ANALYZING COMPUTER SECURITY

This page intentionally left blank

ANALYZING COMPUTER SECURITY

A THREAT / VULNERABILITY / COUNTERMEASURE APPROACH

Charles P. Pfleeger
Pfleeger Consulting Group

Shari Lawrence Pfleeger
Dartmouth College

Upper Saddle River, NJ • Boston • Indianapolis • San Francisco
New York • Toronto • Montreal • London • Munich • Paris • Madrid
Capetown • Sydney • Tokyo • Singapore • Mexico City

Library of Congress Cataloging-in-Publication Data

Pfleeger, Charles P., 1948–
Analyzing computer security : a threat/vulnerability/countermeasure approach / Charles P. Pfleeger, Shari Lawrence Pfleeger.
p. cm.
Includes bibliographical references and index.
ISBN 978-0-13-278946-2 (hardcover : alk. paper)
1. Computer security. 2. Data protection. I. Pfleeger, Shari Lawrence. II. Title.
QA76.9.A25P4485 2011
005.8—dc23

2011013943

ISBN-13: 978-0-13-278946-2
ISBN-10: 0-13-278946-9

Text printed in the United States on recycled paper at Courier in Westford, Massachusetts.
Second printing, December 2011

Publisher
Paul Boger

Acquisitions Editor
Bernard Goodwin

Managing Editor
John Fuller

Full-Service Production Manager
Julie B. Nahil

Project Manager
LaurelTech

Copy Editor
Mary Lou Nohr

Proofreader
LaurelTech

Editorial Assistant
Michelle Housley

Cover Designer
Chuti Prasertsith

Compositor
LaurelTech

Contents

Foreword

Security and privacy are basic human desires. Sometimes those desires are stronger than at other times, and as humans we are often inconsistent in our choices. People seem to have a pretty good idea of when they want security, such as protection against harm from bears or bullies, or against property loss when executing bank transactions. Their interest in privacy also seems clear at times, such as when sending a resume to a prospective employer or accessing grades on a university computer system. However, at other times, people exhibit somewhat less security- or privacy-conscious behavior, such as going bungee jumping or in their use of loyalty cards when shopping. Sometimes the desire for security or privacy varies across nearly identical circumstances. An example of this is when a person receives an email from an unknown party, he might be reluctant to click on an included hyperlink, but when he receives the same email from a friend he happily clicks the hyperlink. These examples illustrate that people decide consciously whether the perceived value they're receiving (adrenaline rush, loyalty points, convenience) exceeds any decrease in security or privacy. It can also be the case that people make these decisions based upon faulty information or when they are unaware of all the relevant facts.

Sometimes the overriding perceived value is simply that the service being provided is convenient or fast. Grocers have certainly demonstrated that customers are willing to disclose their shopping habits in exchange for the convenience of a loyalty card to avoid the inconvenience of clipping, storing, and transporting paper coupons.

When people take the time to think about it, they want to feel as secure and private as the situation seems to warrant. As with most things, the experience one has will color the assessment of the situation. Just as an experienced mechanic would be less intimidated by driving across the country with a balky engine in his prized classic sports car, a skilled practitioner knows just how secure or private a situation is. People assume and trust that security and privacy will be attributes of a system, but they may be disappointed. Often they do not make security and privacy explicit requirements, instead leaving them as implicit assumptions. When that trust is violated, people may be surprised and often a little annoyed, but only if they actually know about the failure and understand its full impact.

Beyond the traditional security and privacy protections provided by the government, such as law enforcement, emergency services, or social services, people are increasingly dependent upon computer and network systems for their happiness and livelihood, often without their knowledge. As a result, the issues surrounding security and privacy are also increasingly complex and confusing. The good news is that this increased dependence has led to some clarity around two points and perhaps a few answers.

CYBER INFRASTRUCTURES

A new set of computerized and networked systems has emerged: many natural and man-made systems are becoming increasingly instrumented, interconnected, and intelligent. These systems, and the man-made infrastructures that use them, are making people's lives safer and more predictable and comfortable. At the same time, they are enabling innovation in reliable and cost efficient energy, better transportation systems, improved agriculture systems, more effective medical service delivery, and many other areas.

The industries that operate the critical infrastructures are primarily from the private sector[1] and are adopting these technologies at an accelerating rate. The improved efficiencies, cost savings, and market access that these systems promise are irresistible. On the other hand, the millions of people affected by this move to "cyber infrastructures" are either unaware of it, or welcome it and the promise of better prices, efficiency, and reliability.

The improvements in efficiency, service, and reliability are indeed attractive. Yet, as with any technological move, there are some concerns.

Many of the critical infrastructures consist of widely distributed systems utilizing subsystems of sensors and actuators to monitor and manage the infrastructure. These subsystems may not have been designed to be connected to a network or to provide any proof of their identity or location. Since such subsystems were not expected to be replaced for years or even decades, upgrading them to more powerful and potentially secure versions will be slow and expensive. In addition, since these subsystems have not been connected to the open Internet before, connecting them now subjects them to a whole new set of threats and vulnerabilities without the benefit of suitable controls.

Many of these subsystems were designed for use on private, non-carrier class networks using proprietary protocols. Much of their security and privacy was implicitly provided by the network. For cost savings these subsystems are being moved to public networks, standard protocols, and open interfaces. As this move occurs, those implicit capabilities should become explicit requirements, lest new or previously shrouded vulnerabilities be exposed.

Many of these same subsystems employ embedded nontraditional operating systems that were most likely developed without strong security in mind, leaving the operating system itself vulnerable to attack from the open Internet as well as insiders.

Concerns about security and privacy are new to most vendors in the highly competitive critical infrastructure industry. As a result, they may make some poor assumptions about what is effective security. For example, having the same short cryptographic key stored into all of a power company's residential electric meters doesn't add much security since the key will likely be guessed and publicized.

Finally, the people in the traditional IT security industry have had to learn that there are some very basic differences between their world and that of the cyber infrastructure. For example, the traditional security goals of confidentiality, integrity, and availability are not necessarily equals in the cyber infrastructure. There, the highest priority is more often availability, since by definition a critical infrastructure is, well, critical, and so the challenges of reliability and fail-safety rise in importance.

This rapid spread of computing and networking into critical infrastructures raises very clear concerns about security and privacy. While more traditional IT security and privacy concerns have been the subject of research and development for many years, there the risks were primarily financial, taking such forms as plummeting stock prices after a very public hack, successful corporate espionage, or a loss of market share due to a drop in customer confidence after a private information leak. When a critical infrastructure has a failure, the effect can vary from widespread inconvenience, to loss of life or property, or a threat to national or international security.

Multidisciplinary Nature of the Problem

The traditional response is to throw more technology at new security and privacy problems. More technology may very well help, especially with the huge challenge of building systems that are secure by design, addressing the needs of legacy systems, and finding ways to actually use the flood of new data pooling around computing systems today.

However, adding more and more technology to the mix has failed to keep up with the rate of new threats and vulnerabilities. In fact, some would argue that the constant addition of more technology has exacerbated the problem by adding complexity, more implicit assumptions, and more vulnerabilities, all of which could potentially interact. Instead of relying solely on technology, people have come to realize that a multidisciplinary approach would be more effective.

For example, Joel Brenner, past U.S. National Counterintelligence Executive and Inspector General for the National Security Agency, said in an article in *Communications of ACM*[2] that "[c]hange in the U.S. is driven by three things: liability, market demand, and regulatory (usually Federal) action." There are now regulations that hold organizations liable for the loss of personal information, and several large cases have been prosecuted and damages assessed. However, liability hasn't played much of a role in driving improvements to cyber security since it has proven difficult to ascertain precisely what was warranted, what failed and for what reason, and who should pay. Some worry that liability would squelch innovation and agility, especially in smaller companies.[3] As for market demand, sometimes the public doesn't realize that a security or privacy breach has occurred, or the degree to which such a breach affects their lives. As a result, the public has only made lukewarm, episodic demands for improvements in cyber security and has a very short memory for past incidents.

So perhaps one way out is via regulatory support. If regulations were to come, the challenge will be to find a way for them to be

- Relevant—addressing a problem that really matters
- Meaningful—addressing the identified problem in an effective manner
- Enforceable—preventing violations or enabling detection and prosecution of violators

While a multidisciplinary approach (combining technology, legislation, and market pressures) will help us move forward, one underlying challenge of security and privacy is common to all of them: People are the ultimate critical infrastructure. Every day, people decide whether to do something in a secure or private way. If it's too difficult, slow, annoying, or otherwise costly to do something securely, most people will almost invariably elect to do it unsecurely. This is especially true when it comes to a person getting his or her tasks done at work. Whether at a school or a bank or an electric power distribution company, if the job depends on performance of a task, people will find a way around anything that gets in the way, including security. Thus, people are both part of the problem and a key part of any multidisciplinary solution.

Whether the threat is from bears or bullies, people need a clear understanding of their vulnerabilities to the threats and the controls they can exert upon them. The real challenge is that in today's interconnected, interdependent, and intelligent world, the threats are many, the vulnerabilities are constantly growing, and the controls must be agile, effective, usable, and expand beyond technology to address the multidisciplinary facets of the challenge.

This is where teachers and authors like Chuck and Shari Pfleeger are indispensable.

In this book, the authors adopt a new approach to explaining the intricacies of the security and privacy challenge, one that is particularly well suited to today's cyber security challenges. Their use of the threat–vulnerability–countermeasure paradigm, combined with extensive real-world examples throughout, results in a very effective learning methodology. The examples illustrate the "hows" of the particular issue being examined, the reliance on implicit assumptions, the indications of a failure, and the spectrum of real-world impacts that successful security and privacy breaches have had. With each of the discussions, the authors include effective strategies for addressing the problems and methodologies for avoiding them in the first place. This refreshing new approach helps the reader not only to understand the problems, but also to gain a deeper appreciation of why they occur and why they are important topics of study. The authors have provided a comprehensive treatment of the important aspects of security and privacy in computing, thoroughly preparing the reader for the new multidisciplinary security and privacy challenges to come.

—Charles C. Palmer
IBM Research
Yorktown Heights, NY

NOTES

1. "Office of Infrastructure Protection: Mission," http://www.dhs.gov/xabout/structure/gc_1185203138955.shtm, viewed on April 4, 2011.
2. Joel F. Brenner, "Why isn't cyberscape more secure?," *Communications of the ACM*, pp. 33–35, Vol. 53, No. 11, November 2010.
3. Robert E. Litan, "The Safety and Innovation Effects of U. S. Liability Law: The Evidence," *The American Economic Review*, pp. 59–64, Vol. 81, No. 2, Papers and Proceedings of the Hundred and Third Annual Meeting of the American Economic Association, May 1991.

Preface

Computer technology surrounds us; from mobile phones to digital cameras, and hybrid vehicles to laser surgery, we achieve results that would be impossible without computers. On any given day you probably interact with computer-controlled or computer-assisted devices tens or hundreds of times, generally without even thinking of the computing involved. And this discussion does not even include the laptops, desktops, netbooks, and other actual computers we use, let alone the Internet and its vast oceans of data. Of course, we could do without all these things, but our lives would be different.

At the same time, as we become more accustomed to and integrated with computers, their weaknesses become our weaknesses. If you lost power, you could write a report by hand or on a manual typewriter, but the process would be a challenge; you might be relieved when the power went back on. You do not worry about changes to paper documents, as long as they are protected from physical hazards such as fire or deterioration, but you must guard against accidentally modifying a file or losing it because of a power surge. When you share a secret with a friend, you do not worry that your secret will become public if someone takes a picture of your friend, but you do need to prevent your files from being copied without your permission. Our use of computer technology has brought with it certain risks.

This book is about bad things that can happen with computers and ways to protect our computing. The title *Analyzing Computer Security* should alert you that this book is intended to help you develop a way of thinking critically about computers and their security.

WHY READ THIS BOOK?

You do not learn algebra by memorizing the names of famous mathematicians or learning the Greek alphabet. You learn algebra by studying its principles, techniques, and results. And then you work problems … lots of problems. You get to the point where you can set up the equations for a mixture problem before you even finish reading or hearing the problem statement. Solving two equations in two unknowns becomes easy. But these tasks were really challenging the first time you did them.

Now let us consider a different kind of learning: completing a crossword puzzle. At the beginning you may have had trouble filling in any cells. Gradually you learned tricks: a plural is likely to end in S, Q is usually followed by U, two Js together may indicate a mistake. Gradually, your analytic skills developed and you may have found you could solve harder puzzles. In a way, you began to think like the person who wrote the puzzle.

This book will do the same kind of thing for you with respect to the security of computers and data: It will make you aware of how such systems can fail—or be made to fail—and how to protect yourself and your use of computing. You will start to look at computing as would an attacker. Your question becomes not *How can I make this work?* but *How could this fail?* Only by figuring out the failure modes can you decide how to protect yourself.

For these reasons, the threat–vulnerability–countermeasure approach is the basis of our presentation. Each chapter starts with an attack, from which we challenge you to develop your ability to identify people or things that could cause harm, locate the weaknesses against which they would work, and learn about the protective tools of the computer security community. For more than forty years, the leaders in our field have been developing a vast array of defenses that we will share with you. Just as with algebra, you need to know the tools of the field, but you also need to develop the insight that guides when to apply which tool.

Who Should Read This Book?

Three groups of people can profit from reading this book: students, computing professionals, and users.

College and university students can use this book in a one- or two-semester course on computer and information security. It covers the most important points such courses address, such as network security, application code, identification and authentication, access control, and operating systems. You will find the expected topics of firewalls, intrusion detection and protection systems, cryptography, viruses, and secure programming techniques, as well as many others. We think you will learn how, when, and why to apply these things for the most benefit.

Computing professionals may have a different context and focus from that of college students. Whereas many students want the full development of the subject, you as professionals may be more comfortable diving into the middle, to learn about a topic that is of immediate importance. From that topic, you can move to neighboring topics that are relevant, or pick another topic in which you have an interest. Although the book has a front-to-back progression, we point to other chapters that have material relevant to what you are currently reading, so you can feel comfortable starting at your point of interest and referring back if you find a concept you need to learn more about.

Computer users can easily find the language of computer security mystifying: Viruses, teardrop attacks, bots, drive-by downloads, backdoors, and rootkits sound dreadful, which they can be, but underneath they are just words to describe methods attackers use to harm you. To protect yourself, ignore the colorful language and focus instead on what valuable things of yours are at risk and how you can defend yourself.

You will find not just definitions of these terms but also examples to which you can relate.

We wrote this book to be useful to all three kinds of readers.

What Will You Learn From This Book?

From this book you will learn how to think critically and creatively about security. Anyone can memorize facts, but mere facts will not address the constantly changing situations in computer security. You need to be able to look at new programs, technologies, requirements, data collections, and objects with an eye for how their security can fail and how those potential failures can be countered.

As you read this book you will encounter many examples: some old, some very recent. We even mention some situations from the days before computers, to amplify or demonstrate a point we want you to understand.

ROADMAP

As you look at the Contents you will not find a networks chapter or the cryptography section or even the privacy pages. That is because computer security, like many disciplines, has interrelationships. We have chosen to work with, rather than against, those connections.

How Is This Book Structured?

We think you will find this book intriguing. We have laid it out in a rather nontraditional way for a textbook, but the structure is designed to help you learn to think critically about security.

Think for a moment of a history book, for example, about the nineteenth century. One conventional way to present history is chronologically: Start at the beginning in 1800 and work by date through all the major events until 1900. That organization is familiar because that is the way our lives unfold, but it is not the only way to present history. Another way to appreciate history is to observe the changes in society. For example, we could look at how artists abandoned realism and classicism for impressionism. We could analyze how inventions and the Industrial Revolution changed the nature of work, or how small city-states united to form large nations. Just as photography lets people see and record events that had formerly been represented only in words, so do we seek to view security through a lens that will help you understand its principles.

Threats–Vulnerabilities–Countermeasures

The lens we have chosen is the threat–vulnerability–countermeasure paradigm. Computer objects are subject to threats from attack sources; those attacks aim to exploit weaknesses or vulnerabilities; and we can take action to protect against the harm those threats could cause. We use case studies to illustrate each attack type.

We have picked real examples for our case studies. In some cases there was an obvious failure: a human error, technology failure, misunderstanding, or an oversight.

We assure you, these failures may be obvious in retrospect, but they were not so apparent before the event. That is precisely the point of this book: You should develop the ability to analyze a situation outside this book, to determine what threats could be raised, what vulnerabilities exploited, and what countermeasures employed. From studying the examples in this book and our explanations, you will acquire both the tools to use as countermeasures and the experience to guide your thinking.

Mapping

In case you want to find a particular topic, Table P-1 shows you where some of the conventional topics of computer security are covered. (This table shows only main locations for these topics.)

TABLE P-1 Conventional Topics and Where They Appear in This Book

Topic	Chapters
Threats, vulnerabilities, and countermeasures	1: Definitions All other chapters: Examples
Identification and authentication	2: Basic concepts 12: Shared secrets, one-time passwords
Cryptography	4: Cryptographic checksums 7: Symmetric encryption 10: Cryptographic weaknesses (WiFi protocols) 11: Key management; asymmetric cryptography 13: Digital signatures, public key infrastructure, code signing 14: SSL, IPsec 16: Block chaining
Malicious code	4: Viruses, Trojan horses, worms 6: Buffer overflows 8: Rootkits 12: Man-in-the-middle attacks, covert channels 15: Denial-of-service attacks, distributed denial of service attacks
Network security	9: Network architecture 9: Firewalls 10: WiFi vulnerabilities 11: Interception 14: Replay attacks; session hijacks 15: Intrusion detection systems
Operating systems	4: Memory separation 6: Memory management 8: Rootkits and operating system subversion, trusted operating systems
Secure software development	3: Techniques 3: Testing 6: Error prevention

Topic	Chapters
System design	5: Security through obscurity 6: Access control models and enforcement 8: Simplicity of design, trusted system design 9: Layered protection 17: Peer-to-peer network model
Assurance	8: Trusted systems 13: Forgeries
Privacy	2: Identities and anonymity 17: Unexpected data distribution 18: Social media applications, inference, and aggregation

Expected Background

What background do you need to appreciate this book? We assume you understand programming, machine organization, operating systems, and networking. We give some background in each of these topics where we introduce them, but because these are the topics of entire books and courses, we cannot really cover all that background in this book. A student in a computer science program or a professional designer or developer probably has most of the background necessary or can check a reference for any needed explanation.

How Does This Book Relate to *Security in Computing*?

You may have seen *Security in Computing*, of which the most recent edition was published in 2007. This book began as a revision; however, as it took shape, we realized it was a dramatically different book. True, both books address many of the same topics, and you will even see some overlap because, for example, there are only so many ways you can explain authentication.

However, not only does this book have more recent coverage of emerging topics, the objectives and structure are completely different. If you want encyclopedic coverage of computer security in a taxonomic progression, you want *Security in Computing*. However, we think a significant number of people will like the analytical approach of this book, so we offer it as an alternative for people who want to be able to identify security weaknesses in any situation and know tools and techniques by which to counter those weaknesses.

IN THE CHAPTERS

Let us now explain how the individual chapters are laid out.

Spotlights

Each chapter begins with a spotlight: a handful of bullet points to tell you the major topics that will be covered in the chapter. This lets you quickly know what you will

find in a chapter, so if you want to skip around in the book, this block will give you a simple guide.

Threats–Vulnerabilities–Countermeasures

We use the same format for each chapter: a case, explanation of the threats, enumeration and expansion on the vulnerabilities, and statement and development of the countermeasures.

Recurring Threads

Some topics are relevant to computer security; we would be remiss if we did not raise them at appropriate points. These topics are privacy, ethics, law and law enforcement, forensics, management, and economics. We pay attention to these topics at points when they are especially relevant in sections labeled "Recurring Thread."

Sidebars

Sometimes we want to view a point from a different perspective, show a historical parallel, or tell an interesting story. We do these things in Sidebars. They are set off typographically so you can tell they are interruptions to the normal flow of content.

Interludes

We have added three mini-chapters to give you a chance to apply the analytic skills you will learn. We call these pieces Interludes, and they raise issues related to cloud computing, electronic voting, and cyber warfare. Currently in an early stage of development, each of these is an important area that we expect will gain in prominence in the future. Although people are beginning to address the security issues for these areas, more analysis and implementation remain to be done.

The Interludes challenge your analytical skills. In each Interlude we lay out the topic and ask some pointed questions for your consideration. However, we leave the bulk of the work to you: Who would have method, opportunity, and motive to attack? What would be the nature of the attack? What harm could occur? Where might there be vulnerabilities that could be exploited? How difficult would an attack be? And what countermeasures could or should be applied now to render each of these situations more secure in the future?

Conclusions

We conclude each chapter by briefly reviewing the salient points, summarizing the current state of and future issues for the chapter's topic, and tabulating the key threats, vulnerabilities, and countermeasures of the chapter.

Exercises

At the end of each chapter you will find a set of exercises. Many of the exercises call for you to analyze, describe, or justify something. You can do these exercises mentally or in writing, and you can use some as debate topics for friends, students, or colleagues.

Afterword

We end the book with a last, unnumbered chapter, to describe where we think the field of computer security is heading. Crystal balls are notoriously cloudy, and we do not think our ability to predict the future is exceptional. Still, this book has pointed out some security strengths and weaknesses in our current computing environment, and we use the Afterword to recommend things to which the community should pay attention.

ACKNOWLEDGMENTS

It is increasingly difficult to acknowledge all the people who have influenced this book. Many colleagues and friends have contributed their knowledge and insight, often without knowing their impact. By arguing a point or sharing explanations of concepts, our associates have forced us to question or rethink what we know.

We thank our associates in at least two ways. First, we have tried to include references to their written works. References in the text cite specific papers relating to particular thoughts or concepts, but the Bibliography also includes broader works that have played a more subtle role in shaping our approach to security. So, to all the cited authors, many of whom are friends and colleagues, we happily acknowledge your positive influence on this book.

Rather than name individuals, we thank the organizations in which we have interacted with creative, stimulating, and challenging people from whom we learned a lot. These places include Trusted Information Systems, the Contel Technology Center, the Centre for Software Reliability of the City University of London, Arca Systems, Exodus Communications, the RAND Corporation, Cable & Wireless, and the Institute for Information Infrastructure Protection. If you worked with us at any of these locations, chances are high that your imprint can be found in this book. And for all the side conversations, debates, arguments, and light moments, we are grateful.

We want to recognize and thank three people for their particular, significant contributions to this book. Mischel Kwon first suggested to us the idea of studying security by exploring threats, vulnerabilities, and countermeasures. As we picked up and began to expand that idea, she offered valuable constructive criticism, as well as friendship and encouragement. We similarly appreciate the contributions of Charles Palmer. In addition to writing the Foreword to this book, Charles has been a great friend and colleague who has gladly shared his insights. We also thank Bernard Goodwin, our editor at Prentice Hall, who has been a solid champion during development of this book.

This page intentionally left blank

About the Authors

Charles P. Pfleeger is an independent consultant with the Pfleeger Consulting Group, specializing in computer and information system security. Among his responsibilities are threat and vulnerability analysis, risk analysis, system security design and review, certification preparation, training, expert testimony, and general security advice. His customers include government and commercial clients throughout the world.

Dr. Pfleeger was previously a Master Security Architect on the staff of the Chief Security Officer of Cable & Wireless, and Exodus Communications, and before that he was a Senior Computer Scientist and Director of Research for Arca Systems, Director of European Operations for Trusted Information Systems, Inc. (TIS), and a professor in the Computer Science Department of the University of Tennessee.

Dr. Pfleeger was chair of the IEEE Computer Society Technical Committee on Security and Privacy from 1997 to 1999 and has been a member of the executive council of that committee since 1995. He is on the board of reviewers for *Computers and Security*, and was a member of the editorial board of IEEE *Security and Privacy* and the board of advisors for OWASP, the Open Web Application Security Project.

Dr. Pfleeger has lectured throughout the world and published numerous papers and books. His book *Security in Computing* (of which the fourth edition—coauthored with Shari Lawrence Pfleeger—was published in 2007) is the standard college textbook in computer security. He is the author of other books and articles on technical computer security and computer science topics.

He holds a Ph.D. in computer science from The Pennsylvania State University and a B.A. with honors in mathematics from Ohio Wesleyan University. He is a Certified Information Systems Security Professional (CISSP).

Shari Lawrence Pfleeger is the Research Director for Dartmouth College's Institute for Information Infrastructure Protection, a consortium of leading universities, national laboratories, and nonprofit institutions dedicated to strengthening the U.S. cyber infrastructure. She joined the I3P after serving for nine years as a senior researcher at the RAND Corporation, where her work focused on software quality and cyber security.

Previously, as president of Systems/Software, Inc., she led a consultancy specializing in software engineering and technology. She has been a developer and maintainer for real-time, business-critical software systems, a principal scientist at MITRE Corporation's Software Engineering Center, and manager of the measurement program at the Contel Technology Center. She has also held several research and teaching positions at universities in the United States and United Kingdom.

Named repeatedly by the *Journal of Systems and Software* as one of the world's top software engineering researchers, Dr. Pfleeger is the author of more than one hundred articles and many books, including *Security in Computing, Fourth Edition* (with Charles Pfleeger), *Software Engineering: Theory and Practice, Fourth Edition* (with Joanne Atlee) and *Solid Software* (with Les Hatton and Charles Howell). She has testified before Congress on cyber security risk, and often appears in the media. She has been associate editor-in-chief of *IEEE Software*, associate editor of *IEEE Transactions on Software Engineering*, and is currently an associate editor of *IEEE Security & Privacy*. Dr. Pfleeger was also the founding chair of ACM's Committee on the Status of Women and Minorities.

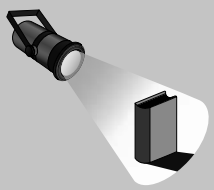
Dr. Pfleeger earned a B.A. in mathematics from Harpur College, an M.A. in mathematics from Penn State, an M.S. in planning from Penn State, a Ph.D. in information technology and engineering from George Mason University, and a Doctor of Humane Letters from Binghamton University.

In their spare time, you can find both Pfleegers on their bicycles or doing volunteer work in the Washington, D.C., area.

This page intentionally left blank

1

Security Blanket or Security Theater?



Chapter Spotlight

- Computer security: the threat–vulnerability–countermeasure paradigm
- Harm to confidentiality, integrity, availability
- Nature of threats; method–opportunity–motive; risk
- Kinds of attackers
- Countermeasure effects: prevent, deter, detect, recover
- Analyzing security

Imagine a series of events unfolding on a single day. First, 20 million U.S. smart phones stop working. Next follow outages in wireline telephone service, problems with air traffic control, disruptions to the New York Stock Exchange, and eventually severe loss of power on America's East Coast. What could cause such crippling outcomes?

You might think first they are isolated events, just coincidentally occurring on the same day. But with several things happening at once, you next start to look for common causes. Perhaps the various organizations providing these services bought some of their software from the same vendor, and the software is failing because of a shared flaw. Possibly this situation is like the Y2K problem, when people were concerned that on January 1, 2000 computer systems would crash because they used only two digits for the date (98, 99) and would fail when computer clocks rolled over the year boundary. Or maybe dependencies in one sector trigger actions that cause the initial failure to cascade into other sectors, for example:

1. A software defect causes disruption in mobile phone service.
2. Consequently, those who need to use phones revert to their wireline service, thereby overloading circuits.
3. Air traffic controllers in some parts of the country depend on wireline communication, so overloaded circuits lead to air traffic control problems.
4. Similarly, the New York Stock Exchange is severely debilitated by its brokers' inability to place and verify trades.
5. At the same time, the power grid experiences problems because its controllers, no longer able to exchange information by using mobile phones, shut down because of a flawed protocol.

There is yet another scenario, used by the Bipartisan Policy Center in its February 2010 Cyber ShockWave exercise: malicious computer software or malware, "planted in phones months earlier through a popular 'March Madness' basketball bracket application, disrupts mobile service for millions" [BPC10].

It is difficult—sometimes impossible—to distinguish between an accident and an attack. Consider, for example, an online gambling site that received a flood of blank incoming email messages that overwhelmed servers and slowed customer traffic to a crawl. Blank messages could easily come from a software or hardware problem: a mail handler caught in a loop with one malformed message that it dispatches over and over. Shortly thereafter, the company received email written in broken English. It told the company to wire $40,000 to ten different accounts in Eastern Europe if it wanted its computers to stay online [MCA05]. So much for the "just an accident" theory.

Are these scenarios realistic or implausible? And are cyber security exercises such as these and the ones described in Sidebar 1-1 designed to confirm our readiness (a security

Testing Cyber Security Readiness — Sidebar 1-1

Governments and the private sector have organized many "cyber security exercises." Although the nature of each exercise varies, the goals of such exercises are similar: to anticipate unwelcome cyber events so that prevention and mitigation plans can be made, to make both public and private officials aware of cyber security risks, and to test existing response plans for both coverage and effectiveness.

For example, in November 2010, the European Union ran its first cyber security "stress test," Cyber Europe 2010. Its objective was to "test Europe's readiness to face online threats to essential critical infrastructure used by citizens, governments and businesses." The activities involved 22 participating nations and 8 observers. Among the lessons learned:

- The private sector must be involved.
- Testing of pan-European preparedness measures is lacking because each member nation is still refining its national approach.
- The exercise is a first step in building trust at a pan-European level. More cooperation and information exchange are needed.
- Incident handling varied a lot from one nation to another because of the different roles, responsibilities, and bodies involved in the process. Some nations had difficulty understanding how similar incidents are managed in other member nations.
- A new pan-European directory of contacts need not be created. The existing directories are sufficient but need to be updated and completed regularly.

Other cyber security exercises have been run around the world. The U.S. Department of Homeland Security involves both public and private sector organizations in its biannual Cyber Storm process. And the Bipartisan Policy Center engaged former U.S. government officials in real-time reaction to its simulated cyber attack. Private enterprise and business sector groups also run cyber security exercises; however, they do not usually make their results public, for fear of revealing problems to possible attackers.

To learn more:

A description of Cyber Europe 2010 and its initial findings is at http://www.enisa.europa.eu/media/press-releases/cyber-europe-2010-a-successful-2019cyber-stress-test2019-for-europe.

Descriptions of the U.S. Department of Homeland Security's Cyber Storm exercises can be found at http://www.dhs.gov/files/training/gc_1204738275985.shtm.

A description of the Cyber Shockwave event, conclusions drawn, and video are at http://www.bipartisanpolicy.org/category/projects/cyber-event.

The nine-part CNN broadcast of the Cyber ShockWave simulation begins at http://www.youtube.com/watch?v=MDWEM2jM7qY.

blanket) or exacerbate our worries (security theater)? What is the likelihood we will be able to determine the causes of these kinds of failures and then prevent or mitigate their effects?

No matter what your work or family responsibilities, it is important for you to understand the nature of these scenarios, make reasoned judgments about their likelihood, and take prudent actions to protect yourselves and the people, data, and things you value.

One way to develop an understanding is to imagine how you might interpret a situation and then react to it. For example, in the unfolding events from mobile phone outage to East Coast power failure, consider these roles:

- You are using your mobile phone to talk with your friend, and the connection drops. You redial repeatedly but never connect. You then try to call your friend on your land line, but again there is no connection. How long does it take you to realize that the problem affects far more people than just you and your friend? Do you contact the telephone company? (And how? You cannot phone, and your Internet connection may very well depend on your telephone carrier!) By the time the power goes out, how do you know the power failure is related to your phone problems? When do you take any action? And what do you do?
- You are using your mobile phone to call your stockbroker because your company's initial public offering (IPO) is scheduled for today—so your company's viability depends on the resulting stock price and the volume of sales. As you begin your conversation with the stockbroker, the connection drops. You redial repeatedly, but never connect. You then try to call your broker on the land line, but again there is no connection. How long does it take you realize that the problem affects your company? Your broker? Others? Whom do you call to report a problem? And when the power goes out, what action do you take?
- You are a government official involved with air traffic control. All morning, you have heard rumors of telephone problems around the country. On your secure government line, you get a call confirming those problems and reporting widening problems with the air traffic control system. How do you determine what is wrong? To whom do you report problems? When you realize that problems with air traffic control may be dangerous to aircraft and their passengers, how do you react? Can you ground all aircraft until the sources of the problems are located and corrected?
- You are a government official involved with regulating the power grid. All morning, you have heard rumors of telephone problems around the country. Your web-based reporting system begins to report sporadic power outages on the East Coast. On your secure government line, you get a call confirming those problems and reporting widening problems with the air traffic control system. How do you determine what is wrong? To whom do you report problems? When you realize that problems with the power grid may threaten the viability of the entire nation's power system, how do you react? The power grid is owned by the private sector. Does the government have authority to shut down the grid until the sources of the problems are located and corrected?

The last situation has precedents. During World War I, the U.S. government took over the railroads [WIL17] and the telephone-telegraph system by presidential proclamations:

> I, Woodrow Wilson, President of the United States, … do hereby take possession and assume control and supervision of each and every telegraph and telephone system, and every part thereof, within the jurisdiction of the United States, including all equipment thereof and appurtenances thereto whatsoever and all materials and supplies [WIL18].

During World War II, the U.S. government encouraged the automotive industry to redirect production toward jeeps, trucks, and airplane parts. The Automotive Council for War Production was formed at the end of 1941, and automobile production was suspended entirely in 1942 so that the industry's total capacity could focus on the war

effort. So possible reactions to our complex scenario could indeed range from inaction to private sector coordination to government intervention. How do you determine cause and effect, severity of impact, and over what time period? The answers are important in suggesting appropriate actions.

Analyzing Computer Security will assist you in understanding the issues and choosing appropriate responses to address these challenges.

In this chapter, we examine our dependence on computers and then explore the many ways in which we are vulnerable to computer failure. Next, we introduce the key concepts of computer security, including attacks, vulnerabilities, threats, and controls. In turn, these concepts become tools for understanding the nature of computer security and our ability to build the trustworthy systems on which our lives and livelihoods depend.

HOW DEPENDENT ARE WE ON COMPUTERS?

You drive down the road and suddenly your car brakes to a stop—or accelerates uncontrollably. You try to withdraw money from your bank and find that your account is overdrawn, even though you think it should contain plenty of money. Your doctor phones to tell you a recent test showed that your usually normal vitamin D level is a fraction of what it should be. And your favorite candidate loses an election that should have been a sure victory. Should you be worried?

There may be other explanations for these events, but any of them may be the result of a computer security problem. Computers are embedded in products ranging from dogs to spaceships; computers control activities from opening doors to administering the proper dose of radiation therapy. Over the last several decades, computer usage has expanded tremendously, and our dependence on computers has increased similarly. So when something goes awry, it is reasonable to wonder if computers are the source of the problem.

But can we—and should we—depend on computers to perform these tasks? How much can we entrust to them, and how will we determine their dependability, safety, and security? These questions continue to occupy policy makers, even as engineers, scientists, and other inventors devise new ways to use computers.

From one perspective, these failures are welcome events because we learn a lot from them. Indeed, engineers are trained to deal with and learn from past failures. So engineers are well qualified to build large structures on which many of us depend. For example, consider bridges; these days, bridges seldom fail. An engineer can study stresses and strengths of materials, and design a bridge that will withstand a certain load for a certain number of years; to ensure that the bridge will last, the engineer can add a margin of safety by using thicker or stronger materials or adding more supports. You can jump up and down on a bridge, because the extra force when you land is well within the tolerance the engineer expected and planned for. When a bridge does fail, it is usually because some bridge component has been made of defective materials, design plans were not followed, or the bridge has been subjected to more strain than was anticipated (which is why some bridges have signs warning about their maximum load).

But computer software is engineered differently, and not all engineers appreciate the differences or implement software appropriately to address a wide variety of security risks. Sidebar 1-2 illustrates some of these risks.

Protecting Software in Automobile Control Systems **Sidebar 1-2**

The amount of software installed in a new automobile grows larger from year to year. Most cars, especially more expensive ones, use dozens of microcontrollers to provide a variety of features aimed at enticing buyers. These digital cars use software to control individual subsystems, and then more software to connect the systems into a network.

Whitehorn-Umphres [WHI01] points out that this kind of software exhibits a major difference in thinking between hardware designers and software designers. "As hardware engineers, they [the automobile designers] assumed that, perhaps aside from bolt-on aftermarket parts, everything else is and should be a black box." But software folks have a different take: "As a software designer, I assume that all digital technologies are fair game for being played with … it takes a special kind of personality to look at a software-enabled device and see the potential for manipulation and change—a hacker personality." That is, hardware engineers do not expect their devices to be opened and changed, but software engineers—especially security specialists—do.

As a result, the hardware-trained engineers designing and implementing automotive software see no reason to protect it from hackers. According to a paper by Koscher and other researchers from the University of Washington and University of California San Diego [KOS10], "Over a range of experiments, both in the lab and in road tests, we demonstrate the ability to adversarially control a wide range of automotive functions and completely ignore driver input—including disabling the brakes, selectively braking individual wheels on demand, stopping the engine, and so on. We find that it is possible to bypass rudimentary network security protections within the car, such as maliciously bridging between our car's two internal subnets. We also present composite attacks that leverage individual weaknesses, including an attack that embeds malicious code in a car's telematics unit and that will completely erase any evidence of its presence after a crash." Their paper presents several laboratory attacks that could have devastating effects if performed on real cars on a highway.

Koscher and colleagues observe that "the future research agenda for securing cyber-physical vehicles is not merely to consider the necessary technical mechanisms, but to also inform these designs by what is feasible practically and compatible with the interests of a broader set of stakeholders."

Security experts have long sought to inform designers and developers of security risks and countermeasures. Unfortunately, all too often the pleas of the security community are ignored in the rush to add and deliver features that will improve sales.

Like bridges, computers can fail: Some moving parts wear out, electronic hardware components stop working or, worse, work intermittently. Indeed, computers can be *made* to fail without even being physically touched. Failures can happen seemingly spontaneously, when unexpected situations put the system into a failing or failed state. So there are many opportunities for both benign users and malicious attackers to cause failures. Failures can be small and harmless, like a "click here" button that does nothing, or catastrophic, like a faulty program that destroys a file or even erases an entire disk. The effects of failures can be readily apparent—a screen goes blank—or stealthy and difficult to find, such as a program that covertly records every key pressed on the keyboard.

Computer security addresses all these types of failures, including the ones we cannot yet see or even anticipate. The computers we consider range from small chips to embedded devices to stand-alone computers to gangs of servers. So too do we include

private networks, public networks, and the Internet. They constitute the backbone of what we do and how we do it: commerce, communication, health care, and more. So understanding failure can lead us to improvements in the way we lead our lives.

Each kind or configuration of computer has many ways of failing and being made to fail. Nevertheless, the analytic approach you will learn in this book will enable you to look at each computer system (and the applications that run on it) to determine how you can protect data, computers, networks, and ultimately yourselves.

WHAT IS COMPUTER SECURITY?

Computer security is the protection of the items you value, called the **assets** of a computer or computer system. There are many types of assets, involving hardware, software, data, people, processes, or combinations of these. To determine what to protect, we must first identify what has value and to whom.

A computer device (including hardware, added components, and accessories) is certainly an asset. Because most computer hardware is pretty useless without programs, the software is also an asset. Software includes the operating system, utilities and device handlers; applications such as word processing, media players, or email handlers; and even programs that you may have written yourself. Much hardware and software is *off-the-shelf*, meaning that it is commercially available (not custom-made for your purpose) and that you can easily get a replacement. The thing that makes your computer unique and important to you is your content: photos, tunes, papers, email messages, projects, calendar information, ebooks (with your annotations), contact information, code you created, and the like. Thus, data items on a computer are assets, too. Unlike most hardware and software, data can be hard—if not impossible—to re-create or replace. These assets are shown in Figure 1-1.

Hardware:
- Computer
- Devices (disk drives, memory, printer)
- Network gear

Software:
- Operating system
- Utilities (antivirus)
- Commercial applications (word processing, photo editing)
- Individual applications

Data:
- Documents
- Photos
- Music, videos
- Email
- Class projects

FIGURE 1-1 Computer Objects of Value

These three things—hardware, software, and data—contain or express things like the design for your next new product, the photos from your recent vacation, the chapters of your new book, or the genome sequence resulting from your recent research. All of these things represent intellectual endeavor or property, and they have value that differs from one person or organization to another. It is that value that makes them assets worthy of protection, and they are the elements we want to protect. Other assets, such as access to data, quality of service, processes, human users, and network connectivity, deserve protection, too; they are affected or enabled by the hardware, software, and data. So in most cases, protecting hardware, software, and data covers these other assets as well.

In this book, unless we specifically distinguish among hardware, software, and data, we refer to all these assets as the computer system, or sometimes as the computer. And because processors are embedded in so many devices, we also need to think about such variations as cell phones, implanted pacemakers, and automobiles. Even if the primary purpose of the device is not computing, the device's embedded computer can be involved in security incidents and represents an asset worthy of protection.

After identifying the assets to protect, we next determine their value. We make value-based decisions frequently, even when we are not aware of them. For example, when you go for a swim you can leave a bottle of water on a towel on the beach, but not your wallet or cell phone. The difference relates to the value of the assets.

The value of an asset depends on the asset owner's or user's perspective, and it may be independent of monetary cost, as shown in Figure 1-2. Your photo of your sister, worth only a few cents in terms of paper and ink, may have high value to you and no value to your roommate. Other items' value depends on replacement cost; some

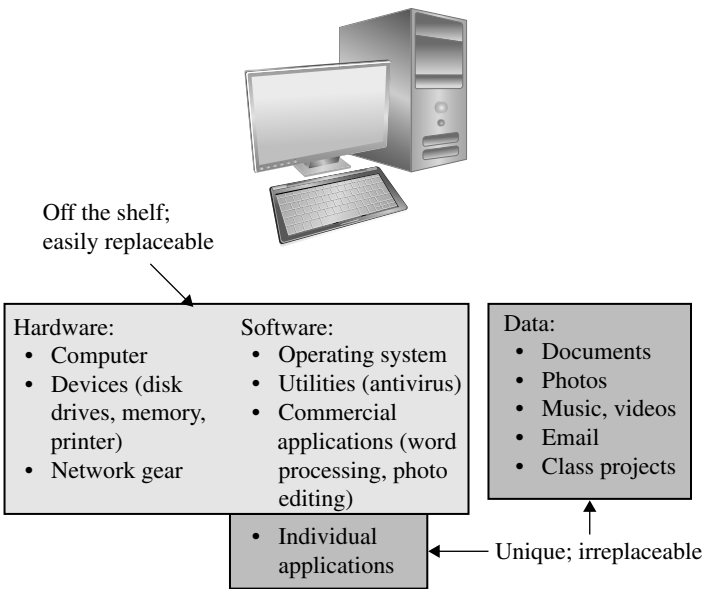


FIGURE 1-2 Values of Assets

computer data are difficult or impossible to replace. For example, that photo of you and your friends at a party may have cost you nothing, but it is invaluable because it can never be replaced. On the other hand, the DVD of your favorite film may have cost a significant portion of your take-home pay, but you can buy another one if the DVD is stolen or corrupted. Similarly, timing has bearing on asset value. For example, the value of the plans for a company's new product line is very high, especially to competitors. But once the new product is released, the plans' value drops dramatically.

The Vulnerability–Threat–Control Paradigm

The goal of computer security is protecting valuable assets. To study different ways of protection, we use a framework that describes how assets may be harmed and how to counter or mitigate that harm.

A **vulnerability** is a weakness in the system, for example, in procedures, design, or implementation, that might be exploited to cause loss or harm. For instance, a particular system may be vulnerable to unauthorized data manipulation because the system does not verify a user's identity before allowing data access.

A **threat** to a computing system is a set of circumstances that has the potential to cause loss or harm. To see the difference between a threat and a vulnerability, consider the illustration in Figure 1-3. Here, a wall is holding water back. The water to the left of the wall is a threat to the man on the right of the wall: The water could rise, overflowing onto the man, or it could stay beneath the height of the wall, causing the wall to collapse. So the threat of harm is the potential for the man to get wet, get hurt, or be drowned. For now, the wall is intact, so the threat to the man is unrealized.

However, we can see a small crack in the wall—a vulnerability that threatens the man's security. If the water rises to or beyond the level of the crack, it will exploit the vulnerability and harm the man.

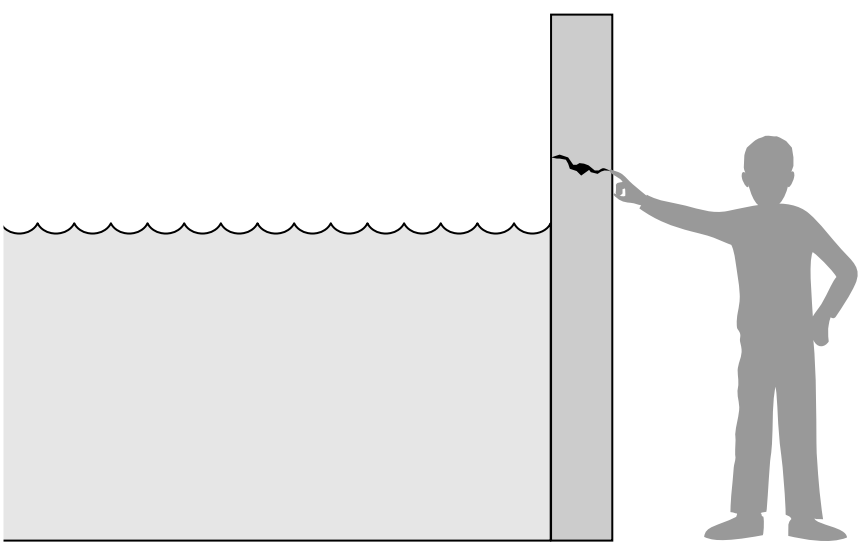


FIGURE 1-3 Threat and Vulnerability

There are many threats to a computer system, including human-initiated and computer-initiated ones. We have all experienced the results of inadvertent human errors, hardware design flaws, and software failures. But natural disasters are threats, too; they can bring a system down when the computer room is flooded or the data center collapses from an earthquake, for example.

A human who exploits a vulnerability perpetrates an **attack** on the system. An attack can also be launched by another system, as when one system sends an overwhelming flood of messages to another, virtually shutting down the second system's ability to function. Unfortunately, we have seen this type of attack frequently, as denial-of-service attacks deluge servers with more messages than they can handle. (We take a closer look at denial of service in Chapters 7 and 15.)

How do we address these problems? We use a **control** or **countermeasure** as protection. That is, a control is an action, device, procedure, or technique that removes or reduces a vulnerability. In Figure 1-3, the man is placing his finger in the hole, controlling the threat of water leaks until he finds a more permanent solution to the problem. In general, we can describe the relationship among threats, controls, and vulnerabilities in this way:

A *threat* is blocked by *control* of a *vulnerability*.

In this book we take the approach of picking a particular type of threat, usually in the form of an attack. From that threat we determine the vulnerabilities that could allow the threat to cause harm. Finally, we explore the countermeasures that can control the threat or neutralize the vulnerability. Thus, this book is about protecting assets by countering threats that could exploit vulnerabilities.

Before we can protect assets, we have to know the kinds of harm we have to protect them against, so now we explore threats to valuable assets.

THREATS

We can consider potential harm to assets in two ways: First, we can look at what bad things can happen to assets, and second, we can look at who or what can cause or allow those bad things to happen. These two perspectives enable us to determine how to protect assets.

Think for a moment about what makes your computer valuable to you. First, you use it as a tool for sending and receiving email, searching the web, writing papers, and performing many other tasks, and you expect it to be available for use when you want it. Without your computer these tasks would be harder, if not impossible. Second, you rely heavily on your computer's integrity. When you write a paper and save it, you trust that the paper will reload exactly as you saved it. Similarly, you expect that the photo a friend passes you on a flash drive will appear the same when you load it into your computer as when you saw it on your friend's. Finally, you expect the "personal" aspect of a personal computer to stay personal, meaning you want it to protect your confidentiality. For example, you want your email messages to be just between you and your listed recipients; you don't want them broadcast to other people. And when you write an essay, you expect no one else to be able to copy it without your permission.

These three aspects, availability, integrity, and confidentiality, make your computer valuable to you. But viewed from another perspective, they are three possible ways to make it less valuable, that is, to cause you harm. If someone steals your computer, scrambles data on your disk, or looks at your private data files, the value of your computer has been diminished or your computer use has been harmed. These characteristics are both basic security properties and the objects of security threats.

We can define these three properties as follows.

- **availability**: the ability of a system to ensure that an asset can be used by any authorized parties
- **integrity**: the ability of a system to ensure that an asset is modified only by authorized parties
- **confidentiality**: the ability of a system to ensure that an asset is viewed only by authorized parties

These three properties, hallmarks of good security, appear in the literature as early as James P. Anderson's essay on computer security [AND73] and reappear frequently in more recent computer security papers and discussions. Taken together (and rearranged), the properties are called the **C-I-A triad** or the **security triad**. ISO 7498-2 [ISO89] adds to them two more properties that are desirable, particularly in communication networks:

- **authentication**: the ability of a system to confirm the identity of a sender
- **nonrepudiation** or **accountability**: the ability of a system to confirm that a sender cannot convincingly deny having sent something

The U.S. Department of Defense [DOD85] adds auditability: the ability of a system to trace all actions related to a given asset. The C-I-A triad forms a foundation for thinking about security. Authentication and nonrepudiation extend security notions to network communications, and auditability is important in establishing individual accountability for computer activity. In this book we generally use the C-I-A triad as our security taxonomy so that we can frame threats, vulnerabilities, and controls in terms of the C-I-A properties affected. We highlight one of these other properties when it is relevant to a particular threat we are describing. For now, we focus on just the three elements of the triad.

What can happen to harm the confidentiality, integrity, or availability of computer assets? If a thief steals your computer, you no longer have access, so you have lost availability; furthermore, if the thief looks at the pictures or documents you have stored, your confidentiality is lost. And if the thief changes the content of your music files but then gives them back with your computer, the integrity of your data has been harmed. You can envision many scenarios based around these three properties.

The C-I-A triad can be viewed from a different perspective: the nature of the harm caused to assets. Harm can also be characterized by four acts: **interception**, **interruption**, **modification**, and **fabrication**. From this point of view, confidentiality can suffer if someone intercepts data, availability is lost if someone or something interrupts a flow of data or access to a computer, and integrity can fail if someone or something modifies data or fabricates false data. These four acts are depicted in Figure 1-4.

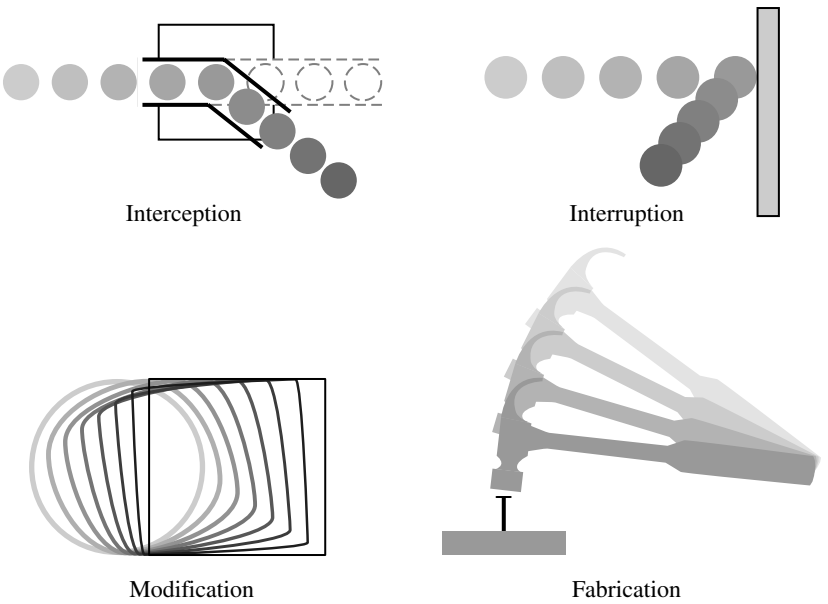


FIGURE 1-4 Four Acts to Cause Security Harm

Thinking of these four kinds of acts can help you determine what threats might exist against the computers you are trying to protect.

To analyze harm, we next refine the C-I-A triad, looking more closely at each of its elements.

Confidentiality

Some things obviously need confidentiality protection. For example, students' grades, financial transactions, medical records, and tax returns are sensitive. A proud student may run out of a classroom screaming "I got an A!" but the student should be the one to choose whether to reveal that grade to others. Other things, such as diplomatic and military secrets, companies' marketing and product development plans, and educators' tests, also must be carefully controlled. Sometimes, however, it is not so obvious that something is sensitive. For example, a military food order may seem like innocuous information, but a sudden increase in the order could be a sign of incipient engagement in conflict. Purchases of food, hourly changes in location, and access to books are not things you would ordinarily consider confidential, but they can reveal something that someone wants to be kept confidential.

The definition of confidentiality is straightforward: Only authorized people or systems can access protected data. However, as we see in later chapters, ensuring confidentiality can be difficult. For example, who determines which people or systems are authorized to access the current system? By "accessing" data, do we mean that an authorized party can access a single bit? the whole collection? pieces of data out of context? Can someone who is authorized disclose data to other parties? Sometimes there is even a question of who owns the data: If you visit a web page, do you own the

fact that you clicked on a link, or does the web page owner, the Internet provider, someone else, or all of you?

In spite of these complicating examples, confidentiality is the security property we understand best because its meaning is narrower than that of the other two. We also understand confidentiality well because we can relate computing examples to those of preserving confidentiality in the real world.

Confidentiality relates most obviously to data, although we can think of the confidentiality of a piece of hardware (a novel invention) or a person (the whereabouts of a wanted criminal). Here are some properties that could mean a failure of data confidentiality:

- An unauthorized person accesses a data item.
- An unauthorized process or program accesses a data item.
- A person authorized to access certain data accesses other data not authorized (which is a specialized version of an unauthorized person accesses a data item).
- An unauthorized person accesses an approximate data value (for example, not knowing someone's exact salary but knowing that the salary falls in a particular range or exceeds a particular amount).
- An unauthorized person learns the existence of a piece of data (for example, knowing that a company is developing a certain new product or that talks are under way about the merger of two companies).

Notice the general pattern of these statements: A person, process, or program is (or is not) authorized to access a data item in a particular way. We call the person, process, or program a **subject**, the data item an **object**, the kind of access (such as read, write, or execute) an **access mode**, and the authorization a **policy**, as shown in Figure 1-5.

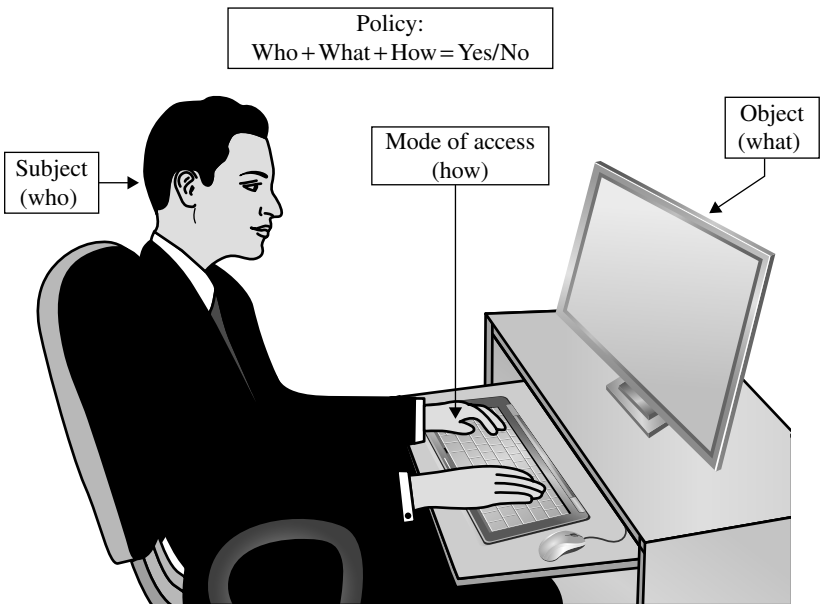


FIGURE 1-5 Access Control

These four terms will reappear throughout this book because they are fundamental aspects of computer security.

One word that captures most aspects of confidentiality is *view*, although you should not take that term literally. A failure of confidentiality does not necessarily mean that someone sees an object and, in fact, it is virtually impossible to look at bits in any meaningful way (although you may look at their representation as characters or pictures). The word *view* does connote another aspect of confidentiality in computer security, through the association with viewing a movie or a painting in a museum: look but do not touch. In computer security, confidentiality usually means obtaining but not modifying. Modification is the subject of integrity, which we consider in the next section.

Integrity

Examples of integrity failures are easy to find. A number of years ago a malicious macro in a Word document inserted the word "not" after some random instances of the word "is"; you can imagine the havoc that ensued. Because the document was generally syntactically correct, people did not immediately detect the change. In another case, a model of the Pentium computer chip produced an incorrect result in certain circumstances of floating-point arithmetic. Although the circumstances of failure were rare, Intel decided to manufacture and replace the chips. Many of us receive mail that is misaddressed because someone typed something wrong when transcribing from a written list; worse is that inaccuracy being propagated to other mailing lists such that we can never seem to correct the root of the problem. Other times we find that a spreadsheet seems to be wrong, only to find that someone typed "space 123" in a cell, changing it from a numeric value to text, so the spreadsheet program misused that cell in computation. Suppose someone converted numeric data to Roman numerals: One could argue that IV is the same as 4, but IV would not be useful in most applications, nor would it be obviously meaningful to someone expecting 4 as an answer. These cases show some of the breadth of examples of integrity failures.

Integrity is harder to pin down than confidentiality. As Steve Welke and Terry Mayfield [WEL90, MAY91, NCS91b] point out, integrity means different things in different contexts. When we survey the way some people use the term, we find several different meanings. For example, if we say that we have preserved the integrity of an item, we may mean that the item is

- precise
- accurate
- unmodified
- modified only in acceptable ways
- modified only by authorized people
- modified only by authorized processes
- consistent
- internally consistent
- meaningful and usable

Integrity can also mean two or more of these properties. Welke and Mayfield recognize three particular aspects of integrity—authorized actions, separation and protection of resources, and error detection and correction. Integrity can be enforced in much the same way as can confidentiality: by rigorous control of who or what can access which resources in what ways.

Availability

A computer user's worst nightmare: you turn on the switch and the computer does nothing. Your data and programs are presumably still there, but you cannot get at them. Fortunately, few of us experience that failure. Many of us do experience overload, however: access gets slower and slower; the computer responds but not in a way we consider normal or acceptable.

Availability applies both to data and to services (that is, to information and to information processing), and it is similarly complex. As with the notion of integrity, different people expect availability to mean different things. For example, an object or service is thought to be available if the following are true:

- It is present in a usable form.
- It has enough capacity to meet the service's needs.
- It is making clear progress, and, if in wait mode, it has a bounded waiting time.
- The service is completed in an acceptable period of time.

We can construct an overall description of availability by combining these goals. Following are some criteria to define availability.

- There is a timely response to our request.
- Resources are allocated fairly so that some requesters are not favored over others.
- Concurrency is controlled; that is, simultaneous access, deadlock management, and exclusive access are supported as required.
- The service or system involved follows a philosophy of fault tolerance, whereby hardware or software faults lead to graceful cessation of service or to work-arounds rather than to crashes and abrupt loss of information. (Cessation does mean end; whether it is graceful or not, ultimately the system is unavailable. However, with fair warning of the system's stopping, the user may be able to move to another system and continue work.)
- The service or system can be used easily and in the way it was intended to be used. (This is a characteristic of usability, but an unusable system may also cause an availability failure.)

As you can see, expectations of availability are far-reaching. In Figure 1-6 we depict some of the properties with which availability overlaps. Indeed, the security community is just beginning to understand what availability implies and how to ensure it.

A person or system can do three basic things with a data item: view it, modify it, or use it. Thus, viewing (confidentiality), modifying (integrity), and using (availability) are the basic modes of access that computer security seeks to preserve.

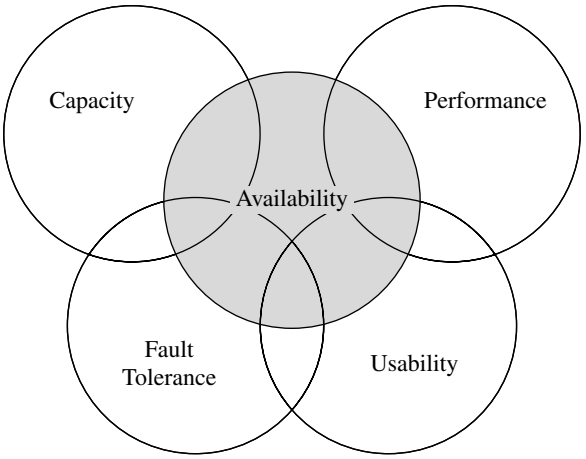


FIGURE 1-6 Availability and Related Areas

A paradigm of computer security is **access control**: To implement a policy, computer security controls all accesses by all subjects to all protected objects in all modes of access. A small, centralized control of access is fundamental to preserving confidentiality and integrity, but it is not clear that a single access-control point can enforce availability. Indeed, experts on dependability will note that single points of control can become single points of failure, making it easy for an attacker to destroy availability by disabling the single control point. Much of computer security's past success has focused on confidentiality and integrity; there are models of confidentiality and integrity, for example, see David Bell and Leonard La Padula [BEL73, BEL76] and Kenneth Biba [BIB77]. Availability is security's next great challenge.

We have just described the C-I-A triad and the three fundamental security properties it represents. Our description of these properties was in the context of things that need protection. To motivate your understanding we gave some examples of harm and threats to cause harm. Our next step is to think about the nature of threats themselves.

Types of Threats

For some ideas of harm, look at Figure 1-7 taken from Willis Ware's report [WAR70]. Although it was written when computers were so big, so expensive, and so difficult to operate that only large organizations like universities, companies, or government departments would have one, Ware's discussion is still instructive. Ware was concerned primarily with the protection of classified data, that is, preserving confidentiality. In the figure, he depicts humans such as programmers and maintenance staff gaining access to data, as well as radiation by which data can escape as signals. From the figure you can see some of the many kinds of threats to a computer system.

One way to analyze harm is to consider the cause or source. We call a potential cause of harm a **threat**. Different kinds of threats are shown in Figure 1-8. Harm can be caused by either nonhuman events or humans. Examples of **nonhuman threats** include natural disasters like fires or floods; loss of electrical power; failure of a component such as a communications cable, processor chip, or disk drive; or attack by a wild boar.

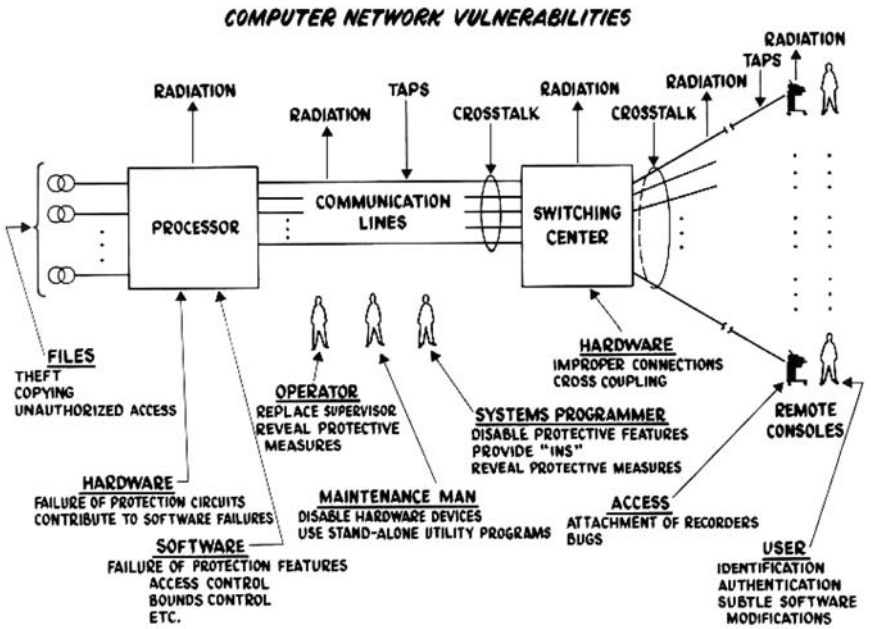


FIGURE 1-7 Computer [Network] Vulnerabilities (from [WAR70])

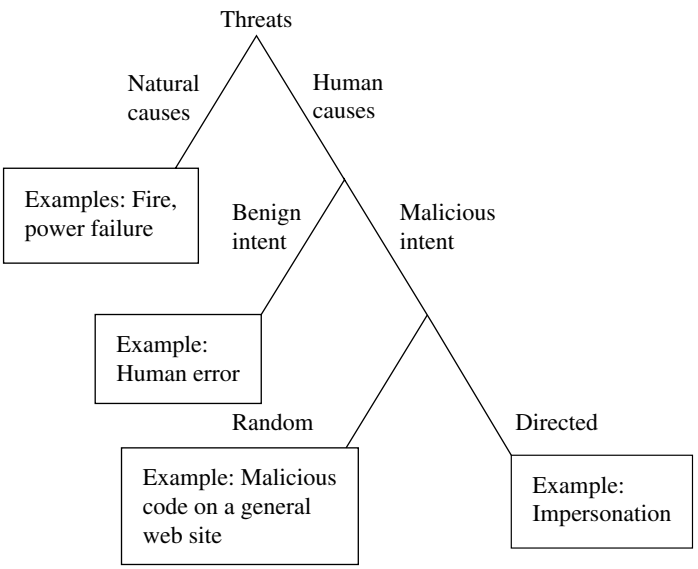


FIGURE 1-8 Kinds of Threats

Human threats can be either benign (nonmalicious) or malicious. **Nonmalicious** kinds of harm include someone accidentally spilling a soft drink on a laptop, unintentionally deleting text, inadvertently sending an email message to the wrong person, and carelessly typing "12" instead of "21" when entering a phone number or clicking "yes"

instead of "no" to overwrite a file. These inadvertent, human errors happen to most people; we just hope that the seriousness of harm is not too great, or if it is, that we will not repeat the mistake.

Most computer security activity relates to **malicious human-caused harm**: A malicious attacker actually wants to cause harm, and so we often use the term *attack* for a malicious computer security event. Malicious attacks can be random or directed. In a **random attack** the attacker wants to harm any computer or user; such an attack is analogous to accosting the next pedestrian who walks down the street. An example of a random attack is malicious code posted on a web site that could be visited by anybody.

In a **directed attack**, the attacker intends harm to specific computers, perhaps at one organization (think of attacks against a political organization) or belonging to a specific individual (think of trying to drain a specific person's bank account, for example, by impersonation). Another class of directed attack is against a particular product, such as any computer running a particular browser. (We do not want to split hairs about whether such an attack is directed—at that one software product—or random, against any user of that product; the point is not semantic perfection but protecting against the attacks.) The range of possible directed attacks is practically unlimited.

Although the distinctions shown in Figure 1-8 seem clear-cut, sometimes the nature of an attack is not obvious until the attack is well under way, or perhaps even ended. A normal hardware failure can seem like a directed, malicious attack to deny access, and hackers often try to conceal their activity to look like ordinary, authorized users. As computer security experts we need to anticipate what bad things might happen, instead of waiting for the attack to happen or debating whether the attack is intentional or accidental.

Neither this book nor any other checklist or method can show you *all* the kinds of harm that can happen to computer assets. There are too many ways to interfere with your use of these assets. Two retrospective lists of *known* vulnerabilities are of interest, however. CVE, the Common Vulnerabilities and Exposures list (see http://cve.mitre.org/) is a dictionary of publicly known information security vulnerabilities and exposures. CVE's common identifiers enable data exchange between security products and provide a baseline index point for evaluating coverage of security tools and services. To measure the extent of harm, CVSS, the Common Vulnerability Scoring System (see http://nvd.nist.gov/cvss.cfm) provides a standard measurement system that allows accurate and consistent scoring of vulnerability impact.

To imagine the full landscape of possible attacks, you may find it useful to consider the kinds of people who attack computer systems. Although potentially anyone is an attacker, certain classes of people stand out because of their backgrounds or objectives. Thus, in the following sections we look at profiles of some classes of attackers.

Types of Attackers

Who are attackers? As we have seen, their motivations range from chance to a specific target. Putting aside attacks from natural and benign causes, we can explore who are attackers and what motivates them.

Most studies of attackers actually analyze computer criminals, that is, people who have actually been convicted of a crime, primarily because that group is easy to identify and study. The ones who got away or who carried off an attack without being

detected may have characteristics different from those of the criminals who have been caught. Worse, by studying only the criminals we have caught, we may not learn how to catch attackers who know how to abuse the system without being apprehended.

What does a cyber criminal look like? In television and films the villains wore shabby clothes, looked mean and sinister, and lived in gangs somewhere out of town. By contrast, the sheriff dressed well, stood proud and tall, was known and respected by everyone in town, and struck fear in the hearts of most criminals.

To be sure, some computer criminals are mean and sinister types. But many more wear business suits, have university degrees, and appear to be pillars of their communities. Some are high school or university students. Others are middle-aged business executives. Some are mentally deranged, overtly hostile, or extremely committed to a cause, and they attack computers as a symbol. Others are ordinary people tempted by personal profit, revenge, challenge, advancement, or job security—like perpetrators of any crime, using a computer or not. Researchers have tried to find the psychological traits that distinguish attackers, as described in Sidebar 1-3. No single profile captures the characteristics of a "typical" computer attacker, and the characteristics of some notorious attackers also match many people who are not attackers. As shown in Figure 1-9, attackers look just like anybody in a crowd.

Individuals

Originally, computer attackers were individuals, acting with motives of fun, challenge, or revenge. Early attackers such as Robert Morris Jr., the Cornell University graduate student who brought down the Internet in 1988 [SPA89], and Kevin Mitnick, the man who broke into and stole data from dozens of computers including the San Diego Supercomputer Center [MAR95], acted alone.

Organized Worldwide Groups

More recent attacks have involved groups of people. An attack against the government of the country of Estonia (described in more detail in Chapter 15) is believed to have been an uncoordinated outburst from a loose federation of attackers from around the world. Kevin Poulsen [POU05] quotes Tim Rosenberg, a research professor at George Washington University, warning of "multinational groups of hackers backed by organized crime" and showing the sophistication of prohibition-era mobsters. He also reports that Christopher Painter, deputy director of the U.S. Department of Justice's computer crime section, argues that cyber criminals and serious fraud artists are increasingly working in concert or are one and the same. According to Painter, loosely connected groups of criminals all over the world work together to break into systems and steal and sell information, such as credit card numbers. For instance, in October 2004, U.S. and Canadian authorities arrested 28 people from 6 countries involved in a global organized cybercrime ring to buy and sell credit card information and identities.

Whereas early motives for computer attackers such as Morris and Mitnick were personal, such as prestige or accomplishment, recent attacks have been heavily influenced by financial gain. Security firm McAfee reports "Criminals have realized the huge financial gains to be made from the Internet with little risk. They bring the skills,

An Attacker's Psychological Profile? Sidebar 1-3

Temple Grandin, a professor of animal science at Colorado State University and a sufferer from a mental disorder called Asperger syndrome (AS), thinks that Kevin Mitnick and several other widely described hackers show classic symptoms of Asperger syndrome. Although quick to point out that no research has established a link between AS and hacking, Grandin notes similar behavior traits among Mitnick, herself, and other AS sufferers. An article in *USA Today* (29 March 2001) lists the following AS traits:

- poor social skills, often associated with being loners during childhood; the classic "computer nerd"
- fidgeting, restlessness, inability to make eye contact, lack of response to cues in social interaction, such as facial expressions or body language
- exceptional ability to remember long strings of numbers
- ability to focus on a technical problem intensely and for a long time, although easily distracted on other problems and unable to manage several tasks at once
- deep honesty and respect for laws

Donn Parker [PAR98] has studied hacking and computer crime for over 20 years. He states "hackers are characterized by an immature, excessively idealistic attitude … They delight in presenting themselves to the media as idealistic do-gooders, champions of the underdog."

Consider the following excerpt from an interview [SHA00] with "Mixter," the German programmer who admitted he was the author of a widespread piece of attack software called Tribal Flood Network (TFN) and its sequel TFN2K:

> Q: Why did you write the software?
>
> A: I first heard about Trin00 [another denial of service attack] in July '99 and I considered it as interesting from a technical perspective, but also potentially powerful in a negative way. I knew some facts of how Trin00 worked, and since I didn't manage to get Trin00 sources or binaries at that time, I wrote my own server-client network that was capable of performing denial of service.
>
> Q: Were you involved … in any of the recent high-profile attacks?
>
> A: No. The fact that I authored these tools does in no way mean that I condone their active use. I must admit I was quite shocked to hear about the latest attacks. It seems that the attackers are pretty clueless people who misuse powerful resources and tools for generally harmful and senseless activities just "because they can."

Notice that from some information about denial-of-service attacks, he wrote his own server-client network and then a denial-of-service attack. But he was "quite shocked" to hear they were used for harm.

More research is needed before we will be able to define the profile of a hacker. And even more work will be needed to extend that profile to the profile of a (malicious) attacker. Not all hackers become attackers; some hackers become extremely dedicated and conscientious system administrators, developers, or security experts. But some psychologists see in AS the rudiments of a hacker's profile.

knowledge, and connections needed for large scale, high-value criminal enterprise that, when combined with computer skills, expand the scope and risk of cybercrime" [MCA05].

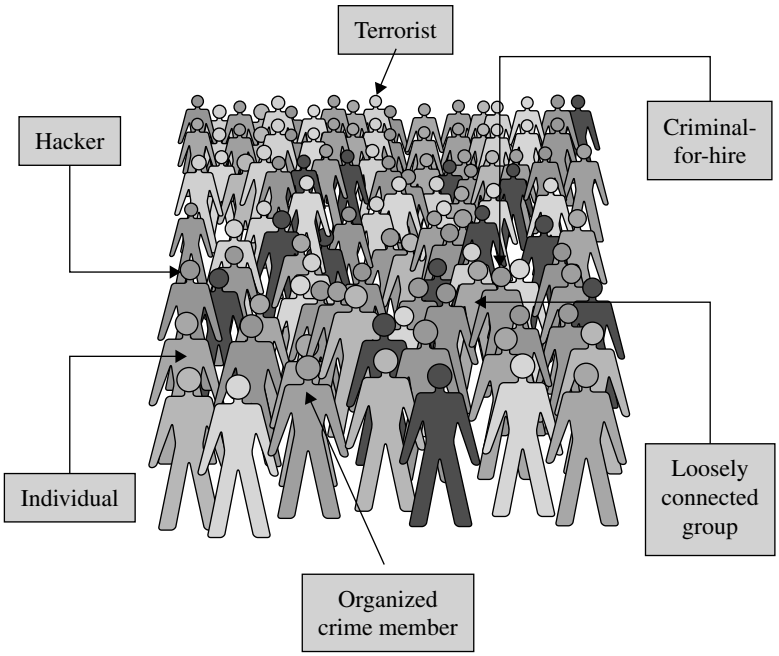


FIGURE 1-9 Attackers

Organized Crime

Attackers' goals include fraud, extortion, money laundering, and drug trafficking, areas in which organized crime has a well-established presence. Evidence is growing that organized crime groups are engaging in computer crime. In fact, traditional criminals are recruiting hackers to join the lucrative world of cybercrime. For example, Albert Gonzales was sentenced in March 2010 to 20 years in prison for working with a crime ring to steal 40 million credit card numbers from retailer TJMaxx and others, costing over $200 million (Reuters, March 26, 2010).

Organized crime may use computer crime (such as stealing credit card numbers or bank account details) to finance other aspects of crime. Recent attacks suggest that organized crime and professional criminals have discovered just how lucrative computer crime can be. Mike Danseglio, a security project manager with Microsoft, said, "In 2006, the attackers want to pay the rent. They don't want to write a worm that destroys your hardware. They want to assimilate your computers and use them to make money" [NAR06a]. Mikko Hyppönen, Chief Research Officer with Finnish security company f-Secure, agrees that today's attacks often come from Russia, Asia, and Brazil; the motive is now profit, not fame [BRA06]. Ken Dunham, Director of the Rapid Response Team for Verisign says he is "convinced that groups of well-organized mobsters have taken control of a global billion-dollar crime network powered by skillful hackers" [NAR06b].

McAfee also describes the case of a hacker-for-hire: a businessman who hired a sixteen-year-old New Jersey hacker to attack the web sites of his competitors. The hacker barraged the site for a five-month period and damaged not only the target companies but

also their Internet service providers (ISPs) and other unrelated companies that used the same ISPs. By FBI estimates the attacks cost all the companies over $2 million; the FBI arrested both hacker and businessman in March 2005 [MCA05].

Brian Snow [SNO05] observes that hackers want a score or some kind of evidence to give them bragging rights. Organized crime wants a resource; such criminals want to stay under the radar to be able to extract profit from the system over time. These different objectives lead to different approaches to computer crime: The novice hacker can use a quick and dirty attack, whereas the professional attacker wants a neat, robust, and undetected method that can deliver rewards for a long time.

Terrorists

The link between computer security and terrorism is quite evident. We see terrorists using computers in four ways:

- *Computer as target of attack*: Denial-of-service attacks and web site defacements are popular activities for any political organization because they attract attention to the cause and bring undesired negative attention to the object of the attack. An example is the massive denial-of-service attack launched against the country of Estonia, detailed in Chapter 15.
- *Computer as method of attack*: Launching offensive attacks requires use of computers. Stuxnet, malicious computer code called a worm, is known to attack automated control systems, specifically a model of control system manufactured by Siemens. Experts say the code is designed to disable machinery used in the control of nuclear reactors in Iran [MAR10]. The persons behind the attack are unknown, but the infection is believed to have spread through USB flash drives brought in by engineers maintaining the computer controllers.
- *Computer as enabler of attack*: Web sites, web logs, and email lists are effective, fast, and inexpensive ways to allow many people to coordinate. According to the Council on Foreign Relations, the terrorists responsible for the November 2008 attack that killed over 200 people in Mumbai used GPS systems to guide their boats, Blackberries for their communication, and Google Earth to plot their routes.
- *Computer as enhancer of attack*: The Internet has proved to be an invaluable means for terrorists to spread propaganda and recruit agents. In October 2009 the FBI arrested Colleen LaRose, also known as *JihadJane*, after she had spent months using email, YouTube, MySpace, and electronic message boards to recruit radicals in Europe and South Asia to "wage violent jihad," according to a federal indictment unsealed in March 2010.

We cannot accurately measure the degree to which terrorists use computers, because of the secret nature of terrorist activities and because our definitions and measurement tools are rather weak. Still, incidents like the one described in Sidebar 1-4 provide evidence that all three of these activities are increasing.

If someone on television sneezes, you do not worry about the possibility of catching a cold. But if someone standing next to you sneezes, you may become concerned. In the next section we examine the harm that can come from the presence of a computer security threat on your own computer systems.

The Terrorists, Inc., IT Department — Sidebar 1-4

In 2001, a reporter for the Wall Street Journal bought a used computer in Afghanistan. Much to his surprise, he found that the hard drive contained what appeared to be files from a senior al Qaeda operative. The reporter, Alan Cullison [CUL04], reports that he turned the computer over to the FBI. In his story published in 2004 in *The Atlantic*, he carefully avoids revealing anything he thinks might be sensitive.

The disk contained over 1,000 documents, many of them encrypted with relatively weak encryption. Cullison found draft mission plans and white papers setting forth ideological and philosophical arguments for the attacks of September 11, 2001. Also found were copies of news stories on terrorist activities. Some of the found documents indicated that al Qaeda was not originally interested in chemical, biological, or nuclear weapons, but became interested after reading public news articles accusing al Qaeda of having those capabilities.

Perhaps most unexpected were email messages of the kind one would find in a typical office: recommendations for promotions, justifications for petty cash expenditures, and arguments concerning budgets.

The computer appears to have been used by al Qaeda from 1999 to 2001. Cullison notes that Afghanistan in late 2001 was a scene of chaos, and it is likely the laptop's owner fled quickly, leaving the computer behind, where it fell into the hands of a secondhand goods merchant who did not know its contents.

But this computer's contents illustrate an important aspect of computer security and confidentiality: We can never predict the time at which a security disaster will strike, and thus we must always be prepared as if it happens suddenly and immediately.

HARM

The negative consequence of an actualized threat is **harm**; we protect ourselves against threats in order to reduce or eliminate harm. We have already described many examples of computer harm: a stolen computer, modified or lost file, revealed private letter, or denial of access. These events cause harm that we want to avoid.

In our earlier discussion of asset, we noted that value is highly dependent on owner or outsider perception and need. Some aspects of value are immeasurable, such as the value of the paper you need to submit to your professor tomorrow; if you lose the paper (that is, if its availability is lost), no amount of money will compensate you for it. Items on which you place little or no value might be more valuable to someone else; for example, the group photograph taken at last night's party can reveal that your friend was not where he told his wife he would be. Even though it may be difficult to assign a specific number as the value of an asset, you can usually assign a value on a generic scale, such as moderate or minuscule or incredibly high, depending on the degree of harm that loss or damage to the object would cause. Or you can assign a value relative to other assets, based on comparable loss: This version of the file is more valuable to me than that version.

In their 2010 global Internet threat report, security firm Symantec surveyed the kinds of goods and services offered for sale on underground web pages. The item most frequently offered in both 2009 and 2008 was credit card numbers, at prices ranging from $0.85 to $30.00 each. (Compare those prices to an individual's effort to deal with the impact of a stolen credit card or the potential amount lost by the issuing bank.)

Second most frequent was bank account credentials, at $15 to $850; these were offered for sale at 19% of web sites in both years. Email accounts were next at $1 to $20, and lists of email addresses went for $1.70 to $15.00 per thousand. At position 10 in 2009 were web site administration credentials, costing only $2 to $30. These black market web sites demonstrate that the market price of computer assets can be dramatically different from their value to rightful owners.

The value of many assets can change over time, so the degree of harm (and therefore the severity of a threat) can change, too. With unlimited time, money, and capability, we might try to protect against all kinds of harm. But because our resources are limited, we must prioritize our protection, safeguarding only against serious threats and the ones we can control. Choosing the threats we try to mitigate involves a process called **risk management**, and it includes weighing the seriousness of a threat against our ability to protect.

Risk and Common Sense

The number and kinds of threats are practically unlimited, because devising an attack requires an active imagination, determination, persistence, and time (as well as access and resources). The nature and number of threats in the computer world reflect life in general: The causes of harm are limitless and largely unpredictable. Natural disasters like volcanoes and earthquakes happen with little or no warning, as do auto accidents, heart attacks, influenza, and random acts of violence. To protect against accidents or the flu, you might decide to stay indoors, never venturing outside. But by doing so, you trade one set of risks for another; while you are inside, you are vulnerable to building collapse. There are too many possible causes of harm for us to protect ourselves—or our computers—completely against all of them.

In real life we make decisions every day about the best way to provide our security. For example, although we may choose to live in an area that is not prone to earthquakes, we cannot eliminate earthquake risk entirely. Some choices are conscious, such as deciding not to walk down a dark alley in an unsafe neighborhood; other times our subconscious guides us, from experience or expertise, to take some precaution. We evaluate the likelihood and severity of harm, and then consider ways (called countermeasures or controls) to address threats and determine the controls' effectiveness.

Computer security is similar. Because we cannot protect against everything, we prioritize: Only so much time, energy, or money is available for protection, so we address some risks and let others slide. Or we consider alternative courses of action, such as transferring risk by purchasing insurance or even doing nothing if the side effects of the countermeasure could be worse than the possible harm. The risk that remains uncovered by controls is called **residual risk**.

A simplistic model of risk management involves a user calculating the value of all assets, determining the amount of harm from all possible threats, computing the costs of protection, selecting safeguards (that is, controls or countermeasures) based on the degree of risk and on limited resources, and applying the safeguards to optimize harm averted. This approach to risk management is a logical and sensible approach to protection, but it has significant drawbacks. In reality, it is difficult to assess the value of each asset; as we have seen, value can change depending on context, timing, and a host of other characteristics. Even harder is determining the impact of all possible threats.

Short- and Long-Term Risks of Security Breaches — Sidebar 1-5

It was long assumed that security breaches would be bad for business: that customers, fearful of losing their data, would veer away from insecure businesses and toward more secure ones. But empirical studies suggest that the picture is more complicated. Early studies of the effects of security breaches, such as that of Campbell [CAM03], examined the effects of breaches on stock price. They found that a breach's impact could depend on the nature of the breach itself; the effects were higher when the breach involved unauthorized access to confidential data. Cavusoglu et al. [CAV04] discovered that a breach affects the value not only of the company experiencing the breach but also of security enterprises: On average, the breached firms lost 2.1 percent of market value within two days of the breach's disclosure, but security *developers'* market value actually *increased* 1.36 percent.

Myung Ko and Carlos Dorantes [KO06] looked at the longer-term financial effects of publicly announced breaches. Based on the Campbell et al. study, they examined data for four quarters following the announcement of unauthorized access to confidential data. Ko and Dorantes note many types of possible breach-related costs:

> Examples of short-term costs include cost of repairs, cost of replacement of the system, lost business due to the disruption of business operations, and lost productivity of employees. These are also considered tangible costs. On the other hand, long-term costs include the loss of existing customers due to loss of trust, failing to attract potential future customers due to negative reputation from the breach, loss of business partners due to loss of trust, and potential legal liabilities from the breach. Most of these costs are intangible costs that are difficult to calculate but extremely important in assessing the overall security breach costs to the organization.

Ko and Dorantes compared two groups of companies: one set (the treatment group) with data breaches, and the other (the control group) without a breach but matched for size and industry. Their findings were striking. Contrary to what you might suppose, the breached firms had no decrease in performance for the quarters following the breach, but their return on assets decreased in the third quarter. The comparison of treatment with control companies revealed that the control firms generally outperformed the breached firms. However, the breached firms outperformed the control firms in the fourth quarter.

These results are consonant with the results of other researchers who conclude that there is minimal long-term economic impact from a security breach. There are many reasons why this is so. For example, customers may think that all competing firms have the same vulnerabilities and threats, so changing to another vendor does not reduce the risk. Another possible explanation may be a perception that a breached company has better security since the breach forces the company to strengthen controls and thus reduce the likelihood of similar breaches. Yet another explanation may simply be the customers' short attention span; as time passes, customers forget about the breach and return to business as usual.

All these studies have limitations, including small sample sizes and lack of sufficient data. But they clearly demonstrate the difficulties of quantifying and verifying the impacts of security risks, and point out a difference between short- and long-term effects.

The range of possible threats is effectively limitless, and it is difficult (if not impossible in some situations) to know the short- and long-term impacts of an action. For instance, Sidebar 1-5 describes a study of the impact of security breaches over time on corporate finances, showing that a threat must be evaluated over time, not just at a single instance.

Perception of the Risk of Extreme Events — Sidebar 1-6

When a type of adverse event happens frequently, we can calculate its likelihood and impact by examining both the frequency and nature of the collective set of events. For instance, we can calculate the likelihood that it will rain this week and take an educated guess at the number of inches of precipitation we will receive; rain is a fairly frequent occurrence. But security problems are often extreme events: They happen infrequently and under a wide variety of circumstances, so it is difficult to look at them as a group and draw general conclusions.

Paul Slovic's work on risk addresses the particular difficulties with extreme events. He points out that evaluating risk in such cases can be a political endeavor as much as a scientific one. He notes that we tend to let values, process, power, and trust influence our risk analysis [SLO99].

Beginning with Fischoff et al. [FIS78], researchers characterized extreme risk along two perception-based axes: the dread of the risk and the degree to which the risk is unknown. These feelings about risk, called *affects* by psychologists, enable researchers to discuss relative risks by placing them on a plane defined by the two perceptions as axes. A study by Loewenstein et al. [LOE01] describes how risk perceptions are influenced by association (with events already experienced) and by affect at least as much if not more than by reason. In fact, if the two influences compete, feelings usually trump reason.

This characteristic of risk analysis is reinforced by prospect theory: studies of how people make decisions using reason and feeling. Kahneman and Tversky [KAH79a] showed that people tend to overestimate the likelihood of rare, unexperienced events because their feelings of dread and the unknown usually dominate analytical reasoning about the low likelihood of occurrence. By contrast, if people experience similar outcomes and their likelihood, their feeling of dread diminishes and they can actually underestimate rare events. In other words, if the impact of a rare event is high (high dread), then people focus on the impact, regardless of the likelihood. But if the impact of a rare event is small, then they pay attention to the likelihood.

Although we should not apply protection haphazardly, we will necessarily protect against threats we consider most likely or most damaging. For this reason, it is essential to understand how we perceive threats and evaluate their likely occurrence and impact. Sidebar 1-6 summarizes some of the relevant research in risk perception and decision-making. Such research suggests that, for relatively rare instances such as high-impact security problems, we must take into account the ways in which people focus more on the impact than on the actual likelihood of occurrence.

Let us look more carefully at the nature of a security threat. We have seen that one aspect—its potential harm—is the amount of damage it can cause; this aspect is the **impact** component of the risk. We also consider how great is the threat's **likelihood**. A likely threat is not just one that someone might want to pull off but rather one that could actually occur. Some people might daydream about getting rich by robbing a bank; most, however, would reject that idea because of its difficulty (if not its immorality or risk). One aspect of likelihood is feasibility: Is it even possible to accomplish the attack? If the answer is no, then the likelihood is zero, and therefore so is the risk. So a good place to start in assessing risk is to look at whether the proposed action is feasible. Three factors determine feasibility, as we describe next.

Method–Opportunity–Motive

A malicious attacker must have three things to ensure success: method, opportunity, and motive, depicted in Figure 1-10. Deny the attacker any of those three and the attack will not succeed. Let us examine these properties individually.

Method

By **method** we mean the skills, knowledge, tools, and other things with which to perpetrate the attack. Think of comic figures that want to do something, for example, to steal valuable jewelry, but the characters are so inept that their every move is doomed to fail. These people lack the capability or method to succeed, in part because there are no classes in jewel theft or books on burglary for dummies.

There are plenty of courses and books about computing, however. Knowledge of specific models of computer systems is widely available in bookstores and on the

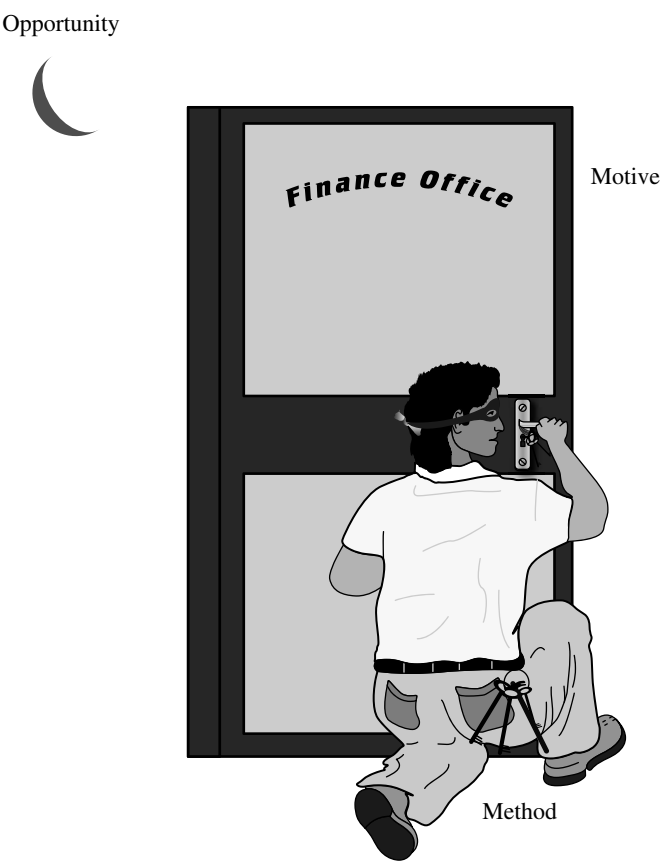


FIGURE 1-10 Method–Opportunity–Motive

Internet. Mass-market systems (such as the Microsoft or Apple or Unix operating systems) are readily available for purchase, as are common software products, such as word processors or database management systems, so potential attackers can even get hardware and software on which to experiment and perfect an attack. Some manufacturers release detailed specifications on how the system was designed or operates, as guides for users and integrators who want to implement other complementary products. Various attack tools—scripts, model programs, and tools to test for weaknesses—are available from hackers' sites on the Internet, to the degree that many attacks require only the attacker's ability to download and run a program. The term **script kiddie** describes someone who downloads a complete attack code package and needs only enter a few details to identify the target and let the script perform the attack. Often, only time and inclination limit an attacker.

Opportunity

Opportunity is the time and access to execute an attack. You hear that a fabulous apartment has just become available, so you rush to the rental agent, only to find someone else rented it five minutes earlier. You missed your opportunity.

Many computer systems present ample opportunity for attack. Systems available to the public are, by definition, accessible; often their owners take special care to make them fully available so that if one hardware component fails, the owner has spares instantly ready to be pressed into service. Other people are oblivious to the need to protect their computers, so unattended laptops and unsecured network connections give ample opportunity for attack. Some systems have private or undocumented entry points for administration or maintenance, but attackers can also find and use those entry points to attack the systems.

Motive

Finally, an attacker must have a **motive** or reason to want to attack. You probably have ample opportunity and ability to throw a rock through your neighbor's window, but you do not. Why not? Because you have no reason to want to harm your neighbor: You lack motive.

We have already described some of the motives for computer crime: money, fame, self-esteem, politics, terror. It is often difficult to determine motive for an attack. Some places are "attractive targets," meaning they are very appealing to attackers. Popular targets include law enforcement and defense department computers, perhaps because they are presumed to be well protected against attack (so that they present a challenge: a successful attack shows the attacker's prowess). Other systems are attacked because they are easy to attack. And other systems are attacked at random simply because they are there.

By demonstrating feasibility, the factors of method, opportunity, and motive determine whether an attack can succeed. These factors give the advantage to the attacker because they are qualities or strengths the attacker must possess. Another factor, this time giving an advantage to the defender, determines whether an attack will succeed: The attacker needs a vulnerability, an undefended place to attack. If the defender removes vulnerabilities, the attacker cannot attack.

VULNERABILITIES

As we noted earlier in this chapter, a **vulnerability** is a weakness in the security of the computer system, for example, in procedures, design, or implementation, that might be exploited to cause loss or harm. Think of a bank, with an armed guard at the front door, bulletproof glass protecting the tellers, and a heavy metal vault requiring multiple keys for entry. To rob a bank, you would have to think of how to exploit a weakness not covered by these defenses. For example, you might bribe a teller or pose as a maintenance worker.

Computer systems have vulnerabilities, too. In this book we consider many, such as weak authentication, lack of access control, errors in programs, finite or insufficient resources, and inadequate physical protection. Paired with a credible attack, each of these vulnerabilities can allow harm to confidentiality, integrity, or availability. Each attack vector seeks to exploit a particular vulnerability.

Our next step is to find ways to block threats by neutralizing vulnerabilities.

CONTROLS

A **control** or **countermeasure** is a means to counter threats. Harm occurs when a threat is realized against a vulnerability. To protect against harm, then, we can neutralize the threat, close the vulnerability, or both. The possibility for harm to occur is called **risk**. We can deal with harm in several ways:

- **prevent** it, by blocking the attack or closing the vulnerability
- **deter** it, by making the attack harder but not impossible
- **deflect** it, by making another target more attractive (or this one less so)
- **mitigate** it, by making its impact less severe
- **detect** it, either as it happens or some time after the fact
- **recover** from its effects

Of course, more than one of these controls can be used simultaneously. So, for example, we might try to prevent intrusions—but if we suspect we cannot prevent all of them, we might also install a detection device to warn of an imminent attack. And we should have in place incident-response procedures to help in the recovery in case an intrusion does succeed.

To consider the controls or countermeasures that attempt to prevent exploiting a computing system's vulnerabilities, we begin by thinking about traditional ways to enhance physical security. In the Middle Ages, castles and fortresses were built to protect the people and valuable property inside. The fortress might have had one or more security characteristics, including

- a strong gate or door to repel invaders
- heavy walls to withstand objects thrown or projected against them
- a surrounding moat to control access
- arrow slits to let archers shoot at approaching enemies
- crenellations to allow inhabitants to lean out from the roof and pour hot or vile liquids on attackers

- a drawbridge to limit access to authorized people
- a portcullis to limit access beyond the drawbridge
- gatekeepers to verify that only authorized people and goods could enter

Similarly, today we use a multipronged approach to protect our homes and offices. We may combine strong locks on the doors with a burglar alarm, reinforced windows, and even a nosy neighbor to keep an eye on our valuables. In each case, we select one or more ways to deter an intruder or attacker, and we base our selection not only on the value of what we protect but also on the effort we think an attacker or intruder will expend to get inside.

Computer security has the same characteristics. We have many controls at our disposal. Some are easier than others to use or implement. Some are cheaper than others to use or implement. And some are more difficult than others for intruders to override. Figure 1-11 illustrates how we use a combination of controls to secure our valuable resources. We use one or more controls, according to what we are protecting, how the cost of protection compares with the risk of loss, and how hard we think intruders will work to get what they want.

In this section, we present an overview of the controls available to us. In the rest of this book, we examine how to use controls against specific kinds of threats.

We can group controls into three largely independent classes. The following list shows the classes and several examples of each type of control.

- **Physical** controls stop or block an attack by using something tangible, such as
  - walls and fences
  - locks
  - (human) guards
  - sprinklers and other fire extinguishers

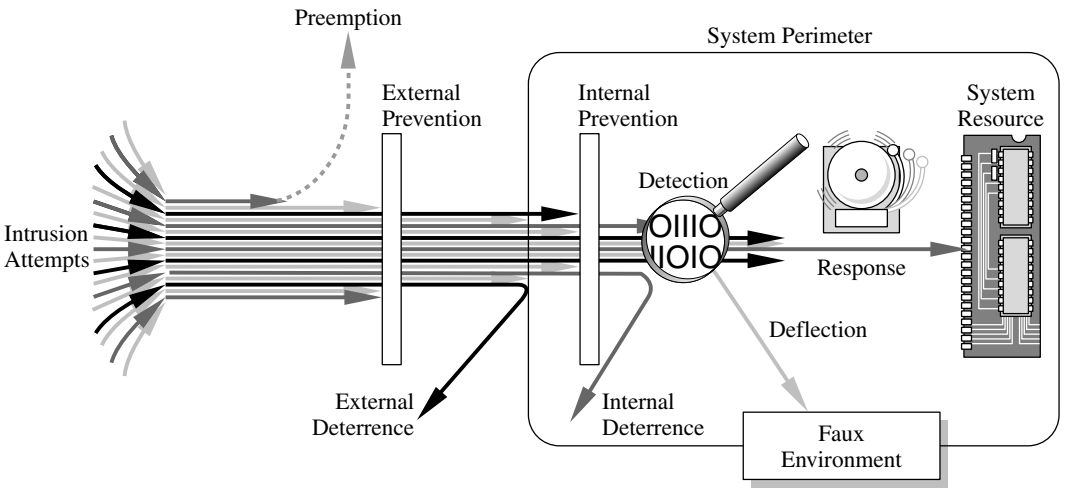


FIGURE 1-11 Effects of Controls

- **Procedural** or **administrative** controls use a command or agreement that requires or advises people how to act; for example,
  - laws, regulations
  - policies, procedures, guidelines
  - copyrights, patents
  - contracts, agreements
- **Technical** controls counter threats with technology (hardware or software), including
  - passwords
  - access controls enforced by an operating system or application
  - network protocols
  - firewalls, intrusion detection systems
  - encryption
  - network traffic flow regulators

(Note that the term "logical controls" is also used, but some people use it to mean administrative controls, whereas others use it to mean technical controls. To avoid confusion, we do not use that term.)

As shown in Figure 1-12, you can think in terms of the property to be protected and the kind of threat when you are choosing appropriate types of countermeasures. None of these classes is necessarily better than or preferable to the others; they work in different ways with different kinds of results. And it can be effective to use **overlapping** controls or **defense in depth**: more than one control or more than one class of control to achieve protection.

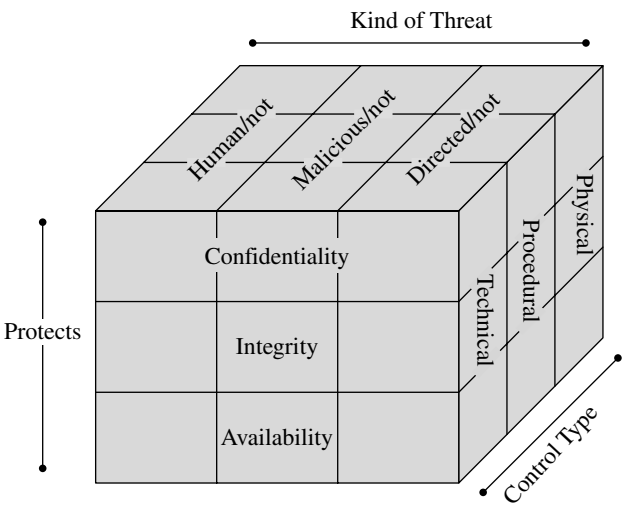


FIGURE 1-12 Types of Countermeasures

ANALYZING SECURITY WITH EXAMPLES

In the remainder of this book we study computer security by using the threat–vulnerability–control paradigm. That is, we begin each chapter with an example of either a real attack that caused harm or a series of attacks. The remaining chapters address confidentiality of messages, integrity of stored code, correctness of data on a video screen, and availability of network access, among other things. Our cases involve political figures, high school students, countries, government agencies, executives, and ordinary users, which should convince you that computer security affects everyone.

You will encounter examples involving email, missile systems, hospitals, mobile phones, spacecraft, and diplomats. Do not fear; you need not know rocket science to appreciate the security aspect of the examples. This variety of examples should help you appreciate (and convince other people) that there are important security aspects of many important current activities. Computer security analysts like to be involved early in the design of a system, product, or solution; there are many possible countermeasures from which to choose, and they can be selected and integrated more easily and effectively during system requirements definition and design rather than later in development. Being handed an already completed product or system and told to "secure this" is often an impossible task.

From each example we identify four things:

1. *Threat*. What threat is being raised? How does it work? On what does it depend? Who are the potential attackers? What are the potential attacks (also called **threat agents**)? What tools and knowledge are needed to realize the attack?
2. *Harm*. What harm can or did this attack cause? If the attack can support other attacks, what are they? How serious is the harm?
3. *Vulnerability*. What vulnerability is being exploited? Is it a general weakness or specific to one computer or situation? Is there more than one vulnerability? Are all vulnerabilities required for the threat to be actualized?
4. *Control*. How can the vulnerability be controlled? Does the control nullify the threat or close the vulnerability? Is there more than one control? If yes, do they overlap (and complement each other)? Are the controls partial or complete? Are the controls strong or can they be defeated or bypassed? Are they expensive or hard to use?

These four categories are the basis of all computer security planning, and they form the structure of the rest of this book.

In this book you will encounter attacks with intriguing names like masquerade, ping of death, salami, and man in the middle, as well as terms you may have heard before like virus, worm, and Trojan horse. We also describe a wide range of countermeasures, from defensive programming to biometric authentication and secure protocol design to digital signatures. Do not worry if any of these terms is unfamiliar; you will find complete explanations of all.

CONCLUSION

Computer security attempts to ensure the confidentiality, integrity, and availability of computing systems and their components. Three principal parts of a computing system are subject to attacks: hardware, software, and data. These three, and the communications among them, are susceptible to computer security vulnerabilities. In turn, those people and systems interested in compromising a system can devise attacks that exploit the vulnerabilities.

In each chapter of this book we include a list of the important points you should have learned in this chapter. For example, in this chapter we have explained the following concepts:

- Security situations arise in many everyday activities, although sometimes it can be difficult to distinguish between a security attack and an ordinary human or technological breakdown. Alas, clever attackers realize this confusion, so they may make their attack seem like a simple, random failure.
- A threat is an incident that could cause harm. A vulnerability is a weakness through which harm could occur. These two problems combine: Either without the other causes no harm, but a threat exercising a vulnerability means damage. To control such a situation, we can either block or diminish the threat, or close the vulnerability (or both).
- Seldom can we achieve perfect security: no viable threats and no exercisable vulnerabilities. Sometimes we fail to recognize a threat, or other times we may be unable or unwilling to close a vulnerability. Incomplete security is not a bad situation; rather, it demonstrates a balancing act: Control certain threats and vulnerabilities, apply countermeasures that are reasonable, and accept the risk of harm from uncountered cases.
- An attacker needs three things: method—the skill and knowledge to perform a successful attack; opportunity—time and access by which to attack; and motive—a reason to want to attack. Alas, none of these three is in short supply, which means attacks are inevitable.

In this chapter we introduced the notions of threats and harm, vulnerabilities, attacks and attackers, and countermeasures. Attackers leverage threats that exploit vulnerabilities against valuable assets to cause harm, and we hope to devise countermeasures to eliminate means, opportunity, and motive. These concepts are the basis we need to study, understand, and master computer security.

Countermeasures and controls can be applied to the data, the programs, the system, the physical devices, the communications links, the environment, and the personnel. Sometimes several controls are needed to cover a single vulnerability, but sometimes one control addresses many problems at once.

Throughout this book we use a scenario-based format to explore examples of attacks and countermeasures that can control them: First the attack that could or did occur; then the weakness that allowed the attack to succeed, with perhaps some attention to tools, techniques, or knowledge the attacker needed; and finally the countermeasures that can or could offer protection. When possible we present a range of countermeasures so you have a palette of options to apply to future scenarios or situations outside this book.

As you look at countermeasures, keep in mind the balance between risk and control: Does this situation warrant that level (degree, severity, cost) of countermeasure and are there simpler countermeasures that would provide adequate security?

Because the book is organized around types of attacks, we describe vulnerabilities and countermeasures relevant to the specific attacks. Some countermeasures, such as authentication and access control, are effective against many attacks; consequently, we sometimes (as with access control) introduce the topic in one chapter and expand upon it in later chapters. In other cases, as with program development controls, we explore the topic once and simply refer to it when it is relevant in a later scenario.

We think the threat–vulnerability–countermeasure structure gives you the opportunity to analyze these cases on your own. You may think of vulnerabilities we have not listed, and you will almost certainly be able to think of additional countermeasures that could be effective. Computer security is always changing to address new attacks and new technological advances; you do not learn one set of tools or one approach and say you know all there is to know. The breadth and nature of attacks continues to change and grow, as do the means of defense. Our goal is to help you to think critically and creatively in order to be able to address ever-changing threats.

Several themes recur throughout the book: privacy, legal matters, economics, ethics, usability, and forensics. These areas are tangential to security: Each is an important area of study by itself, but at points throughout this book, one or another will be relevant to a particular topic. Rather than have a chapter on each that might get lost or overlooked, we treat these topics when they are relevant, as part of the flow of the main chapters. This arrangement emphasizes that these themes relate to the core content of computer and information security.

To give you additional practice analyzing security, we include three chapters, which we call interludes, in which we present just a bare scenario and invite you to derive the threats, potential vulnerabilities, and countermeasures. The three topics are cloud computing, electronic voting, and cyberwarfare; these interludes are placed among the other chapters.

We also conclude each chapter with exercises to help reinforce what you have learned and let you apply that knowledge in different settings.

EXERCISES

1. List at least three kinds of harm a company could experience from electronic espionage or unauthorized viewing of company confidential materials.
2. List at least three kinds of harm a student could experience from electronic espionage or unauthorized viewing of personal materials.
3. Describe a situation in which complete denial of service to a user (that is, the user gets no response from the computer) is a serious problem to that user. Describe a situation in which 10% denial of service (that is, the response from the computer is 10% slower than normal) is a serious problem to a user.
4. Consider the web site of an organization many people would support, for example, an environmental group or a charity. List at least three classes of people who might attack that web site. What are their motives? Consider the web site of a controversial organization, for example, a group of extreme ideology. List at least three classes of people who might attack

that web site. What are their motives? Can you build a list of three classes that would attack both types of sites?

5. Do you think attempting to break in to (that, is obtain access to or use of) a computing system is ethical? Why or why not? Do you think that act should be illegal? Why or why not? Base your answer on harm: Who is harmed, to what degree, and does benefit to the person breaking in override the harm?

6. Consider electronic medical records. Which of confidentiality, integrity, and availability do their users require? Cite examples of each of these properties you think are required. Describe at least two kinds of people or situations that could threaten each property you name.

7. Distinguish among threat, threat agent, vulnerability, harm, and control.

8. Not all kinds of computer harm are illegal. List five examples of harm that are not illegal.

9. Consider the example with which this chapter began: a series of seemingly unrelated events, including failure of the communications and electrical power networks. Describe a scenario in which these could all occur concurrently but not be related. Describe a way at least one could lead to another. Describe a way you could determine the root cause of each failure.

10. Continuing from Exercise 9, suppose you were a malicious agent assigned to cause failure of the telecommunications and electric power systems. What steps could you take to make it difficult to determine who you are? What steps could you take to make it difficult to determine that the attack was malicious and not a natural accident? What steps could you take to make it seem as though the cause was someone else, for example, a particular foreign country?

11. Consider a restaurant with an online reservation system for patrons. What confidentiality, integrity, and availability threats might such a system experience? Hypothesize vulnerabilities in such a system that an attacker might try to exploit. What countermeasures could be applied against these threats?

12. Suppose a payroll system secretly leaks a list of names of employees earning more than a certain amount each pay period. Who would be harmed by such a vulnerability? How could such a vulnerability come about? What controls could be instituted to counter such a vulnerability? Suppose the leakage was not just names but also employees' identification numbers and full pay amounts. Would the people harmed or the degree of harm be different? Why or why not? If the employees are the ones suffering the greatest harm, who should be responsible for countering this vulnerability: the employee or the employer? Why?

13. A letter arrives in the surface mail apparently from your bank, but you are skeptical of its origin. What factors would make you skeptical? How could the bank help allay your skepticism in advance of sending the letter? What could the bank put in the letter itself that would reduce your skepticism? Would your answers be the same if the bank sends email instead of a surface mail letter?

14. Consider a program you could install on your own personal web page to display your city's current time and temperature. What threats could this program cause to you? To people who visit your web site? What controls could counter those threats?

15. Consider a program that allows people to order goods over the Internet. What threats could this program cause to users (purchasers)? What threats could this program cause to the merchant? Hypothesize three vulnerabilities that could allow these threats to be actualized.

16. Suppose you are a talented sailor about to race your boat in a yachting competition. A possible threat to winning is cancellation of the event because of adverse weather conditions. List three other threats you might encounter as you try to win by posting the fastest finishing time. List three vulnerabilities those threats might exploit. List three countermeasures against those threats.

17. Suppose you are a spy, and you need to pass secret materials to another spy, Agent Smart. However, you and Smart have never before met. You are aware that hostile forces are all around, any one of whom might try to impersonate Smart; if you approach someone and asked if she is Agent Smart, she might say she is even if she is not. Suggest a control for this threat—that is, a way you could be convinced the person to whom you are talking is really Agent Smart. Would your technique work if you assumed your telephone and mail were being monitored by the hostile agents? Suggest a way that would work even if your communications were monitored.

Index

Index pages in **bold** point to the definition or explanation of an item.

Symbols

A

B

C

D

E

F

G

H

J

K

L

M

N

O

P

S

T

U

V

W

X

Y

Z

This page intentionally left blank

Register the Addison-Wesley, Exam Cram, Que, and Sams products you own to unlock great benefi ts.

To begin the registration process, simply go to **informit.com/register** to sign in or create an account. You will then be prompted to enter the 10- or 13-digit ISBN that appears on the back cover of your product.

Registering your products can unlock the following benefits:

- Access to supplemental content, including bonus chapters, source code, or project files.
- A coupon to be used on your next purchase.

Registration benefits vary by product. Benefits will be listed on your Account page under Registered Products.