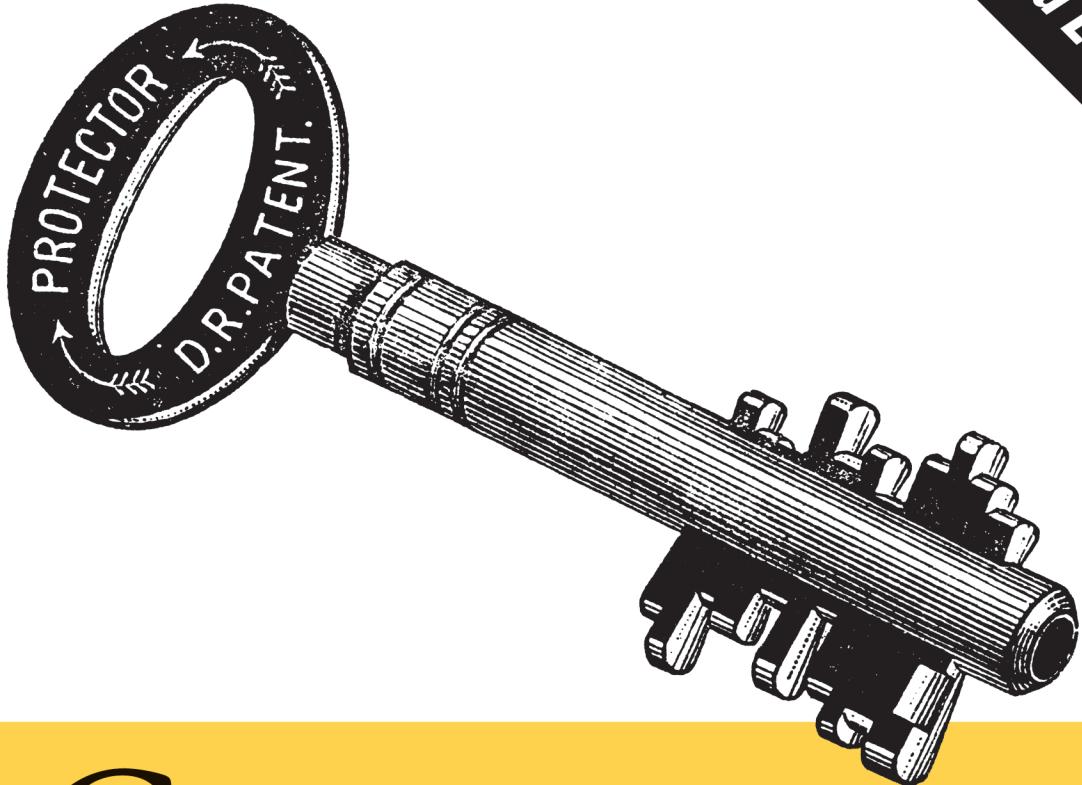


Computer Security for the 21st Century

2nd Edition

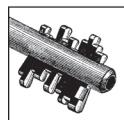


Computer Security Basics

O'REILLY®

*Rick Lehtinen, Deborah Russell
& G.T. Gangemi Sr.*

Computer Security Basics



This is the must-have book for a must-know field. Today, general security knowledge is mandatory, and if you need to understand the fundamentals, *Computer Security Basics* is the book for you.

This new edition builds on the well-established principles developed in the original edition and updates that core knowledge thoroughly. For anyone who is involved with computer security, including security administrators, system administrators, developers, and IT managers, *Computer Security Basics* offers a clear overview of the security concepts you need to know. Topics range from access controls, malicious software, security policy, cryptography, and biometrics, to government regulations and standards.

This handbook describes complicated concepts—such as trusted systems, encryption, and mandatory access control—in simple terms. It tells you what you need to know to understand the basics of computer security, and it will help your employees to practice safe computing.

Topics include:

- Computer security concepts
- Security breaches, such as viruses and other malicious programs
- Access controls
- Security policy
- Web attacks
- Communications and network security
- Encryption
- Physical security and biometrics
- Wireless network security
- Computer security and requirements of the Orange Book
- OSI Model and TEMPEST

“Computer Security Basics is an excellent treatment of a complex subject. [It] is a complete introduction to computer security and is a valuable reference for both security novices and experts.”

—Michael McChesney, Chairman and CEO of M1 Global Solutions, Inc.

oreilly.com

US \$44.99

CAN \$56.99

ISBN: 978-0-596-00669-3



9 780596 006693



Includes
FREE 45-Day
Online Edition

Computer Security Basics

Other computer security resources from O'Reilly

Related titles

- | | |
|--|---|
| Network Security Tools | Network Security with OpenSSL |
| Internet Forensics | Practical Unix and Internet Security |
| Digital Identity | SSH, The Secure Shell: The Definitive Guide |
| Security Warrior | Network Security Assessment |
| Security and Usability | Apache Security |
| Linux Security Cookbook™ | |
| Windows Server 2003 Security Cookbook™ | |

Security Books Resource Center

security.oreilly.com is a complete catalog of O'Reilly's books on security and related technologies, including sample chapters and code examples.



Conferences

oreillynet.com is the essential portal for developers interested in open and emerging technologies, including new platforms, programming languages, and operating systems.

O'Reilly brings diverse innovators together to nurture the ideas that spark revolutionary industries. We specialize in documenting the latest tools and systems, translating the innovator's knowledge into useful skills for those in the trenches. Visit *conferences.oreilly.com* for our upcoming events.



Safari Bookshelf (*safari.oreilly.com*) is the premier online reference library for programmers and IT professionals. Conduct searches across more than 1,000 books. Subscribers can zero in on answers to time-critical questions in a matter of seconds. Read the books on your Bookshelf from cover to cover or simply flip to the page you need. Try it today for free.

SECOND EDITION

Computer Security Basics

*Rick Lehtinen, Deborah Russell,
and G.T. Gangemi Sr.*

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Sebastopol • Taipei • Tokyo

Computer Security Basics, Second Edition

by Rick Lehtinen, Deborah Russell, and G.T. Gangemi Sr.

Copyright © 2006, 1991 O'Reilly Media, Inc. All rights reserved.

Printed in the United States of America.

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (safari.oreilly.com). For more information, contact our corporate/institutional sales department: (800) 998-9938 or corporate@oreilly.com.

Editor: Tatiana Apandi

Developmental Editor: Mary Dageforde

Production Editor: Darren Kelly

Copieditor: Mary Anne Weeks Mayo

Proofreader: Darren Kelly

Indexer: Julie Hawks

Cover Designer: Edie Freedman

Interior Designer: David Futato

Illustrators: Robert Romano and Jessamyn Read

Printing History:

July 1991: First Edition.

June 2006: Second Edition.

Nutshell Handbook, the Nutshell Handbook logo, and the O'Reilly logo are registered trademarks of O'Reilly Media, Inc. *Computer Security Basics*, the image of a key, and related trade dress are trademarks of O'Reilly Media, Inc.

Figure 7-1 is reproduced by permission of the Smithsonian Institution. Figure 10-14 used by permission of Berkeley Varitronics Systems, Inc.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and O'Reilly Media, Inc. was aware of a trademark claim, the designations have been printed in caps or initial caps.

While every precaution has been taken in the preparation of this book, the publisher and authors assume no responsibility for errors or omissions, or for damages resulting from the use of the information contained herein.



This book uses RepKover™, a durable and flexible lay-flat binding.

ISBN: 978-0-596-00669-3

[M]

Table of Contents

Preface	ix
---------------	----

Part I. Security for Today

1. Introduction	3
The New Insecurity	3
What Is Computer Security?	9
Threats to Security	12
Why Buy Security?	18
What's a User to Do?	21
2. Some Security History	22
Information and Its Controls	22
Computer Security: Then and Now	25
Early Computer Security Efforts	27
Building Toward Standardization	32
Computer Security Mandates and Legislation	37

Part II. Computer Security

3. Computer System Security and Access Controls	49
What Makes a System Secure?	49
System Access: Logging into Your System	50

4. Viruses and Other Wildlife	79
Financial Effects of Malicious Programs	79
Viruses and Public Health	80
Viruses, Worms, and Trojans (Oh, My!)	80
Who Writes Viruses?	90
Remedies	92
The Virus Hype	93
An Ounce of Prevention	94
5. Establishing and Maintaining a Security Policy	96
Administrative Security	97
Overall Planning and Administration	98
Day-to-Day Administration	103
Separation of Duties	109
6. Web Attacks and Internet Vulnerabilities	112
About the Internet	112
What Are the Network Protocols?	116
The Fragile Web	124

Part III. Communications Security

7. Encryption	137
Some History	138
What Is Encryption?	141
The Data Encryption Standard	153
Other Cryptographic Algorithms	163
Message Authentication	169
Government Cryptographic Programs	170
Cryptographic Export Restrictions	171
8. Communications and Network Security	173
What Makes Communication Secure?	174
Modems	177
Networks	179
Network Security	187

Part IV. Other Types of Security

9. Physical Security and Biometrics	203
Physical Security	204
Locks and Keys: Old and New	207
Biometrics	212
Gentle Reminder	218
10. Wireless Network Security	220
How We Got Here	220
Today's Wireless Infrastructure	221
How Wireless Works	225
Playing the Fields	228
What Is This dB Stuff?	231
Why Does All This Matter?	232
Encouraging Diversity	233
Physical Layer Wireless Attacks	233

Part V. Appendixes

A. OSI Model	249
B. TEMPEST	252
C. The Orange Book, FIPS PUBS, and the Common Criteria	258
Index	283

Preface

About This Book

This book is about computer security—what it is, where it came from, where it's going, and why we should care about it. It introduces the many different areas of security in clear and simple terms: access controls, worms and viruses, cryptography, firewalls, network and web security, biometric devices, and more. If you're at all interested in computer security or if computer security is a part of your job (whether you want it to be or not!), you should find this book useful. I've tried to give you the big picture and quite a few helpful details.

This book is not a technical reference. I've tried to pull together the basics about many different areas of computer security and put that information together comprehensively. If you need particularly technical information about a specific area of computer security (for example, making your specific system or operating system more secure, securing your web site, or configuring a router or firewall), you should refer to other, more specialized books.

Summary of Contents

This book is divided into 10 chapters and 3 appendixes.

Part I, Security for Today

This section presents a brief overview of what computer security is, where it came from, and where it's going.

Chapter 1, *Introduction*

This chapter introduces computer security: what it is and why it's important. It summarizes the threats to computers and the information stored on them, and it introduces the different types of computer security.

Chapter 2, Some Security History

This chapter briefly describes the history of computer security: where it came from, and what government mandates, laws, and standards address it.

Part II, Computer Security

This section discusses computer security methods of protecting information stored in a computer system, primarily by controlling access to that information.

Chapter 3, Computer System Security and Access Controls

This chapter introduces computer system security and describes how that security controls access to systems and data.

Chapter 4, Viruses and Other Wildlife

This chapter explores viruses, worms, Trojans, and other types of malicious code.

Chapter 5, Establishing and Maintaining a Security Policy

This chapter describes the administrative procedures that improve security within an organization. It also introduces business continuity and disaster recovery as part of security.

Chapter 6, Web Attacks and Internet Vulnerabilities

This chapter introduces the perils that can attack your system or network while it is connected to the Internet.

Part III, Communications Security

This section discusses communications security methods of protecting information while it's being transmitted over communications lines and network backbones.

Chapter 7, Encryption

This chapter explains what encryption is and how it protects data.

Chapter 8, Communications and Network Security

This chapter introduces network concepts and discusses some basic communications security issues.

Part IV, Other Types of Security

This section describes several additional types of security.

Chapter 9, Physical Security and Biometrics

This chapter introduces physical security and describes different types of biometric devices.

Chapter 10, Wireless Network Security

This chapter describes the workings of wireless networks and the security ramifications of this access medium.

Part V, Appendixes

This section provides a number of quick references to computer security requirements and programs.

Appendix A, *OSI Model*

This appendix describes the seven layers of OSI and how each relates to security.

Appendix B, *TEMPEST*

This appendix describes what TEMPEST is and why it's important.

Appendix C, *The Orange Book, FIPS PUBS, and the Common Criteria*

This appendix provides a summary of legacy Orange Book requirements, the Federal Information Processing Publications (FIPS PUBS), and the Common Criteria, which is the international successor to the Orange Book.

Using Code Examples

This book is here to help you get your job done. In general, you may use the code in this book in your programs and documentation. You do not need to contact O'Reilly for permission unless you're reproducing a significant portion of the code. For example, writing a program that uses several chunks of code from this book does not require permission. Selling or distributing a CD-ROM of examples from O'Reilly books *does* require permission. Answering a question by citing this book and quoting example code does not require permission. Incorporating a significant amount of example code from this book into your product's documentation *does* require permission.

We appreciate, but do not require, attribution. An attribution usually includes the title, author, publisher, and ISBN. For example: *Computer Security Basics*, Second Edition, by Rick Lehtinen, Deborah Russell, and G.T. Gangemi Sr. Copyright 2006 O'Reilly Media, Inc., 0-596-00669-1.

Comments and Questions

Please address comments and questions concerning this book to the publisher:

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472
800-998-9938 (in the United States or Canada)
707-829-0515 (international or local)
707-829-0104 (fax)

We have a web page for this book that lists errata, examples, and any additional information. You can access this page at:

<http://www.oreilly.com/catalog/csb2>

To comment or ask technical questions about this book, send email to:

bookquestions@oreilly.com

For more information about our books, conferences, Resource Centers, and the O'Reilly Network, see our web site at:

<http://www.oreilly.com>

Safari® Enabled



When you see a Safari® Enabled icon on the cover of your favorite technology book, that means the book is available online through the O'Reilly Network Safari Bookshelf.

Safari offers a solution that's better than e-books. It's a virtual library that lets you easily search thousands of top tech books, cut and paste code samples, download chapters, and find quick answers when you need the most accurate, current information. Try it for free at *<http://safari.oreilly.com>*.

Acknowledgments

Any author of a second edition is incredibly indebted to the authors of the first. Deborah Russell, you were great to work with from start to finish. The text authored by you and G.T. Gangemi Sr. gave me an excellent starting point. Let's do it again some day. And to my buds at Cisco's Network Academy and to Gwen, my CISSP coach, who put me up to this—warm thanks, I owe you.

Thanks to Kathy, Jana, Jon, Kyle and a few cats who did not see enough of me during this book's production, and Louise who said I shouldn't undertake it, but was kind of proud once it got done.

Many people from O'Reilly helped to produce this second edition: Deborah Russell for seeing the value in the project and Tatiana Apandi for keeping me at it and cheering me on when it conflicted with my day job. Thanks to my technical reviewers: Mark Lucking, Simon Biles, and especially Mary Dageforde for all her help. Many thanks!

Some of the chapters in the first edition were based on an internal document that Deborah Russell prepared for Wang Laboratories. Thanks to a lot of other people who contributed to the first edition of this book: Dennis K. Branstad, James Burrows, Daniel Faigin, Perry Flinn, Simson Garfinkel, Irene Gilbert, Nick Hammond, Stuart W. Katzke, F. Lynn McNulty, Paul Mei, Andrew Odlyzko, Victor Oppenheimer, Tim O'Reilly, Robert Rosenthal, Bradley Ross, Len Schneider, Miles Smid, Gene Spafford, Bob Tinkelman, Gene Troy, and Mitch Wright.

PART I

Security for Today

CHAPTER 1

Introduction

The New Insecurity

Since the terrorist attacks on September 11, 2001, computer security has taken on some new meanings. The first is positive. As part of a global tightening of belts and rolling up of sleeves, there emerged several outreaches designed to provide security training and certification to folks in all walks of life, from the consumer being alerted about identity theft, to the soldier and sailor and weapons scientists taking greater precautions with items of national security, to the common person on the street gaining a heightened awareness of hackers and crackers and cyber attackers. Gradually this new emphasis on computer and network safety has percolated down to the ordinary user's computer in the den or living room. And because it really is a small Internet, and what affects one usually affects all, the safer individual users are, the safer the Net is for everybody.

Unfortunately, in return for a perception of security, both physical and on the Internet, some computer users have begun to accept unprecedented compromises in privacy as being part of the price to be paid to counter an envisioned terrorist threat associated with computer usage. In return for a feeling of "protection" with vague ties to national defense, more and more of what used to be private data and folks' own business is now available for inspection by corporate and legal observers. Giving up the proven checks and balances that are the underpinnings of a free society may do more harm than good. Recent reports, such as a summer 2003 incident in which one or more airlines turned over to a contract firm working for the Department of Defense the transaction records of a half million passengers for use in an experiment on database profiling, have demonstrated that relaxed restraints against law enforcement agencies can lead to egregious actions. Numerous press reports have indicated that the expanded powers granted to law enforcement agencies in the name of homeland defense have resulted in those powers being used increasingly to investigate and prosecute crimes under laws not related to homeland defense at all. This, in turn, has resulted in a mini-backlash designed to rein in the security promoters, heightening the debate.

Possibly in response to a perceived decrease in privacy, a large number of new laws have come into play that attempt to protect individuals against widespread dissemination of personal information and regulate the creation and exchange of financial information regarding corporations. These new laws have long names, such as the Health Insurance Portability and Accountability Act (HIPAA), Sarbanes-Oxley, and the Family Educational Rights and Privacy Act (FERPA). These laws make it a crime to reveal personal information gathered in the course of doing business, and often require the reporting of computer crimes that were formerly swept under the carpet to avoid embarrassing the agency or company allowing such a lapse.

The ordinary user, such as the salesperson or secretary who logs on in the morning and shuts down at night, would rather not think twice about security. In fact, she might not think of it at all until a worm or some other attack affects the machine on which she has to work.

Some of the most invasive computer attacks against individuals may not involve infecting a computer, but merely listening to one. With machine patience, sniffers and database programs can accumulate data about people—lots of people—over as long a time as is needed to gather enough information to make an attack. Usually, the attack takes the form of making credit card purchases, or applying for credit in the name of the victims whose details have been pieced together. Such crimes, often called *identity theft*, can be devastating. It is not that the victim is always left liable for the fraudulent purchases; consumer protection laws and the rapid closing of accounts help a great deal to prevent that. It is that the victim may be left unable to exercise his own credit, or establish more because vendors can't easily be sure if any new transactions after the ID theft is reported are being made by the customer or by the thief. And it is highly likely that the victim will be unaware of any of these activities until the damage has been done.

Now that it increasingly impacts the average user, public awareness of computer security has risen dramatically. Computer security has hit the newsstands, with more and more articles warning the public about viruses and other perils. The media also describes an increasing array of preventatives, ranging from changing network habits to adding firewalls and intrusion protection systems. Mix in the specter of terrorism, and the stakes get even higher.

Who You Gonna Call?

A new generation of security consultants—what *Business Week* once termed “hacker-busters”—have hung out their shingles. A number of organizations stand ready to provide expert assistance in case a computer virus outbreak threatens the Internet:

- Funded by the Defense Advanced Research Projects Agency (DARPA), the Computer Emergency Response Team (CERT) at the Software Engineering Institute at Carnegie Mellon University was created to provide information and support against any Internet crises, cyber attacks, accidents, or failures. Now officially

named the CERT Coordination Center, this clearinghouse is the mother-of-all-CERTs, and regional and corporate incident response centers are springing up to handle crises locally.

- The Federal Computer Incident Response Center (FedCIRC) is the federal government's trusted focal point for computer security incident reporting, providing assistance with incident prevention and response. In 2003, the FedCIRC officially became part of the Department of Homeland Security's Information Analysis and Infrastructure Protection (IAIP) Directorate. IAIP will continue to provide the FedCIRC services.
- The Department of Energy has also established a Computer Incident Advisory Capability (CIAC) oriented to its own agency needs, including a "hoaxbusters" page dedicated to helping users recognize which attacks are real and which are based on hysteria. The gentle gags clog up networks as users frantically alert their friends and neighbors of the supposed hazard. The vicious gags encourage users to take "protective measures" that might actually damage their own computers in an attempt to avoid worse calamity.
- US-CERT is a partnership between CERT and the U.S. Department of Homeland Security.

Other national incident response teams have been formed in many countries:

- In the United Kingdom, there is the National Infrastructure Security Co-ordination Centre (NISCC), pronounced "nicey", which is charged with protecting essential system and services known collectively as the Critical National Infrastructure (CNI).
- AusCERT (Australian CERT) monitors and evaluates global computer network threats and vulnerabilities.
- CanCERT is Canada's first national Computer Emergency Response Team.
- CERT Polska deals with security-related incidents related to Polish networks.
- SingCERT (Singapore CERT) serves Singapore and parts of Southeast Asia.
- SI-CERT is the Slovenian Computer Emergency Response Team, a service offered by ARNES (Academic and Research Network of Slovenia).

In addition to government response organizations, many commercial providers of security services and virus protection systems have also set up organizations that are prepared to come to the aid of any customers who find security holes or face attacks.

- OXCERT provides CERT services for Oxford University in the United Kingdom.
- Linux and Unix users have ample organizations that report new exploits and post cures for easy update.

Information Sharing and Analysis Centers

Akin to CERTs, Information Sharing and Analysis Centers (ISACs) help develop and promulgate "best practices" for protecting critical infrastructures and minimizing

vulnerabilities. Many industries have established ISACs to allow these critical sectors to share information and work together to help better protect the economy.

In the United States, Presidential Directive Number 63 and the Patriot Act establish that the ISACs will receive governmental sponsorship. The Department of Homeland Security lists links to various industry ISACs on its web site. ISACs are established for the food industry, water industry, emergency services (police and fire), state governments, and the telecommunications and information technology industries. There are also ISACs in place for the energy, transportation, banking and finance, chemical, and real estate industries.

Vulnerable broadband

Just as corporate and government users are bonding together to provide mutual protection, however, a huge emerging class of users is expanding rapidly, and for the most part they are unprotected. As broadband Internet access becomes increasingly popular, more users set up home computers and leave them running 24/7. The result is they become targets for attackers.

One study estimated that the time between when a new computer is turned on and the first attack is underway is usually less than 10 minutes. This is because attackers often use automated scanning tools that probe constantly, looking for opportunity. An exploit can often be placed in seconds, often before countermeasures can be installed to complete an installation. Other studies claim the situation is worse still, figuring the time before attack is equal to 2 minutes. I've seen instances in which newly updated computers became infected by a virus within a few minutes, even though the computers were protected by a secure network. This happened because the infecting computers were inside the network, likely becoming infested by pathogens carried in on media workers brought from home.

As the pool of computer users has increased, ways are emerging to illicitly profit off of them. The computer of a naive user may be forced into participating in a distributed denial of service (DDoS) attack aimed toward a designated target and timed to fire off with hundreds of thousands of others so as to overwhelm the victim. Alternatively, users' broadband computers can be turned into unwilling web sites for pornography or other products, or made into relays for unsolicited email (spam).

Fortunately, help is on the way:

- Microsoft, for instance, offers easy software security updates over the Internet.
- Help sites are available for every kind of Linux and Unix.
- Many antivirus software publishers offer not only antivirus programs but also some kind of information service documenting viruses and what to do to prevent or handle specific attacks.
- Most companies today are adding their own internal security forces. Increasingly, corporate want ads request a computer security certificate or two as a prerequisite for hiring.

No computer is an island

While once it was easy to ignore most warnings and scares as mere nuisances because most sites were isolated and unconnected, in today's world, few computers stand alone. Viruses occur and spread with amazing speed, sometimes spanning the globe in hours or days (usually by stealing information, such as an email address book from one victim, and using it to infect others).

Even corporations that have secure perimeters can find themselves with significant internal virus problems. Often this is due to users who bring in infected laptops, use removable data drives, or burn information onto recordable CDs or DVDs that are infected and then brought into the office network.

The Sorry Trail

The story of network attacks, bugs, viruses, and criminal actions stretches as far as the computer industry itself. One of the first bugs to develop in a computer system was precisely that: a moth was found squished inside some relay contacts at a government installation. Lieutenant Grace Hopper collected that moth and duly pasted it into the facility logbook. She eventually became a rear admiral, and went on to invent the computer compiler and was the driving force behind the COBOL computer language.

With each advance of technology came new threats and attacks. Rogue self-replicating programs nearly overwhelmed a research facility in Palo Alto, California; they were the first computer worms. Unchecked, worms can multiply until they fill up a hard disk. Viruses, similar to worms but requiring a host program of some kind to live in and take over, came soon after. Attacks and countermeasures followed one after another until the present. Vulnerabilities continue to be sniffed out by attackers who create viruses and worms to exploit them. Manufacturers then create patches intended to counter the attacks.

The whole adventure of viruses and worms can all be summed up in the term *malicious software* or *malware*. Malware will be covered in some detail in later chapters.

While early malware exploited single systems or multiuser systems, it took the Internet to really give malware life. The Internet forms a massive distributed environment. Malicious software can steal control of computers on the Internet, direct DDoS attacks at given hosts or servers, or pose as someone they are not in order to intercept data. The latter action is known as a *masquerade attack* or *spoofing*.

The most elaborate malware can scan a victim machine for links to other machines, then replicate itself to those other machines while working its attack on the victim machine. The infamous Code Red worm worked over the Internet in this way. After replicating itself for the first 20 days of each month, it replaced web pages on the victim machines with a page that declared "Hacked by Chinese," then launched an attack on the White House web server.

Computer crime

Computer crime has also become a major threat to business. According to the Federal Bureau of Investigation, computer crime is the most expensive form of commercial crime. In 2003, theft of information cost over \$70 million, with an average cost of \$2.6 million per theft. Also in 2003, denial of service attacks, which deprived companies of revenue and idled IT investments, cost over \$66 million, with an average loss of \$1.4 million. Estimates of the dollar figure for theft by computer intrusion and attack total \$201 million.



Although almost 75 percent of organizations reported some kind of attack in 2003, only about 40 percent of those attacked could quantify the loss. It is estimated that roughly 50 percent of intrusions were not reported at all, either because their scope was unknown or the publicity was undesired.

Even though there has been substantial publicity in recent years about computer system risks and attacks, it turns out that many organizations are unwilling to report system intrusions. Doing so can result in adverse publicity, the loss of public confidence, and the possible charge of managerial incompetence. Many organizations fear lawsuits based on the emerging “standard of due care.”

In fact, there are reports that in the days before regulations such as Sarbanes-Oxley, which requires increased justification of the figures used in business accounting, some businesses paid hush money to intruders. In London, a number of firms have reportedly signed agreements with computer criminals offering them amnesty for returning part of the money stolen and, more importantly, for keeping quiet about their thefts. In one case, an assistant programmer at a merchant bank diverted eight million pounds to a Swiss account. In an agreement that protected him from prosecution, the programmer promised not to disclose the system penetration—and he got to keep one million pounds!

Recent statistics indicate that payment of hush money is decreasing, often due to increasingly automated nature of the attacks. Most attacks today are run by unsophisticated youth who learn a few tricks and gather a few scripts from true gurus, and then do what amounts to vandalism for the thrill of it. However, the thrill of penetration and creating havoc is increasingly offset by the penalties. The legal fate of some big time virus writers has been widely reported on TV and in the newspapers. Some murderers and rapists have gotten away with lighter sentences.

More recently, skillful intruders are attacking computers with criminal or military goals in mind. These attackers may outwit even sophisticated security systems, and can leave dormant sleeper programs that will lay low to avoid detection until their owners summon them to action.

What Is Computer Security?

The term *computer security* has different interpretations based on what era the term describes. Early on, computer security specialized in keeping the glass houses in which the computer core was positioned safe from vandalism, along with providing constant cooling and electricity. As computers became more dispersed, security became more of an issue of preserving data and protecting its validity, as well as keeping the secrets secret. As computers moved onto the desktop and into the home, computer security took the form of protection against data thieves and network attackers. Modern computer security includes considerations of *business continuity*. This ability mitigates interruption or loss regardless of the threat, and more importantly, develops rational systems that estimate and offset risk. These values are incorporated into procedures and policies that make computer security a priority from the top down. Today, industrial security, in terms of loss control due to theft, vandalism, and espionage, involves the same personnel controls and physical security provisions that protect the enterprise as a whole.

You can get a good thumbnail sketch of computer and network security by examining the principles on which it is founded. Computer and network security are built on three pillars, commonly referred to by the C-I-A acronym:

- Confidentiality
- Integrity
- Availability

Data is confidential if it stays obscure to all but those authorized to use it. Data has integrity as long as it remains identical to its state when the last authorized user finished with it. Data is available when it is accessible by authorized users in a convenient format and within a reasonable time. (Note: the C-I-A acronym will be repeated like a mantra throughout the course of this book.)

Following shortly on the heels of C-I-A are a host of other terms and acronyms. Each of these has its own shade of meaning, but all of them are part of the C-I-A model:

Identification

Who do you say you are?

Authentication

How do I know it's really you?

Authorization

Now that you are here, what are you allowed to do?

Accountability

Who did what, and, perhaps, who pays the bill?

Different groups emphasize different combinations. To “simple is best” administrators, a favored authentication would likely be the *username* (who you say you are)

and *password* (prove it to me!) combination. Devotees of *biometric security* identification, on the other hand, who use some physical attribute as a means of identification, point with pride to the fact that a retina scan can identify and authenticate simply by taking a picture of the blood vessels in the back of someone's eye. (The crack to this system was demonstrated by actor Tom Cruise in the film *Minority Report*. It lent a whole new meaning to the phrase "He's got his father's eyes.") Other groups promote acronyms within acronyms. For example, "authentication, authorization, and accounting" (AAA) is Cisco shorthand meaning that user verification and rights determination can be accomplished in the same process as transaction record keeping, or *audit logging*.

Computer security and network security are part of a larger undertaking that protects your computer and everything associated with it—your building, your terminals and printers, your cabling, and your disks and tapes. Most importantly, computer security protects the information you've stored in your system. That's why computer security is often called *information security*.

The International Information Systems Security Certification Consortium, or (ISC)², encompasses the following 10 domains in its common body of knowledge:

- Information security management
- Access control systems and methodology
- Cryptography
- Physical security
- Enterprise security architecture
- Application security
- Telecommunications, network, and internet security
- Law, investigation, and ethics
- Business continuity planning
- Operations security

Each domain includes five functional areas:

- Information protection requirements
- Information protections environment
- Security technology and tools
- Assurance, trust, and confidence mechanisms
- Information protection and management services

A Broader Definition of Security

The popular conception of computer security is that its only goal is secrecy, such as keeping the names of secret agents from falling into the hands of the enemy, or keeping a nationwide fast food chain's new advertising strategy from being revealed to a

competitor. Secrecy is a very important aspect of computer security, but it's not the whole story.

In some systems or application environments, one aspect of security may be more important than others. Your own assessment of what type of security your organization requires will influence your choice of the particular security techniques and products needed to meet those requirements.

Secrecy and Confidentiality

A secure computer system must not allow information to be disclosed to anyone who is not authorized to access it. For example, in highly secure government systems, *secrecy* ensures that users access only information that they are allowed, by the nature of their security clearances, to access. Similarly, in business environments, *confidentiality* ensures the protection of private information (such as payroll data) as well as sensitive corporate data (such as internal memos and competitive strategy documents).

Of course, secrecy is of paramount importance in protecting national defense information and highly proprietary business information. In such environments, other aspects of security (e.g., integrity and availability), while important, may be less critical. Chapter 3 discusses several major methods of enforcing secrecy or confidentiality in your system, including controlling who gets access and specifying what individual users are able to do. Chapter 7 discusses encryption, another excellent way to keep information a secret.

Accuracy, Integrity, and Authenticity

A secure computer system must maintain the continuing integrity of the information stored in it. *Accuracy* or *integrity* means that the system must not corrupt the information or allow any unauthorized malicious or accidental changes to it. It wasn't deliberate, but when a simple software error changed entries in Bank of New York transactions many years ago, the bank had to borrow \$24 billion to cover its accounts until things got straightened out—and the mistake cost \$5 million in extra interest.

In network communications, a related variant of accuracy known as *authenticity* provides a way to verify the origin of data by determining who entered or sent it, and by recording when it was sent and received.

In financial environments, accuracy is usually the most important aspect of security. In banking, for example, the confidentiality of funds transfers and other financial transactions is usually less important than the verifiable accuracy of these transactions. Chapter 7 discusses *message authentication*, a method that ensures the accuracy of a transmission. With this method, a code is calculated and appended to a message when that message is sent across a network. At the receiving end, the code is

calculated again. If the two codes are identical, the message sent is the same as the message received—proof that it wasn’t forged or modified during transmission.

Availability

A secure computer system must keep information available to its users. *Availability* means that the computer system’s hardware and software keeps working efficiently and that the system is able to recover quickly and completely if a disaster occurs.

The opposite of availability is *denial of service*, or *DoS*. Denial of service means system users are unable to get the resources they need. The computer may have crashed. There may not be enough memory or processes to run a program. Needed disks, tapes, or printers may not be available. DoS attacks can be every bit as disruptive as actual information theft, attacking system availability by spreading through networks, creating new processes, and effectively blocking all other work on the infected computers.

In some ways, availability is a baseline security need for everyone. If you can’t use your computer, you won’t be able to tell whether your secrecy and accuracy goals are being met. Even users who abhor “security” agree that their computer systems have to keep working. Many of them don’t realize that keeping systems running is also a type of security.

Chapters 5 and 6 discuss two important ways to ensure the availability of a network system: careful system administration and sound system design.

Threats to Security

There are three key words that come up in discussions of computer security issues: vulnerabilities, threats, and countermeasures. A *vulnerability* is a point where a system is susceptible to attack. A *threat* is a possible danger to the system. The danger might be a person (a system cracker or a spy), a thing (a faulty piece of equipment), or an event (a fire or a flood) that might exploit a vulnerability of the system. The more vulnerability you see in your system, and the more threats you believe are out there, the more carefully you’ll need to consider how to protect your system and its information. Techniques for protecting your system are called *countermeasures*.

Computer security is concerned with identifying vulnerabilities in systems and protecting against threats to those systems.

Vulnerabilities

Every computer and network is vulnerable to attack. Security policies and products may reduce the likelihood that an attack will actually be able to penetrate your system’s defenses, or they may require an intruder to invest so much time and so many resources that it’s just not worth it—but there’s no such thing as a completely secure system.

The following sections demonstrate the typical points of vulnerability in a computer system.

Physical vulnerabilities

Your buildings and equipment rooms are vulnerable. Intruders can break into your server room, just as they can break into your home. Once in, they can sabotage and vandalize your network equipment, and they can steal backup media and printouts, or obtain information that will allow them to more easily hack their way in at a later time.

Locks, guards, and biometric devices (devices that test a physical or behavioral trait—for example, a fingerprint, a voiceprint, or a signature—and compare it with the traits on file to determine whether you are who you claim to be) provide an important first defense against break-ins. Burglar alarms and other ordinary types of protection are also effective deterrents.

Natural vulnerabilities

Computers are very vulnerable to natural disasters and to environmental threats. Disasters such as fire, flood, earthquakes, lightning, and power loss can wreck your computer and destroy your data. Dust, humidity, and uneven temperature conditions can also do damage.

In areas where obtaining stable power is a problem, facilities employ back-up generators. These can also help during times of extreme weather. Localized protection can be obtained through installing an uninterruptible power supply (UPS). A properly sized UPS will keep a computer energized long enough to shut down properly and without data loss, and provide power conditioning as well. Dust and other hazards are usually controlled by proper filters on the air conditioning and heating systems. If the environment itself tends to be dusty, a simple cloth cover can protect the computer when not in use. Do not cover a computer while it is operating, however, to avoid blocking the internal cooling fans and let the case radiate excess heat. Even temperature will help eliminate some problems, as well. The components and cards in a computer may expand and contract at different rates; they can become loose in their sockets. Avoid dampness in areas where removable media, such as floppy disks, CDs, DVDs, and backup tapes, are stored; mold and fungus are lethal to some media.

Hardware and software vulnerabilities

Certain kinds of hardware failures can compromise the security of an entire computer system. If protection features fail, they wreak havoc with your system, and they open security holes. It is also possible to open some “locked” systems by introducing extra hardware, or to use external devices to make a copy of the contents of disks or memory.

Software failures of any kind may cause your system to fail, open your system to penetration, or simply make the system so unreliable that it can't be trusted to work properly and efficiently. Thriving exploration into vulnerabilities by the hacking

community means that exploits will be published in online forums, paving the way for those who wish to write and publish viruses or other malicious software to do so. In particular, bugs in security features can open the floodgates to intrusion.

Even if individual hardware and software components are secure, an entire system can be compromised if the hardware components are connected improperly or if the software isn't installed correctly.

Media vulnerabilities

Backup media, such as disk packs, tape reels, cartridges, and printouts, can be stolen, or can be damaged by such mundane perils as dust and stray magnetic and electromagnetic fields. Most hard-drive erase operations involve rewriting header files, not actually erasing the entire disk, so sensitive data may be left on magnetic media, easily decoded after a computer is retired or discarded. Even the memory chips in some electronic devices can be scanned for remnants of data or files.



Media is useful only if it is usable. As mentioned previously, keep backup tapes and removable disks clean and dry.

Emanation vulnerabilities

All electronic equipment emits electrical and electromagnetic radiation. Electronic eavesdroppers can intercept the signals emanating from computers, networks, and wireless systems, and decipher them. The information stored and transmitted by the systems and networks then becomes vulnerable.

Communications vulnerabilities

If your computer is attached to a network or if it can be accessed by a dial-in modem or over the Internet, you greatly increase the risk that someone will penetrate your system. Messages can be intercepted, misrouted, and forged. Communications lines connecting computers to each other, or connecting terminals to a central computer, can be tapped or physically damaged. Radio transmissions, the basis of wireless interconnections such as IEEE 802.11 (Wi-Fi) or IEEE 802.15 (Bluetooth), are particularly susceptible to surreptitious interception.

Human vulnerabilities

The people who administer and use your computer system represent the greatest vulnerability of all. If your administrator is poorly trained, or decides to take to a life of crime, your network is in grave peril. Ordinary computer users, operators, and other people on your staff can also be bribed or coerced into giving away passwords, opening doors, or otherwise jeopardizing security in your system.

Exploiting vulnerabilities

There's a lot of variation in how easy it is to exploit different types of vulnerabilities. For example, tapping a wireless network can require nothing more than special software installed on a laptop. Logging into a system that has no password protection, minimal controls, or inadequate password policies (e.g., allowing users to leave passwords on sticky notes at their workstations) is almost as easy. Tapping an encrypted fiber-optic communications link, on the other hand, or intercepting emanations from TEMPEST-shielded equipment is much more difficult, even for a dedicated intelligence operation. (See Appendix B for more information on TEMPEST.)

Threats

Threats fall into three main categories: natural, unintentional, and intentional.

Natural and physical threats

These threats imperil every physical plant and piece of equipment: fires, floods, power failures, and other disasters. You can't always prevent such disasters, but you can find out quickly when one occurs (with fire alarms, temperature gauges, and surge protectors). You can minimize the chance that the damage will be severe (e.g., with certain types of sprinkler systems). You can institute policies that guard against hazards posing special dangers to computers (such as smoking or soda spills). You can also plan for a disaster by backing up critical data off-site and by arranging for the use of a backup system that can be used if an emergency does occur.

Unintentional threats

Ignorance creates dangers: for example, a user or a system administrator who hasn't been trained properly, who hasn't read the documentation, and who doesn't understand the importance of following proper security procedures. A user might inadvertently delete a file, or a system administrator might change the protection on the password file or on critical system software, locking out programs and applications that need to access that data. Generally, more information is compromised, corrupted, or lost through ignorance than through malice.

Intentional threats

These villains come in two varieties: outsiders and insiders. Some types of attacks are feasible only for certain types of attackers. For example, a casual "browser" isn't likely to intercept and decipher electromagnetic emanations, or to perform a determined cryptographic analysis. Attacks of those kinds can typically be mounted only by sophisticated attackers who have substantial skill and resources (in computing power, money, time, and personnel) behind them.

Outsiders include a number of different categories:

Foreign intelligence agents

They're not lurking behind every bush, but they really do exist! Products using sophisticated encryption devices are most appropriate at installations where attacks on classified information are a realistic threat.

Terrorists

Luckily, we haven't seen too much computer terrorism yet, though there have been attacks on university computers, various DoD networks and web sites, court buildings, and the like. The government worries about computer terrorism. So do airlines, oil companies, and other businesses that protect information that's vital to the national interest. While some experts repeatedly predict that an "electronic Pearl Harbor" is imminent, others feel that computer terrorism, if it ever occurs, will just be a diversion, augmenting any terrorist attack by slowing down the communications needed to respond to the attack.

That said, there is evidence that some nations increasingly engage in routine interruption of communications within other nations, apparently with the intention of advancing political agendas. Mirroring offline diplomatic clashes, Internet users in Japan, China, and Korea have reportedly launched cyber attacks against each other. Information can be beamed into countries that suppress it. Denial of service attacks can be launched against government and company web sites. Often these attacks coincide with national holidays or protests.

Criminals

Computer crime is lucrative, and, unlike many other types of crimes, can be carried out in a tidy, anonymous electronic fashion. The goal may be outright theft or embezzlement, or it may be extortion of some kind; for example, "I have just encrypted an important hard drive that is part of your important database. If you don't pay me, I will throw away the key and tell the world that you could not keep your corporate data secure."

Corporate raiders

Corporations rely on computers, network connections, and electronic mail. Corporate records, memos, and informal messages have become more vulnerable than ever to attacks by competitors intent on ferreting out weaknesses and plans.

Crackers

This category consists of "computer joy riders." When people talk about crackers, or hackers,* they usually mean intruders who are more interested in the challenge of breaking in than in the spoils of victory. These intruders may browse

* The word "hacker" has a long and honorable history. It originally meant anyone with a strong interest in computers and an eagerness to experiment with them and test their limits. More recently, the word has been used to refer to those who break into systems in an unlawful way. Because many law-abiding "hackers" object to this pejorative meaning of the word, I've chosen to call those who deliberately break into systems "attackers" or "intruders," rather than "hackers."

through systems, peeking at interesting data and programs, but they usually don't do it for monetary or political gain. More typically, they break into systems for the challenge of defeating each new security feature they encounter. They may share their knowledge with other crackers via electronic bulletin boards, newsgroups, IRC channels, and web logs (blogs). They may also document their successes in hardcopy or electronic publications such as *2600 Magazine*, *Phrack*, and the *Computer Underground Digest*.

Insiders and outsiders

Outsiders may penetrate systems in a variety of ways: simple break-ins of buildings and computer rooms; disguised entry as maintenance personnel; anonymous, electronic entry through modems and network connections; and bribery or coercion of inside personnel.

Although most security mechanisms protect best against outside intruders, survey after survey indicates that most attacks are by insiders. Estimates are that as many as 80 percent of system penetrations are by fully authorized users who abuse their access privileges to perform unauthorized functions. As Robert H. Courtney Jr. put it, "The enemy is already in—we hired them."

There are a number of different types of insiders. The fired or disgruntled employee might be trying to steal; more likely, he's just trying to wreak revenge by disrupting office operations. The coerced employee might have been blackmailed or bribed by foreign or corporate enemy agents. The greedy employee might use her inside knowledge to divert corporate or customer funds for personal benefit. The insider might be an operator, a systems programmer, or even a casual user who is willing to share a password.

Don't forget, one of the most dangerous insiders may simply be lazy or untrained. He doesn't bother changing passwords, doesn't learn how to encrypt email messages and other files, leaves sensitive printouts in piles on desks and floors, and ignores the paper shredder when disposing of documents. More energetic types may take advantage of this laziness and do serious damage.

Often, the most effective system attacks are those that combine a strong outside strategy (for example, breaking into competitors' files to steal their marketing plans) with access by an insider (for example, a marketing assistant who's been bribed to give away a password or steal reports).

Countermeasures

There are many different types of countermeasures—methods of protecting computers and information. This book provides a survey of these methods in several basic categories.

Computer security

Earlier in this chapter, the term “computer security” was used in a broad sense to cover the protection of computers and everything associated with them. It’s more precise to say that computer security is the protection of the information stored in a computer system, as opposed to protecting information that’s being transmitted (which is network security), or protecting the equipment and the facility itself (physical security). Computer security focuses on operating system features that control who can access a system and the data stored in it.

Part II of this book discusses computer security controls—including passwords, auditing of security actions, and administrative procedures such as backups that protect stored data. That part also draws distinctions between the various types of security policies available in multiuser computer systems: discretionary access control (DAC), mandatory access control (MAC), and role-based access control (RBAC). Part II also briefly discusses how the government sets standards for computer security and certifies products that meet those standards. Additional coverage is included in Appendix C.

Communications security

Communications security is the protection of information while it’s being transmitted by telephone, cabling, microwave, satellite, or any other means. This branch of security focuses on network access to computer systems, and the technologies that increase the security of systems allowing such connections to the outside world.

Part III of this book describes encryption, a highly effective method of protecting data either in storage or during transmission. Encryption takes many forms and is applied in several different ways. Part III also discusses a number of other ways to increase network security.

Physical security

Physical security is the protection of physical computer equipment from damage by natural disasters and intruders. Physical security methods include old-fashioned locks and keys, as well as more advanced technologies such as smart cards and biometric devices. Part IV of this book discusses physical security and wireless security (steps that can be taken to protect information during wireless transmission).

Why Buy Security?

Computer security has historically been viewed as being an unnecessary impediment to getting work done. With pressure from the government, the courts, and the press, security now seems to have graduated to a necessary evil. In the latest versions of the Linux and the Windows operating systems, security is automated and is becoming a full-fledged system feature.

Estimates of the size of the computer security market vary. The Freedonia group estimates that computer security now represents roughly a \$9 billion a year market opportunity for the United States, and this number is expected to increase dramatically over the next decade. As you'd expect, the U.S. government drives much of the security market. Because of its special concern for classified information relating to national defense and intelligence, the U.S. government has historically been the major force behind security research and technology. The government has a great many secrets (millions of new pieces of information are classified each year!), and computer security products thrive on secrecy.

It's difficult to get hard numbers on government security spending because military and other classified programs account for a large piece of the security market, and dollar figures for those classified programs aren't publicly available. Best estimates are that as much as half the total computer security purchases are government-related, but this ratio ebbs and flows as security concerns enter and leave the public mind.

The Department of Defense, the intelligence agencies, and government contractors are particularly heavy users of security products—especially cryptographic products, highly secure computer systems, and systems that use TEMPEST technology. (The TEMPEST market is almost exclusively a government one, although the private sector is likely to wake up to it as decoding systems decrease in complexity and increase in availability.) Virtually every government department and agency buys security products. Most of them have little choice; they're required by government regulations to protect the information they process. (Not to say that the security of every government entity is exemplary; numerous “report cards” frequently implicate government watchdogs as being somewhat behind the security curve.)

Businesses and government agencies have different goals and different cost/risk tradeoffs. Because financial considerations are the focus of business, security has to be cost-effective, or business won't buy or build it. A big market question is who will be willing to pay for security over time—both in dollars and in potential loss of convenience and user friendliness.

Why buy security? There are two especially good reasons, described next.

Government Requirements

If you sell to the government, you almost certainly need to use many of the security technologies described in this book. If you're a computer vendor who's trying to sell a lot of computer workstations, for example, you may be forced to build security into your products or to buy the technology from others.

Most government requisitions specify security requirements along with operational requirements. Most operating systems must conform to a specified level of the Common Criteria for security. Common Criteria has replaced the security levels specified

in the former “Orange Book” (see Appendix C for more information) standard for trusted systems, but the language of the Orange Book still permeates the security business.

You may need to use a particular form of encryption to protect stored and transmitted data; for high-security applications, consider using TEMPEST shielding.

In addition to being the major purchaser of computer security, the government has historically been the driving force behind the development of security products and the standardization of what makes a system “secure.” Chapter 2 describes what these security standards are and how they developed.

Information Protection

Government agencies are required by law to protect both classified information and also what’s called “sensitive unclassified” information. Examples include such information as productivity statistics (from the Department of Commerce), currency production and transfer information (Department of the Treasury), and embassy personnel information (Department of State). What makes this information sensitive is the fact that its theft or modification could potentially disrupt the nation’s economy or compromise its employees. Similarly, the breach of individual health and financial records maintained by such agencies as the Social Security Administration, the FBI, the IRS, and others could have severe legal and personal repercussions.

If your business uses sensitive corporate information, you need security too. In some cases, you need to keep information secret. Obvious examples of such information include banking funds transfers, oil resource data, stock futures strategies, medical research data, personal medical data, and airline reservation information.

In other cases, you need to ensure the integrity of the information. A primary example is electronic funds transfer (EFT). The Society for Worldwide Interbank Financial Telecommunications (SWIFT), for example, provides EFT services for over 7,500 financial institutions in 200 countries, securely completing millions of transactions a day, and billions of transactions each year (2 billion in 2003 alone). SWIFT and other financial institutions require absolute accuracy in their transactions. As described earlier, message authentication and related techniques play an important part in ensuring the accuracy of such financial information.

Even if your business doesn’t involve national defense secrets or international funds transfers, the information you process is critical to your own business. Information may well be your most important business asset. Any theft or compromise of information is as much an attack on your business as is the theft of any other company asset. And the loss of information is more likely to damage your business than would a more tangible loss. Even if you’re not convinced that you need security, your insurance company and your shareholders may be. The concept of “reasonable safeguards” is having an impact on users of computer systems now. You may find that if

you do not provide adequate security for your information, your insurance may not cover a loss, and you may lose a court battle over “computer malpractice,” “preventable loss,” or the “standard of due care.”

What's a User to Do?

If you use a computer of any kind, anywhere, computer security not only affects you, it is your responsibility. If your device is compromised, you could be an unwitting partner in crime, or at least a source of inconvenience. And before you can even worry about computer abuse, you need to worry about power failures, natural disasters, making backups in case of a disk failure or virus attack, and making sure no one walks off with your equipment or backup media. If you work on a network, you have to observe network access and security regulations. You'll find that as quickly as manufacturers release cures for network or computer exploits, you will need to adopt them and incorporate them into your daily routine.

If your organization has installed a highly secure system, you may have to accept substantial restrictions on the administrative tasks you might have performed in the past—sheer torture for power users used to configuring their systems or at least their desktops to be just the way they want them. If your system supports mandatory access controls, you'll find that even if another user wants to let you read or print one of her files, the system may not let you.

Conversely, some organizations that really should know better sometimes display a stunning lack of security. In this case, you're on your own recognizance: sure, you could reprogram the boss's spreadsheet and plot yourself a big raise, but you would not want to be you the day you are caught.

Computer security is a multibillion dollar industry that addresses a threat that now impacts everyone. Major software companies warn users to install personal firewall software on their PCs in addition to performing frequent software updates to avoid the latest hazards. These days, only a fool or one uninformed would go too long without periodically updating his virus definition tables.

Summary

Hopefully, you are now convinced that security is good for you and your data—that it's worth it for you to spend a small amount of extra time worrying about viruses, protecting your login account, and otherwise practicing safe computing to the best of your ability. Remember that security means more than keeping the bad guys out. It also means doing what you can to protect, or at least to avoid endangering, the network and computers used by yourself and others.

CHAPTER 2

Some Security History

Computer security is a hot issue today, but it's an issue that's been simmering for many years. The development of government security regulations and standards, research into security mechanisms, and debates over the threats to information and the costs of protecting against these threats—all of these activities are well into their fourth decade. Computer security itself isn't new. What's new is security's broader focus (security means more than just keeping outsiders out) and its wider appeal (security is important to business and folks at home as well as government).

This chapter describes how we got to where we are today. It summarizes key events in the history of computer security, discusses some of the government standards and programs involved with computer security, and introduces the concept of computer databases and the preservation of privacy.

Information and Its Controls

Information security is almost as old as information itself. Whenever people develop new methods of recording, storing, or transmitting information, these innovations are almost inevitably followed by methods of harnessing the new technologies and protecting the information they process. They're also followed by government investigations and controls. For example:

- In 1793, the first commercial semaphore system (use of mechanized flags) was established between two locations near Paris. Semaphore signaling came to be used throughout France, Italy, Germany, and Russia. Thousands were employed manning the stations, which operated at a speed of about 15 characters per minute. Code books were used so that whole sentences could be represented by a few characters. Semaphores weren't very successful in England because of fog and smoke, but in the United States, systems of this kind are the reason so many communities have geographic names such as Signal Hill, Beacon Rock, Signal Butte, and Semaphore Pointe.

- With Samuel F.B. Morse's introduction of the telegraph came concerns for protecting the confidentiality of transmitted messages. In 1845, just a year after the invention, a commercial encryption code was developed to keep the transmitted messages secret.
- Within five years of the introduction of the telephone in 1881, a patent application was filed for a voice scrambler.
- In the 1920s, the use of telephone wiretaps by both government and criminal forces resulted in a public outcry, Congressional hearings, and, ultimately, legislation prohibiting most wiretapping.
- In the 1930s, Title VI of the Communications Act of 1934 prohibited unauthorized interception and publication of communications by wire or radio, while giving the President certain powers to deal with communication matters in the event of war or other national emergency.
- In the 1940s, concerns about controlling the proliferation of information about atomic energy led to the Atomic Energy Act of 1946. This act created a Restricted Data category of information requiring special protection and penalties for dissemination. Similar controls have been imposed on new advances in other scientific fields.
- In the 1980s, the Defense Authorization Act specified controls on technical information about emerging military and space technologies.
- In the 1990s, attention turned to not just keeping out the bad guys, but keeping out the bad codes. Viruses had been around for some time, but it took dramatic increases in the number of home and business computers to make viruses a plague for everyone.
- In the 2000s, viruses have diverged into spyware, which detects and reports a user's Internet activities, adware, which presents the user with targeted advertisements for goods and services, and malware, which is code designed to cause harm or further the illicit ends of the perpetrator.
- In the months following the terrorist attacks on 9/11, the Homeland Protection Act granted broad executive powers related to gathering of information from private communications.

Like any other new technology, computers have raised substantial questions about the degree to which the technology should be controlled—and by whom. Even newer technologies—for example, imaging systems that may impact the integrity of legal and financial documents, including U.S. currency—will no doubt raise the same types of complex issues. Widely available software that allows users to doctor electronic photographs makes it hard to determine what is real and what is designed to entertain or to persuade. Computers and wireless communications also make it easy to eavesdrop.

One ongoing debate in the computer security world is over the government's restriction of technological information. Government needs to protect certain kinds of

information, such as national defense data and the take from intelligence gathering activities. Particular security technologies—for example, cryptographic products—are very effective at safeguarding such information. Should the government be able to control who can and cannot buy such technologies? Should there be any limits on such sales? For example, should enemy governments be able to buy cryptographic products that may make it more difficult for U.S. intelligence operations to monitor these nations' communications? What about information concerning the technologies themselves, for example, technical papers about cryptographic algorithms? Should these have to be submitted for government examination and possible censorship? Encryption technologies have been variously classified as munitions or as a normal part of software. Is there a need to stifle the development of products developed privately that may inadvertently mimic (or possibly outperform) existing or proposed government communications technologies? Can technology and the free exchange of intellectual data flourish in an environment that tries to control certain kinds of intellectual exchanges?

A somewhat more alarming trend has been the government role in limiting or suppressing the use of cryptographic techniques between private parties in the United States. One such method, Pretty Good Privacy (PGP), was promulgated at tremendous sacrifice to its developer. At length, PGP prevailed, but some in the cryptographic community are concerned that its commercialization may have encouraged a government-sponsored backdoor that allows easy transmission decoding within the constraints of the legal system. A similar situation has been seen in the telecommunications industry: “law-enforcement ports” have begun to appear in commercial telephone switchgear, creating the possibility of wiretapping without due process.

Another debate concerns the involvement of the government in mandating the protection of nongovernment information. Should the government have any control over the protection of such information? Who gets to decide whether information such as productivity statistics, geological surveys, and health information must be protected from public scrutiny? From whom is it being protected? In 2003, a graduate student compiled a list of all the connections into and out of a major city using publicly available data. Debate ranged from whether the document should be classified to whether the student had gone beyond the scope of a standard research paper and had in fact committed a crime by assembling such a document. Should the government impose the same security standards on systems used to process commercial information as those imposed on systems for government information? The importance of the commercial infrastructure to the economy suggests that the commercial infrastructure deserves attention, similar to a bridge, tunnel, or airport.

As you'd expect, different people have a variety of opinions about these questions. We'll discuss such questions and representative opinions throughout this book.

Computer Security: Then and Now

In the early days of computing, computer systems were large, rare, and very expensive. Naturally enough, those organizations lucky enough to have a computer tried their best to protect it. Computer security was simply one aspect of general plant security. Computer buildings, floors, and rooms were guarded and alarmed to prevent outsiders from intruding and disrupting computer operations. Security concerns focused on physical break-ins, the theft of computer equipment, and the physical theft or destruction of disk packs, tape reels, punched cards, and other media. (This was in the day when you could destroy a program by grabbing its card deck and scattering it to the wind. Incorrect reassembly of the card deck could cause some memorable computer errors. It was important to choose carefully one's victim in such a prank, lest they be faster or stronger than they appeared.)

Insiders were also kept at bay. Few people knew how to use computers, and only those who knew the secrets of the machine were privileged to stand in its presence. Most users never saw the computers that crunched their numbers. Batch processing meant that users submitted carefully screened jobs—often through protected slots in the doors of computer rooms—to operators who actually put the machine through its paces.

Times changed. During the late 1960s and 1970s, computer technology was transformed, and with it the ways in which users related to computers and data. Multiprogramming, time-sharing, and networking dramatically changed the rules of the game. Users could now interact directly with a computer system via a terminal, giving them more power and flexibility but also opening up new possibilities for abuse. Acoustic couplers—modems with foam pads into which a telephone handset was inserted—allowed connectivity not just in the computer room or building, but from cities far away.

Telecommunications—the ability to access computers from remote locations and to share programs and data—radically changed computer usage. Large businesses began to automate and store online information about their customers, vendors, and commercial transactions. Networks linked minicomputers together and allowed them to communicate with each other and with mainframes containing large online databases. It became much easier to make wholesale changes to data—and much easier for errors to wreak widespread damage. Banking and the transfer of assets became an electronic business.

The increased ease and flexibility of computer access also had a dramatic impact on education. Universities and other schools that could not afford their own computer installations now found it possible to tie into computer networks and centralized computers and databases.

Finally, personal computers put the processing power on the desktop. Expensive mainframe connections were useful, but not always needed. More and more students had an opportunity to experiment with computers, and computers increasingly became a part of the curriculum. The result was a huge increase in the number of people who knew how to use computers. The desire to access shared resources led to interconnection of computers, which led eventually to the Internet. Today, if some resource is not available in the machine on your desk, that machine can connect to another that has the resource.

Inevitably, the increased availability of online systems and information led to abuses. Computer security concerns broadened. Instead of worrying only about intrusions by outsiders into computer facilities and equipment (and an occasional computer operator going berserk), organizations now had to worry about computers that were vulnerable to sneak attacks over telephone lines, and information that could be stolen or changed by intruders who didn't leave a trace. Incidents of computer crime began to be reported. Individuals and government agencies expressed concerns about the invasion of privacy posed by the availability of individual financial, legal, and medical records on shared online databases. As computer terminals and modems became more affordable, these perils intensified.

The 1980s saw the dawn of a new age of computing. With the introduction of the personal computer, individuals of all ages and occupations became computer users. Computers appeared on desks at home and at the office. Small children learned to use computers before they could read. As the price of systems dropped, and as inexpensive accounting packages became available, more and more small businesses automated their operations. PC technology introduced new risks. Precious and irreplaceable corporate data was now stored on diskettes, which could too easily be lost or stolen.

Where Are We, Really?

Some pundits have noted that computers actually have not changed the amount of work done in an office, merely its speed. The same accountant that used to make trial balances near the end of the month now can close the books each evening.

Another artifact of broad computerization is the shifting of work. For instance, authors once typed manuscripts, leaving the entire production process to artists and others who saw the publication through to the presses, once the words were right. Today, word processing programs include formatting commands that place the burden of typography and type selection at the feet of those who create the content.

A final observation is that computers may actually do little to save money. Where once a firm may have employed several accountants, today there may be one or two, but in their place there is now a squad of IT professionals and telecommunications experts needed to keep the systems running.

As PC networks proliferated, so did the use of electronic mail, and bulletin boards, dramatically increasing the ability of users to communicate with other users and computers, and vastly raising the security stakes. In the past, even a skilled attacker might have breached security only in a single computer, installation, or local area network; now she had the potential to disrupt nationwide or even worldwide computer operations.

The 1980s also saw systems under attack. Theoretical possibilities came to life as *War Games*-like break-ins played themselves out on the front pages of local newspapers: the Internet worm, the Friday the 13th virus, the West German hackers of Cuckoo's Egg fame, and players at espionage. Government, business, and individual users suddenly saw the consequences of ignoring security risks.

The 1990s faced the challenges brought by open systems, distributed computing, and massive interrelationships between network elements. Along with an increasing dependence on networks and the need to share data, applications, and hardware/software resources across vendor boundaries came increasing security risks. In this decade the security business came of age, with more vendors developing trusted systems, bundling security functions, and beginning the development of biometric devices and network security products such as firewalls and intrusion detection systems. Still, security systems lagged behind the technologies they sought to control. Both businesses and individuals lagged still further behind the growing ranks of attackers, many of whom were no longer elite hackers, but an annoying and dangerous breed of vandal known as the *script kiddie*, short on skill, long on a desire to interrupt computation for the sake of doing so.

In the 2000s, particularly after the attacks of 9/11, security took on a serious tone. Corporations and government alike became more willing to make security an integral part of their products and their jobs. New certifications for IT workers, such as the CompTIA's Security+, the (ISC)² Certified Information System Security Professional CISSP, and the Cisco Certified Security Professional (CCSP) popped up, making security a hot ticket in an otherwise flagging industry.

The challenge of this decade will be to consolidate what we've learned—to build computer security into our products and our daily routines, to protect data without unnecessarily impeding our ability to access it, and to make sure that both security products and government and industry standards grow to meet the ever-increasing scope and challenges of technology.

Early Computer Security Efforts

The earliest computer-related security activities began in the 1950s, with the development of the first TEMPEST security standard, the consideration of security issues in some of the earliest computer system designs, and the establishment of the first government security organization, the U.S. Communications Security (COMSEC)

Board. The board, which consisted of representatives from many different branches of the government, oversaw the protection of classified information.

Emanations and Computing

The protection of computer data has always traveled hand in hand with the protection of computer signals. Like radio transmitters, the circuits and wires of a computer system or other device with an embedded computer radiate electromagnetic waves that vary in nature according to the information being processed. Because they pose a security threat, there are standards that limit these emanations as well as standards that strive to prevent outright data theft.

Although these events set the scene for later computer security advances, the 1960s marked the true beginning of the age of computer security, with initiatives by the Department of Defense, the National Security Agency, and the National Bureau of Standards (now the National Institute of Standards and Technology or NIST), coupled with the first public awareness of security. The Spring Joint Computer Conference of 1967 is generally recognized as being the locale for the first comprehensive computer security presentation for a technical audience. Willis H. Ware of the RAND Corporation chaired a session that addressed the wide variety of vulnerabilities present in resource-sharing, remote-access computer systems. The session addressed threats ranging from electromagnetic radiation to bugs on communications lines to unauthorized programmer and user access to systems and data.

The Department of Defense, because of its strong interest in protecting military computers and classified information, was an early partisan of computer security efforts. In 1967, DoD began to study the potential threats to DoD computer systems and information. In October of that year, DoD assembled a task force under the auspices of the Defense Science Board within the Advanced Research Projects Agency (ARPA), now known as the Defense Advanced Research Projects Agency, or DARPA. The task force worked for the next two years examining systems and networks, identifying vulnerabilities and threats, and introducing methods of safeguarding and controlling access to defense computers, systems, networks, and information. Published as a classified document in 1970, the task force report, *Security Controls for Computer Systems*, was a landmark publication in the history of computer security. Its recommendations, and the research that followed its publication, led to a number of programs dedicated to protecting classified information and setting standards for protection.

The Department of Defense took to heart the recommendations of the task force and began to develop regulations for enforcing the security of the computer systems, networks, and classified data used by DoD and its contractors. In 1972, DoD issued a

directive* and an accompanying manual† that established a consistent DoD policy for computer controls and techniques. The directive stated overall policy as follows:

Classified material contained in an ADP system shall be safeguarded by the continuous employment of protective features in the system's hardware and software design and configuration.

The directive also stipulated that systems specifically protect both the computer equipment and the data that it processes by preventing deliberate and inadvertent access to classified material by unauthorized persons, as well as unauthorized manipulation of the computer and associated equipment.

During the 1970s, under the sponsorship of DoD and industry, a number of major initiatives were undertaken to better understand the system vulnerabilities and threats that early studies had exposed and to begin to develop technical measures for countering these threats. These initiatives fell into three general categories: tiger teams, security research studies, and development of the first secure operating systems.

Tiger Teams

During the 1970s, *tiger teams* first emerged on the computer scene. Tiger teams were government- and industry-sponsored teams of crackers who attempted to break down the defenses of computer systems in an effort to uncover, and eventually patch, security holes. Most tiger teams were sponsored by DoD, but IBM aroused a great deal of public awareness of computer security by committing to spend \$40 million to address computer security issues—and tiger teams were an important part of finding security flaws in the company's own products.‡

Tiger teams were an effective way to find and fix security problems, but their efforts were necessarily piecemeal. U.S. Air Force Lieutenant General Lincoln D. Fauser, former Director of the National Security Agency, wrote that the efforts of the tiger teams resulted in two significant conclusions:§

Attempts to correct (patch) identified security vulnerabilities were not sufficient to prevent subsequent repenetrations. New tiger teams often found security flaws not found

* *Security Requirements for Automatic Data Processing (ADP) Systems* (DoD 5200.28), 1972, revised 1978.

† *ADP Security Manual—Techniques & Procedures for Implementing, Deactivating, Testing, and Evaluating Secure Resource-Sharing ADP Systems* (DoD 5200.28-M), 1973, revised 1979.

‡ Reports describing the efforts of tiger teams included:

C.R. Attanasio, P.W. Markstein, and R.J. Phillips, "Penetrating an Operating System: A Study of VM/370 Integrity," *IBM Systems Journal*, Volume 15, Number 1, 1974.

R. Bisbey, G. Popek, and J. Carlstedt, *Protection Errors in Operating Systems*, USC Information Sciences Institute, 1978.

P.A. Karger and Schell, R.R., *Multics Security Evaluation: Vulnerability Analysis*, (ESD-TR-74-193), Electronic Systems Division, U.S. Air Force, Hanscom Air Force Base, Bedford (MA), 1974 (available from NTIS: AD A001120).

§ Lincoln D. Fauser, "Computer Security Goals of the Department of Defense," *Computer Security Journal*, Summer 1984.

by earlier tiger teams, and one could not rely on the failure of a penetration effort to indicate that there were no exploitable security flaws.

The only apparent means of guaranteeing the protection of system resources would be to design verifiable protection mechanisms into computer systems.

Tiger teams served a useful function by identifying security flaws and demonstrating how easily these flaws could be exploited. In fact, the current model in the hacker community of discovering vulnerabilities through probing, alerting manufacturers, and after a suitable waiting period, publishing a proposed exploit has its roots in tiger team methodology. The commercial practice of *penetration testing*, or running *pen tests*, has its roots in tiger teams as well.

By the end of the 1970s, however, it was apparent that a more rigorous method of building, testing, and evaluating computer systems was needed. Although this was achieved in certain defense-related systems, the commercial world in many instances is still waiting.

Research and Modeling

During the 1970s, DoD and other agencies sponsored a number of ground-breaking research projects aimed at identifying security requirements, formulating security policy models, and defining recommended guidelines and controls.

In the research report of the Computer Security Technology Planning Study,* James P. Anderson introduced the concept of a reference monitor, an entity that “enforces the authorized access relationships between subjects and objects of a system.” The idea of a reference monitor became very important to the development of standards and technologies for secure systems. The concept of reference monitors and the function of subjects and objects in secure systems are described in Appendix C.

DoD went on to sponsor additional research and development in the 1970s focusing on the development of security policy models. A *security policy* defines system security by stating the set of laws, rules, and practices that regulate how an organization manages, protects, and distributes sensitive information. The mechanisms necessary to enforce a security policy usually conform to a specific *security model*. This is also discussed in Appendix C.

A number of additional technical research reports published during the 1970s defined secure systems and security requirements.[†] During the 1970s, David Bell and

* J.P. Anderson, *Computer Security Technology Planning Study*, (ESD-TR-73-51), Electronic Systems Division, U.S. Air Force, Hanscom Air Force Base, Bedford (MA), 1972 (available from NTIS: AD 758206).

† These reports included:

Secure Minicomputer Operating System (KSOS) Department of Defense Kernelized Secure Operating System, (WDL-7932), Ford Aerospace and Communications Corporation, 1978.

P.G. Neumann et al, *A Provably Secure Operating System: The System, Its Application, and Proofs*, Final Report, Project 4332, SRI International, Menlo Park (CA), 1977.

Leonard LaPadula developed the first mathematical model of a multilevel security policy. The Bell and LaPadula model* was central to the development of basic computer security standards and laid the groundwork for a number of later security models, and their application in government security standards. Although current security literature tends to overlook the mathematical basis of security and encryption in favor of a more practical approach, behind almost any durable, robust security concept is a solid string of mathematics proving it valid.

Secure Systems Development

A number of government-sponsored projects undertook to develop the first “secure” systems during the 1970s. Most of these efforts were devoted to developing prototypes for security kernels. A *security kernel* is the part of the operating system that controls access to system resources. The most significant was an Air Force-funded project that led to the development of a security kernel for the Multics (Multiplexed Information and Computing Service) system.[†]

Multics allowed users with different security clearances to simultaneously access information that had been classified at different levels. Because it embodied so many well-designed security features, the Multics system was particularly important to the development of later secure systems. Multics was a large-scale, highly interactive computer system that offered both hardware- and software-enforced security. Specific features of Multics included extensive password and login controls; data security through access control lists (ACLs), an access isolation mechanism (AIM), and a ring mechanism; auditing of all system access operations; decentralized system administration; and architectural features such as paged and segmented virtual memory and stack-controlled process architecture.

Other security kernels under development during the 1970s included Mitre Corporation’s Digital Equipment Corporation PDP-11/45[‡] and UCLA’s Data Secure Unix PDP-11/70.[§]

* D.E. Bell and L.J. LaPadula, *Secure Computer Systems: Mathematical Foundations and Model*, (M74-244), Mitre Corporation, Bedford (MA), 1973 (available from NTIS: AD 771543).

† For information about Multics, see J.C. Whitmore et al, *Design for Multics Security Enhancements*, (ESD-TR74-176), Honeywell Information Systems, Cambridge (MA), 1973. (Available from NTIS: AD A030801.) See also Elliot Organick, *The Multics System: An Examination of Its Structure*, MIT Press, Cambridge (MA), 1975.

‡ W.L. Schiller, *The Design and Specification of a Security Kernel for the PDP-11/45*, ESD-TR-75-69, Mitre Corporation, Bedford (MA), 1975 (available from NTIS: AD A011712).

§ Popek et al., “UCLA Secure UNIX,” *Proceedings*, National Computer Conference, New York (NY), 1979.

Building Toward Standardization

Late in the 1970s, two important government initiatives significantly affected the development of computer security standards and methods. In 1977, the Department of Defense announced the DoD Computer Security Initiative under the auspices of the Under Secretary of Defense for Research and Engineering. The goal was to focus national attention and resources on computer security issues. The initiative was launched in 1978 when DoD called together government and industry participants in a series of seminars. The goal of the seminars was to answer these questions:

- Are secure computer systems useful and feasible?
- What mechanisms should be developed to evaluate and approve secure computer systems?
- How can computer vendors be encouraged to develop secure computer systems?

The second important initiative came from the National Bureau of Standards (NBS), now known as the National Institute of Standards and Technology. NIST has historically been responsible for the development of standards of all kinds. As a consequence of the Brooks Act of 1965 (described in “Computer Security Act” later in this chapter), NIST (as NBS) became the agency responsible for researching and developing standards for federal computer purchase and use, and for assisting other agencies in implementing these standards. The bureau has published many federal standards known as Federal Information Processing Standards publications (FIPS PUBs) in all areas of computer technology, including computer security. Over the course of the next decade or so after the Brooks Act, NBS focused on two distinct security standardization efforts: development of standards for building and evaluating secure computer systems, and development of a national standard for cryptography.

Standards for Secure Systems

NBS’s first charge was to evaluate the federal government’s overall computer security needs and to begin to find ways to meet them. Early efforts, based on NBS’s Brooks Act mandate, included the following:

1968

NBS performed an initial study to evaluate the government’s computer security needs.

1972

NBS sponsored a conference on computer security in collaboration with the ACM.

1973

NBS initiated a program aimed at researching development standards for computer security.

1977

NBS began a series of Invitational Workshops dedicated to the Audit and Evaluation of Computer Systems. These had far-reaching consequences for the development of standards for secure systems.

At the first Invitational Workshop in 1977, 58 experts in computer technology and security assembled to define problems and develop solutions for building and evaluating secure systems. Invitees represented NBS, the General Accounting Organization (GAO), other government agencies, and industry. Their goal? To determine:

What authoritative ways exist, or should exist, to decide whether a particular computer system is “secure enough” for a particular intended environment or operation, and if a given system is not “secure enough,” what measures could or should be taken to make it so.

Workshop participants considered many different aspects of computer security, including accuracy, reliability, timeliness, and confidentiality. The NBS workshops resulted in the publication of several reports.* These concluded that achieving security required attention to all three of the following:

Policy

What security rules should be enforced for sensitive information?

Mechanisms

What hardware and software mechanisms are needed to enforce the policy?

Assurance

What needs to be done to make a convincing case that the mechanisms do support the policy even when the system is subject to threats?

The NBS report stated:

By any reasonable definition of “secure” no current operating system today can be considered “secure” . . . We hope the reader does not interpret this to mean that highly sensitive information cannot be dealt with securely in a computer, for of course that is done all the time. The point is that the internal control mechanisms of current operating systems have too low integrity for them to . . . effectively isolate a user on the system from data that is at a “higher” security level than he is trusted . . . to deal with.

This conclusion was an important one in terms of the multilevel security concepts discussed in Part II of this book.

The NBS workshops recommended that a number of actions be taken. One action was to formulate a detailed computer security policy for sensitive information not covered by national security policies and guidelines. Another was to establish a formal

* Z. Ruthberg and R. McKenzie, ed., *Audit and Evaluation of Computer Security*, Special Publication 500-19, National Bureau of Standards, Gaithersburg (MD), 1980 (SN 003-003-01848-1).

Z. Ruthberg, ed., *Audit and Evaluation of Computer Security II: System Vulnerabilities and Controls*, Special Publication 500-57, (MD78733), National Bureau of Standards, Gaithersburg (MD), 1980 (SN 003-003-02178-4).

security and evaluation and accreditation process, including the publication of a list of approved products to guide specification and procurement of systems intended to handle sensitive information. A third was to establish a standard, formalized, institutionalized technical means of measuring or evaluating the overall security of a system.

As an outgrowth of the NBS workshops, the Mitre Corporation was assigned the task of developing an initial set of computer security evaluation criteria that could be used to assess the degree of trust that could be placed in a computer system that protected classified data. Beginning in 1979, in response to the NBS workshop and report on the standardization of computer security requirements, the Office of the Secretary of Defense conducted a series of public seminars on the DoD Computer Security Initiative. One result of these seminars was that the Deputy Secretary of Defense assigned to the Director of the National Security Agency (NSA) responsibility for increasing the use of trusted information security products within the Department of Defense.

National Computer Security Center

As a result of NSA's new responsibility for information security, on January 2, 1981, the DoD Computer Security Center (CSC) was established within NSA to expand upon the work begun by the DoD Computer Security Initiative. The official charter of the CSC is contained in the DoD Directive entitled "Computer Security Evaluation Center" (5215.1).

Several years later, the computer security responsibilities held by CSC were expanded to include all federal agencies and the Center became known as the National Computer Security Center (NCSC). The Center was founded with the following goals:

- Encourage the widespread availability of trusted computer systems
- Evaluate the technical protection capabilities of industry- and government-developed systems
- Provide technical support of government and industry groups engaged in computer security research and development
- Develop technical criteria for the evaluation of computer systems
- Evaluate commercial systems
- Conduct and sponsor research in computer and network security technology
- Develop and provide access to verification and analysis tools used to develop and test secure computer systems
- Conduct training in areas of computer security
- Disseminate computer security information to other branches of the federal government and to industry

In 1985, NSA also merged its communications and computer security responsibilities together under the Deputy Directorate for Information Security Systems (INFOSEC).

Birth of the Orange Book

The Center met an important goal by publishing the *Department of Defense Trusted Computer System Evaluation Criteria* (TCSEC), commonly known as the Orange Book because of the color of its cover. Based on the computer security evaluation criteria developed by Mitre,* and on such developments as the security model developed by Bell and LaPadula, this publication was distributed to government and industry experts, revised, and finally released in August 1983.

The Orange Book is the bible of secure system development. It describes the evaluation criteria used to assess the level of trust that can be placed in a particular computer system. It effectively makes security a measurable commodity so a buyer can identify the exact level of security required for a particular system, application, or environment. The Orange Book presents a graded classification of secure systems. It defines four broad hierarchical divisions, or levels, of protection—D, C, B, and A, in order of increasing security. Within each division, the Orange Book defines one or more classes, each defined by a specific set of criteria that a system must meet to achieve a rating in that class. Some divisions have only a single class, others have two or three. The original Orange Book was revised slightly and reissued in December 1985.

Using the Orange Book criteria, NCSC performed evaluations of products submitted by vendors for certification at a particular level of trust. Products that are successfully evaluated through the NCSC Trusted Products Evaluation Program (TPEP) are placed on the Evaluated Products List (EPL). Appendix C describes the Orange Book evaluation criteria (and also mentions some of the complaints about these criteria). The Orange Book is so pervasive that although the standards have transferred to its successor, the Common Criteria, Orange Book designations are often used synonymously with Common Criteria equivalents, and students research one by studying the other.

In the days since the Orange Book, the focus on common security has shifted to the Common Criteria. This set of guidelines describes parameters for secure computing and has a scale to rate the performance of an examined system against those parameters. Based on the European White Book, Common Criteria, in conjunction with numerous FIPS, is the basis of computer security in the United States today. Orange Book culture is so enduring, however, that you can barely speak of one without invoking the other. An overview of the interrelationship of these standards is contained in Appendix C.

* G.H. Nibaldi, *Proposed Technical Evaluation Criteria for Trusted Computer Systems*, (M79-225), Mitre Corporation, Bedford (MA), 1979 (available from NTIS: AD A108832).

G.H. Nibaldi, *Specification of a Trusted Computing Base*, (M79-228), Mitre Corporation, Bedford (MA), 1979 (NTIS publication: AD 108831).

Standards for Cryptography

During the 1970s, interest in a national cryptographic standard began to build within the government. The idea was to find an algorithm that could be used to protect sensitive unclassified government information (classified algorithms were already being used to protect classified information) and sensitive commercial data such as banking electronic funds transfers. In 1973, the National Bureau of Standards, part of the Department of Commerce, invited vendors to submit data encryption techniques that might be used as the basis of an encryptions algorithm.

Under the auspices of the Institute of Computer Science and Technology (ICST), later known as the National Computer Systems Laboratory, NBS organized a series of workshops for government and industry representatives to select a national encryption algorithm. The method eventually selected by NBS became known as the Data Encryption Standard (DES).

The DES was adopted as a Federal Information Processing Standard (FIPS PUB 46) in 1977 as the official method of protecting unclassified data in the computers of U.S. government agencies, and was subsequently adopted as an American National Standards Institute (ANSI) standard.

The DES consists of two components: an algorithm and a key. The DES algorithm is a complex, iterative process that is public information. This algorithm uses a secret value—the key—to encode and decode messages.

DES technology has been embedded in the products of many commercial products. Until 1986, the National Security Agency endorsed products containing DES-based algorithms. In 1986, NSA announced that it would no longer endorse such products. There was a substantial reaction to this decision by vendors, users, and other government agencies. Chapter 7 describes DES in greater detail and outlines some of the issues surrounding the use of the algorithm.

DES has now been cracked, both by special-purpose devices (not necessarily computers) constructed of microchips and by clusters of computers operating in tandem over the Internet. While the DES algorithm is likely to remain in use for some time (it's still efficient in certain two-way voice encryption systems), cryptographic researchers have continued to work on the development of more advanced algorithms. A competition was held in the late 1990s to determine which encryption standard would become the Advanced Encryption System (AES). The winner of the competition was an algorithm called Rijndael. The “losers,” many of which were powerful encryption tools, are also enjoying success in the world today. Most are available as open source programs.

Through the Commercial Communications Security Endorsement Program (CCEP), government and industry representatives develop, test, and endorse new cryptographic products.

Standards for Emanations

As early as the 1950s, concerns began to develop about the possibility that the electrical and electromagnetic radiation that emanates from computer equipment (as it does from all electronic equipment) could be intercepted and deciphered.

It works like this: any time a current flows, magnetic fields form around it. Conversely, when magnetic fields change size or shape, they induce currents in nearby conductors. Finally, any voltage that exists on one side of an insulator has the ability to cause changes to a voltage on the other side of an insulator due to the coupling of charges. All put together, operating any device that uses electricity can create signals that are detectable elsewhere. Often you see this as a disturbance of some kind, such as the interference on a televised football game caused by someone operating a vacuum cleaner or blender nearby.

In an effort to counter this threat, the U.S. government established the first standard for the level of emanations that was acceptable for equipment used to process classified information in the late 1950s. During the 1960s and 1970s, as standardization efforts proceeded in areas of secure systems and cryptography, they also resulted in the refinement of the initial TEMPEST standard and the establishment of a program to endorse products that met the requirements of this standard.

The Industrial TEMPEST Program was established in 1974 with three main goals:

- Specify a TEMPEST standard that sets allowable limits on the levels of emission from electronic equipment.
- Outline criteria for testing equipment that, according to its vendors, meets the TEMPEST standard.
- Certify vendor equipment that successfully meets the TEMPEST standard.

The National TEMPEST Standard, known as NACSEM 5100 (National Communications Security Emanations Memorandum 5100), was published in 1970. Much of the document was classified. This standard has been revised several times. The current standards family, NSTISSAM/1-91 (Compromising Emanations Laboratory Test Requirements, Electromagnetic) was published in 1971. It is superseded by NSTISSAM/1-92; NSTISSAM/2-91 (Compromising Emanations Analysis Handbook) was published in 1991; NSTISSAM/3-91 (Maintenance and Disposition of TEMPEST Equipment) was published in 1991, with certain augmentations published in 1995.

Government and industry representatives have worked together to set standards and to develop, test, and certify TEMPEST equipment. The U.S. government approves laboratories that evaluate TEMPEST products.

Computer Security Mandates and Legislation

Throughout history, new advances in the availability, processing, and transmission of information have inevitably been followed by new security methods, federal laws,

and procedural controls. These are typically aimed at protecting information that's considered to be essential to national security or other national interests.

In the 1970s and into the 1980s, national concerns about the Soviet interception of domestic communications intensified. Since the 1990s, that threat has diminished, but other nations, primarily those in the Middle East, but certainly Communist China and North Korea as well, have become threats. All this has led to a large number of security-related pieces of legislation, Presidential directives, and national policy statements. These fall into several categories:

Protection of classified or sensitive information

Legislation mandating computer security practices by federal agencies and contractors. The idea of this legislation is that organizations that process classified or sensitive unclassified government information must be careful to protect that information from unauthorized access.

Computer crime

Legislation defining computer crime as an offense and extending other regulations to cover thefts and other abuses carried out by computers and other new techniques. In addition to federal policies, virtually all U.S. states have enacted their own legislation prohibiting computer crime and abuse.

Privacy

Legislation protecting the privacy of information maintained about individuals (e.g., health and financial records). Another consideration for computer privacy is the practice of merging records from multiple, seemingly benign databases into profiles that may reveal devastating amounts of information about an individual.

The Balancing Act

Although it may be a mischaracterization, much of the concern about computer security centers on the government. Data needs to be classified to avoid exposing sensitive information and the means used to collect it, or protected to avoid allowing unfriendly investigators to compile data and expose national weaknesses. In addition, the communications of terrorists and criminals can be a rich source of information to prevent or solve crime, but access to that information by law enforcement or military agencies can constitute an infringement of personal liberty (which is one of the principles upon which this country was founded). Thus government is both the protector and the cause of concern, and untying the hands of enforcement agencies that protect us while keeping their agents from scrounging secrets out of our trash is one of the enduring concerns of national computer policy.

National Security Decision Directive 145 (NSDD 145), entitled the National Policy on Telecommunications and Automated Information Systems Security, signed by President Reagan on September 17, 1984, had far-reaching significance in the world of computer security. NSDD 145 mandated the protection of both classified and unclassified sensitive information. It also gave NSA the obligation to “encourage,

advise, and if appropriate, assist” the private sector. Because it gave NSA jurisdiction in the private sector, NSDD 145 was a controversial directive.

NSDD 145 required systems that handle classified information to be secured as necessary to prevent access by unauthorized individuals. It also required that systems protect sensitive information, whether it originated in the government or outside it, in proportion to the potential damage that disclosure, alteration, or loss posed to national security. Examples of sensitive information include productivity statistics; information that might relate to the disruption of public services (e.g., air traffic control information); and virtually all information collected by such organizations as the Social Security Administration, the Federal Bureau of Investigation, the Internal Revenue Service, and the Census Bureau (e.g., individual health and financial records). These provisions rankled private enterprise, which used such data for planning purposes. Post 9/11 concerns have caused the pendulum to swing in favor of guarding such information.

In an effort to clarify the meaning of sensitive information and better interpret the requirements for its protection, the National Telecommunications and Information Systems Security Publication 2 (NTISSP 2), “National Policy on Protection of Sensitive but Unclassified Information in Federal Government Telecommunications and Automated Systems,” was published on October 29, 1986.

NTISSP 2 defined sensitive information as follows:

Sensitive, but unclassified information is information the disclosure, loss, misuse, alteration, or destruction of which could adversely affect national security or other federal government interests. National security interests are those unclassified matters that relate to the national defense or the foreign relations of the U.S. government. Other government interests are those related, but not limited to the wide range of government or government-derived economic, human, financial, industrial, agricultural, technology, and law enforcement information, as well as the privacy or confidentiality of personal or commercial proprietary information provided to the U.S. government by its citizens.

NTISSP 2 applied to all government agencies and contractors. It described the general categories of information that might relate to national security, foreign relations, or other government interests. It instructed the heads of departments and agencies to determine what information is sensitive but unclassified and to provide system protection for that information when it is electronically communicated, transferred, processed, or stored.

Computer Fraud and Abuse Act

Issued in 1986, the Computer Fraud and Abuse Act (18 U.S. Code 1030, now called Public Law 99-474) prohibits unauthorized or fraudulent access to government computers and establishes penalties for such access. Anyone convicted under this act faces a fine of \$5,000 or twice the value of anything obtained via unauthorized access, plus up to five years in jail (one year for first offenders). The act prohibits

access with the intent to defraud, as well as intentional trespassing. For example, posting passwords to federal computers on pirate electronic bulletin boards is a misdemeanor under this act. Robert T. Morris, author of the infamous Internet worm, was the first person convicted under the Computer Fraud and Abuse Act (Section 1030 (a)(5)), setting a precedent for future cases.

There are complaints about the wording of the Computer Fraud and Abuse Act on both sides of the issue.

The frustration of the Justice Department in prosecuting espionage cases in which classified information has been obtained by computer has led the Department to try to change the wording of the Computer Fraud and Abuse Act. The current law says it's a felony for anyone knowingly to gain unauthorized access to a computer and obtain classified information "with the intent or reason to believe that such information so obtained is to be used to the injury of the United States or to the advantage of any foreign nation." The Justice Department wants to drop that clause. The revised law would simply require proof that the intruder obtained certain information, not that the information was delivered or transmitted to anyone else.

Amendments to the Computer Fraud and Abuse Act have been proposed in Congress to expand the act's current government and banking focus to any systems used in interstate commerce or communications. The amendments would also change the orientation of the act from simple unauthorized access to the use of a computer system in performing other crimes.

On the other side of the issue, there have been complaints that the language of the Computer Fraud and Abuse Act is too general and can apply to anyone who writes or teaches about computer security. There have also been suggestions that the act should explicitly treat different types of offenses in different ways. At present, there is no clear distinction between people who use computers for hacking, for computer crime, or for terrorism.

As yet, no changes have been made to the language of the Computer Fraud and Abuse Act.

Computer Security Act

An important outgrowth of NSDD 145 was the development of the Computer Security Act of 1987 (H.R. 145), which later became Public Law 100-235. This act has expanded the definition of computer security protection and has increased awareness of computer security as an issue. It's the closest thing the United States has to a federal data protection policy. The Computer Security Act, which went into effect in September 1988, requires every U.S. government computer system that processes sensitive information to have a customized computer security plan for the system's management and use. In this, it mirrors the Common Criteria used in Europe. It also requires that all U.S. government employees, contractors, and others who directly

affect federal programs undergo ongoing periodic training in computer security. All users of systems containing sensitive data must also receive computer security training corresponding to the sensitivity of the data to which they have access.

The Computer Security Act further defines sensitive information as information whose “loss, misuse, unauthorized access to, or modification of could adversely affect the national interest, or the conduct of federal programs, or the privacy to which individuals are entitled under...the Privacy Act” (described in the section “Privacy Considerations” later in this chapter).

This act gave NIST new responsibility for federal computer security management. It assigned to the ICST within NIST responsibility for assessing the vulnerability of federal computer systems, for developing standards, and for providing technical assistance, as well as for developing guidelines for the training of federal personnel in computer security-related areas. NSA was assigned a role as advisor to NIST regarding technical safeguards.

The Computer Security Act does not affect the protection of classified information. It also allows requirements to be waived if they disrupt or slow down the implementation of what’s considered to be an important federal agency mission.

Searching for a Balance

Following the adoption of the Computer Security Act, concerns were voiced by Congress, industry, professional groups, and the general public about the potential for abuse. These concerns focused on NSA’s role in the private sector—in particular, relating to the control of unclassified information. Banks and other data-intensive industries feared that the act would impose disruptive restrictions on their operations. Civil libertarians were concerned about infringements on personal privacy. These concerns culminated in a series of Congressional hearings during 1987. During these hearings, NTIISP 2 was rescinded (March of 1987), and a review of NSDD 145 was ordered. Debate continues about the appropriate role of government in protecting and mandating the protection of information (see the sidebar “Hackers’ Rights”).

In 1990, NSDD 145 was revised and reissued as National Security Directive 42 (NSD 42). The new directive narrowed the original scope of NSDD 145 to primarily defense-related information.

Recent Government Security Initiatives

NSA, NIST, and DoD have all played an important part in developing computer security standards and in carrying out security programs. The exact balance of responsibility has not always been clear, and the boundaries continue to shift. Typically, NSA has had responsibility for the protection of classified military and intelligence information via computer security techniques, while NIST has been

Hackers' Rights

Computer hackers have come under increasing scrutiny, to the point that it may be technically illegal to perform many legitimate research activities. Legislation did not leave too many holes for small independent researchers, and large firms and universities do not want to risk a technical violation of the law. As a result, current activity by hackers takes place in an underground movement.

This is actually a return to conditions that existed about a decade ago. In the wake of early 1990s law enforcement sweeps such as Operation Sun Devil, which seized 42 computers and 23,000 floppy disks in 14 cities, shut down several bulletin board systems, and made four arrests in the spring of 1990 (some resulting from publication of information about the BellSouth 911 emergency telephone service and other “hacker tutorials”), a group was formed to provide legal aid to those it said had been victimized.

Mitch Kapor (of Lotus and ON Technology fame) and John Barlow (author and Grateful Dead songwriter) banded together, in conjunction with some well-known New York and Boston law firms, to form a defense team known as the Electronic Frontier Foundation, which seeks to protect the civil liberties of legitimate computer users. It was the EFF that successfully cracked the DES encryption algorithm using a specialized hardware device even as government witnesses were testifying in committees and before Congress that DES would take centuries and millennia to crack, even with a fast computer. Obviously, a great debt is owed these pioneers. Without them, trust in DES would have been misplaced, and private information compromised.

Members of the team charged that in trying to protect the government, industry, and individuals from cracker attacks, government agents violated constitutional rights. They also warned that the intimidation of today could thwart the technology of tomorrow. Senator Patrick Leahy, Democrat from Vermont, insisted that lawbreakers should be punished, but commented, “We cannot unduly inhibit the inquisitive 13-year-old who, if left to experiment today, may tomorrow develop telecommunications or computer technology to lead the United States into the 21st century. He represents our future and our best hope to remain a technologically competitive nation.”

His plea went unheard. Harsh prison sentences seem to rule now, as public anger has started to rise against spam and other invasions of privacy. In one recent case, an attacker sent spam to hundreds of thousands of people using the return addresses of certain reporters at some Philadelphia newspapers. The attacker wanted to rant about how his favorite sports team was getting a raw deal. Unfortunately, when thousands of users who received the email logged on to protest, and when the out-of-date addresses on the attacker’s list started bouncing back as returned to sender, the newspaper workers whose email addresses had been spoofed started getting deluged with mail, none of which was rightfully theirs. The attacker was eventually caught and sentenced to numerous consecutive sentences, 471 years in total. Manslaughter can carry a sentence as light as 5 to 10 years. Obviously the sentencing structure is currently out of whack.

responsible for developing standards and for developing computer security training programs. At times, both NSA and NIST have claimed responsibility for safeguarding unclassified, sensitive information, though most recently this type of information has been in NIST's bailiwick. Although standards for the evaluation of secure systems came about under NIST's auspices, actual evaluations are performed by the National Computer Security Center, a part of NSA. Different pieces of legislation seem to shift the balance one way or the other.

As a consequence of NSDD 145, the balance of responsibility seemed to shift to NSA (only to shift back again to NIST with the Computer Security Act and the revision of NSDD 145). NIST's Computer Security Program now encompasses a wide range of security activities, including developing and publishing computer security standards (in conjunction with organizations such as ANSI, ISO, and IEEE, described later in this chapter); conducting research in areas of security testing and solutions; and providing computer security training and support to other government agencies.

NSA and NIST have signed a memorandum of understanding about how they will cooperate on issues affecting the protection of sensitive unclassified information, mainly focusing on the implementation of the Computer Security Act. Their agreement is still subject to interpretation, and the balance continues to be a fragile one.

Modern Standards for Computer Security

In late 1990, NSA and NIST announced they were embarking on a joint venture that might result in new computer security criteria for computer procurements by federal agencies. The joint venture was also expected to take into consideration the European Community's Information Technology Security Evaluation Criteria (ITSEC), requirements that are under consideration as an international security standard. The result of this collaboration is known as the Common Criteria, and it is the foundation of computer protection documents today.

Under funding by DARPA, the National Research Council published a report, entitled *Computers at Risk*, that expressed concern about the state of computer security in the United States, and made recommendations about the need for a more coordinated security structure. The committee's report also suggested the publication of a comprehensive set of what are known as Generally Accepted System Security Principles (GASSP), which would clearly define necessary security features and requirements.

GASSP and GAISP Overview

Creation of the GASSP began in response to Recommendation #1 of the *Computers at Risk* report. Originally carried by the International Information Security Foundation (IISF), the GASSP has drawn from an array of existing guidelines, such as those created by the Organization for Economic Cooperation and Development (OECD) and the United Kingdom Department of Trade and Industry. As a global initiative,

participation and support have been gained from respected groups such as the International Information Systems Security Certification Consortium (ISC)², the International Standards Organization (ISO), the Institute of Internal Auditors (IIA) and the international Common Criteria effort.

The Information Systems Security Association (ISSA) decided, with the concurring support of the IISF, to take on the leadership needed to finalize and promote this important body of work as the Generally Accepted Information Security Principles (GAISP).

In the wake of the attacks of 9/11, the security industry has seen a dramatic increase in awareness and participation from the U.S. federal government, with new initiatives such as the Department of Homeland Security and the Partnership for Critical Infrastructure Security. Security professionals have an opportunity to work with these efforts and establish self-regulations rather than permit government-mandated security policies and regulations to fill the perceived void.

The GAISP is a comprehensive guidance hierarchy that provides a globally consistent, practical framework for information security. The final body of the GAISP will provide three levels of guiding principles to address security professionals at all levels of technical and managerial responsibility:

Pervasive Principles

Targeting governance and executive-level management, the Pervasive Principles outline high-level guidance to help organizations solidify an effective information security strategy.

Broad Functional Principles

Broad Functional Principles are the building blocks of the Pervasive Principles and more precisely define recommended tactics from a management perspective.

Detailed Principles

Written for information security professionals, the Detailed Principles provide specific, comprehensive guidance for consideration in day-to-day information risk management activity.

Privacy Considerations

The ability to collect and manage information doesn't necessarily confer the right to save, analyze, and publicize that information, but several recent attacks suggest that this is occurring. In one high profile case, an airline was approached and asked to turn over the records of millions of passengers who had purchased tickets for trips on the airline. Apparently the purpose was to combine the flight plans of customers with other data available commercially, such as reports from credit bureaus, and determine which fliers may fit the profile of a terrorist. This is feasible only if you can rapidly combine information from several different databases, and that, in the view of many, represents a massive invasion of privacy.

The concern that the compilation of more benign databases could provide potentially harmful information began at about the time commercial use of computers first accelerated. During the 1960s, the increasing availability of large-scale computers made possible for the first time the development and use of centralized, computerized databases. In 1965, recommendations were made for the establishment of a National Data Bank to serve as a central repository of all personal information gathered by federal agencies about U.S. citizens. This proposal awakened concerns about the computer's potential for invading individual privacy. Extended and heated testimony was heard before the U.S. House of Representatives Subcommittee on the Computer and Invasion of Privacy. There was considerable national discussion about the potential for abuse of centralized databases and about the need for legal action to protect society against such abuses.

Since the dawn of the computer age, there has been tension between, on the one hand, the technologies that enable huge amounts of information to be stored and accessed with accuracy and efficiency, and, on the other hand, the right to personal privacy. In the case of the airline issue, the defense agency that requested the data even used a private company to merge the lists and perform the analysis. This circumvented laws prohibiting the government from forming such databases. Private industry, even as a contractor to government, faces fewer such restrictions. This issue is explored forcefully in *Database Nation*, by Simson Garfinkel (O'Reilly).

Summary

Computer security, originally envisioned to protect the denizens and equipment contained in the “glass houses” of computer centers has grown into a network of laws, standards, and best practices designed to promote the protection of the computer and its network from internal and external attack, as well as the misuse of its contents by owners and protectors alike.

PART II

Computer Security

Computer System Security and Access Controls

Computer security covers a lot of territory: locking your server and telecommunications rooms, locking your machine, protecting your login accounts with strong passwords, using file protection and adhering to a regular backup schedule to keep your data from being destroyed, encrypting network communications lines, and using special shields to keep electromagnetic emanations from leaking out of your computer (TEMPEST). But when people talk about computer security, they usually mean what is called *computer system security*, which is a fancy way of saying data protection.

What Makes a System Secure?

In the most basic sense, computer system security ensures that your computer does what it's supposed to do—even if its users don't do what they're supposed to do. It protects the information stored in it from being lost, changed either maliciously or accidentally, or read or modified by those not authorized to access it.

How does computer system security provide protection? There are four primary methods:

System access controls

These methods ensure that unauthorized users don't get into the system and encourage (sometimes force) authorized users to be security-conscious—for example, by changing their passwords on a regular basis. The system also protects password data and keeps track of who's doing what in the system, especially if what they're doing is security-related (e.g., logging in, trying to open a file, using special privileges). System access controls are the soul of authentication.

The next section introduces the basics of system access controls. Appendix C describes the Orange Book accountability requirements, which specify the system access controls defined for different levels of secure systems. The Orange Book is still an important reference for computer security, although technically it has been replaced by the Common Criteria.

Data access controls

These methods monitor who can access what data, and for what purpose. Another word for this is *authorization*, that is, what you can do once you are authenticated. Your system might support discretionary access controls; with these, you determine whether other people can read or change your data. Your system might support mandatory access controls; with these, the system determines access rules based on the security levels of the people, the files, and the other objects in your system. Role-based access controls are a hybrid system; these methods extend individual authorization to group memberships.

System and Security Administration

These methods perform the offline procedures that make or break a secure system—by clearly delineating system administrator responsibilities, by training users appropriately, and by monitoring users to make sure that security policies are observed. This category also involves more global security management; for example, figuring out what security threats face your system and what it will cost to protect against them.

Chapter 5 introduces the basics of system security planning and administration. Appendix C shows the Orange Book system administration requirements defined for different levels of secure systems.

System Design

These methods take advantage of basic hardware and software security characteristics; for example, using a system architecture that's able to segment memory, thus isolating privileged processes from nonprivileged processes.

Although a detailed discussion of secure system design is outside the province of this book, the major Orange Book design requirements for different levels of secure systems are available in Appendix C.

System Access: Logging into Your System

The first way a system provides computer security is by controlling access to that system. Who's allowed to log in? How does the system decide whether a user is legitimate? How does the system keep track of who's doing what in the system?

Trying to log into a system is a kind of challenge/response scenario. You tell the system who you are, and the system requests that you prove it by providing information that matches what the computer has stored about you. In security terms, this two-step process is called *identification* and *authentication*.

Identification and Authentication

Identification is the way you tell the system who you are. *Authentication* is the way you prove to the system that you are who you say you are. In just about any multi-user system that involves local area networks, and in most desktop and laptop PCs,

you must identify yourself, and the system must authenticate your identity, before you can use the system. There are three classic ways to do so:

What you know

The most familiar example is a password. The theory is that if you know the secret password for an account, you must be the owner of that account. There is a problem with this theory: you might give your password away or have it stolen from you. If you write it down, someone might read it. If you tell someone, that person might tell someone else. If you have a simple, easy-to-guess password, someone might guess it or systematically crack it.

What you have

Examples are keys, tokens, badges, and smart cards you must use to unlock your terminal or your account. The theory is that if you have the key or equivalent, you must be the owner of it. The problem with this theory is that you might lose the key, it might be stolen from you, or someone might borrow it and duplicate it. Electronic keys, badges, and smart cards can be used as authentication devices and as access devices for buildings and computer rooms. Some of the most sophisticated new security tokens are physical devices that continually calculate new passwords based on time-of-day or according to secure algorithms. These passwords are similarly calculated back to the system for which entrance is sought, and the password from the petitioning party must match the password calculated locally.

What you are

Examples are physiological or behavioral traits, such as your fingerprint, handprint, retina pattern, iris pattern, voice, signature, or keystroke pattern. Biometric systems compare your particular trait against the one stored for you and determine whether you are who you claim to be. Problems sometimes occur with false positives and false negatives; occasionally, a valid user is rejected and an invalid user is accepted. In addition, a big problem with these authentication systems is that, on the whole, people aren't comfortable using them.

Multifactor authentication

Multifactor authentication is a way to cascade the three methods listed previously such that if an attacker gets past one safeguard, they still have to pass another. Passwords are still, far and away, the authentication tool of choice. In a multifactor authentication system, username and password would be augmented with one of the other two systems. In fact, almost every time authentication devices such as tokens and biometric devices are used, they usually supplement, not replace, conventional login IDs and passwords. See Chapter 9 for additional discussion of authentication devices.

Login Processes

In most systems, you identify yourself to the system by entering some kind of unique login identifier or username, followed by a password. The identifier is typically a

name, initials, department, or an account number assigned by the system administrator based on your own name and/or group. The password is typically a string of letters and/or numbers known only to you.

The exact method of the username/password authentication process varies depending on the network operating system. Generally, however, it involves one or more of the following:

Encryption

This method scrambles a password so that it cannot be deciphered by someone who monitors storage or transmissions. The opposite of encryption is *clear text*, in which password or other information is transmitted unmodified.

Challenge and response

With this method, the user is asked to authenticate at the beginning of the exchange, and frequently at random intervals thereafter.

The following subsections describe several models for these mechanisms.

Password Authentication Protocol

In Password Authentication Protocol (PAP), the user provides a username and password, and these are compared with values stored in a table to see if they match. PAP is essentially the same as the classic Unix login. With PAP, login information is transmitted in the clear, that is, unencrypted, or using clear text.

Challenge Handshake Authentication Protocol (CHAP)

The Challenge Handshake Authentication Protocol (CHAP) is a type of authentication in which the device doing the authenticating, usually a network server, sends the client program an ID value and a random number, and both the sender and peer share a predefined secret word, phrase or value. The client strings together the random value (or nonce), the ID, and the secret, and calculates a new value from them using what is called a *hash function*. This new value is sent to the authenticator, which has built the same string and calculated a similar hash. The authenticator compares the result with the value it received from the client. If the values match, the peer is authenticated. To increase security, the authenticator can be set to repeat the challenge-response process periodically throughout the communications session.

Mutual authentication

Mutual authentication can be thought of as two-way authentication. The client authenticates to the server, and then the server authenticates to the client or workstation. This allows the server to verify that the user is at an authorized workstation. If not, access is denied. Mutual authentication can prevent *masquerade attacks*, in which identity is pretended in order to gain access, or *man-in-the-middle attacks*, in

which an attacker represents itself to the server as a valid client, and to the client as the valid server, taking advantage of the information flow to steal secrets or inject malicious data.

One-time password

The one-time password (OTP) is a variation of the username/password combination. With OTP, the user creates a password, and the system creates a variation of the password each time a password is required. In this way, the same password is never used twice. With OTP, even if an attacker learns a password by snooping, he won't be able to use it again.

Per-session authentication

Setting up an authentication method that requires the client to reauthenticate for each exchange of information is burdensome, but it provides a great deal of security. One way to do per-session authentication is to increment a counter with every transmission. Because the password changes constantly, this form of authentication protects against attackers who snoop.

Tokens

A token can strengthen the authentication process by providing a “what you have” layer to the authentication process. A token or token card is usually a small device that supplies the response to a challenge that is received when trying to log on.

One type of a token is a credit card-size device with a built-in keypad. At login, the server issues a challenge with a number. The user keys this number into the token card, and the card displays a response. The user inputs this response and sends it to the server, which calculates the same result it expects to see from the token. If the numbers match, the user is authenticated.

Other tokens are based on time. They display a number that changes at regular intervals, usually several times each hour. The user logs in by entering her username and password, along with the time-based value from the token. If the value from the token matches a value the server has calculated, the account is authenticated, and access is granted.

The disadvantage of tokens is their small size and their price. If the token breaks or becomes lost, a replacement will be needed to gain access. Depending on the manufacturer and quantity purchased, tokens can run from \$30 to \$100 each. Software-based tokens are also available. At login, users enter a personal identification number (PIN), with which the soft token creates a one-time password. The PIN is never transmitted. The software token protects itself by refusing to work if an incorrect PIN is entered too many times.

Biometrics

Biometrics, using personal measurements such as fingerprints, hand outlines, iris and retina scanners, voice recognition, handwriting analysis, and keyboard analysis can be a one-stop shop for authentication, but it is seldom used that way. Biometric technologies are most effective if used as a “what you are” layer of authentication on top of a “what you know” or “what you have” layer. Biometric systems are prone to false positives (saying you are who you aren’t) and false negatives (saying you aren’t who you are), and most systems have an index of reliability that is calculated by combining the two ratios. Biometrics could be great for verifying a username or password, however, so look for more of it soon. Biometrics is covered in greater detail in Chapter 9.

Remote access (TACACS and RADIUS)

Even though broadband connections such as cable and DSL are becoming more widespread, dial-up type access is still widely used. To allow remote users to connect to the network, many businesses use modem banks that allow users to dial into the network. These connections must be secured and audited, and users must be able to be authenticated easily, in order to keep information available. But instead of a hundred or a thousand potential clients, anyone with a telephone and a computer becomes a potential login.

Specialized login and authentication systems have been developed to meet this need. Here are two of them:

- Remote Authentication Dial-In User Service (RADIUS)
- Terminal Access Controller Access Control System (TACACS)

Handling large amounts of login traffic can bog down a server. These protocols allow a network server to offload the user authentication and authorization to a central server. A typical enterprise network may have an access server attached to a modem pool, along with a RADIUS or TACACS server to provide authentication services. Remote users dial into the access server, and the access server sends authentication requests to the RADIUS or TACACS server. The RADIUS or TACACS server then authenticates users and authorizes access to internal network resources. Remote users are clients to the access server, and the access server is a client to the RADIUS server.

In addition to providing authentication, both services can track users, the duration of sessions, the protocols used during sessions, ports used, addresses contacted, and even the cause of session termination. Because these services can create logs, the audit trail can be examined to determine who is accountable for each individual session. Essential information about dial-up activity can be organized, charted, and examined. This means that authentication, authorization, and accounting functionality is often delegated to RADIUS or TACACS.

TACACS is documented in RFC 1492. RADIUS is defined in RFC 2139 (RADIUS Accounting, April 1997) and RFC 2865 (Remote Authentication Dial In User Service (RADIUS), June 2000).

DIAMETER

DIAMETER is a protocol that authenticates remote dial-up users, and also provides authorizations and accounting. DIAMETER grew out of the RADIUS protocol, which has limitations in how it can connect. RADIUS works only with modem-based protocols such as Serial Line Interface Protocol (SLIP) and Point to Point Protocol (PPP). In the past, RADIUS functionality was adequate; today's environment, however, uses smart phones and hand-held devices that feature different protocols, so more flexible access methods are required.

DIAMETER adds new features to RADIUS, such as the ability to ask for additional logon information beyond basic authentication. DIAMETER uses the initial logon and authentication to identify a user, and then queries the user for additional information that might be used to strengthen the authentication, or gives her access to additional systems.

DIAMETER supports the following extensions to various network services:

Roaming Operations (ROAMOPS)

Procedures, mechanisms, and protocols to support user roaming among groups of Internet service providers (ISPs). Global operations with a single account should be used with applications requiring roaming.

Network Access Server Requirements (NASREQ)

Extends network access server design beyond simple dial-up and into virtual private network (VPN) support, improved authentication, and roaming.

IP Routing for Wireless/Mobile Hosts (MobileIP)

Development of the needed routing support to allow IP roaming using either IPv4 or IPv6 over multiple subnetworks and to different types of media.

DIAMETER is described in various Internet Engineering Task Force (IETF) draft standards, including:

RFC 3589: Diameter Command Codes for 3GPP Release 5, September 2003

RFC 3588: Diameter Based Protocol, September 2003

RFC 2486: The Network Access Identifier, January 1999

RFC 2607: Proxy Chaining and Policy in Roaming, June 1999

Kerberos

One of the most important and complicated authentication schemes is a network authentication protocol developed in the 1980s at the Massachusetts Institute of Technology (MIT), called *Kerberos*. Kerberos provides secure transport across insecure media, such as the Internet. Named after the three-headed watchdog guarding the underworld in Greek mythology, Kerberos follows a three-step login process:

1. The user provides a username and password. This login request is sent to an *authentication server* (AS), which performs the user authentication. When the AS receives the login request, it creates two copies of a new key called a *session key*.

These session keys are used between the user and the resource she is trying to access.

One session key is encrypted with a key held by the user; the other is encrypted with a key held by the server. This server key is called a *ticket*. The AS sends both the user key and ticket back to the user. The user opens one copy using her key as the password, extracts the session key, and checks to see that it is valid. The user can open only the copy encrypted with her key because the other was encrypted with the server's key.

2. The user next creates another message, called the *authenticator*, adds to it the current time, and creates a *checksum*, which is a mathematical check performed on a series of characters in which the characters are added up (summed) and appended to the series. Subsequent users repeat the checksum process to determine if the checksum is identical. If not, the characters have changed in storage or transmission, and the data is usually discarded as a precaution. The user encrypts the checksum using the session key and then sends both the ticket and the authenticator to the server that is providing the desired resource.
3. The server receives both encrypted copies of the session key, uses its ticket (server key) to open the copy that was encrypted with that key, extracts the session key, and verifies it as coming from the user, which is determined by the fact the user's key is present. The other message is opened with the session key. The server next compares the timestamp and checksum to the current time to identify the user and check the integrity of the message. If everything is valid, the server grants the user access.

Easy, right? In truth, Kerberos requires you to understand a few more details, but this explanation covers the basics. Kerberos can be considered a response to the advance of technology. Early networks were concerned with communication. Security was second to reliability. Most traffic was sent in the clear. The advent of tools such as packets allowed operators to see any unencrypted traffic as it traveled across the wire, which was not acceptable. Kerberos tries to solve this problem by providing strong encryption and sure authentication. Even though it introduces considerable complexity, Kerberos is a model on which many similar systems are built.

Two good resources for Kerberos information are the MIT Weblink for Kerberos (<http://web.mit.edu/kerberos/www/>) and the IETF RFCs (<http://www.ietf.org/rfc/rfc1510.txt>).

Passwords

In spite of all the options, far and away the most common method for authentication is the username/password combination, as is found on any Unix, Linux, or Windows system. For example, Unix systems display the prompt:

login:

and expect a first or last name or some prearranged combination of both, such as first initial followed by up to seven letters of the last name, or any other handle you've been given, in response.

After you enter your login ID, the system prompts for a password:

Password:

You type the password (which is ordinarily never echoed, or displayed, on the terminal screen, unless it is by a series of asterisks, which may or may not correspond to the number of keystrokes). The system then authenticates your identity by verifying that the entered password is valid for your account. You'll be able to proceed only if the password you enter matches the password stored for you in the system.

Hints for Protecting Passwords

Both system administrators and users share responsibility for enforcing password security. Remember, password security is everyone's responsibility. In addition to damaging your own files, someone who uses your password to break into a system can also compromise all the files in the system or network.

- Don't allow any logins without passwords. If you're the system administrator, make sure every account has a password.
- Don't keep passwords that may have come with your system. Change all test, vendor, or guest passwords—for example, *root*, *system*, *test*, *demo*, and *guest*—before allowing users to log in.
- Don't ever let anyone use your password. If you must share it, say because you are at home, and someone at work needs it, change it as soon as possible, or consult a security administrator to create a new one until you can change yours.
- Don't write your password down—particularly on your terminal, computer, or anywhere around your desk. If you ever do write your password down, don't identify it as a password. Better yet, add a few leading or trailing characters, and write two or three more potential passwords along with it. Consider writing it backwards.
- Don't type a password while anyone is watching.
- Don't record your password online or send it anywhere via electronic mail. In his book *The Cuckoo's Egg*, Cliff Stoll reports how his intruder scanned electronic mail messages for references to the word "password."
- Don't keep the same password indefinitely. Even if your password hasn't been compromised, change it on a regular basis.

An old saying from the USENET: "A password should be like a toothbrush. Use it every day; change it regularly; and DON'T share it with friends."

Passwords are your first defense against intruders. To protect your system and your data, you must select good passwords, and you must protect them carefully.

This all assumes that you are the only user of your computer. If you share a computer with multiple users, say workers on different shifts, or supervisors who keep accounts on your machine to aid maintenance, any one of the passwords could be the target of the attack.

Hints for Picking Passwords

If you're allowed to choose your own password, pick passwords that are hard to guess. Here are some suggestions:

- Pick passwords that aren't words (English or otherwise) or names (especially your own, that of a fictional character such as Hamlet or Gandolf, or that of a family member, a pet, or a car).
- Pick a mix of alphabetic and numeric characters. Never use an all-numeric password (especially your phone number or social security number).
- Pick long passwords. If your password is only a few letters long, an attacker can easily try all combinations. Most systems insist that your password be at least 6–8 characters. Some systems support passwords of up to 40 or more characters.
- Pick different passwords for the different machines or network nodes you access.
- Mixing special characters such as & or \$ can help increase the pool of available characters a brute force attacker must cycle through. Be careful about using numbers that resemble letters, or 31337-speak, because the attackers know it well. Check with your system administrator to find which characters are allowed.

The best passwords contain mixed uppercase and lowercase letters, as well as at least one number and/or special character. The password you pick doesn't need to be gibberish. In fact, if it is, you'll be tempted to write it down, defeating the purpose of your careful selection. Some suggestions are:

- Combine several short words with numbers or special characters; for example, I;did3it.
- Use an acronym you've built from a phrase you'll remember. For example, the acronym for "When in the course of human events" is Witcohe, but that one could be guessed. It's better to pick a phrase that's not recognizable. For example, the acronym for "Oh no, I forgot to do it" is 0niftdi.
- Add a number or a special character for more security. For example :0nif;tdi or 0n5iftdi.
- Pick a nonsense word that's still pronounceable; for example 8Bektag or shmoaz12.

Protecting passwords

Access decisions are the heart of system security, and access decisions are based on passwords, so it's vital that your system protect its passwords and other login information.

Most security administrators protect passwords in three important ways: they make passwords hard to guess, they make login controls hard to crack, and they protect the file in which passwords are stored.

Making passwords hard to guess is a matter of user training and occasionally testing to see if the system yields to a common dictionary attack. Such penetration testing can smoke out users who have weak, easily guessed passwords.

Making passwords hard to steal is a matter of hiding the file in which they are stored, and in some cases, performing a one-way encryption called a *message digest*, or *hash* which stores the passwords in an encrypted form.

Protecting your login and password on entry

Most vendors offer a whole smorgasbord of login controls and password management features that the system administrator can mix and match to provide optimal protection of a particular system. Because these security features are commercially attractive and relatively easy to implement, most systems tend to have a lot of them. Examples of such features are shown in Table 3-1.

Table 3-1. Sample login/password controls

Feature	Meaning
System messages	Most systems display warning banners and announcement messages before and/or after you successfully log in. Formerly, the banners were friendly identifiers, but court cases showed that to be unwise. Some systems allow the system administrator to suppress these messages because they can provide a clue to an observer as to the type of system being accessed. If an intruder telnets in and finds out he's talking to a Solaris system, for example, that's a valuable clue.
Limited attempts	After a certain number of unsuccessful tries at logging into the system (the number can be specified by the system administrator), the system locks you out and won't let you log in from that terminal. Some systems lock you out without informing you that this has happened. Modern attack scripts overcome this by trying to open several sessions at once.
Limited time periods	Certain users or terminals may be limited to logging in during business hours or other specified times.
Incrementing login failure wait times	Each time a login fails, a longer time must pass before allowing another attempt. That is, after the first attempt, it takes one second to reset; after the second attempt, two seconds; after the third, four seconds; and after the fourth, eight seconds. This thwarts brute force attacks.
Last login message	When you log in, the system may display the date and time of your last login. Many systems also display the number of unsuccessful login attempts since the time of your last successful login. This lets you know if your account was accessed by someone else—for example, you may notice a login in the middle of the night or a pattern of repeated attempts to log in. If you weren't responsible for these attempts, notify your system administrator right away.
User-changeable passwords	In many systems, you're allowed to change your own password any time after its initial assignment by the system administrator and may be required to change it after a certain interval.

Table 3-1. Sample login/password controls (continued)

Feature	Meaning
System-generated passwords	Some systems require you to use passwords generated randomly by the system, rather than relying on your own selection of a difficult-to-guess password. Some systems let you view several random choices from which you can pick one you think you'll be able to remember. A danger of system-generated passwords is that they're often so hard to remember that users may tend to write them down. Another danger is that if the algorithm for generating these passwords becomes known, your entire system is in jeopardy.
Password aging and expiration	When a specified time is reached—for example, the end of the month—all passwords in the system may expire. The new passwords usually must not be identical to the old passwords. The system should give reasonable notice before requiring you to change your password; if you have to pick a password quickly, you're likely to pick a poor one. In some systems, the system administrator can respond to a security breach by forcing a particular password, or all passwords, to expire immediately. The system may keep track of your passwords for an extended period to make sure you don't reuse one that might have been guessed.
Minimum length	Because short passwords are easier to guess than long ones, some systems require that passwords be a certain length, usually six to eight characters, but longer is better.
Password locks	Locks allow the system administrator to restrict certain users from logging in or to lock login accounts that haven't been used for an extended period of time.

Protecting your password in storage

Every system needs to maintain its authentication data. Typically, valid passwords are stored in a password file. This file is accessed only under certain limited circumstances—when a new user is registered, when you change your password, or when you log in and need to be authenticated. As a security protection against imposters or compromised passwords, in most cases, administrators *cannot* access stored passwords. The best they can do is reset a password to a temporary value, such as “let-me-in” or “password,” and require a user to change the temporary password at the next login.

Protection of passwords is extremely critical to system security. Systems commonly use both encryption and access controls to protect password data.

Encryption

Most systems encrypt the data stored in the system's password file. Encryption (described in Chapter 7) transforms original information into altered information that usually has the appearance of random text. Encryption ensures that even if file security is somehow breached, the intruder won't be able to read the passwords in the file; they'll look like gibberish.

Most systems perform one-way encryption of passwords. One-way encryption means that the password is not intended to be decrypted—that is, deciphered into its original form. When the system administrator supplies you with your initial password, it's encrypted before it's stored in the password file. The original password is not preserved, not even in memory. Each time you log in and enter

your password, the system encrypts the password you enter and compares the encrypted version with the encrypted password stored in the password file to be sure you've entered a valid password. Remember too that the password is never displayed on the terminal screen.

Access controls

Even encrypted passwords can be cracked by a determined foe. Many systems store encrypted password data in files known as *shadow password* files, which have the most restrictive protection available in the system. In most systems, access is limited to the system administrator, usually by specifying only the administrator's ID in an ACL on the file. (See the discussion of ACLs in the section "Access Control in Practice.")

Password attacks

There are two primary methods of password cracking. The first is to try to guess a password by trying every possible combination of characters, one attempt at a time (a so-called *brute force attack*). Like everything else, this process has been automated. Crackers now use computers to do the guessing. In theory, the longer the password, the longer it takes to try every combination. For example, with a password containing eight random characters, there are 2.8 trillion combinations. Even a fast computer could take weeks to saw through all possible combinations.

The second method is a *dictionary attack*. The dictionary attack works because most users don't select random, or even decently secure passwords. Most users typically pick passwords that are laughably easy to guess—their initials, their childrens' names, their license plates, etc. Studies indicate that a very large percentage of users' passwords can easily be guessed. With the help of online dictionaries of common passwords (English words, names of people, animals, cars, fictional characters, places, and so on), crackers are quite likely to be able to guess a good many of the passwords most people are likely to choose. But if you select a good password (see the sidebar "Hints for Picking Passwords"), an intruder shouldn't be able to guess it—with or without a dictionary.

Authorization

Once you've been authenticated, the system uses your ID (and the security information associated with it) to determine what you're allowed to do in the computer or on the network. The process of determining your bounds is called *authorization*. For example, if you try to modify a sensitive file, the system checks your authenticated user ID against the list of IDs representing users who are authorized to read and write the data in that file. Only if your ID appears in that list will the system allow you to access the file.

Systems typically maintain a file containing information about your privileges and characteristics; in some systems, this is called a security profile, an authentication

profile, or a user list. Your profile might tell the system what your clearance is (e.g., SECRET), whether you’re allowed to change your own password, whether you can log in on weekends, whether you can run backup programs and other privileged programs, and a myriad of other information. In some cases, your profile is in the same file as the password list; in other cases, the information might be kept in separate files. In any case, it’s vital that your system protect this information; any compromise can jeopardize the security of the entire system.

One of the important pieces of information that appears in your authentication profile or user list is an indication of what kind of user you are. Most systems support several categories of users, or roles; a typical set includes regular users, a system administrator, and an operator. Highly secure systems may define a security officer as a separate category. Each category of user has specific privileges and responsibilities—for example, specific programs the user can run. The system administrator, for example, may effectively be able to do *anything* in the system, including overriding or circumventing security requirements and creating new, spurious accounts for the use of attackers. (The power of the system administrator is a major issue in secure systems; see the discussion of administrative controls and least privilege in Chapter 5.)

Sensitivity labels

Every subject and object in a system supporting mandatory access controls has a sensitivity label associated with it. A sensitivity label consists of two parts: a classification and a set of categories (sometimes called *compartments*). For example, the label shown in Figure 3-1 has both parts.

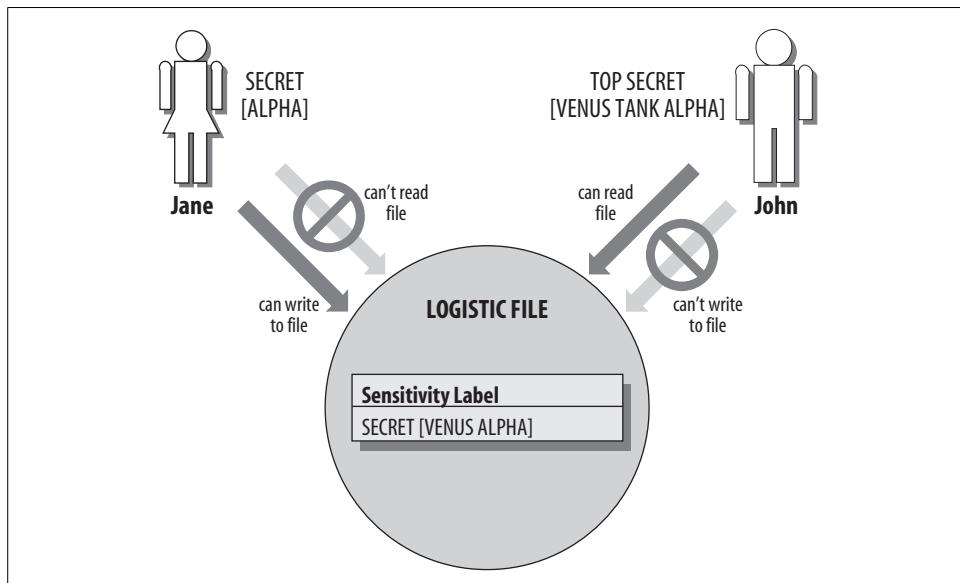


Figure 3-1. Sensitivity label that consists of two parts

The *classification* is a single, hierarchical level. In the so-called military security model (based on the Department of Defense multilevel security policy), there are four distinct levels:

TOP SECRET
SECRET
CONFIDENTIAL
UNCLASSIFIED

Each classification is more trusted than the classification beneath it.

The actual definition of classifications is up to a site's system administrator or security officer. If your site is processing government classified information, your labeling setup has to support the military model. But in general, labels represent whatever set of classifications and categories make sense in your own environment. For example, at a commercial site, you might define this corporate hierarchy:

CORPORATE
BRANCH
DEPARTMENTS

Or you might define gradations in levels of trust:

EYES ONLY
OFFICERS ONLY
COMPANY PROPRIETARY
PUBLIC

Or:

RESTRICTED
CONFIDENTIAL
PUBLIC

The *categories* or *compartments* are nonhierarchical and represent distinct areas of information in your system. Together, the categories make up a category set or a compartment set. A set may contain an arbitrary number of items.

In a military environment, you might have categories such as:

ANTI-TERRORISM
TANK
SUB
VENUS
STEALTH

In a commercial environment, your categories might correspond to company departments, product names, ad campaigns, or any other setup you wish to implement:

ACCOUNTING
PR
MARKETING

The idea is that even someone who has the highest classification isn't automatically cleared to see all information at that level. For example, to see information in the ANTI-TERRORISM category, you must "need to know" that information.

Compartmentalized Information

Military information is often compartmentalized as well as rated for its sensitivity. Think of compartmentalization as a *need to know*. Even if an individual has a high security clearance, if there is no plausible need to know the facts of some matter or another, the data is off limits. This was once demonstrated rather abruptly by a Marine sergeant who was the crew chief of a helicopter equipped with a secret electronics package, presumably used for intelligence.

One day a curious captain walked into the 'copter, and the sergeant summarily pushed him back out the door and down the stairs, spilling him onto the deck. This would not normally be considered a wise career move. However, the sergeant was likely demonstrating good security practice. Rank—Captain versus Sergeant—was in this case trumped by need to know. The sergeant knew good and well that the captain had no need to know about the special package on his ship, and took immediate precautions against compromising it.

In spite of the physical nature of the encounter, the fine traditions of the Corp were apparently upheld. As he sent the captain sprawling, the sergeant was heard to shout "Sorry, Sir!"

Access models

There are two general models of access you are likely to encounter. The Bell-LaPadula model is optimized for confidentiality and is used in defense circles; the Biba model is optimized for integrity and is used in financial transactions and business.

Bell-LaPadula model. In 1973, David Bell and Leonard LaPadula first described the DoD multilevel military security policy in abstract, formal mathematical terms, as mentioned in Chapter 2. In their Bell and LaPadula model, they used mathematical notation and set theory to define the concept of a secure state, the modes of access, and the rules for granting access. They called it the Basic Security Theorem. In that model, a subject's (usually a user's) access to an object (usually a file) is allowed or disallowed by comparing the object's security classification with the subject's security clearance. The three basic rules are as follows:

- The *-property (star property)
- The simple security condition
- The tranquility property

The **-property*, or *star property*, states that a subject is allowed write access to an object (usually a file) only if the security level of the object is greater than or equal to the clearance level of the subject. This makes it impossible for data from a highly cleared subject to become available to users with a lower security clearance in an object (file/directory) with a low security level. Without this rule, a user with a high security clearance could copy sensitive data into a low security clearance document—thus allowing “confidential” data to be *written down*, or to flow from a “top secret” to an “unclassified” level. Sometimes this is called the *write up rule* or *no write down rule* or the *confinement rule*.

The *simple security condition* is so named because it is not hard at all to grasp. The simple property states that a user (a subject) may only read a file (an object) if she has a security level equal to or greater than that of the file. It means that someone with a “secret” security level cannot read a file with a “top secret” security level; but can read a file with a “secret” or “confidential” security level. Sometimes this is called the *read down rule* or *no read up rule*.

The *tranquility property* states that the security level of an object cannot be changed while it is being processed by a computer system. This keeps a program or attack from modifying the sensitivity of a file while it is open and vulnerable.

Biba model. The military model is not the best for all cases. Another model you should know about is the Biba model, sometimes called the *Bell-LaPadula upside down* model. In business and banking, the integrity of data (that is not slipping a decimal point) may be of greater importance than secrecy. (For instance, if one purchases 10 million shares of a company’s stock, it will likely be in the financial pages soon as a major transaction, so secrecy is not terribly likely. But during the transaction, it would not be good to shift the decimal a spot and change each share’s price by 10 cents, which would alter the sale price by an amount even Bill Gates might notice.)

Whereas in Bell-LaPadula, a highly secured individual is not allowed to write documents with lower security to avoid letting information flow down, the Biba model denies a user of a lesser sensitivity from writing documents of a higher sensitivity. It is assumed that the accuracy and trustworthiness of data increases with the sensitivity of the document. To *write up* would be to risk contaminating higher integrity data with lower integrity data. Similarly, for a higher sensitivity user, to read lower integrity data could result in polluting a higher integrity document.

Access Control in Practice

Fortunately, you, the user, are isolated from the mathematical intricacies of access control models. These principles are codified in one of several access control mechanisms.

Why Access Controls?

If you work alone on a PC, you don't need to worry about access controls. You own all of your files, and you can read and write them as you wish. If you want to share a file with someone, you can copy it to a disk, open a network share to a folder in which you store the file, save the file on a common area of a server, or burn a CD-ROM or DVD and hand it over.

With shared computers, it isn't as easy. As soon as you begin to work on a system that supports multiple users, you must start worrying about data protection and access controls. You may not want every user in the system to be able to read your files. You certainly won't want them to change your files.

Even if you trust everyone on the system to keep away from your data, you need to protect against accidents. Suppose you and Joe Slow are working in separate areas of a shared directory. Both of you are working on the same projects, and you've both picked identical names for some of your files. With a few unfortunate keystrokes, Joe can change directories and delete one of your files, thinking it's his own. If your files aren't protected in any way, the system doesn't put any obstacles in his path. Access controls on files (they usually apply to other system objects, such as directories and devices as well) provide protection against such disasters.

There are at least three basic types of access controls that provide different levels of protection to the files in your system:

- Discretionary access control (DAC)
- Mandatory access control (MAC)
- Role-based access control (RBAC)

With *discretionary access control* (DAC) you decide how you want to protect your files and whether to share your data. With the more complex *mandatory access control* (MAC), the system protects your files. In a MAC system, everything has a label; using the security policy relationships established for your organization, the system decides whether a user can access a file by comparing the label of the user with the label of the file. Role-based access control (RBAC) grants you certain privileges based on your job function. If you are an accountant, you are allowed access and control similar to the other accountants. The following sections describe these access control types.

Discretionary access control

Discretionary access control is an access policy that restricts access to files (and other system objects such as directories and devices) based on the identity of users and/or the groups to which they belong.

What's discretionary about discretionary access control? In contrast to mandatory access control, in which the system controls access, DAC is applied at your own discretion. With DAC, you, the file owner, can choose to give away your data; with MAC, you can't.

Not only does DAC let you tell the system who can access your data, it lets you specify the type of access allowed. For example, you might want to let everyone in the system read a particular file, but you might want only yourself and your manager to be able to change it. Most systems support three basic types of access:

Read

If you have read access for a file, you can read the file.

Write

If you have write access for a file, you can write (change or replace) the file.

Execute

The execute permission is relevant only if the file is a program. If you have execute permission for a file, you can run the program.

Ownership. There are many types of discretionary access control. One simple method involves ownership of files, directories, and devices.

If you create a file, you're the owner of the file. Your login ID, or some other identifier, is entered in the file header. A system might base all its access decisions on file ownership. If you're the owner of the file, the system lets you read and change the file. If you're not the owner, you have no rights to the file. This is a simple scheme, but not a very practical one. For one thing, it doesn't let you share the file.

Virtually every system keeps track of file ownership and bases many access decisions on it (for example, regardless of other mechanisms, the system might let you delete a file only if you're the file owner).

Self/group/public controls. In many systems, you control access to your files by dividing the world of users into three categories and telling the system what the users in each category can do to your file. Some systems call these *self/group/public* controls. In Unix, they're called *user/group/other* (UGO) controls:

Self

You—the creator or owner of the file.

Group

A set of users. For example, all the users in a particular department may be in the R&D group.

Public

Everyone else—users other than you and the other members of your group.

File permissions. Each file has a set of bits called *file permissions* associated with it. File permissions often have the meanings shown in Figure 3-2.

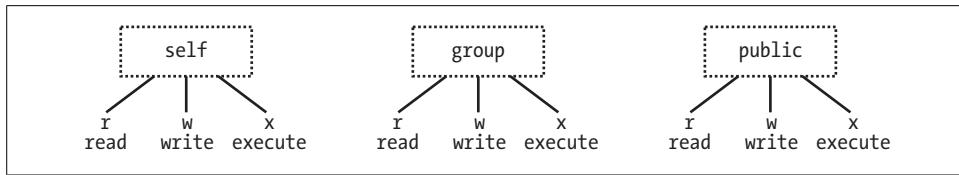


Figure 3-2. Self/group/public controls

If you list your files (with the correct option) in a Unix or Linux system, you'll see such file permissions as the following:

```
-rw-rw-r-- 1 frank r&d 81904 Nov 7 13:25 UPDATES
```

If a dash (-) appears in place of a permission, the user does *not* have the corresponding permission to read, write, or execute the file. For instance, in the example, the file owner (frank) can read and write the UPDATES file (rw-), members of the file group (r&d) can read and write the file (rw-), and the rest of the world can only read the file (r--). (Ignore the first dash above; it has a special meaning to Unix.)

Consider a few more examples.

The CHESS file contains a game; its permissions look like this:

```
-rwxrwxrwx 1 libr games 61799 May 19 10:11 CHESS
```

Everyone can read, write, and execute this file.

The SRC95 file is a segment of code that several people in the r&d group are working on; its permissions look like this:

```
-rwx-rw--- 1 sarah r&d 55660 Dec 19 11:42 SRC95
```

The owner and the other members of the group can read and change the file; no one else can access it.

The self/group/public controls are a good way to protect files. But what happens if you need to protect a file in different ways for different users, or if you want to keep one user from accessing a file?

If Sarah owns the FLAG file and wants Joe (a member of her group) to be able to read and change it, she'll specify the following permissions:

```
-rw-rw--- 1 sarah r&d 22975 Jan 19 10:14 FLAG
```

If Sarah wants Joe to be able to read FLAG, and Mary to be able to read and change it, she could make Mary the owner of the file (with r and w permissions) and leave Joe as a member of the group that can read (r) only. But what if Joe's group contains other users who aren't trusted enough to read FLAG? How can Sarah exclude the sinister Sam, for example?

Discretionary access control may seem burdensome, but it is highly flexible. With some complicated maneuvering, it's possible to accomplish these goals with self/group/public controls, but the more special cases you have, the more unwieldy this kind of file access becomes. The system we will explore next offers little such flexibility.

Mandatory access control

Mandatory access control is an access policy supported for systems that process especially sensitive data (e.g., government classified information or sensitive corporate data). Systems providing mandatory access controls must assign sensitivity labels to *all* subjects (e.g., users, programs) and *all* objects (e.g., files, directories, devices, windows, sockets) in the system. A user's sensitivity label specifies the sensitivity level, or level of trust, associated with that user; it's often called a clearance. A file's sensitivity label specifies the level of trust that a user must have to be able to access that file. MACs use sensitivity labels to determine who can access what information in your system.

Together, labeling and MAC implement a *multilevel security policy*—a policy for handling multiple information classifications at a number of different security levels within a single computer system.

In the past, military systems were set up to handle one, and only one, level of security. The policy for such systems, known as a *system high policy*, required everyone who used the system to have the highest clearance required by any data in the system. A system that handled SECRET data, for example, could not be used by anyone who did not have a SECRET clearance (regardless of how well protected the SECRET data was).

Although many government sites still operate in system high mode, others support multiple security levels. This is done by dividing, or compartmentalizing, the data. Such systems support simultaneous use by users with the highest and the lowest clearances (or, potentially, with no clearances at all), and with access to many different types of sensitive compartmented intelligence (SCI).

Although this section tries to touch on the major issues involved with labeling, security levels, and MAC, these are complicated topics, with a lot of security history behind them. Appendix C discusses briefly how the Orange Book specified requirements in these areas.

Data import and export. In a mandatory access control system, it's very important to control the import of information from other systems, and the export of information to other systems. MAC systems have a lot of rules about data import and export. They also control which system devices you can use to copy and print information; for example, you might not be allowed to print sensitive information on a printer located in a public area of your building. There are also rules for labeling devices and printed output (with banner pages and page headers and trailers). The discussion of specific labeling requirements is discussed in Appendix C.

Access decisions

In a MAC system, all access decisions are made by the system. Unlike DAC, which allows you to specify, at your own discretion, who can and cannot share your files,

MAC puts all such access decisions under the control of the system. The decision to allow or deny access to an object (e.g., a file) involves an interaction between all three of the following:

- The label of the subject—for example, your clearance:

TOP SECRET [VENUS TANK ALPHA]

- The label of the object—for example, a file named LOGISTIC with a sensitivity label:

SECRET [VENUS ALPHA]

- An access request—for example, your attempt to read the LOGISTIC file.

When you try to read the LOGISTIC file, the system compares your clearance to the file's label to determine whether you'll be allowed to read it.

Although multilevel security systems provide many benefits, they're the source of many frustrations as well. For example, under some circumstances, you may write a file and then find you can't read it!

Role-based access control

Role-based access control determines a user's access based on that user's role. For instance, financial managers may need to access all accounting data: taxes, payroll, receivables, collections, credit. A clerk in accounts receivable, however, will need only a subset of the accounting data, and an engineer over in R&D will need very little of it. The role a user is assigned to is based on the *least privilege concept*. The role is defined with the least amount of permissions or functionalities that is required to get the job done. If the privileges for a role change, permissions can be added or removed. This offers greater flexibility by changing the role instead of changing the user's permissions.

Since a user can belong to more than one role at a time, there is the chance for conflict. One role may allow access to a resource where another role denies access to that resource. Conflicts like this need to be corrected on a user-by-user basis. As a general rule, the least permissive combinations are used.

Access control lists

It would be reasonable to ask how the models mentioned in the previous section are implemented. Access control lists (ACLs) are lists of users and groups, with their specific permissions. They offer a more flexible way of providing discretionary access control. ACLs are implemented differently on different systems. For example, in a Unix-based trusted system that uses a Unix security, you'd protect PAYROLL with ACLs in the form:

```
<john.acct, r>  
<jane.pay, rw>
```

where:

- john and jane are login IDs of users who are allowed access to the PAYROLL file.
- acct and pay are group IDs of the users.
- r and w indicate the type of access allowed; r means that the user can only read PAYROLL, w that he or she can also change it.

If john is in the acct group, he can only read the file. If he belongs to any other group, by default he has no access. Similarly, if jane is in the pay group, she can read and write the file.

ACLs usually support wildcard characters that let you specify more general access to files. For example, you might specify:

`<*.*, r>`

to indicate that any user (*) in any group (*) can read (r) the file. You might specify:

`<@.*, rw>`

to indicate that only the owner (@) of the file can read (r) and modify (w) it.

In some systems, you can indicate that a particular user is specifically *not* allowed to access a file—for example, by specifying a null character or the word none or null in place of an access character such as r or w.

`<sam.*,none >`

Least and Most Permissive Access

There is a good example of an access conflict hiding in the example about john and the PAYROLL file. The ability of user john to access the file is affected by which groups he belongs to. As long as he is a member of only the acct group, he can read PAYROLL. Once he is part of an additional group, he loses access.

This conflict arises because of the *least permissive principle*. Cumulative permissions are handled differently by different network operating systems, such as NetWare, NetWare using Novell Directory Services, WindowsNT, and Windows 200x using Active Directory Services.

Fortunately, understanding the Unix UGO system will prepare you for any of the others. You will recognize the principles and will just need to learn their new names and how they are handled in that particular system. Also, this is an important but somewhat theoretical concept. In the real world, most administrators handle these issues by cloning a user who has the appropriate permissions and then changing the username. Don't skip over this issue lightly: you will see access lists again, when you program perimeter defensive systems such as firewalls, firewall software packages for routers, and intrusion detection systems.

Directory Services

As I described previously, several systems are used to keep track of authorized users. Some of these systems, such as biometrics, are still waiting to catch on. Others, such as the username/password, are getting fairly aged. One of the improvements to occur in login technology is the integration of Authentication, Authorization, and Accounting (AAA) technologies. These improved directory systems keep track of the identity of each user, as well as attributes of that user, such as contact information, personal attributes, including what they can access, in some kind of directory services database. A *directory service* is essentially a large database that stores information about all the objects in a network and information about how these objects interact. Objects can be users, groups, hardware, or software.

One common directory service is Active Directory, used by Windows servers and domain controllers. The Active Directory implementation helps to organize all objects and resources in the network environment while acting as the central point for security as well. Active Directory provides a means to audit all the events that happen on login servers (domain controllers). Audit policies can be set to monitor specific activities, create reports, and notify specific personnel of these events. Group policies can be defined that govern groups of users and computers. These policies can be applied to sites, domains, or organizational units that are defined within the Active Directory structure.

Most current authentication services depend on a system of parameters defined in the ISO X.500 standard that defines a standardized way to keep a user and her attributes and permissions easily accessible. The X.500 standard is a comprehensive system, so much so that an X.500 directory is usually accessed using a lighter, simpler tool called Lightweight Directory Access Protocol (LDAP). This section describes X.500, and then discusses the use of LDAP, its applications, its problems, and its possibilities.

Email example

Perhaps the easiest way to gain appreciation for directory-based authentication is to consider the situation faced by email administrators in the days before LDAP and X.500. Imagine that a large company merges with another large company. Now imagine that the two companies use different email systems that are only nominally compatible. Every user must appear (have a mailbox) on each system, in order to log on. That means that two sets of administrators must frantically import the complete user list of the other company into each other's systems. This also means, however, that each user ends up with two mail clients that must be checked frequently. Both halves of the new company will have a hard time efficiently communicating with each other, to say nothing of the confusion that will be faced by clients and customers.

Obviously, one way to simplify this situation is to import all users into one or the other mail system. This may not be the best approach, however, because email servers seem to operate in clusters that communicate amongst themselves according to

users' locations and system load. Migrating to one system or the other may force massive infrastructure changes and the possible interruption of services as the system is cut over. It may be easier just to scale up the systems on both sides and then sort it out later, using some kind of gateway servers to translate between the two systems.

Using an LDAP directory, however, all users appear in a common database that the system automatically replicates to login servers all over the network. Mail stores can be forwarded as users move from city to city. Changes to personal information, such as surnames, or changes in preferred salutation, or job title, can be updated in one place, and the change will propagate to wherever it is needed.

About X.500

ISO X.500 is an international standard. There are two outstanding features of X.500:

- X.500 is object-based.
- X.500 uses a hierarchical structure.

X.500 treats every listed computer and user as an *object*. Server “Framingham 12,” backup operator “tdiaz,” and System Admin “IMJasmine” are all objects. Every object has attributes. “Framingham 12” may have the IP address attribute of 200.16.212.12. User tdiaz has the attribute of belonging to the Backup Operator group. Admin Jasmine may have the *firstname* attribute of Ina and the *lastname* attribute of Jasmine. Ideally, the database schema for each class of object should be consistent. If there is a need to modify the schema to accommodate items of data from one company that are not present in the other, this should not be an unbearable task.

The second outstanding feature of X.500 is its hierarchical structure. Individuals and computers are given *common names* (CN). A common name is the name you or I might call something. Objects with common names are gathered into container objects called *Organizational Units* (OU). OUs are combined to form objects called *organizations* (O). The organizations in the hierarchy are grouped into objects called *countries* (C). Countries exist on the root, which is the top of the filesystem. This hierarchical structure is spelled out in Figure 3-3.

The directory information tree (DIT) is not intended to be a general-purpose database. It is optimized for frequent reading but infrequent writing. It resembles a database, but the standard was developed to store information about objects relevant to telecommunications.

The advantage of the X.500 hierarchical system is that it standardizes the storage of files regarding user attributes and permissions. This makes it easier to share directory data or to copy it. Whether an object is a single user or an entire corporate division, copying addresses is as easy as directing that a copy of the required object be moved into the new directory tree, and all the subsidiary objects will follow. And once the copy gets moved, it is readily usable.

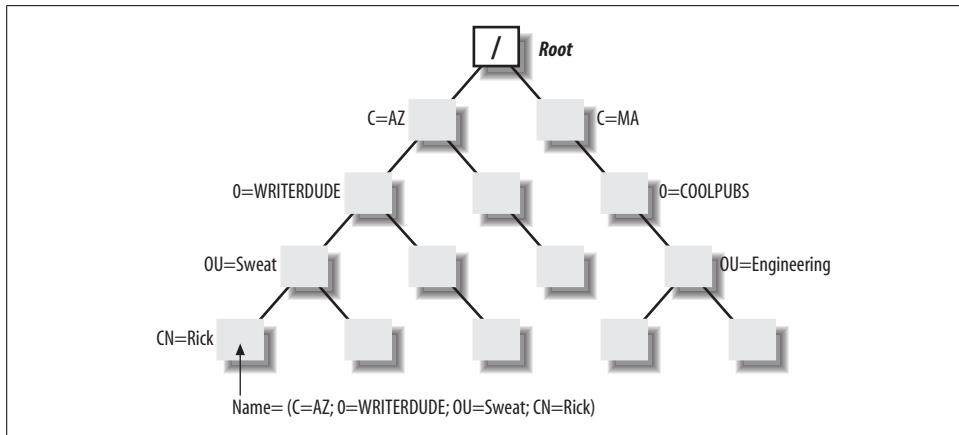


Figure 3-3. Discretionary access control with an access control list

Each X.500 site has its own directory system. The administrators at each site need to keep only their own site's data up to date. Users access the local X.500 system only. If the person being sought works at the local site, their entry will be found in the local X.500 system. If the person works at a remote site, the local X.500 system automatically contacts the remote X.500 system to retrieve the information. X.500 switching between local and remote access is transparent to the user. If the X.500 access to the remote sites proves to be too slow, the systems can be configured to automatically copy data to each other, and to regularly update the data. This process is called *replication*.



A key point of these directory-based systems is that they center on the individuals that wish to access the system, rather than on a static list of things such as passwords. Individual users are mapped in the directory according to where they fit into the larger scheme of organizations and enterprises.

The X.500 directory system can store lots of information, such as telephone and fax numbers, physical addresses, computer addresses, positions, responsibilities, and so on. The system can also be extended to store customized information. This large amount of data, and the inherent complexity of X.500, has led to the broad acceptance of a lightweight directory access protocol, called LDAP.

Lightweight Directory Access Protocol

The Lightweight Directory Access Protocol (LDAP) standard defines a network protocol for accessing information from a directory. When it became apparent that directories would need to be standardized, IBM, Microsoft, Lotus, and Netscape agreed to support LDAP. LDAP was designed to adapt the complex X.500 directory system to the needs of modern networks, including the Internet. With LDAP, directory servers

X.500 History

The X.500 directory is specified in the ISO/IEC 9594 multipart standard and in the ITU-T X.500 Series of Recommendations. There have been four versions of the X.500 standard.

1988 edition (First edition)

The first edition was issued as the multipart standard ISO/IEC 9594:1990 and as the CCITT X.500 (1988) Series of Recommendations.

1993 edition (Second edition)

The second edition was released as ISO/IEC 9594:1995 and as ITU-T X.500 (1993). This edition had some useful functions, such as shadowing of directory information, access control, and a significantly expanded information model and administrative capabilities.

1997 edition (Third edition)

The third edition was ISO/IEC 9594:1998 and ITU-T X.500 (1997). It provided several minor and some major extensions. It added a feature called *contexts*, which allowed information to be distinguished according to the context in which it was being accessed. Also added was OSI management of the directory. Important security features were also added and extended.

2001 edition (Fourth edition)

The fourth edition was issued as ISO/IEC 9594:2001 and as ITU-T X.500 (2001). This edition has several major extensions. It adds service management features, mapping-based matching, families of entries, and support for a pure TCP/IP stack.

run on host computers on the Internet, which means that client programs that are LDAP-compliant understand the LDAP protocol, and can log on to the LDAP server and look up directory entries.

LDAP servers first index all the data in their entries. Then, when they process a *query*, they run the data through *filters*. These are short, user-configurable statements that describe what should be included or discarded. The filters weed out unneeded entries, leaving the user with just the selection she wants. It can be a person, a group, or an exotic set, such as “every user in the Pacific Northwest who says they have a portable 802.11 appliance.” Users can also select how much data they want returned in their queries so that their answer fits their needs, such as last name, first name, business title, email, and phone number.

The LDAP namespace. In the LDAP protocol, each listing in the directory is called an entry. Each entry can have one or more attributes. Each attribute has a type, and one or more values. Here’s an entry:

```
cn = mylittletestentry
objectclass = person
```

Where `cn` stands for `commonName`; `mylittlestentry` is the value of `commonName` for this entry; and the entry is given a type by use of an `objectClass` attribute.

The system and users tell one entry from another by giving it a name. Entries are named by one of their attributes. `cn=mytestentry` is this entry's *relative distinguished name* (RDN).

Hierarchy. The LDAP namespace, like X.500, is hierarchical. This means that the full name of an object must include the path of the object. This full name is called the *distinguished name* (DN). (It may help to think of "distinguished" in this case as meaning "fully and uniquely identified" rather than "outstanding, noteworthy, or gray-haired.")

LDAP storage capabilities. Almost anything can be stored in an X.500 directory. This can include text, ID photographs, biometric information used for identification and authentication, web addresses, FTP addresses, and other pointers. However, the more information is stored, the more memory is required.

Different types of data are held in attributes of different types. Each attribute type has a particular syntax. User-defined attributes, syntaxes, and object classes, make it possible for security administrators to make the directory fit their needs precisely.

LDAP is particularly useful in situations where the Internet is an integral part of the network or its authentication scheme. LDAP contains protocols that allow it to be self-replicating over multiple sites. These updates travel over the network as if they were regular network traffic.

Identity Management

One would think that with X.500 and LDAP, that directory-based login had reached perfection. Such is not the case. What happens to a given user who must interact with four or five networks in the course of a day? The administrative burden is heavy, even just keeping up with all the logins and the required changes and rotations. The burden is shared by network administrators, who have to maintain several databases for user logins and keep them all in sync. It would be much easier if a user could be authenticated and authorized by one process in a single database, and then have the network generate appropriate logon tokens or certificates that could serve many purposes. This would roughly be the equivalent of the Microsoft Passport system, in which a single logon authenticates the user in many services.

This emerging trend in single-point authentication and authorization is called *identity management*, or sometimes *federated identity management*. It is called federated because it provides unification—one standard login serving across several different applications, say a network logon, telephone long distance accounting, an email password, and a PIN, to gain access to secured areas.

At the root of identity management is a human need to reduce the burden of tracking multiple passwords; an economic need to avoid supporting and maintaining several different databases for the same pool of people; and a legal need to protect the privacy of users who may have private information, such as emergency medical information or information on disabilities or health conditions, stored in company directories. For instance, a switch could ensure that employees who are visually handicapped will receive corporate email announcements in a format suitable for text-to-speech software.

Software and standards for building interoperable identity-management systems are evolving rapidly; for example, the Security Assertion Markup Language (SAML) provides a common path to share end-user credentials. An example of an early deployment of a SAML integration application is a project between Boeing and Southwest Airlines that gives the airline's mechanics single sign-on, web-based access to repair manuals stored on Boeing's corporate networks. Service Provisioning Markup Language (SPML) is another new protocol nearing ratification. It is important because it promises to integrate systems that provision user accounts.

Financial and legal pressures

Organizations today face increased regulatory pressure to protect user privacy, ensure the accuracy of corporate financial data, and audit and log their efforts to ensure compliance with new laws. The laws of greatest concern at present are the Sarbanes-Oxley Act (adherence to standards in financial record keeping), the Gramm-Leach-Bliley Act (protection of personal financial data), the Health Insurance Portability and Accountability Act (secure transmission and storage of medical records), the Family Educational Rights and Privacy Act (privacy of student records), and the U.S. Patriot Act (antiterrorism and law enforcement). In the United Kingdom, there is the U.K. Data Protection Act and Regulation of Investigatory Powers Bill, as well. Businesses are struggling to adapt to these sometimes sweeping changes, especially regarding issues of privacy. It is far easier to control personal information in one well-secured database than it is when protected information is spread over a series of disparate databases. As owners and executives wake up to the possibility of fines and perhaps even imprisonment for failure to heed these new laws, the executive visibility of advanced management and control of records and data increases in importance and funding.

Summary

This chapter began with an introduction of simple username/password authentication systems, and continued into various authentication and authorization systems, including RADIUS, TACACS, DIAMETER, and Kerberos. The development of authentication, authorization, and accounting system continued on to the directory-based systems, X.500 and LDAP. Finally, we discussed the wave of the future, identity management.

Proper passwords are an effective first layer in the organization's security policy. It is not difficult to develop strong passwords and change them frequently, but users tend to resist the practice, or else subvert it by writing the passwords down and leaving them where they can be seen. Multifactor authentication schemes, including biometrics, promise to greatly increase the security of networks and facilities.

If the need for increased security is needed, tokens can be added to the login process. This multifactor authentication method adds yet another layer to the defense in depth concept. Other factors discussed included using one-time passwords, Kerberos, and biometrics as a secure means to be authenticated.

Current directory-based login systems offer authentication, authorization, and accounting. As multiple authentication databases merge into federated identity management systems providing single-point login, increasing amounts of private information about users can be incorporated into the login process.

Viruses and Other Wildlife

The word *virus* has become a generic term describing a number of different types of attacks on computers using malicious code. Just about everybody has heard of computer viruses, worms, Trojan horses, and other malicious software. Many have been infected at least once, either by one of the famous attacks such as Melissa, ExploreZip, MiniZip, Code Red, NIMDA, BubbleBoy, I LoveYou, NewLove, Killer-Resume, Kournikova, NakedWife, or Klez; or perhaps by a lowly pest picked up in an email or through visiting some web site. A virus or worm could even be active in your machine right now, lying dormant until some trigger activates it. And in today's world, the line is blurred between viruses and marketing tools, such as pop-ups, adware, or spyware, each of which uses a certain amount of the computer's resources to display or gather data about the user.

Financial Effects of Malicious Programs

Although estimates vary widely, it is a safe bet that billions of dollars worth of damage have been done over the two decades since malicious code hit the big time. While some of this harm has been due to destruction of data and even damage to hardware, the bulk of the loss is likely lost time.

Spending time recovering from a virus steals opportunity in a few ways:

- The time and effort it takes to takes to root out the virus and repair the damage.
- The diversion of time and effort from what may have been revenue production.
- The out and out loss of computer hardware (rare these days) or documents, files, and applications that either cannot be recovered, or for which the time and expense of recovery can't be justified.

Putting an actual dollar value on the loss due to viruses requires a lot of guesswork. This is because one affected machine may be used mainly for trivial pursuits and can be easily repaired by simply reloading the machine with fresh copies of applications and any available data, while another may contain the details of an elaborate business deal, your contact book, your masters' thesis, or digital pictures of your loved

ones, which must be carefully located, scanned for infections, and copied onto the machine after replacing or repairing any infected files.

Thus the hard number of actual dollar losses due to malware activity is obscured by emotional losses. Just because the actual amount of damage is fuzzy, however, does not degrade the importance of malicious software.

Viruses and Public Health

A primary reason today to care about malicious code, such as viruses and worms, is the same reason it is important to keep one's vaccination records up to date while traveling. You would not want to become a carrier of some awful disease. Most malicious code today is concerned not only with trashing your machine, but also in using your machine to infect others.

A classic example is the software used to create a DDoS attack. After hiding itself in your computer, modern malware typically seeks information from you to use to infect others, and it usually finds it in your address book or by prowling your local area network. The malware then stalks its new victims, often by sending an email in your name and infects them as well. This is akin to the way a virus may propagate itself in a living organism, and it may be a separate operation from executing the payload—performing the dirty work of the virus—which follows.

Upon command, your machine and all of those you have inadvertently helped infect may then zero in on some target (such as the White House, or a critical server that helps make the Internet function) with tens of thousands of commands and requests coming from all directions. This brings the target to its knees, accomplishing the attacker's goal. To avoid this, every machine must be a healthy one, so that it does not unwittingly infect its neighbor, which may be you.

Viruses, Worms, and Trojans (Oh, My!)

While historically important, the classification of malicious code into categories such as “virus” or “worm” is today somewhat quaint. Attackers who want to harm your system will get there any way they can and whipping up a software half-breed that blurs definitions would be the least of their worries.

For this reason, modern attack tools tend to be labeled by their function more than their genealogy. Hence there are *rootkits*, *Trojan horses*, *exploits*, *password sniffers*, and *zombies*, more than there are viruses and worms. Although we will briefly explore the historic roots and definitions of such malicious programming in a moment, in this book we shall call all such programs *malicious code*, or for short, *malware*. And to keep in sync with the general public, we will use the term *virus* as its synonym.

A Malware Taxonomy

This chapter will define many modern attacks, but here are some brief definitions:

Denial of service attack (DoS)

Attack that produces so many requests of system resources in the computer under attack—such as calls to the operating system, or opening dialogs with other machines and then hanging onto the line to tie it up—that normal functions on the targeted computer are overwhelmed and cease.

Distributed DoS attack (DDoS)

DoS attack launched from many different computers, usually zombies hijacked for this purpose.

Exploit

Malware that capitalizes on known or undiscovered vulnerabilities, which are bugs or weakness in software applications or operating systems.

Rootkit

Malware, usually a small suite of programs, that install a new account or steal an existing one, and then elevate the security level of that account to the highest degree (*root* for Unix, Administrator for Windows) so that attackers can do their will without obstruction.

Script

File containing specific instructions of the attacker and commands to make them occur.

Sniffer

An attack, usually a Trojan horse, that monitors computer transactions or keystrokes. A keystroke logger, for instance, detects sensitive information by monitoring the user's keystrokes.

Trojan horse

Malware named for its method of getting past computer defenses by pretending to be something useful.

Zombie

A corrupted computer that is waiting for instructions and commands from its master, the attacker.

Knowing a little bit about germs may help you prevent the spread of disease. Similarly, knowing something about malicious code may help prevent the spread of its effects. Most viruses and worms tend to be upgrades or variations of others seen before. For this reason we will review the historic terms and how they came to be, before returning to the present to fight viruses in the here and now.

Viruses

A *virus* is a code fragment that copies itself into a larger program, modifying that program. Unlike a worm, described in the next section, a virus is not an independent

program but depends upon a host program, which it infects. A virus executes only when its host program begins to run. The virus then replicates itself, infecting other programs as it reproduces. After seeing to its own reproduction, it then does whatever dirty work it carries in its programming, or *payload*. There may not be a payload, or it might shoot blanks. Several powerful viruses have been uncovered that can spread at an alarming rate, but that don't really do much. Not to say they couldn't, of course, but their programmer chose to make them weak, or in some cases made a programming error that caused the payload portion to misfire. Nevertheless, every packet that traverses a network means that another packet has to wait its turn. Even if a virus is nothing more than a clever prank, sort of like graffiti on the computerscape, it still has negative impact. And because it is a rogue program turned loose within your computer, it is definitely an intrusion. It has gotten past your defenses, and has demonstrated that you are vulnerable until you take defensive measures.

While viruses generally see to their reproductive needs before their destructive ones, not all viruses launch at the first opportunity. A virus might start reproducing right away, or it might lie dormant for some time, until it's triggered by a particular event. For example, a particular triggering date may arrive (as with the infamous Friday the 13th virus), or a particular event may occur (for example, deleting a particular person's name from a payroll file, perhaps indicating that he's been fired). See the section on "Bombs" later in this chapter for more information about triggers.

Different pathogens may infect different parts to the body, and it is the same for viruses. A virus may infect memory, a floppy disk, a hard drive, a backup tape, or any other type of storage. Like a biological virus, a computer virus invades other organisms, causing those organisms to proliferate and spread the virus. It degrades system performance in addition to whatever other damage (e.g., data deletion) it does. At one time, viruses passed themselves about by hanging around in memory and jumping onto each new floppy disk that was inserted. Today, viruses (and worms too) can spread via a local area network, the Internet, or through the use of infected software on diskettes, recordable CDs, MP3 files, as email attachments, or in any situation in which computers lower their guard and trust each other.

While it is conceivable to catch a virus from electronic gathering places such as electronic bulletin boards or newsgroups, or via email or instant messaging software, viruses also can move about as macros, such as those written in the scripting language used to automate keystrokes in office programs such as Microsoft Word or Excel. As we shall see in Chapter 6, infected ActiveX files from the Web, or even Java code, can also be a source of infection.

Until recently, it was necessary for the user to activate a virus by running a corrupted program, such as opening an attachment to an email. Several email reader programs today have the ability to launch attached software automatically, such as music files or graphics. Attackers have figured out how to program malware into such files. As a result, when the user opens the email, the code executes.

Viruses and worms are often confused with each other. During a 1988 outbreak on the Internet, there was much debate about whether the invader was a virus or a worm. Even though we will discuss worms in a moment, in truth, the difference is somewhat academic. Both viruses and worms can deliver a payload, and because it takes processor time to open and activate them, either can result in system and network slowdowns.

Both viruses and worms can protect themselves by hiding in their host programs, by operating in delayed fashion, or by changing their physical characteristics to avoid detection. Sometimes, as is the case with rootkits, they also attempt to destroy the evidence of their wrongdoing. All this may hinder attempts to control and eradicate the offending code. And again, as is the case with a logic bomb that waits until a particular time or event to fire, some malware just entrenches itself and then lays low until called for by its master. A computer thus infected is often called a *zombie*.

Stealthy attackers create and collect zombies by running programming scripts that probe for vulnerable computers, using various attacks to corrupt them, and then at a time of their choosing, call them into play for their own purposes. This practice is so highly automated that many researchers report that an unguarded computer on the Internet is likely to become infected in this manner within a few minutes or hours. And yes, some zombies include code that will erase their malware after execution, making them difficult for experts to trace. Of course, the computer owners are usually completely oblivious to this activity.

The history of viruses

According to computer consultant and columnist Rob Rosenberger, the roots of the modern computer virus go back to 1949. This was when computer pioneer John von Neumann presented a paper on the “Theory and Organization of Complicated Automata,” in which he postulated that a computer program could reproduce.

Bell Labs employees gave life to von Neumann’s theory in the 1950s in a game they called “Core Wars.” In this game, two programmers would unleash software “organisms” and watch as they vied for control of the computer. Core Wars was described in three issues of *Scientific American* in 1983 and 1984. According to some pundits, the code snippets used to illustrate the concepts of this useful game may have provided the first roadmaps to taking viruses out of the arena and into the wild.

This is echoed in science fiction of the day. Gene Spafford of Purdue* credits David Gerrold with being the first to use the word “virus” as a computer attack in Gerrold’s

* Eugene H. Spafford, “The Internet Worm Program: An Analysis,” Purdue Technical Report CSD-TR-823, West Lafayette (IN), November 29, 1988.

science fiction stories about the G.O.D. machine.* Spafford describes the origins of the virus:

A subplot in that book described a program named VIRUS created by an unethical scientist. A computer infected with VIRUS would randomly dial the phone until it found another computer. It would then break into that system and infect it with a copy of VIRUS. This program would infiltrate the system software and slow the system down so much that it became unusable (except to infect other machines). The inventor had plans to sell a program named VACCINE that could cure VIRUS and prevent infection, but disaster occurred when noise on a phone line caused VIRUS to mutate so VACCINE ceased to be effective.

In 1984, Ken Thompson described the development of what can be considered the first practical computer virus (though he didn't give it that name). Thompson wrote a self-reproducing program in the C programming language. He modified the program to "learn" new syntax and then planted a Trojan horse (described later) that deliberately miscompiled the Unix login command, enabling him to log into the system as any user. Next, he added a second Trojan horse aimed at the C compiler. After compiling the modified source with the normal C compiler to produce a bugged binary, he installed this binary as the official C, and removed the bugs from the source of the compiler. Then, whenever the new source of the compiler was compiled, the new binary reinserted the bugs. The login command remained bugged, with no trace in the source.[†]

A number of people developed computer viruses on IBM PC and Apple II computers (though they were not called viruses) during the early 1980s. (According to Spafford, "Festering Hate" and "Cyberaids" were among the infections.)

Fred Cohen from the University of Southern California was the first to define formally the term "computer virus." (The term was suggested by Cohen's advisor, Len Adleman, after Cohen developed an experimental program for a security seminar.) According to Cohen, a computer virus is:

...a program that can "infect" other programs by modifying them to include a possibly evolved copy of itself. With the infection property, a virus can spread throughout a computer system or network using the authorizations of every user using it to infect their programs. Every program that gets infected may also act as a virus and thus the infection grows[‡].

In his experiments, Cohen showed how quickly a virus could propagate throughout an entire operating system.

* These stories were later combined and expanded into the book *When Harlie Was One*, Ballantine Books, First Edition, New York (NY), 1972). In later editions of the book, the virus plot was removed.

† Kenneth Thompson, "Reflections on Trusting Trust," *Communications of the ACM*, Volume 27, Number 8, August 1984.

‡ Fred Cohen, "Computer Viruses: Theory and Experiments, *Computers and Security*," Volume 6, Number 1, 1987 (first presented at 1984 meeting of IFIP Technical Committee 11 on Security and Protection in Information Systems).

Hundreds of thousands of viruses now inhabit computer systems—particular in uncontrolled PC environments. Precise calculation is impossible because a generally agreed upon taxonomy of virus types has never come about. Formerly, viruses were often named for the place where they are first discovered (e.g., the Bulgarian Factory of viruses), or for some message displayed by the virus (e.g., the AIDS virus). Examples of early PC viruses include the Brain virus, the Fu Manchu virus, the Icelandic virus, and the Bouncing Ball virus. More recently, as competing teams of virus hunters have tried to eliminate duplication of effort, the names make reference to some distinctive feature in the code of the virus. Examples include SoBig, Code Red, and NIMDA. Even so they don't always get it right, hence several popular viruses are known by more than one name.*

Worms

A *worm* is an independent program that reproduces by copying itself in full-blown fashion from one computer to another, usually over a network. Like a virus, a worm compounds the damage it does by spreading rapidly from one site to another. Unlike a virus, which attaches itself to a host program, a worm keeps its independence; it usually doesn't modify other programs. Like a virus, however, a worm can include malicious instructions that cause damage or annoyance, in addition to whatever inconvenience it causes by tying up the resources of the network as it maintains and reproduces itself.

The notion of a worm as a computer intruder apparently dates from John Brunner's 1975 science fiction novel, *The Shockwave Rider*. In Brunner's book, programs called "tapeworms" lived inside computers, spread from machine to machine, and were "indefinitely self-perpetuating so long as the net exists."†

Around 1980, John Shoch and Jon Hupp, researchers at Xerox Palo Alto Research Center, developed the first experimental worm programs as a research tool.‡ The worms were designed to spread from one computer to another. Shoch and Hupp described their worms as follows:

A *worm* is simply a computation which lives on one or more machines...The programs on individual computers are described as the *segments* of a worm...The segments in a worm remain in communication with each other; should one segment fail, the remaining pieces must find another free machine, initialize it, and add it to the worm. As segments (machines) join and then leave the computation, the worm itself seems to move through the network.

* For a full taxonomy, see Eugene H. Spafford, Kathleen A. Heaphy, and David Ferbrache, *Computer Viruses: Dealing with Electronic Vandalism and Programmed Threats*, ADAPSO, Arlington (VA), 1989. (Order from ADAPSO: (703) 522-5055.)

† John Brunner, *Shockwave Rider*, Ballantine, New York (NY), 1975.

‡ John F. Shoch and Jon A. Hupp, "The Worm Programs—Early Experience with a Distributed Computation," *Communications of the ACM*, Volume 25, Number 3, pp. 172-180, March 1982. (An earlier version was presented at the Workshop for Fundamental Issues in Distributed Computing, ACM/SIGOPS and ACM/SIGPLAN, December 1980.)

The Xerox PARC worms were, on the whole, useful creatures; they handled mail, ran distributed diagnostics, and performed other distributed functions. A few errant, experimental worms did get out of control, however, before a “worm watcher” was added to the network.

The Morris Worm

The worm that took the spotlight may not actually have been a worm at all. It was the creation of Robert Tappan Morris, a 23-year-old doctoral student from Cornell, who on the second of November 1988, at about 6:00 p.m., released a self-replication bit of code onto the Internet designed to spread itself freely, but to do little else. There was no dangerous payload. It erased no files. It did not attempt to establish accounts with which to exploit other computers. It logged no keystrokes and stole no passwords. It was intended to spread, nothing more, and once it did so, it was pretty much supposed to just lay low in the operating system and let the world go by.

Soon, however VAX and Sun machines (the only systems targeted) across the country started to bog down. The process load caused by hundreds of instances of the worm running at once used up all the systems' available power, and no other users could log on. Desperate administrators disconnected their equipment from the Internet, or powered down their expensive machines and brought them up again, only to discover that they were quickly reinfected and were often worse off than before.

Had the Morris worm been a true virus, it would have attached itself to some other piece of computer code and remained inert until its host program was called. It also would have had to depend on some action of the user to allow it to spread from system to system. As a worm however, the program was self replicating, and required no action on the part of a user to spread. It sought out new victims on its own, and was transmitted to them via the Internet.

Whether virus or worm, a possible flaw in the code caused the code to attach itself very quickly to new processes inside the targeted computer, and then for each of these to attempt to do the same. The result was rapidly expanding numbers of copies of the virus running in the victim computer at the same time, squeezing out legitimate traffic, and rendering the machine useless.

This same scene was replayed at the sites of over 6,000 machines across the country. While no physical damage was caused by the worm, the U.S. General Accounting Office estimated that the worm cost between \$100,000 and \$10,000,000 due to lost access. Robert Morris was tried and convicted of violating the 1986 Computer Fraud and Abuse Act. After appeals he was sentenced to three years probation, 400 hours of community service, and a fine of more than \$10,400. He went on to have a very successful career in computer science and education.

Trojan Horses

A *Trojan horse* is a code fragment that hides inside a program and performs a disguised function. In classical mythology, a Trojan horse was a large hollow horse made of wood by Odysseus during the Trojan War. The Greeks hid soldiers inside the horse and left it at the gates of Troy. After the Trojans were persuaded to bring the horse inside the gates, the hidden soldiers opened the doors for the rest of the army, which attacked the city and won the war.

In the modern computer world, a Trojan horse hides in an independent program that performs a useful or appealing function—or appears to perform that function. Along with the apparent function, however, the program performs some other unauthorized operation. A typical Trojan horse tricks a user into running a program, often an attractive or helpful one. When the unsuspecting user runs the program, it does indeed perform the expected function. But its real purpose is often to penetrate the defenses of the system by usurping the user's legitimate privileges and thus obtaining information that the penetrator isn't authorized to access. An example of this would be the modern *rootkit*, which is a script that controls a small suite of programs that create an administrative level account on the targeted system, and then create a *backdoor*, an unmonitored entrance way that evades the security mechanisms, through which the attacker can later gain convenient access. Trojan horses are often hidden in programs that entice users by displaying information about new system features or by hiding in downloadable applications or games.

Many web site operators today insert Trojan horses that create content that pops up on your screen while you are doing other functions. While the marketers may claim they are merely displaying additional advertisements that may be in sync with what you were currently browsing, what they are actually doing is monitoring your actions, and then sending out codes over your network that pull in content you did not request, and then put it in front of you. For this service they charge the content owner a fee.

Dan Edwards of NSA is credited as being the first to use the term “Trojan horse” in the context of malicious software. One classic Trojan horse attack was described by Dennis M. Ritchie.[†] An attacker wrote a “password grabber” program that simulated the normal login process and ran it on the victim’s computer. When an unsuspecting user saw the `login:` prompt and attempted to log in, the program captured her username and password and transmitted to whomever was doing the spying. The Trojan then prompted the victim to reenter the username or password, claiming

* Morrie Gasser, *Building a Secure Computer System*, New York (NY), Van Nostrand Reinhold, 1988. Also, Donn B. Parker, “The Trojan Horse Virus and Other Crimoids,” in Denning, Peter J., ed., *Computers Under Attack: Intruders, Worms, and Viruses*, ACM Press, Addison Wesley, Reading (MA), 1990.

† Dennis M. Ritchie, “On the Security of UNIX”, *UNIX System Manager’s Manual (SMM)*, 4.3 Berkeley Software Distribution, University of California, Berkeley (CA), 1986.

there had been an error, and then shut itself off. The user then ran through the normal login sequence in the ordinary way, but perhaps resolving to take more care in typing. One unpublicized defense against such an attack may have been to enter a curt two-word epithet as the first logon attempt, and then to log in normally on the second try.

Some Trojan horses do not actively interact with the user, but instead simply monitor keystrokes on the keyboard. This technique can seize passwords more readily with less risk of detection. Other versions of this ploy snatch the users' name, credit card numbers, and expiration dates for transmission to a storage facility. The numbers are later tested by placing a small order. If the numbers prove valid, they are sold to unscrupulous customers.

A clever Trojan horse will leave no trace of its presence, may reside indefinitely in unsuspecting software, and may even be programmed to self-destruct before it can be detected.

Bombs

Not all malware goes to work immediately. Attacking code can be stealthier if it just lays low for awhile, often using an assumed filename or a filename that is very close to a legitimate system file. Later, the attack can be launched when it is not expected. Sometimes this approach is called a *bomb*. A bomb is a type of Trojan horse, used to release a virus, a worm, or some other system attack. It's either an independent program or a piece of code that's been planted by a system developer or a programmer. A bomb works by triggering some kind of unauthorized action when a particular date, time, or condition occurs.

Technically, there are two types of bombs: time and logic. A bomb that's set to go off on a particular date or after some period of time has elapsed is called a *time bomb*. The Friday the 13th Virus, which started doing damage on the first Friday the 13th in 1988, is an example of a time bomb. A bomb that's set to go off when a particular event occurs is called a *logic bomb*. Software developers have been known to explode logic bombs at key moments after installation—for example, if the customer fails to pay a bill or tries to perform an illicit copy.

A.K. Dewdney described a logic bomb in the French spy novel by Thierry Breton and Denis Beneich, *Softwar: La Guerre Douce*:*

...they spin a chilling yarn about the purchase by the Soviet Union of an American supercomputer. Instead of blocking the sale, American authorities, displaying studied reluctance, agree to the transaction. The computer has been secretly programmed with a "software bomb." Ostensibly bought to help with weather forecasting over the vast

* A.K. Dewdney, "A Core War Bestiary of Viruses, Worms, and Other Threats to Computer Memories," *Scientific American*, Volume 252, Number 3, pp. 14-23, March 1985.

territory of the Soviet Union, the machine, or rather its software, contains a hidden trigger; as soon as the U.S. National Weather Service reports a certain temperature at St. Thomas in the Virgin Islands, the program proceeds to subvert and destroy every piece of software it can find in the Soviet network.

Trap Doors

A *trap door*, or a *back door*, is a mechanism that's built into a system by its designer. The function of a trap door is to give the designer a way to sneak back into the system, circumventing normal system protection. Unlike a logic bomb, which usually explodes in someone else's system, a trap door gives the original designer a secret route into the software.

Sometimes, programmers leave trap doors (entry points) in a program to allow them to test the program, or monitor its operation, without having to follow what may be cumbersome access rules or security measures. These trap doors also provide a way to get into the program in case there's a problem with the access routines. Although such trap doors are ordinarily removed before the program is shipped to the customer, sometimes they're left in the code by accident or by design.

A trap door is typically activated by the person who planted it. Usually, the means of access is not apparent—for example, an unlikely set of keystrokes or sequence of events, or a particular login. Once the developer gets in through the trap door, he may get special program privileges. In the 1983 movie, *War Games*, the hero gained access to a NORAD computer system by inadvertently entering a trap door planted by its creator (allowing him to log in without an authorized account), setting off a global thermonuclear war game that became all too real.

Another common example of a trap door, although harmless, is the Easter egg. This is hidden code that can only be actuated by an unlikely set of keystrokes, often while doing a specific task. Typically the egg is a set of credits for the unsung authors of the code who have toiled for months or years in anonymity, and feel they deserve at least a mention. Other times, complete games, MIDI tunes, and photo arcades have been included as a little hidden treasure. Various web sites list current Easter eggs as they are discovered and reported.

Spoofs and Masquerades

A *masquerade* is a generic name for a program that tricks an unsuspecting user into giving away privileges. Often, the ruse is perpetrated by a Trojan horse mechanism in which an authorized user is tricked into inadvertently running an unauthorized program. (The Trojan horse login described earlier is an example of such an attack.) The program then takes on the privileges of the user and may run amuck!

A *spoof*, on the other hand, is an important technique used for misdirection and concealment. Sometimes, a communication that the sender wishes to transmit anonymously is tagged with a false return address. Spoofing return addresses in this way is

one of the techniques used to create unsolicited junk email advertisements (spam) with no way to track down the offending sender.

Who Writes Viruses?

In the early days, most virus-like programs were written by geniuses, mainly computer scientists at major corporations and labs, who were still exploring the theoretical limits of computing. The question “What do computers do?” had not been answered. Attempts at producing independent, self-replicating program elements were a valid inquiry into the question of how computers should be used and organized. Nothing was easy about programming in these pioneering days. Assembly language, often considered today the most arcane of the programming arts, was still a bit of a dream. So, by and large, the early experimenters with viral technology, operating before the metaphor was even coined, were the giants.

In today’s world, you buy a computer; you plug it in, or have the kid who sold it to you come out and do so; you connect it to the Internet via a broadband connection; and you are flying. For many, it is an entertainment and communications tool, on which to write the occasional letter, using self-correcting, autoformatting, and grammar-checking office software. The computer has become as far removed from its early users as has the automobile from most of its drivers. True, most mechanics and automotive engineers drive to work, and some shade tree mechanics tinker in the innards, and a few actually make improvements. Most, however, decide on their choice of colors, and whether or not they want chrome rims or white sidewalls for their wheels. Customization has replaced engineering, and in many cases, disposal has replaced repair.

It is in this environment that the new breed of virus writer has emerged. The computer is no longer an awe-inspiring oracle in a glass enclosed tabernacle, into which only the chosen may enter. Today the computer, in the United States at least, is ubiquitous. Almost as many homes have computers as subscribe to newspapers and magazines. And as the entertainment value of the Internet increases, that number may begin to approach the market penetration of the telephone and television, which in the early 1990s was already present in slightly more homes than the number having indoor plumbing. There are so many computers in use today that states such as California have adopted laws concerning their disposal, so that old PCs don’t clog landfills.

In this computer-saturated society, the mystique of the box is replaced by the aura of that to which it connects. The Internet is the new hang-out, and to hang there in style requires you to make your mark. Joyriding and hacking, defacing web sites and tagging walls with spray paint, setting fires in dumpsters, and unleashing viruses, all are manifestations of a similar juvenile, typically angst permeating affluent nations. The difference is that the street tough guy skins his knuckles, and the script kiddie, which is the pejorative applied to wannabes on the Internet, grows pale and soon needs glasses.

The script kiddie does not usually write viruses: that would take programming skill. Instead, he (or she, though most crackers arrested are male) obtains canned code strings, called *scripts*, and runs them against targeted machines. In time, this attacker may develop some skill at evading network protection mechanisms, such as firewalls, and be attuned enough to stay ahead of virus protection systems. When all is said and done, however, a few really bright individuals discover flaws in common software, and write exploits against the flaws. Script kiddies hang on these deities for any scripts they may hand down to their adoring followers, who execute them.

The great mass of computer users have unprotected home machines plugged into broadband outlets without any strategy or tools for protection (firewalls and virus-scanning software are discussed later in this chapter). This low hanging fruit is easily sensed and attacked by the scripts, and, before long, the script kiddie has a sufficient herd of zombies that she can launch a DoS attack against someone she doesn't like or maybe shut down a municipal web site.

If he does not have access to a bright mentor, the script kiddie can download scripts from a cracker web site, likely one that is hard to read, with lots of dirty words, and sprinkled with 1337-script (pronounced "elite"—with certain letters replaced by similar looking letters, 1 for l, 3 for E, 7 for t). Fortunately, something happens to most of these: they grow up and some of them actually find work in the information technology industry.

Then there are the bright, criminally-minded ones who don't leave their hacking days behind and who become cyber thieves, spies, anarchists, and terrorists. Skillfully manipulated, lots of zombied computers could cause a lot of trouble. One Polish organization claims it can push spam with impunity regardless of changes in the law, simply by activating a few of its 500,000 zombies. A half-million machines throwing out millions of emails a day can be an inconvenience to a lot of people, but the real question may be "What else could they do?"

Any worthwhile technique learned by phone phreakers (those who attempt to outwit the public telephone network by synthesizing its control mechanisms so as to avoid charges) is likely in the hands of various militaries, as are the best work of the crackers. Electronic countermeasures going well beyond mere listening (ELINT) are likely practiced daily, using say, cell phones or phone emulators to access PBXs, which then are used to access—who knows?

Why bother considering electronic high jinks as a form of attack? Perhaps armies do not travel on their cell phones, but economies do. Economic pressure will eventually force a convergence of the myriad of proprietary protocols controlling the devices on factory floors, the SCADA systems controlling dams and waterways, the remote well heads in petroleum exploration, and the power grids supplying the homes and industrial facilities of a nation. Already, convergence is occurring between the Internet and the telephone network, as in Voice over Internet Protocol (VoIP), and some wireless providers have gone full circle, using wireless to transmit IP, on which voice is encoded.

Economic pressure will also force a centralization of control, with one center monitoring several plants rather than each installation being under local control. At that point, when one protocol dominates one network, the third class of virus maker will have their swing, either robbing from us, or extorting us, or simply turning key systems off as a cover for a bigger move. For all we know, World War III might begin with a loss of dial tone, and a string of “Error 404, site not found.” That will be the day to be truly fearful of viruses.

Remedies

There are many programs that can help you keep viruses and other wildlife away from your system—and can wipe out the critters if they gain access. Known as *virus protection programs*, these programs are available from both commercial and public domain sources. These products, and the system administration procedures that go along with them, have two overlapping goals: they don’t let you run a program that’s infected, and they keep infected programs from damaging your system.

Firewalls

A firewall protects your computer by examining each information packet that travels over the network. Clues to a packet’s purpose can be read from its destination address. Firewalls contain a list of allowed and disallowed destinations and functions. If a packet is heading for a forbidden address or comes from one, the firewall stops it. If a packet is heading for a valid address, but its port identifier (the clue to packet function) is unknown or disallowed, the firewall stops that packet as well. Advanced firewalls even keep track of outgoing packets, and open up only if a packet is expected and returning.

The role of a packet in stopping the prevention of active threats such as worms and viruses is that these pests often attempt to enter a computer using forbidden paths, such as port numbers that are unmonitored or unusual. The firewall examines each packet, and it quashes those that are unexpected or disallowed.

Antivirus

Virus protection software uses two main techniques. The first uses signatures, which are snapshots of the code patterns of the virus. The antivirus program lurks in the background watching files come and go until it detects a pattern that aligns with one of its stored signatures, and then it sounds the alarm and maybe isolates or quarantines the code. Alternatively, the virus protection program can go looking for trouble. It can periodically scan the various disks and memories of the computer, detecting and reporting suspicious code segments, and placing them in quarantine.

One problem with signature-based virus protection programs is that they require a constant flow of new signatures in response to evolving attacks. Their publishers stay

alert for new viruses, determine the signatures, and then make them available as updated virus definition tables to their users. To access the new tables, users typically download them from the World Wide Web.

Of course, as the number of viruses increases (and it shows no signs of abating), the tables get progressively larger, making frequent updates somewhat of a chore. This is particularly a problem in the case of memory-limited devices such as palm-top computers or intelligent cell phones.

Another problem is called the *Zero Day* problem. Basically, this occurs when a user trips over a new virus before the publisher discovers it and can issue an updated signature.

A third problem is that, just as with biological pathogens, viruses can mutate. Sometimes this happens accidentally; other times, it happens because a clever programmer uses file compression software to change the signature of the virus when it is not active or even gives it the ability to be self-garbling. This means it can change its own form by introducing extra statements or adding random numbers, to elude signature detection. (A similar technique is sometimes used by bulk emailers to elude subject line scanners.)

To counter these worries, virus protection publishers are adding what is called *heuristic detection* features to their wares. Basically, a heuristic is a rule or behavior. If a virus exhibits that behavior, the antivirus software tries to stop it in the act. For instance, a code snippet that suddenly accesses a critical operating system area or file, such as a file table definition sector on a hard drive, is likely up to no good, and should be stopped. Other risk indicators include unexplained changes in file size, particularly in system files, sudden decreases in available hard disk space, or changes in file time or date stamps.

The Virus Hype

Unfortunately, a backlash has recently started against virus alarmists. Like those who too many times cry “wolf” when there is no wolf and so must face danger alone when a wolf actually appears, those who trumpet the dangers of viruses often find the public to be increasingly apathetic when new attacks are discovered. The issue is not helped by an industry that is faintly tainted. Many journalists have noticed that often those who raise loudest the warning cry are owners of companies that publish virus protection software. It makes sense to scare the public into buying protection, but the practice may have questionable ethics.

Almost as troublesome are those who pass on emails that contain warnings of viruses to everybody in their address lists. Often these users are somewhat new to computers. Fearful of losing or damaging their own investments, they wish to spare their friends from calamity. The result is a big stream of well-intentioned junk email, often followed by a chagrined stream of “oops” emails, once they become aware of their apparent naiveté.

The result is that fear of viruses can often be as devastating as the viruses themselves. Not to say that viruses do not wreak havoc. Any administrator who has followed up on the results of a really wicked one, such as the Majester virus that has infected law firms, can tell stories about computers that were crippled beyond recovery. This means that failure to install at least some form of antivirus protection is simply negligent. However, it must be part of a balanced program that includes user education, regular backups, normal security precautions, and intrusion detection software or hardware.

An Ounce of Prevention

A final issue, and perhaps the most important one, is to determine why viruses spread in the first place. These days, software manufacturers are alert to the threat of security holes. Hordes of hackers regularly test network defenses or code vulnerabilities. When a hole is found, in most cases a warning is raised before an exploit is published. Manufacturers scramble to produce small updates, or *patches*, that improve the affected code and eliminate the vulnerability. It is a sad statement that today many attacks come days or months after a manufacturer learns of a problem and posts a cure.

In short, most people who have virus problems have them because they have not availed themselves of available protections. In most cases, all that would have been needed to prevent a disruption is to have obtained virus protection software. This can be done either by purchasing a commercial product, by downloading a free product from the Web, or by periodically logging on to a web-based scanning service. Also, make sure to keep your virus definition tables updated, and install patches and security updates on operating systems and applications as required.

Do this, and the virus problems that make headlines will for the most part pass you by. To some, the cure must seem worse than the disease. The phenomenon roughly parallels that of smoking, which even some tobacco manufacturers now acknowledge has adverse effects. To some, the risks seem acceptable—until the consequence arrives.

Summary

The worms, viruses, and other wildlife that exist in computers and networks are the descendants of early experiments with autonomous or symbiotic programs. Once unleashed into the wild, these rogue programs can multiply freely. They enter networks from the outside—that is, from connected networks or the Internet—and they enter networks from the inside—that is, from computers and media that users bring from home. Most viruses are nearly harmless. They deprive the network of little more than computer processing time and communications bandwidth, of which

there is usually plenty. These viruses exist as a chance for programmers to demonstrate their skills. Others can be devastating, crippling computers, robbing network owners of massive amounts of bandwidth, stealing secrets, defeating security, corrupting data or holding it hostage, even taking down entire systems. These extreme examples demonstrate how creating and propagating viruses and worms can be criminal, even terroristic, in scope.

Effective computer security policies and practices can do much to eliminate the spread of viruses and worms. In the end, however, nothing can do more to stop the spread of pathogenic programs than educating and training users in virus prevention.

CHAPTER 5

Establishing and Maintaining a Security Policy

Secure system planning and administration is the human side of computer security. Even in a highly trusted system, security isn't automatic. Administrators need a written guideline, spelled out beforehand, that clearly outlines what steps to take and what procedures to follow in the pursuit of security. The assault on trusted systems seems relentless these days, as vulnerability after vulnerability has riddled both the Windows and Linux worlds, and perfidy abounds abounds both inside and outside an organization's walls. If there is safety in the changing world of security, it seems to lie not in what our equipment or software does for us, but in what we do for ourselves. The first step in maintaining security today is to set security policies for our organizations, and then to exercise diligence in promulgating and maintaining them. This effort cuts across all layers. Although the security administrators carry out the security policy in terms of protection, detection, and enforcement, it is the users who must keep the security, and the owners and managers who must authorize and sustain it, and administer the required sanctions against those who violate it.

For example, your organization's security policy may require regular backups, but it's the administrator who must actually run the backups. Once administrators train users to copy files to areas that will be protected, managers must deal with noncompliance. Similarly, administrators may tell users to avoid writing down machine-issued passwords and laying them near their keyboards, but management must give the policy teeth.

The most critical area for all layers to come together is likely incident response—what to do once a breach occurs. Decisions about evidence preservation, notifying authorities (and which ones), and what to do next must be hashed out before the fact. These should be codified in writing and distributed to all persons likely to be affected.

The security policy is a living document that must be examined and updated regularly. Training users, administrating passwords, backing up system-critical files, setting up and tuning firewalls and intrusion detection systems, and examining audit logs: these are some of the many ways that a system's abstract security policy gets translated into real world defenses. This is the role of the security administrator.

Administrative Security

Administrative security falls into three general categories:

Overall security planning and administration

This category includes working with management to set a security policy for your organization, publicizing it and gaining management support for it, performing risk analysis and disaster planning, monitoring employees, training users, answering their questions, and so on.

Day-to-day security administration

This category includes creating accounts and assigning security profiles for users—for example, their initial passwords, their password controls (e.g., how often they must change their passwords), their login controls (e.g., what hours they can log in), making sure there aren’t security holes in your system, and so on.

Day-to-day system administration

This category includes keeping the system running, doing daily backups, trolling for breaches, and testing the condition of hardware and software used to sustain operations in times of stress or attack. This type of system administration is vital to any system. Although these mundane tasks may not seem especially security-relevant, they’re actually vital. Remember that “availability” is a key goal of overall computer security. Day-to-day system administration keeps the system available.

This chapter provides some guidelines for sound administration. Most enterprises today are a heterogeneous environment, that is some legacy functions reside on creaky old mainframes, some business critical processes are accomplished on a fairly modern computing core, and lots of client server networks unite a mixture of Windows and Unix users. Wireless implementations accommodate drop-in users and guests, as well as worker who insist on carrying their work to the lunchroom with them. There may even be a layer of mobile devices—PDAs with radios built in—by which a few workers keep in touch with each other during meetings. Each of these systems requires its own series of administrative practices, and each requires administrators to carefully develop security policies regarding its use.

Actually, in some respects, the highest levels of security are the easiest to attain. Most ultra-sensitive systems use an *air wall*, that is nothing goes in or out. Each terminal or workstation connects to its own network, and that network goes nowhere else. None of the users’ devices are equipped with floppy drives or removable media. And rogue wireless devices (or in the old days, rogue modems) are usually considered contraband in this environment.

This environment is practical only in a few disciplined organizations. Most organizations connect users to some kind of server or server cluster via a local area network, and the LAN usually connects at some point to the Internet, usually via a firewall. As this arrangement is most common, the bulk of security policies apply to it, although

password administration and certain other tenets, such as division of duties and least privilege certainly apply no matter what server and network configuration is in effect.

Fortunately, the more esoteric the network, the more administrator documentation the vendors supply to describe the security features of their systems. If your organization has government contracts, you may need to observe more stringent security policies established by the government for high-security sites. When your organization gets a security clearance, you'll find out the details of what you need to do.

It is in the vast bog of PC LANs and wireless networks that most security is made or broken. Because of size, staffing, or budget, some organizations may not have dedicated system administrators charged with security administration. In this case, the burden of security administration is likely to fall on the existing system administrators. If your organization can't afford full-time system administration, or if you don't have the appropriate staff to administer a security policy that adequately protects your equipment and information, you should consider hiring a security consultant on a short-term or periodic basis. Such a person can analyze your security risks and needs, help you set up a workable security policy, and conduct periodic security audits. (See the discussion in the later section "Performing a Security Audit.")

Overall Planning and Administration

System administrators once may have found themselves in the uncomfortable position of being the security advocates within their organizations—having to sell security both to users (who may question why they need to use features they may find cumbersome) and to upper management (who may question why it all costs so much money).

It is a different environment today. In the wake of 9/11, several industries found themselves regarded as part of their nations' critical infrastructure. While it is nice to be wanted, this carries with it the responsibility of confidentiality and availability that meets someone else's imposed standards, rather than what internal leadership considered previously to be good enough to get by. In addition, many recent corporate accounting scandals appear to have defrauded millions of people and have led to an increased accountability regarding the creation and maintenance of financial data. Finally, and somewhat belatedly, privacy has been raised as an issue, likely because of fear of identity theft, and in some cases of fear of discrimination based on personal or health data. This too has lead to a raft of new statutes. These regulations go by titles such as PDD-63, HIPAA, GLBA, the Patriot Act, the Sarbanes-Oxley Act, and others. Many of them carry fines, some of them per incident, for security infractions.

In addition, it has been determined that the storage or transmission of certain materials, such as pornography, obscene materials, or materials designed to incite racial hatred or discrimination based on gender, may produce a threatening environment for employees subjected to them. This may be considered a form of job discrimination, and federal and state law enforcement agencies can investigate.

Even if no discrimination is proved by the authorities, the party who feels aggrieved may choose to pursue legal action on his own after the official investigation.

Finally, there is the issue of copyrighted material. If software or MP3 music files are downloaded or distributed using company property, it may subject the company to legal action.

With the increasing liability surrounding their computers and networks, management is now much more interested in security. Unfortunately, moving towards a more secure network is often handled on a piecemeal, sporadic basis. A much more regulated approach is usually indicated.

Analyzing Costs and Risks

Computer security is a tradeoff. When you're considering building, buying, or even using a security product, you have to balance the cost of the product against the risk of doing without it. Most organizations formalize this process and call it a risk analysis. *Risk analysis* is a procedure used to estimate potential losses that may result from system vulnerabilities and to quantify the damage that may result if certain threats occur. The ultimate goal of risk analysis is to select cost-effective safeguards that reduce risks to an acceptable level. Basically, risk analysis is a way to figure out how important your system is, and how far you're willing to go—in terms of equipment, people, and budget—to protect it.

Standard risk analysis involves looking at your tangible assets—for example, your buildings, computers, and other equipment—and determining how to protect them. Because your organization's most valuable asset may be the information processed by your computers, not the computers themselves, you need to take a good look at how best to protect that information as well.

When you're evaluating your organization's information asset and considering whether and how to protect it, you'll have a number of important questions to ask.

What information do you have, and how important is it?

There are many different types of information: national defense information describing military resources and deployment; corporate records showing projected profits, losses, and strategies; personnel records describing health, financial, academic, and employment history. You'll need to assess how important that information is to your own organization. Information of inestimable value to one organization may have little or no value to another organization.

However, holding information is a two-edged sword. Legal safeguards today make it an expensive offense to release information in your possession. The Health Insurance and Portability and Accountability Act protects patient records, the Gramm-Leach-Bliley Act of 1999, protects customer records in the financial industry, and its January 2003 extensions apply to all data in the financial industry. Other specific

legislation may apply to other areas, such as the Family Educational Rights and Privacy Act, which applies to educational records.

How vulnerable is the information?

Some information may be very important to you, but may be of little interest to anyone else. The novel you’re writing on your PC may fall into this category. In this case, simple backups may suffice (and hard disk encryption if you write controversial material). Other information may be of great interest, but may be so inaccessible that additional security controls aren’t really justified. (Classified military information that’s stored on a heavily guarded computer with one authorized user and no network connections may fall into this category.)

Everyone needs to worry about physical threats (e.g., fire and power loss) and accidents caused by careless or untrained employees. Beyond these obvious perils, you’ll need to evaluate whether realistic attempts are being made, or could be made, to break into your system, and to assess how likely it is that a break-in will occur in the future. If you’re responsible for national defense information, you’ll have to worry about foreign intelligence. If you’re protecting your business’s data, you’ll be concerned about your competitors, crackers, and insider threats. Remember, too, that threats to information tend to grow as people learn about your system’s vulnerabilities, and as methods of exploiting those vulnerabilities get cheaper and easier.

What is the cost of losing or compromising the information?

There are many different costs and consequences for failure to secure computers and networks effectively. If we’re talking about the loss of vital national defense information, the cost of information loss or leakage might be cataclysmic. If a medical experiment is disrupted, or if patient records are lost or compromised, people might die. If the security of an ATM is breached, a bank might lose a lot of money—and, when the news hits the press, the bank might also suffer a loss of confidence by customers and possibly lawsuits by shareholders.

What about corporate strategy information? Personal health or financial information? Loss or compromise of each has its own risks, costs, and consequences, both tangible and intangible, ranging from the loss of competitive advantage to the risk of losing government benefits to personal embarrassment. Again, legislative action has increased the stakes to system operators, often making data safety the responsibility of the computer system owner. Any lapses in security may be reportable. The California Data Security Act (California SB 1386), for instance, requires disclosure of computer-security breaches in which confidential information of *any* California resident *may* have been compromised.

What is the cost of protecting the information?

There are certain basic costs that you must incur. You *must* back up your data. No matter what security violation occurs—a natural disaster, a user mistake, or a break-in—having a recent backup of your data will allow you to go on.

There are many different types of additional costs. Will you need to buy new equipment? Will the use of a security product slow down response time and performance in a system that must provide quick customer service? Will security controls detract from the user-friendliness of a system that you’re marketing as easy to use?

Here, too, you’ll have to consider the different types of costs within your own organization and to assess the impact of security costs in relation to expected security benefits. One rule of thumb is that the cost of securing information shouldn’t exceed the financial and administrative cost of recovering that information—although certain types of information, such as national defense information, can’t necessarily be quantified in this way. It’s also hard to quantify the damage done by publicity and the loss of public confidence. Dennis Steinauer of the National Institute of Standards and Technology put it this way, “Controls that are more expensive than the value of the information they protect are not cost-effective. Absolute security is achieved only at unlimited cost.”

Based on the answers to these questions, you’ll need to make a determination, balancing your assessment of the value of your information asset against the risks of losing it and the financial and human costs of protecting that information. Then you’ll need to decide what your priorities are, and what types of security—physical, operating system, communications lines, encryption, biometric devices, and so on—best fit your information, your risks, and your budget. Finally, you’ll need to make an educated guess about what to protect and how.

Who are you going to call?

One issue that requires considerable attention in advance is to whom to report computer incursions. A computer attacker may be wading through your dumpster right now, trying to find old employee directories, in which case the local police or sheriff would be a good start. However, electronic breaking and entry is not always simple to prosecute. The attacker may be in your city or across the globe. Local agencies may not have the expertise or jurisdiction to help you much. A good rule of thumb is to start with higher authorities first. Federal investigators can always refer you to state agencies in your area.

A “don’t cry wolf” policy may be a good thing to apply. If a break-in you experienced has definitely resulted in the release of confidential records, you might want to take steps to preserve any evidence and call the authorities. If you suspect some script kiddie is probing your resources after school is out, you may want to tighten your firewall and hope she just goes away. If your network is violated, some of the new disclosure laws may limit your ability to keep it secret. Review with counsel

your rights and obligations before any incursions occur. Then put the steps for your plan into the security policy, and distribute it to all personnel likely to be affected.

Of course, there is an obvious reason to state the course of action to be taken in advance in writing. It helps preclude an executive from committing a crime and then ordering the staff to avoid documenting or investigating it. For further information, see Chapter 25 of *Practical Unix and Internet Security* (Garfinkel et al., O'Reilly).

Planning for Disaster

One of the most important things you can do to protect your organization from disaster is to plan for that disaster. A disaster recovery plan is a plan for keeping your computer equipment and information available in case of an emergency. Disaster planning may spell the difference between a problem and a (possibly business-threatening) catastrophe.

Your organization's disaster recovery plan will involve such activities as backing up data for storage at remote secure facilities and arranging for the use of other computer facilities or equipment in case of an emergency. Such arrangements may be informal (for example, you might make a reciprocal agreement with another department or organization to use each others' equipment if a disaster occurs), or they may be formal (for example, you might prepare a separate emergency site or contract with an organization that handles disaster preparedness).

Emergency sites are usually characterized as cold, warm, or hot. *Cold sites* are emergency facilities containing air conditioning and cabling, but no computers. You can hustle up some servers and desktops, move other replacement equipment into this site, and continue processing. *Hot sites* are emergency facilities containing computers, backup data—the works! *Warm sites*, a hybrid, are sites in which computers and equipment are preinstalled, but not programs or backup data. With increasing awareness of the frailty of the interconnected power grid, more companies are incorporating an additional requirement into their back up sites—geographic distance. It is no good having the backup site and the main site both fail due to the same calamity.

In addition to protecting your organization's equipment and information, a disaster recovery plan may greatly increase public confidence—as well as the confidence of your employees and managers—in your ability to safeguard data and continue to provide service.

Remember that backups are the key to disaster planning. If a disaster occurs and you've backed up your system, you'll be able to recover eventually. See the discussion in the section "Performing Backups" later in this chapter.

Chapter 9 discusses some of the natural disasters that face your organization and describes what you can do to reduce your risks.

Setting Security Rules for Employees

Some aspects of security are simply good management. Be sensible about who you hire, what computer resources you let them use, and what you do when they leave your organization. See the sidebar “Hints for Employee Security Management” for the most basic rules.

Training Users

No matter how diligent and careful a system administrator you are, you can't underestimate the ability of your users to undermine your efforts. In polite language, this is called “the human factor,” and it has grown exponentially since the PC became commonplace. The users in your organization have to take some responsibility for security. Teach your users how to use the hardware and software, be sure they understand your organization's security policy, and impress upon them the importance of observing good security practices. (See the sidebar “Hints for Safe Computing” for some very basic guidelines for individual user security.) Most important, be sure they know how to recognize security problems and what to do if they occur. Remember, improperly trained users are more of a peril to system security than attackers.

Day-to-Day Administration

Day-to-day system administration encompasses many activities, but most focus on keeping your computers and networks running smoothly by maintaining equipment, making sure there's sufficient space on the system disks, and protecting the system and its software from damage. Examples include making sure users can't modify system software; checking each new release of a vendor's software, especially fixes to security problems, to be sure such problems have really been fixed; and insisting that users or system administrators promptly patch any security holes or other bugs that are discovered.

It is essential to monitor various groups and news wires, as well as official sites of your vendors, so that you are aware of potential problems. Unfortunately, there are still instances where a patch to one problem breaks something else, especially in cross-vendor situations. The most affluent of organizations maintain test networks in which checks are made to make sure the cure is not worse than the disease prior to pushing out software updates. If you would prefer to get a holiday bonus rather than get more problems to worry about, stay tuned to security web sites for news of troubles with bug fixes and patches.

Hints for Employee Security Management

- Make sure the system administration staff is well-trained and that your organization has a policy for handling turnover and training new people. Inexperienced system administrators are a major threat to security.
- Monitor your employees' security practices. If your employees are careless about logging off when they leave their desk, if they fail to do backups, or if they're not diligent about protecting their passwords, remind them of the importance of these activities!
- Put your organization's security policy in writing. Be sure that all levels of management agree with it and that employees understand it and agree to abide by it.
- Check out your employees before they're hired or given security-related work. Different organizations believe in different levels of checking. Some might simply check references. Others might do personal background checks, administer lie detector and drug tests, and insist on written contracts and agreements of various kinds. Online subscription services can perform cursory versions of these checks for a nominal fee. Make sure that you understand local right-to-work and equal opportunity laws before you exclude someone from employment based on a background check. One class of employee deserves very close attention—those who have power to disable security mechanisms and perform maintenance around them. Enforce vacation policies and try to rotate certain types of assignments. Many security attacks take a long time to complete (e.g., slicing tiny amounts daily off bank balances) or involve daily monitoring. Shaking up staff assignments periodically may uncover such long-term attacks.
- Limit the access that users have to equipment and information. If someone doesn't need access to a particular network or server, or a certain set of files, don't grant access automatically. Also, lock server rooms and telecommunications closets. If someone from outside your organization must enter a locked space to do maintenance, see that they are accompanied by a trusted employee.

When an employee leaves your organization, be sure to review with the employee his continuing obligation to keep company information confidential:

- Revoke all of that employee's authorizations immediately; get back keys, smart cards, tokens, badges, and the like. Consider changing the locks to the facility and/or computer room.
- Delete or deactivate the employee's account and password from all systems and networks. Some operating systems specify that user profiles can be renamed to replacement employees rather than creating a new security profile from scratch.
- Save the employee's files in case they're needed for proof if you discover wrongdoing.

Hints for Safe Computing

Security features and trusted systems do a lot to make your computer environment a secure one. But in most systems, the final word on security is your own. Here's a collection of general hints for protecting your computer and your data. Other chapters contain additional hints in specific areas.

- Obey your site's security policy. Follow the rules, make sure your work habits are secure, and don't try to bypass security. Taking a few extra minutes to protect your login, your password, and your data is a pretty good bargain compared with trying to reconstruct your work and deal with the consequences if PCs or files are stolen, lost, or damaged.
- Never leave your computer, workstation, or terminal unattended. If you're going out to lunch, log out first. The easiest way for someone to crack a system is simply to use your account.
- Sanitize the hard drives on old computers before you discard them. Be sure to delete all data by overwriting what's there. Don't just reinitialize your tapes or disks. That typically rewrites only the header. Be aware that skilled computer forensics professionals can recover data that has been overwritten several times.
- Don't eat or drink near your computer or any computer media. A soda spilled into the ventilation holes of your PC can cause a tremendous amount of damage. And even after all these years, the CD drive is still not a cupholder.
- Be careful not to damage your disks and other media. For example, don't write directly on their labels with a ballpoint pen. Remember that the surfaces of recordable CDs and DVDs must remain scratch free. The label may actually be a protective covering for a mirror surface behind.
- Use any security controls and products available to you. These may include locks, security boards, and software packages and features.
- All data theft does not have to be electronic. Be careful about leaving sensitive documents within easy access.
- Most manufacturers now recommend that you use physical security, such as a firewall and possibly an intrusion detection system for your network, as well as software security, such as an antivirus software and a personal firewall.

Performing Backups

Backups of your system and all the data stored on your system are absolutely essential if you expect to be able to recover from a disaster. What kind of disaster? It might be a natural disaster, such as a fire or a flood. It might be a crime, such as a system intruder's meddling, vandalism of your computer room, or theft of a computer or a disk. It might be a hardware or software failure or a user error (e.g., deleting the latest version of a document or the latest release of some development

software). Whatever the cause, and whatever the extent of the damage, you *will* be able to recover eventually if you have recent backups of all your system data.

In a PC environment, many system administrators discover that critical documents on a user's machine often disappear when a disk fails. They can help protect against this by providing personal folders in common space on a server. Users are responsible for the contents of their own hard disks. Failure to have these files in a public storage area is not an excuse at your performance review, when a PC failure necessitates rework.

There are many systems for backup. You should do it regularly. Many organizations have well-defined rules about performing backups; if you don't follow the rules, you'll lose your job. But many other organizations have much looser policies. The scheduling and the extent of backups is far more discretionary. In these cases, it's really up to you. You'll find some general guidelines in the "Hints for Backups" sidebar.

Hints for Backups

Remember that your backups are the key to recovering your system in case of a disaster. Back up all of your files, and follow these rules:

- Encrypt your backups if they contain sensitive data.
- Keep extra backups off-site in a locked, fireproof location. You don't want a fire, lightning, or some other disaster to wipe out your system and your backups at the same time. Usually third-party couriers can provide this service.
- Secure your backup tapes or disks in locked areas with a sign-out sheet. Don't leave them on a desk for someone to steal or mistakenly use.
- Verify your backups. Check periodically to make sure they've been produced correctly and haven't been damaged in any way. Many consultants tell horror stories about systems that owners thought were backed up that were not, either because backup tapes were faulty or were not being properly rotated.
- Sanitize your backups before discarding them. Be sure to delete all data by overwriting what's there; don't just reinitialize your tapes or disks. (That typically rewrites only the header.) The process of overwriting and deleting may need to be repeated several times to make the backup completely secure. In some cases, bulk degaussing may be required.
- If you're throwing backups away, destroy the media first (by burning, crushing, or shredding.) Commercial service providers can degauss, shred, and disintegrate media, rendering it particle-sized. Locate these services in the Yellow Pages or on the Internet. Of course, make sure to provide for secure transport to the destruction facility.
- Consider buying an automatic backup program that runs full or incremental backups (without your intervention) every night. There are also services available to encrypt and back up your data over a phone line to an "electronic vault."

What does it mean to perform regular backups? That's an organizational decision: it depends on the number of users in your system, the volume of work, and many other variables. Many organizations perform a full backup (of every file in the system) every night. Others may do a full backup only once a month, or more commonly, once a week, but they do an incremental backup (of everything that's changed since the last full backup) every day. The best rule of thumb is to back up frequently enough that you can afford to recreate the work that may be lost since the last backup.

Like most security practices, however, backups have a cost associated with them. In this case, it is usually network bandwidth and server capability. You'll need to schedule backups in less desirable parts of the day, so that they will inconvenience the fewest users. If your organization operates 24/7, it may be necessary to host redundant systems, so that one can be backed up while the other is live. Fortunately, improvements in fault tolerance, using technical means to limit any single points of failure, and clustering technology, which entails running several computers in parallel to spread the load and provide redundancy, make this economically feasible. It is not necessary for the redundant system to just sit there when it is not being used, it can share the load of normal processing as well.

Hardware and Software Security Tools

Fortunately, today there's a good variety of hardware and software tools designed to prevent network incursion. As I mentioned previously, one of the most important is the firewall. A firewall monitors communications that pass through it, and it can take action against users that seem to be abusing or attacking the network. In some cases, the firewall monitors the Internet Protocol (IP) address of a packet, and if it is not found on a safe list, or is discovered to be on a "deny entry" list, it deletes the packet from the transmission stream, and usually any that follow from the same unauthorized addresses.

A firewall can also monitor the ports used by a communications session. Each protocol has a unique combination of ports available to it over which to communicate information. Using ports allows several different conversations to take place using the same IP address. However, the presence of communications from unexpected ports may indicate that an attack is underway. A firewall can also silence packets to and from undesired ports.

An *intrusion detection system* (IDS), on the other hand, usually listens to the circuit, taking note if any unusual activity is taking place. For instance, a certain user that constantly connects to a little used disk drive may be storing information there, either for later theft, or perhaps to be used as a tool in a future incursion. Intrusion detection systems usually have large libraries of attack signatures, that is, lists of the steps attackers typically take or have taken in the past to accomplish some attack. If the pattern of these attacks is repeated in a system being monitored by the IDS, the IDS will likely stop the transaction if it can, and place a page or call to an administrator informing of the attempted attack.

A *honeypot*, sometimes called a *honeynet*, is a decoy. It is usually placed in an unprotected portion of the network as a lure to attackers. While unauthorized users are checking out the honeypot, their movements are recorded. This helps further develop the library of attack signatures.

Penetration testing, or *pentesting* is a programmed, usually automated series of attacks that administrators carry out on their own network. The purpose of pentesting is to locate overlooked vulnerabilities. These are then patched, and communications proceeds. Pentesting may be performed by network personnel or by outsiders contracted for the purpose.

Performing a Security Audit

It's a good idea to check on the security of your system by performing periodic security audits. A *security audit* is a search through your system for security problems and vulnerabilities.

Check your system files and any system logs or audit reports your system produces for dangerous situations or clues to suspicious activity. These might include:

Accounts without passwords

They might have come with the system, or they might have been set up for guests or demos. Anyone can log in using such accounts.

Accounts with easily guessed passwords

These might include passwords selected by users or passwords associated with administrator or guest accounts. In addition, most attackers are well aware of the passwords and usernames that come with equipment from the factory. Change these immediately.

Group accounts

Long lists of privileges for individual accounts sometimes create confusion. Group management of accounts can simplify security administration by allowing precise, predetermined groups of privileges to be assigned to groups such as accountants, HR, engineering, and so on, in accordance with the organization's security policy.

Dormant accounts

These include accounts of users who have left your organization, have gone on vacation, or have moved to a different group or system.

New accounts

Be sure these are accounts you have assigned and not accounts that an intruder has created.

Default accounts

Many operating systems create "Everybody" or "Guest" or even "Administrator" accounts automatically. In some cases, these accounts are disabled, but an attacker may be able to make them live, or use them as a foothold to deeper

penetrations. For this reason, some administrators delete them or provide more subtly labeled replacements.

Recent changes in file protection

An intruder may have given special privileges to certain programs or may have made system files accessible to ordinary users. Individual users may have carelessly made their files accessible to everyone in the system. Monitor logs for privilege escalation to make sure attackers aren't gradually trying to obtain administrative ability.

Suspicious user activity

Basically, this means that a user (or someone using that user's account) is acting in an unexpected way—for example, someone logs in from a number of different terminals, logs in at odd times of the day or the week, runs protected system programs, transmits or dials out an unusual amount, uses new networks, etc.

Separation of Duties

Separation of duties is the principle that it's better to assign pieces of security-related tasks to several specific individuals. If no one user has total control of the system's security mechanisms, no one user can completely compromise the system. This principle is related to another important security principle, that of *least privilege*, the idea that the users and the processes in a system should have the least number of privileges—and for the shortest amount of time—needed to do their work.

In many systems, the system administrator has total control of the system's daily operations and security functions. In secure systems, this concentration of power in a single individual isn't allowed. It's obvious that in such systems, ordinary users shouldn't be allowed to perform security-related functions (except those that are discretionary to them, such as protecting files they, themselves, own). It may not be so obvious that security-related functions should not automatically be in the bailiwick of the system administrator, who takes responsibility for other important system operations.

In highly secure systems, as many as three distinct, complementary administrative functions, or roles, may be required: a system administrator, a security administrator (sometimes called an Information System Security Officer or ISSO), and an operator.

Typical system administrator/operator functions include:

- Installing system software
- Starting up and shutting down the servers in the system
- Adding and removing system users
- Performing backup and recovery
- Handling and servicing printers

Typical security administrator functions include:

- Setting user clearances, initial passwords, and other security characteristics for new users, and changing security profiles for existing users
- Setting or changing file sensitivity labels
- Setting security characteristics of devices and communications channels
- Reviewing audit data

If an operator role is defined, the operator may perform some of the more mundane system administrator duties, such as doing backups.

The system administrator, the security administrator, and the operator may not always be different people, but in a secure system their roles must be clearly divided. Whenever the system administrator assumes the role of security administrator, for example, the person must switch hats thoroughly enough so the system is aware that the person is changing roles.

Suppose the person serving as system administrator needs to perform a security function—for example, starting up an auditing program. She will typically have to exit from the system administrator interface and switch, in some system-defined way, to the security administrator interface before being able to run the program. Although cumbersome, this process clearly reinforces the system administrator/security administrator's understanding that the two roles are very different, with clearly delineated responsibilities that are monitored by the system. The system administrator and the security administrator play complementary roles that provide checks and balances on each other.

In some ways, these roles meet the objective of the so-called *two-man control* discussed in government security guidelines—the idea that it's much less likely that two people will conspire to breach security. For example, the system administrator's job is to add new users to the system; the security administrator's is to assign a password, a clearance, and other security information to that user's account. The security administrator usually must create system administrator accounts; the system administrator cannot create his own.

Summary

Security grows down into an organization once a written policy dictates it is required. Administratively, this means that management creates and sustains the demand for things to be done according to certain standards and levels. This requires that risks be categorized and prioritized, and the value of the asset to be protected is weighed against the cost of its protection.

Security policies require procedures. Security procedures include holding regular security audits, and implementing rules such as separation of duties and use of two-man controls. To insure people know how execute security procedures requires security training. To make sure people actually follow policies and procedures requires oversight and enforcement. For there to be enforcement, management must be involved. Management, after all, sets the policies.

CHAPTER 6

Web Attacks and Internet Vulnerabilities

If you are at all interested in computer and network security, you'll need to know something about the Internet, and its subset, the World Wide Web (WWW). This chapter deals with the basics of the Internet and the Web, as well as several important Internet protocols that keep the Internet humming from behind the scenes. The chapter then discusses vulnerabilities of several of these services, as well as exploits that can be used to attack them. Finally, this chapter gives several suggestions of how users can defend against those who misuse the Internet to steal or annoy.

About the Internet

The Internet may have been invented by former Vice President Al Gore, but it has since taken on a life of its own.* When many people think of the Internet, the first thing that comes to mind is often the World Wide Web; but the Web is only part of the story, if a highly visible one. This works to the advantage of the attacker, who crafts exploits based on less familiar parts of the Web in order to shut down the parts more readily seen.

In truth, the Internet is composed of many different connection schemes called *protocols*, all of which transmit over a common system of packetized communication called Transmission Control Protocol/Internet Protocol (TCP/IP). Among these are the following:

* A flurry of press clippings to the contrary, Gore appears never to have made the claim that he invented the Internet, except perhaps to point out that he promoted technical progress during his terms in the House of Representatives and as the Vice President. It is all a ruse, and a well-executed one. There was even circulated a bogus RFC, RFC 3000, in which Gore purportedly detailed his involvement. This was published before the RFC counter got that high. The real RFC 3000 is way less entertaining.

File Transfer Protocol (FTP)

The File Transfer Protocol allows rapid, reliable transfer of data files between repositories, called FTP servers, and between computers with FTP client software installed, called FTP clients.

Hypertext Transport Protocol (HTTP)

The Hypertext Transport Protocol allows users to access pages of text that are marked up using a special format called the Hypertext Markup Language (HTML). HTML tags are inserted into a web document to indicate the desired font, color, and position of text, and it facilitates linking to different different web sites, files, or pages. This allows an author to create the displayed page one time and to have it display more or less the same on any platform.

Simple Mail Transfer Protocol (SMTP)

The SMTP service allows a standardized method of electronic mail transmission.

Domain Name Service (DNS)

The Domain Name Service resolves the easy-to-read names familiar to Internet users, such as *www.oreilly.com*, to the Internet Protocol addresses that actually guide information around the network, such as 172.16.32.15.

Dynamic Host Configuration Protocol (DHCP)

When requested, DHCP automatically provides an Internet Protocol (IP) address, such as 172.16.32. 15, to a computer on a local area network. An IP address is required to communicate with other network devices that exist beyond the immediate proximity of the computer requesting the address.

Each of these is very useful to the Internet as we know it. But each is subject to attacks that can cause no end of problems. This chapter will describe several of these protocols, what they do, how they can be subverted, and provide some insight into what can be done about it.

History of Data and Voice Communications

When you call a friend on the phone, you imagine that the telephone company forms a direct connection between you and your friend. You don't care if the connection is always there, or if the phone company just whips it up to order, provided it is there when you need it. This method of connection is called *circuit-switched*. Circuit switching solved a ton of problems in the early days of the phone network because it saved the telcos from having to run a wire from each subscriber to every other subscriber. Instead, a single wire could go from a subscriber to a central office (CO), from which the call could be switched down a *trunk* to the CO serving the recipient, and from there to the desired recipient.

When you use a modem to connect to school, office, or to a dial-up Internet provider, you are using circuit switching. Provided you are willing to wait out the time required to initiate the call, and are willing to take the risk that sometimes there will

be more callers on the network than there are circuits (as happens when you hear the rapid busy signal), you pay for the connection only when you use it.

The opposite of circuit switching is to use a *dedicated circuit*. In a dedicated circuit, the wire you use is yours alone. You use it anytime you want, with no competition, and you pay for it whether you use it or not.

Dedicated and switched circuits were the basis of all wired and wireless communications from the beginning of electronic communications (about 1876) until a time just after World War II, when network supervisors decided to do something about a lingering problem with both systems. The problem was one of continuous underutilization, and the solution was a system of forced sharing between many users, called multiplexing. In a voice conversation, there are pauses (at least for the majority of people). During these intervals of silence, the two callers need not be connected. The line can do something else while two people are saying nothing to each other, as long as the parties are hooked up again before either of them has the next thing to say.

Over the years, multiplexing has taken many forms. Sometimes the contents of the line were shifted in frequency, so that one conversation theoretically could not be heard by the other. This system was called frequency division multiplexing (FDM). Other systems seized the line and rotated it between a number of conversations, returning it to you before you noticed it was gone. This was called time division multiplexing (TDM). A few systems capitalized on the nature of the phone systems themselves to create extra, *phantom circuits* by borrowing one of the two lines of your circuit and one of the two lines of a neighbor's, and allowing other users to conduct conversations over them.

All of these systems increased the capability and capacity of systems, but they did little to overcome another key drawback: how to handle retransmission of missed or scrambled message segments, or missed or garbled speech? If a message was mangled in transmission, it was necessary to determine that fact, then to notify the sender, and await retransmission. All of this took time and resources. And while it was all getting sorted out and resent, current messages often had to wait.

Packets, Addresses, and Ports

To increase the reliability of communications, a new paradigm developed called *packet switching*. In a packet-switched network, messages are chopped up into chunks of uniform length, called *packets*. A packet-switched network gives each packet an individual address label and then shoots it out onto the network, trusting that each packet will eventually make it to its destination, although the packets of a single message may travel by different routes. The packet address includes mechanisms to insure that all packets are accounted for and assembled in the proper order at the other end of the link. (Broadly speaking, this is called User Datagram Protocol, or UDP.) Other mechanisms provide the capability to recognize which packets have been corrupted or delayed, and to facilitate retransmission of replacements.

This is generally called Transport Control Protocol (TCP). TCP and UDP together work with the IP addressing system to facilitate all the services that the Internet provides to users. Additional protocols and services are also at work behind the scenes to keep the network smoothly running.

So how do packets help make networks and the Internet reliable? First, a packet travels over the circuit quickly. If it goes missing, its replacement can be retransmitted without taking a long time. When dealing with larger messages, the entire message has to traverse the circuit before it can be checked for errors in transmission. If an error is detected, the entire message must be transmitted again instead of just the errant packet or packets. Second, because it is understood that packets may take one of several possible routes to their destination, there is a possibility that packets may actually spend part of their journey traveling in parallel, rather than waiting constantly for the packets in front of them to move along. This parallel movement may increase overall transmission speed.

Packet transmission revolutionized communications, first remaking data transmission networks, then reforming the methods used for voice traffic, and finally, video. Packets have been with us since the early ARPANET, the precursor to today's Internet. They are likely to be with us through the foreseeable future. In fact, the successor to packet communications, asynchronous transfer mode (ATM), the system on which most of today's heavy communications backbones are built, refines packets further. Instead of using packets of roughly even length, ATM uses *cells* that are precisely 53 bits long.

There is much more that can be said about digital communications and packets, but we'll go into only two more concepts: addresses and ports.

An Internet Protocol address describes a location on the network. Technically speaking, a network address, or an IP address in the case of the Internet, is a *logical address*. Users don't often need to concern themselves with this logical, network address because the network has services that correlate such addresses with more friendly names, such as those that end in *.com* or *.org*. Beneath this layer of friendly conversion, however, logical addresses hold sway. The use of logical address allow the network to route packets to the correct part of the network, without concerning itself where in the world that location is. In fact, it is somewhat difficult to know by inspection where a packet actually goes based on its IP address alone, a concept that lends the Internet its "no-borders" characteristic.

IP addresses generally take the form of four numbers, separated by periods, in which each number is between 1 and 255. For instance, the IP address, 192.168.32.12 leads you to a destination on your network that's administrator-designated. Between the action of the conversion layer and the protocols that work with the IP address, the packets that compose the messages get delivered.

Just as there is a conversion layer between the common name and the IP address, there is a conversion system that connects the IP address to a specific network

device. This system is called the Address Resolution Protocol (ARP), and the address of each device is called the *physical address*, or in network speak, the machine access code (MAC) address. Think of the MAC address as you would the vehicle identification number (VIN) that uniquely identifies every automobile that is manufactured, and the IP or logical address as the license plate, or the identification that is required to operate the vehicle on public roadways.

There is one more component to addresses, and in some ways it is the most important of all, at least from a security perspective. It is important that packets are identified by function, that is, that their addresses contain some clue as to what they are intended to do. This allows them to be switched to the correct location by inspection (like reading the license plate) without having to open them up and examine their contents. For this reason, it is not uncommon for packets to have the same IP address, and to use the port number to state the packet function, so that the packet will be routed properly once it makes it to the device. This is similar to the various destinations used for things that come into a house. It is traditional, for instance, for mail to be delivered to the mail box or slot, for the newspaper to be placed or thrown onto the porch, and for water and gas to come into the house through the appropriate pipes. (To awaken and find natural gas coming out of the kitchen faucet, water pouring out of the oven, the mail in the flower bed, and the paper through the front room window is generally a sign that it will be a bad day.) In the physical world, getting these delivery functions right is the job of the respective couriers and plumbers. In the network world, it is the job of the *port number*.

Each IP address comes with roughly 65,000 port numbers, which can be thought of as cubbyholes, or individual message slots in a hotel lobby. Different types of network traffic use different ports. This is how the network keeps things straight, or knows to deliver mail in one place and newspapers in another.

Why do you need to understand this multitier system of addressing? Simple: *most network attacks in some way involve falsely manipulating or replacing the IP address, MAC address, or port*. In fact, one of the most important tools used today for network safety, the firewall, is based almost entirely on recognizing suspicious or invalid combinations of addresses and ports.

What Are the Network Protocols?

A protocol is a defined procedure for interconnecting and interacting. Within a protocol, defined behaviors allow the participants to move towards an anticipated condition or result. The protocols that determine how data are transported over the Internet, or over a LAN that uses TCP/IP, provide a variety of services. Some are directly visible to the end user, others operate in the wings or serve as aids in troubleshooting. Some protocols move web pages, some move email, some move files, and some move streaming media. Others exist to provide watchdog functions over the

system or to allow various network components to communicate. Many of the most important network protocols, which also happen to be most commonly attacked, are the protocols needed to make communication over a network possible.

Data Navigation Protocols

The fundamental network protocol is likely Internet Protocol, which describes how packets will navigate from network to network. It does this by defining the structure of IP addresses. The IP also provides a fragmentation and reassembly function, which means that if a message, or datagram, is too long, an IP packet can be split into smaller chunks for transmission through the network and then put it back together when it gets to its final destination.

What IP does not do is keep track of whether messages actually make it to where they are going, or pace the transmission of messages so that a link does not overfill. IP treats each piece of a message, or Internet datagram, as an independent entity unrelated to any other Internet datagram. The IP must link up with several other protocols to insure reliable end-to-end delivery and retransmission of missing messages.

For instance, the Transmission Control Protocol wraps itself around the IP packet and provides the information needed to see a packet through multiple hops to its destination and determines if all packets made the trip. TCP can figure out which packets were lost and order up replacements. UDP, on the other hand, is the stripped-down version of TCP, moving packets with speed but sacrificing end-to-end reliability.

The File Transfer Protocol mentioned previously operates using TCP. All the data travels reliably over the network, and the transmission is not finished until the packets have all made the trip and been reassembled in order at the destination. FTP's UDP cousin, however, Trivial File Transfer Protocol (TFTP), transmits the packets as a firehose transmits water. It streams, but it has no way to determine by itself if the water is hitting the target.

TCP can detect errors because each packet uses a cyclical redundancy check (CRC), which is like a parity or checksum, to check itself. A checksum is a mathematical mechanism that detects errors in transmission, usually by adding up the numeric value of all the characters transmitted and seeing if the total is the same at both ends of the link.

If IP needs to report errors to the sender, it uses helping protocols from a suite called Internet Control Message Protocol (ICMP). IP includes a facility for limiting the transmission of misdirected messages, a self-destruct mechanism called time-to-live (TTL). Every time a packet passes through a network node that retransmits it, the TTL counter gets decremented, that is, subtracted by one. If the TTL reaches zero before the Internet datagram reaches its destination, the Internet datagram is considered lost, and is destroyed.

Data Navigation Protocol Attacks

These four protocols: IP, TCP, UDP, and ICMP are the basis for Internet communications. They are also the basis of many attacks that use the Internet or of attacks against the Internet itself.

For instance, setting the TTL counter to a high or infinite value causes bad packets to crash on indefinitely. This takes up bandwidth and clogs the pipes so other traffic can't get through. False ICMP messages can put the network on alert against threats that aren't there, slowing or stopping communications, or causing unneeded reroutes. This can be particularly vexing when heavy traffic gets shipped down a narrow street. Network administrators will have a hard time limiting the flow without cutting off the desirable traffic.

TCP assures reliability by introducing sequence counters and acknowledgments to IP. Also, TCP transmissions proceed after the communicating systems give each other a *handshake*, that is, after both ends go through a short three-part exchange to confirm which other system they are talking with, similar to exchanging business cards before sitting down to make a deal. A favorite hacker trick is to open up a session (begin a communication) with a system under attack, receive an acknowledgment, and then leave the connection half-completed, tying up resources and memory on the attacked device. Do this enough times, and unprotected systems will buckle under the load, similar to meeting too many interesting people at one time at a party. Affected systems can hang up or cease functioning, denying services to legitimate users, or they can crash, possibly allowing attackers to modify the operating software with illicit changes that can create secret entrances to open the device to attackers.

Why is IP such a pushover? Because it's not being used for that which it was built. The military wanted a network protocol that would survive a worst case scenario—something along the lines of global thermonuclear war. The network needed to pass traffic to every location smoothly and efficiently, and to be able to reconfigure itself around bad routes and sudden outages. Were the balloon to go up, and everyone except people in deep bunkers had to spend two weeks hiding in basements and under doors in slit trenches, while waiting for fallout to decay, the network was supposed to reconfigure itself and be ready to go once humanity reemerged, serving every place that was still a place.

Instead, the Internet became an “information superhighway” that led to economic growth, prosperity, and jobs. It became a tool of enhanced communications, helping to bring the entire human family closer together. True enough, there are robbers in the bushes around that highway, and attacks for money, both of the travelers and of the destinations, are increasingly common. There is also increasing concern about pedophiles who use the Internet to form associations with unsupervised innocents. These are unintended consequences against which the Internet was never fortified.

Other Internet Protocols

FTP, SMTP, and lots of the behind-the-scenes protocols use datagrams to communicate. These protocols can be subjected to attack. The easiest way to attack these datagrams is by monitoring the network using a *packet sniffer*. A packet sniffer surreptitiously monitors and decodes packets, allowing the attacker to gather information about the network and the devices and persons attached to it. A more sophisticated attack would be to change the contents of a datagram (data modification) or to make it appear as if it came from a different party (spoofing). Packet sniffers are, however, very useful tools for network administrators because they allow you to see what protocols are on the network.

File Transfer Protocol

The File Transfer Protocol was designed to promote sharing files, such as computer programs or data, by connecting machines reliably and efficiently, without getting tangled up in whether or not the host machine was the same brand or used the same operating system as the client. As a result, remote access of computers became more commonplace. In fact, even though simple FTP terminal programs are available, web browsers can often perform such transfers simply and transparently. This appears to be in line with the original intent of the standard.

However, the FTP protocol is subject to abuse. In the first place, it transmits in the clear without encryption shielding. Just sit and listen to a network connection, and in time, files by the boatload will come streaming by for the copying. FTP is also very subject to anonymous access. This is highly desirable in many environments, where to regulate access requires issuing passwords to every applicant, creating a tremendous administrative burden. It also means, however, that while you are in the process of gathering usage information from your FTP server, you will not learn anything more about me than I choose to reveal. Like an honor system coffee club, FTP is vulnerable to those lacking honor.

Simple Mail Transfer Protocol

The SMTP is designed to transfer email messages reliably and efficiently, again without regards to the particular computers or operating systems encountered along the way. It does this by setting up a channel between the initial sender and a receiver, which can be either the ultimate destination or some waypoint. Once the transmission channel is established, the mail sender issues a MAIL command, which identifies the sender and states that there is traffic to send. If the mail receiver can accept mail, it responds with an OK reply. The mail sender then sends a RCPT command identifying the mail recipient. If the mail receiver can accept mail for that recipient, it responds with an OK reply. If not, it responds with a reply rejecting that recipient (but not the whole mail transaction).

The mail sender and mail receiver may negotiate with several recipients. When the recipients have been negotiated, the sender sends the mail data. If the SMTP receiver successfully processes the mail data, it responds with an OK reply.

In the case that mail is sent to an intermediary stop, or waypoint, the process is repeated. If the mail receiver is the intended destination, the message is forwarded to a mailbox for storage until the recipient calls for it with her mail client.

Mail that can't be delivered because of incorrect or invalid addresses are returned with a note from whichever mail server determined the problem, stating that delivery was impossible.

The SMTP system works so well that email has become an important means of doing business. This same reliability, however, is its undoing. Email is normally transmitted in the clear, which means that a host that pretends to be an email relay can access all email that passes through it; mail could then be copied or modified. When an attacker suspects that a user or administrator is getting suspicious, it is relatively easy to disconnect the relay and lay low. The flow of message receipts and returns may be delayed but will likely not be disrupted because of the self healing nature of the robust SMTP protocol.

Further, it is very easy to create an email message that looks as if it was sent from someone other than the true sender. This can create problems in its own right (for example, a university student notifies everyone in a class that a certain test has been cancelled, and the message appears to emanate from the professor's computer). This also makes it easy to formulate an attack that sends tens of thousands of emails out to various addresses on the Internet, valid or not, using the spoofed return address of someone you wish to annoy or attack. As the emails bounce off the bad recipient addresses, your target will get a flood of annoying messages saying that the address is no longer valid. A few of the addresses will be valid, so your victim may get a couple of irate responses from legitimate but uninterested recipients as well.

SMTP and spam. The ability to spoof a return address and easily mail the same message to multiple recipients has lead to the uncontrolled outbreak of junk email, or *spam*. Spam, by some accounts, represents up to 50% of email traffic and is popular for one reason: email is dirt cheap. Junk physical mail is annoying, but it has a certain cost of delivery, and each piece must be handled and addressed, even if the name used is "occupant." This inherent cost limits the total amount of junk mail sent out because someone has to pay the bill. Email, on the other hand, has few costs: scraping up a few million email addresses off newsgroups and chain letters is not really that hard, especially if software is used to scan newsgroups and other places where addresses are densely displayed. Plus, launching and sending such messages is largely automatic; just turn on the spambot machine, and forget it.

Some email recipients resent the intrusion, and send responses asking that they no longer be disturbed. This proves the user's email is valid and increases the value of

the address when compiled into lists sold to other spammers. Other addresses are nonexistent or already abandoned. Email systems will send back notices to the sender to this effect. As the bad addresses bounce back and fill up the sender's inbox, there are no worries: it probably wasn't a valid inbox. Most spammers are careful to avoid using their real Internet addresses.

Tracking spammers down requires a lot of detective work; for example, you have to request that ISPs and the owners of various intermediate systems check their logs. This makes detection difficult. Because most spam today leads respondents to a web page anyway, dead-letter information is not really important to the sender. In fact, as stated, the most malicious spammers mimic a return address of someone they wish to annoy, letting them deal with the crush of dead-letter notifications, which often come in such volume as to shut down the unlucky victims email account in order to protect the server.

The cost-per-thousand of spam is so low that any spammer with technical savvy and a little software can send millions of junk emails a day from a very humble facility. The potential returns make it worthwhile grubbing about for addresses.



In an experiment I completed recently, a valid but unadvertised email account was left active but unopened for one year's time. At the end of that period, several hundred megabytes of email had accumulated—over 17,000 pieces in all. All of it spam.

Recent rule changes have made it illegal to send spam. However, just because a law is in effect in one country does not mean it's valid in others. Many spammers operate remotely, from countries that don't get too negative about earning revenue from a nonpolluting, high-tech industry such as bulk email. Other spammers use hacking techniques to turn ordinary computers, perhaps yours, into machines that will either generate or relay their spam for them. The best way to cope is likely the same way you cope with unsolicited physical mail. Use the antispam features of your email client software to filter undesired email into the recycle bin before you even see it.

Domain Name Service

Of the many other protocols inherent in the Internet, most are subject to attack or subversion. The Domain Name Service (DNS) has its own vulnerabilities.

DNS is used to resolve a friendly name, such as *www.oreilly.com*, to an IP address, such as 192.168.32.10. DNS is needed because while the Internet runs with IP addresses, people tend to think in words. The DNS service keeps a distributed directory handy, which allows you, the user, to type a uniform resource locator (URL) or Internet address, something that in most cases is fairly straightforward and easy to remember, into the address block on your web browser, and the computer will sweat the numbers.

DNS is not usually the first step in address resolution. To save time and prevent wasted bandwidth, a table of address and their URLs is usually stored or cached on the local machine. Your computer starts at this table when you make a web request, looking to see if it already has the IP address of the site you desire. When your local machine cannot find where to send a web request, it contacts the nearest DNS server, which tells the computer every thing it knows about the desired IP address. If the address is unknown at the DNS server, that DNS server consults the next DNS server up the chain, until your address is found or you hit the top, come up empty, and are sent an error message.

This suggests three very convenient DNS attacks. First, if you seed the local machine's cache with incorrect data, it sends the user's communications to the wrong place, including possibly a decoy site of the attacker's own design. Second, if you pollute the database of one of the nation's big DNS servers, you may shut down a major portion of the Internet, which is always good for achieving status in the cracker underworld. Fortunately, the distributed nature of the DNS system makes this a little far-fetched because backup systems will likely kick in. Third, if you deny access to the DNS server that provides address resolution to a population of users, say the LAN that serves your company, users of that LAN are not going to be able to contact web sites for which they do not already have IP addresses. Attackers do this in at least two ways: take out the server with some kind of attack or change the place that your desktop computer looks for DNS resolution. It may be easier to force a DNS error by changing the place computers look for DNS by modifying the local information in cache than it would be to take down the server.

Poisoning the DNS system doesn't only slow down or prevent the access of web pages and services. Mail may not work, remote filesystems may be rendered inaccessible, and network printing may go down. Essentially everything that involves an external communication is at risk when DNS fails.

Dynamic Host Configuration Protocol

To access the Internet, users need an IP address. But there are situations in which there may not be enough IP addresses to serve all users. (This used to happen frequently when networks expanded beyond their original projected size, and administrators discovered that they had not reserved enough address numbers.)

To overcome this shortage, a system to share IP addresses was created, called the Dynamic Host Configuration Protocol. DHCP provided an IP address to those users who were actually logged on at the moment, drawing them from a pool of all available IP addresses. The pool was usually much smaller than the number of users, but that was okay, because all the users were rarely in the office or using the Internet at once. Shortly after the user logged off, the IP address could be reassigned to another user that was just logging on. Oversubscribing, that is, operating with fewer addresses than computers, is a tactic also used by phone companies to provide everybody sharing a common bank of equipment.

Network Address Translation

Network Address Translation (NAT) has also become a popular way to share addresses among many users. With NAT, the IP addressing system inside the network is known only to those in the network. Outsiders see only a small number of external addresses, which are rotated among all the users who may need one. A table in the NAT server keeps track of which internal addresses map to which external addresses. The number of internal addresses can be almost unlimited (after all, they stay inside the closed network). Because these addresses never appear to the outside world, NAT network administrators usually adopt the Private Address space allocations set aside in the IP address system.

RFC 1918 Address Allocation for Private Internets

The Internet Assigned Numbers Authority (IANA), the organization that allocates the assignment of IP addresses, created the following three blocks of the IP address space for private internets.

- Class A 10.0.0.0 to 10.255.255.255 (10/8 prefix)
- Class B 172.16.0.0 to 172.31.255.255 (172.16/12 prefix)
- Class C 192.168.0.0 to 192.168.255.255 (192.168/16 prefix)

Home networks and Wi-Fi wireless systems likely use addresses that fall into this assignment.

In some Microsoft networks, the Automatic Private IP Addressing (APIPA) addressing scheme may come be used in cases in which a DHCP system is nonfunctional. When the DHCP server is again able to service requests, clients update their addresses automatically.

Before NAT, DHCP was used to ration IP addresses. With NAT, DHCP is used to tell a PC which IP address it has been issued. This is in line with the root of DHCP, which is called the Bootstrap Protocol (BOOTP). BOOTP and DHCP tell a recently awakened PC many things it needs to know about its configuration, such as what address it can use and where it can locate various network resources. This is needed in case these things have changed while the computer was turned off. It is also necessary to receive this configuration information in case a computer is new to the network.

If the DHCP server is compromised and starts issuing wrong information, the ability of the computers on the internal network to access the external network or the Internet will be severely limited.

Attacks against DHCP usually involve interrupting these processes. For instance, one item that is frequently shared by DHCP is the location of DNS servers. If the DHCP server is compromised and starts issuing wrong information, the ability of the computers on the net to access the Internet will be severely limited.

This is not the only DHCP attack, however. Another popular attack is to change the pool assignments so that DHCP starts to issue IP addresses that are either invalid, or which are in use elsewhere. When this occurs, the routers and switches learn these new addresses and share them, and soon much of the traffic on the network can be going to the wrong place. Further, it may not be long before duplicate IP addresses begin to appear on the network. Many pieces of network equipment will blacklist devices that are using illicit, duplicate addresses. Finally, the routers and switches themselves will begin to labor under the strain of having to update so much information, and soon the network will be severely degraded. And this is not all. Bad address data must be purged, and good data must repopulate all the cache tables that need it. This takes time, is a burden to the network equipment, and it consumes bandwidth as well.

Port Address Translation

It is possible to overload a single external NAT address so that it can be used by several internal users. You can use port numbers in addition to network addresses to keep all ongoing exchanges organized; the resulting system is called Port Address Translation (PAT). (Ports were compared previously to the various services such as gas and water that entered a house separately, even though they were at the same physical address.) In a sense, PAT fills part of the role of DHCP because it shares a small number of public IP addresses, that would be one (1), with a larger number of users. Unlike DHCP, which may open some security holes, NAT and PAT can actually increase security because they obscure the true addresses used by users.

The Fragile Web

Of all the languages in the world, it is somehow fitting that the one that unifies most people in a common experience is a product of the Electronic Age. It's not English, Chinese, or even Japanese; rather it is Hypertext Markup Language (HTML). One of the most important protocols on the Internet, HTML is used to format the text and images people place in their web pages.

The actual heavy lifter for web work on the Internet is Hypertext Transfer Protocol, a special protocol that carries HTML codes and other information over the Internet efficiently.

It would follow that anything so useful as HTML and HTTP would be subject to special attacks, and this is precisely the case. A whole family of attacks can be perpetrated against these formats, and when web pages become corrupted, they can actually become instruments of attack and destruction themselves. Attacks can take place on the user's end, in which case they are called *client-side attacks*, or on the servers that send out the web pages that clients view, in which case they are called *server-side attacks*. This section describes the most important client- and server-side attacks.

How HTML Formats the Web

HTTP is a sparse language with a few terse commands such as “GET.” What HTTP “gets” is a small file of HTML code.

HTML is blocks of text surrounded by commands. The commands are generally embedded within greater than and less than signs, which are called start tags (`<command>`) and end tags (`</command>`), respectively. The start tag instructs the user’s web browser to do something, say turn on the bold text function, and the end tag tells the browser to stop doing whatever the start tag kicked off. Thus the command that makes text bold applies to all the text between `<bold>` and `</bold>`. Given the many available text and commands in the HTML language, lots of effects are possible.

More advanced commands deal with inserting images on the page. Image commands are actually a call back to a server to download a picture, which the HTML code positions on a user’s screen.

Hyperlinks are also important. A user activates a hyperlink by clicking on it. This calls a new file from the server, which may be a whole new web page. Hyperlinks greatly increase the interactivity of a web site, while making it easy to organize a lot of information with a point-and-click interface.

You can view HTML code simply by calling up a web site and clicking on the browser’s View Source command. It is usually located on the command line at the top of the browser window. If you cannot find it, use the browser’s Help function to look it up.

Advanced Web Services

Merely displaying text, graphics, and photos, might make an attractive web site, but is unlikely to make much money. With increased advertising, the Web has become commercialized, and innovation, not all of it strictly beneficial, has become commonplace. In the first place, ads, instead of images, can be downloaded to users. Banner ads, for instance rotate among a variety of sponsors, who pay for page views, knowing that some percentage of those who view a page will be exposed to the ad. In order to make the ads change from time to time, the embedded commands needed to be a little bit more sophisticated. Various scripting languages have been adapted to this purpose.

A web page script is a little bit of computer code that executes either as the page loads into the computer from the server or on the user’s computer as the page is displayed. The scripts that execute on the server are called *server-side scripts*; scripts that run on the user’s machine are called *client-side scripts*. Using a script, complex functions and displays can be generated programmatically, greatly increasing the functionality and interactivity of web-based applications.

With the power of scripting comes also the opportunity for abuse. It is possible, for instance, to use scripting languages to force disallowed states on either the client or server machines, which can result in unpredictable results and crashes. Some scripts can even exploit vulnerabilities in the underlying operating systems. In this case, the scripts become exploits, and can cause serious trouble or damages. A good deal of work has now gone into trying to ensure that scripting languages on the Web are secure, and for the most part they are. However, every now and again, somebody somewhere discovers a new code sequence that can get around the controls and checks. Most browsers enable users to disable scripts and applets, or to run them in a restrictive fashion. Assuming users stay in the mainstream in terms of web sites, it should not be an issue.

What is a script?

The term script in a computing context dates back to the early 1970s and comes from the Unix operating system term shell script. A *shell script* is a sequence of commands that are read by the computer from a file. Using scripts, multistep commands can be embedded in a web page. Scripting languages have increased in flexibility and robustness; today, scripts can be executed either at the server as the page is called from memory or at the user's computer as the material is called up for display.

The advantage to client-side scripting is that it allows the local computer to check against data entry errors such as proper abbreviation of a state name, or the format of something such as a phone number, before the data crosses the network and ties up the server, or worse, creates an entry in the server which has to be corrected. The advantage to scripting on the server side is that it helps preserve confidentiality. Processes that involve company secrets are executed in the safety of the server room rather than distributed out to where they can be captured and analyzed. In addition, some processes are so intensive that it might bog down the user's machine to execute them on the client side. Anything that involves lots of tables or uses a database is usually best kept server side.

Scripting allows a lot of interaction between the client and the server. Responses from the field may be organized and checked by a client-side script, which transmits them to the server. On the server side, a script may use the information contained in a client's response to modify on the fly what the server sends back to the user. This way the application shows the user exactly what is required or what has changed as a result of the user's entry. A raft of conventional business applications can be operated in real time on the Web using judicious scripting. Focusing demands for computational power on the server also allows for the use of simpler clients. This means that cell phones, video games, and personal digital assistants (PDAs) can be viable terminals in complex interactions.

Client-side scripting languages

There are several different scripting languages. While a few are used both client side and server side, most languages are used on one or the other. Language choice is generally based on the specific features or characteristics of the language, adherence to company- or project-wide standards, or simply personal preference.

Occasionally the native browser does not have the program code required to execute everything that is asked of it. Certain media files, for instance, may require a special code to play back properly. A *plug-in* is a software component that loads automatically to extend the range of file types that a web browser can understand and execute.

These are the most popular scripting languages for client-side applications:

- JavaScript
- VBScript
- Java (requires plug-ins on the receiving machine)
- ActiveX Controls
- Macromedia Flash (requires a plug-in on the receiving machine)

Multiple scripts can run on the same web page, but languages cannot be mixed within the same script.

JavaScript (not to be confused with scripts written in Java) is one of the earliest languages and is supported by both major browsers. It was introduced first for Netscape Navigator and, when it was adopted by Microsoft for Internet Explorer, it was called Jscript.

JavaScript is based on computer languages such as C, C++, and Perl. VBScript, on the other hand, is based on Microsoft's Visual Basic. The major difference between JavaScript and VBScript is the command syntax. However, only Internet Explorer natively supports VBScript. To make VBScript work on Netscape Navigator requires a plug-in.

Java may require the use of a browser plug-in, but it is a popular language because it can operate on any computer that can host a Java Runtime Engine (JRE). The programmer must write the Java code and compile it, but thereafter it will run on any type of computer or device that has a JRE, with little extra programming effort. Java scripts for the client side are usually called *applets*. When they are constructed to use as server-side, they are called *servlets*.

Java has lots of power, but it has certain built-in security features that are designed to keep it from doing anything damaging to the machine on which it is running. These features create a virtual environment for the Java code to run in, formerly called a sandbox. The Java security manager keeps the Java code from crawling out of the sandbox. A Java that tries to escape the sandbox and do harm to the client is called a *malicious applet*.

ActiveX is a powerhouse language that features a lot of capabilities and a fair amount of danger. ActiveX controls can be integrated into web pages and use numerous functions such as buttons, drop-down list boxes, text entry, and display fields. ActiveX controls can take nearly complete control of the host machine, unlike Java, which is at least partially constrained by the security manager. ActiveX can cause real trouble if malicious code is included. On the other hand, ActiveX is powerful and flexible, making it highly attractive to programmers and developers.

Server-side scripting languages

When a server-side script runs, it is often because a user has submitted answers to an application or form. The user then downloads a page that shows the effect of the user's submissions. A general property of all server-side processing is that when a user submits a form, the server processes it and creates a new HTML page. The user downloads this page to observe the effect of what they have requested, in effect, the completed form.

The use of scripts on the server is divided between two main paths. Some set ups communicate with the server directly; others communicate via the Common Gateway Interface (CGI).

The Common Gateway Interface (CGI) is a middleman between the web server and the clients. CGI establishes the rules under which various programs will talk to each other. The purpose of the CGI is to translate between unlike languages and systems. CGI programs can be written in many languages. C, C++, Java, Perl, Visual Basic, or any other language that can accept user input, process it, and output a response, can be used in CGI scripts.

If the scripting language is portable between platforms, such as Java, the script can play in multiple environments, and CGI might not be employed.

Scripting languages for server-side applications include:

- Perl
- Active Server Pages (ASP)
- PHP
- ColdFusion
- Java Server Pages (JSP)
- Python

Which programming language to use is a matter of preference and performance. Perl, PHP, and Ruby tend to look like code written in C, C++, or Java. They bristle with curly braces. ColdFusion code, on the other hand, is stuffed with HTML-like tags. This bothers some programmers because the resulting code seems to lack precision. Metrics that matter include reliability, flexibility, ease of making changes, speedy performance, and an ability to wring the most performance out of the computer it runs on.

Support for the application is also important. A firm that specializes in Windows applications that also uses some Unix or Linux merely because one of the employees is a whiz at it may find themselves in a difficult position if that employee leaves.

Perl is a language tightly focused on the Web and web applications. It has a long history of service. Veteran programmers swear by it. Perl is also an open source language, which means it is produced and supported by a community that allows members to access the internal workings of the language. This may increase its inherent security because any changes will be seen by many eyes, rather than a few employees working under the cover of confidentiality agreements. Combining Perl with an open source web server, such as Apache, creates a powerful combination for web servers.

Active Server Pages can be written in JavaScript, Perl, Python, and VBScript. ASP has the advantage of being able to work with COM objects. These small, portable code objects (one line often does it) can provide similar capabilities in different applications. By reusing previously developed COM objects, developers can reduce the amount of code they must write and can more easily import advanced or complicate functions into the web server.

PHP is open source. Because PHP resembles C, Java, and Perl, this can shorten the time it takes to learn it if you are already an experienced programmer. PHP features easy interface with numerous databases and allows you to program on both Windows and Unix machines.

ColdFusion is a popular web development tool because it uses a system of tags called ColdFusion Markup Language (CFML), which resembles HTML. The ColdFusion development environment tries to automate the coding process as much as possible. This may limit coding errors, since less is done by hand.

Java Server Pages (JSP), or servlets, are a set of Java classes optimized for client server interactions. JSP operates with true Java, which brings portability, multithreading, extensive class libraries, strong safety features, robust security measures, and several other of Java's advantages. Java servlets are efficient because they are persistent. Once created, usually at the time of the first request, the servlets stay resident in memory as compact Java objects. When a subsequent request comes by, the servlet can quickly build the HTML page the client requires. With ASP, the code may need to be reinterpreted for every client request, slowing down the process of page generation.

Python is an interpreted, interactive, object-oriented programming language that resembles Perl and Java. The syntax is clear and avoids nested curly braces. Python is copyrighted but freely usable and distributable, even for commercial use. Python can run on many brands of Unix, Windows, Macintosh, and others.

Each of these programming languages can make the web experience better for the user and more efficient for the provider. However, each of these environments can be a source of trouble. Every language or technology has its own set of bugs and exploits

that server operators must track and patch against. It is important to monitor language-specific newsgroups and bulletins to learn about the problems and solutions.

Web Attacks and Preventions

The following sections describe both client-side and server-side web attacks, and what you can do about them.

Client-side web attacks

Client-side web attacks include the following:

Malicious HTML tags in web requests

Malicious code in a form window can cause the server to generate pages that are unpredictable or dangerous if run on the server. Malformed pages sent back to the client for execution may cause further problems.

Curative: Webmasters must not allow nonvalidated input. Use client-side scripting to clean up form data before it is transmitted.

Malicious code from other clients

A web site with a discussion group may be open to attacks of the form:

```
Hello Group- Here is my message!  
<SCRIPT>malicious code</SCRIPT>  
That is all!
```

If a victim client has scripting enabled, their browser may run this code unexpectedly.

Curative: Users should turn off script functions, web servers should screen for embedded tags that show a script may be present.

Clients sending malicious code to themselves

An attacker can slip a client a message or file and encourage them to post it to the server. When the server echoes or displays the posting, the client's machine may execute it.

Curative: Webmasters should screen data, even if the intended recipient is the client that sent it.

Abuse of tags

Tags such as <FORM>, normally harmless enough, can cause trouble if they're embedded at the wrong place. An intruder can trick users into revealing sensitive information by modifying the behavior of an existing form or can display information that may have been held in the form of a previous user.

Other HTML tags can alter the appearance of a page, insert unwanted or offensive images or sounds, break things, and otherwise disturb the peace by interfering with the page's intended appearance and behavior.

Curative: Set browser security to high and lower it only for those users you are sure will not violate that trust.

Poisoned cookies

While visiting a web site, a simple text file called a *cookie* is often placed in the user's computer. At the next visit, the web server scans for cookies, and if it locates one, can use the cookie data to recall the previous conversation. A poisoned cookie is one that has been altered to trigger the download of malicious code.

Curative: Keep security settings high until trust is earned. Scan all incoming files (cookies included) for viruses to prevent the injection of malicious code.

Using the wrong character set

Browsers interpret the information they receive according to the character set chosen by the user. If the user fails to specify a character set, the web server uses a default setting, which can result in garbled displays or unintended meanings.

Curative: Users should declare their character set when configuring their browsers.

General client-side attack preventatives. General measures that can be taken to help prevent client-side attacks include the following:

- There are a host of security options built into most browsers, but it is up to the user to tailor them to her specific situation and needs. Web page developers should filter their page output to eliminate these types of problems.
- Users should set security high and set scripting off. This may disable some web functionality, so users should know how to make changes to browser configurations as required.
- Remember the client may not be the intended victim. A carefully constructed attack may execute script on the client machine that is designed not to hurt the client, but rather some other computer or network to which the client is connected. For instance, the client may have cached security information when it last connected to a particular server, and this authorization information may be co-opted by the attacker in order to attack the server via the client.

Server-side web attacks

Servers can be attacked just as easily as clients, or perhaps more readily. Servers have the dual disadvantage of having to be exposed to many users, and possibly also to the Internet.

Buffer overflows

One of the most serious attacks against a server involves causing an intentional buffer overflow. Although the arrangement varies slightly from computer to computer and from operating system to operating system, in most computers, RAM memory is organized by roping off a piece for the operating system, then roping off a section to be used for temporary variable storage called the *stack*. Above the stack is cordoned off yet another section of memory, this one called the *heap*, after which is the memory storage spot for code waiting for execution.

If one of these areas, often the stack, suddenly grows too large, it may overwrite the area above it. This is called *smashing the stack*. When this happens the values that were stored in those regions are changed to whatever was being written into memory at the time the overwriting occurred. This may cause the computer to behave erratically or to crash.

If the values designed to be overwritten are chosen with extreme care, they may actually end up being stored, as if they were instructions. They may execute the next time the computer reads those memory locations. This is one way to inject arbitrary code into the server; such code could be instructions that allow an attacker to take over the computer.

One popular approach is to install a *back door*—code that allows an intruder to enter the machine without going through security first. Another approach is to install a *rootkit*—code that promotes the privileges of the attacker to that of the system administrator. Either would be a home run. It would be enough to install a code stub to facilitate installing more code later on. This can be as simple as telling the computer to go to a previously prepared web address or URL, where the malicious code is waiting.

Curative: The defense against buffer overflows is good programming practice. No user input should ever be permitted without first verifying that it is of the correct length and that it contains no characters that may be invalid or that may be misinterpreted.

Stolen cookies

Because most Internet transactions last only as long as you are logged on, all the information about that session usually dissolves as soon as the link is broken. Cookies remind the server who you are and what you talked about last time. If someone copies your cookies, the potential exists for them to trick the server into revealing what it knows about you. This may not be critical, but if the server has a credit card number on file, or personal information, this can expose too much information about you.

This becomes a more powerful attack if a really complex business process is involved. For instance, if one server hosts your credit card information and address information, and a second holds your academic records, while a third holds your current schedules, it may be possible to use minimal access to one server to trick the others into revealing a great deal. This exploits the server-to-server trust. These attacks are called *cross-site scripting* (CSS).

Curative: Cross-site scripting is a complicated exploit. Once the attacker is communicating from server-to-server as a peer, many unpredictable things can happen. The strongest defense is for each server to be properly protected in the first place.

Malicious ActiveX controls

As part of the .NET initiative, ActiveX allows powerful interactions between computers. ActiveX operates in a way that resembles a plug-in for Java, but

ActiveX does not use an isolating area of memory as does Java. Today's fine-grained access controls prevent a lot of Java problems because it confines the code to allowable portions of the machine it is running on. Malicious code using ActiveX, on the other hand, can have much more power. Such code can facilitate many computers working together, or it can be exploited, with devastating consequences.

TIME (XNTP) exploits

Many network processes require that servers and clients keep track of time, and that there is no variance between elements in the network. Time is kept the same all over using the XNTP Network Time Protocol. If an attacker can assign legitimate times to fake messages, the messages could masquerade as real ones. Alternatively, a legitimate message can be rendered suspect by shifting its time component to an illegitimate value.

Summary

Internet protocols, while they have allowed the internetworking of computers and the distribution of applications, have opened up a host of security issues. The protocols were meant to be robust, not secure. The enemy was never thought to be within. The trust by which computers are able to interact and complete complicated business processes turns out to be the undoing of much business, because sensitive information can be spilled and confidence lost. Further, the attacks that can be accomplished via the Web are crippling, because they can use the Web's flexibility to invade and cripple the computers that are attached to it.

In a way, society might be fortunate that the Internet was built first to be durable. This appearance of dependability allowed it to gain a following, then a foothold, then a share of mind, and finally enough dependence upon it that we now demand its weaknesses be fixed. Had the Internet been created with an isolationist mind, many of its applications and potential applications that we now struggle to protect may never have been developed.

PART III

Communications Security

CHAPTER 7

Encryption

The study of encryption is called *cryptography*, from the Greek *kryptos* meaning “hidden,” and *graphia*, meaning “writing.” The process of trying to decrypt encrypted information without the key (to “break” an encrypted message) is called *cryptanalysis*. The study of code creation and breaking together is sometimes referred to as *cryptology*. To write something in a cipher so that only those authorized to do so can decode and read it is called *encryption*.

Encryption is an ancient form of information protection that dates back 4,000 years. Encryption has taken on new significance in the modern computer age. It’s a particularly effective way to protect sensitive information—for example, passwords—that’s stored in a computer system, as well as information that’s being transmitted over radio or microwave channels and communications lines.

By changing or substituting or scrambling the order of letters and words, encryption has through the ages protected communications while they were being transmitted through a hostile environment—usually one involving war or diplomacy. Of course, once the message is received, it must be unencrypted to be meaningful. Thus begins the fascinating story of cryptography, literally “secret writing,” much of which forms the basis for computer and network security today.

Hundreds or even thousands of years ago, messages worthy of encryption might have included letters from a battlefield general to the home front. Encryption protected the communication in case the soldier carrying the letter was captured. In modern times, this might mean encrypting an electronic mail message containing sensitive information (of military, corporate, or personal importance) transmitted across a network. Encryption protects the information in case an intruder taps into the network.

Information that’s encrypted remains secure even when it’s transmitted over a network that doesn’t provide strong security—in fact, even if the information is publicly available. In most versions of the Unix operating system, for example, the file containing user passwords stores those passwords in encrypted form. Encryption protects these passwords effectively, to the point that if somebody does access the

file, encryption would make it very difficult for an attacker who obtained the file to be able to decipher the passwords.

Because encryption has historically been an expensive method of computer security (expensive in terms of product cost as well as computer time needed to encrypt), it has most often been used to protect only classified or particularly sensitive information—for example, military information, intelligence information, information about funds transfers, and information about the passwords in a computer system. Encryption is now becoming a more popular and inexpensive method of protecting both communications and sensitive stored data. For example, numerous web browser providers offer an encryption service to users. Inexpensive or free encryption software that can help ensure message security is available for PCs and wireless devices. As awareness of encryption benefits grows, as more laws mandate penalties for failing to protect information, and as encryption technology becomes more accessible and affordable, encryption is likely to be used as a matter of course to protect data—whether it's classified information being transmitted over a network, or ordinary user data stored on an office computer system.

This chapter describes basic encryption techniques and how they're used to protect data. Chapter 8 discusses communications security (of which encryption is an important part) and networking concepts, and elaborates on how encryption fits into overall communications security.

Cryptography is a complex topic. This chapter provides an introduction to basic encryption techniques, but it doesn't try to describe the mathematical basis of encryption algorithms or explore all the complexities of the topic. For detailed information, an excellent reference is *Network Security: Private Communication in a Public World*, by Charlie Kaufman, Radia Perlman, and Mike Speciner (Prentice Hall).

Some History

The earliest encryption systems, sometimes called codes or ciphers, date back to early Egyptian days—around 2000 B.C., when funeral messages consisting of modified or encrypted hieroglyphs were carved into stone—and were designed not to keep the messages a secret, but to increase their mystery. (This leads to the question: considering that hieroglyphs often contain of pictures of animals—could a dyslexic stone cutter have gotten into trouble for flipping the bird?)

The history of cryptography extends through the centuries from ancient Egypt to India, Mesopotamia, Babylon, Greece, and on into Western civilization and eventually to the dawn of the computer age.

From the Spartans to Julius Caesar, from the Old Testament ciphers to the Papal plotters of the fourteenth century, from Mary, Queen of Scots to Abraham Lincoln's Civil War ciphers, cryptography has been a part of war, diplomacy, and politics. Mary, Queen of Scots, for example, lost her life in the sixteenth century because an

encrypted message she sent from prison was intercepted and deciphered. During the Revolutionary War, Benedict Arnold used a codebook cipher to communicate with the British. Throughout history, governments and individuals have protected secret communications by encoding them. The development of ciphers and ciphering devices over the centuries has culminated in the complex computer-based codes, algorithms, and machines of modern times.

The protection of communications has always been particularly critical in times of war and political strife. The development of modern cryptography owes much to the research conducted under the pressures of World War II, and particularly to the breaking of codes produced by a machine known as the Enigma machine.

The Enigma machine was originally developed in Germany by an electrical engineer, Arthur Scherbius, during World War I. He offered an early version of the machine to the German navy and foreign office as early as 1918. The machine was originally rejected, but after some additional security enhancements were made to the commercial model, the German navy began using Enigma machines early in 1926.

The Enigma machine (shown in Figure 7-1) worked as follows: an operator typed the original text of the first letter of the message to be encrypted on the machine's keyboard-like set of buttons. The battery-powered machine encrypted the letter and, using a flashlight-type bulb, illuminated a substitute letter on a glass screen. What was special about the Enigma machine was a set of wheels known as *rotors*. Made of rubber or some other nonconducting material, the rotors contained electrical contacts, which were wired in such a way that turning the rotors would change the correspondence between letters. Before the encryption began, the operator would set these rotors to an initial position. When the operator typed the first character—"A", for example, the machine might illuminate a light corresponding to "P". The operator would then copy the letter "P" onto an encryption worksheet, the machine would advance the rotors, and the operator would enter the next letter. With the new rotor settings, the correspondence between letters would change. An "A" might now be translated to an "X". This process would continue until the entire message was encrypted. The encrypted message could now be transmitted by radio to its destination, usually a U-boat in the Atlantic.

At the other end of the communication, the operator trying to decrypt a message coded by the Enigma machine would need another, identically built Enigma machine and would also need to know the original settings of the rotors.

The first breakthrough in solving the Enigma codes came from Poland. In the late 1920s, the Poles formed a code cracking team that applied the science of cryptanalysis to work on breaking the German codes. Marian Rejewski and two other mathematicians cracked some of the early Enigma messages.

In the early 1930s in France, a German named Hans-Thilo Schmidt offered French intelligence some information about setting the Enigma keys. The French cryptanalysts didn't have the resources to take advantage of this information and the British

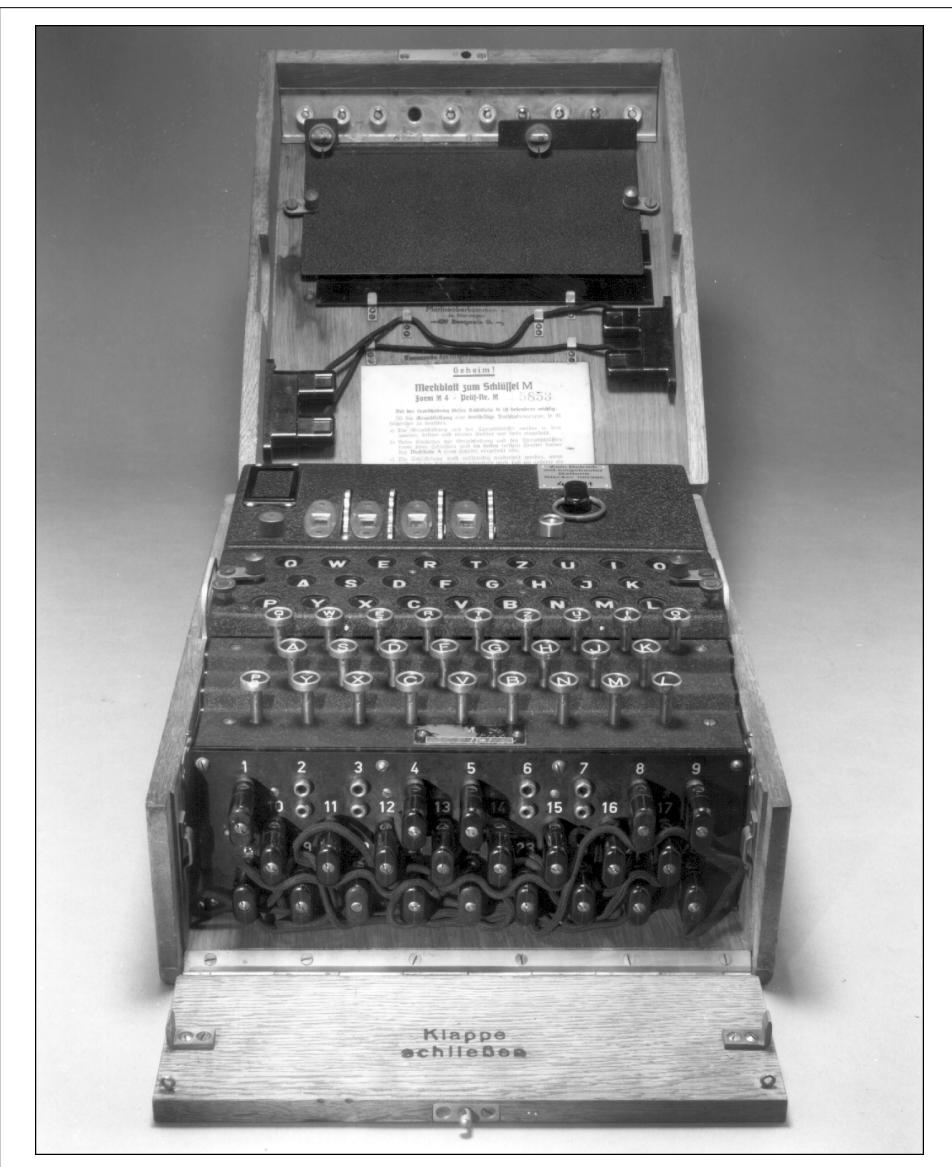


Figure 7-1. The Enigma machine (reproduced by permission of the Smithsonian Institution)

also rejected the information as being insufficient. The French offered the information to Poland, where Rejewski used it to make additional brilliant advances in cracking the Enigma codes.

After the fall of Poland in 1939, the Poles passed their information on to the French and the British. As the Germans continued to change keys and to modify the design of the Enigma machine, the British built on the Polish solution. Under the direction

of mathematician Alan Turing, and with the help of Enigma documents captured from U-boats sunk during the remainder of the war, the highly secret “Ultra” project began to decrypt German naval messages on a regular and timely basis. In the early 1940s, Americans—some of them from IBM—made their own contributions, based on knowledge they’d gained reconstructing Japanese diplomatic cipher machines through the “Purple” project.

In the decades since World War II, the use of computers to break codes has transformed the codebreaking game and has contributed greatly to the use of cryptography in military and intelligence applications, as well as in systems used in everyday computer systems.

What Is Encryption?

Encryption (sometimes called *enciphering*) transforms original information, called *plaintext* or *cleartext*, into transformed information, called *ciphertext*, *codetext*, or simply *cipher*, which usually has the appearance of random, unintelligible data. The transformed information, in its encrypted form, is called the *cryptogram*.

When encryption is used to send messages, it is reversible. After transmission, when the information has reached its destination, the inverse operation (*decryption*, sometimes called *deciphering*) transforms the ciphertext back to the original plaintext. (There are cases in which encryption is one way only. These will be explained later.)

The technique or rules selected for encryption—known as the *encryption algorithm*—determines how simple or how complex the process of transformation will be. Most encryption techniques use rather simple mathematical formulas that are applied a number of times in different combinations. Most also use a secret value called a *key* to encrypt and decrypt the text. The key is a kind of password, usually known only to the sender and the recipient of encrypted information. The encryption algorithm mathematically applies the key, which is usually a long string of numbers, to the information being encrypted or decrypted.

Unlike a regular password, a key doesn’t directly give you access to information. Instead, it’s used by the algorithm to transform information in a particular way. With the key, information that’s been locked (encrypted) by the key can readily be transformed; without the key, that information is inaccessible. The examples shown later in this chapter will help make encryption keys more understandable.

The type of encryption algorithm, the secrecy of the key, and a number of other characteristics together form what’s called the *strength* of the encryption; cryptographic strength determines how hard it is to break an encrypted message.

An important consideration in assessing the strength of any encryption algorithm is not whether it can be broken (given sufficient pairs of plaintext and ciphertext, any secret message—except one encoded with a so-called “one-time pad,” described later in this chapter—can theoretically be decrypted) but how likely it is that decryption

can be performed in a reasonable amount of time. A message that can be broken, but only with a network of supercomputers grinding away for decades, is very safe indeed.

Early cryptographic systems depended on the secrecy of the encryption algorithm to provide security. Gradually, cryptography has come to depend upon the secrecy, and usually the length, of the key to keep messages secret. The most reliable cryptographic algorithms, in fact, are now expected to pass through several rounds of public scrutiny before most organizations will trust them. The most secret codes, of course, have secure algorithms as well as secret keys. There is no point in giving the opponent a head start. It is tremendously difficult to create reliable algorithms and to be able to test them thoroughly, however. Only a few nations have such a capability.

Remember that a poorly chosen, or improperly protected, encryption key opens the door to an intruder, just as a shared or stolen password does. If an intruder gets access to an encryption key, even the strongest encryption algorithm won't protect your data. Also, public key encryption, to be discussed later, depends on the mathematical difficulty of deriving a private key when presented with a public one. Certain efforts to increase the strength of encryption keys have led to research in a number of areas. Most current codes deal with the relationship between prime factors of very large numbers. Others depend on a relationship established by elliptic curves crossing an axis. One field, called *quantum encryption*, uses the curious relationship of spinning photons of light to increase security and make it easier to detect if a message has been read.

Figure 7-2 shows simple encryption and decryption.

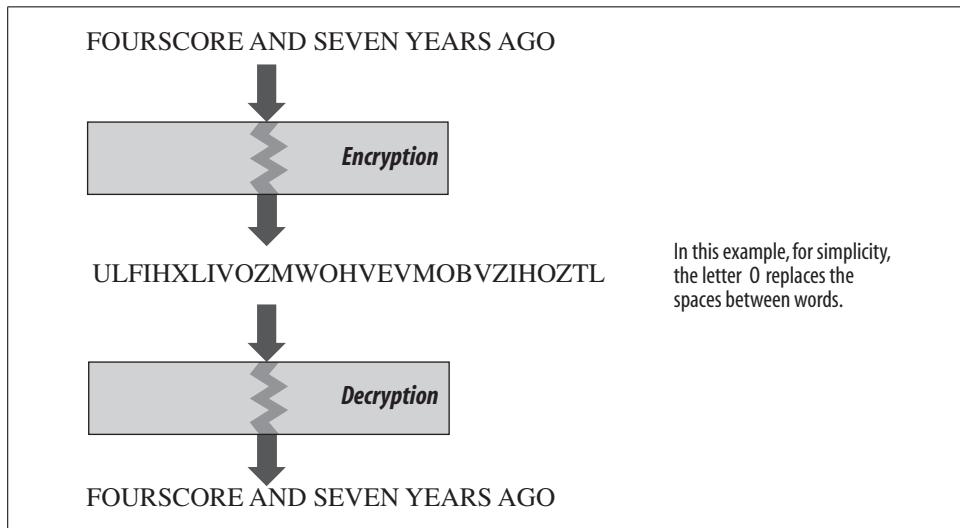


Figure 7-2. Simple encryption and decryption

Why Encryption?

Encryption provides security in three of the four security categories introduced in Chapter 1. (Encryption is not a particularly effective way to achieve the fourth category, availability.)

Secrecy or confidentiality

Encryption is very good at keeping information a secret. Even if someone is able to steal your computer or to access an encrypted file, that person will find it extremely difficult to figure out what's in the file.

Accuracy or integrity

Encryption is also very good at ensuring the accuracy or the integrity of information. In addition to keeping information secret, certain types of encryption algorithms protect against forgery or tampering. This type of processing detects even the slightest change—malicious or inadvertent—in the information. While military, intelligence, and many corporate users care a lot about secrecy, financial institutions are more concerned about accuracy: making sure that a decimal point or a zero hasn't slipped, or that an electronic embezzler hasn't rounded off a few transactions here and there. Integrity checking is also a way that network users can ensure that their communications have not been affected by viruses or other penetrations.

Authenticity

Encryption is also very good at making sure that your information is authentic, that is, that it comes from who it says it does. Certain encryption techniques let you confirm absolutely who sent a particular piece of information. This is extremely important to financial or legal transactions. An important authentication technique is a digital signature. A digital signature is unique for every transaction and is very difficult to forge. Digital signatures are described later in this chapter.

Transposition and Substitution Ciphers

There are two basic types of encryption ciphers:

Transposition ciphers

Rearrange the order of the bits, characters, or blocks of characters that are being encrypted or decrypted. They are sometimes called *permutation ciphers*.

Substitution ciphers

Replace the actual bits, characters, or blocks of characters with substitutes (for example, one letter replaces another letter).

With a very simple transposition cipher (shown in Figure 7-3), the letters of the original text (the plaintext) are scrambled. With this type of cipher, the original letters of the plaintext are preserved; only their positions change.

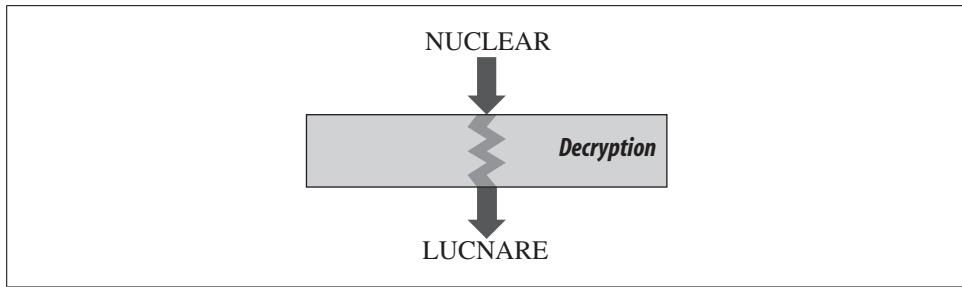


Figure 7-3. A simple transposition cipher

With a very simple substitution cipher (two variations are shown in Figure 7-4), the letters of the plaintext are replaced with other letters, numbers, or symbols. With this type of cipher, the original positions of the letters of the plaintext are preserved, but the letters themselves change.

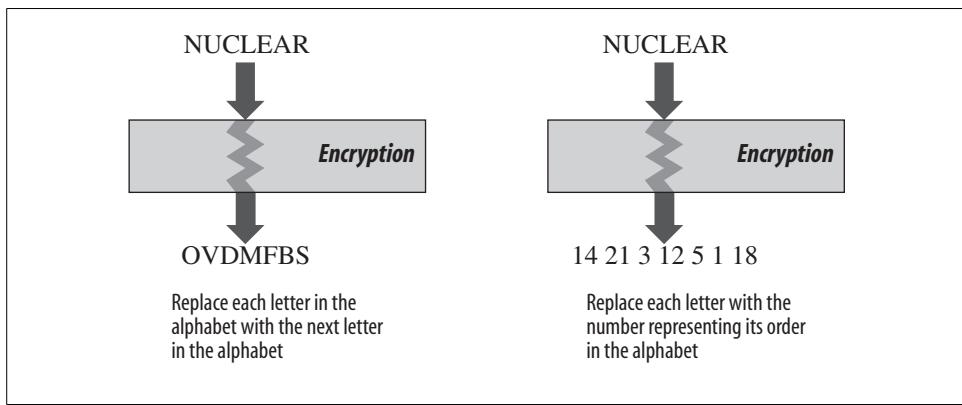


Figure 7-4. Simple substitution ciphers

More about transposition

In the fifth century B.C., the Spartans used a particularly interesting type of transposition cipher. During the Peloponnesian War, Spartan rulers encoded official messages by writing them on a long strip of parchment wound in a spiral around a wooden staff called a skytale. A message written in this fashion could be deciphered only by an official Spartan reader who had been given a baton of identical diameter. Thucydides, Plutarch, and Xenophon all have written about the use of this early cryptographic device.

Figure 7-5 shows another example of a transposition cipher.

More about substitution

Although earlier substitution ciphers existed, Julius Caesar's military use of such a cipher was the first clearly documented case. Caesar's cipher, shown in Figure 7-6,

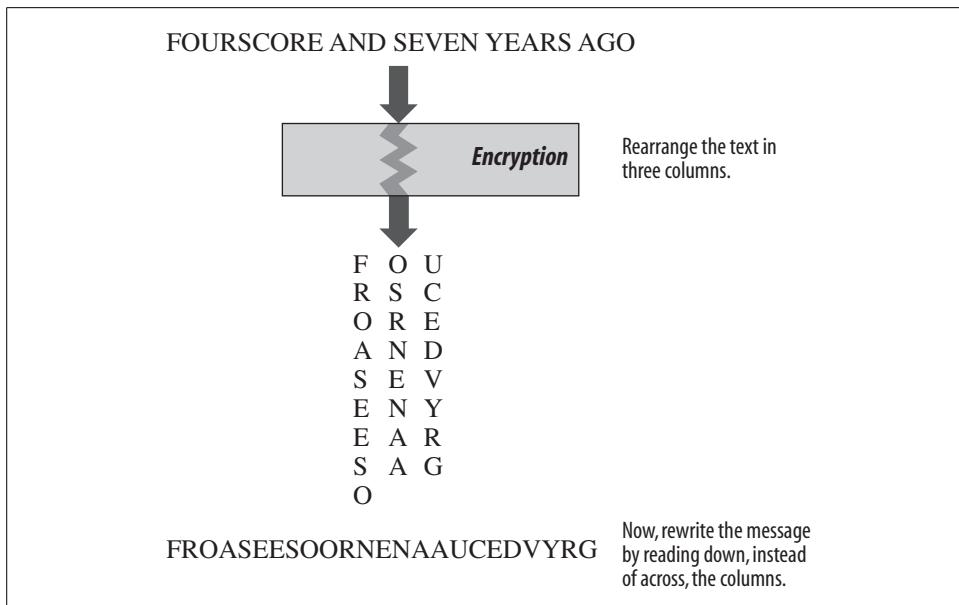


Figure 7-5. Another transposition cipher

was a simple form of encryption in which each letter of an original message is replaced with the letter three places beyond it in the alphabet.

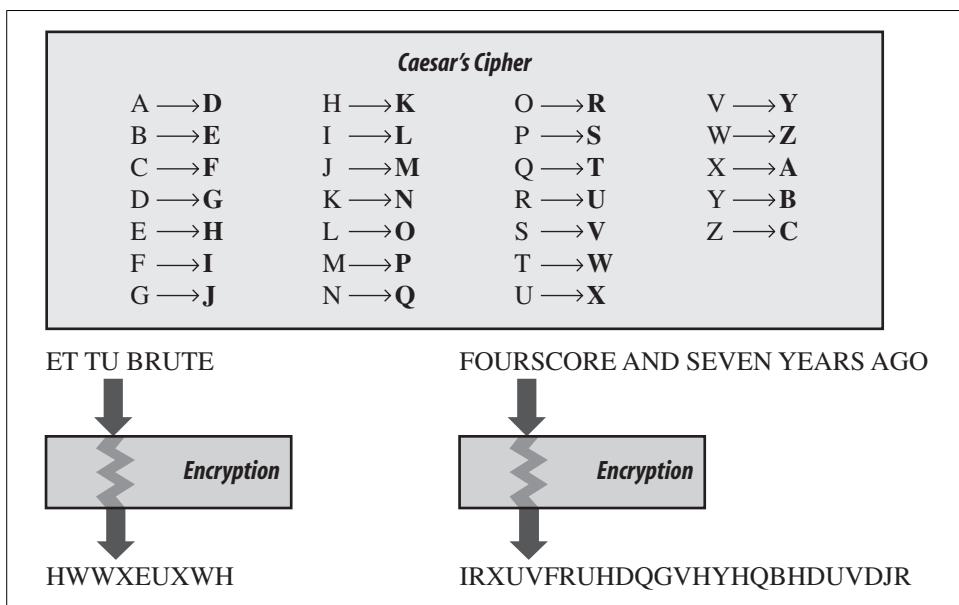


Figure 7-6. The Caesar substitution cipher

The cipher used in Edgar Allan Poe's short story, "The Gold Bug," is a good example of a substitution cipher. Another example from literature is the cipher used in Sir Arthur Conan Doyle's Sherlock Holmes tale, "The Adventure of the Dancing Men."

Usually, cipher alphabets are much more complex than these examples. Sometimes an alphabet will have multiple substitutes for a letter, sometimes the alphabet will include substitutes that mean nothing, and sometimes several alphabets are used in rotation or combination. This is called a polyalphabetic cipher.

The Enigma machine described earlier in this chapter used substitution to encrypt communications.

Cryptographic Keys: Private and Public

More complex ciphers do not use simple substitutions or transpositions. Instead, they use a secret key to control a long sequence of complicated substitutions and transpositions. The key and the algorithm work together to change the original information into an encrypted version of itself. In most cases, the operation of the encryption algorithm is fixed and well known. It is the key that produces the unique encrypted version of the information.

Modern cryptographic systems fall into two general categories (identified by the types of keys they use): private key and public key systems.

Private key cryptography

Private key (sometimes called symmetric key, secret key, or single key) systems use a single key. That key is used both to encrypt and to decrypt information. (See Figure 7-7.) A separate key is needed for each pair of users who exchange messages, and both sides of the encryption transaction must keep the key secret. The security of the encryption method is completely dependent on how well the key is protected. The Data Encryption Standard (DES) algorithm, described later in this chapter, is a private key algorithm.

Public key cryptography

Public key (sometimes called asymmetric key or two key) systems use two keys: a public key and a private key. Within a group of users—for example, within a computer network—each user has both a public key and a private key. A user must keep his private key a secret, but the public key is publicly known; public keys may even be listed in directories of electronic mail addresses.

Public and private keys are mathematically related. If you encrypt a message with your private key, the recipient of the message can decrypt it with your public key. Similarly, anyone can send anyone else an encrypted message, simply by encrypting the message with the recipient's public key; the sender doesn't need to know the recipient's private key. When you receive a message encrypted with your public key,

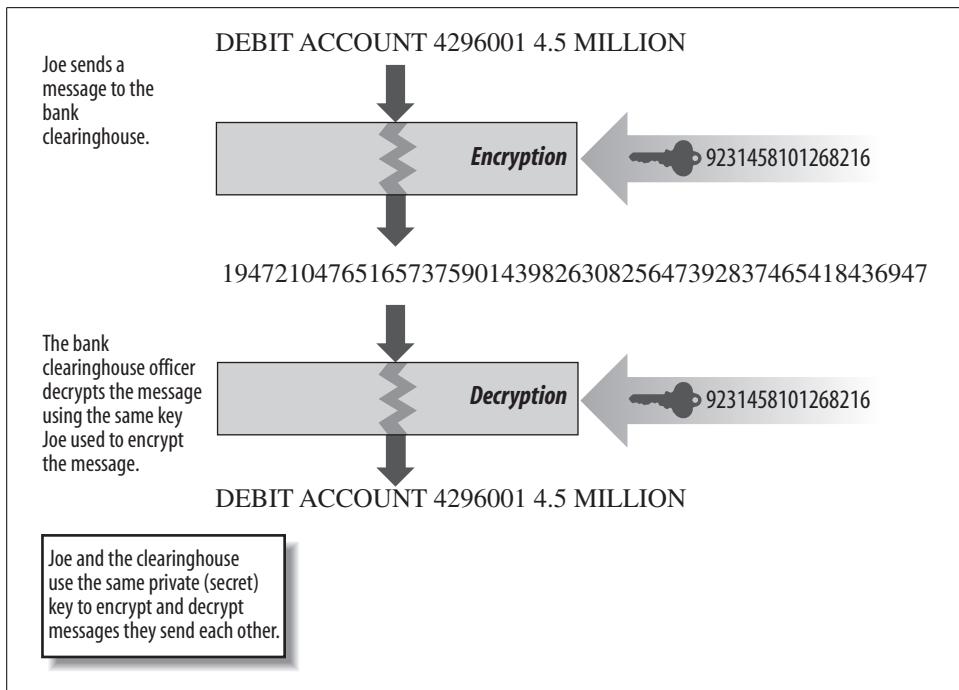


Figure 7-7. Private key encryption/decryption

you, and only you, can decrypt it with your private key. The RSA cryptographic algorithm, described later in this chapter, is an example of a public key algorithm.

In addition to providing an encryption facility, some public key systems provide an authentication feature which ensures that when the recipient decrypts your message he knows it comes from you and no one else.

In Figure 7-8, a banker named Joe uses his private key (known only to him) to encrypt a message. When the message is sent to the bank clearinghouse, the clearinghouse officer applies Joe's public key (known to everyone within the bank). Because decryption produces an intelligible message, the officer knows that only Joe could have created the message, and proceeds to follow Joe's instructions.

This example shows one of the disadvantages of public key encryption. Using a private key to encrypt a message and a public key to decrypt it proves that the message originated with who it is claimed, but anyone who can access the public key can decrypt the message. This provides authenticity without confidentiality. On the other hand, encrypting a message with a public key and decrypting it with a private key means that the message is secure (only the private key can decrypt it), but because anyone could have obtained the public key, it does nothing to demonstrate authenticity.

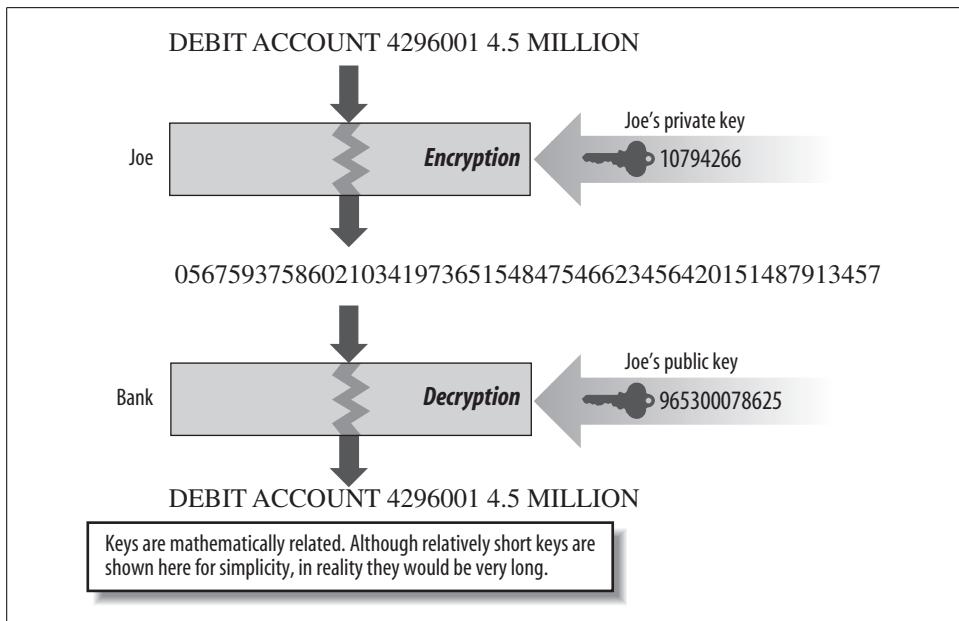


Figure 7-8. Public key encryption/decryption

The solution lies in combining a little bit from each, as shown in Figure 7-9. The sender can encrypt the message using his own private key, and then encrypt the result using the intended receiver's public key. Since only the receiver's private key can be used to decrypt the final result, secrecy is maintained. The receiver uses her private key to decode the doubly encrypted message and thereby obtains the message originally encrypted by the sender's private key. She then uses the sender's public key to decode that message, thereby obtaining the original message. She is confident in the authenticity of the message because if the message can be decrypted with a given public key, the message was encoded by the corresponding private key. It is assumed, of course, that the sender still has control of the private key or the computer or equipment that contains it, and that the sender was operating the equipment intentionally, and not under duress.

Encoding and decoding the entire message two times is unnecessarily burdensome. And as long as the security is being provided by the sender using the recipient's public key, only a small portion of the message need be encoded using the sender's private key to show authenticity. In fact, using a snippet in this way is the principal behind digital signatures.

Key Management and Distribution

A major problem with encryption as a security method is that the distribution, storage, and eventual disposal of keys introduces an expensive and onerous administrative burden. This is called the *key distribution problem*. Historically, cryptographic

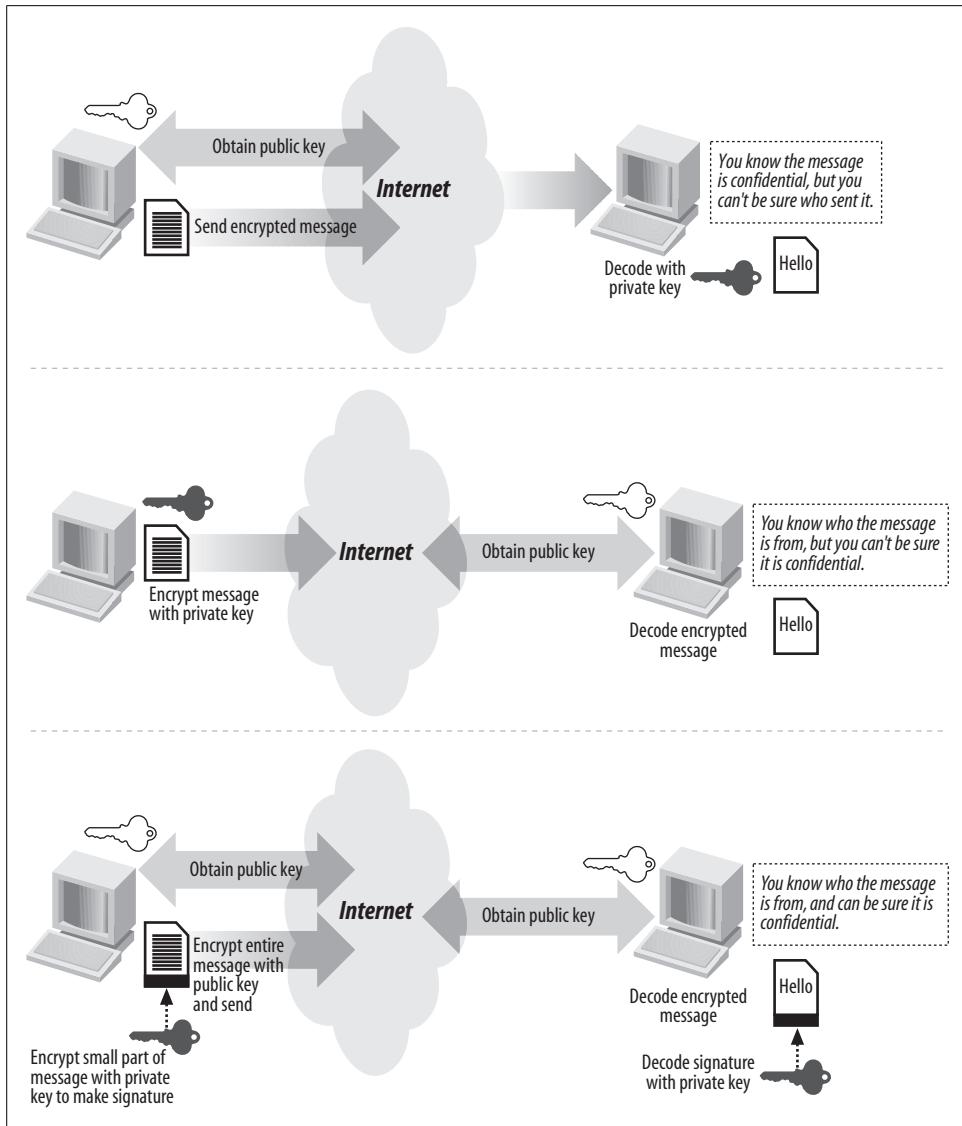


Figure 7-9. Public and private key encryption used together

keys were delivered by escorted couriers carrying keys or key books in secure boxes. In some cases, this is still the way it's done. With most modern high-security cryptographic products, government agencies do the actual key distribution, delivering the keys on magnetic media or semiconductor media to individual sites.

Another approach is to distribute a master key, which is then used to generate additional session keys. A site must follow strictly enforced procedures for protecting and monitoring the use of the key, and there must be a way to change keys at every site

that will be receiving the encrypted messages, preferably at the same time. Even with all these restrictions, there's always a chance that the key will be stolen or compromised. Keys are kept in secure areas in safes. Instructions given to couriers for modern mobile encryption units usually require that the security modules of these devices never leave the sight of cleared individuals. This includes trips to the restroom. Fortunately, the secure modules are usually pocket-sized, and can be detached from the actual encryption units.

Of course, if a key is lost, there's another problem. Because deciphering encrypted information depends on the availability of the key, the encrypted information will be lost forever if you can't locate the key.

The difficulty of key distribution, storage, and disposal has limited the wide-scale usability of many cryptographic products in the past. Automated key variable distribution is problematic because it's difficult to keep the keys secure while they're being distributed, but this approach is finally becoming more widely used. The Department of Defense-sponsored Secure Telephone Unit (STU-III) project is an example of a system that uses automated key distribution.

Standards for key management have been developed by the government and by such organizations as ISO, ANSI, and the American Banking Association (ABA).

One-Time Pad

One approach for encryption, called a *one-time pad* (see Figure 7-10) or a *one-time cipher key*, can be proven mathematically to be foolproof. As its name indicates, the pad is used only once, and the key must then be discarded or destroyed.

With a one-time cipher, you create two copies of a pad containing a set of *completely* random numbers. (These are numbers produced by a secure random number generator, possibly one based on some physical source of randomness. Sometimes, one-time pads are based on the process of nuclear radioactive decay.) The set contains at least as many numbers as the number of characters in your message. The sender of the message gets one copy of the pad; the recipient gets the other. On a computer system, one way to encrypt or decrypt a one-time message is to use a mathematical function called an exclusive OR, or XOR. When the sender XORs the message with the first copy of the pad, the process creates the encrypted message. When the recipient XORs the encrypted message with the second copy of the pad, the process recreates the original message. This method of encryption is also known as the Vernam Cipher, named for its inventor, Gibeert Vernam, who developed the cipher in 1918 for use with telegraphy.

One-time pads are sometimes used to encrypt important diplomatic communications, but they're not practical for most communications because of the difficulty of key distribution. (For each possible pair of users who might wish to communicate, a key has to be generated and distributed to those users; the key must be longer than

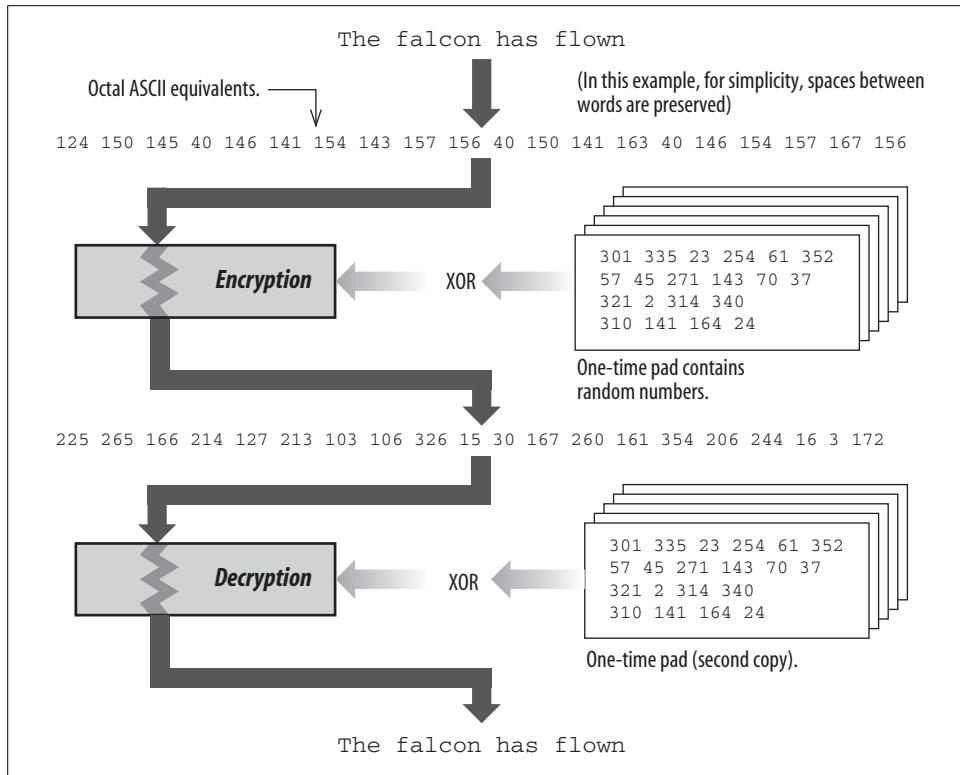


Figure 7-10. A one-time pad

all the messages they might wish to exchange.) One-time pads must never be reused. Most of the successful cryptographic attacks against these systems have involved operators attempting to reuse old pads.

Modern military communications equipment is often equipped with a long list of keys, up to several hundred. This may allow secure communications for up to a period of years. The keys reside with the unit and are never transmitted, but indexing messages communicate to the receiving system which of the keys it should start with, and how the keys increment thereafter. These units also come with instructions to operators instructing them to never leave the unit out of the care of the cleared person. An “overrun” command allows emergency erasure of the keys in the event the unit appears about to fall into untrusted or enemy hands.

End-to-End and Link Encryption

There are two basic communications levels at which encryption can be performed. With *end-to-end encryption* (sometimes called *off-line encryption*), a message is encrypted when it is transmitted and is decrypted when it is received. The network

may not even need to be aware that the message is encrypted. This type of encryption may sometimes be selected as an option by the user. The message remains encrypted through the entire communications process, from start to finish. This has the advantage of not depending on secure waypoints at every stage in the message path. This is the principle used by tunnel-style message encryption systems such as SSL, or TLS as it is known today.

With *link encryption* (sometimes called *online encryption*), a message is encrypted when it is transmitted, but is decrypted and then encrypted again each time it passes through a network communications node. The message may therefore be encrypted, decrypted, and reencrypted a number of times during the communications process, and the message is exposed within each node. With link encryption, the encryption is performed just before the message is physically transmitted. Encryption is typically invisible to the user; it is simply part of the transmission process.

There are advantages and disadvantages to each method of encryption. Advantages of end-to-end encryption are as follows:

- It is more flexible; the user may be able to encrypt only certain information, and each user can have a distinct key.
- It makes key management and distribution easier.
- It protects data from start to finish through the entire network.
- It is more efficient; the network doesn't need to have any special encryption facilities.

Disadvantages of end-to-end encryption are as follows:

- It may need to process some information (e.g., headers and routing information) in unencrypted form.
- Each system needs to be able to perform the same kind of encryption.
- It only secures the contents of a message, but does nothing to hide the fact that a message transmission has taken place.

Advantages of link encryption are as follows:

- It is easier; the user doesn't need to take any action.
- It is more convenient in a network that provides many nodes.
- It doesn't compromise the entire network if a single key is compromised. Each pair of network nodes can use a distinct key.
- It encrypts all information, including headers and routing information.

Disadvantages of link encryption are as follows:

- Key distribution and management are more difficult because all nodes in the network must receive a key.
- There are more points of vulnerability because the message is reduced to its original form several times along its path.

The Data Encryption Standard

During the 1960s, with the burgeoning of computer technology and concerns about the secrecy and privacy of communications, interest in a national encryption standard began to build. The idea of this standard was that it could be used by the many different types of government computer systems and networks, as well as in the systems used by government contractors, and potentially in commercial systems as well. The drive toward a national cryptographic standard culminated in the development of the Data Encryption Standard (DES).

Since 1965, when the Brooks Act was passed, the National Bureau of Standards (NBS, now known as NIST) had held the authority to research and develop standards for the protection of computer systems. The NBS study of government computing security needs, spanning the years 1968–1971, touched upon the need for an encryption standard. Development of the standard was clearly in NBS's bailiwick, and, with the cooperation of NSA, NBS initiated a cryptography program. The goal from the beginning was to develop a single public standard for protecting unclassified government or sensitive private-sector information—a standard that would be viable for approximately 10–15 years (a goal the DES far exceeded) and that would be able to be used on different types of systems (interoperability).

NSA already had its own encryption algorithms used for the protection of classified military and intelligence information. NSA lent a lot of technical support to NBS, including the evaluation of proposed encryption standards. However, for national defense reasons, NSA never intended either to share its own secret algorithms with the public or to use the public standard to encrypt classified communications.

In the *Federal Register* of May 1973, NBS invited vendors to submit data encryption techniques that might be used as the basis of a high-quality public cryptographic standard. Only a few responses were received, and these were unacceptable to the NSA evaluators. In August of 1974, NBS tried again. This time, with the prodding of NSA, IBM submitted an algorithm, which proved acceptable to NSA.

At IBM, work had been proceeding for some time on the development of several encryption algorithms. One was a 64-bit algorithm used to protect financial transactions. Another was a 128-bit algorithm known as Lucifer. IBM was particularly interested in the protection of automated funds transfers, especially those involving communication between online terminals.

There have been charges that NSA deliberately weakened the Lucifer algorithm before accepting it as the basis for a national cryptographic standard—some say to allow the agency to crack encrypted communications. These charges were fueled by the fact that NSA urged IBM to submit its algorithm, and by the modifications made in the Lucifer algorithm (the shortening of the key from 128 bits to 56 bits and changes in the algorithm's substitution functions, or S-boxes). A U.S. Senate panel has investigated these charges and upheld the integrity of DES.

NBS solicited comments about the DES in the *Federal Register* of March and August, 1975, and in a letter sent to Federal Information Processing Standards (FIPS) contacts in federal agencies. In an effort to be responsive to comments and to the controversy brewing around NSA's involvement in the algorithm, NBS sponsored two workshops. One examined the feasibility—both technical and financial—of cracking the DES through computational brute force. The other examined the mathematical basis of the DES. In addition, NBS discussed with the Department of Justice issues regarding competition.

The approval of the DES by the Department of Commerce in 1976, and its publication in 1977 as the *Data Encryption Standard* (FIPS PUB 46, updated and revised as FIPS PUB 46-1 in 1988) as the official method for protecting unclassified data in the computers of U.S. government agencies was a landmark in the history of cryptography. The approval included a provision that NBS review the algorithm every five years to determine whether it should be reaffirmed as a public standard. The DES was subsequently adopted as an ANSI standard.

The following FIPS PUBs and ANSI X3 (Information Processing) Committee publications contain standards for the DES and its use:

FIPS PUB 46-1

Data Encryption Standard

FIPS PUB 74

Guidelines for Implementing and Using the NBS Data Encryption Standard

FIPS PUB 81

DES Modes of Operation

FIPS PUB 113

Computer Data Authentication

ANSI X3.92

Data Encryption Algorithm (DEA)

ANSI X3.105

Data Link Encryption

ANSI X3.106

DEA Modes of Operation

ANSI's X9 (Financial Services) Committee has also published standards related to the use of DES in the banking community.

What Is the DES?

FIPS PUB 46 describes the DES as follows:

The Data Encryption Standard (DES) specifies an algorithm to be implemented in electronic hardware devices and used for the cryptographic protection of computer data ... Encrypting data converts it to an unintelligible form called cipher. Decrypting cipher

converts the data back to its original form. The algorithm... specifies both enciphering and deciphering operations which are based on a binary number called a key ... Data can be recovered from cipher only by using exactly the same key used to encipher it.

FIPS PUB 46 recommends that certain kinds of data be protected by the DES:

Data that is considered sensitive by the responsible authority, data that has a high value, or data that represents a high value should be cryptographically protected if it is vulnerable to unauthorized disclosure or undetected modification during transmission or while in storage. A risk analysis should be performed under the direction of a responsible authority to determine potential threats.

Federal Standard 1027, *Telecommunications for Use in the DES, in the Physical Layer of Data Communications*, published by the General Services Administration in 1982, is a standard for how the DES algorithm should be built into cryptographic hardware or firmware—for example, in unclassified link encryption, voice encryption, and satellite systems. Equipment bought by the government for unclassified use must meet the 1027 standard.

The DES consists of two components: an algorithm and a key. The published DES algorithm involves a number of iterations of a simple transformation, which uses both transposition and substitution techniques applied alternately. This algorithm uses a single key to encode and decode messages. DES is a so-called private key cipher. As you may recall from earlier in this chapter, with this type of cipher, data is encrypted and decrypted with the same key. Both the sender and the receiver must keep the key a secret from others. Because the DES algorithm itself is publicly known, learning the encryption key would allow an encrypted message to be read by anyone.

The DES key is a sequence of eight bytes, each containing eight bits (seven key bits and a parity bit). During encryption, the DES algorithm divides a message into blocks of 64 bits (plaintext). It operates on a single block at a time, dividing the block in half and encrypting the characters one after another. The characters are scrambled 16 times, under control of the key, resulting in 64 bits of encrypted text (ciphertext). The key has 56 meaningful bits (the eight parity bits are discarded by the first permutation). Figure 7-11 (adapted from a diagram included in FIPS PUB 46-1) shows the basic processing performed during DES encryption. If you are interested in the details, you can consult that publication for an explanation of DES processing.

It must be noted that the government's own acceptance of DES was halfhearted at best. It never was allowed to be used in sensitive government communications, but only for sensitive but nonclassified materials. At the very time that politicians and government agency officials were making speeches about its strength (usually including references to its not being able to be cracked for multiples of the known age of the universe), a determined group of researchers working for the Electronic Freedom Foundation developed a specialized device named Deep Crack out of Field Programmable Gate Arrays (FPGAs) that demonstrated an uncanny knack of sawing through

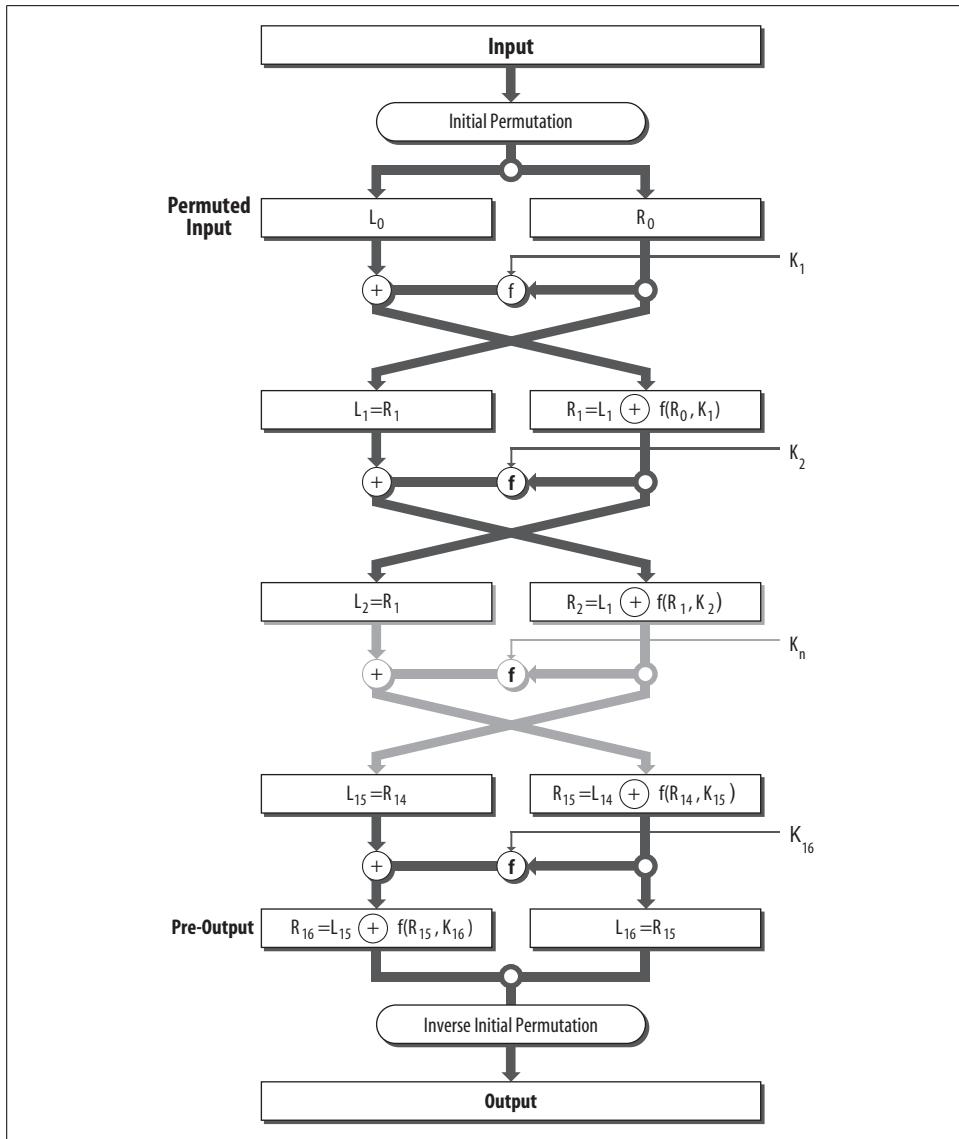


Figure 7-11. How DES works

supposedly uncrackable DES ciphertext. Also, in 1997, a cooperative effort on the Internet using over 14,000 computers, deciphered a test message after using 18 quadrillion of the 72 quadrillion possible DES keys.

There were three immediate effects. First the industry stopped listening for a time to almost anything the government said about privacy. Second, successive government initiatives to provide “free encryption for all,” such as the “clipper chip,” were laughed back into oblivion. Third, it was discovered that DES cracks did not really

matter. In most transactions, there is a time window in which information must be kept absolutely secret. After that window expires, disclosure doesn't much matter, especially in financial transactions. Whatever information was urgent will be old news, and its effect will already have been discussed in the newspapers.

Because DES has fallen, many organizations use triple DES (3DES), which is essentially DES repeated three times. This effectively increases the key length, boosting security, but no one is under the illusion 3DES will forever be adequately secure. Estimates of the length of key required to offer true privacy grow larger with each increase in computer power. In the early 2000s, math coprocessors began to appear in some personal computers again, along with dual-core architectures, drastically multiplying decryption power while lowering its costs.

DES lives on, curiously, because everybody can use it, and it can keep secrets for a little while, which is enough for many applications. For material that has to be kept absolutely confidential, or kept confidential over the long term, DES is not the best choice. Look to its replacement, the Advanced Encryption System (AES) or one of its siblings. (AES will be discussed later in this chapter.) For many applications, however, including telecommunications and mobile radios, DES will do fine.

Analyzing DES is highly instructive. Understanding it makes it easier to understand its successors. The DES provides four distinct modes of operation that differ in complexity and use. The following summarizes these modes very briefly. For a description of how the modes work, and an explanation of chaining and other technical terms, see FIPS PUB 81.

ECB

Electronic Codebook. A basic block encryption method. This mode operates like a codebook; for a given block of plaintext and a given key, it always produces the same block of ciphertext. The ECB mode is sometimes used to encrypt keys.

CBC

Cipher Block Chaining. An enhanced version of ECB that chains together blocks of ciphertext. Unlike the ECB mode, which encrypts identical input blocks to produce identical output blocks (perhaps revealing a pattern), the CBC mode encrypts each block using the plaintext, the key, and a third value, which is based on the previous block. This repetitive encryption, called chaining, hides repeated patterns.

CFB

Cipher Feedback. Uses previously generated ciphertext as input to the DES to generate pseudo-random output. This output is combined with plaintext to produce ciphertext, thus chaining together resulting ciphertext. The CFB mode is often used to encrypt individual characters.

OFB

Output Feedback. Very similar to the CFB. OFB does not chain the ciphertext as does CFB, although it does reprocess some plaintext. The OFB mode is often used to encrypt satellite communications.

The CBC and CFB modes perform message authentication as well as encryption. Message authentication ensures that the information received matches the information sent. During encryption by the DES, the blocks of text are linked; in CBC and CFB modes, the encryption of each block depends on the results of encoding the block that preceded it. Because of this link, the final encrypted block is changed if a single character is altered anywhere in the message. The final block serves as a message authentication code—a cryptographic checksum used to check the accuracy of the transmission and to detect whether there's been any tampering with the message.

See the discussion in the section “Message Authentication” later in this chapter.

Application of the DES

DES technology has been embedded in many commercial products and remains a popular choice for applications requiring fast and continuous processing, such as stream ciphers and has been used for devices such as police radios or secure telephones. One example of a DES-based product would be a mobile radio that uses a chip built on the DES algorithm to encrypt voice communications. Other DES products include encryption boxes for use with microwave, satellite, and other types of communications.

Although the government cannot use the DES to protect classified or extremely sensitive unclassified information, DES products have been very popular in all but the most secret government agencies. For example, DES is found in applications in the Department of Energy communications systems. The DES was also the basis of the Department of the Treasury's electronic funds transfer program, and the Federal Reserve used DES to encrypt connections between Depository Financial Institutions and Federal Reserve banks. Off the shelf military communications systems are often built on DES.

Historically, NSA has supported the DES through the Government Endorsed Data Encryption Standard Equipment Program. Through this program, NSA evaluates DES-based products, places successfully evaluated products on the NSA Endorsed Data Encryption Standard Products List (NEDESPL), and publishes this list in the *Information Systems Security Services and Products Catalogue*, available from the Government Printing Office. In some cases, this role is now delegated to approved independent laboratories.

In 1986, the agency surprised the industry by announcing that as of January 1988, it would not endorse DES-based products as complying with Federal Standard 1027. In addition, the agency said that it would recommend that NIST not reaffirm the DES when the standard next came up for review (in 1988). NSA did say that products already endorsed would continue to be available and would continue to be listed, and that NSA would continue to provide keys as needed for these products for a time. (NSA historically provided keys to owners of DES equipment.) In addition, the

agency would continue to evaluate modifications to previously endorsed products, as long as the modifications did not affect the security of the products.

What's the alternative to the DES? NSA for a time tried to get both government and industry to use its own classified algorithms. Through the Commercial Communications Security Endorsement Program (CCEP), described in the later section "Government Cryptographic Programs," qualified vendors may be allowed access to these algorithms, embedded in tamperproof integrated circuit modules, that could be used in in their products.

There was a sizable reaction—much of it negative—to NSA's decision to drop DES endorsement. In 1987, in the wake of NSA's announcement, when NIST published a request for comments, 31 of 33 responding organizations (including federal agencies, the American Bankers Association, and the Computer and Business Equipment Manufacturers Association) supported reaffirmation of the standard. (Of the other two, one did not oppose reaffirmation, and the other supported reaffirmation for financial encryption.*) Respondents pointed out that the DES was used extensively in existing products and applications, and that there was no clear and currently available alternative. Without the DES, information might be left unprotected while organizations waited for an appropriate alternative algorithm.

Vendors had other concerns about losing the DES. They were worried about the future market for existing DES-based products. There was also concern about using government-supplied or -mandated encryption chips, for fear that this would somehow help government agencies such as NSA to more easily penetrate messages sent using the vendors' equipment, decreasing its attractiveness to buyers. (Revelations about a global monitoring network called Echelon eventually made it plain that this concern may have been valid.) Finally, because the classified algorithms have more stringent export restrictions than DES-based products, companies that use these products in their U.S. and foreign offices (and in communications between these offices) could be seriously affected. There was a time in which versions of software with built-in encryption were not approved for export.

Industry complaints, many coordinated by the American Bankers Association on behalf of banks and financial institutions (prime users of the DES for encryption of financial transactions), led NSA to reconsider its decision. NSA eventually announced that it would continue to support the DES for the encryption of financial data—for example, for FedWire transactions—"until transition to a new cryptographic technology is possible." The government also eased export restrictions in the late 1990s, to all but a few nations considered hostile to the United States. This was very helpful to businesses with worldwide operations.

* Miles E. Smid and Dennis K. Branstad, *The Data Encryption Standard: Past and Future*, National Institute of Standards and Technology, Gaithersburg (MD).

Overwhelming support for the DES, caused NIST to recommend to the Secretary of Commerce that the DES be reaffirmed for five more years, until 1992, and the cycle for creating a new encryption algorithm was begun in earnest within a few years after that.

Realizing the DES is no longer secure for many applications, and suspicious of how long 3DES may last, those who are concerned about real security cast about for other algorithms that may allow them to continue in reasonable confidentiality. The government, through NIST, was heavily involved in the selection process.

The Advanced Encryption Standard

The government, via NIST, determined that the next nationwide algorithm should be called the Advanced Encryption Algorithm, to be used in the Advanced Encryption Standard (AES). Several rounds of competition were held to determine the algorithm that should be the basis of AES. From five semifinalists, the winning algorithm was selected to be the Rijndael algorithm, created by Joan Daemen and Vincent Rijmen.

NIST announced the approval of the Federal Information Processing Standard (FIPS) for the Advanced Encryption Standard, in FIPS-197. This standard specifies the Rijndael algorithm as a FIPS-approved symmetric encryption algorithm that may be used by U.S. government organizations (and others) to protect sensitive information. Because of the difficulty of proving that an algorithm is truly secure, many nations and entities besides the United States have joined in supporting the AES. The more people there are to test an algorithm, and problems are not found, the more likely it is that people will gain confidence that it is secure.

Overview of the AES Development Effort

NIST worked for years with the cryptographic community to develop an Advanced Encryption Standard. The overall goal was to develop a FIPS that specifies an encryption algorithm(s) capable of protecting sensitive government information for a decade or more (into the early decades of this century). The algorithm or algorithms were expected to be used by the U.S. Government and, on a voluntary basis, by the private sector.

NIST announced the initiation of the AES development effort in January 1997, with a formal call for algorithms in September 1997. NIST listed the following requirements:

- The design for AES would be that it would be unclassified, that is, the algorithm would be subject to public scrutiny.
- AES was to use a symmetric key algorithm.
- The algorithm was to be made freely available, without charging royalties, worldwide.
- Additionally, the algorithm was to be a block cipher, and at a minimum support block sizes of 128 bits and key sizes of 128, 192, and 256 bits.

In August 1998, NIST announced that 15 algorithms, submitted by cryptographers worldwide, had been chosen as the first candidate group. NIST requested public comments on the candidate algorithms. Candidates met to review and discuss the various algorithms and the analysis in March 1999, two weeks after the period for comments closed. Based on the review, NIST selected 5 finalists from the 15 algorithms. The AES finalist candidate algorithms were MARS, RC6, Rijndael, Serpent, and Twofish. NIST then solicited public comments on the remaining algorithms, including but not limited to cryptanalysis, reviews of the intellectual property involved, and overall recommendations and implementation issues. NIST also provided a discussion forum for interested parties to discuss the AES finalists and relevant AES issues.

In April 2000, NIST sponsored a third meeting—an open, public forum for discussion of the analyses of the AES finalists. Submitters of the AES finalist algorithms were invited to attend and engage in discussions regarding comments on their algorithms, deliver papers, and make presentations. After studying all available information, NIST announced that it had selected Rijndael as the proposed AES algorithm. After another comment period, a draft FIP was published for public comment in February 2001. After a review, the standard was at last implemented in the summer of 2001.

How AES Works

Rijndael works on a byte-by-byte basis, matching up the input cleartext to the cipher key in a matrix of 16 bytes (4×4 matrix). The key is divided, or scheduled, so as to be injected in several repetitive rounds a little bit at a time. The first part of the key is added in before beginning 10 processing rounds. In each of these rounds, bytes are substituted, rows are shifted, and columns are mixed. (See Figure 7-12.)

SubBytes

In the substitution process, the bytes of the input text are placed into a substitution box (S-box). This is a 16×16 matrix with a byte stored at the intersection of each row and column. The correct substitution is achieved by taking the first nibble (hex integer) of a cleartext byte and using it to determine the row of the S-box, and then the second nibble to determine the column. The character stored at the intersection of the selected row and column becomes the substitute byte (SubByte) for the cleartext. This process is repeated for each of the 16 bytes in the matrix.

Row shift and mix columns

The bytes to be encoded, now swapped with replacements, are next subjected to a row shift. Imagine the first row being undisturbed, the second row moved right one space, with the byte that falls out rolling over and assuming the empty spot at the start of the row. The third row shifts two bytes, and the final row three. This is followed by a Mix Columns phase, in which each column of the matrix is multiplied by another matrix to change its position.

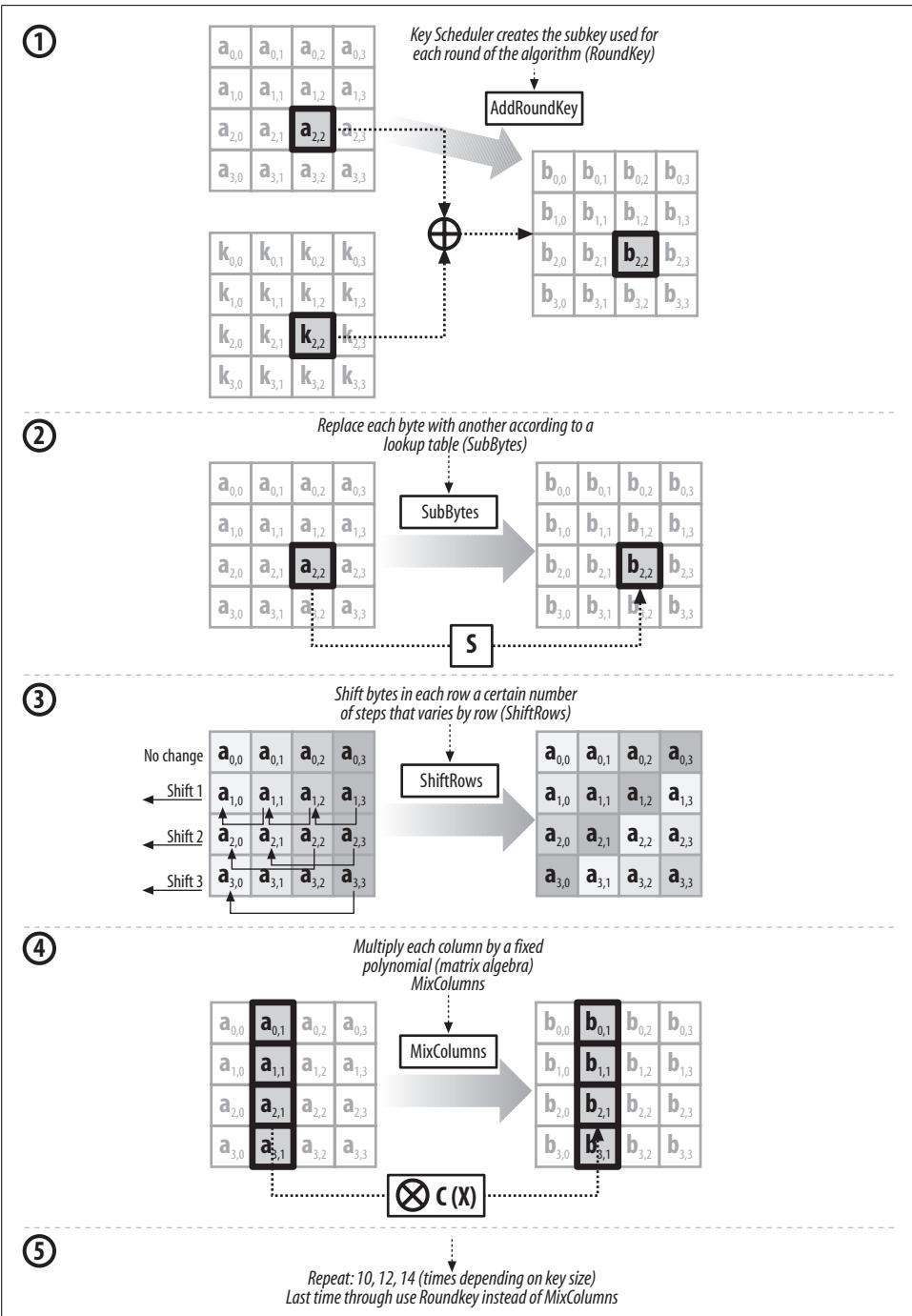


Figure 7-12. Operation of AES

Round keys

Next, a round key is added to each column. Think of the round key as a little bit of the secret key, injected so as to result in further encryption.

Do it again

These transforms are applied for nine more rounds. There is no operation to mix columns on the final round. A round key is added to the final result, yielding the ciphertext. The key is shifted and rounded and added to itself as well.

The AES algorithm resulted from a multiyear evaluation process led by NIST with submissions and review by an international community of cryptography experts. AES is gradually showing up in many different encryption protocols including IEEE 802.11, IPsec, S/MIME, and TLS. RSA Laboratories—one of the five finalists in the AES competition—recently submitted a proposal for using AES to encrypt wireless data packets, a system called AES-CCM.

A great deal of information about the AES algorithm is available at the NIST web site: <http://csrc.nist.gov/encryption/aes/>.

Other Cryptographic Algorithms

Not all of the encryption algorithms in existence were presented for consideration for the AES. Some were reserved for copyright reasons. And there were also algorithms that were submitted but not selected for the original 15. These algorithms have not disappeared. It is likely that they will be encountered from time to time. Among them are:

- DESX
- 3DES
- BLOWFISH
- CRYPTON
- DEAL
- GOST
- CAST
- RC2
- RC4_
- DIAMOND2
- TEA
- SAFER
- 3WAY
- SHARK

- SQUARE
- SKIPJACK
- REDOC II
- IDEA
- TWOFISH
- SERPENT
- E2
- Frog
- HPC (The Hasty Pudding Cipher)
- LOKI97

You can obtain more information about these algorithms at the following web sites:

- <http://csrc.nist.gov/cryptval/des.htm>
- <http://www.counterpane.com/>
- <http://cryptography.org>
- <http://citeseer.nj.nec.com/253207.html>
- <http://www.tropsoft.com/strongenc/serpent.htm>

AES Round 1 Candidate Algorithms

The algorithm which were considered for the first round of the AES selection process, as well as some of their proponents are listed next:

- CAST-256: Entrust Technologies, Inc. (represented by Carlisle Adams)
- CRYPTON: Future Systems, Inc. (represented by Chae Hoon Lim)
- DEAL: Richard Outerbridge, Lars Knudsen
- DFC
- CNRS: Centre National pour la Recherche Scientifique, Ecole Normale Supérieure (represented by Serge Vaudenay)
- E2 NTT: Nippon Telegraph and Telephone Corporation (represented by Masayuki Kanda)
- FROG
- MAGENTA
- DFC
- CNRS
- HPC
- LOKI97

Public Key Algorithms

The DES and AES are examples of private key algorithms. The keys are kept secret, and the key is the same on both ends of the link. A completely different type of cryptographic algorithm, the public key algorithm, was introduced by Whitfield Diffie and M.E. Hellman in 1976.* (See the section “Cryptographic Keys: Private and Public” earlier in this chapter for an introduction.) Two examples of public key cryptography (PKC) are the Merkle and Hellman trap-door knapsack encryption method† (which was broken in 1982) and the RSA algorithm introduced by Ronald Rivest, Adi Shamir, and Leonard Adleman.‡

The very first public key algorithm, proposed by Diffie and Hellman, cannot be used for encrypting messages or files by itself, but can be used to exchange keys used with other cryptosystems, and also for identification and related purposes. Drs. W. Diffie and M.E. Hellman disclosed this method in a 1976 paper entitled “New Directions in Cryptography.” The security of the Diffie-Hellman public key system is based on the mathematical difficulty of the discrete logarithm problem. For this scheme to be secure, the keys have to be long.

One particular implementation of the Diffie-Hellman algorithm, included by Sun Microsystems as part of its widely used Network File System (NFS) protocol, used keys of 192 bits. In 1989, at AT&T’s Bell Laboratories, Andrew Odlyzko and Brian LaMacchia broke this particular discrete logarithm problem. With their solution, it’s possible to calculate the secret key using only a few minutes of computer time (although the program that implements the calculation took months of time by cryptography experts to develop). One important result of the work of Odlyzko and LaMacchia is that for secure operations, it shows that you have to use public key algorithms with long keys. This increases the computational burden on the machines used to encode and decode the messages.

The RSA Algorithm

Typically, private key algorithms such as DES can’t protect against fraud by the sender or the receiver of a message. (The reason being that the key used to encrypt the DES message could have been stolen and used to encrypt a message by a false entity.) The RSA algorithm, on the other hand (as well as some other public key algorithms), provides authentication (a way of ensuring that a message was sent by a certain user), as well as encryption. Named for its developers, Rivest, Shamir, and

* W. Diffie and M.E. Hellman, “New Directions in Cryptography,” *IEEE Transactions on Information Theory*, Volume IT-22, Number 6, November 1976.

† Merkle and M.E. Hellman, “Hidden Information and Signatures in Trapdoor Knapsacks,” *IEEE Transactions in Information Theory*, September 1978.

‡ Rivest, A. Shamir, and L. Adleman, “A Method for Obtaining Digital Signatures and Public Key Cryptosystems,” *Communications of the ACM*, Volume 21, Number 2, February 1978.

Adleman, who invented the algorithm at MIT in 1978, the algorithm uses two keys: a private key and a public key. With RSA, there is no distinction between the function of a user's public and private keys. A key can be used as either the public or the private key.

The keys for the RSA algorithm are generated mathematically—in part, by combining prime numbers. Most big numbers can be built by multiplying smaller ones together. Once you do this, the smaller numbers will be mathematically related, and knowing one can help you determine the other. This is the situation you are looking for with private/public key pairs. But once the numbers get really huge, knowing one of the factors does not mean it is easy to find its companion. There are many false choices, and only the right one will help decrypt the message. The security of the RSA algorithm, and others like it, depends on the use of very large numbers. (Most versions of the RSA use 154-digit or 512-bit keys.) It is hard, sometimes impossible, to reliably factor extremely large numbers, such as those that are hundreds of digits long. Some research efforts have succeeded in cracking very large numbers, but it's still highly unlikely that an intruder will choose number-cracking as a cost-effective way to break into a system.

Rivest, Shamir, and Adleman licensed the patent on the algorithm from MIT, and in 1982 began offering the algorithm as a commercial product. A number of government and corporate users now use the RSA algorithm; these include Lotus Development (in its Notes groupware product), the U.S. Navy, the Department of Labor, and a number of other federal agencies and universities. Although the RSA algorithm is patented inside the United States, it cannot be patented abroad (because the algorithm was published before it was patented), so it is in fairly wide use outside this country.

Digital Signatures and Certificates

In addition to providing encryption and message authentication, some encryption systems also use an authentication tool called a *digital signature* to verify the origin of the message and the identity of the sender and to resolve any authentication issues between sender and receiver. A digital signature is distinct for each specific transaction. It is unforgeable and can potentially be used as a valid signature in legal contracts. Public key encryption systems such as the RSA can produce digital signatures quite readily. When a message is encrypted at the sender's end, the sender's key digitally signs the message. When the message is decrypted at the recipient's end, the receiver's key is used to validate the digital signature. If any alteration in either signature or message occurs, the signature won't verify any more.

An algorithm that provides both encryption and a digital signature might work like this. Suppose Joe is sending a message to Claudia:

1. Joe encrypts the message with his private key (to sign it).
2. Joe now applies Claudia's public key to the message (to keep it a secret from anyone but Claudia).

Now, suppose Claudia has received a message, supposedly from Joe:

1. Claudia decrypts the message with her private key (to validate the signature that is part of the message).
2. Claudia now applies Joe's public key to the message to verify that he sent the message. (It won't validate unless Joe's private key was used to encode it.)

There has been discussion about expanding the use of digital signatures so they can be used as digital pseudonyms. This would be a way of preventing attackers (people trying to intercept a message) from figuring out a sender's identity via cross-matching or other techniques.

Certificates

In cryptography, a public key certificate (or identity certificate) is a short document that can be used to verify that a public key belongs to an individual. The certificate uses a digital signature to bind together a public key with information such as the name of a person, an organization, or address information.

A certificate usually follows the ITU-T X.509 standard, which includes the following:

- The public key being signed.
- A name, which can refer to a person, a computer, or an organization.
- A validity period during which the certificate should be considered to be reliable.
- The Internet address (URL) of a revocation center that can be consulted to determine if the certificate has been declared to be invalid.

Certificates make it possible to use public-key cryptography on a large scale. To securely exchange secret keys between network users becomes impractical as the number of users increases beyond a few (the key distribution problem). The system used to exchange keys as networks scale in size is called the Public Key Infrastructure (PKI). If user Beatrice wants others to be able to send her encrypted messages, she publishes her public key. Anyone who obtains it can then send her a secure message.

Certificate Authorities

Unfortunately, user Mark can outwit Beatrice by publishing a public key for which he, not Beatrice, has the associated private key. Mark then claims that the public key belongs to Beatrice and intercepts some of the messages intended for her. Beatrice can counter this, however, by building her public key into a certificate requesting that a third party digitally sign it. Anybody who trusts the third party can then check the certificate to see whether the embedded public key is truly Beatrice's.

In this PKI, the third party is a Certificate Authority (CA). The CA is must be trusted by all participants, that is, each user must decide whether to trust the CA when the CA asserts that a particular public key belongs to a particular user.

There are many commercial CAs that charge for their services. Institutions and governments may have their own CAs. You can find free CAs on the Internet. In a large network, users may not be familiar with each other's certificate authority. In this case, appeal is made to a higher level Certificate Authority, sometimes called CA2. This can be accomplished by including the public key of the local CA, signed by a higher level CA, which is likely to be recognized by the message recipient. This leads to a hierarchy of certificates, with the root certificate at the top. The root certificate represents a CA that is so central and well known that no additional third-party authentication is needed.

If it is discovered or suspected that a given private key has been compromised, or if the relationship between a certificate and a private key is no longer valid (by a person changing jobs, for instance), it is possible to revoke a certificate. This is done by placing it onto a certificate revocation list (CRL), which is stored at the revocation center, and which should be scanned when two computers need to trust each other.

A final note about certificates is that they are built on varying levels of trust. The level of trust dictates how sure you can be that the individual who owns the digital certificate is real, or is who she claims to be. Validation checking can encompass checking the corporate registry and legal status of a company, verifying whether or not they truly own the domain for which they wish to obtain a certificate. You should also check the ID of the person making the request; make sure he is an authorized representative of the requesting organization.

Government Algorithms

NSA has always had secret algorithms that it guards very closely. Some of these are now available to members of the Commercial Communications Security Endorsement Program (CCEP) for inclusion in so-called "high-grade" cryptographic products. Founded in 1984, CCEP is a business relationship designed to combine government cryptographic knowledge with corporate product development expertise. Through CCEP, several dozen companies have developed cryptographic products.

NSA and NIST have also joined forces to create a joint committee that's developing a set of new algorithms for use with sensitive, unclassified information. Several algorithms are known to be under consideration:

- A DES-like algorithm used to protect confidentiality.
- A public key algorithm used to distribute keys for the confidentiality algorithm.
- An algorithm used to provide a digital signature for messages. This algorithm would perform a one-way hash of the message.
- A second algorithm used to provide a digital signature for messages. This algorithm would sign the hash in digital fashion.

Message Authentication

As mentioned previously, encrypting a message makes the message a secret, but it doesn't automatically authenticate it. Encryption protects a message from disclosure. Only authentication protects a message from modification.

What does authentication do for a message? It ensures that the message has not been altered, either maliciously or inadvertently, during transmission. It has arrived exactly as it was sent. Authentication also ensures that the message is not a repeat (called a *replay*) of a message previously sent, that the message came from the origin stated in the message (was not forged by an imposter), and that the message went to the intended recipient (was not falsified—for example, to alter the date of receipt).

Authentication can be used alone—just to ensure against modification and forgery—or it can be used in conjunction with encryption. In the first case, the plaintext is authenticated. In the second case, the ciphertext is authenticated.

Historically, computer and communications systems have used such techniques as *checksums* (adding up the digits in a message before and after transmission to see if they have changed), *parity checks* (counting the ones and zeroes in a message to determine if the number of ones is odd or even, and verifying that whatever the state was before transmission, that it is the same after), and *test words* (special codes or words inserted in or preceding a message to signify authenticity) to check that the information received matched the information sent, and that nothing had been modified—either intentionally or unintentionally. The message authentication capabilities included in modern encryption technologies provide an extremely reliable replacement for the old checking techniques. Both private-key and public-key encryption algorithms allow for message authentication, though public key systems are better equipped to perform this function due to their two-key mechanism.

With the DES algorithm, for instance, certain modes (CBC or CFB) perform message authentication. Encrypting the data also produces a message authentication code, which is appended to the encrypted message. At the receiving end, the DES independently calculates the code for the message and compares it to the message authentication code sent with the message. If the two codes are identical, it's extremely likely that the message was sent without alteration.

There are a number of elaborations on this theme. To provide privacy as well as authentication, the DES can be used with two different secret keys, one for authentication and one for encryption. To safeguard against the sender or the receiver of a message forging or denying that he sent the message, you can use a digital signature (described earlier). Although it's possible to create a digital signature via the DES (using a message authentication code), public-key systems such as the RSA algorithm often do it more efficiently.

Government Cryptographic Programs

The government has a keen interest in cryptographic products. NSA, NIST, and the Department of the Treasury have all developed programs for evaluating cryptographic algorithms and products.

NSA

NSA's Communications Security Cryptographic Endorsement Program, introduced earlier in this chapter, evaluates so-called "high-grade" cryptographic products. All algorithms used in high-grade products are designed by NSA and are classified. Chip implementations of the algorithms are provided to vendors with protective coating so they can't be reverse-engineered.

NSA classifies high-grade cryptographic products developed under CCEP as either Type 1 or Type 2:

Type 1

Type 1 products are designed to encrypt classified data; they can also be used to encrypt sensitive unclassified data. An example of a Type 1 product is the Secure Telephone Unit (STU), a telephone that encrypts voice and data communications and provides secure key distribution. STU-II and STU-III are NSA-sponsored projects aimed at developing secure telephones for government agencies and government contractors. The telephones operate over ordinary telephone circuits and use encryption to provide secure voice and data communication. The first versions are being developed by AT&T, General Electric, and Motorola. DoD has recently bought thousands of STU-III units for use by both government employees and contractors. Other Type 1 products include trunk encryption devices and network communications products.

Type 2

Type 2 products are designed to encrypt sensitive unclassified data; the government doesn't allow these products to be used to encrypt classified data. Examples of Type 2 products include authentication devices, transmission security devices, and secure LANs. Type 2 equipment is effectively intended as a replacement for DES-based equipment.

Until recently, NSA's Government Endorsed Data Encryption Standard Equipment Program evaluated products based on the DES algorithm. Although NSA no longer endorses new DES-based products through this program, it does continue to list and provide keys, as necessary, for already endorsed products.

NIST

NIST's cryptographic responsibilities include the development of both standards and validation systems. NIST assists the Department of the Treasury by offering a system

that tests the conformance of vendors' systems to the ANSI X9.9 message authentication standard. The system also checks for conformance to FIPS 113 (*Computer Data Authentication*). The validation is automated and can be initiated remotely via telephone lines. NIST is currently developing a system that tests the conformance of systems to the ANSI X9.17 key management standard. NIST is also working on systems that use digital message authentication codes in place of written signatures in government transactions.

Treasury

Since 1988, the Department of the Treasury has required that all of the department's electronic funds transfer messages be authenticated. The Treasury certifies authentication devices developed by vendors to ensure that they conform to Federal Standard 1027 (DES implementation) as well as to ANSI standard X9.17 (key management). The Electronic Funds Transfer Certification Program for Authentication Devices is aided by technical input and testing services provided by NSA and NIST.

Cryptographic Export Restrictions

The U.S. government closely regulates the sale and export of cryptographic products developed within the United States. Export regulations are intended to restrict the use of products that ultimately could make an enemy nation's communications more difficult for U.S. intelligence agencies to decipher.

Most cryptographic export restrictions were eased under the Clinton administration, nevertheless, it is imperative that you verify that there are no restrictions in place against each nation with which you use cryptography.

In general, licenses are issued only for the governments and government contractors of the NATO countries, plus certain "friendly" governments, such as Canada, Australia, and New Zealand. License applications are considered on a case-by-case basis. NSA sometimes requires a vendor to change its own encryption algorithm to qualify for an export license. Licenses are easier to get for internationally based financial institutions with recognized needs for encryption.

Often, once overseas customers have acquired a product stripped of its cryptographic capabilities, they'll insert a different, home-grown encryption algorithm in the product.

By the same token, there are countries whose governments do not wish their populace to have access to powerful encryption. This was in fact the case here in the United States, where the developer of the Pretty Good Privacy encryption system found himself in no end of trouble for a few years for developing a powerful and efficient method of encrypting documents and email.

Generally speaking, encryption control falls into the categories of input, export, and domestic use, with some countries denying all three. A useful table is maintained by RSA Laboratories at the following web address: <http://www.rsasecurity.com/rsalabs/node.asp?id=2333>.

Summary

From the secret writing systems of early times to the secure encryption algorithms of today, encryption has been part of the arts of war and diplomacy. Commercial security has now increased in importance to the point that encryption is a common tool in daily business. A requirement for privacy—say for patient records and other sensitive data—has moved encryption into the medical market. Even educational records deserve a certain amount of encryption to protect family privacy.

Modern cryptology depends upon thoroughly vetted encryption algorithms, coupled with the secure use and transmission of keys, either public or private. This is the basis of maintaining message confidentiality, integrity, and authenticity.

Communications and Network Security

The cost of maintaining secure communications links can be great. In a time when remote access to the corporate IT infrastructure is critical, imagine the expense of securing the link to every traveling salesperson or home-based telecommuter. Encryption is an obvious answer, but as we saw in the previous chapter, to provide encryption and do it right is burdensome. Further, the Internet is a hostile environment for the security conscious. Crackers and other attackers seek to co-opt communications both to eavesdrop and to steal.

And the problem is coming home. In the “always on” mode that tends to follow broadband communications such as DSL and cable modems, attackers can attempt to acquire sensitive data, steal identities, and take over hardware by planting malicious code, all while the computer owner is sleeping.

Still, the Internet is such an effective way to put the people of an organization near where the action is and where the customers are, that rolling back to an earlier era seems unlikely. If we are to effectively leverage the Internet, there must be a way to use it without becoming a victim to those who seek to misuse it. Bad guys will attempt to intercept our communications, or worse.

The modern approach to network security, particularly for remote access, but also for point-to-point links, is called the Virtual Private Network. It is “private” because while it appears to travel over the same wide area network as other traffic, its internal encryption mechanisms serve to keep the transmitted data safe from prying eyes. It is “virtual” because there is no need to invest in the infrastructure of buildings, wires, cable plant, and technical services, other than the nominal (by comparison) fee charged by the service provider. More VPN information is available in the section “Through the Tunnel.”

Whether you work on a lone laptop or a clustered supercomputer, you undoubtedly need access to a network and to the Internet. Networks let you share information (such as messages and files), as well as resources (such as printers and other remote computers). This type of communication is vital to most organizations. Information

that can't be shared in a timely fashion among users—in different parts of an office, a country, or the world—may rapidly lose its value. But sharing information over communications lines creates increased dangers of interception. From a computer security point of view, networks are the most vulnerable component of a system configuration. The number of possible users, the ease of access from remote, and sometimes anonymous, locations, and the opportunity for error introduced by the global complexity of the Internet, all contribute to this vulnerability. For this reason, computer and network security have increased steadily in their importance to individuals, commercial users, and to government and military organizations.

What Makes Communication Secure?

Communications security protects information while it's being transmitted over unsecured networks, which would be any network that you do not control, whether owned by you or by a service provider. Secure communication enforces the security principles introduced in earlier chapters.

Secrecy or confidentiality

Secure communication keeps information from being transmitted to anyone not authorized to receive it. Secrecy means that intruders can't tap into communications lines and messages will go to their intended recipients, without being viewed by others. In highly secure systems, secrecy might also mean that information can't be passed—either deliberately or because of delivery errors—to systems or networks not cleared to process that level of information.

Accuracy or integrity

Secure communication keeps information from being lost, changed, or repeated during transmission. The information is delivered exactly as sent.

Authenticity

Secure communication keeps users on either side of a transmission from being able to forge a message.

Nonrepudiation

Secure communication means that a receiver is certain that the message sender is who it is expected to be. Security also means that a message sender will be unable to later on deny having sent a given message.

Availability

Secure communication keeps the network working efficiently. Availability is a particularly important concept for networks, where even a minor slowdown in service can have a reverberating effect on an entire network. “Denial of service,” described later in this chapter, is a particularly virulent problem for network security. Think of availability as having to do with robustness, and the ability to resist attacks which could slow down or compromise the communications process.

How can you make a communication secure? Here are the main approaches:

Keep the communication from being intercepted

Protect your communications equipment (so an intruder can't attack a network switch or any other equipment), and choose the most secure communications medium (so an intruder can't tap the line).

Encrypt the data you're sending

If you're concerned primarily about protecting the secrecy of a communication, encryption (described in Chapter 7) is an excellent solution, regardless of whether the medium and the equipment are physically secure. Certain encryption techniques can also ensure the accuracy and authenticity of a communication. A powerful method of automatically encrypting data during communications is the Virtual Private Network. Using VPNs, workers at home can have secure access to the same resources they would be able to access at the corporate site. Corporate VPNs are used in point-to-point links.

Control access to your systems and to your networks

Proper authentication of users helps ensure that only authorized users can enter the network and that once in, they can access only approved resources. Configure your network for security. For complete protection from the outside, set up an *airwall*: no external connections, no floppy drives, no USB-port memory sticks. Just the computer on your network, nothing more. To protect your own trusted local area network without completely cutting off communication to the outside world, set up a gateway computer sometimes called a *firewall* to isolate your local users. A firewall looks at each transmission and decides if it originates from or is destined to an Internet address that is listed as desirable to receive (i.e., a friend or a foe).

Within the security perimeter of a local network, users may be able to communicate freely

It is possible that an eavesdropping attack could take place within the secured area. An intrusion detection system (IDS) can monitor the internal traffic and look for packets that seem malicious or are sourced from known trouble spots.

Communications Vulnerabilities

There are many points of vulnerability when information is being communicated.

The media itself (e.g., the telephone line, the cabling, or the radio transmission) may be vulnerable. Different types of media vary significantly in how easy they are to tap. Simple cordless telephones can be intercepted very easily, and conventional telephone taps are both easy and inexpensive to implement. Cellular phones vary in complexity. Coaxial cable is somewhat more difficult to tap. Fiber-optic cable is probably the most secure medium. The IEEE 802.11x wireless standard (Wi-Fi) has gone through several iterations of security in order to keep it safe from prying eyes, although in the eyes of some, the effort has not succeeded. The 802.11i standard addresses security directly. (See the section "Network Media" later in this chapter.)

Appropriate Technology

I'll discuss some of the most important methods for message security in this chapter, but there are other approaches as well. Consider one of the methods used to protect radio communications in the Pacific during World War II, as reported by David Kahn in *The Codebreakers* (Scribner). The Marines protected against eavesdroppers by having native Navahos on either side of the communication speak to each other in their own language. Because so few people speak Navaho, and because Navaho is an extremely difficult language to learn, particularly as an adult, there was virtually no chance that communications between "codetalkers" could be understood or counterfeited by the enemy. As with operating-system security, you'll have to decide on the best method, or combination of methods, for your own equipment, environment, and assessment of the importance of your information.

Communications equipment (e.g., switching systems, routers, signaling equipment, or testing equipment) is another point of vulnerability. The nodes where communications lines meet are very vulnerable to attack. It's extremely important to keep unauthorized people away from all communications equipment, starting with card-key access to the premises and continuing on through putting a lock on every telecommunications room door. Damage to a switching system can have a disastrous effect on a network. Communications equipment is also vulnerable to natural disasters (e.g., power problems) and to error (e.g., noise on the line).

Telephone and network connections are very vulnerable to attack. It's often easier to break into a system over a network than it is onsite. Physical controls obviously aren't effective against remote access. Using telephone and network connections, an attacker can spend a lot more time trying to break into a system remotely than would be possible onsite—and can usually stay anonymous as well.

Communications Threats

There are a number of special terms commonly used to describe communications threats.

Masquerade

Occurs when someone (an imposter) pretends to be an authorized user.

Playback or replay

Occurs when someone records a legitimate message (perhaps a funds transfer), and later sends it again.

Repudiation

Occurs when someone denies that she sent or received a message.

Denial of service

Occurs when someone or something dominates system resources, stopping or slowing down system or network performance.

Distributed denial of service

A coordinated denial of service attack that originates from many sources at once.

Denial of service is a problem for operating systems as well as networks. If someone shuts off power, fills up a storage device or disk, or creates more processes than the system can support, no one will be able to get any work done. If server resources aren't available on an equitable basis, some users will be very unhappy. One example of a network denial of service attack is *message flooding*, in which someone sends so many requests (perhaps meaningless messages) to a system that the system's resources are overloaded, and the system may crash as a result.

Communications interceptions, or taps, are another special network concern. There are two basic types of taps:

Passive taps

Threaten the secrecy of the information that's being transmitted. Taps of this kind usually involve wiretapping or radio interception of transmitted data. Through electronic eavesdropping or monitoring, the intruder intercepts the information but doesn't attempt to modify it. It's very easy to tap telephone lines and lines connecting terminals, especially in telecommunications rooms where connections are plentiful, and splice points abound. A single splice, or an induction loop around a terminal wire, can successfully intercept many different types of communications.

Active taps

Threaten the authenticity of the information that's being transmitted. Active taps usually involve breaking into a communications line and deliberately modifying information. In addition to tampering directly with the contents of the information, the intruder might threaten the transmission by tampering with its routing or authenticity—by changing the apparent origin of a message, by rerouting it to another destination, by replaying a previous message (to create a false message), or by falsifying an acknowledgment of a genuine message.

Modems

The simplest type of communication involves a single user communicating with a computer via a modem. A *modem* (short for modulator/demodulator) is a device that lets you connect to a computer from a terminal using an ordinary telephone line. Your computer can also use a modem to communicate with another computer. Many point-of-sale terminals, credit-card scanners, and automatic teller machines (ATMs) work using modems, which convert digital bits to tones that can be transmitted over the telephone line. Using a modem is efficient because the technology is

low cost and the Public Switched Telephone Network is readily accessible. Modems introduce security risks because they allow anyone to call your computer or modem-connected device. Attackers can track down your computer by dialing endless numbers using software that operates automatically, trying every number in turn until it finds one that answers with a modem's characteristic noise. Once connected, the intruder still needs to crack your login defenses, but connecting to the computer is the first step.

In most modern configurations in which modems are used, remote access servers (RAS) handle most security chores. These servers, using protocols such as TACACS and RADIUS, and DIAMETER employ techniques such as automatic callback, in which a user dials into the remote access server to notify the server that the user wishes to connect, and then hangs up. The server then calls the user. This not only ensures security, it also makes the telephone connect charges the responsibility of the server end, where bulk calling rates are more likely to apply. Once the RAS server authenticates the caller, the user then can execute network transactions according to whatever privileges have been stored in the user's security profile.

In most parts of the world, modem access is being replaced by wireless access or by some form of broadband access. With a broadband system, the security procedures employed resemble network security in general. This is because most high-speed access networks use the same protocols as local area networks, such as TCP/IP. An authentication server may still be employed to handle the chore of verifying users' identities—and keeping visitors isolated until it is certain they are supposed to be present.

Hints for Modem Security

- Be sure unauthorized users can't easily get access to your telephone and modem.
- Don't publicize your computer's telephone number.
- Don't put call forwarding on your telephone line if you have a callback modem on your computer. Call forwarding can let someone who's learned a password forward the call from the authorized terminal to his own (unauthorized) terminal.
- Be sure your modem works properly with the systems you're accessing. For example, make sure the modem hangs up the telephone when you log out. Make sure the modem hangs up the telephone, and the system logs you out if you get disconnected. Otherwise, someone else might get access to your account—either accidentally or deliberately.

Networks

This section defines the most important terms you'll encounter when reading about networks, and it provides a brief history of where networks came from. Because the goal of this chapter is to summarize a number of network issues relevant to security, not to provide a comprehensive description of networking as such, we're only touching on the major networking topics, without trying to be too rigorous or complete. There are a number of good books describing network concepts in much greater detail.

Network Terms

A *network* is a data communications system that allows a number of systems and devices to communicate with each other. Networks allow users to send and receive messages, and to access network services such as shared information and devices. A *message* is a generic name for a single unit of communication that's transmitted over a network. A message might actually be an electronic mail message, a file, a document, an image, or any other integral piece of information.

A PC or other system that is capable of processing information can be called a *network node*. A computer system that is accessed by a user working from a remote location is called a *host*, while the device by which a user connects to the host is called a *terminal* or *remote terminal*. In modern practice, most terminals are actually PCs that are running terminal emulation software. A PC that is connected to a network may thus be called either a host or a node, depending on the context. Generally speaking, a host is a PC or server, while a node is more of a generic term for any device or appliance that connects to the network, PCs and servers included.

At a very low level of message communication and routing, we can discuss network communications in two categories: connection-oriented and connectionless.

Connection-oriented communications are often compared to telephone communications. With a telephone call, you pick up the telephone, dial the number, establish that the person you want to talk to is there, carry on your conversation, say goodbye, and hang up. For the duration of your conversation, a dedicated connection called a *circuit* is established between you and the person you're talking to. No other conversations take place on the circuit until your conversation is complete, and you give up the circuit. In network terms, you establish a *session*—an environment in which you can send and receive messages. Think of setting up a session as being the equivalent of establishing eye contact with someone you wish to communicate with before speaking. The two sides of the communication typically agree upon, or *negotiate*, the characteristics of the communication. With connection-oriented communications, the order of your messages is clear and predictable. The first sentence of your telephone conversation is immediately followed by the second. Connection-oriented communications are said to be reliable. *Reliable* means the network guarantees that it

will deliver your data. It detects and reports any data that's missing, duplicated, or out of order.

In contrast, *connectionless communications* are often compared to U.S. mail communications. You compose a letter, write an address on the envelope, and put the letter in a mailbox. You don't need to establish that the person you're writing is available at the other end. Eventually, the letter will be left at its destination, and the recipient will open, read, and possibly respond to it. With this type of communication, the order of delivery can't be predicted. Two letters, placed in the same mailbox on the same day may be delivered on two different days. Even if they arrive together, there's no way to control which the recipient will open first. Connectionless communications are said to be unreliable. *Unreliable* means the network does not guarantee that it will deliver your data. There's no sure way of telling whether a message has been delivered, or whether data is missing, duplicated, or out of order. Typically, network software deals with the problem of unreliable communications by simply retransmitting a communication if it doesn't receive an acknowledgment after a certain amount of time.

Many networks use packet-switching technologies. With *packet-switching networks*, all communications traffic is broken into small blocks called *packets*. Each message may consist of many packets. Each packet has identifying information associated with it. At the sending end, a message is broken into individual packets, each of which is transmitted through the network as an individual entity. At the receiving end, the message is reassembled from its component packets. Using the identifying information associated with the packets, the message is then routed to its proper destination. With packet-switching technology, a computer connected to a network via a single telephone line could simultaneously hold many conversations over that channel.

Packet switching is the most popular network paradigm in use today, but the packets may not be identically sized. This complicates switching because buffers must be set to accommodate the largest packet anticipated. Asynchronous Transfer Mode uses cell-based switching. ATM cells are very small but contain uniform packets of 53 bytes in length. Five of these are for addressing and control; the other 47 are for data. The small cells allow all switching circuits to be optimized for the same length, greatly increasing throughput.

Multiple networks can connect to form interconnected networks known as *internetworks*. The Internet can be thought of as the ultimate network of interconnected networks. A common method of connecting networks is via a *gateway*—a system, or node, that's part of two networks. Communications from one network to another pass through the gateway that's attached to both of them. From a user's point of view, networks connected by gateways appear to be a single network.

There are many types of network configurations, called topologies. A *topology* is the way the nodes of a network are connected together. Examples of topologies are bus, ring, and star configurations. Interestingly, a logical topology of a bus, for example, may

actually be implemented by running a single wire to each node from a central point or hub. This makes the physical topology a star. A logical ring may take the same shape of a physical star, as is the case with a technology such as token ring, or the topology may be an actual ring, as is the case with a fiber-optic technology such as FDDI.

Protocols and layers

Two systems or users who want to exchange messages must agree on a common protocol. A *network protocol* is a set of rules for how information is exchanged over a communications network. The protocol dictates the formats and the sequences of the messages passed between the sender and the receiver. It establishes the rules for sending and receiving messages and for handling errors. The protocol doesn't need to know the details of the hardware being used or the particular communications method.

The purpose of a *protocol model* is to provide a conceptual basis for describing how to communicate within a network in a way that's independent of the specific rules of the protocol that's being used.

The concept of layering is central to the development of a *protocol suite* or a *protocol family*. Layering divides the communications process into several, relatively independent component processes called *layers*. Each layer provides specific functions and communication with the layers above and beneath it. A protocol model specifies the general characteristics of each layer of services in a network protocol suite. The purpose of a protocol layer is to provide network services (i.e., to transmit and receive data) to the systems or users who are communicating in the layer above it. Within each layer, the two sides of the communication implement the protocols appropriate to the layer.

Examples of protocol suites are Open Systems Interconnection (OSI), TCP/IP, IBM's Systems Network Architecture (SNA), Xerox's Xerox Network System (XNS), Digital Equipment Corporation's DECnet and Digital Network Architecture (DNA), and Apple's AppleTalk. Of these, TCP/IP is the most popular, and OSI is the most theoretical. In fact, the OSI protocol model has become the reference model to which other protocols are compared. This greatly simplifies talking and writing about network layers, because any protocol or system can be spoken of by comparing and contrasting it to the OSI model. You can't get very deep into networking without speaking of OSI layers: WANs, MANs, and LANs.

Computer networks fall into three general categories:

WANs

Wide area networks are large-scale (sometimes called long-haul) networks that span a large geographic area, usually larger than a single city or metropolitan area. Worldwide military networks, public telephone networks, and large funds transfer networks, are examples of WANs. WAN systems are usually connected by leased, high-speed, long distance data circuits, typically using a packet-switched technology.

MANs

Metropolitan area networks are middle-sized networks that may serve an intermediate area such as a small city or a metropolitan area. MAN systems are usually connected by technologies such as coaxial cable or microwave.

LANs

Local area networks are networks that are optimized for a small- to moderate-sized geographic area. Although LANs are typically intended for use with a smaller population of users than WANs and MANs, some LANs can support 1,000 or more users. Office and department networks are examples of LANs. LANs often connect PC systems via technologies such as Ethernet cable or fiber optics. Usually LANs are *broadcast networks*. Instead of routing a message directly from one system to another, LANs typically broadcast messages to every system in the network. Although inherently any system can hear every transmission, by convention each system listens only for messages intended for it.

For completeness, there are a few more ANs that have entered the lexicon:

WLANs

Wireless local area networks are networks that roughly parallel LANs, except that the connection to the work area (the line to the end user) is a radio link, typically some form of IEEE 802.11.

CANs

Campus area networks are the enterprise networks that serve a number of related structures, as in a large company or a college campus.

BANs

Building area networks consist of the integrated cabling systems that comprise the building lifelines, for example, the wires connecting environmental systems, including heating, ventilation, and air conditioning (HVAC), surveillance and security (but not fire or life safety, which use an independent network). Telecommunications and network cabling is a possible target for assimilation into the BAN. Office towers would provide connectivity just as they now provide vertical transport (elevators and escalators) and hot and cold running water. In network security terms, a BAN must be highly secure lest an attacker quite literally shut down the building.

PANs

Personal area networks consist of the myriad of devices worn or carried by individuals that can interconnect, sometimes on an ad hoc basis or autonomously. Thus the pocket organizer that receives updates via Bluetooth when you walk near your desk is part of the PAN.

Some Network History

During the early days of computing, communications links connected central processors to remote terminals and other devices such as printers and remote job entry stations. This technology provided the basis for the first computer networks.

In the 1960s, there was a great expansion in the development of computers and the use of remote multiplexers and concentrators. These devices made network communication more economical by collecting all traffic from a set of peripheral devices in the same area and sending it on a single link to a central processor. Concurrently, special processors called *frontends* were developed to free the CPU from having to handle all communications functions. The challenge during these early days of communications was to figure out how to transmit information efficiently and reliably.

The late 1960s and early 1970s saw the establishment of the first large-scale, general-purpose data networks. The ARPANET network, funded by the Department of Defense Advanced Research Projects Agency (ARPA, now known as DARPA), connected geographically distributed military, university, and research computer systems. The ARPANET was the first wide-area network and the first to use packet-switching technology, which revolutionized computer communication. The original ARPANET allowed different host systems to communicate on the same network via a standard network control program.

In the 1970s, IBM and Xerox also introduced their first networks—IBM's SNA and Xerox's XNS. Xerox PARC also introduced Ethernet packet-switching technology, which was standardized by Xerox, Digital Equipment Corporation, and Intel in 1978 as a network technology that allowed systems to communicate directly, without requiring the use of a central network authority. Ethernet was the first true local area network.

The present-day Internet began to take shape in the 1970s, when DARPA started converting machines to use the TCP/IP protocol suite. By 1983, TCP/IP had become the network standard for the ARPANET. TCP/IP allows systems on different networks to communicate. It's named for its two major protocols—TCP (Transmission Control Protocol) and IP (Internet Protocol). TCP/IP has become tremendously popular because it provides a way to connect systems based on different computers and communications equipment without being concerned about the details of their physical connections.

In the 1980s, IBM introduced the first PC local area network, and interest in the use of networks in small areas such as offices grew dramatically. The 1980s also saw the introduction of the Open Systems Interconnection Basic Reference Model, which is described in Appendix A.

Over the years, new communications technologies have developed, new network media have been introduced, and tremendous growth has occurred in the use of both

wide area networks and local area networks. Today's network challenges include building workable network products based on standards such as OSI, developing standards for network security, and incorporating trusted system concepts and requirements into network implementations.

Hints for Network Security

- Protect your physical cables. A vandal can disable a local area network or a part of a larger network by cutting a single wire.
- Consider the use of conduit, even if it exceeds local fire or building codes. For the most security, seal the cables in plastic and fill the conduits with pressurized gas. If an intruder breaks through a pipe of this kind, a pressure sensor detects the drop in pressure and shuts off traffic or sounds a warning.
- Lock all telecommunications rooms and wiring closets.
- Provide extra physical security for any special systems on your network, for instance, a backup device or backup tape or disk library.
- Use secure offsite storage for backups so that if anything happens to the main facility, a copy of the data will still be intact.
- Provide firewalls and intrusion detection systems for protection both on the network perimeter and interior.
- Use trusted network authentication and encryption products.

Network Media

In communications systems, electronic signals may be carried on any of the following types of network media: twisted pair cable, coaxial cable, fiber-optic cable, microwave, and satellite. Each has functional advantages and disadvantages, and each has security consequences. A network may combine several of these media—for example, each building on a campus might be cabled with local Ethernet cable, but fiber may be used between buildings or floors of buildings. The guidelines in the previous sidebar, “Hints for Network Security,” provide some general network security hints. Supplement these with specific rules for your own cabling and environment.

The entire collection of cables in your facility is generally referred to as the cable plant. Interior cabling generally follows a specific and logical architecture called a structured cabling system. Cabling between buildings is called outside plant, or OSP cabling. OSP follows a different set of rules dealing with rights of way, usage of poles, and depth of burial. Adhering to the appropriate codes and standards will almost always make your network more reliable, and in most cases it will be easier to secure, because unauthorized attachments will be easier to spot.

Twisted pair cable

Twisted pair is the type of cable used most often for telephone systems and for LANs. Twisted pair cable is the cheapest type of conventional cabling, but it's limited in distance and bandwidth (and thus in the number of communications it can carry on a single line). It's called twisted pair because it consists of two insulated wires twisted together. Twisting wires together allows them to mutually cancel out the creation of magnetic fields, which can rob energy and create a potential for eavesdropping. Twisting pairs also allows both halves of each circuit to experience noise and interference in the same way, simplifying filtering.

These days a single twisted pair cable is a rarity. In most cases, the pairs will be bundled into groups of four, and covered with a common sheath or jacket. Sometimes a shield is added to control electromagnetic emissions. There are security problems with twisted pair cable, because it's very easy to tap into a twisted pair communication. For this reason, wiring pathways and spaces should be locked where possible.

Coaxial cable

Coaxial cable was formerly used to connect network devices, although its current role is primarily for video, as a medium for cable modems, and as antenna leads on wireless access points. Coaxial cable is frequently made of copper and it's more expensive than twisted pair cable, and more resistant to electromagnetic interference. Like twisted pair, coaxial cable may be shielded to control emissions.

There are two techniques for transmitting a signal over a coaxial cable: baseband and broadband. With *baseband*, only a single channel is transmitted. With *broadband*, many channels, including video, voice, and data, can be carried simultaneously over greater distances. Coaxial cable has some of the same security problems as twisted pair; it's very easy to tap into a coaxial communication.

Fiber-optic cable

Fiber-optic cable carries signals as light waves rather than as electrical impulses. It offers many functional advantages (e.g., speed, longer distances, cost), and it provides far better security than other types of cable. For example:

Fiber is difficult to tap

Fiber-optic cable is a very difficult medium for intruders to tap because it's not electrical, and it doesn't radiate. It can't be tapped by induction devices or located by metal detectors. Any tampering with a fiber-optic network is readily detected, so it's difficult to insert a listening device in the cable.

Fiber is resistant to interference

Because fiber-optic cable is immune to electromagnetic interference (EMI) and radio frequency interference (RFI) and doesn't create its own electromagnetic interference, it's particularly appropriate in environments where such interference

might be a problem (e.g., radar corridors, areas with heavy electrical equipment, or areas with equipment used to monitor critical data).

Fiber is resistant to hazards

Fiber-optic cable is the right choice in hazardous environments such as wet or corrosive areas, areas near high-voltage lines, and areas of frequent lightning or power surges.

Microwave

Microwave or wireless usually isn't an exclusive medium for a network. Instead, it's used in conjunction with other networks—for example, as a gateway between two LANs separated by some geographical distance (e.g., across a campus, a body of water, or a city). Microwave is less secure than fiber, coaxial, and twisted pair cable because communications can be intercepted through the air via an antenna. The only way to increase wireless security is to encode the transmissions; however, that adds a layer of complexity to the communications process. Unfortunately, the Wired Equivalency Protocol (WEP) or as some say, Wired Equivalent Privacy (WEP), security system that was originally issued for Wi-Fi can be cracked. An alternative, Wi-Fi Protected Access (WPA), is similar to Microsoft's Simple Secure Networking (SSN), which is built into the Windows XP operating system. WPA offers greatly improved security. Users who have Cisco equipment can use a technology called LEAP. The ultimate answer—so far—seems to be the IEEE 802.11i standard, which defines the Temporal Key Integrity Protocol (TKIP), and which is designed to accommodate the Advanced Encryption Standard.

An older kind of microwave system, called a fixed microwave system, provides longer paths for wireless links. Often these systems work as relays, with a signal taking several hops before reaching its eventual destination. These paths too are vulnerable to eavesdropping. Fixed microwave has been replaced in many applications by fiber optics, which offers greater bandwidth and easier maintenance.

Another microwave system, IEEE 802.16 (WiMAX) promises to offer wider range than 802.11. This system grew out of proposals for dispersed, localized low-power wireless cable TV systems, similar in topology to a cellular telephone network.

Satellite

Like microwave, satellite is often used with other networks to connect two distant points. Because of the delays implicit in satellite communication, satellite may be appropriate for certain types of computer communications (e.g., file transfers), but not for other types (e.g., terminal interactions). From a security point of view, satellite is not very secure. Like microwave communications, satellite communications can be intercepted through the air via an antenna. Nevertheless, satellite is increasing in importance in those areas that lack good wired infrastructure. As with microwave communications, encryption services can protect data if required, but at a cost of bandwidth or reduced speed.

Network Security

In the early days of networks, a network administrator usually had tight control over whether a system could connect to another remote system. These days, with the proliferation of interconnected networks and easy remote access and resource sharing, it's often impossible to identify—never mind to trust—all of the points of access to a system.

There are a number of different strategies for accomplishing security in a network environment. The choice of which and how many strategies to use depends largely on the type and scope of the network, the level of trust that can be placed in the users, and the value of the data that's being transmitted.

Network Security Standards

Several standards are important to network security:

The Institute of Electrical and Electronic Engineers (IEEE) has developed a set of standards, called *802 standards*, primarily for local area networks; some of these standards, such as those used for wireless, are still in the proposal stage and have not yet been published as final standards. Others have been through extensive modification, with additions being noted as letters at the end of the standard's name. For instance, the standard for transmission of power over the cables used to connect devices by Ethernet (Power over Ethernet, or PoE) has the name IEEE 802.3af.

The early 802 standards, which include 802.1 through 802.10, basically address the lowest two layers of the OSI model. 802.10 is a standard for interoperable LAN security, known as S1LS, which is oriented to the exchange of data in multivendor networks.

The *X.400 standards* developed by ISO and CCITT (Comité Consultatif Internationale Télégraphique et Téléphonique, today called International Telecommunications Union—Telecommunications division, or ITU-T) are OSI-oriented protocols for message handling (for example, in electronic mail systems). They include standards for secure messaging.

The *X.500 standards* developed by ISO and CCITT are standards for naming. They allow users and programmers to identify an object (e.g., a file, a disk, etc.) without knowing the location of the object in a network or the path required to reach it. X.500 includes standards for authentication and secure naming. Most current authentication services depend on a system of parameters defined in the X.500.

The X.500 standard is such a comprehensive system that in most cases an X.500 directory is usually accessed using a streamlined tool called Lightweight Directory Access Protocol.

Most network security mandated by government projects follows a series of standards found in a series of publications called the Federal Information Processing Standards.

Access Control Methods

An interesting problem with security is that not only must information be protected from outsiders, it must sometimes be protected from insiders as well. For instance, patient information in a doctor's office can be accessed by medical staff, and in fact in an emergency should be readily available. However, vendors who visit the office must not be allowed to see it, nor should cleaners or facility maintenance personnel. Keeping information stratified inside an organization is a form of *access control*. Various methods that control access to network environments are described in the following sections.

Discretionary access control

In an operating system, discretionary access control (DAC) can be used to restrict file access to certain users or groups. In a network environment, DAC may restrict access to certain remote users and/or systems. A particular network service might be available only to a certain group, which might be defined in a network environment as a particular Internet address (e.g., all the users of a particular system in the network).

Role-based access control

In many cases, it is not so much the person as their position in the organization that determines whether or not they should have access to a given record or file. Engineers rarely need access to payroll data. They would likely look up coworkers' salaries, get jealous or gloat, and perhaps post the information on the Internet somewhere. Accountants rarely require access to the wind-tunnel test data of the secret new fighter aircraft. It might not mean much to them, and they may try to correct minor math errors they may come across. (On the other hand, this can be an advantage. One very sensitive national defense project was discovered to be flawed when the calculations regarding thrust of a rocket engine were found to contain an incorrect formula. The engineers did not detect their error mathematically; it first showed up when a computer artist modeled the burn on PC and the graphic results changed unexpectedly at the point where the formula was flawed.) To separate information by department rather than by person is called *role-based access control*.

Mandatory access control

Every system in a trusted network must label its data with security attributes (e.g., sensitivity labels, information labels, login IDs, etc.). This way, the sensitivity of the data will be recognized if the data is sent to another system. Because different networks support different security policies, these labels are not necessarily in the same format.

In certain types of secure networks, each system may effectively have a label. Mandatory access control keeps TOP SECRET data, for example, from being sent over the network to a system labeled as SECRET. It is actually quite challenging to insure that

no TOP SECRET documents are read by persons with only SECRET clearance, a condition called *read down*. Similarly, it must be made impossible for persons with SECRET clearance to save documents with a TOP SECRET classification (write up).

Auditing

In a network environment, additional networking events must be audited on both sides of the network connection. Examples include establishing or dropping a network connection, security violations such as lost or misrouted data, and failure of a network component.

Perimeters and Gateways

The simplest way to protect a network from access by unauthorized users is to keep that network physically secure—for example, to provide physical protection of all internal network switches and connections, no telephone connections, and no network cabling to the outside world. As mentioned previously, such a system can be nicknamed as an airwall, because there is a effectual gap between external elements and internal elements. But with trends toward wider communication, most organizations will at least occasionally have to communicate with outside systems and networks. Communication between trusted and untrusted networks must have very clear rules associated with it.

A trusted local area network can be thought of as being inside a security perimeter. Inside the perimeter, access controls and other security features determine who can access what information. For example, in a trusted system supporting multilevel military security, some users are cleared to access TOP SECRET data, others only SECRET, CONFIDENTIAL, or UNCLASSIFIED data. In a fairly simple network environment, all information originating outside the trusted network might be treated at a single sensitivity level. In such an environment, a gateway system, known as a firewall, or a firewall computer if it is built out of a PC with two network cards, might separate the trusted system or network from the untrusted systems or networks outside it. Untrusted systems can communicate with trusted systems only through a single communications channel controlled by a trusted gateway. The gateway controls traffic from both inside and outside the network and effectively isolates the trusted network from the outside world. Because the firewall protects the other machines within the perimeter, security can be concentrated on the firewall.

In the simplest case, where everything inside the perimeter is trusted and everything outside is untrusted, the gateway system labels and filters data. When information is imported into a trusted system from an untrusted system, the gateway system puts a sensitivity label (usually “UNCLASSIFIED”) on this data. When information is exported from a trusted system to an untrusted system, the gateway system filters that information by exporting only data that the untrusted system is allowed to process (usually UNCLASSIFIED data). If multilevel security is

supported, more complex network solutions are required to regulate access to the different security levels, as described in the next section.

Security in Heterogeneous Environments

More and more modern networks are attempting to serve heterogeneous computing environments. Networks must have facilities for supporting a whole range of security environments, corresponding to the host systems they serve.

Using a trusted gateway to partition trusted networks from untrusted networks, as described in the previous section, works only if communications from outside the trusted network can be treated at a single level of security. In more complex environments supporting multilevel communications, more complex solutions are needed.

On the trusted network side, a system must make decisions about requests originating outside the trusted network. It must regulate which information remote users are cleared to access, and it must control which system services remote users are granted. In standard networks, in which the security attributes of the remote user aren't transmitted, the local system doesn't have sufficient information to make access decisions. This can lead to major security problems. For example, if a trusted network server on the trusted network side processes requests on behalf of an untrusted remote user, the security of the whole system is at risk.

There must be some way to propagate to the local process (the process on the trusted network side) the security attributes of the remote user (the process on the untrusted network side). This isn't as simple as sending a "TOP SECRET" label, for example, from one side to the other. Even if a user is cleared for TOP SECRET information on one system, he's not necessarily given the same courtesy when he accesses another system. Each system in the network must be able to compare the attributes of the process performing an operation with those of the file or other object on which the operation is performed, and make access decisions that are appropriate to the local system's security policy. To do otherwise invites incredible risk, especially in this day when snippets of code are passed readily between computers for execution.

Encrypted Communications

Encryption, a process that transforms information into another form that cannot be read by unauthorized users, is a very important part of network security. Because information is so vulnerable to attack when it's being transmitted over a network, encryption offers a strong assurance that even if the information is intercepted, it won't be comprehensible. (See Chapter 7 for a detailed description of encryption.)

In addition to protecting the secrecy of messages transmitted over a network by transforming them into data that appears to be unintelligible, encryption can also ensure the integrity and authenticity of those messages. *Message authentication* provides a critical network security tool; it ensures that a message was received in

exactly the form sent. At the sending end, the encryption process appends a message authentication code to the encrypted message. At the receiving end, the decryption process independently calculates a message authentication code and compares it with the code sent with the message. If the two are identical, the message was sent and received accurately.

A *digital signature* is another network security tool that provides electronic evidence that you, and only you, sent a signed message. Because the message incorporates a secret key that only you possess, it can't be forged. It's an immutable proof that you sent a message.

A message that's proved to be authentic by an outside authority, sometimes called a *notary*, is said to be *arbitrated*. As with a traditional notary, who provides independent evidence that someone is who she claims to be, a digital notary attests that a message is genuine. The notary may also provide proof that the message was sent and received at a particular, recorded point in time. This keeps the message from being repudiated by the sender and/or the receiver later on.

There are two communications levels at which encryption can be performed, each with different implications for network security: end-to-end encryption and link encryption.

End-to-end encryption

With *end-to-end encryption* (sometimes called *off-line encryption*), a message is encrypted when it's transmitted and decrypted when it's received. The network may not even need to be aware that the message is encrypted. This type of encryption sometimes may be selected as an option by the user. The message remains encrypted through the entire communications process, from start to finish, as shown in Figure 8-1. See "End-to-End and Link Encryption" in Chapter 7 for more information.

Link encryption

With *link encryption* (sometimes called *online encryption*), a message is encrypted when it is transmitted, but is decrypted and then encrypted again each time it passes through a network communications node. The message may therefore be encrypted, decrypted, and reencrypted a number of times during the communications process, and the message is exposed within each node, as shown in Figure 8-2. With link encryption, the encryption is performed just before the message is physically transmitted. Encryption is typically invisible to the user; it is simply part of the transmission process. See "End-to-End and Link Encryption" in Chapter 7 for more information

Through the Tunnel

One popular technique of encrypting traffic as it travels over a link is called *tunneling*. As mentioned previously, one of the problems with encrypting a link is that the

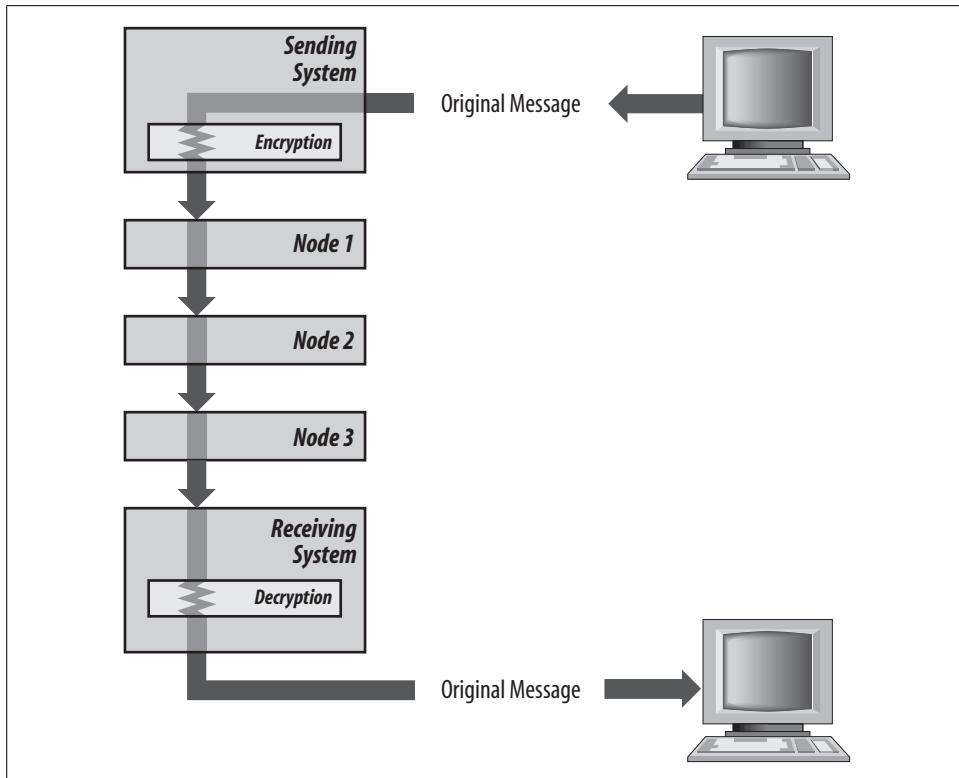


Figure 8-1. End-to-end encryption

headers and trailers of each packet, which are the points in the packet where the addressing occurs, have to remain in the clear. This may reveal as much information as if the packet itself was open. (Encoded messages to the enemy's Weapons of Mass Destruction department must always be looked at carefully.)

To encrypt the addresses requires that the network be secured in its entirety, drastically increasing expenses. At the very least it requires a system of leased lines, to which no unauthorized parties have access. Maintaining such a network can also be quite expensive, and it doesn't truly guarantee that the lines are not monitored by spies, only that they are not supposed to be. In addition, the cost of such networks may vary by the mile or by the amount of capacity used. This means that the more you use your network, the more costly it can become. In other words, it is difficult to scale a network that consists of many dedicated circuits.

Tunneling is the basis of a host of secure technologies. To create a tunnel, packets are placed in a wrapper, which contains the network addressing information. While the wrapper is in place, it handles the navigation through the network. The actual packets are securely encoded inside the wrapper. Once the wrapper carries the packet

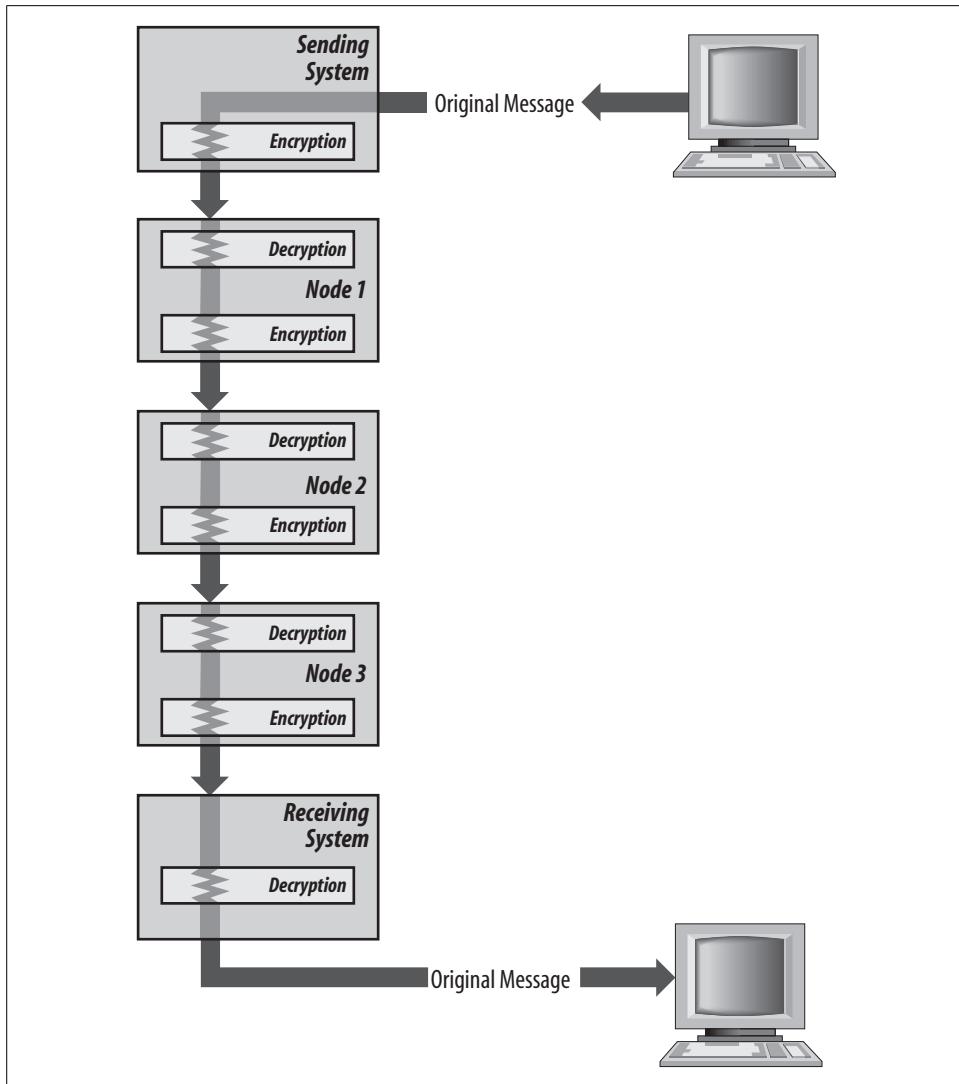


Figure 8-2. Link encryption

across the uncontrolled network, the wrapper can be taken off, and the packet decrypted so that its address can be interpreted normally within the secure domain.

Tunnels are the basis of virtual private networking (VPN). Basically, a VPN is a private network that uses a public network, often the Internet, to connect remote users and sites together. Because the network behaves as if it was under private control even though it is not, it is called a virtual private network. A VPN uses virtual, non-physical connections that are routed through a public network or the Internet to tunnel packets from the company's private network to the remote site. This means that the user can obtain the utility of dedicated links without having to maintain all those links.

VPNs for remote access

VPNs have grown in popularity because they support remote access service. In recent years, many organizations have increased the mobility of their workers by allowing more employees to telecommute. Employees also continue to travel and face an increasing need to stay “plugged in” to the company network. Leased lines don’t support mobile workers well because the lines fail to extend to people’s homes or their travel destinations. Companies that don’t use VPNs must resort to implementing specialized “secure dial-up” services. To log in to a dial-up intranet, a remote worker must call into a company’s remote access server using either a toll-free number or a local number. The overhead of maintaining such a system internally, coupled with the possibility of high long distance charges incurred by travelers, make use of VPNs instead an appealing option here.

Figure 8-3 illustrates a VPN remote access solution. A remote node (client) wanting to log into the company VPN calls into a local server connected to the public network. The VPN client establishes a connection to the VPN server maintained at the company site. Once the connection has been established, the remote client can communicate with the company network just as securely over the public network as if it resided on the internal LAN itself.

VPNs for internetworking

A simple extension of the VPN remote access architecture allows an entire remote network (rather than just a single remote client) to join the local network. Rather than a client-server connection, a server-server VPN connection joins two networks to form an extended intranet or extranet.

VPNs inside the firewall

Intranets can also use VPN technology to implement controlled access to individual subnets on the private network. In this mode, VPN clients connect to a VPN server that acts as a gateway to computers behind it on the subnet. Note that this type of VPN use does not involve an ISP or public network cabling. However, it does take advantage of the security features and convenience of VPN technology.

VPN tunneling protocols

Several interesting network protocols have been implemented specifically for use with VPN tunnels. The three most popular VPN tunneling protocols are listed next. These protocols are generally incompatible with each other.

Point-to-Point Tunneling Protocol (PPTP)

Several corporations worked together to create the PPTP specification. People generally associate PPTP with Microsoft because nearly all flavors of Windows include built-in client support for this protocol. Security has improved over the years.

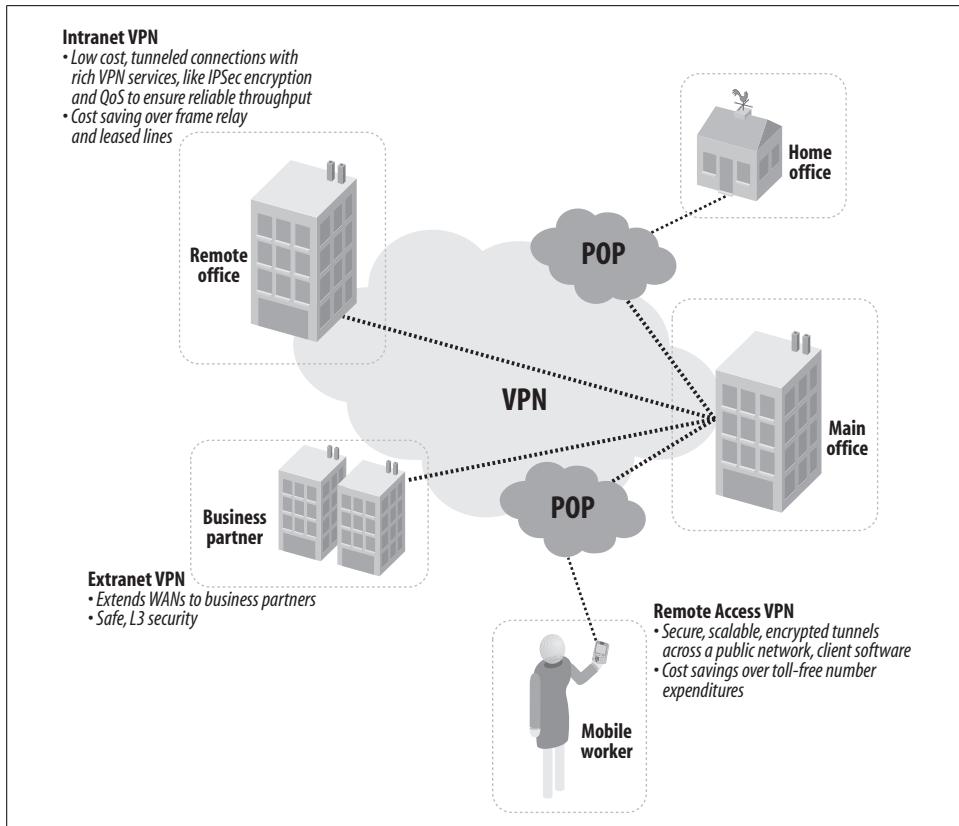


Figure 8-3. VPN Remote Access Architecture

Layer Two Tunneling Protocol (L2TP)

The original competitor to PPTP for VPN tunneling was L2F, a protocol implemented primarily in Cisco products. In an attempt to improve on L2F, its best features were combined with PPTP to create a new standard called L2TP. Like PPTP, L2TP exists at the data link layer (Layer Two) in the OSI model—thus the origin of its name. Refer to Appendix A for more information on the OSI mode.

Internet Protocol Security (IPSec)

Internet Protocol Security (IPSec) is actually a collection of multiple related protocols. It can be used as a complete VPN protocol solution, or it can be used simply as the encryption scheme within L2TP or PPTP. IPSec exists at the network layer (Layer Three) in OSI. IPSec is covered more fully in a later section.

Network Security Tasks

Providing a secure connection between sender and receiver is a technical challenge. It is divided into several subtechnologies that are described in the following sections.

Communications integrity

Communications integrity services ensure that network communications are transmitted accurately. This means that messages aren't forged, modified during transmission, or repudiated by either the sender or the receiver of the messages. (See Table 8-1.)

Table 8-1. Communications integrity services

Service	Meaning
Authentication	Proves the identity of the user and system sending a message. Ensures that unauthorized users can't pretend to be another (called a <i>masquerade</i>). Authentication also ensures that an unauthorized user can't record and then resend a previously sent message (called a <i>playback</i> or a <i>replay attack</i>). Network security techniques: encryption, passwords, digital signature, timestamp on message.
Communications field integrity	Protects the accuracy and integrity of the message. Ensures that the message, including specific message fields (such as the protocol header used for routing) are not changed (via message stream modification)—either deliberately or accidentally.
Nonrepudiation	Proves that a message has been sent and received. Ensures that the sender can't deny that he sent the message, and that the recipient can't deny that he received it.

Denial of service

Denial of service services are designed to ensure that the network keeps working and that all services needed by users are fully available. This means that the network supports good system administration and that there are methods for keeping threats such as message flooding and worms out of the network. (See Table 8-2.)

Table 8-2. Denial of service services

Service	Meaning
Continuity of operations	Keeps the network working efficiently, even if components fail or the network is attacked. Network security techniques: redundant or fault-tolerant systems and devices, ability to reconfigure network in an emergency.
Protocol-based protection	Detects network problems (e.g., slowdown in transmission) using existing protocol services. Network security techniques: measurement of transmission rates between systems (compare with minimum) and waiting time for responses (compare with threshold).
Network management	Monitors overall network performance to detect network attacks (e.g., message flooding, replays), failures (e.g., message overload, noise), or inequities (some users or processes having a processing advantage over others). Network security techniques: restriction of resources via network administration (e.g., limited CPU time, number of jobs, number of disks and tapes, number of files, file size).

Compromise protection

Compromise protection services are designed to keep information transmitted over the network a secret from those not authorized to access it. This means that the network provides methods for keeping intruders out of the network—both physically and remotely. (See Table 8-3.)

Table 8-3. Compromise protection services

Service	Meaning
Data confidentiality	Protects data from being intercepted by unauthorized users during transmission. Network security techniques: physical protection of cabling, access controls.
Traffic flow confidentiality	Protects data characteristics (e.g., message length, frequency, destination) from being analyzed by an intruder (called <i>traffic analysis</i>). Such an analysis might allow an observer to infer information; a classic example is that a pattern of messages between a general and troops in the field could predict that the troops are about to launch an attack. Network security techniques: covert channel analysis, padding messages (to disguise their actual characteristics), sending noise or spurious messages.
Selective routing	Avoids particular threats to data by routing messages to avoid certain networks or systems (e.g., systems in certain countries or certain suspicious network nodes). Network security techniques: network configuration, periodic deletion or modification of messages.

Securing Communications

To sum up, you need to provide data privacy, authenticity, integrity, and protection against replay attacks for network traffic whether communications are between a client and a server, between two servers, or between two clients. You also need to provide protection when clients access the network remotely, in case you do not want to use a VPN. PPTP and L2TP can provide the security that you need for remote access. The technique discussed next, IPSec, is not designed to be used for VPN remote access.

Internet Protocol Security (IPSec)

To secure communications over the Internet, which is an open network, requires some additional tools. Internet Protocol Security (IPSec) can provide security across public networks or private networks that you may lease but do not control or for which you cannot guarantee physical security.

IPSec was designed by the Internet Engineering Task Force. It defines IP packet formats and related infrastructure to provide end-to-end strong authentication, integrity, antireplay, and if you add an encryption layer, message confidentiality for traffic on networks.

An automatic key management service, using the IETF-defined Internet Key Exchange (IKE) based on RFC 2409, can be integrated into the connection. Using

IKE provides Kerberos authentication, public/private key signatures, and passwords to establish authenticity and to establish trusted communications between computers. The latter two technologies have been discussed earlier. Kerberos will be detailed in the next section.

Once two communicating computers have authenticated each other, they use computer algorithms to generate encryption keys that are known only to the two computers. This allows them to protect the data against modification or interpretation by attackers who may be lurking in the network. The generated keys are created in bulk, because they are automatically refreshed according to IPSec policy settings defined by the administrator, and only a true communications partner would have the correct next key. Making sure both computers start from the same position is the task of the IKE, which insures that the type and strength of keys to use for authentication, and which type of security should be applied to the *application traffic* being communicated, are the same at each end of the link.

Kerberos

Kerberos is an authentication system for open systems and networks. Developed by Project Athena at the Massachusetts Institute of Technology, Kerberos can be added to any existing network protocol. Historically, Kerberos has been used with Unix-oriented protocols such as Sun’s Network File System and in the exchange of certificates. Kerberos uses an encryption system based on the Data Encryption Standard (described in Chapter 7). Each user has a private authentication key.

How does Kerberos work? Like its namesake, the many-headed dog who guards the entrance to the underworld, Kerberos guards the data transmitted between machines that communicate over the network. Kerberos uses cryptographic keys known as *tickets* to protect the security of the messages you send to the system—and the messages the system sends back to you. Kerberos never transmits passwords, even in encrypted form, on the network. Passwords reside only in a highly secure machine called a *key server*. Kerberos performs authentication both when you log into the system and when you request any type of network service (e.g., a printer or a mail system).

The Kerberos authentication sequence works like this:

1. When you log in, you enter your login name. The login process sends your login name to the Kerberos key distribution service, which returns the following to you:
 - An encrypted session key—a temporary key you use to communicate with the Kerberos ticket granting service (described later).
 - An encrypted ticket for the Kerberos ticket granting service.
2. You enter your password. The login process uses your password as a private key to decrypt the session key and the ticket sent to you by the key distribution service. If the decryption works, you’re authenticated.

3. When you request a network service (e.g., mail), the system sends your temporary session key and your ticket granting ticket to the Kerberos ticket granting service. Each service has its own password. The ticket granting service returns a temporary key and a ticket for use with the service. The system uses your session key to decrypt the key and the ticket.
4. To make the connection to the service, the system sends the service your session key, your temporary service key, and your service ticket. If the server can decrypt the request sent to it, you're allowed to use the service.

Summary

The computer and operating system safeguards described in this book protect information very effectively as long as the information remains safely in the computer, under the operating system's control. But, in the world of networks, and the Internet, information is increasingly on the move—being shared and communicated among different users on different systems across the globe. Information that's protected securely by an operating system becomes much more vulnerable when it's being transmitted—over telephone lines or network connections, via satellite or microwave. Whether you are sending your thoughts via electronic mail, presenting your new sales plan using a shared document session, or simply transferring data files, once the information enters the wild, your materials are potentially available to anyone. Instead of being available to only a relatively small population of users within your own organization, your computer system potentially becomes open to attack by anyone who has access to a modem or a network connection. Appropriate steps, such as are detailed in this chapter, must be taken.

PART IV

Other Types of Security

Physical Security and Biometrics

Physical security protects your physical computer and networking facilities—your building, your server room, your computers, your backup media, and increasingly, your people. *Biometric devices* (devices that sample a physical or behavioral trait—for example, a fingerprint—and compare it with the traits on file to determine whether you are who you claim to be) provide an important first defense against break-ins.

Sound physical security is the basis on which a security policy and its resulting activities must rest. The best computer security activities can easily be negated by careless physical security. The importance of good physical security can be illustrated by referring once again to the two security mnemonics: CIA and IRA. The first refers to the following:

Confidentiality

That which is secret should stay that way.

Integrity

That which is received is identical to what was sent, what is retrieved should be identical to what was stored.

Accessibility

What is stored should be easy to retrieve, what you need or where you wish to go should always be available to you.

The stepping stones to these are in the IRA:

Identification

Being able to prove you are who you say you are.

Reliability

All systems function as they should on demand, even in a crisis.

Authentication

Making sure users can access only those areas for which they are authorized.

To show how these relate to physical security, consider the following story.

In December 2002, social security numbers and other personal information for a half million military personnel, family members, and retirees were stolen by heisting the system's backup devices. The theft occurred on a Saturday and was detected the following Monday. Medical insurance claims data for active military personnel in the western portion of the United States were involved, and the matter was treated seriously, including posting of a reward, providing periodic updates of the case on a web site, and providing information to those affected about what to do if they suspected their stolen information had been misused. From backups (early news reports stated that the stolen drives were in fact "expensive backup devices"), the facility quickly determined which records were compromised and notified those affected with details of the stolen information, including copies of forms they may have filled out, so that the victims could see what data was at risk. No cases of identity theft or other crime have been attributed to this theft and the facility has since increased both electronic and physical security, and in fact has been awarded a contract for a considerably increased service area.

Upon further investigation, it became clear that the IT processes in place at this organization were reasonable. However, with a break-in on a Saturday being reported on Monday, there may have been a breakdown in physical security. It is clear that problems with physical security can undermine the best of intentions with electronic security.

Physical Security

In the early days of computing, computers—and the information they processed—were protected in the most fundamental way: they were locked up, with entry limited to a few authorized operators and users. Today, a few servers may remain behind bars, but personal computers are widely distributed, taking the place of terminals and having access to nearly all the information that once was locked away in secured areas. Using wireless access points, it is not even necessary to make a physical connection to the network to get a clear shot at almost any resource. Easy electronic access has logically led to an increased emphasis on electronic security.

However, despite advances in computer security and communications security, physical security remains a vitally important component of your total security plan. Physical security measures are tangible defenses that can protect your facility, equipment, and information from theft, tampering, careless misuse, and natural disasters. In some ways, physical security is the easiest and the most rewarding type of security. It's very visible and reassuring. It's a tangible signal to employees and visitors that you take security seriously. Building, server room, and telecommunications room locks provide an important outer, physical perimeter of security, within which electronic measures and other types of security provide finer-grained protection of information.

Natural Disasters

The discussion of information security risks throughout this book has focused on man-made disasters such as sabotage, hacking, and human error. But don't forget that computers and networks are affected by the same kinds of dangers that imperil all of your organization's equipment—fire, flood, lightning, earthquakes, and other natural disasters.

In fact, many natural threats are actually more of an issue for computers than for other types of equipment because computers and associated equipment are particularly sensitive to temperature changes, moisture, power loss, and surges in electricity. And while computers are easily replaceable, the information they contain may not be.

The suggestions provided in the following sections are very brief and basic ones. If you're in a high-risk area for any of these hazards, consider the importance of planning for redundancy, disaster recovery, and business continuance, as discussed in Chapter 5.

Fire and smoke

- Install smoke detectors near your equipment—and check them periodically.
- Keep fire extinguishers in and near your server rooms, telecommunications rooms, and work areas, and be sure that everyone knows they are there.
- Make sure that fire extinguishers are inspected regularly and are of the correct type and rating (ABCD code).
- Enforce no-smoking laws and policies; these are also important to controlling smoke, another hazard to computers.
- Ensure that specialized gas systems for fire control, such as Halon and carbon dioxide, are operable, cannot be accidentally or carelessly discharged, and are in compliance with environmental laws.
- Depending on local codes, it may be a good idea to have the air conditioning system interface with the fire alarm system, so the AC can be shut down if a fire in another part of the building threatens to inject smoke into the server room via the AC ductwork.

Climate

- Keep all rooms containing computers at reasonable temperatures (approximately 50–80 degrees Fahrenheit or 10–26 degrees Celsius).
- Keep telecommunications rooms and server rooms decidedly cool; if you need a sweater while working in them, that's about right.
- Keep the humidity level at 20–80 percent.

- Install gauges and alarms that warn you if the temperature or humidity is getting out of range.
- Equip your heating and cooling systems with air filters to protect against dust (another peril to computers and especially to older tapes and disk packs, and to certain optical media).

Earthquakes and vibration

- Keep computers and telecommunications equipment away from glass windows and high surfaces, particularly if you're in a high-risk area.
- Rack-mount equipment where possible, remembering to secure the floor plates. Use the ANSI/TIA/EIA-569-A standard on telecommunications pathways and spaces, with local seismic variations as your guide.
- Be sure that if strong vibration occurs (because of earthquakes, construction, or other sources), objects won't easily fall on your computers and network equipment.

Water

- There are various types of water damage. Flooding can result from rain or ice buildup outside, toilet or sink overflow inside, or the water from sprinklers used to fight a fire. Air conditioners and other cooling units may create water due to condensation. This is usually held in trays, but these can rust out or overflow. Be sure you've protected against all types of moisture.
- If your computer does get wet, let it dry thoroughly before you attempt to turn it on again.
- Install a water sensor where appropriate. Simple ones are available for the price of a smoke alarm.
- Remember that the presence of water increases the likelihood of electrical shock. Use greater caution in the case of flooding emergencies in equipment areas.

Electricity

- Your computer will suffer if it gets too much or too little electricity.
- For best results, install an uninterruptible power supply. It will absorb surges and provide extra voltage during brownouts, and if power fails completely, it will provide power until you're able to shut down the system. An unprotected power loss can result in serious damage. Note that surge protection won't work unless your electrical system is well-grounded.
- Install a line filter on your computer's power supply; a voltage spike caused by lightning or a power fault can destroy your computer.
- Verify that the protective grounding system is adequate. This may require an electrician or grounding specialist. Local applicable standards, such as ANSI standard 942 and J-STD 607-A should be your guide, along with the local electrical code.

- If you can, install a special electrical circuit with a clearly labeled circuit breaker for each of your systems.
- Install antistatic carpeting in your facility. This carpeting contains special filaments that dissipate static electricity.
- Have a telecommunications specialist and an electrician verify the effectiveness of your equipment or signal grounding system. In most cases, the signal ground must be electrically bonded to the electrical or protective grounding system.

Lightning

- If a lightning storm hits, try to turn off your computer and unplug it. Lightning generates an enormous power surge that can damage your computer even if you have a surge protector on your computer.
- If you use magnetic media as a back up, protect it from the magnetic field created if lightning strikes your building. Store the media as far away as possible from the building's steel supports. Even metal shelving may pose a hazard.

Risk Analysis and Disaster Planning

One of the most important things you can do to protect your organization from disaster is to plan for that disaster. Risk assessment and disaster planning are vital security activities, and they're rarely performed, except by the most informed organizations. For a description of what these activities are all about, see the section "Planning for Disaster" in Chapter 5.

Locks and Keys: Old and New

The first line of defense against intruders is to keep them out—out of your building, out of your server room, out of your telecommunications closets. In the past this was easier. The locked or guarded computer room has historically been the primary means of protecting an organization's computer equipment and information from physical intrusion and unrestricted access. The terminals that were on desktops were very thin clients, "dumb terminals" in fact, capable of manipulating data only in the mainframe. In most organizations these days, everyone has a workstation, from which information can be removed easily on a USB memory stick or floppy disk, for instance. Printers, from which documents can be collected, are distributed around the office. Locking up is harder in this kind of environment, but it still is a good idea. Kick the janitor out of the telecommunications closet and spill the cleaning supply storage shelves into the hallway. Boot the network administrators out of the server room and put on a combination lock. Gather up the backup tapes and the old hard drives and installation CDs for operating systems and office applications, and store them in a locked metal cabinet.

To gain access to a locked facility, a user should have to pass an authentication test. Remember from Chapter 3 that there are three classic ways in which you identify yourself (i.e., prove that you are who you say you are):

- What you know—for example, a password.
- What you have—for example, a key, a token, a badge, or a smart card.
- What you are—for example, the fingerprint on your finger (which matches the one on file).

All of these authentication techniques can be used for physical security (e.g., building or computer room access) as well as for system access control. When a smart card or a fingerprint is used for computer access, it's usually only a first step. Passwords are typically required as well. When two distinct techniques are used for authentication in this way, it's called *two-factor authentication*. One factor is something you have; for example, you present your smart card or have your fingerprint or voiceprint scanned. The other factor is something you know; for example, you type a personal identification number (PIN) or a password into the system. Multifactor identification systems promote a “defense in depth” environment.

Hints for Keeping Intruders Out

Don't forget that the front door isn't the only way into your facility. Be careful about:

Dropped ceilings

Be sure the walls extend above the dropped ceiling so intruders can't climb over the walls.

Raised floors

Be sure the walls extend down beyond the raised floor so intruders can't crawl under the raised floor.

Air ducts

Be sure the air ducts are small enough so intruders can't crawl through them.

Glass walls

They're too easy to break, and breakage will cause a lot of damage. People can also look through them and potentially get access to sensitive information that way.

Photographs

Follow the lead of several national monuments and take a digital photograph. Sneaks don't like to be documented. Perhaps you can use the photo on a “guest ID” badge so it can't be pocketed and reused by someone else.

Network connections

Intruders can't sneak in over communications lines (it just seems that way), but they can cause a lot of damage by unplugging or cutting cables. Physical access to cabling also opens up the possibility of wiretapping. Current standards require that wiring pathways and spaces be kept locked.

Some organizations add another dimension of security to a locked facility by using surveillance devices, such as closed-circuit television and more sophisticated access detectors that use infrared, ultrasonic, laser, or audio technologies. Advanced digital video recorders can record multiple cameras and cause an alarm when images change, such as people appearing in certain hallways or going through certain doors.

Highly secure facilities can install elaborate turnstiles called *mantraps* (see Figure 9-1). Systems of this kind route personnel through a double-doored facility in which you show a badge to a guard or are subjected to verification techniques, such as weight checks (to ensure against your entering the facility with an unauthorized buddy), key checks, and biometric checks, such as those described later. If you fail to pass the tests, you are trapped between the double doors, unable to enter or exit the facility until a security officer investigates the incident!

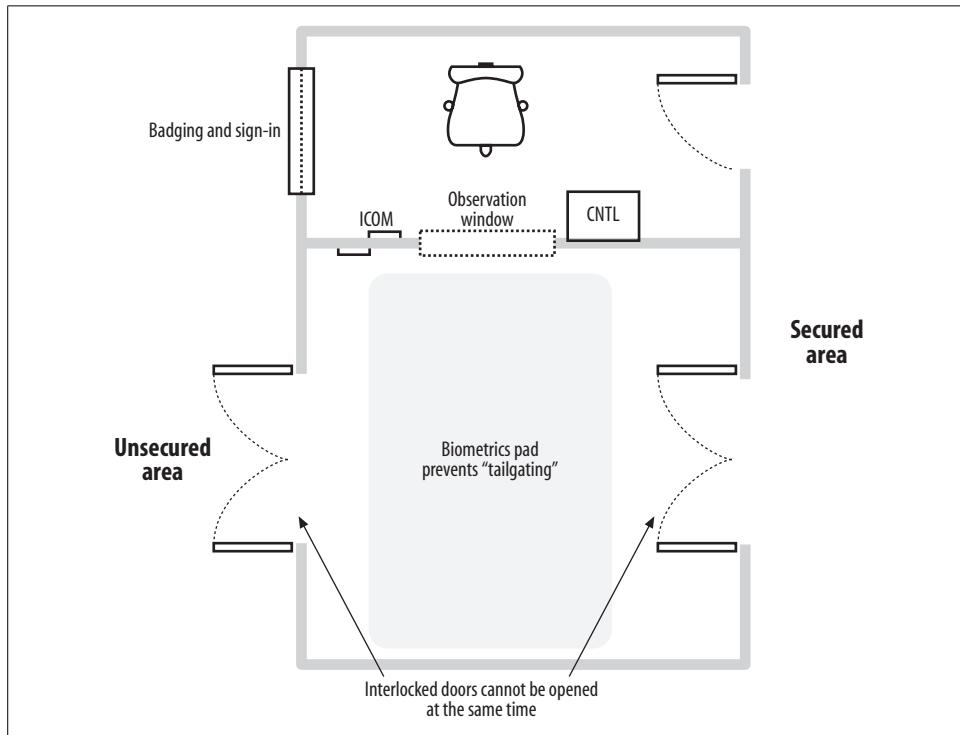


Figure 9-1. A mantrap

Types of Locks

In addition to locking up buildings and computer rooms, you can also secure your computer, your network, your disk drives, and your disks.

Here are two lock examples:

Equipment locks

The simplest way to keep someone from walking out with your PC, router, switch, or other network device is simply to bolt it down. Computers, workstations, and cables may also be equipped with locks that can be unlocked only by special keys, electronic tokens, or smart cards.

Cryptographic locks

Some ultra-secure products are equipped with electronic devices known as *smart keys*. These keys are used to load initial cryptographic key information (usually supplied by a government agency) into the product. They typically have tamper-detection circuits, which erase the secure key storage if the circuit is broken.

Tokens

A token is an object that you carry to authenticate your identity.

In ancient times, a trusted courier might have carried the king's ring to a foreign kingdom to prove that he could speak for the king. Modern tokens are electronic devices, usually containing encoded information about the user who's authorized to carry it. Typically, a token is used in conjunction with another type of authentication, in a two-factor authentication system. For example, with certain types of PC security packages, you must insert an electronic, key-shaped token during login and authentication. After the system recognizes the token, it prompts you to type identifying information (e.g., ID and password) and compares your entry with the information encoded on the token. If the two match, you'll be allowed access. If they don't, you'll usually be given a few more chances. After multiple failures, you'll be locked out, and an alarm will sound. These tokens do not require user interaction; for this reason they are sometimes referred to as *passive tokens*.

Modern token systems are packaged as PCMCIA (PCCARD) or USB form factor devices that fit into your computer. Soft tokens, secure passwords that are unique to your computer, can also be installed.

In most cases, tokens are used as part of a VPN secure transmission path, which allows a portable computer to log onto a corporate network from out of the office or on the road, and implements encryption on the data that is transferred.

Challenge-Response Systems

Some public key authentication products are more sophisticated versions of electronic tokens. These can be called *active tokens*. Challenge-response systems typically use a hand-held device containing an encryption program and a key. When you try to log in, the system challenges you with a random number. You type this number into the hand-held device, which encrypts it and displays the result. Now, you

type that number into the system. The system compares the typed response with the result of its own encryption of the random number. If the two numbers match, you're allowed access.

Versions of this kind of device include *smart cards* that display a code periodically. To log into a system or enter a secure facility, you identify yourself (by typing your PIN or password), and you also type the code that's currently displayed on your smart card. The card is synchronized with network timing signals, so that a stolen sequence is valid for only a short time. Such systems offer a number of other special features. For example, the card can be designed to stop and erase its memory at the end of its programmed lifetime. If you attempt to open the card to replace batteries or change it in any way, the card is permanently disabled.

Cards: Smart and Dumb

For many years, ID badges, often with photos, have served as credentials. You must present your license to the bank teller or supermarket clerk before you're allowed to cash a check. You must flash your employee badge before the building guard allows you to enter the building. Authentication works by having someone visually match your face to your picture.

Automatic teller machine cards and certain types of credit cards use a more reliable type of matching that magnetically encodes identifying information on the card. For example, an ATM works by comparing the information on the card to the information you enter at the ATM—usually some combination of account number, PIN, and/or password—and allowing you to withdraw money only if the match is successful. Increasingly, more advanced types of cards are being used to control access to buildings, computer rooms, and computers themselves.

The typical access card is the size of a credit card. It usually contains an encoded identifying number, password, or other type of prerecorded information, often in encrypted form. Depending upon the sophistication of the system, the card may contain a large amount of additional information.

The newest types of access cards are called *smart cards*. Smart cards come in different sizes and shapes. Some look like credit cards, some look like memory sticks, and some are shaped like a watch fob. One popular version provides secure transmission information using state-of-the-art encryption algorithms, and fits on a standard key ring. A disadvantage to tokens is that they are small and easily misplaced, and they can be costly, from \$25 to over \$100 dollars. These cards contain microchips that consist of a processor, memory used to store programs and data, and some kind of user interface. Sensitive information, which typically includes the user's PIN and/or password, is kept in a secret zone of the read-only memory. This zone is encoded during manufacturing, using cryptographic techniques, and is inaccessible even to the card's owner.

The newest forms of smart cards are ultra-small modules that communicate via radio with sensors on the device or doorway to be protected. These usually contain not only the information required to access the protected area or device, but also personal information related to what you can or cannot do once you have entered or logged on. Advanced versions of these identifiers may contain health related information, which can be used if you are found unconscious or disabled, or man-down sensors, which can alert personnel if you fall or come under duress.

Unlike most of the other types of access cards that are typically used (like badges) simply to gain entry to a facility, smart cards are often used for authentication. When a user attempts to log in or enter a secure facility, the computer system may transmit information to the smart card, which performs a series of complex calculations on it and transmits the result back to the computer. If the transmitted result matches the expected result (which is possible only if you have an authentic smart card), you're allowed to enter.

Many smart cards are built to work with card readers. You insert the card in the reader. The system displays a message, and you enter your personal identifier in response. If the identifier matches the one expected, you're allowed access.

Biometrics

Every person has a set of unique physiological, behavioral, and morphological characteristics that can be examined and quantified. *Biometrics* is the use of these characteristics to provide positive personal identification. Fingerprints and signatures have been used for many years to prove an individual's identity, but individuals can be identified in many other ways as well. Computerized biometric identification systems examine a particular trait and use that information to decide whether you have the right to enter a building, unlock a secured area, or access a system.

Biometric systems are available today that examine fingerprints, handprints, retina patterns, iris patterns, voice patterns, signatures, and keystroke patterns. Devices have also been proposed for such traits as body weight, footprints, signatures, lip-prints, wrist vein patterns, brainwaves, skin oil characteristics, facial geometry, and weight/gait patterns.

Although the human body is intrinsically difficult to measure and quantify accurately, biometric devices are generally reliable (especially when sensible thresholds are established for determining, for example, how closely a signature needs to match to be called "identical"). For best results, use biometric devices in a two-factor authentication system, in conjunction with another authentication measure such as a password.

Of the devices currently on the market, only fingerprint, handprint, and retina pattern systems are properly classified as biometric systems, because they test actual physical characteristics. Voice, signature, and keystroke systems are more properly classified as behavioral systems, because they test patterns of physiology or behavior.

Types of Access Cards

Access cards are distinguished by the technologies used to encode information on them. The government publications, *Guideline on User Authentication Techniques for Computer Network Access Control* (FIPS PUB 83), and *Guideline For The Use Of Advanced Authentication Technology Alternatives* (FIPS 190) described the following types of cards in 1980 and 1994, respectively. Although some of these methods have been supplanted by newer technologies, they're all included here for historical interest:

Photo ID card

Contains a facial photograph that is checked visually by a person.

Optical-coded card

Contains a geometric array of tiny, photographically etched or laser-burned dots representing binary zeros and ones that typically encode the user's identification number. The card is laminated with a protective layer that can't be removed without destroying the data and invalidating the card.

Electric circuit card

Contains a printed circuit pattern. When inserted in a reader, the card selectively closes certain electrical circuits.

Magnetic card

Contains magnetic particles that encode the card's permanent identification number. Data can be encoded on the card, but the identifying structure of the tape itself can't be altered or copied.

Magnetic stripe card

Contains only a stripe of magnetic material, typically on one edge of the card. This technique is used by most commercial credit cards.

Metallic strip card

Contains rows of copper strips. The presence or absence of strips determines the code pattern.

Capacitance card

Contains an array of small conducting plates. The capacitance of the plates determines which are isolated and which are connected.

Passive electronic card

Contains electrically tuned circuits. The card is read using a radio frequency field, which decodes the tuned circuits to encode the unique card number.

Active electronic card

Contains electrical circuits. The card is read by an interrogation unit that examines the encoded information transmitted by the badge.

Memory and microprocessor tokens

These are fitted with both EPROM and EEPROM programmable nonvolatile memory technologies.

Radio-based card

Communicates between user and desktop computer via tiny radio frequency circuits embedded in the access card.

The typical biometric identification system obtains data from you—for example, a handprint, a retina pattern, an iris scan, or a voice pattern. It then converts that analog signal into a digital representation and compares that representation to the many “templates” stored in the system. These templates are obtained when you are originally enrolled in the system. For example, in a signature verification system, you’re required to sign your name several times to allow the system to sample traits and construct a template for later comparisons.

Biometrics have shown great promise, and in fact have long history, but they are not yet extremely popular. Users have expressed reluctance to submit to such surveillance for all but the most crucial operations. False positives and false negatives (either allowing an unauthorized person in or preventing an authorized person from entering) remain a concern. Metrics involving the allowable ratio of false positives to negatives are part of the purchase process. Potential buyers seem to be mainly deterred by cost and by fears that unauthorized users will either impersonate authorized users or somehow bypass the devices altogether. Some recent experiments with gelatin hand and fingerprints, for instance, have indicated that biometrics alone is not adequate for personnel identification.

This is not to say the technology does not have great promise. As biometric devices gain more of a foothold in the market, it’s expected that useful techniques will follow, which may make such systems desirable. One example is a program that keeps track of unsuccessful attempts to gain access and then stores the characteristics (e.g., fingerprints) of the unsuccessful intruder so that person can be tracked down at a later time.

Another feature sends a distress signal if the system determines that you (an authorized user) are being coerced into helping an unauthorized individual to gain access. To make this work, you need an agreed-upon signal. For example, with a fingerprint system, you could signal that you needed help by pressing the left index finger, not the expected right index finger, on the glass plate of the scanner. The intruder would be unlikely to notice this subtlety. The software could be programmed to allow entry, but to alert your organization’s security forces that you’d been forced to help an intruder gain entry.

There are a lot of tradeoffs associated with biometric systems. Because such systems are new and because many people just don’t like being measured, there’s quite a bit of personal resistance to using them. Although most of these methods are quite effective from a technical point of view, they may prove completely ineffective if people reject them and if they’re seen as being intrusive, time-consuming, or even dangerous.

Some biometric methods are viewed as being quite threatening. Retina identification systems tend to be the most frightening; despite reassurances, people fear that the system will run amuck and blind them. That’s quite an obstacle to acceptance! Other systems, because they’re more familiar and less threatening, don’t meet with the same degree of resistance. Signature systems, for example, are well-accepted because

people are accustomed to having their signatures verified during banking and credit card transactions.

Surveys indicate that biometric devices, in order of effectiveness, rank as follows (most secure to least secure):

- Retina pattern
- Fingerprint
- Handprint
- Voice pattern
- Keystroke pattern
- Signature

In order of social acceptance, the order is practically the opposite:

- Keystroke pattern
- Signature
- Voice pattern
- Handprint
- Fingerprint
- Retina pattern

Trade organizations for developers of biometric products are addressing technical issues associated with biometric devices and are also working on increasing public acceptance of these devices.

Retina Patterns

Everybody has a unique retinal vascular pattern. Unlike a fingerprint, the pattern of blood vessels in the retinal tissue can't be recorded or even photographed with ordinary equipment. Retina pattern verification systems examine the unique characteristics of an individual's retina and use that information to determine whether the individual should be allowed access.

A retina pattern verification system uses an infrared beam to scan your retina, measuring the intensity of light as it is reflected from different points and producing a digital profile of the blood vessel patterns in the retina. The system allows access only if your retina pattern sufficiently matches those of the retina pattern stored for you in the system. The newer systems also perform iris and pupil measurements. Hand-held devices are being developed for workstation access.

Retina systems are very reliable. Their ability to work properly is affected only by very serious injuries and a few rare diseases. They have been used successfully in national laboratories, office buildings, and prisons, but they are not well-accepted as access devices. Of all of the biometric systems, retina systems seem to be the most threatening because of the public's fear that scanners will blind or otherwise injure them.

Iris Scans

Rather than examining the retina at the back of the eye, an Iris scan looks at the colored part of the front of the eye. This is much easier to image, and can be very accurate for identification. Iris scans may provide a feasible biometric where retina scans still meet resistance.

Fingerprints

Everybody has a unique set of fingerprints. Fingerprint verification systems examine the unique characteristics of your fingerprints and use that information to determine whether or not you should be allowed access.

The use of fingerprints to identify people dates from the 1800s. In the past, manual methods were used to classify and cross-check fingerprints according to certain patterns of ridges and whorls—in particular, detailed features of the print called *minutiae*. A fingerprint may have up to 150 of these minutiae. In the late 1960s, the FBI automated its system for cross-checking fingerprints, and all fingerprint checking was converted to automated systems by 1983.

A fingerprint system works like this: you place one finger on a glass plate. Light flashes inside the machine, reflects off the fingerprint, and is captured by a scanner, which transmits the fingerprint information to the computer for analysis. The fingerprint system digitizes the ridges and other characteristics of the fingerprint and compares these characteristics against the fingerprint templates stored in the system (or, in more primitive systems, against a print on a card that you carry). The system allows access only if your fingerprint sufficiently matches the template.

The more sophisticated fingerprint verification systems also perform a three-dimensional analysis of the fingerprint including infrared mechanisms for ensuring that a pulse is present. This means that an intruder can't gain entry by presenting a mold of an authorized user's finger or, worse still, an authorized finger that's no longer attached to its owner (a particularly grisly type of hacking!).

Because fingerprinting has historically been used as a law-enforcement tool, fingerprint systems are pretty well-accepted by potential users of such systems—particularly in criminal justice organizations, in the military, in high-security organizations such as defense plants, and, increasingly, in banks. They have several disadvantages. They are slower than certain other types of biometric systems. In addition, their ability to work properly depends on the condition of the fingers being presented. Burns or other physical problems can affect the system's ability to match fingerprints, as can any substance (e.g., dust, perspiration, grease, glue) on fingers. And as mentioned, gelatin coatings can allow someone to “forge” a fingerprint, and then discard or eat the evidence once past the barrier.

Handprints

Everybody has unique handprints. Handprint or hand geometry verification systems examine the unique measurements of your hand and use that information to determine whether you should be allowed access.

With a handprint verification system, you place your hand on a reader, aligning all of your fingers along narrow grooves with glass between. A sensor beneath the plate scans the fingers, recording light intensity from an overhead light, and measuring fingers from tip to palm to within 1/10,000 of an inch. The information is digitized and compared against a handprint template stored for you in the system. The system allows access only if your handprint sufficiently matches that of the stored template.

The older handprint systems examined finger length and the thickness and curve of the webbing between fingers. The newer hand geometry systems examine a whole set of topographical characteristics, such as the depth of the skin creases in the palm.

Very few handprint systems are in use today, though the technology is pretty well-accepted because it's not considered to be as intrusive as other types of biometric systems. Handprint systems are said to be less reliable than fingerprint systems. Like fingerprint systems, their ability to work properly depends on the physical condition of the hand. Injuries, swelling, or the presence of rings, or even nail polish, on your fingers may affect the system's ability to match a handprint.

Voice Patterns

Everybody has a unique vocal and acoustic pattern. Voice verification systems examine the unique characteristics of your voice. Some systems also examine your own phonetic and linguistic patterns and use that information to determine whether you should be allowed access.

With a voice verification system, you speak a particular phrase. The system converts the acoustic strength of a speaker's voice into component frequencies and analyzes how they're distributed. The system compares your voice to a stored voiceprint. The voiceprint is a "voice signature" constructed by sampling, digitizing, and storing several repetitions of a particular phrase. The system allows access only if your voice signature sufficiently matches those of the stored voiceprint.

Voice systems are fairly well-accepted (they are viewed as being nonthreatening) in financial organizations such as banks (particularly vaults), credit card authorization centers, and certain types of ATMs. Their ability to work properly depends to some extent on the physical condition of the larynx. Respiratory diseases, injuries, stress, and background noises may affect the system's ability to match a voiceprint.

Keystrokes

Everybody has a unique pattern or rhythm of typing. Keystroke verification systems examine the unique characteristics of your keystrokes (your own electronic signature) and use that information to determine whether you should be allowed access.

With a keystroke system, you must type until the system can construct a reliable template of your keyboard rhythm. Once a template is available, the system will be able to examine the speed and timing of your typing during the login process, and compare it to the keystroke template stored for you. The system will allow access only if your keystroke patterns sufficiently match those of the stored template.

Because keystroke verification may be built into the ordinary login process and doesn't require a separate verification cycle, it may eventually win wide acceptance.

Certain keystroke systems are passive systems that continuously sample your keystrokes. The goal is to determine whether in fact, the person who logged onto your system under your account (presumably that would be you) remains in that position, or whether an intruder has somehow supplanted you to gain access. Because such systems can be used to perform surveillance of your work habits (What are you typing? At what rate?), they raise privacy issues.

Signature and Writing Patterns

Everybody has a unique signature and signature-writing pattern. Signature verification systems examine the unique characteristics of your signature, and the way in which you write your signature, and use that information to determine whether you should be allowed access.

With a signature verification system, you sign your name, using a biometric pen, typically attached by a cable to a workstation. The pen, or the pad on which you write, converts your signature into a set of electrical signals that store the dynamics of the signing process (e.g., changes in pressure as you press down lightly on one stroke and more forcefully on another). The system compares the signature to a signature template stored for you. It may also analyze various timing characteristics, such as pen-in-air movements, that are unique to you and that are much more difficult to forge than the actual static signature on a page. The system allows access only if your signature and related characteristics sufficiently match those of the stored template.

Signature systems are a very well-accepted type of biometric system because people are accustomed to having their signatures scrutinized. Such systems are also much cheaper than many of the other biometric systems described in this section.

Gentle Reminder

It does not so much matter which technology is used to authenticate users to access facilities and computers if the correct steps are not taken in other areas. For example,

consider the situation where physical access to certain areas is typically denied, but after 5:00 p.m., janitors can roam freely.

Proper disposal of documents would be another classic example. As has been demonstrated from several embarrassing incidents, mere shredding is often not enough. Unless a cross-cut shredder is used, software can detect which pieces join together by noticing the perforations and patterns left by the shredder cutting wheels. Computer graphics allows researchers to manipulate the bits of paper with a mouse. Sophisticated algorithms can reassemble sentences, performing trial fits that archivists can use to detect how a torn page might be reassembled. And all of this assumes that the document was even shredded, not merely ripped or folded. Physical security must be augmented by coaching users to destroy sensitive documents (with a cross-cut shredder). And everybody must keep an eye out for dumpster divers.

Similarly, access systems that cannot easily be fooled can usually be evaded and talked around. Would it take more than acquiring a pair of coveralls in the color of an overnight delivery firm? Or wandering in with a telephone communications test set on the belt and holding a clipboard? Also, what good is it if every employee is given a background check, equipped with a fancy token and prohibited from entering certain areas, while transient employees and vendors wander freely with their pails, vacuums, and clipboards? To be effective, physical security must apply to all hands.

Summary

Physical security is based on keeping intruders out. Out of the facility, out of the network and its wiring and wireless links, out of the computers, and out of the data stores. Much of physical security can be attained by common sense precautions, similar to keeping the cash safe in a business. Other measures are more obscure, such as adhering to the standards in data cabling and facility construction, such as providing adequate telecommunications pathways and spaces, and properly securing telecommunications rooms.

An important part of physical security is making sure that the people who must enter a facility are actually who they say they are. Proper identification procedures can include requiring badges and card readers at critical points, or actually measuring some attribute of the person, such as physical weight, appearance, fingerprints, or retina patterns. This kind of biometric security is increasing in its importance, and is appearing in more facilities all the time.

CHAPTER 10

Wireless Network Security

Wireless network communications, systems that interconnect several units without using wires, has been a goal of network administrators for a long time. This is because seasoned administrators know that sometimes wires just do not go where you need them to. Further, there has always been a need for mobility in computing. Wireless provides that. The need for portability has increased dramatically in recent years, however, probably to keep up with the cellular telephone, which has granted unparalleled flexibility to wandering users.

Unfortunately, wireless is based on a shared medium, called at one time “the Ether.” Because wireless communications work by injecting electromagnetic waves into the ether, everyone who can poke an antenna into the path of a radio wave can intercept a wireless transmission.

So, the question is, how do you maintain security over a medium that is without bounds? The answer is careful encryption, limited power, and attention to antenna placement. Although the folks at the neighborhood electronics store would have you believe that all you need to do is plug a card into your computer and you can join the wireless world, if you fail to pay attention to some basics, you are entering a threatening world in which your secrets are no longer your own.

How We Got Here

Before we talk about where wireless is and where it is going, let's look at where it came from. Radio waves happen when electrons run down a wire, and the movement of current produces magnetic fields and electrostatic fields that ebb and flow with the current. Using that phenomenon to convey information is a matter of controlling the current changes and characteristics of this ebb and flow, a process called *modulation*. The names of two types of modulation can be seen on any radio dial: *amplitude modulation* (AM) increases and decreases the strength of the current to convey information. *Frequency modulation* (FM) changes the current's frequency of oscillation. Additional ways to change or modulate the currents abound. *Pulse*

modulation (PM) emits sharp bursts of current, carrying information in the timing of bursts. PM was the basis of early digital transmission systems, but today's modulation methods are much more sophisticated. Without getting too lost in details, the modulation systems used in wireless today use combinations of these three to efficiently transmit digital information. Because these systems spread their information over wide sections of the radio spectrum, they are called *spread spectrum* systems. Because they vary their operating frequency rapidly, they are also called *frequency hop* systems. To understand too much more about them would require some serious arm waving discussion at a white board, or better yet, some military experience with the systems where these techniques were first used. For the purposes of this book, it is enough to know that the modulation systems used in wireless today can pack an amazing amount of information into a low-power signal. The result has been a proliferation of IEEE 802.11-type radio systems. Nearly any laptop purchased today has wireless connectivity built in. This is great, because it allows a computer to join the local area network without first having to string a wire to it. It has a number of security issues, however, which will be discussed in this chapter.

Today's Wireless Infrastructure

Many network administrators and most home users have the belief that setting up a wireless network is a self-managing affair. *Wireless cards* are available for most computers. The thinking in many facilities today seems to be: "Throw in the wireless cards, and you are off."

This is called an *ad hoc* infrastructure. Sure, wireless will work under these conditions but to do so is to invite a security disaster. In this arrangement, units attach freely to each other, and there is no central authority to govern authentication (see Figure 10-1).

In contrast, an *infrastructure architecture* is a wireless network in which all units communicate with one unit that connects to a wired network. *Wireless access points* (APs) provide an interface between the physical network and the various computers containing wireless cards. This makes sense; after all, most people use wireless not to connect to each other, but to a LAN, or to the Internet.

Infrastructure architecture improves security and performance because administrators can monitor central login points to detect illicit access attempts. This architecture also makes network resources available to wireless users in the network, which increases the value of the connection.

Each network attachment point and the area it serves is called a *basic service set* (BSS). If it exists alone on a network, it is called an *independent service set* (ISS). Two or more basic service sets can exist on the same wired network. Traffic hands off between them depending on which has the stronger signal; this process is called *roaming*.

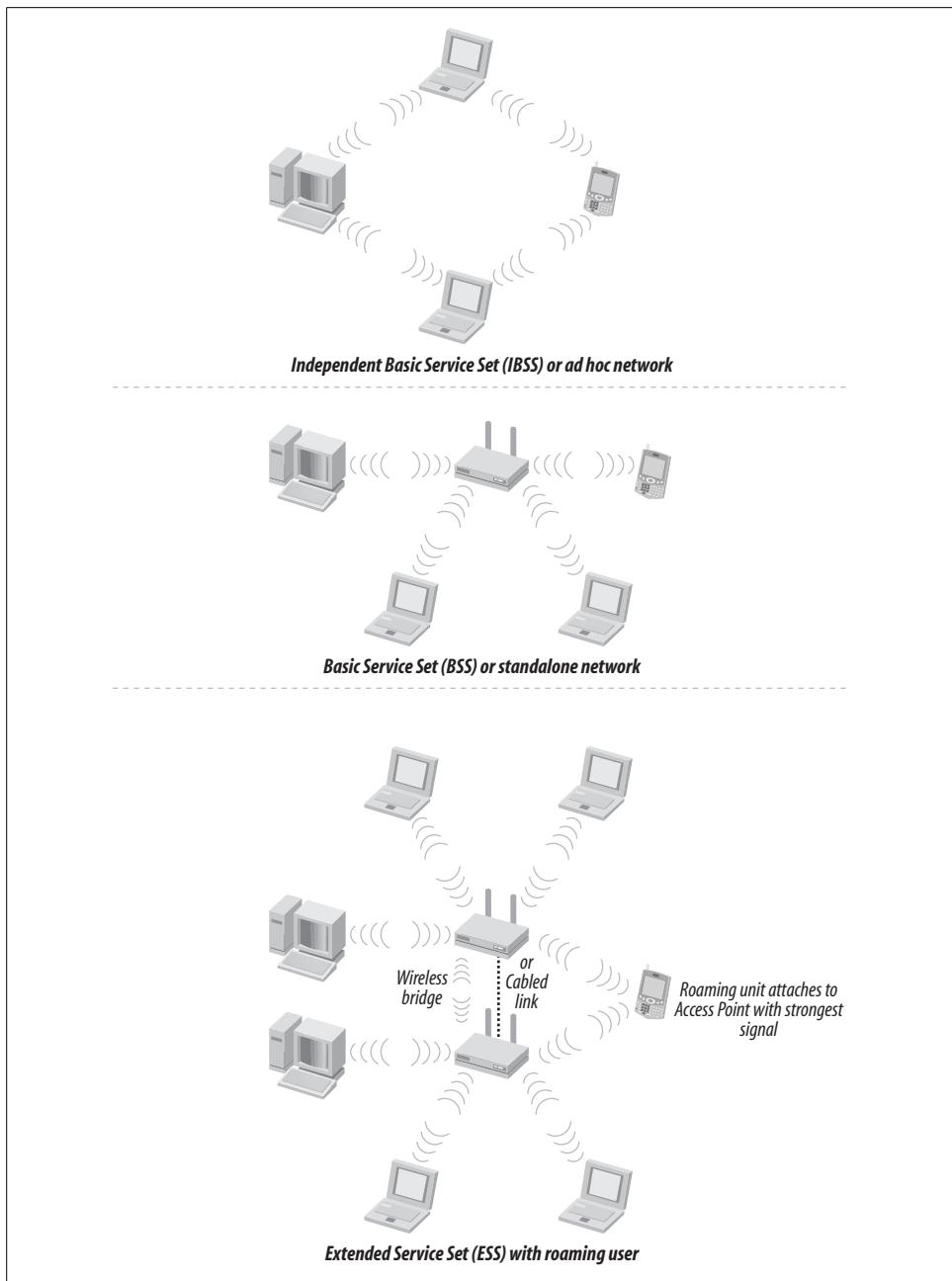


Figure 10-1. Wireless architectures

It is possible to take advantage of wireless to overcome an impassible barrier between two network segments. Used in this way, wireless equipment forms a network

bridge. Most access points will function as bridges; however, if the interconnection is between two buildings, there may be an advantage to using specialized hardware that is optimized to withstand exposure to the elements.

Home users too have pined for the advantages of a wireless world. Even something as simple as working one's laptop while watching a TV show used to require stringing a long network cable across the floor. This was a trip hazard at the least, and such a cable was often a luxury in its own right. Well-wired houses can be a rarity. The time and tools required to route wires through walls made it a daunting task. A wireless access point connected to a cable or DSL modem makes wireless highly desirable for home use.

Wireless Costs

Wireless was formerly an expensive and cumbersome solution. Until it became possible to lower its cost and increase its performance, wireless in data networking remained an expensive niche solution, justified only in a few unusual circumstances in which running cable simply was not feasible. These usually took the form of bridge links between buildings or floors.

With the adoption of the IEEE 802.11 series of standards in 1997 and subsequent updates, practical and easy wireless networks became available at reasonable cost. Wireless suddenly found a place in corporate networking and coffee shops, public places and poolsides. Anywhere you desired network computing without the necessity of running cables first, wireless hit the ground running. Although there are many forms of radio and light wave communications that can be called "wireless," wireless systems, or Wi-Fi, usually refer to those that align with the 802.11 standards.

There are real advantages to wireless. Without it, the *work area* outlet near your desk (the RJ-45 connector where you plug your computer in) is served by a consolidation point, which leads to a telecommunications room, where it ends on a patch panel before being wired into some kind of expensive equipment, such as a switch or router. This orderly structured cabling system is designed to be durable, reliable, and serviceable, but at a price of flexibility and spontaneity. It is literally built around a concept of servicing a work area, not a person. Different people may come and go, but the work area remains the same.

In contrast, wireless sort of just—well, connects. You loft a signal from your computer, and if you are within radio earshot of the *access point*, you are asked to authenticate (provide your username and password), and then you are on the air, or *attached*. You are connected at a speed of anywhere from one-half to about three times that of a 10BaseT (10 megabits per second) LAN, depending on the flavor of 802.11 used, and you can sit wherever you please and move about at will.

But wireless is fraught with dangers that come from not being tied to a cable. Unauthorized users can attempt to make connections in your name without having to set

Radio Rundown (Wireless Words)

Standards for Wi-Fi communications systems are put forth by the IEEE and licensed by local governments. The IEEE 802.11 family of equipment operates in the unlicensed portions of the 2.4 GHz industrial-scientific-medical (ISM) band and 5 GHz unlicensed national information infrastructure (UNII) radio bands.

The original wireless, IEEE 802.11, an old 1-Mbps service, has largely been superceded by 802.11b, which operates at 11 Mbps on 2.4 GHz. The newer IEEE 802.11a system operates at 54 Mbps on 5 GHz. In June 2003, the IEEE 802.11g standard was formalized. This standard put the superior 802.11a transmission scheme onto the friendlier 802.11b frequencies, allowing 54 Mbps on 2.4 GHz as well, although most 802.11g units maintain backward compatibility to 802.11b, and multiband radios are available. The IEEE 802.11n standard offers increases in range and significant increases in speed over a, b, or g. The new IEEE 802.11i standard adds security to the 802.11 series, helping to plug a serious deficiency. Here are some other versions of radio communications:

Fixed microwave

Licensed channels link at various frequencies. The low-cost, license-free 802.11x systems are a direct response to the difficulty and expense of these services, but a license carries the advantage of legal protection against interfering services.

Multi-Channel Multipoint Distribution Service (MMDS) or “wireless cable”

This repurposing of terrestrial microwave channels, including the instructional television fixed service (ITFS), is used for schools.

Local Multipoint Distribution Service (LMDS)

A wireless competitor to fiber optic to the pedestal (FTTP), LMDS is a licensed service that operates in the 28–31 GHz band to carry voice, video, and high speed data communication.

Satellite

Commercial satellite data services have existed for years, often using very small aperture terminal (VSAT) antennas, which were precursors to the small antenna satellite systems available for homes. Satellite home video services usually offer a data capability as well.

IEEE 802.15 (Bluetooth) wireless personal area network (WPAN)

This system offers low complexity and low-power short-range connectivity for personal devices and peripherals. Many systems use Bluetooth to synchronize PDAs and their desktop computers.

IEEE 802.16 wireless metropolitan area network (wireless MAN or WiMAX)

These licensed and unlicensed broadband wireless access systems operate in a variety of frequencies as an alternative to fiber optic for last mile distribution to businesses.

Infrared

Infrared offers generally low bandwidth line of sight communications for control devices, such as TV remote controllers.

Free-space laser

These systems offer lightwave-based license free communications and are an alternative to fixed microwave.

foot within your walls. If they get in on your radios, they can use up bandwidth, access files without proper identification, and potentially gain a free rein on your network and beyond. Wireless provides a modern-day version of the old problem of users installing unauthorized modems on their desktop PCs. And regardless of whether they can actually attach to your network, they can still *eavesdrop* freely without making a connection, something that is difficult, though not impossible, to do with wired networks. That is the trade off of wireless; you can get high speed and easy connection with great ease—but so can everybody else.

Too Free Access?

I was once on a job site in a large office park where a local affiliate of a large chain was replacing their main data link to the corporate offices in the East. The plan was to connect two wireless access points to a router, which connected via a T-1 link to the corporate headquarters. A firewall was employed to keep outsiders out. Because of a glitch in the service provider's programming, we could not turn on our system and provide service as planned. After an hour or so, the provider corrected the issue. We then went to apologize to the branch president, who had anticipated using his new wireless laptop card to browse the Internet over lunch. His response was a lecture in wireless security packed into one sentence. "Is there something wrong?" he asked. Although our own system was down, he had not missed a beat. There were enough carelessly secured wireless access points operating in the surrounding businesses that he never lacked connectivity. Good for browsing, but bad for those businesses.

How Wireless Works

When a magician suspends an assistant in thin air, it has a way of getting the audience's attention. There is something counterintuitive about seeing someone apparently defy the law of gravity when you spend your entire life learning to obey it.

Wireless is like that. It follows its own rules and conforms to its own logic. A top-rated world-class antenna engineer once confided to a friend of mine, a lowly Navy technician at that time, that he really didn't truly understand how antenna systems actually worked, and doubted if anybody did. However, that didn't permit him to give up and change jobs: people were depending on him. He had to stick with it, even when in his heart of hearts, he knew that much of his success was by luck.

Fortunately, while the advanced parts of wireless are sheer mystery, the basic bits are understandable enough, once you've been properly introduced. If you know this, you will be prepared to deal with an emerging round of hacks and countermeasures. Just as the script kiddies don't usually have a clue how things work, many wireless crackers will be just as in the dark. If you know what is actually taking place, you will be better able to outwit them. We will now look at wireless security starting with the business end, the radios and antennas.

Both wired and wireless network communications work on the same principles. When electrons flow through a wire, they cause the creation of electromagnetic fields around the cable, as you can see in Figure 10-2.

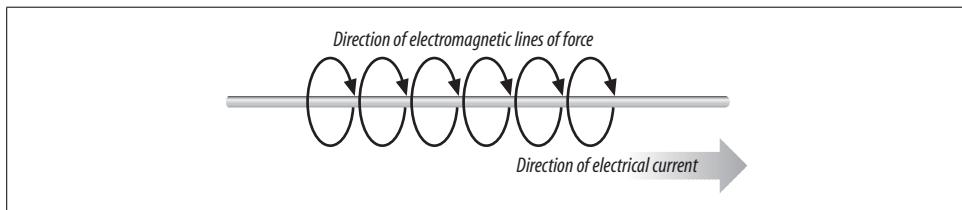


Figure 10-2. Creation of electromagnetic field around cable

If you now shoot a current flow into a cable that does not connect to anything, but simply ends, the electrons of the current flow tend to accumulate until they reach a charge equal to or exceeding whatever force propelled them down the cable in the first place.

At this point, the accumulated charge starts to swim back upstream towards the source and may even oscillate back and forth a few times until the charge ends up evenly spread over the length of the conductor. If the exciting voltage is pulsed or alternated in polarity, it becomes possible to set up a pattern of waves in the conductor as charges are pushed to the end of the wire and bounce back. As the electrons surge back and forth, the electrical and magnetic field surrounding the wire ebbs and flows as well. (See Figure 10-3.)

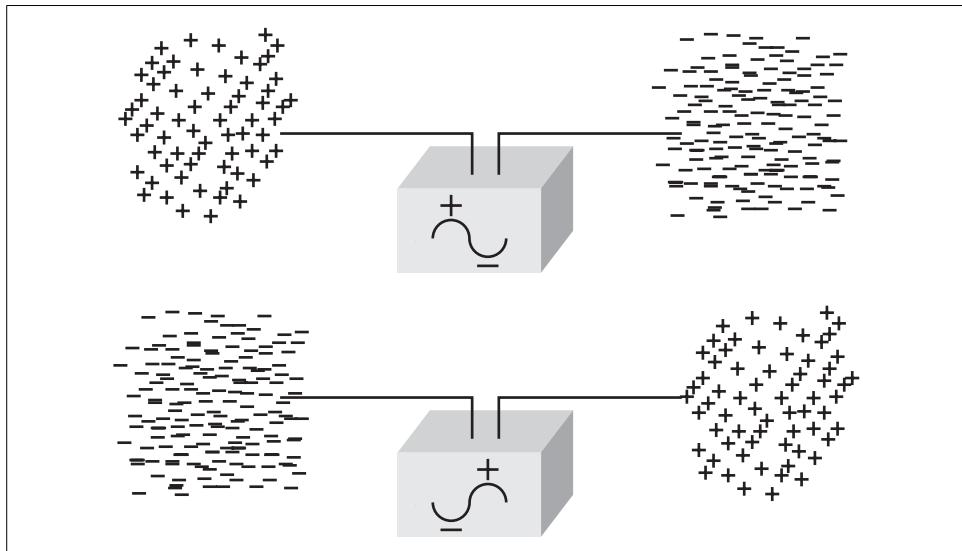


Figure 10-3. Accumulation of charge at the end of an unterminated piece of cable

If you adjust the length of the wire, which is called an *antenna*, to a length that is just right, a *launch effect* occurs (see Figure 10-4). The electric fields essentially step off the wire and travel through the fabric of space, theoretically going forever, except they decrease in power or intensity with the square of the distance from the launch point.

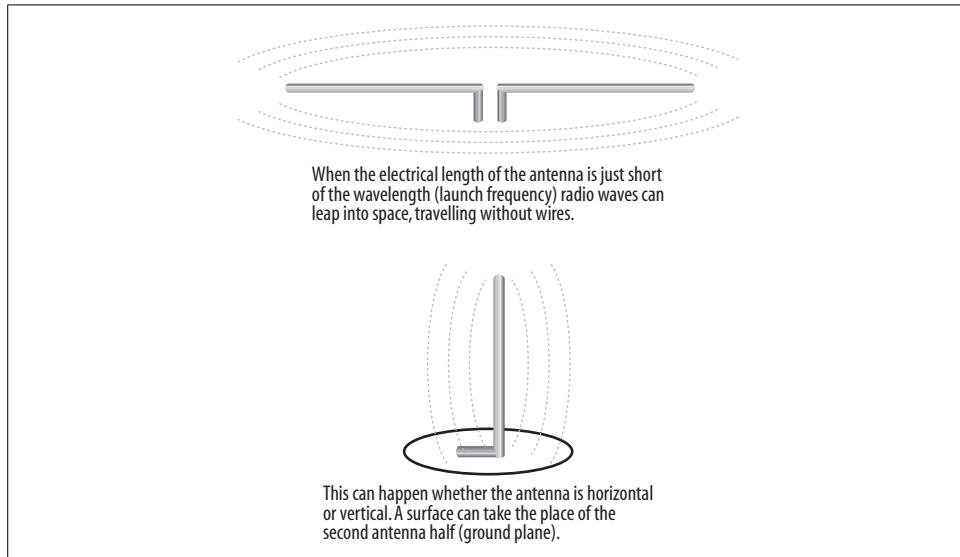


Figure 10-4. Cut to the correct length, the wire becomes an antenna and launches electromagnetic waves into space

In the period of just over a century that this effect has been recognized, people have learned much about how to control it and put it to good use. For instance, early on it was discovered that the orientation of the antenna at launch determined the orientation of the wave as it traveled through space. For convenience, these orientations are called *horizontal polarization* and *vertical polarization*, although many other combinations are possible, including circular and elliptical polarizations, and even mysterious shifting polarizations used for some radars, that appear to have the ability to hide radiation. (Radio waves are a form of nonionization radiation.)

A horizontally oriented transmitting antenna will be most effective if it is received by a horizontally oriented receive antenna, and will be least effective when received by a vertically oriented one. Understanding antenna polarity and knowing that radio waves sometimes change polarity as they pass through or bounce off of various materials and structures helps explain how sometimes shifting the antenna orientation just a little can dramatically improve the performance of an antenna system—or the ability to better reject certain interference (see Figure 10-5).

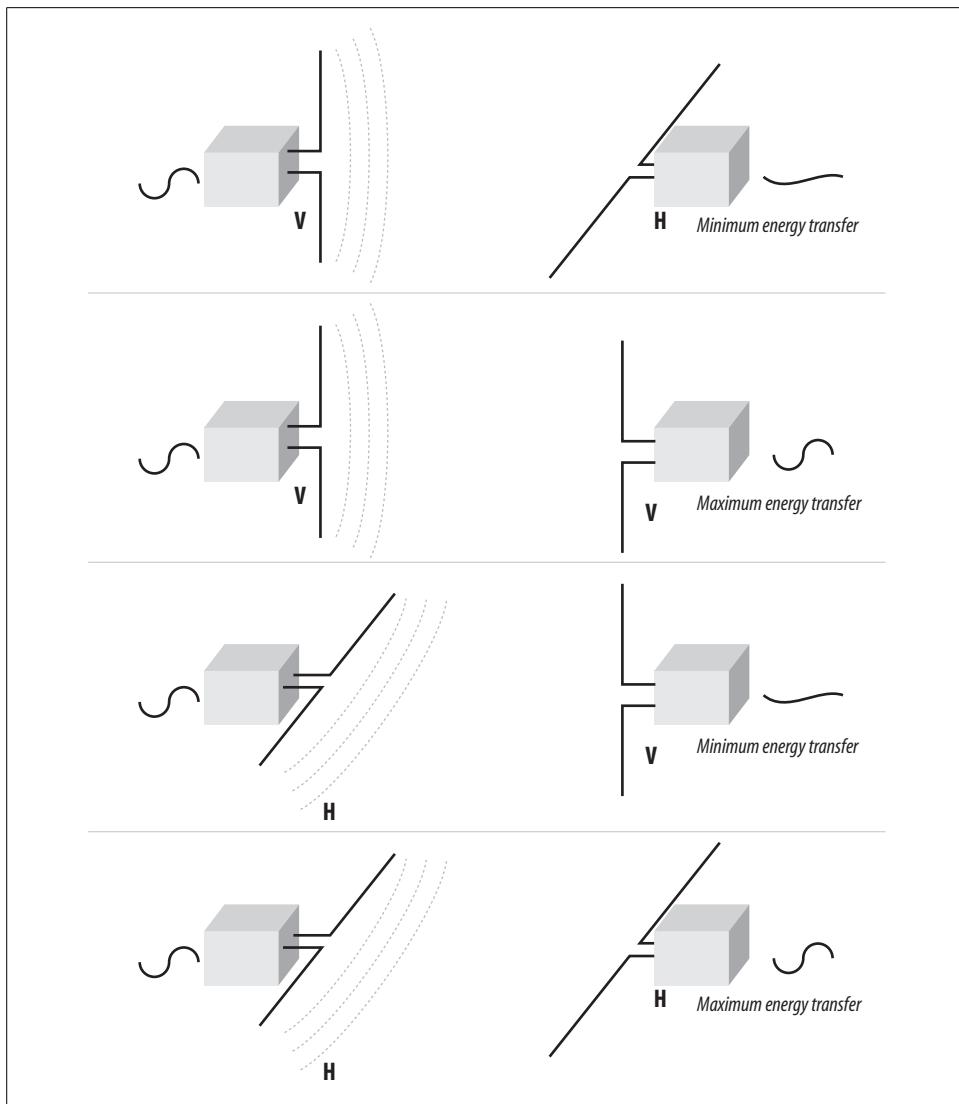


Figure 10-5. Observing correct polarity can help maximize antenna performance

Playing the Fields

One of the cardinal rules for installing a wireless system is to overcome the notion that antenna wire is simply a path for electricity. Coaxial cable is a mechanically precise system that must be gently handled if it is to perform well over time. Consider: the wire from your battery to the starter in your car is designed to move electrons. The power cord from the wall to your computer is designed to move electrons as

well. At frequencies below a hundred cycles per second or so (hertz, or Hz), the efficiency with which cables pass their signals is determined by the thickness of the cable. Literally, the more copper, the more current the wire can carry. Above a couple of hundred Hz, however, the electrons begin to travel closer to the outer edge of the cable, due to a process called *skin effect*. By the time the frequencies approach that of LANs, most of the current is traveling on the surface of the wire, not the center. In the range of microwaves and radars, (which is roughly the same frequency as 802.11x) so much of the current travels on the surface that sometimes signals are routed through pipes called *waveguides* instead of wires.

What this means is that you must be gentle while installing cables and mount them in a way that supports them without stressing them. The condition of the surface of the wire is the highway the electric fields travel over. Don't create potholes by mishandling the cable. The condition of the cable once it is in place, and the ability for it to resist damage as it sags and settles over the years, is what separates a good installation from a mediocre one. Good installers never nick a cable, never bend it tighter than four times the cable radius—even while installing it—and do not exert a lot of pressure while pulling it into place. This is even more important with the relatively low-power transmitters used in wireless.

Keeping the Waves Inside

To further explore the importance of electrical fields as they apply to security requires us to get a little more familiar with this invisible phenomenon. The patterns of the waves that emanate from antennas create a field pattern that can be measured and plotted. If we choose the length and orientation of our antenna carefully, we can make the fields that radiate from it interact in ways that shape that antenna's sensitivity and pattern of coverage.

For instance, if we energize a simple vertical antenna, we create a pattern of energy that extends around it like a doughnut. (Well, maybe a flattened doughnut.) Interestingly, if we lengthen the vertical antenna, we flatten the doughnut some more; this is because the fields tend to interact (see Figure 10-6).

Besides just squishing doughnuts, this interaction can be used to make an antenna directional. Let's add a second vertical antenna of identical dimension and distance from the first (that is, we will keep the wires to each antenna the same length). If we space the two antennas at some multiple of one wavelength from each other (wavelength is the physical length of the one complete cycle of our fields), something interesting happens. The waves from the first antenna will arrive at the second antenna in phase, that is, at the same part of the electrical cycle. For this reason they will add together in their travels, nearly doubling the strength along the line of the two antennas. Perpendicular to that line, however, the waves conflict with each other, nulling out the signal. The resulting antenna pattern, therefore, is highly directional (see

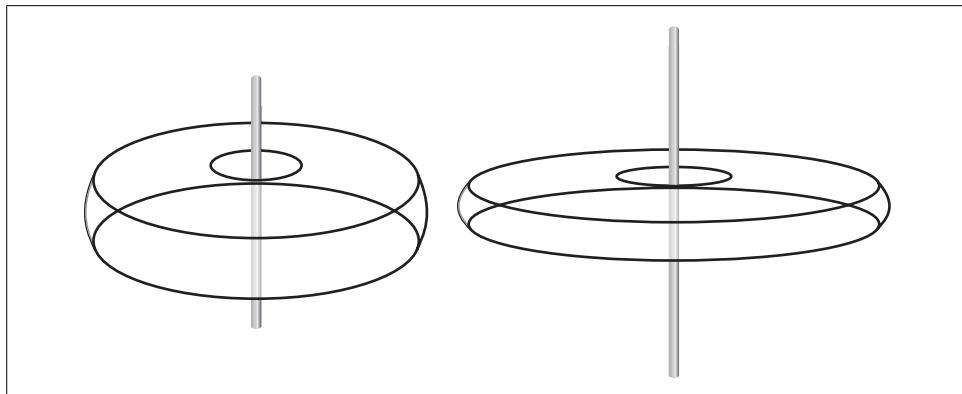


Figure 10-6. A vertical antenna radiation pattern where the doughnut's width depends on the antenna length

Figure 10-7). For the record, it doubles the strength along the main axis, providing an apparent six-decibel (6 db) gain, or roughly four times the power.

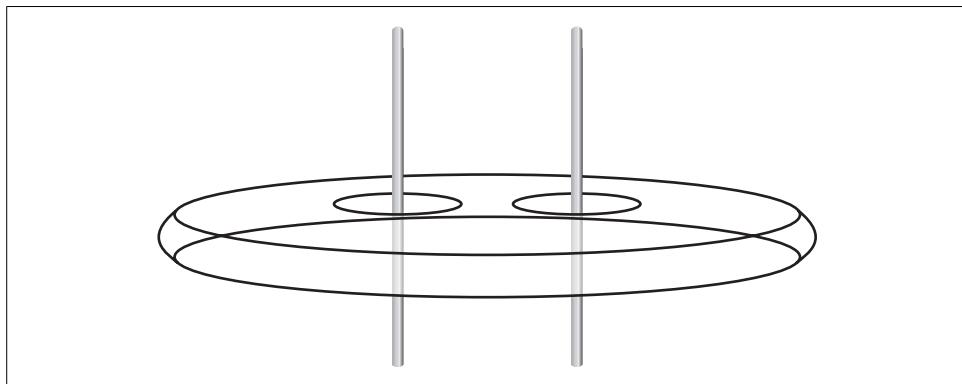


Figure 10-7. Spacing two vertical antennas one wavelength apart provides a signal gain along the line of the antennas

On the other hand, if we space the antennas at some multiple of a half wavelength apart, energy from one antenna will arrive at the other antenna out of phase. Instead of adding, the signals in the line of the two antennas cancel. The signals are increased, however, along the line formed between the antennas. The resulting pattern is wider than the pattern at the full-wave spacing, but not as long, because the antennas interact over a broader area. This half wave separation forms a pattern that is about half as strong, or about 3 db.

We can back-stop the two vertical antennas with a large piece of metal to form a panel antenna. This antenna uses the backing plate to reflect some of the energy back toward the two antennas and also prevents energy from passing through the barrier presented by the plate. The result is a pattern that has a large forward pattern, but

has a dead spot behind. This kind of antenna would probably produce a gain of about 5 db. Panel antennas (see Figure 10-8) are usually wall mounted, and are used to project a signal forward into the work area, not backward into the parking lot, where it can be intercepted by sneaks.

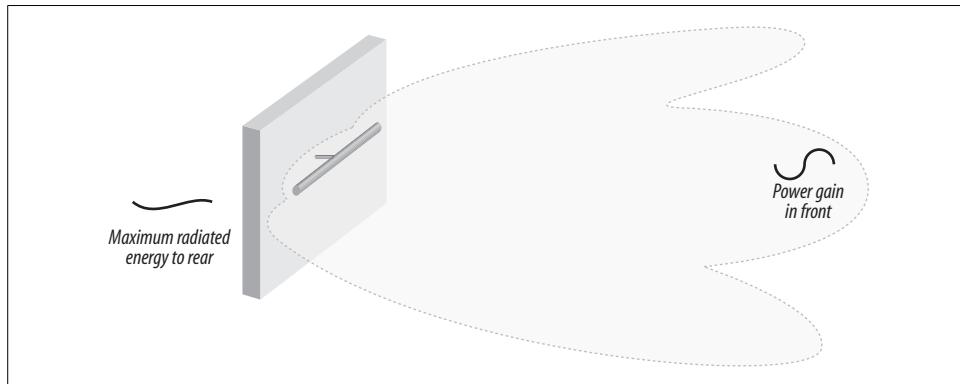


Figure 10-8. A panel antenna offers a wide, compressed unidirectional pattern

Adding an extra antenna element behind the radiating element that is slightly longer than the radiator creates a *reflector*. Adding elements in front of the radiator that are slightly shorter than the radiator creates a *director*. A reflector and one or more directors creates a very directional antenna called a *Yagi*, after the last name of its inventor. Yagis are very popular in wireless today. The gain of a Yagi increases with the number of elements, but 15 dB is not uncommon.

Finally, placing the radiator in front of a parabolic dish creates an antenna called a *parabolic reflector* or *dish*. The dish antenna is likely the most highly directional antenna available today, with the exception of antennas designed specifically for scanning and hacking. A dish can produce a gain of about 21 dB.

Using the manufacturer's data sheets (see Figure 10-9), it is possible to tailor an antenna system that will fully cover or illuminate the desired areas of a floor or department, while minimizing the amount of energy leaving the building.

What Is This dB Stuff?

The theoretically plain vanilla, no-padded-seat, economy class antenna is the simple dipole. It is considered to add no increase to the power of the transmitter, hence it is a *unity gain* antenna. Unfortunately, it is easier to mathematically create a unity gain antenna than it is to construct one, so this antenna is usually theoretical, and it is called an *isotropic radiator*. This is a useful concept in wireless, because if we build a real antenna and measure its performance, we can express its performance and radiation pattern in terms of the amount of power higher or lower than the theoretical

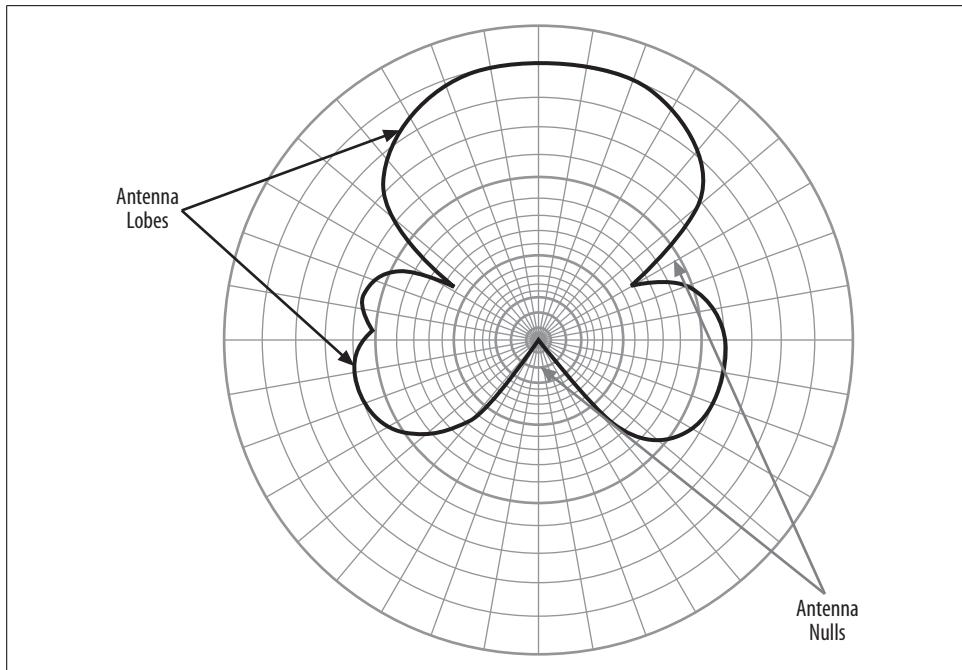


Figure 10-9. Antenna data sheets include charts that outline the antenna pattern

antenna. This information is usually plotted on some kind of graph that shows where the signal strength is greatest and weakest.

One decibel (dB) is a tenth of the power of whatever you start with, either added or subtracted. Because of the way things multiply out, each 3 dB increase in power is a doubling of power with respect to the isotropic antenna (technically 3dBi , where i stands for isotropic). Thus a dish antenna offering 21 dBi in the zone along its focus offers over a hundred times the power of the simple isotropic antenna. This multiplying factor is called *antenna gain*. A Pringles can Wi-Fi antenna (that's right, an antenna manufactured out of an empty potato chip container), if kept dry, will have a gain of 8–12 dBi.

Why Does All This Matter?

Wireless is currently in a honeymoon period. People are throwing access points on the air with abandon, showing little regard for careful design. An emerging tenet of wireless administration, however, is that you should not illuminate areas you don't intend to service. Splashed-over signal into the parking lot creates an inadvertent hot zone that attackers can take advantage of. Also, it is an emerging tenet that you should not accept logins from signals that are lower than a certain power level (as would occur if you log in from the park bench across the street) or that cannot operate at a link speed below a certain threshold (as would occur if you were an outsider

trying to log in from a distance, say the parking lot). Carefully choosing the type of antennas to use and planning their physical layout so as to not splash out of bounds, will keep the radio energy where it is needed, minimizing interference from competing sources. This forms a first line of defense against external attackers seeking to eavesdrop or join in a network where they are uninvited. More about this in a moment.

Encouraging Diversity

There is one more principle that you should know about because it shows up over and over again in wireless. The electrical field of a radio signal, which is called the *electrostatic vector*, pulses alternately strong and weak. This is an artifact of the flow of charges that lead to its creation and launch into space from its antenna. At a rate equal to its frequency, the electrical vector starts out positive, moves through zero (at the zero crossing), goes negative, and returns to positive. At the same time, there is an electromagnetic field, or vector, which is a result of the antenna's moving charge (remember, as current flows, it produces a magnetic field). This flows in a manner just opposite to the electrostatic, so that when the electrostatic vector is the strongest, the electromagnetic is the weakest, and vice versa. Let's ignore this electromagnetic field for present, but it will come up again when we discuss advanced systems of eavesdropping.

Think for a moment about receiving a signal. Signals bounce as they travel, a condition called *multipath*. It is possible that any two paths of propagation will overlap, and the distances between the paths will be different enough that the signals null each other out (similar to what happens in a twisted pair cable). On the other hand, about a quarter wavelength away, the two signals are likely to be in phase, making the signal at that point twice as strong. If we place two antennas approximately one quarter wavelength from each other, the signal received at one of them will be at its peak energy while at the other it will be at zero. If there was only one antenna, and it was located at the null point, it might not be able to detect the signal at all.

Because of the extremely small wavelengths of the frequencies we are dealing with (they are after all *microwaves*) spacing the antennas about four inches apart usually ensures that one or the other or both will always be in signal, and sure enough, many access points use dual antennas at about this spacing. In the roving environment of wireless, the chances that a multipath will occur as the users move with respect to each other is just about certain. Using two or more antennas spaced a quarter wavelength apart to avoid losing the connection is called *diversity reception*.

Physical Layer Wireless Attacks

The nature of wireless networks and their infrastructure lays the groundwork for a set of attacks that can be mystifying because they operate on different principles than

many network administrators are used to dealing with. Each has a defense, and if it is left undefended, it becomes a future watch point for security administrators.

Hardening Wireless Access Points

Of course, the first attack is no mystery: steal the access point. I have seen access points mounted on walls at just above head height, just below the suspended ceiling. In many cases, they mount by two screws that click in to the back of the unit. If someone removes the unit, she creates a denial of service attack for any users that normally associate with that point.

This attack is effective because, for whatever reason, network administrators are at the time of this writing failing to take seriously the concepts of structured cabling as they apply to wireless. No one interested in maintaining their employment would leave a server, switch, or hub, in an exposed physical location. To do so defies the ANSI/TIA/EIA standard 569A regarding telecommunications pathways and spaces. This standard states clearly that telecommunications equipment rooms (formerly called wiring closets) are to be equipped with doors that lock. Cable runs are also in the preview of paths and spaces, so keeping the wires out of harm's way makes sense; the same warning applies to access points.

A wireless installation that can be effective involves mounting a locking enclosure from the ceiling, and suspending it to align with the suspended ceiling. The cover to the enclosure, usually hinged to allow maintenance access, should be equipped with holes that allow the antennas to protrude down into the work area.

The cost of the enclosure shown in Figure 10-10 is likely several times the price of most access points available from consumer electronics stores. Add the cost of the installation (the enclosure must be suspended from the deck above the suspended ceiling, similar to a large fluorescent light fixture), and possibly the cost of an electrician to bring power to it, and it may become an overwhelming proposition for all but the most serious installations. However, it is often feasible to mount the access point on a wall *just above* the suspended ceiling. In this way the access point is secured by obscurity. Those who remember back to the days of ThickNet, which also ran above suspended ceilings, can guess what comes next. Do not allow service technicians to remove the suspended ceiling tiles with dirty hands, or at least make sure they lift them with the knuckles instead of the fingers. If not, soon, the location of the access point will become obvious by the accumulated fingerprints.

By the way, do not consider borrowing some electricity from a nearby fluorescent light. Most modern buildings operate their lights off a higher voltage circuit than what serves the electrical outlets. The higher voltage is more efficient for lighting, but could be disasterous for equipment. Electricians are paid to deal with things that spark. Use them.



Figure 10-10. A reasonably secure wireless access point that conforms to network wiring standards

Those who wish to disrupt your network, however, know that denying power to an access point is an excellent attack, one that will just about guarantee that the wireless part of the network will shut down. In most buildings, the servers and switches are, or should be, on UPS or backup power. Any laptops attached to the access points will be capable of operating on batteries. The weak link will be the access point, and a failure to protect it may indicate that you did not do enough planning for contingencies.

Securely providing power to an access point is actually a simple matter. There are several products that feature the ability to use Power over Ethernet. PoE works because there are four pairs of wires in an Ethernet cable, but only two are required for most communications at the speed that will serve an access point. The unused pairs carry power to the AP.

Back in the telecommunications room, either the switch feeding the AP is equipped to provide PoE, or the cable is run through a small device that inserts power on the cable. Devices in the telecommunications rooms have the opportunity to use protected power, that is, power that will continue after normal power is gone. This has the potential of making the wireless portion of the network even more robust than the wired section in some cases, because the desktop PCs and possibly even the telephones are likely to use unprotected power.

An alternative approach may be one of several systems that put Ethernet over power lines. The advantage is that two wiring systems, one for data and one for power, are not required. The security aspects of this system are not yet fully explored. In theory, the signals could leave the home or building and be available for interception.

The Tie That Binds

Although stealing an access point is a feasible attack, it leaves the attacker with the problem of having to carry off the unit, and then to dispose of it. It might be attractive to collect various units, but as long as they exist, they are evidence. A more reasonable attack is to simply cut the wires. If the access point is mounted on a wall in a hallway, this can be accomplished with any cutting tool. If PoE is employed, the attacker will likely use a pair of insulated grip wire cutters the second time he tries it because most systems use -48 volts, a telephone company standard.

Conduit or a flexible metallic “whip” can prevent an assault on cabling; however, installation may be costly and, to be truly effective, you may need to modify the case of the access point. A simpler solution again may be to mount the access point above the suspended ceiling. In this case, PoE is highly desirable because appropriate electrical outlets may be hard to find up there. Many facilities run all of their telecommunications wiring above ceilings, which leads to some important considerations:

- If the area above the ceiling is used as an air return for the HVAC system, that space is known as a *plenum*. Electrical codes require the use of plenum-rated cables, which are jacketed with materials that will not give off poisonous gas when heated by a fire. These gases could threaten people in other areas of the building because of the shared air handling.
- If you need to get through a wall that extends from the floor to the ceiling, don’t just poke a hole: it could be a firewall. A firewall is designed to seal off smoke and fire for a specified period, giving rescue and salvage workers a chance to work in unaffected areas. Poking a hole in that wall may violate its fire rating, and fire marshals and building inspectors get all excited about that, as do building owners, because it may have insurance ramifications. If you absolutely must go through that wall, look for an existing penetration and try to join it without disturbing the packing and putties that are part of a firewall penetration. Contact a telecommunications wiring specialist if there is any doubt or if you must unavoidably make a penetration.
- Once it was common to lay cables on top of the suspended ceiling grid. Standards no longer allow this because the metal of the grid can bend the cables tighter than their minimum bend radius and affect performance. ANSI/TIA/EIA-569-A specifies that cable above suspended ceilings must be supported using an independent system of approved devices. That means you cannot use nylon cable ties to suspend network cables on the wires that hang the lights or the ceiling grid, or tie up to a pipe or electrical conduit. But I won’t tell.

On the chance that there is no way to avoid hanging an access point on a wall in an exposed area, at least take this simple precaution: run the wire from the telecommunications room to the wall, and terminate it in a wall-mounted jack. Then, use a short patch cord to go from the jack to the access point. This way, if an attacker tries to damage your network by cutting the cable, you can easily replace the patch cord with one of the spares that you keep in your desk drawer. The installation will then adhere to standards. It is also far easier than trying to install a modular 8-pin connector while perched on a stepladder.

Sophisticated Physical Layer Attacks

A more sophisticated physical-layer denial of service attack involves disturbing the medium, which in this case are radio waves. The classic attack is to overwhelm the receiver with noise, a process called *jamming*. Any generator of radio frequencies (RFs) in the 2.4-GHz region for 802.11b and 802.11g, and the 5-GHz region for 802.11a, will suffice. As many surprised owners can tell you, DECT phones, Bluetooth devices, and cordless telephones, can (all in some cases) present jamming signals.

A steady jamming signal would be fairly easy to track down using a directional antenna and a laptop set up to do a site survey. This would be an effective attack, because it would require a sophisticated user to track down the jamming source. Much more devious and hard to trace, however, would be an attack that misuses some principle of the medium itself, for instance, the diversity reception system that is designed to prevent fading as users change their position with respect to each other. As explained previously, due to the wavelengths and the interference patterns involved, sometimes one antenna will have all of the signal, and the other will have none. However, most of the time, both antennas will have a portion of the signal. An effective DoS attack, then, would switch signals from one antenna to the other by introducing one that is stronger, yet invalid, on one of the diversity receive antenna pairs. This signal would not need to take place full time, anything that gates the signal away from where it is supposed to be periodically enough to cause the receiver to drop the link will suffice. This would be enough to cause missed packets, to necessitate retransmissions to replace them, and in general to plug up the network. And it would be very difficult to track down because it would resemble an interference source.

An example of such interference that causes failure in wireless networks occurred in late 2003 and from a surprising source. Some versions of a Centrino chipset contained a wireless adapter specifically designed to facilitate Wi-Fi communications. This chipset may somehow have interfered with a popular chip family used in many wireless access points. This caused many wireless access point owners to have to either upgrade their BIOSs or retire their access points and switch to a different brand.

Any DoS attack based on mimicking interference is a cause of concern. The attack signatures can be almost invisible to all but the most sophisticated detection systems. Worse, the intruding RF does not have to be locally produced but can be transmitted toward the affected network using a directional antenna or perhaps some form of

external amplification, neither too hard to purchase. The advantage of the latter course to an attacker is that it would be much easier to camouflage.

While such a chirping jamming signal eventually can be tracked down, it is not hard to imagine such an attack being used to shift traffic from wireless to wired forms, possibly to enable a secondary exploit of some kind. There is precedent to this. Immediately prior to the Normandy invasion, allied signals and intelligence forces did all that was possible to shut down radio communications among occupation forces, shifting it to wired means, which had been compromised by wiretaps and sabotage. This prevented the command infrastructure from shifting forces from diversionary targets in the north to the actual fronts, giving the allied advance time to establish a beachhead and land resources.

The sure cure for such attacks is based on a two-pronged defense. First, control the pathways and spaces. Unknown persons with stepladders should politely be asked to account for themselves. This is a basic principle of physical security. Second, monitor the network. Many successful administrators keep an eye out for unauthorized *rogue access points* (access points installed by endusers without permission) by using wireless monitoring programs such as NetStumbler or one of several mapping programs designed for the purpose.

Forced Degradation Attacks

In a large wireless installation, users can move from access point to access point, seamlessly being handed off from one BSS (cell) to another. These cells are spaced close enough together that there is some overlap, which insures that the user never experiences a gap in service. In order to achieve this overlapping coverage, access points are set to one of several patterns of alternating channels. Three channels are enough to cover almost any size building or campus, and channel patterns are well known.

However, the Wi-Fi service is unlicensed, which means that if there is interference, it is up to you to work it out with the interferer. There is no governmental recourse, as exists with licensed services. If more than one organization inhabits a given building, frequency coordination is left to the organizations themselves. This assumes that all involved parties are cooperative and communicative.

The government rightly refuses to regulate Wi-Fi, not that it does not need doing, but that it would take tremendous resources to do so. Years ago, regulators removed a portion of the Amateur radio band and created the Citizens band radio service, so that the common person could use CBs to communicate short distances and from car to base. Unfortunately, the band that was selected, 29 MHz, or 11 meters, was able to skip through the ionosphere, and long distance communication was popular. Operators who modified their equipment to increase its power created great interference. The original 23-channel allotment was not adequate to accommodate all users; it was then increased to 40 channels by taking spectrum away from another service. These additional channels did not help much, and even today CB can often be complete bedlam.

The government got it better with the Family Radio service: its frequency does not skip, and the FM modulation system it uses does not interfere with others. Instead, one frequency wins one over another. WiFi similarly does not travel further than line of sight (about 23 miles because the earth curves, and it becomes impossible to see the receiving station, even if on a tower). For this reason, illicit power enhancers will not provide greater range. They may, illicitly, help fill out a building that has a dark spot, perhaps, but will never make the system into a long-range communication medium. Extra power and channel assignments to increase throughput generally serve only to interfere with neighbor users. If the interference is intentional, it becomes a denial of service attack via jamming.

Mother Nature can, of course, provide degradation, although it may not be appropriate to call them attacks. Any atmospheric phenomenon that puts water in the air has the potential to disrupt microwave communications. Snow, rain, hail, and sleet, for instance, can cause *rain fade*. You may have noticed this if you use a satellite television system and have observed changes in the signal that are related to the weather. Wind and ice loading can also temporarily bend an antenna off axis or out of alignment. Lightning can cause momentary interference as well as lasting damage. Even the sun is a powerful generator of radio waves, and can cause interference, particularly if it rises or sets in the path of a wireless bridge or microwave link.

Man-made interference can also present a problem. Airport radar systems can create interferences, as can high-powered transmitters of any kind that can react inside of the Wi-Fi receive and transmit electronics, setting up spurious signals. And of course, electromagnetic pulse (EMP), the burst of radio frequency energy that accompanies nuclear explosions, or electrochemical or electromechanical devices that simulate EMP, can destroy the electronics of any unhardened radio system.

Eavesdropping Attacks

Most of the concern about wireless is currently around eavesdropping. Most wired communications systems are considered to be *bound*, that is the media is contained in a controlled manner, such as a conduit. This applies to wire and fiber alike. *Unbound* media, on the other hand, are not restrained to conduits. Wireless, infrared, and free-space laser communication are unbound.

Unbound media are subject to *eavesdropping* attacks. Although the dictionary definition of eavesdropping is “to listen secretly to the private conversation of others,” the name developed based on the position of the eavesdropper—literally hanging from the eaves of a roof, hoping to catch a conversation snippet via a window or other opening.

Wireless eavesdroppers need not hang from roofs; the park bench across the way will do just fine. The WiFi signal that passes through walls and ends up in the parking lot is fair game to someone equipped with a sensitive receiver and a directional antenna. In fact, a culture has sprung up around locating the stray signals of others. Searching for such *hot spots* by car is called *war driving*. Do it by aircraft, and it’s *war*

flying. Do so on foot, and it is *war walking*. The idea is to travel about until a hot zone is located and then see if it is accessible. If it's an unguarded site, the war driver (walker, pilot) can make a mark on the sidewalk that resembles the old hobo code. Generally, it is two parentheses back to back, with diacritical marks indicating frequencies, available bandwidth, and security (or lack thereof). Placing such marks is often called *war chalking* (see Figure 10-11).

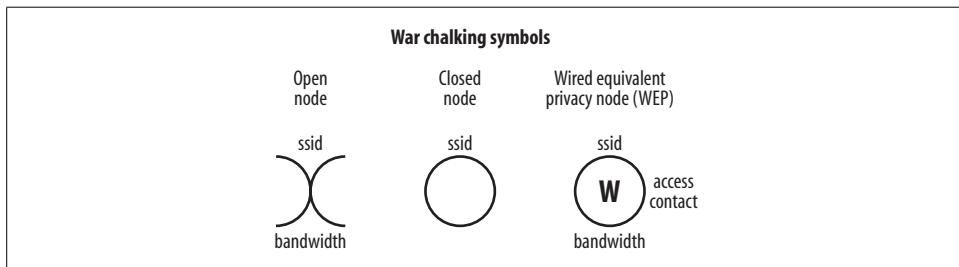


Figure 10-11. War chalking codes indicates the presence of Wi-Fi hot spots

Eavesdropping Defenses

The defense against eavesdropping is to stay off the air—not the goal of wireless communication. Clandestine operators adhere to strict schedules, transmitting only at pre-arranged times and powering down immediately after the message is sent. A wireless node that you would have to turn on each time you wished to use it would defeat the purpose of wireless, which is mobility and quick access. (It is always a good idea to turn off wireless access points when no one is using them, however, such as at night or on weekends in the case of most offices. There is no point in giving a determined attacker an unattended path into the network or in advertising the network's location.) Besides, didn't your parents tell you to turn off the lights when you left the room?

One potential downside to turning off your access point is that others who wish to install their own wireless systems over the weekend may determine the frequencies are unoccupied and set up shop. This sets up the potential for interference. Some administrators prefer to leave their access points on the air but remove as many identifying signals as possible. This allows them to show up in site surveys, but precludes attackers from gathering information that can be used against the network.

In addition to turning off the access point or suppressing ID, there are two main defenses against wireless eavesdropping attack:

- Intrusion detection systems
- Encryption systems

As they are somewhat more rare, let's discuss the intrusion detection system (IDS) first. An intrusion detection system operates by monitoring something about a given computer (host-based IDS) or given network (network IDS). Logs are kept during monitoring; such logs can analyze attacks and provide documentation for possible legal action.

The IDS monitors the network for baseline changes, that is, any events that can signal an attack, such as someone unfamiliar logging on, or traffic of one kind or another spiking up or dipping suddenly. The IDS reports each such event, usually by paging a network administrator. The network admin can turn off the threatened access unit or isolate it from the network. The admin can also contact security forces, if they are available, and track down the eavesdropper.

The disadvantage of an IDS is that without an accurate baseline, it is hard for an IDS to detect anomalous traffic. Paging an admin in the middle of the night over a false alarm, or an attack that isn't really happening, is called *false positive*. Too many false positives and the administrator is likely to disable the IDS, or change its settings and thresholds. These new settings may not pick up or relay a legitimate alarm; this situation is called a *false negative*. Obviously, neither false positives nor false negatives are desirable.

A full time method of protection that requires less fiddling uses encryption. Wi-Fi has several encryption options; unfortunately, an increasing number of them are falling to attacks by eavesdroppers.

Wired Equivalent Privacy (WEP) is the most common security protocol in older wireless networks. Its original algorithm is not very effective. Readily available software can monitor WEP transmissions until there is enough data to take advantage of a few loopholes, and thereafter will print out cleartext.

Wi-Fi Protected Access (WPA)—which is similar to Microsoft's SSN—is less vulnerable to hacking. If you can choose between WEP and WPA, use WPA. Cisco users can take advantage of the LEAP protocol between other Cisco units.

AES and the Temporal Key Integrity Protocol (TKIP) can provide the most secure wireless networking. They will likely appear as the wireless standard IEEE802.11i.

Advanced Eavesdropping Attacks

Today, eavesdropping isn't just eavesdropping. There are now advanced forms of it, and wireless may play a role in it. The easiest way to protect against encryption is to read messages before or after they are encrypted. Unfortunately, that is exactly how it is done, using techniques put forward by Dutch scientist Wim van Eck, who in 1985 demonstrated that he could pick up emissions from a VDU (video display unit) or computer screen, and reproduce them clearly several hundred meters away.

The fields that steer the beam that sweeps across your computer monitor or TV form a repetitive pattern, and like any other radio emission, they radiate and can be detected. If you monitor these fields using specialized equipment, it is possible to reinsert into the received signal the sync pulses that tell the beam when to start scanning the following line, or when to return to the top of the screen; this can be done by carefully tuning the correct oscillators. With skill, the display that you see on your screen is also visible to an operator sitting in the pizza van that has been mysteriously parked across the street for several hours. And since your screen is not likely to show encrypted data (after all, you have to be able to read it), the men in black get the cleartext just as you do.

HIPAA and Wireless—Prescription for Trouble?

I went to my doctor this summer for a checkup. When he was finished with his exam, he started clicking a stylus against a Compaq iPaq. I noticed a radio pack and an antenna on the back of his unit, so I asked “What are you doing about security for that thing?”

His answer was startling: “Security?”

This MD’s ignorance unfortunately represents the state of the knowledge of many healthcare practitioners today. Wireless is the new toy. But with wireless comes a responsibility for intense patient-record security, courtesy of the Health Insurance Portability and Accountability Act of 1996.

HIPAA created strong requirements for those who keep and transmit private patient records. The cavalier attitude my MD demonstrated can have big consequences, and the penalties for intentional violation are stiff. Clearly, decisions about the deployment of wireless technologies must take into account the implications of HIPAA security rules, which were issued in final form in February 2003.

Any electronic transmission of protected health information (PHI) comes under HIPAA auspices, and the watchwords are privacy and security. Practitioners must be sure that electronic communications are encrypted and that electronic devices are secure—that is, protected from loss, theft or unauthorized use.

There is a way to deal with this situation and it needs an acronym: Telecommunications Electronics Material Protected from Emanating Spurious Transmissions, or TEMPEST. TEMPEST is a set of standards that determine whether or not a device produces emanations that can be detected and decoded. Almost every device produces an electromagnetic signature that can be decoded, so when dealing with classified materials, TEMPEST certification becomes important. For more information, see Appendix B.

At the cutting edge of wireless weirdness is the idea that even if the field coming from a device is extremely small, to the point that the pizza truck guys cannot make it out, those same emanations may be able to exert an influence on a local source of emission, such as a radio transmitter. The transmitter signal may be detectable at a distance, hence the subtle influences from the signal you wish to eavesdrop may also be recoverable. There is a possibility that the electrostatic and electromagnetic vectors of a radio wave mentioned previously may contribute to this effect. Needless to say, a building with a floor full of wireless devices is a radio frequency cornucopia waiting to be sniffed by those who know what to look for.

Rogue Access Points

A much more down-to-earth threat for wireless networks is the same type of threat that faced networks back in the days of modems. Users would drag in a modem from home, hook it up unbeknownst to the network administrator, and proceed to create massive gashes in the security infrastructure.

In the same vein, *rogue access points* are extremely easy to install. Some are small enough to plug into a USB port. The defense against such installations is to monitor the environment, looking for unknown stations that attempt to attach to your network. Locating the stray can be a challenge but, generally speaking, a highly directional antenna can be used either with a laptop or with a handheld scanner designed especially for network maintenance (see Figure 10-12).



Figure 10-12. Wireless network maintenance tools, such as Yellowjacket, can guard against rogue access points

The culprit may well be you. As embarrassing as it sounds, it's not unheard of for a network administrator to set up a wireless device in a conference room for a meeting, and then to forget about it.

80/20 Rules

Nothing makes a wireless network completely secure, but most networks can be tightened enough that attackers move to easier prey:

Cloak

Don't broadcast service set identifiers (SSIDs). SSIDs, the 32-character identifiers in WLAN packet headers, can be easily sniffed, and can function as a sort of password. Turn them off. Legitimate users will still find the access point because they know its name. Always change the default user names and passwords that come in equipment from the factory. Hackers know these by heart.

Stealth

Don't hand out a map of your network. Make it harder to navigate or to deduce the layout of the network by avoiding common names for network items exposed to the outside. Only trusted souls need to know what goes where.

Watch your channel plan

Most facilities use a three-channel plan to keep geographically overlapped access points from stepping on each other's radiations. An attacker trying to set up a man-in-the-middle exploit will have to use off-channels. If you detect these, something is afoot.

A time of your choosing

If your office works 9-to-5, don't run your wireless 24/7. Shut down during off hours. You can also control access via signal strength and speed. Signals decline with distance, so users who are in the parking lot instead of the lobby sound farther away. If users do not exceed a certain threshold for strength, cut them off. If a user is struggling to keep up, it may be because they are farther away. Don't create a network security hole while trying to make up for a problematic network design or poor equipment.

Use serious security

Going without security is foolish. WEP is simply not adequate, because a fast network will give out enough packets for a crack in the time it takes to change a tire. Use WPA; use LEAP (look first); use TKIP or AES. Stay tuned for vulnerability notices.

Monitor, monitor

The most brilliant attackers will be able to get past your defenses, no matter what you do. Monitor for the attacks you cannot defend against. Your organization has a lobby ambassador to greet guests, and to call the cops if they act rowdy or threatening. Use an intrusion detection system to monitor against unruly wireless visitors who probe the network or attempt to access it at strange hours. Know what access points are where. Check to make sure that no new access points materialize, and no users inadvertently relay inside traffic outside.

Summary

Radio waves, being unbounded by cable or wiring, can reach many users who otherwise would be tied to a given desk or location. However, the fact that radio waves transmit information invisibly also allows them to leak information without detection. Anyone who can put a wireless device on the air can passively monitor, or actively invade, a wireless network. Further, wireless systems are prone to interference which can deny users intended services.

Proper antenna selection will do much to keep radio signals confined to the appropriate area. Careful attention to network cabling—all wireless networks end up connecting to a wired network at some point—will also help assure security for your wireless environment. Strictly discouraged are unauthorized or rogue access points. Installing such equipment without the knowledge, and approval of, network administrators can create a tremendous security hazard, even if the installer has only the best of intentions.

PART V

Appendices

APPENDIX A

OSI Model

The Organisation Internationale de Normalisation (International Standards Organization, or ISO), which has been developing international standards since 1946, introduced the Open Systems Interconnection (OSI) model in an effort to define a conceptual model for accomplishing communication and interoperability between systems, regardless of the specific hardware or networking characteristics of the systems.

Adopted in 1984, the OSI Basic Reference Model defines a network architecture consisting of seven layers in the communications process, as shown in Figure A-1. In very rough terms, the top layer of the OSI model (layer 7) deals with the end user interface. The next three layers (layers 6, 5, and 4) define the characteristics of the systems at the two ends of the communication. The bottom three layers (layers 3, 2, and 1) define the network facilities necessary to transfer a message.

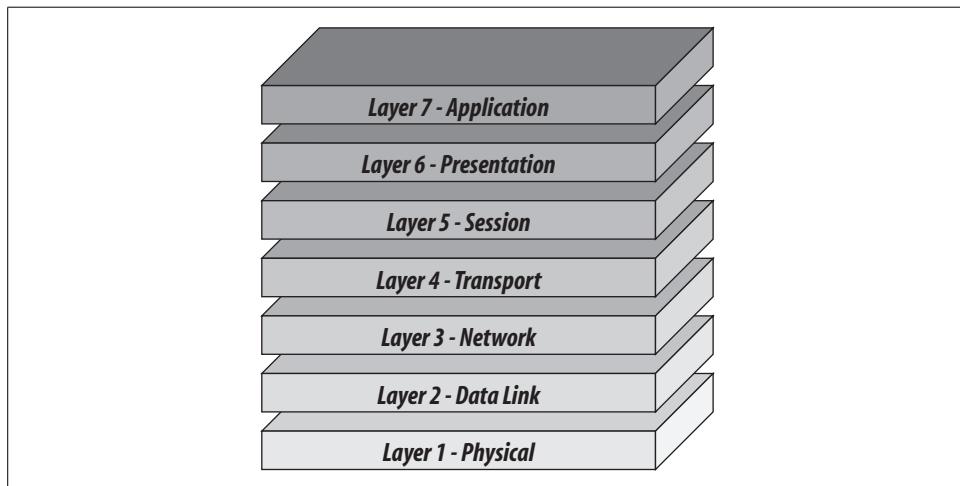


Figure A-1. Open Systems Interconnection (OSI) model

At the highest layer (the application layer), an application program in one computer communicates with an application in another computer. For example, you might send an electronic mail message to a user or to another computer. The mail program on the sending and receiving computers each resides in its own application layer.

Within the transmitting computer, the information to be transferred is passed down from one layer to the next layer, with each layer performing the work needed to prepare the data for physical transmission. Each side of the transmission performs an equivalent function. For example, a layer on one side adds routing information; the other side removes it.

Each step in the process is a type of transformation because eventually the information is transferred from one computer to the next in a very different form—for example, as radio waves, telephone line signals, or pulses of light in an optical fiber. At the other end of the communications link, the transformations have to be reversed so the other side of the communication can understand the information.

Table A-1 shows the functions—particularly those that are security-related—performed by each layer of the OSI model.

Table A-1. OSI model

Layer	Function
7 (Application)	Defines how a user accesses the network. Provides the end-user interface and services—for example, the specifics of electronic mail and file transfer.
6 (Presentation)	Prepares information for applications, often by data formatting—for example, performing blocking, data compression, and code conversion. Ensures that information is delivered in a form that the recipient can understand. May perform end-to-end encryption.
5 (Session)	Establishes a communications session between the two sides of the communication, and synchronizes the communication. Selects the necessary network services. The session layer typically deals with starting up new tasks, if necessary, and with security (for example, authentication of nodes).
4 (Transport)	Provides transparent, reliable data transmission and end-to-end transmission control, including correcting errors and reestablishing communication after a network failure. This layer is responsible for general network management functions and resource optimization.
3 (Network)	Adds routing information, and selects the appropriate facilities for transmitting the message. Controls the destination addressing and physical routing and flow of data. In packet switching networks, breaks messages into packets at the sending end and reassembles the packets into messages at the receiving end. This layer handles routing through gateways to other networks. The network layer is often X.25, CCITT's protocol for packet-switching networks.
2 (Data Link)	Formats the messages for transmission. Handles point-to-point synchronization and error control for information transmitted over the physical link. May perform link encryption.
1 (Physical)	Establishes the physical, mechanical, and electrical connection (e.g., the actual cable connection) and transmits the actual bits.

If you are involved at all with networking, you have likely already memorized this table. Certainly, you cannot expect to converse with a network engineer without hearing reference to “Layer 3 addressing,” or “Application Layer issues.” Those new

to the model can keep the layers straight using some simple mnemonics, in which the first letter of each word corresponds in order to the layers of the OSI model:

Top (Layer 7) to bottom (Layer 1):

- All People Seem To Need Data Processing.
- All Programmers Seem To Need Double Pepperoni.

Bottom (Layer 1) to top (Layer 7)

- Please Do Not Throw Sausage Pizza Away.
- Please Do Not Tell Sales People Anything.

APPENDIX B

TEMPEST

It took a while for the computer industry to realize that the emanations from most computer terminals and screens radiated fields strong enough that a sensitive receiver could decipher the screen contents. Although intuitive to someone with a background in radio transmission, many computer geeks simply didn't get it. To help them understand the nature of the threat, some researchers have developed files that create patterns of emanations from monitors that can be decoded by an AM shortwave receiver as music. Although spreadsheets may not be pretty to listen to, it sinks the message home. Emanations are a threat to security.

This capability, first discussed publicly by authors such as Wim van Eck, was exposed to the world as a technical curiosity (much to the chagrin of agencies that may have secretly used the techniques to eavesdrop or gather intelligence).

It is a particularly nasty effect from a security point of view because whatever is transmitted is whatever is on the screen, before any encryption systems are employed at the transmission side or after the text is decoded at the receive side.

The code word used to describe the U.S. government countermeasures to such risks is TEMPEST. Tempest is supposedly an acronym, but so many reliable sources report it to mean so many things that most users just leave it as TEMPEST.

The Problem of Emanations

All electronic equipment—hair dryers, typewriters, telephones, microwave ovens, personal computers, laptops, and personal digital assistants—emits electrical and electromagnetic radiation through the air or through conductors. It has long been recognized that such emanations can cause interference to radio and television reception. In addition, concerns about possible health hazards associated with emanations have led to increased shielding of monitors (see the sidebar “Hazardous to Your Health?”).

As early as the 1950s, government and industry observers became concerned about the possibility that electronic eavesdroppers could intercept emanations to decipher them, or could obtain information about the signals used inside the equipment, and use this information to reconstruct the data being processed. They speculated that eavesdroppers could breach security even some distance from the equipment.

Studies of signal interception and decoding have borne out these speculations. It turns out that with virtually no risk of detection, eavesdroppers using relatively unsophisticated equipment can intercept and decipher signals from an electronic source. Modern listening devices allow an eavesdropper to detect emissions and reproduce data streams or video screen images—for example, to read the computer display screens on the desktops in a remote building. Although opinions about the ease of interception vary, in theory a modified TV could do the job if its sweep circuits were adjusted to match common computer monitor frequencies. The components needed to perform such a penetration are garden variety. Some early computer terminals broadcast signals so strong that an ordinary television set, placed beside the terminal, could broadcast everything displayed on the terminal's screen.

The TEMPEST Program

In the late 1950s, the U.S. government established the TEMPEST program to attack the emanations problem. TEMPEST has become an umbrella name for the technology that contains or suppresses signal emanations from electronic equipment, and for the investigations and studies of these emanations. An unclassified government publication describes TEMPEST emanations as “unintentional, intelligence-bearing...signals which might disclose sensitive information transmitted, received, handled, or otherwise processed by an information processing system.”

In 1974, government and industry began to work more closely together through the Industrial TEMPEST Program (ITP). ITP was founded with the following objectives:

- Specify a TEMPEST standard that sets allowable limits on the levels of emission from electronic equipment. The idea was to state clearly how much the equipment could leak and still be acceptable.
- Outline criteria for testing equipment that, according to its vendors, meets the TEMPEST standard.
- Certify vendor equipment that successfully meets the TEMPEST standard.

The idea of ITP was to standardize TEMPEST requirements and technologies, and to encourage vendors to develop and test off-the-shelf TEMPEST equipment that the government could buy. The early TEMPEST products were typically standalone computer systems. Today, TEMPEST versions of most types of computer products have become available, and the actual certification efforts are supervised by the National Security Agency.

Hazardous to Your Health?

In the security world, fears about uncontrolled electromagnetic emissions focus on the interception and deciphering of these emissions by intruders. Of more immediate concern to most of us may be the growing evidence that emissions are physically dangerous as well. Since 1977, studies have looked at the health consequences of exposure to three types of fields: VLF (very low frequency, such as those given off by a computer's horizontal-scan frequency), ELF (extremely low frequency, such as those given off by a computer's vertical-scan frequency), and 60-Hz AC (alternating current, such as those given off by power lines and computer monitors' power transformers):

- In 1979, epidemiologists Nancy Wertheimer and Ed Leeper reported on an investigation showing that children living in Denver homes located near high-current electric wires died of cancer at twice the expected rate.
- In 1988, Kaiser Permanente researchers reported that of 1,583 case-controlled women who attended their clinics, women who worked with VDTs for more than 20 hours a week suffered miscarriages at a rate 80 percent higher than women performing similar work without VDTs.
- In 1989, Johns Hopkins epidemiologists reported that the risk of leukemia for New York Telephone Company cable splicers, who work close to power lines, was seven times greater than that of other company workers.

In December of 1990, the Environmental Protection Agency reported that environmental studies have shown "a causal link [between power lines and] EM fields and certain forms of site-specific cancer." Large-scale studies have since been undertaken, but the results have received more attention in Europe than in the United States. Nevertheless, monitor shielding has improved, to the point that they greatly limit exposure to emissions. LCD monitors may create radio frequency (RF) energy, but they produce a tiny fraction of the magnetic flux on which CRT based monitors relied. Nevertheless, there is a good chance that the safety of VDTs and related equipment may be a recurring concern in the future, along with the safety of cell phones and other radio frequency-emitting apparatus. The topic already occupies much attention among alternative medical providers, who usually advocate long walks in tree-filled areas as a curative. The inertia of the established interests, however, has kept this topic on the fringes.

In the past, TEMPEST has focused almost exclusively on the protection of classified information. As people discover that TEMPEST-type shielding can protect people as well as data, the TEMPEST technology described in this appendix may get a new lease on life, providing human safety as well as data security.

Because they're built to control electromagnetic emanations, TEMPEST products are larger, heavier, and more expensive than comparable commercial products. TEMPEST products control emanations either by shielding the signals—building a container around them so they can't emanate beyond the container—or by suppressing the signals—engineering the equipment so signals don't emanate at all. (Sometimes, a product combines both methods.)

Faraday Screens

A *shield* attenuates electromagnetic signals, conducting them to ground before they can escape. A shield, which can be as small as a cable casing or as large as an entire building, is constructed in such a way that signals can't emanate outside it. This shielding to stop the flow of electromagnetic radiation is commonly called a *Faraday screen*.

The simplest but most expensive shield approach is to install regular computer equipment in a shielded room that provides special protection against electromagnetic leaks. Smaller shields or containment devices serve the same purpose as a shielded room; shields can be constructed for computers, workstations, peripheral devices, circuit boards, and inside wiring. Modern PCs and monitors are usually coated with radiation-dampening materials that help to prevent emanations.

The containment approach to TEMPEST security resembles the steps that were once taken to protect equipment and buildings from electromagnetic pulse—which is a product of nuclear explosions in the atmosphere. This in fact may have contributed to the secrecy that surrounded TEMPEST. Explaining how to avoid an influx of nuclear EMP also explained some things about preventing electromagnetic leakage of information, and discussing secrets about one technology may have inadvertently given away some information about the other. In short strokes, electromagnet containment was tricky. Every cable and pipe that entered or left the screened portion of a facility required special treatment. But trying to curb the electromagnetic effects of the bomb was no easy task either.

Source Suppression

Some TEMPEST products use a different engineering approach. With *source suppression*, products are engineered in such a way that compromising signals are suppressed at the source. Sometimes this is done by adding confusing or spurious signals.

The source suppression approach can be technologically more difficult than the shield approach, but it's more foolproof. Its effectiveness doesn't depend on the proper use of the equipment by human beings, and it tends to be a more appropriate approach for products installed in an office environment.

Recent advances in microchip design have led to chips that do not radiate as much as their predecessors. Some of the best source suppression may take place in modern software. The hard edges of dark letters against white pages generate a great deal of square waves, which throw off electromagnetic radiation much more readily than gentle transitions. New techniques produce softer edges on letters, which decrease the sharp transitions and their radiations. Of course, a similar effect may be achieved by darkening the background. You may have seen this effect on some of the more hardcore cracker sites.

TEMPEST Standards

The National TEMPEST Standard specifies the level of emanation permitted for TEMPEST equipment. Since the original TEMPEST standard was published in the late 1950s, this standard has been revised a number of times. At one point, even accessing the standards on TEMPEST was a tricky business. The TEMPEST documents NACSIM 5100A and NACSI 5004 were classified and accessible only on need-to-know basis.

Basic TEMPEST standards were made available to the public in 1995. The TEMPEST standard, NSTISSAM TEMPEST/1-92 is now publicly available. The U.S. army acknowledges its TEMPEST testing facility, the U.S. Army Information Systems Engineering Command, at Fort Huachuca, Arizona. Information about TEMPEST certified products, which may be needed to service or provision certain government contracts, is readily available at the NSA web site: <http://www.nsa.gov/ia/industry/tempest.cfm>.

The TEMPEST Endorsement Programs (TEP) today consists of three closely related NSA programs:

- The Endorsed TEMPEST Products Program
- The Endorsed TEMPEST Test Services Program
- The Zoned Equipment Program

In the Endorsed TEMPEST Products Program, NSA provides lists of commercially developed and produced TEMPEST telecommunications equipment, which NSA has endorsed. (Formerly, NSA provided the testing, but that is now handled by certified laboratories.) The equipment lists are used by government entities and their contractors to select products for processing classified information. Separate lists are provided for products that meet different levels of compliance.

In the Zoned Equipment Program, NSA provides a listing of commercial off-the-shelf (COTS) telecommunications equipment that is not designed to meet the National Standard, but has been tested against a portion of that standard and has been assigned a “zone” of B or C. The zoning system calculates a TEMPEST *risk*, which takes into account such characteristics as geographic location (rural or city), type of facility (public or private), construction of the building (wood, cinder block, brick), and the likelihood of reconnaissance. If a facility’s mathematically computed penetration index is below a stated cutoff, certain COTS equipment can be used.

Hard As You Try

Just as you thought it was safe to discuss TEMPEST countermeasures, another form of emanation eavesdropping has come onto the scene. “Optical TEMPEST” is the nickname given to compromising emanations based on the LED status indicators of data communication equipment. Apparently, under certain conditions, the LEDs

that indicate the status of links and data flows can end up being modulated by the information passing through the device. Exposure to prying eyes may yield information about the traffic being passed. Most LEDs connect to regulated power supply rails, but if the equipment is constructed in a certain way, and if the operating voltages are such that voltage regulators are forced to operate in difficult regions, there may be enough correlation to deduce a message, or at least a class of message. One suggested countermeasure: put a small patch of black electrical tape over anything in the telecommunications room that glows.

APPENDIX C

The Orange Book, FIPS PUBS, and the Common Criteria

When the U.S. government writes the standards, and then becomes itself one of the largest customers for equipment that meets requirements defined by those standards, those standards become important very quickly. Add to this the fact that once the government overcomes its own bureaucratic forces to the point it actually accomplishes something, the corollary is that the work stays in force for a long time.

Such is the case of the Orange Book. Different organizations required different levels of security, and because security professionals needed a metric to gauge if a computer system was secure enough for the intended purpose, the government developed the Trusted Computer System Evaluation Criteria (TCSEC) and published them in a book that had an orange cover, hence the nickname “Orange Book.” The Orange Book was part of a family of publications on security with different colored covers called the Rainbow Book series. See the sidebar “Somewhere Over the Rainbow.”

In theory, the Orange Book, et al., have been superseded by the international Common Criteria, but you would not know it in some circles. The language of the Orange Book, and its rating system, is so pervasive that if you’re at all interested in computer security, you’ll need to know something about the Orange Book.

Concurrent with the development of the Orange Book is a series of government standards known as the Federal Information Processing Standards Publications. FIPS PUBS are the language of computer system acquisition and information technology operations, as viewed by the U.S. government.

The Common Criteria is the name given to a unification of TCSEC and its European counterpart, the Information Technology Security Evaluation Criteria. Common Criteria represents where the definition of security standardization is going, but the actual state of the art is found somewhere in the nexus of TCSEC, ITSEC, and the national security evaluation policies from several other nations.

The following sections describe these three standards.

Somewhere Over the Rainbow

The Orange Book was an abstract, very concise description of computer security requirements. It raised more questions than it answered. In an attempt to help system developers, the government has published a number of additional books interpreting Orange Book requirements in particular, puzzling areas. These were known collectively as the Rainbow Series because each had a different color cover, and the influence of these publications is still felt in government computing today.

Chief among the documents in the Rainbow Series was the *Trusted Network Interpretation* (TNI), the Red Book, which interpreted the criteria described in the Orange Book for networks and network components. Published in 1987, this book identified security features not mentioned in the Orange Book that applied to networks, and it described how these features fit into the graded classification of systems described in the Orange Book. For example, the Red Book discussed how the concept of group identification for DAC might be extended to Internet addresses.

Another book that left traces of language that can sometimes be seen today is *Trusted Database Management System Interpretation* (TDI), the Lavender Book. This book was directed toward developers of database management systems and interpreted Orange Book requirements for DBMS products. For example, this book discussed how the concept of security labels might be extended to labels for stored view definitions (obtained via DBMS query commands).

There were more than 20 books in the Rainbow Series. Some others included:

Green Book

Password Management Guideline

Tan Book

A Guide to Understanding Audit in Trusted Systems

Purple Book

Guidelines for Formal Verification Systems

Burgundy Book

A Guide to Understanding Design Documentation in Trusted Systems

About the Orange Book

The Orange Book defined four broad hierarchical divisions of security protection. In increasing order of trust, they were:

- D: Minimal security
- C: Discretionary protection
- B: Mandatory protection
- A: Verified protection

Each division consisted of one or more numbered classes, with higher numbers indicating a greater degree of security. For example, division C contained two distinct classes (C2 offered more security than C1); division B contained three classes (B3 offers more security than B2, which offered more security than B1). Each class was defined by a specific set of criteria that a system must meet to be awarded a rating in that class.

Why Was There an Orange Book?

What was the purpose of the Orange Book? According to the book itself, the evaluation criteria were developed with three basic objectives:

Measurement

To provide users with a metric with which to assess the degree of trust that can be placed in computer systems for the secure processing of classified or other sensitive information. For example, a user can rely on a B2 system to be “more secure” than a C2 system.

Guidance

To provide guidance to manufacturers as to what to build into their trusted commercial products to satisfy trust requirements for sensitive applications.

Acquisition

To provide a basis for specifying security requirements in acquisition specifications. Rather than specifying a hodge-podge of security requirements, and having vendors respond in piecemeal fashion, the Orange Book provides a clear way of specifying a coordinated set of security functions. A customer can be confident that the system she acquires has already been checked out for the needed degree of security.

As the Orange Book put it, the criteria “constitute a uniform set of basic requirements and evaluation classes for assessing the effectiveness of security controls built into... systems.”

These philosophical underpinnings echo through the international replacement for the Orange Book, the Common Criteria.

Orange Book Security Concepts

The Orange Book criteria had four pillars:

- Security policy
- Accountability
- Assurance
- Documentation

Security policy

A *security policy* is the set of rules and practices that regulate how an organization manages, protects, and distributes sensitive information. It's the framework in which a system provides trust.

A security policy is typically stated in terms of subjects and objects. A *subject* is something active in the system; examples of subjects are users, processes, and programs. An *object* is something that a subject acts upon; examples of objects are files, directories, devices, sockets, and windows.

The Orange Book defined a security policy as follows:

Given identified subjects and objects, there must be a set of rules that are used by the system to determine whether a given subject can be permitted to gain access to a specific object.

At the lower levels of security, a security policy could be informal, and the Orange Book did not even require that it be written. For B1 systems and above, the Orange Book required a written security policy. At the higher levels, the security policy became increasingly formal and had to be expressed in mathematically precise notation.

A security model is usually required to give life to the security policy. The Orange Book criteria were based on the state-machine model developed by David Bell and Leonard LaPadula in 1973 under the sponsorship of the U.S. Air Force Electronic Systems Division (ESD). The Bell and LaPadula model was also the first mathematical model of a multilevel secure computer system. This involved the concept of a security lattice, which is now seen in most other security models. The difference between various models today often boils down to what subjects can access which objects from what level of the lattice.

Accountability

Accountability requirements defined in the Orange Book centered on the idea that a system must identify all users and use that information to decide whether the user could legitimately access information. The system was also required to keep track of any security-related actions taken in the system. There was also a requirement that the system produce audit reports, if required, so that system access and usage could be examined.

Generally, the requirements increased in rigor as the trust classification increased. For instance, for C1, the system distinguished only between authorized users and unauthorized users; for C2 and above, each user had an individual ID. Once logged in, the system used the user ID, and the security profile associated with it, to make access decisions. The system also used the ID to audit user actions. The Orange Book required user auditing beginning at the C2 level.

Assurance

Assurance is a guarantee (or at least a reasonable confidence) that the security policy of a trusted system has been implemented correctly and that the system's security features accurately carry out that security policy. The Orange Book identified two types of assurance: operational assurance and life-cycle assurance. *Operational assurance* focused on the basic architecture and features of the system; *life-cycle assurance* focused on controls and standards for building and maintaining the system.

Moving up the ladder from less trusted to more trusted, systems encountered more and more assurance requirements only to a point. At the highest levels (B3 and A1), there were very few additional security features beyond those required for lower-level systems. What was added was a greater demand that a far greater assurance, or trust, could be placed in the security features.

A very important element of system security is the assurance that the system will function as expected and will remain available for operation over the course of its lifetime. Many vendors met the system integrity requirement by providing a set of integrity tests. These tests occurred regularly, often each time the system was powered up, reminiscent of the power on self test (POST) performed by most computers and network appliances. More substantial security diagnostics were usually performed at scheduled preventive maintenance periods.

In theory, every operation performed by every piece of information in a network can provide information that is useful to an attacker. A *covert channel* is an information path that's not ordinarily used for communication in a system and therefore isn't protected by the system's normal security mechanisms. Covert channels have a nice subversive sound to them. They're a secret way to convey information to another person or program—the computer equivalent of a spy carrying a newspaper as a signal: for example, the *New York Times* means the blueprints have been smuggled out of the plant, and the *Washington Post* means there's been a problem.

The reason covert channels aren't more of a problem is that getting meaningful information in this way is quite cumbersome. The Orange Book wisely reserved covert channel analysis and protection mechanisms for the highest levels of security (B2 systems and above), where the information gained by exploiting covert channels is more likely to be worth the quest.

Trusted facility management, another Orange Book criterion, is the assignment of a specific individual to administer the security-related functions of a system. Trusted facility management is closely related to the concept of least privilege, which means that the users in a system should have the least number of privileges—and for the shortest amount of time—needed to do their work. It's also related to the administrative concept of *separation of duties*, the idea that it's better to assign pieces of security-related tasks to several specific individuals; if no one user has total control of the system's security mechanisms, no one user can completely compromise the system.

The Orange Book concept of *trusted recovery* ensures that security is not breached when a system crash or some other system failure occurs. Trusted recovery actually involves two activities: preparing for a system failure and recovering the system. A system failure represents a possibly serious security risk because it is a time when the system is not functioning in the ordinary way. For example, if the system crashes while sensitive data is being written to disk (where it is protected by the system's access control mechanisms), the possibility exists that the data will be left unprotected in memory. Alternatively, a data label representing a lower level of security may have been assigned to a file that was in the process of having secret materials removed from it, but the operation had not been completed when the failure occurred.

Life-cycle assurance. Life-cycle assurance ensures that once assurance activities are begun, they will continue for the rest of the time the computer or network remains in service. It is a metric of diligence to protect hardware and software from tampering or inadvertent modification at any stage in the system's life cycle—during design, development, testing, updating, or distribution. An example of a key life-cycle assurance is configuration management, which carefully monitors and protects all changes to the system's hardware, software, firmware, test suites, and documentation.

These and other Orange Book assurances are subject to verification by testing. The Orange Book even carries a requirement for the highest levels of security that all material and parts into and out of the computer or network be protected by *trusted distribution*, which means that materials must be protected while they are at vendors' facilities and while they are transported to and from. This may require secure storage, protective, tamper-proof packaging, and the use of couriers.

Documentation

The final set of requirements defined in the Orange Book was the set of *documentation requirements*. Developers of trusted systems sometimes complained that the documentation required for the evaluation of a trusted system was formidable. The Orange Book mandated four specific documents, although this varied by trust classification. As you'd expect, the amount and depth of the documentation increased as you moved up the trusted system ladder.

The documentation requirements were:

- *Security Features User's Guide* (SFUG)
- *Trusted Facility Manual* (TFM)
- Test documentation
- Design documentation

The *Security Features User's Guide* (SFUG) was aimed at ordinary, unprivileged system users. It told them what they needed to know about system security features and

how to enforce them, including: logging in, protecting and moving files, and warnings about why certain file moves, adds, and changes were disallowed by the system.

The *Trusted Facility Manual* (TFM) was aimed at system administrators and/or security administrators. It told them everything they needed to know about setting up the system so it would be secure, and how to enforce system security. The Orange Book required that the TFM contain warnings about the functions and privileges that must be controlled in a trusted system. The TFM was a requirement at all levels of security. As security requirements became more complex at the higher levels, the TFM addressed increasingly complex topics as well.

The *test documentation* requirement was also an important piece of security testing. Good test documentation is usually simple but voluminous. It was not uncommon that the test documentation for even a C1 or C2 system to consist of many volumes of test descriptions and results. For all Orange Book levels, the test documentation had to “show how the security mechanisms were tested, and results of the security mechanisms’ functional testing.”

The *design documentation* was a formidable requirement for most system developers. The focus of the design documentation was to document the internals of the system’s (or at least the Trusted Computing Base’s [TCB]) hardware, software, and firmware. A key task was to define the boundaries of the system and to clearly distinguish between those portions of the system that are security-relevant and those that are not.

There were two major goals of design documentation: to prove to the evaluation team that the system fulfills the evaluation criteria, and to aid the design and development team by helping them define the system’s security policy and how well that policy was implemented.

Because there were so many questions about the scope, organization, and quantity of design documentation during system development and evaluation, the government published *A Guide to Understanding Design Documentation in Trusted Systems* (another in the Rainbow Series: the Burgundy Book). This document summarizes all the specific design documentation requirements (36 in all!).

Rating by the Book

As discussed earlier, a security policy states the rules enforced by the system to provide the necessary degree of security. The Orange Book defined the following specific requirements in the security policy category:

- Discretionary access control
- Object reuse
- Mandatory access control
- Labels

What's a Trusted System?

The Orange Book talked about “trusted” systems rather than “secure” systems. The words aren’t really synonymous. No system is completely secure. Any system can be penetrated—given enough tools and enough time. But systems can be trusted, some more than others, to do what we want them to do, and what we expect them to continue to do over time.

The Orange Book provided its own definition of a trusted system, and the language lingers to this day:

...a system that employs sufficient hardware and software integrity measures to allow its use to simultaneously process a range of sensitive unclassified or classified (e.g., confidential through top secret) information for a diverse set of users without violating access privileges.

The concept of the Trusted Computing Base is central to the notion of a trusted system. The Orange Book used the term TCB to refer to the mechanisms that enforce security in a system. The book defined the TCB as follows:

The totality of protection mechanisms within a computer system—including hardware, firmware, and software—the combination of which is responsible for enforcing a security policy. It creates a basic protection environment and provides additional user services required for a trusted computer system. The ability of a trusted computing base to correctly enforce a security policy depends solely on the mechanisms within the TCB and on the correct input by system administrative personnel of parameters (e.g., a user’s clearance) related to the security policy.

Not every part of an operating system needs to be trusted. An important part of an evaluation of a computer system is to identify the architecture, assurance mechanisms, and security features that comprise the TCB, and to show how the TCB is protected from interference and tampering—either accidentally or deliberately.

No Guarantees

An Orange Book rating, even a very high rating, can’t guarantee security. A “trusted” system is still only as secure as the enforcement of that system. Much of actual system security is in the hands of your system administrator and/or your security administrator. Your administrative staff must understand the full range of system features, must set up all necessary security structures, and must train users to do their part. Often, their efforts will seem toothless unless high management lends them weight. The more highly trusted a system is, the less individual users can compromise the system. But system administration remains the vital link between a system’s theoretical security and its actual ability to function securely in the real world.

Discretionary and Mandatory Access Control

The discretionary access control method restricts access to files (and other system objects) based on the identity of users and/or the groups to which they belong. DAC is the most common type of access control mechanism found in trusted systems.

The DAC requirement specifies that users should be able to protect their own files by indicating who can and cannot access them (on a “need-to-know” basis) and by specifying the type of access allowed (e.g., read-only, read and modify, etc.). In contrast to mandatory access control, in which the system controls access, DAC is applied at the discretion of the user (or a program executing on behalf of the user). With DAC, you can choose to give away your data; with MAC, you can’t.

Object Reuse

Object reuse requirements protect files, memory, and other objects in a trusted system from being accidentally accessed by users who aren’t authorized to access them. Consider a few obvious examples of what could happen if object reuse features are not part of a trusted system. Suppose that when you create a new file, your system allocates a certain area of disk to that file. You store confidential data in the file, print it, and eventually delete it. However, suppose the system doesn’t actually delete the data from the physical disk, but simply rewrites the header of the file to indicate deletion. There are many ways to bypass ordinary system procedures and read the data on the disk without using the normal filesystem. Object reuse features provide security by ensuring that when an object—for example, a login ID—is assigned, allocated, or reallocated, the object doesn’t contain data left over from previous usage.

Common object reuse features implemented in trusted systems include:

- Clearing memory blocks or pages before they are allocated to a program or data.
- Clearing disk blocks when a file is scratched or before the blocks are allocated to a file.
- Degaussing magnetic tapes when they’re no longer needed.
- Clearing X Window System objects before they are reassigned to another user.
- Erasing password buffers after encryption.
- Clearing buffered pages, documents, or screens from the local memory of a terminal or printer.

Labels

A *sensitivity label* describes briefly who can access what information in your system. Labels and mandatory access control are separate security policy requirements, but they work together. Beginning at the B1 level, the Orange Book required that every subject (e.g., user, process) and storage object (e.g., file, directory, window, socket)

have a sensitivity label associated with it. (For B2 systems and above, *all* system resources (e.g., devices, ROM) must have sensitivity labels.) A user's sensitivity label specifies the sensitivity level, or level of trust, associated with that user; a user's sensitivity label is usually called a *clearance*. A file's sensitivity label specifies the level of trust that a user must have to be able to access that file.

Label integrity

Label integrity ensures that the sensitivity labels associated with subjects and objects are accurate representations of the security levels of these subjects and objects. Thus, a file sensitivity label, such as:

TOP SECRET [VENUS]

must actually be associated with a TOP SECRET file containing information about the planet Venus. A clearance, such as:

COMPANY CONFIDENTIAL [PAYROLL AUDIT]

must be assigned to an accountant who's responsible for the payroll and auditing activities in your company. Similarly, when information is exported (as discussed in the next few sections), the sensitivity label written on the output must match the internal labels of the information exported.

Suppose that a user edits a TOP SECRET file to remove all TOP SECRET and SECRET information (leaving only UNCLASSIFIED information), and then changes the sensitivity label to UNCLASSIFIED. If the system crashes at this point, has it ensured that the label didn't get written to disk before the changes to the file were written? If such a thing could happen, you'd end up with a file labeled UNCLASSIFIED that actually did contain TOP SECRET data—a clear violation of label integrity. Systems rated at B1 and above must prove that their label integrity features guard against such violations.

Exportation of labeled information

A trusted system must be sure that when information is written by the system, that information continues to have protection mechanisms associated with it. Two important ways of securing exported information are to assign security levels to output devices and to write sensitivity labels along with data. Systems rated at B1 and above must provide secure export facilities.

The Orange Book defined two types of export devices: multilevel and single-level. Every I/O device and communications channel in a system must be designated as one type or the other. Any changes to these designations must be able to be audited. Typically, a system administrator designates devices during system installation and setup.

Exportation to multilevel devices

A *multilevel device* or a *multilevel communications channel* is one to which you can write information at a number of different sensitivity levels. The system must

support a way of specifying the lowest (e.g., UNCLASSIFIED) and the highest (e.g., TOP SECRET) security levels allowed for data being written to such a device.

To write information to a multilevel device, the Orange Book required that the system have some way to associate a security level with it. Mechanisms may differ for different systems and different types of devices. Files written to such devices must have sensitivity levels attached to them (usually written in a header record preceding the data in the file). This prevents a user from bypassing system controls by simply copying a sensitive file to another, untrusted system or device. Of course, there must be system controls on what devices can actually be used in this way—and what happens to the transferred data. A sensitivity label on a file isn't much protection if a user can simply carry a CD home, or can post the file to a web page. In most trusted systems, only disks are categorized as multilevel devices.

Exportation to single-level devices

A *single-level device* or a *single-level communications channel* is one to which you can write information at only one particular sensitivity level. Usually, terminals, printers, tape drives, and communication ports are categorized as single-level devices. The level you specify for a device is usually dependent on its physical location or the inherent security of the device type. For example, your installation might locate printers in a number of different computer rooms and offices. You'd designate these printers as having sensitivity levels corresponding to the personnel who have access to the printers.

You might designate a printer in a public area as UNCLASSIFIED, whereas a printer in a highly protected office used only by an individual with a TOP SECRET clearance might be designated TOP SECRET. Once you've designated a device in this way, the system will be able to send to that device only information at the level associated with the device.

Unlike output sent to multilevel devices, output sent to single-level devices is not required to be labeled with the security level of the exported information, although many trusted systems do label such output. The Orange Book does require that there be some way (system or procedural) to designate the single level of information being sent to the device.

Labeling human-readable output

The Orange Book included very clear requirements for how to label hard-copy output (output that people can see). This included pages of printed output, maps, graphics, and other displays. The system administrator must have some way of specifying the labels that appear on the output.

Subject sensitivity labels

The *subject sensitivity label* requirement states that the system must notify a terminal user of any change in the security level associated with that user during an interactive session. This requirement applies to systems rated at B2 and above.

The idea of subject sensitivity labels is that you must always know what security level you're working at. Trusted systems typically display your clearance when you log in, and redisplay it if the security level changes, either automatically or at the user's request.

Device labels

The *device labels* requirement states that each physical device attached to your system must have minimum and maximum security levels associated with them, and that these levels are used to "enforce constraints imposed by the physical environments in which the devices are located."

Summary of Orange Book Classes

The following sections summarize very briefly the high points of each of the Orange Book evaluation classes described in this appendix. (See Figure C-1.)

D Systems: Minimal Security

According to the Orange Book, division D "is reserved for systems that have been evaluated but that fail to meet the requirements for a higher evaluation class." The Orange Book listed no requirements for division D. It's simply a catch-all category for systems that don't provide the security features offered by systems more highly rated.

In fact, there are no evaluated systems in this class; a vendor wouldn't go to the trouble of getting a system evaluated if it didn't offer a reasonable set of security features. It's likely that the most basic personal computer systems would fall into the division D category if they were evaluated.

C1 Systems: Discretionary Security Protection

C1 systems provide rather limited security features. The Orange Book described C1 systems as an environment of "cooperating users processing data at the same level(s) of security." At this level, security features are intended primarily to keep users from making honest mistakes that could damage the system (e.g., by writing over system memory or critical software) or could interfere with other users' work (by deleting or modifying their programs or data). The security features of a C1 system aren't sufficient to keep a determined intruder out.

Trusted Computer System Evaluation Criteria Summary Chart						
	C1	C2	B1	B2	B3	A1
Discretionary Access Control	█	█	█	█	█	█
Object Reuse	█	█	█	█	█	█
Labels	█	█	█	█	█	█
Label Integrity	█	█	█	█	█	█
Exportation of Labeled Information	█	█	█	█	█	█
Exportation of Multilevel Devices	█	█	█	█	█	█
Exportation of Single-Level Devices	█	█	█	█	█	█
Labeling Human Access Control	█	█	█	█	█	█
Mandatory Access Control	█	█	█	█	█	█
Subject Sensitivity Labels	█	█	█	█	█	█
Device Labels	█	█	█	█	█	█
Identification and Authentication	█	█	█	█	█	█
Audit	█	█	█	█	█	█
Trusted Path	█	█	█	█	█	█
System Architecture	█	█	█	█	█	█
System Integrity	█	█	█	█	█	█
Security Testing	█	█	█	█	█	█
Design Specifications and Verification	█	█	█	█	█	█
Covert Channel Analysis	█	█	█	█	█	█
Trusted Facility Management	█	█	█	█	█	█
Configuration Management	█	█	█	█	█	█
Trusted Recovery	█	█	█	█	█	█
Trusted Distribution	█	█	█	█	█	█
Security User's Guide	█	█	█	█	█	█
Trusted Facility Manual	█	█	█	█	█	█
Test Documentation	█	█	█	█	█	█
Design Documentation	█	█	█	█	█	█

Legend:

- No requirements for this class
- New or enhanced requirements for this class
- No additional requirements for this class

Vertical labels on the right side of the chart:

- Security Policy
- Accountability
- Assurance
- Documentation

Figure C-1. Comparison of evaluation classes

The two main user-visible features required in the C1 class are:

Passwords (or some other mechanism)

These identify and authenticate a user before letting him use the system.

Discretionary protection of files and other objects

With systems of this kind, you can protect your own files by deciding who will be allowed to access them.

C1 systems must also provide a system architecture that can protect system code from tampering by user programs. The system must be tested to ensure that it works properly and that the security features can't be bypassed in any obvious way. There are also specific documentation requirements. Many people feel that if an ordinary Unix or Linux system (with no enhanced security features) were submitted for evaluation, it would be given a C1 rating. (Others claim that it would get a C2 rating.)

C2 Systems: Controlled Access Protection

C2 systems provide much more stringent security than C1 systems. In addition to providing all the features required at the C1 level, C2 systems offer the following additional user-visible features:

Accountability of individual users

Through individual password controls and auditing (maintaining a record of every security-related action every user performs), the system can keep track of who's doing what in the system.

More detailed discretionary controls

The Orange Book describes C2 requirements as follows: "These access controls shall be capable of including or excluding to the granularity of a single user." Through access control lists or some other mechanism, you must be able to specify, for example, that only Mary and Joe can read a file and that only Sam can change it.

Object reuse

This feature makes sure that any data left in memory, on disk, or anywhere else in the system doesn't accidentally become accessible to a user.

The C2 system architecture must allow system resources to be protected via access control features. C2 systems also require more rigorous testing and documentation.

B1 Systems: Labeled Security Protection

B1 (and higher) systems support mandatory access controls. In a MAC system, every file (and other major object) in the system must be labeled. The system uses these sensitivity labels, and the security levels of the system's users, to enforce the system's security policy. In MAC systems, protection of files is no longer discretionary. You can't give another user access to one of your files unless that user has the necessary clearance to the file. The Orange Book imposed many specific labeling requirements on B1 systems, including labeling of all information that's exported from the system.

B1 systems must have a system architecture that more rigorously separates the security-related portions of the system from those that are not security-related. B1 systems

also require more stringent testing and documentation. In the B1 class, your documentation must include a model of the security policy supported by the system. This policy need not be a mathematical one (as is required in the higher classes), but it must be a clear statement of the rules enforced by the system's security features.

B2 Systems: Structured Protection

B2 and higher systems don't actually add many more specific user-visible security features to the set already required of B1 systems. Instead, they extend these features, and they require additional assurance that the features were designed and work properly.

In the B2 class, labeling features are extended to include all objects in the system, including devices. B2 systems also add a trusted path feature, allowing the user to communicate with the system in a direct and unmistakable way—for example, by pressing a certain key or interacting with a certain menu. The system design must offer proof that an intruder can't interfere with, or “spoof,” this direct channel.

B2 systems support *least privilege*, the concept that users and programs should possess the least number of privileges they need, and for the shortest time necessary, to perform system functions. This concept is implemented via a combination of design, implementation, and assurance requirements. One example is the requirement that there be a separate system administrator and operator.

From a system design point of view, B2 systems require substantially more modularity and use of hardware features to isolate security-related functions from those that are not security-related. B2 systems also require a formal, mathematical statement of the system's security policy, as well as more stringent testing and documentation. They also require that a configuration management system manage all changes to the system code and documentation and that system designers conduct a search for covert channels—secret ways that a developer could use the system to reveal information.

Few systems have been successfully evaluated at B2 and higher.

B3 Systems: Security Domains

There are no requirements for new user-visible features at the B3 level, but the system design and assurance features are substantially more rigorous. Trusted facility management (assignment of a specific individual as security administrator) is required, as is trusted recovery (procedures ensuring that security doesn't fail if the system does) and the ability to signal the administrator immediately if the system detects an “imminent violation of security policy.”

The Orange Book indicated that B3 systems must be “highly resistant to penetration.” The TCB is very tightly structured to exclude code not required for the enforcement of system security.

B3 systems aren't very common. It's difficult enough to obtain a B3 rating that vendors may choose to go all the way for an A1 rating.

A1 Systems: Verified Design

At present, A1 systems are at the top of the security heap, although the Orange Book does discuss the possibility of defining requirements for systems that exceed current A1 requirements in areas of system architecture, testing, and formal verification.

A1 systems are pretty much functionally equivalent to B3 systems. The only actual feature provided by A1 systems beyond B3 requirements is trusted distribution, which enforces security while a trusted system is being shipped to a customer. What makes a system an A1 system is the additional assurance that's offered by the formal analysis and mathematical proof that the system design matches the system's security policy and its design specifications.

Only two or three systems ever received an A1 rating.

Complaints About the Orange Book

Not everyone thought the Orange Book answered all needs, and it tended to be government-specific. Some respected security practitioners disagreed with the government's reliance on this book alone as a way of measuring trust. Here are some of the main claims about the inadequacies of the Orange Book.

- The Orange Book model worked only in a government-classified environment, and the higher levels of security weren't appropriate for the protection of commercial data, where data integrity rather than confidentiality was the chief concern.
- The Orange Book emphasized protection from unauthorized access, while most security attacks actually involve insiders. (Orange Book defenders pointed out that the book's principle of *least privilege* addressed this issue to some degree.)
- The Orange Book did not address networking issues, although the Red Book did.
- The Orange Book contained a relatively small number of security ratings.

In addition, there was a need to harmonize the computer security requirements with those of nations with which there was a likelihood of joint military and commercial operations. The European Information Technology Security Evaluation Criteria (ITSEC) was similar to U.S. TCSEC, of which the Orange Book was central. A merger of the two was in order. It came to be known as the Common Criteria.

FIPS by the Numbers

The Federal Information Processing Standards Publications describe how the U.S. government acquires computer systems. If you or your company sells computers or anything that connects to them to the government, you will need to be aware of the

FIPS PUBS that deal with that equipment and the procedures that surround it. There are actually some computer security industry certifications that are based largely on the mastery of the FIPS PUBS.

The FIPS PUBS list is maintained by the NIST, an agency of the U.S. Computer Departments Technology Administration. In 2005, over a dozen FIPS PUBS were withdrawn. Current information can be obtained by checking the FIPS web site at: <http://www.itl.nist.gov/fipspubs/index.htm> or calling the department at 301-975-2832.

Some of the following references include dates, which are written in the government format of YR(YR)-MONTH-DATE.

FIPS 4-2, Representation of calendar date to facilitate interchange of data among information systems.

Adopts American National Standard ANSI X3.30-1997: Representation of Date for Information Interchange (revision of ANSI X3.30-1985 (R1991)). FIPS 4-2 supersedes FIPS PUB 4-1, dated January 27, 1988, and updates the standard for representing calendar date and implements the U.S. government's commitment to use four-digit year elements (e.g., 1999, 2000, etc.) in its information technology systems.

FIPS 5-2 Codes for the Identification of the States, the District of Columbia and the Outlying Areas of the United States, and Associated Areas—87 May 28

Provides a set of two-digit numeric codes and a set of two-letter alphabetic codes that represent the 50 states, the District of Columbia and the outlying areas of the United States, and associated areas such as the Federated States of Micronesia and Marshall Islands, and the trust territory of Palau.

Additional information on the content of this FIPS may be obtained from the Bureau of the Census at (301) 763-1522.

FIPS 6-4 Counties and Equivalent Entities of the U.S., Its Possessions, and Associated Areas—90 Aug 31

Provides the names and three-digit codes that represent the counties and statistically equivalent entities of the 50 states, the District of Columbia, and the possessions and associated areas of the United States, for use in the interchange of formatted machine-sensible data. Implements ANSI X3.31-1988. Minor editorial corrections made in January 2005.

Additional information on the content of this FIPS may be obtained from the Bureau of the Census at (301) 763-9031.

FIPS 10-4 Countries, Dependencies, Areas of Special Sovereignty, and Their Principal Administrative Divisions—95 Apr

Provides a list of the basic geopolitical entities in the world, together with the principal administrative divisions that comprise each entity. Each basic geopolitical entity is represented by a two-character, alphabetic country code. Each principal administrative division is identified by a four-character code consisting of

the two-character country code followed by a two-character administrative division code. These codes are intended for use in activities associated with the mission of the Department of State and in national defense programs.

Note: Change notices for FIPS 10-4 are issued by the National Geospatial-Intelligence Agency (NGA), and are available on NGA's GEOnet Names Server (GNS) at http://earth-info.nga.mil/gns/html/fips_files.html.

Additional information on the content of this FIPS may be obtained from the National Geospatial-Intelligence Agency (NGA) at (301) 227-1407.

FIPS 113 Computer Data Authentication—85 May 30

Specifies a data authentication algorithm (DAA) which, when applied to computer data, automatically and accurately detects unauthorized modifications, both intentional and accidental. Based on FIPS PUB 46, this standard is compatible with requirements adopted by the Department of Treasury and the banking community to protect electronic fund transfer transactions.

FIPS 140-2 Security Requirements for Cryptographic Modules—01 May 25

This Federal Information Processing Standard (140-2) was recently approved by the Secretary of Commerce. It specifies the security requirements that will be satisfied by a cryptographic module, providing four increasing, qualitative levels intended to cover a wide range of potential applications and environments. The areas covered, related to the secure design and implementation of a cryptographic module, include specification; ports and interfaces; roles, services, and authentication; finite state model; physical security; operational environment; cryptographic key management; electromagnetic interference/electromagnetic compatibility (EMI/EMC); self-tests; design assurance; and mitigation of other attacks.

This version supersedes FIPS PUB 140-1, January 11, 1994.

FIPS 161-2 Electronic Data Interchange (EDI)—96 May 22

FIPS 161-2 adopts, with specific conditions, the families of EDI standards known as X12, UN/EDIFACT, and HL7 developed by national and international standards developing organizations. FIPS 161-2 does not mandate the implementation of EDI systems within the federal government but requires the use of the identified families of standards when federal agencies and organizations implement EDI systems.

FIPS 180-2 Secure Hash Standard (SHS)—02 Aug

Specifies a secure hash algorithm to be used by both the transmitter and intended receiver of a message in computing and verifying a digital signature.

FIPS 180-2 superseded FIPS 180-1 as of February 1, 2003.

FIPS 181 Automated Password Generator (APG)—93 Oct 05

Specifies a standard to be used by Federal organizations that require computer-generated pronounceable passwords to authenticate the personal identity of an automated data processing (ADP) system user, and to authorize access to system

resources. The standard describes an automated password generation algorithm that randomly creates simple pronounceable syllables as passwords. The password generator accepts input from a random number generator based on the DES cryptographic algorithm defined in Federal Information Processing Standard 46-2.

FIPS 183 Integration Definition for Function Modeling (IDEF0)—93 Dec 21

Describes the IDEF0 modeling language (semantics and syntax), and associated rules and techniques, for developing structured graphical representations of a system or enterprise. Use of this standard permits the construction of models comprising system functions (activities, actions, processes, and operations), functional relationships, and data (information or objects) that support systems integration.

FIPS 184 Integration Definition for Information Modeling (IDEFIX)—93 Dec 21

Describes the IDEF1X modeling language (semantics and syntax), and associated rules and techniques, for developing a logical model of data. IDEF1X is used to produce a graphical information model that represents the structure and semantics of information within an environment or system. Use of this standard permits the construction of semantic data models, which may serve to support the management of data as a resource, the integration of information systems, and the building of computer databases.

FIPS 185 Escrowed Encryption Standard (EES)—94 Feb 09

This nonmandatory standard provides an encryption/decryption algorithm and a Law Enforcement Access Field (LEAF) creation method that may be implemented in electronic devices and may be used at the option of government agencies to protect government telecommunications. The algorithm and the LEAF creation method are classified and are referenced, but not specified, in the standard. Electronic devices implementing this standard may be designed into cryptographic modules that are integrated into data-security products and systems used in data-security applications. The LEAF is used in a key escrow system that provides for decryption of telecommunications when access to the telecommunications is lawfully authorized.

FIPS 186-2 Digital Signature Standard (DSS)—00 Jan 27

Specifies algorithms appropriate for applications requiring a digital, rather than written, signature. A digital signature is represented in a computer as a string of binary digits; it is computed using a set of rules and a set of parameters such that the identity of the signatory and integrity of the data can be verified. An algorithm provides the capability to generate and verify signatures. Signature generation uses a private key to generate a digital signature; it uses a public key that corresponds to, but is not the same as, the private key. Each user possesses a private and public key pair. Private keys are kept secret; public keys may be shared. Anyone can verify the signature of a user by using that user's public key. Signature generation can be performed only by the possessor of the user's private key.

This revision supersedes FIPS 186-1 in its entirety.

FIPS 188 Standard Security Label for Information Transfer—94 Sept 6

Defines a security label syntax for information exchanged over data networks and provides label encodings for use at the Application and Network Layers. ANSI/TIA/EIA-606-1993

FIPS 190 Guideline for the Use of Advanced Authentication Technology Alternatives—94 Sept 28

Describes the primary alternative methods for verifying the identities of computer system users, and provides recommendations to federal agencies and departments for the acquisition and use of technology that supports these methods.

FIPS 191 Guideline for the Analysis of Local Area Network Security—94 Nov 9

Discusses threats and vulnerabilities and considers technical security services and security mechanisms.

FIPS 192 Application Profile for the Government Information Locator Service (GILS)—94 Dec 7

Describes an application profile for the GILS, which is based primarily on ANSI/NISO Z39.50-1992. GILS is a decentralized collection of servers and associated information services that is used by the public either directly or through intermediaries to find public information throughout the federal government.

FIPS 192-1 (a) & (b) Application Profile for the Government Information Locator Service (GILS)—97 Aug 1

Describes the U.S. federal government use of the international application profile for the GILS. The GILS Profile is based primarily on ISO 23950, presently equivalent to the ANSI/NISO Z39.50-1995/Version 2. GILS is a decentralized collection of servers and associated information services that is used by the public either directly or through intermediaries to find public information throughout the federal government.

FIPS 196 Entity Authentication Using Public Key Cryptography—1997 Feb 18

Specifies two challenge-response protocols by which entities in a computer system may authenticate their identities to one another. These protocols may be used during session initiation, and at any other time that entity authentication is necessary. Depending on which protocol is implemented, either one or both entities involved may be authenticated. The defined protocols are derived from an international standard for entity authentication based on public key cryptography, which uses digital signatures and random number challenges.

FIPS 197 Advanced Encryption Standard (AES)—01 Nov 26

The AES specifies a FIPS-approved cryptographic algorithm that can be used to protect electronic data. The AES algorithm is a symmetric block cipher that can encrypt (encipher) and decrypt (decipher) information. Encryption converts data to an unintelligible form called ciphertext; decrypting the ciphertext converts the data back to its original form, called plaintext.

FIPS 198 The Keyed-Hash Message Authentication Code (HMAC)—02 March

This standard describes a keyed-hash message authentication code (HMAC), a mechanism for message authentication using cryptographic hash functions. HMAC can be used with any iterative approved cryptographic hash function, in combination with a shared secret key. The cryptographic strength of HMAC depends on the properties of the underlying hash function. The HMAC specification in this standard is a generalization of Internet RFC 2104, HMAC, Keyed-Hashing for Message Authentication, and ANSI X9.71, Keyed Hash Message Authentication Code.

FIPS 199 Standards for Security Categorization of Federal Information and Information Systems—04 Feb

FIPS 199 addresses one of the requirements specified in the Federal Information Security Management Act (FISMA) of 2002, which requires all federal agencies to develop, document, and implement agency-wide information security programs for the information and information systems that support the operations and the assets of the agency, including those provided or managed by another agency, contractor, or other source. FIPS 199 provides security categorization standards for information and information systems. Security categorization standards make available a common framework and method for expressing security. They promote the effective management and oversight of information security programs, including the coordination of information security efforts throughout the civilian, national security, emergency preparedness, homeland security, and law enforcement communities. Such standards also enable consistent reporting to OMB and Congress on the adequacy and effectiveness of information security policies, procedures, and practices.

FIPS 201 Personal Identity Verification for Federal Employees and Contractors—05 Feb

This standard specifies the architecture and technical requirements for a common identification standard for federal employees and contractors. The overall goal is to achieve appropriate security assurance for multiple applications by efficiently verifying the claimed identity of individuals seeking physical access to federally controlled government facilities and electronic access to government information systems.

I Don't Want You Smelling My Fish

At the time the United States was assembling the TCSEC into the Orange Book, many other nations did the same. Europe put together its ITSEC program. The Canadians developed the Canadian Trusted Computer Product Evaluation Criteria (CTCPEC) guidelines. Each project had similar goals and several differences.

With all the computers and networks in a country being responsible for an increasing part of the security, or at least the economic well being, of that country, it is only natural that each country has its own national interests at heart when it develops its security evaluation programs and capabilities.

It is a little like buying fresh seafood. Each of us has pet procedures for picking piscatorial products. (Find a good supplier, avoid fish that smell, look for clear eyes and bright pink or red gills, and well-defined fins.) The individual who has no idea what he is looking for is the most likely to be sold flounder that is past its prime. However, with networking equipment and computers, it is more than a dinner party that is being threatened. The nation that can keep itself free of intruding eavesdroppers, capacity-robbing viruses, infrastructure-damaging threats, and terrorists' activities, is so much ahead on the world stage. And to insure that, each nation needs to develop and administer its own methods of ensuring secure computing.

And yet we do all have to play together. Equipment is cheaper for all if it is manufactured for a world, rather than for a national, market. Operating system software is more likely to run universally if there is a commonality to the platforms on which it plays. Working together simplifies standardization and provides a uniform landscape. Even bad artifacts of the Internet age, such as phishing, spam, and online scams are easier to prosecute if both the victim and the attacker live in countries with computer crime laws. It's that much more difficult if the attack is launched from a country that does not acknowledge illicit computer activity as a criminal act.

The Common Criteria is an attempt to take the best of many methods and provide the world with a single method of determining a product's security and protection levels. The project began as a joint effort between the United States (TCSEC), Europe (ITSEC), and Canada (CTCPEC). It has continued to develop into an international equipment certification system that can serve many nations at the same time. Common Criteria also frees governments from direct involvement, because a Common Criteria testing laboratory may in fact be a commercial entity that is certified for testing.

Common Criteria recognizes that equipment intended for different uses can be subjected to different tests, unlike the Orange Book, which tended to view everything from its attitude of security and defense potential. The Common Criteria process is now part of an international standard known as ISO 15408 (Common Criteria). The web site for the consortium is <http://www.commoncriteriaportal.org/>.

Common Criteria Evaluation Assurance Levels (EALs)

The evaluation assurance level defines the thoroughness of the testing to be done on a product. The protection profile (PP) states the need for a security solution and the rationale for the testing. Protection profiles are meant to be reusable so that every sponsor does not need to reinvent the wheel for each new product to be evaluated. If, for instance, a profile exists that defines the nature of "medium strength robustness" for security algorithms, a developer who has developed a new algorithm can submit it for Common Criteria evaluation according to that profile, if it is relevant and will give an adequate test. A protection profile should include the intended evaluation assurance level for the product to be tested.

The next requirement is to provide the definition of the security target, which is a fancy way of explaining how a product is intended to operate, in what environment it is to operate, and what components should be subjected to testing. The product to be tested is called the target of evaluation.

Next, the functionality requirements and the assurance requirements are spelled out, followed by the actual evaluation. Presumably, the product will meet the requirements of the EAL specified in the protection profile.

The EAL is actually a measure of how much rigor, or the depth of the science, is involved in the test. There are seven possible levels that can result from a successful evaluation. One, EAL0 indicates failure. The various EALs and their relationship to the corresponding Orange Book ratings are displayed in Table C-1.

Table C-1. Common Criteria to Orange Book Harmony

EAL Level	Description of the Level	Orange Book Level (TCSEC)
EAL0	Inadequate assurance	n/a
EAL1	Functionally tested	n/a
EAL2	Structurally tested	C1
EAL3	Methodically tested and checked	C2
EAL4	Methodically designed, tested, and reviewed	B1
EAL5	Semiformally designed and tested	B2
EAL6	Semiformally verified design and tested	B3
EAL7	Formally verified design and tested	A1

Returning to the analogy of selecting your own fish, the result of this standards activity is that, instead of subjecting equipment to its own evaluation process, the U.S. government has transferred equipment verification duties to signatories of the Common Criteria agreements. This is by virtue of what is called the Common Criteria Recognition Arrangement (CCRA). Within the CCRA only evaluations up to the trust level called EAL 4 are mutually recognized, however some European countries that were formerly part of the ITSEC agreement may recognize higher EALs as well.

Evaluations at EAL5 and above tend to involve the security requirements of the host nation's government and likely will be performed by those nations' own national laboratories. In other words, for small items, Common Criteria works well. For important stuff, the job is still done in-house.

Common Criteria documentation and further information can be accessed at this web site: <http://niap.nist.gov/cc-scheme/index.html>.

Does ISO 15408 Certification Mean a Product Is Always Secure?

Interestingly, the closer an item moves toward being a commodity, the more flexible its security issues can be. Let's consider a common PC operating system, such as Linux or some version of Microsoft Windows. It happens that Windows 2000 is ISO 15408-certified, but regular security patches to address security vulnerabilities are still published by Microsoft for Windows 2000.

This seeming contradiction is possible, in fact necessary, because of the broad potential use for such products. As an adaptation to the process of getting an ISO 15408 certification, a vendor is allowed to stipulate certain assumptions about the operating environment. In addition, the strength of the threats to be faced in that environment can be stated in advance. Finally, the manufacturer can state what configuration of the product is to be evaluated.

As a result, this kind of broadly deployed, multiple use product can be ISO 15408 certified, but should be considered only secure in the assumed, specified circumstances of the test, also known as the *evaluated configuration*.

Index

Numbers

2600 Magazine, 17
3DES, 163
3WAY, 163
60-Hz AC (alternating current), 254

A

access cards, types of, 213
access controls, 188
 ACLs, 70
 data, 50
 discretionary, 266
 discretionary access control
 (DAC), 66–68
 ownership, 67
 self/group/public controls, 67
 in practice, 66–71
 mandatory, 266
 mandatory access control (MAC), 66, 69–70
 access decisions, 69
 data import and export, 69
 passwords, 61
 role-based access control (RBAC), 66, 70
 system, 49
 why use, 66
access decisions, 69
access points, 221
 hardening, 234–236
 rogue, 243
accessibility, 203
accountability, 9, 261
accuracy, 11

accuracy of information, 174
Active Directory, 72
active electronic card, 213
Active Server Pages, 129
active taps, 177
ActiveX, 128
 malicious controls, 132
ad hoc infrastructure, 221
Address Resolution Protocol (ARP), 116
Adleman, Leonard, 165
administrative security, 97
 analyzing costs and risks, 99–102
 employees, setting security rules for, 103
 overall planning and
 administration, 98–103
 planning for disaster, 102
 training users, 103
Advanced Encryption Standard
 (AES), 160–163, 186, 277
 development effort, 160
 how it works, 161
Advanced Encryption System (AES), 36
 protocol, 241
Advanced Research Projects Agency
 (ARPA), 28
algorithms, government, 168
American National Standards Institute
 (ANSI), 36
amplitude, 220
amplitude modulation (AM), 220
Anderson, James P., 30
ANSI X3.105, 154
ANSI X3.106, 154
ANSI X3.92, 154

We'd like to hear your suggestions for improving our indexes. Send email to index@oreilly.com.

- antenna, 227
gain, 232
antivirus programs, 92
AppleTalk, 181
applets, 127
malicious, 127
application traffic, 198
arbitrated, 191
ARNES (Academic and Research Network of Slovenia), 5
ARPANET, 115
assurance, 262
life-cycle, 262, 263
operational, 262
asymmetric key, 146
asynchronous transfer mode (ATM), 115
ATMs, 177
Atomic Energy Act of 1946, 23
audit logging, 10
audit, security, 108
auditing, 189
AusCERT (Australian CERT), 5
authentication, 9, 11, 50, 169, 190, 196, 203
multifactor, 51
mutual, 52
per-session, 53
server (AS), 55
two-factor, 208
authenticator, 56
authenticity, 174
authorization, 9, 61–65
access models, 64
Bell-LaPadula model, 64
Biba model, 64, 65
categories or compartments, 63
classification, 63
sensitivity labels, 62
Automated Password Generator (APG), 275
availability, 9
(see also C-I-A acronym)
availability of networks, 174
- B**
- back door, 89, 132
backups, 105
hints for, 106
badges, 51
BANs, 182
Barlow, John, 42
baseband, 185
basic service set (BSS), 221
Bell, David, 30, 261
- Bell-LaPadula model, 31, 64, 261
Beneich, Denis, 88
Biba model, 64, 65
biometrics, 54, 212–218
devices, 203
fingerprints, 216
handprints, 217
iris scans, 216
keystroke verification, 218
overview, 212–215
retina patterns, 215
security, 10
signature-writing pattern, 218
voice verification, 217
BLOWFISH, 163
bombs, 88
Bootstrap Protocol (BOOTP), 123
Bouncing Ball virus, 85
Brain virus, 85
Breton, Thierry, 88
bridge, 223
broadband, 6, 185
broadcast networks, 182
Brooks Act, 153
Brooks Act mandate, 32
Brunner, John, 85
brute force attack, 61
BubbleBoy virus, 79
buffer overflow attacks, 131
building area networks, 182
Burgundy Book, 259
buying security, 18–21
government requirements, 19
information protection, 20
- C**
- Caesar, Julius, 144
calendar date standards, 274
California Data Security Act (California SB 1386), 100
campus area networks, 182
Canada's first national Computer Emergency Response Team, 5
Canadian Trusted Computer Product Evaluation Criteria (CTCPEC), 278
CanCERT, 5
CANs, 182
capacitance card, 213
CAST, 163
CAST-256, Entrust Technologies, Inc.
(represented by Carlisle Adams), 164

cells in ATM, 115
central office (CO), 113
CERT Polska, 5
certificates, 166, 167
challenge and response, 52
Challenge Handshake Authentication Protocol (CHAP), 52
challenge-response systems, 210
channels, 244
checksum, 56
C-I-A acronym, 9
Cipher Block Chaining (CBC), 157
Cipher Feedback (CFB), 157
ciphers, 141
 substitution, 143, 144
 transposition, 143, 144
 types, 143
 (see also keys)
ciphertext, 141
circuit, 179
circuit switching, 113
Cisco Certified Security Professional (CCSP), 27
civil libertarians, 41
classification, 63
cleartext, 141
client-side attacks, 124
 web, 130
client-side scripts, 125
 languages, 127
climate, 205
cloak, 244
CNRS, 164
 Centre National pour la Recherche Scientifique, Ecole Normale Supérieure (represented by Serge Vaudenay), 164
coaxial cable, 185
Code Red virus, 79
codetext, 141
Cohen, Fred, 84
cold sites, 102
ColdFusion, 129
 ColdFusion Markup Language (CFML), 129
Commercial Communications Security Endorsement Program (CCEP), 36, 168
Common Criteria, 258, 279–281
Common Criteria Evaluation Assurance Levels (EALs), 279
Common Criteria Recognition Arrangement (CCRA), 280
Common Gateway Interface (CGI), 128
Communications Act of 1934, 23
communications and network security, 173–199
 active taps, 177
 appropriate technology, 176
 passive taps, 177
 threats, 176
 vulnerabilities, 175
 what makes communication secure, 174–177
communications integrity, 196
communications security, 18
communications vulnerabilities, 14
compartmentalized information, 64
compromise protection services, 197
CompTIA's Security+, 27
computer crime, 8
Computer Emergency Response Team (CERT), 4
Computer Fraud and Abuse Act, 39, 40
Computer Incident Advisory Capability (CIAC), 5
computer security, 18
 accuracy, 11
 audit logging, 10
 authentication, 11
 availability, 12
 biometric security, 10
 broad definition of, 10
 changes over the years, 25–27
 confidentiality, 11
 early efforts, 27–31
 research and modeling, 30
 security systems development, 31
 history, 22–45
 information (see information security)
 integrity, 11
 mandates and legislation, 37–45
 balancing act, 38
 Computer Fraud and Abuse Act, 39
 Computer Security Act of 1987, 40
 recent initiatives, 41
 overview, 9–12
 secrecy, 11
 standardization, 32–37
 cryptography, 36
 emanations, 37
 Orange Book (see Orange Book)

computer security (*continued*)
 threats (see threats)
 what users can do, 21
Computer Security Act of 1987, 40
Computer Security Evaluation Center (CSC), 34
computer system security, 49
 data access controls, 50
 security administration, 50
 system access controls, 49
 system administration, 50
 system design, 50
 what makes system secure, 49–50
Computer Underground Digest, 17
COMSEC (U.S. Communications Security), 27
confidentiality, 9, 11, 174, 203
 (see also C-I-A acronym)
connectionless communications, 180
connection-oriented communications, 179
cookies, 131
 poisoned, 131
 stolen, 132
Core Wars, 83
corporate raiders, 16
costs of security, analyzing, 99–102
countermeasures, 12
 communications security, 18
 computer security, 18
 physical security, 18
counties and statistically equivalent entities, 274
Courtney, Robert H., Jr., 17
crackers, 16
criminals, 16
Critical National Infrastructure (CNI), 5
cross-site scripting (CSS), 132
cryptanalysis, 137
cryptogram, 141
cryptographic
 export restrictions, 171
 locks, 210
 modules, 275
 products, 24
cryptography, 137
 standardization, 36
 (see also encryption)
cryptology, 137
CRYPTON, 163
 Future Systems, Inc. (represented by Chae Hoon Lim), 164

D

Daemen, Joan, 160
data access controls, 50
data and voice communications, history of, 113
data authentication algorithm (DAA), 275
Data Encryption Standard (DES), 36, 146, 153–163
 application of, 158–160
 Cipher Block Chaining (CBC), 157
 Cipher Feedback (CFB), 157
 ECB (Electronic Codebook), 157
 Output Feedback (OFB), 157
 what is, 155–158
data import and export, 69
data modification, 119
data navigation protocols, 117
 attacks, 118
data protection, 49–78
day-to-day system administration, 97, 103–109
 backups, 105
 hints for, 106
 hardware and software security tools, 107
 security audit, 108
DEAL, 163
 Richard Outerbridge, Lars Knudsen, 164
decibel (dB), 232
deciphering, 141
decryption, 141
dedicated circuits, 114
Deep Crack, 155
default accounts, 108
Defense Advanced Research Projects Agency (DARPA), 4, 28
Defense Authorization Act, 23
denial of service, 177, 196
denial of service attack (DoS), 12, 81
Department of Defense, 28
Department of Energy, 5
Department of Homeland Security, 6, 44
Department of the Treasury, 171
DES (see Data Encryption Standard)
design documentation, 264
DESX, 163
device labels, 269
Dewdney, A.K., 88
DFC, 164
DIAMETER, 55, 178
DIAMOND2, 163

dictionary attack, 61
Diffie, Whitfield, 165
Diffie-Hellman public key system, 165
Digital Equipment Corporation's DECnet and Digital Network Architecture (DNA), 181
Digital Signature Standard (DSS), 276
digital signatures, 166, 191
director, 231
directory services, 72–76
 Active Directory, 72
 Lightweight Directory Access Protocol (LDAP), 74–76
 X.500, 72–75
disaster, planning for, 102
discretionary access control (DAC), 18, 66–68, 266
 ownership, 67
 self/group/public controls, 67
dish antenna, 231
distributed denial of service, 177
distributed denial of service attack (DDoS), 6, 81
division multiplexing (TDM), 114
Domain Name Service (DNS), 113
 vulnerabilities, 121
dormant accounts, 108
Dynamic Host Configuration Protocol (DHCP), 113, 122

E

E2, 164
E2 NTT
 Nippon Telegraph and Telephone Corporation (represented by Masayuki Kanda), 164
earthquakes, 206
Easter egg, 89
eavesdroppers, 14
eavesdropping attacks, 239
 advanced, 241
 defenses, 240–241
ECB (Electronic Codebook), 157
Edwards, Dan, 87
EEE 802.11x wireless standard, 175
electric circuit card, 213
electricity, 206
electromagnetic emissions, 252, 254
electromagnetic radiation, 14
Electronic Data Interchange (EDI), 275
electronic eavesdroppers, 14

Electronic Frontier Foundation, 42
electrostatic vector, 233
ELF (extremely low frequency), 254
email, spam and, 120
emanations
 electronic, 28
 standardization, 37
 vulnerabilities, 14
emergency sites, 102
employee security management, 104
employees, setting security rules for, 103
enciphering, 141
encrypted communications, 190–191
encryption, 52, 60, 137–172, 175
 algorithm, 141
 ciphers, 143
 cryptographic export restrictions, 171
 data standards (see Data Encryption Standard (DES))
 defined, 141–142
 end-to-end, 151
 end-to-end link, 151–152
 government algorithms, 168
 government cryptographic programs, 170–171
 history, 138–141
 keys (see keys)
 link, 152
 off-line, 151
 one-time pad, 150–151
 online, 152
 other algorithms, 163–168
 quantum, 142
 RSA algorithm, 165
 table maintained by RSA Laboratories, 172
 uses for, 143
encryption technologies, 24
Endorsed TEMPEST Products Program, 256
Endorsed TEMPEST Test Services Program, 256
end-to-end encryption, 151, 191
end-to-end link encryption, 151–152
Enigma machine, 139
Environmental Protection Agency, 254
equipment locks, 210
Escrowed Encryption Standard (EES), 276
estimates, 79
European Community's Information Technology Security Evaluation Criteria (ITSEC), 43
exclusive OR (XOR), 150

exploiting vulnerabilities, 15
exploits, 80, 81
ExploreZip virus, 79

F

false positive, 241
Family Educational Rights and Privacy Act, 77
Faraday screen, 255
Faurer, Lincoln D., 29
Federal Bureau of Investigation, 8
Federal Computer Incident Response Center (FedCIRC), 5
federal employees and contractors, 278
Federal Information Processing Standard (FIPS), 160
Federal Information Processing Standards Publications (see FIPS PUBS)
Federal Information Security Management Act (FISMA), 278
federated identity management, 76
fiber-optic cable, 185
Field Programmable Gate Arrays (FPGAs), 155
file protection changes, 109
File Transfer Protocol (FTP), 113, 117, 119
financial effects of viruses, 79
fingerprints, 216
FIPS PUBS (Federal Information Processing Standards Publications), 258, 273–278
FIPS 10-4, 274
FIPS 113, 275
FIPS 140-2, 275
FIPS 161-2, 275
FIPS 180-2, 275
FIPS 181, 275
FIPS 183, 276
FIPS 184, 276
FIPS 185, 276
FIPS 186-2, 276
FIPS 188, 277
FIPS 190, 277
FIPS 191, 277
FIPS 192, 277
FIPS 192-1 (a) & (b), 277
FIPS 196, 277
FIPS 197, 277
FIPS 198, 278
FIPS 199, 278
FIPS 201, 278
FIPS 4-2, 274

FIPS 5-2, 274
FIPS 6-4, 274
FIPS PUB 113, 154
FIPS PUB 46, 36
FIPS PUB 46-1, 154
FIPS PUB 74, 154
FIPS PUB 81, 154
fire and smoke, 205
firewalls, 92, 107
fixed microwave, 224
foreign intelligence agents, 16
Freedonia group, 19
free-space laser, 224
frequency division multiplexing (FDM), 114
frequency hop systems, 221
frequency modulation (FM), 220
FROG, 164
Frog, 164
Fu Manchu virus, 85

G

gateways, 180, 189
Generally Accepted Information Security Principles (GAISP), 44
Generally Accepted System Security Principles (GASSP), 43
geopolitical entities, 274
Gerrold, David, 83
Gore, Al, 112
GOST, 163
government algorithms, 168
government cryptographic programs, 170–171
Government Information Locator Service (GILS), 277
government mandating protection of nongovernment information, 24
government's restriction of technological information, 23
Gramm-Leach-Bliley Act, 77
Green Book, 259
group accounts, 108

H

hackerbusters, 4
hackers, 27
 rights, 42
handprints, 217
handshake, 118
hardware security tools, 107
hardware vulnerabilities, 13

- hash, 59
Health Insurance Portability and Accountability Act (HIPAA), 4, 77, 242
Hellman, M.E., 165
heterogeneous computing, 190
heuristic detection, 93
hints for safe computing, 105
Homeland Protection Act, 23
honeypot, 108
horizontal polarization, 227
host, 179
hot sites, 102
hot spots, 239
HPC (The Hasty Pudding Cipher), 164
human vulnerabilities, 14
Hupp, Jon, 85
hyperlinks, 125
Hypertext Markup Language (HTML), 113, 124
Hypertext Transport Protocol (HTTP), 113
- I**
- I LoveYou virus, 79
IBM's Systems Network Architecture (SNA), 181
Icelandic virus, 85
IDEA, 164
identification, 9, 50, 203
identity management, 76
identity theft, 4
IEEE 802.11, 224
IEEE 802.11i standard, 186
IEEE 802.11-type radio systems, 221
IEEE 802.15 (Bluetooth) wireless personal area network (WPAN), 224
IEEE 802.16 wireless metropolitan area network (wireless MAN or WiMAX), 224
independent service set (ISS), 221
Industrial TEMPEST Program, 37
information protection, 20
information security, 10
history, 22–24
Information Sharing and Analysis Centers (ISACs), 5
Information System Security Officer (ISSO), 109
Information Systems Security Association (ISSA), 44
information vulnerabilities, 100
- infrared, 224
infrastructure architecture, 221
infringements on personal privacy, 41
Institute of Computer Science and Technology (ICST), 36
Institute of Internal Auditors (IIA), 44
Integration Definition for Function Modeling (IDEF0), 276
Integration Definition for Information Modeling (IDEFIX), 276
integrity, 9, 11, 196, 203
(see also C-I-A acronym)
integrity of information, 174
interception, 175
International Information Systems Security Certification Consortium (ISC)2, 10, 44
International Standards Organization (ISO), 44
Internet Control Message Protocol (ICMP), 117
Internet Key Exchange (IKE), 197
Internet Protocol (IP) address, 113
Internet Protocol Security (IPSec), 195, 197
internetworks, 180
intruders, hints for keeping out, 208
intrusion detection system (IDS), 107, 175
iris scans, 216
(ISC)2 Certified Information System Security Professional CISSP, 27
ISO 15408 certification, 281
isotropic radiator, 231
ITSEC, 278
- J**
- Java, 127
Java Runtime Engine (JRE), 127
Java Server Pages (JSP), 129
JavaScript, 127
Johns Hopkins epidemiologists, 254
- K**
- Kaiser Permanente researchers, 254
Kerberos, 55, 198
information, 56
key server, 198
Keyed-Hash Message Authentication Code (HMAC), 278
keys, 51, 146–150
distribution problem, 148
management and distribution, 148

keys (*continued*)
private, 146
public, 146
 disadvantages, 147
(see also locks and keys)
keystroke verification, 218
KillerResume virus, 79
Klez virus, 79
Kournikova virus, 79

L

label integrity, 267
labeling human-readable output, 268
labels, sensitivity, 266
LaMacchia, Brian, 165
LAN security, 277
LANs, 182
LaPadula, Leonard, 31, 261
Lavender Book, 259
Law Enforcement Access Field (LEAF), 276
Layer Two Tunneling Protocol (L2TP), 195
layers, 181
Leahy, Patrick, 42
LEAP protocol, 241
least permissive principle, 71
Leeper, Ed, 254
legislation and mandates, 37–45
 balancing act, 38
 Computer Fraud and Abuse Act, 39
 Computer Security Act of 1987, 40
 recent initiatives, 41
leukemia, risks for, 254
life-cycle assurance, 262, 263
lightning, 207
Lightweight Directory Access Protocol
 (LDAP), 72, 74–76
link encryption, 152, 191
Local area networks, 182
Local Multipoint Distribution Service
 (LMDS), 224
locks and keys, 207–212
 active tokens, 210
 challenge-response systems, 210
 cryptographic locks, 210
 equipment locks, 210
 hints for keeping intruders out, 208
 smart cards, 211
 smart keys, 210
 tokens, 210
 types of locks, 209
logging into system, 50–77
logic bombs, 88

logical addresses, 115
login processes, 51–56
 biometrics, 54
Challenge Handshake Authentication
 Protocol (CHAP), 52
Kerberos, 55
mutual authentication, 52
one-time password (OTP), 53
Password Authentication Protocol
 (PAP), 52
per-session authentication, 53
Radius DIAMETER, 55
remote access (TACACS and
 RADIUS), 54
tokens, 53
LOKI97, 164

M

machine access code (MAC) addresses, 116
MAGENTA, 164
magnetic card, 213
magnetic stripe card, 213
malware, 7, 80
 (see also viruses)
 elaborate, 7
MAN systems, 182
mandatory access control (MAC), 18, 66,
 69–70, 266
 access decisions, 69
 data import and export, 69
mantraps, 209
masquerade, 89, 176, 196
media vulnerabilities, 14
Melissa virus, 79
memory and microprocessor tokens, 213
message, 179
message authentication, 11
message digest, 59
metallic strip card, 213
metropolitan area networks, 182
microwave, 186
MiniZip virus, 79
modems, 177
 hints for security, 178
modulation, 220
monitoring wireless networks, 244
Morris worm, 86
Morris, Robert Tappan, 40, 86
Morse, Samuel F.B., 23
Multi-Channel Multipoint Distribution
 Service (MMDS) or “wireless
 cable”, 224

Multics (Multiplexed Information and Computing Service) system, 31
multifactor authentication, 51
multilevel security policy, 69
multipath signals, 233
multiplexing, 114
mutual authentication, 52

N

NACSEM 5100, 37
NakedWife virus, 79
National Bureau of Standards (NBS), 28, 32
National Computer Security Center (NCSC), 34
National Computer Systems Laboratory (NBS), 36
National Infrastructure Security Co-ordination Centre (NISCC), 5
National Institute of Standards and Technology (NIST), 28, 32
National Security Agency, 28
National Security Decision Directive 145 (NSDD 145), 38–41
National Telecommunications and Information Systems Security Publication 2 (NTISSP 2), 39
natural disasters, 205
natural vulnerabilities, 13
negotiate, 179
network access, 175
Network Address Translation (NAT), 123
Network File System (NFS) protocol, 165
network node, 179
network protocols, 116–124
network security, 187–199
access control, 188
auditing, 189
communications integrity, 196
compromise protection services, 197
gateways, 189
heterogeneous computing, 190
Internet Protocol Security (IPSec), 195, 197
Kerberos, 198
Layer Two Tunneling Protocol (L2TP), 195
perimeters, 189
Point-to-Point Tunneling Protocol (PPTP), 194
standards, 187
VPN (see VPN)

networks, 179–186
hints for network security, 184
history, 183
media, 184–186
Neumann, John von, 83
new accounts, 108
NewLove virus, 79
NIMDA virus, 79
NIST, 170
nonrepudiation, 174, 196
notary, 191
NSA, Communications Security Cryptographic Endorsement Program, 170
NSTISSAM/1-91 (Compromising Emanations Laboratory Test Requirements, Electromagnetic), 37
NSTISSAM/2-91 (Compromising Emanations Analysis Handbook), 37
NSTISSAM/3-91 (Maintenance and Disposition of TEMPEST Equipment), 37

O

object reuse, 266
Odlyzko, Andrew, 165
off-line encryption, 151, 191
one-time cipher key, 150–151
one-time pad, 150–151
one-time password (OTP), 53
online encryption, 152, 191
Open Systems Interconnection (OSI), 181
operational assurance, 262
Optical TEMPEST, 256
optical-coded card, 213
Orange Book, 258–273
about, 259–264
birth of, 35
complaints, 273
discretionary access control, 266
documentation requirements, 263
design documentation, 264
Security Features User's Guide (SFUG), 263
test documentation requirement, 264
Trusted Facility Manual (TFM), 264
evaluation classes, 269–273
A1 systems, 273
B1 systems, 271

Orange Book (*continued*)

- B2 systems, 272
- B3 systems, 272
- C1 systems, 269
- C2 systems, 271
- D systems, 269
- labels, 266–269
 - device labels, 269
 - exportation of labeled information, 267
 - exportation to single-level device, 268
 - human-readable output, 268
 - integrity, 267
 - sensitivity, 266
 - subject sensitivity label, 269
- mandatory access control, 266
- object reuse, 266
- purpose of, 260
- rating by the book, 264–269
- security concepts, 260–264
 - accountability, 261
 - assurance, 262
 - life-cycle assurance, 263
 - security policy, 261
 - trusted facility management, 262
 - trusted recovery, 263
- Organization for Economic Cooperation and Development (OECD), 43
- OSI (Open Systems Interconnection)
 - model, 249–251
 - functions, 250
 - layers, 250, 251
- OSI Basic Reference Model, 249
- Output Feedback (OFB), 157
- ownership of files, directories, and devices, 67
- OXCERT, 5

P

- packet sniffers, 119
- packet switching and packets, 114
- packets, 180
- packet-switching networks, 180
- PANs, 182
- parabolic reflector, 231
- Partnership for Critical Infrastructure Security, 44
- passive electronic card, 213
- passive taps, 177
- Password Authentication Protocol (PAP), 52
- password sniffers, 80

passwords, 51, 56–61, 108, 137

- access controls, 61
- cracking, 61
- encryption, 60
- hash, 59
- hints for picking, 58
- hints for protecting, 57
- message digest, 59
- one-time password (OTP), 53
- protecting, 58
- protecting in storage, 60
- sample controls, 59
- shadow password files, 61

Patriot Act, 6, 77

- payload, 82
- PCMCIA (PCCARD), 210
- perimeters, 189
- Perl, 129
- per-session authentication, 53
- Personal area networks, 182
- phantom circuits, 114
- photo ID card, 213
- PHP, 129
- Phrack, 17
- physical addresses, 116
- physical security, 18, 203–207
 - climate, 205
 - earthquakes, 206
 - electricity, 206
 - fire and smoke, 205
 - lightning, 207
 - natural disasters, 205
 - risk analysis and disaster planning, 207
 - water damage, 206
- physical vulnerabilities, 13
- plaintext, 141
- playback, 176
- playback or a replay attack, 196
- plug-ins, 127
- PoE, 236

Point-to-Point Tunneling Protocol (PPTP), 194

- Polish security consultants, 5
- Port Address Translation (PAT), 124
- port numbers, 116
- Presidential Directive Number 63, 6
- Pretty Good Privacy (PGP), 24
- privacy
 - considerations, 44
 - personal, 41
- private keys, 146

protocols, 112, 181
family, 181
model, 181
suite, 181
public key algorithms, 165
public keys, 146–148, 277
disadvantages, 147
Public Law 99-474, 39
pulse modulation (PM), 220
Purple Book, 259
Python, 129

Q

quantum encryption, 142

R

radio communications, 224
radio-based card, 213
RADIUS, 178
Radius DIAMETER, 55
Rainbow Series, 259
RC2, 163
RC4, 163
read down, 189
Red Book, 259
REDOC II, 164
reflector, 231
relative distinguished name (RDN), 76
reliability, 179, 203
packets, 115
remote access (TACACS and RADIUS), 54
remote access servers (RAS), 178
Remote Authentication Dial-In User Service (RADIUS), 54
remote terminal, 179
replay, 176
repudiation, 176
retina patterns, 215
Rijmen, Vincent, 160
Rijndael, 36, 161
(see also Advanced Encryption Standard (AES))
risks, analyzing, 99–102
Ritchie, Dennis M., 87
Rivest, Ronald, 165
roaming, 221
rogue access points, 243
role-based access control (RBAC), 18, 66, 70, 188
rootkits, 80, 81, 132
Rosenberger, Rob, 83

routers, 176
RSA algorithm, 165
RSA cryptographic algorithm, 147

S

safe computing, hints for, 105
SAFER, 163
sandbox, 127
Sarbanes-Oxley Act, 77
Family Educational Rights and Privacy Act (FERPA), 4
satellite, 186, 224
Schmidt, Hans-Thilo, 139
Schoch, John, 85
script kiddies, 27, 91
scripting, 126
scripts, 81, 91
secrecy, 11, 174
secret key, 146
Secure Hash Standard (SHS), 275
Secure Telephone Unit (STU-III) project, 150
security administration, 50
security administrator functions, 110
Security Assertion Markup Language (SAML), 77
security audit, 108
security classification, comparing with subject's security clearance, 64
security consultants, 4–7
international, 5
Security Controls for Computer Systems, 28
Security Features User's Guide (SFUG), 263
security kernel, 31
security label syntax, 277
security model, 30
security policy, 30, 261
administrative security, 97
overall planning and administration, 98–103
establishing and maintaining, 96–111
multilevel, 69
self/group/public controls, 67
semaphore system, 22
sensitivity labels, 62, 266
separation of duties, 109, 262
SERPENT, 164
server key, 56
server-side attacks, 124
web, 131
server-side scripts, 125
languages, 128

Service Provisioning Markup Language (SPML), 77
servlets, 127
session, 179
session keys, 56
shadow password files, 61
Shamir, Adi, 165
SHARK, 163
shell scripts, 126
Shockwave Rider, The, 85
SI-CERT, 5
signaling equipment, 176
signature-writing pattern, 218
Simple Mail Transfer Protocol (SMTP), 113, 119
simple security condition, 65
SingCERT (Singapore CERT), 5
single key, 146
single-level device, 268
SKIPJACK, 164
Slovenian Computer Emergency Response Team, 5
smart cards, 51, 211
smart keys, 210
smashing the stack, 132
sniffers, 80, 81
Softwar: La Guerre Douce, 88
software security tools, 107
software vulnerabilities, 13
source suppression, 255
Spafford, Gene, 83
spam, 120
spoofing, 89, 119
spread spectrum systems, 221
Spring Joint Computer Conference of 1967, 28
spyware, 23
SQUARE, 164
standardization, 32–37
cryptography, 36
emanations, 37
Orange Book (see Orange Book)
star property, 65
state two-digit numeric codes, 274
stealth, 244
Steinauer, Dennis, 101
Stoll, Cliff, 57
subject sensitivity label, 269
substitution ciphers, 143, 144
summarizes, 264
suspicious user activity, 109
switching systems, 176

symmetric key, 146
system access, 50–77
system access controls, 49
system administration, 50
day-to-day (see day-to-day system administration)

system administrator/operator functions, 109
system design, 50

T

TACACS, 178
tag abuse, 130
Tan Book, 259
taps, 177
TCSEC, 278
TEA, 163
telephone connections, 176
TEMPEST, 19, 242, 252–257
Endorsement Programs (TEP), 256
Faraday screen, 255
Industrial TEMPEST Program, 37
initial standards, 37
NACSEM 5100, 37
Optical, 256
Products Program, 256
program overview, 253
security, containment approach, 255
source suppression, 255
standards, 256
Test Services Program, 256
Zoned Equipment Program, 256
Temporal Key Integrity Protocol (TKIP), 186, 241
terminal, 179
Terminal Access Controller Access Control System (TACACS), 54
terrorists, 16
test documentation requirement, 264
testing equipment, 176
Thompson, Ken, 84
threats, 12–18
countermeasures, 12
insiders and outsiders, 17
intentional and unintentional, 15
natural, 15
physical, 15
vulnerabilities, 12
tickets, 56, 198
tiger teams, 29
time bombs, 88

time-to-live (TTL), 117
tokens, 51, 53, 210
 active, 210
topology, 180
traffic analysis, 197
training users, 103
tranquility property, 65
Transmission Control Protocol/Internet Protocol (TCP/IP), 112
Transport Control Protocol (TCP), 115
 attacks, 118
transposition ciphers, 143, 144
trap door, 89
Trivial File Transfer Protocol (TFTP), 117
Trojan horses, 80, 81, 87
 bombs, 88
trunk, communications, 113
Trusted Computer System Evaluation Criteria (TCSEC), 258
Trusted Database Management System Interpretation (TDI) (Lavender Book), 259
trusted distribution, 263
trusted facility management, 262
Trusted Facility Manual (TFM), 264
Trusted Network Interpretation (TNI) (Red Book), 259
trusted recovery, 263
trusted system, 265
tunneling, 191
Turing, Alan, 141
twisted pair cable, 185
two key, 146
two-factor authentication, 208
TWOFISH, 164

U

U.K. Data Protection Act and Regulation of Investigatory Powers Bill, 77
U.S. Communications Security (COMSEC), 27
U.S. Computer Departments Technology Administration, 274
U.S. Department of Homeland Security, 5
United Kingdom Department of Trade and Industry, 43
unity gain antenna, 231
unreliability, 180
USB form factor devices, 210
US-CERT, 5
User Datagram Protocol (UDP), 114

V

van Eck, Wim, 241
vertical polarization, 227
Virtual Private Network (VPN), 173, 175
viruses, 7, 23, 79–94
 antivirus programs, 92
 creators, 90–92
 financial effects of, 79
 firewalls, 92
 history, 83
 hype, 93
 overview, 81
 prevention, 94
 public health and, 80
 remedies, 92–93
VLF (very low frequency), 254
Voice over Internet Protocol (VoIP), 91
voice verification, 217
VPN
 inside firewall, 194
 internetworking, 194
 remote access, 194
 tunneling protocols, 194
vulnerabilities, 12
 communications, 14
 emanation, 14
 exploiting, 15
 hardware, 13
 human, 14
 media, 14
 natural, 13
 physical, 13
 software, 13

W

WAN systems, 181
war chalking, 240
war driving, 239
war flying, 239
war walking, 240
Ware, Willis H., 28
warm sites, 102
water damage, 206
Web attacks, 112–133
Web services, advanced, 125
Wertheimer, Nancy, 254
wide area networks, 181
Wi-Fi Protected Access (WPA), 186, 241
Wired Equivalent Privacy (WEP), 186, 241
wireless, 186
wireless cards, 221

- wireless costs, 223
wireless local area networks, 182
wireless network security, 220–244
 access points (APs), 221
 ad hoc infrastructure, 221
 AES protocol, 241
 background, 220
 basic service set (BSS), 221
 bridge, 223
 channels, 244
 cloak, 244
 decibel (dB), 232
 electrical fields, 228–231
 director, 231
 dish antenna, 231
 parabolic reflector, 231
 reflector, 231
 Yagi, 231
 electrostatic vector, 233
 how wireless works, 225–227
 independent service set (ISS), 221
 infrastructure, 221–225
 infrastructure architecture, 221
 LEAP protocol, 241
 monitoring wireless networks, 244
 multipath signals, 233
 physical layer wireless attacks, 233–244
 advanced eavesdropping attacks, 241
 eavesdropping, 239
 eavesdropping defenses, 240–241
 forced degradation attacks, 238
 hardening access points, 234–236
 PoE, 236
 rogue access points, 243
 sophisticated attacks, 237
 roaming, 221
 stealth, 244
 Temporal Key Integrity Protocol (TKIP), 241
 why it matters, 232
 Wi-Fi Protected Access (WPA), 241
 Wired Equivalent Privacy (WEP), 241
 wireless cards, 221
 wiretaps, 23
 WLANs, 182
 worms, 7, 85
- X**
- X.500, 72–75
Xerox Network System (XNS), 181
Xerox PARC worms, 86
XNTP Network Time Protocol exploits, 133
XOR, 150
- Y**
- Yagi, 231
- Z**
- Zero Day problem, 93
zombies, 80, 81
Zoned Equipment Program, 256

About the Author

Rick Lehtinen started his career in electronic communications, from installing two-way radios to building television stations and satellite uplink facilities. Frequent contribution of articles to the trade press led to a position as an editor at *Broadcast Engineering* magazine, as well as a sister publication, *Video Systems*, which was followed by nationwide travel promoting desktop video production as “Dr. Video.”

After relocating to Phoenix, Lehtinen researched and produced marketing information for the semiconductor industry. This led to a survey of computers and networking, which eventually became his passion. Lehtinen financed his studies by teaching and writing, and is on the adjunct faculty of several community colleges and vocational schools. As a course writer for the Cisco Network Academy, he has worked with materials from conception through post-production, including learner assessment and the creation of annoying multiple choice questions. Lehtinen holds certifications in computer security (CISSP), networking (CCNP, CCDP), and telecommunications design (BICSII RCDD). His hobbies include spending time with his family, playing the tuba, and writing about complimentary/alternative medical technologies.

Colophon

The tool on the cover of *Computer Security Basics*, Second Edition, is a key. Locks and keys were first developed by the Egyptians for use on doors. The Egyptian wooden door lock was opened using a long wooden implement with thin wooden pegs of varying lengths set in one end. When inserted and lifted, the pegs on this key would raise similar pegs within the lock. When all the pegs were raised past a certain point, the door would open.

Metal locks and keys were developed by the Romans using a design similar to that of the Egyptian door lock. Over time, the Romans refined this lock technique and embellished both locks and keys; locks were camouflaged as animals, flowers, or birds. Keys were highly decorated and frequently worn as pendants.

Portable locks, otherwise known as padlocks, were invented by the early Chinese, Turkish, Indians, and Russians. In the late 16th century this was taken a step further with the invention of keyless locks, otherwise known as combination locks. Further refinement continued throughout the 20th century as greater precision in metal work allowed for closer parts tolerance and therefore greater security.

The cover image is from The Dover Pictorial Archive. The cover font is Adobe ITC Garamond. The text font is Linotype Birk; the heading font is Adobe Myriad Condensed; and the code font is LucasFont’s TheSans Mono Condensed.

