# 1

# Introduction

**In this chapter:**

- Threats, vulnerabilities, and controls
- Confidentiality, integrity, and availability
- Attackers and attack types; method, opportunity, and motive
- Valuing assets

On 11 February 2013, residents of Great Falls, Montana received the following warning on their televisions [INF13]. The transmission displayed a message banner on the bottom of the screen (as depicted in Figure 1-1).

**FIGURE 1-1** Emergency Broadcast Warning

And the following alert was broadcast:

> [Beep Beep Beep: *the sound pattern of the U.S. government Emergency Alert System. The following text then scrolled across the screen:*]
>
> Civil authorities in your area have reported that the bodies of the dead are rising from their graves and attacking the living. Follow the messages on screen that will be updated as information becomes available.
>
> Do not attempt to approach or apprehend these bodies as they are considered extremely dangerous. This warning applies to all areas receiving this broadcast.
>
> [Beep Beep Beep]

The warning signal sounded authentic; it had the distinctive tone people recognize for warnings of serious emergencies such as hazardous weather or a natural disaster. And the text was displayed across a live broadcast television program. On the other hand, bodies rising from their graves sounds suspicious.

What would you have done?

Only four people contacted police for assurance that the warning was indeed a hoax. As you can well imagine, however, a different message could have caused thousands of people to jam the highways trying to escape. (On 30 October 1938 Orson Welles performed a radio broadcast of the H. G. Wells play *War of the Worlds* that did cause a minor panic of people believing that Martians had landed and were wreaking havoc in New Jersey.)

The perpetrator of this hoax was never caught, nor has it become clear exactly how it was done. Likely someone was able to access the system that feeds emergency broadcasts to local radio and television stations. In other words, a hacker probably broke into a computer system.

You encounter computers daily in countless situations, often in cases in which you are scarcely aware a computer is involved, like the emergency alert system for broadcast media. These computers move money, control airplanes, monitor health, lock doors, play music, heat buildings, regulate hearts, deploy airbags, tally votes, direct communications, regulate traffic, and do hundreds of other things that affect lives, health, finances, and well-being. Most of the time these computers work just as they should. But occasionally they do something horribly wrong, because of either a benign failure or a malicious attack.

This book is about the security of computers, their data, and the devices and objects to which they relate. In this book you will learn some of the ways computers can fail—or be made to fail—and how to protect against those failures. We begin that study in the way any good report does: by answering the basic questions of what, who, why, and how.

## 1.1 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 its 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 recreate or replace. These assets are all shown in Figure 1-2.

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.

**Computer systems—hardware, software, and data—have value and deserve security protection.**

In this book, unless we specifically distinguish between hardware, software, and data, we refer to all these assets as the computer system,

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 1-2** Computer Objects of Value

or sometimes as the computer. And because processors are embedded in so many devices, we also need to think about such variations as mobile phones, implanted pacemakers, heating controllers, 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.

## Values of Assets

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 and 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 1-3. 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 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 there is no other copy. 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

**Assets' values are personal, time dependent, and often imprecise.**

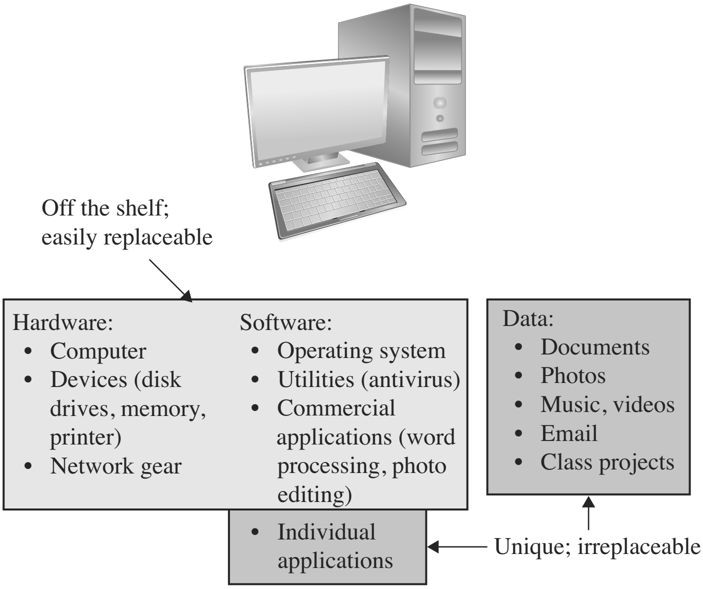


**FIGURE 1-3** Values of Assets

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 vulnerability is a weakness that could be exploited to cause harm.**

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 1-4. 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.

**A threat is a set of circumstances that could cause harm.**

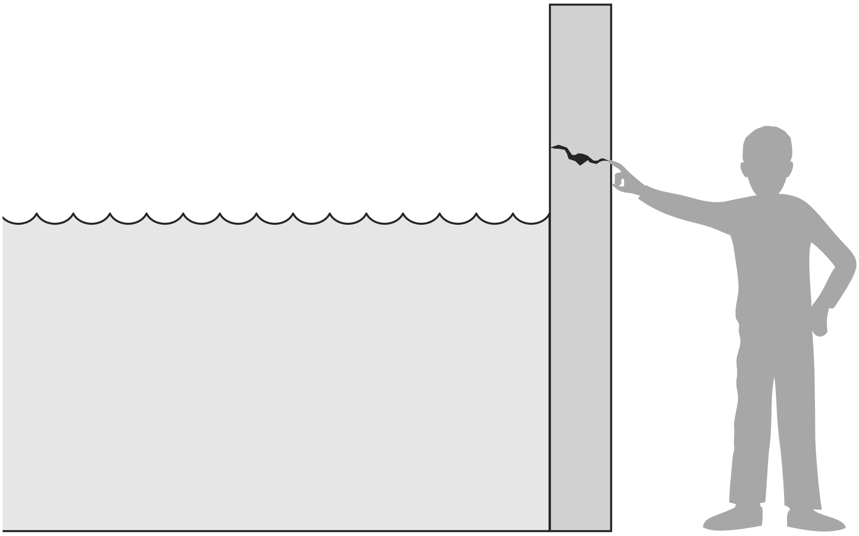


**FIGURE 1-4** Threat and Vulnerability

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.

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 Chapter 6.)

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 1-4, 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 between threats, controls, and vulnerabilities in this way:

**Controls prevent threats from exercising vulnerabilities.**

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

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

## 1.2 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 computer. 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 that no one can copy it without your permission.

These three aspects, confidentiality, integrity, and availability, 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

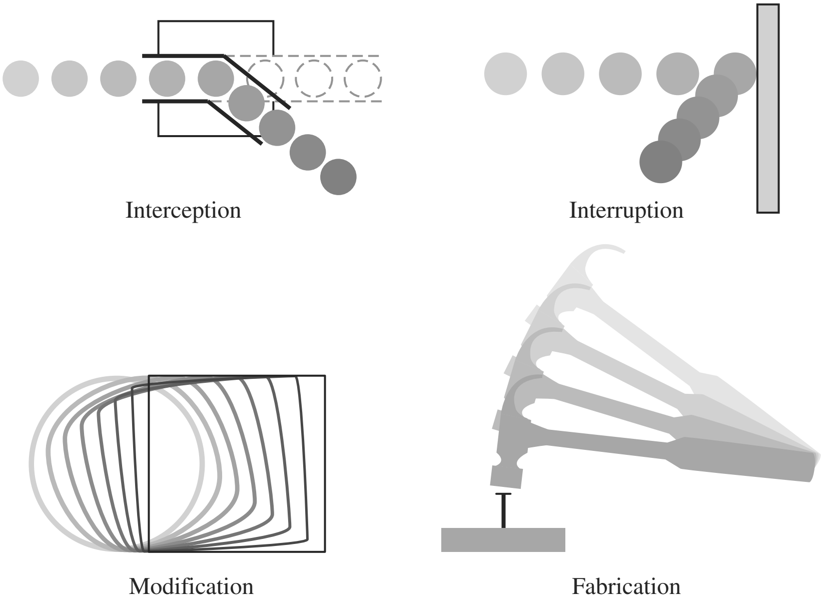
These three properties, hallmarks of solid security, appear in the literature as early as James P. Anderson's essay on computer security [AND73] and reappear frequently in more recent computer security papers and discussions. Taken together (and rearranged), the properties are called the **C-I-A triad** 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. Authenticity 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.

**C-I-A triad: confidentiality, integrity, availability**

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 compromised. 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.



Interception

Interruption

Modification

Fabrication

**FIGURE 1-5** Four Acts to Cause Security Harm

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**. These four acts are depicted in Figure 1-5. 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. 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 underway 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 1-6. These four terms 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.

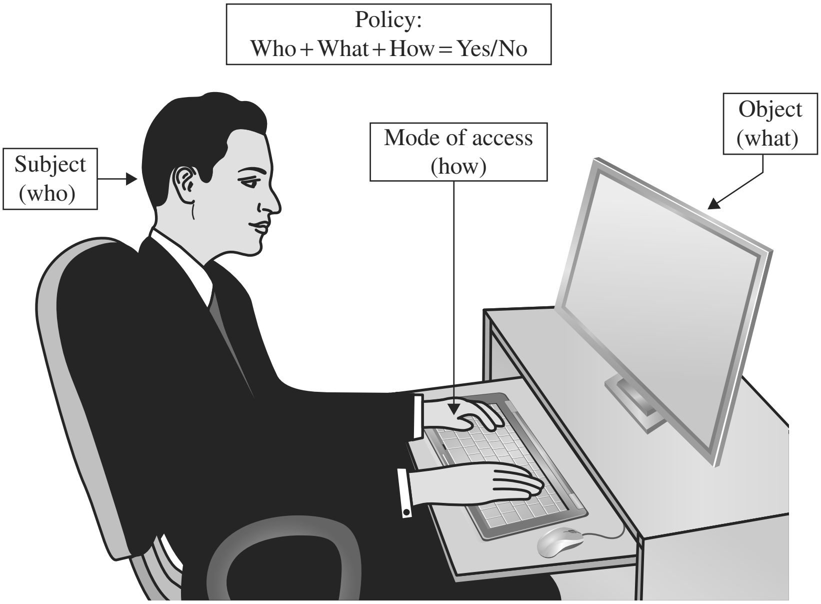


FIGURE 1-6 Access Control

## 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. A worse situation occurs when that inaccuracy is 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 Stephen Welke and Terry Mayfield [WEL90, MAY91, NCS91a] 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 confidentiality, 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 1-7 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.

**Computer security seeks to prevent unauthorized viewing (confidentiality) or modification (integrity) of data while preserving access (availability).**

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

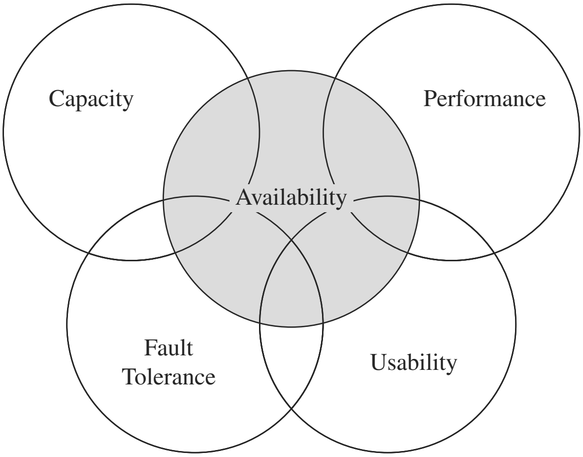


**FIGURE 1-7** Availability and Related Aspects

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 1-8, 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, major corporations, or government departments would have one, Ware's discussion is still instructive today. 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**. Harm can be caused by either nonhuman events or humans. Examples of **nonhuman threats** include natural disasters

**Threats are caused both by human and other sources.**

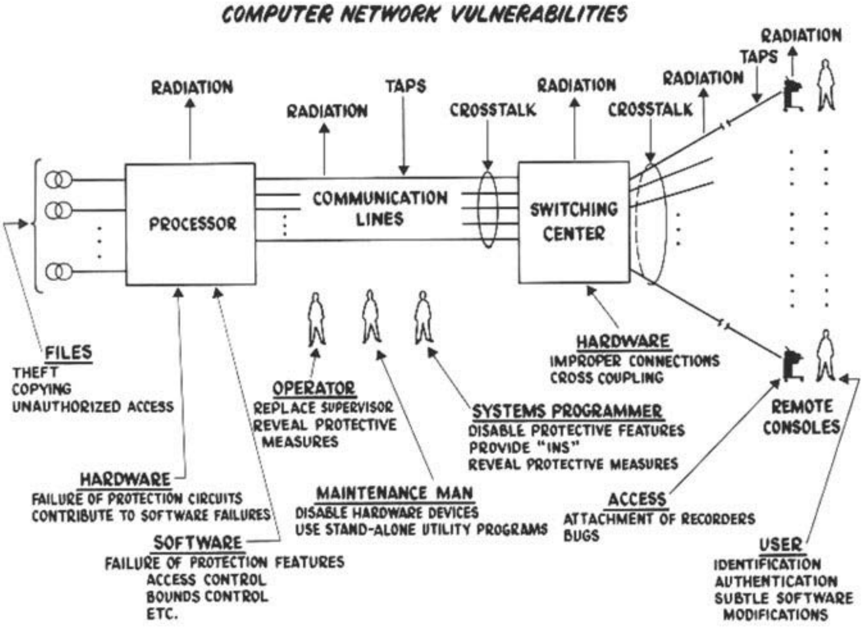


**FIGURE 1-8** Computer [Network] Vulnerabilities (from [WAR70])

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.

**Human threats** can be either benign (nonmalicious) or malicious. **Nonmalicious** kinds of harm include someone's 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.

**Threats can be malicious or not.**

Most computer security activity relates to **malicious**, **human-caused harm**: A malicious person 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 website 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. Different kinds of threats are shown in Figure 1-9.

**Threats can be targeted or random.**

Although the distinctions shown in Figure 1-9 seem clear-cut, sometimes the nature of an attack is not obvious until the attack is well underway, 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 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. The Common Vulnerabilities and Exposures (CVE) list (see http://cve.mitre.org/) is a dictionary of publicly known 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, the Common Vulnerability Scoring System (CVSS) (see http://nvd.nist.gov/cvss.cfm) provides a standard measurement system that allows accurate and consistent scoring of vulnerability impact.

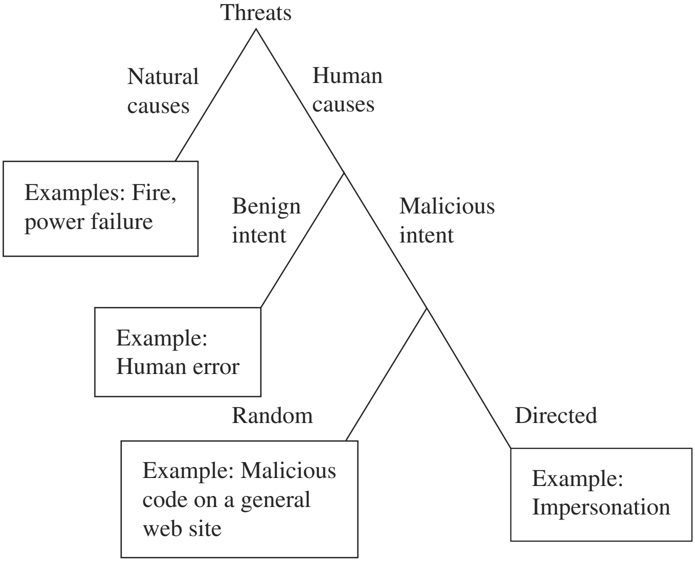


**FIGURE 1-9** Kinds of Threats

### Advanced Persistent Threat

Security experts are becoming increasingly concerned about a type of threat called **advanced persistent threat**. A lone attacker might create a random attack that snares a few, or a few million, individuals, but the resulting impact is limited to what that single attacker can organize and manage. A collection of attackers—think, for example, of the cyber equivalent of a street gang or an organized crime squad—might work together to purloin credit card numbers or similar financial assets to fund other illegal activity. Such attackers tend to be opportunistic, picking unlucky victims' pockets and moving on to other activities.

Advanced persistent threat attacks come from organized, well financed, patient assailants. Often affiliated with governments or quasi-governmental groups, these attackers engage in long term campaigns. They carefully select their targets, crafting attacks that appeal to specifically those targets; email messages called spear phishing (described in Chapter 4) are intended to seduce their recipients. Typically the attacks are silent, avoiding any obvious impact that would alert a victim, thereby allowing the attacker to exploit the victim's access rights over a long time.

The motive of such attacks is sometimes unclear. One popular objective is economic espionage. A series of attacks, apparently organized and supported by the Chinese government, was used in 2012 and 2013 to obtain product designs from aerospace companies in the United States. There is evidence the stub of the attack code was loaded into victim machines long in advance of the attack; then, the attackers installed the more complex code and extracted the desired data. In May 2014 the Justice Department indicted five Chinese hackers in absentia for these attacks.

In the summer of 2014 a series of attacks against J.P. Morgan Chase bank and up to a dozen similar financial institutions allowed the assailants access to 76 million names, phone numbers, and email addresses. The attackers—and even their country of origin—remain unknown, as does the motive. Perhaps the attackers wanted more sensitive financial data, such as account numbers or passwords, but were only able to get the less valuable contact information. It is also not known if this attack was related to an attack a year earlier that disrupted service to that bank and several others.

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 the attackers are 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 1-1. These studies are far from conclusive, however, and the traits they identify may show correlation but not necessarily causality. To appreciate this point, suppose a study found that a disproportionate number of people convicted of computer crime were left-handed. Does that result imply that all left-handed people are computer criminals or that only left-handed people are? Certainly not. 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 1-10, attackers look just like anybody in a crowd.

**No one pattern matches all attackers.**

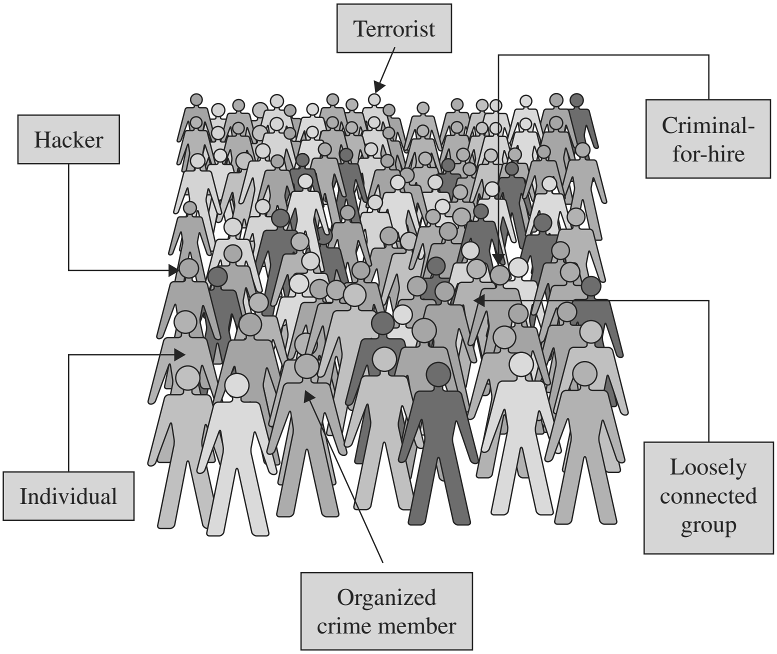


FIGURE 1-10 Attackers

## SIDEBAR 1-1 An Attacker's Psychological Profile?

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

(*continues*)

## SIDEBAR 1-1 *Continued*

Donn Parker [PAR98] has studied hacking and computer crime for many 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 piece of attack software] 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 sophisticated attack. But he was "quite shocked" to hear they were used for harm.

More research is needed before we can 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.

### Individuals

Originally, computer attackers were individuals, acting with motives of fun, challenge, or revenge. Early attackers acted alone. Two of the most well known among them are 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].

### 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 13) 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 an international, 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, 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]

### 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*, 26 March 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 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]

**Organized crime groups are discovering that computer crime can be lucrative.**

McAfee also describes the case of a hacker-for-hire: a businessman who hired a 16-year-old New Jersey hacker to attack the websites 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 crude attack, whereas the professional attacker wants a neat, robust, and undetectable 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 website 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 13.
- *Computer as method of attack*: Launching offensive attacks requires the use of computers. Stuxnet, an example of 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. (We examine the Stuxnet worm in more detail in Chapters 6 and 13.)
- *Computer as enabler of attack*: Websites, 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.

We cannot accurately measure the degree to which terrorists use computers, because terrorists keep secret the nature of their activities and because our definitions and measurement tools are rather weak. Still, incidents like the one described in Sidebar 1-2 provide evidence that all four of these activities are increasing.

### SIDEBAR 1-2 The Terrorists, Inc., IT Department

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 11 September 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 to act immediately if it suddenly happens.

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.

## 1.3 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 denied access to data. These events cause harm that we want to avoid.

In our earlier discussion of assets, we noted that value depends 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 you 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 effect 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 websites 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 website administration credentials, costing only $2 to $30. These black market websites 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 management involves choosing which threats to control and what resources to devote to protection.**

## 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 entirely eliminate earthquake risk. 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 basic model of risk management involves a user's 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. 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 1-3 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.

### SIDEBAR 1-3 Short- and Long-term Risks of Security Breaches

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

(*continues*)

## SIDEBAR 1-3 *Continued*

> 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.

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 1-4 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.

## SIDEBAR 1-4 Perception of the Risk of Extreme Events

When a type of adverse event happens frequently, we can calculate its likelihood and impact by examining both 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 by using reason and feeling. Kahneman and Tversky [KAH79] 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.

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 the magnitude of 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.

**Spending for security is based on the impact and likelihood of potential harm—both of which are nearly impossible to measure precisely.**

## Method–Opportunity–Motive

A malicious attacker must have three things to ensure success: method, opportunity, and motive, depicted in Figure 1-11. Roughly speaking, method is the how; opportunity, the when; and motive, the why of an attack. 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.

Anyone can find plenty of courses and books about computing, however. Knowledge of specific models of computer systems is widely available in bookstores and on

**FIGURE 1-11** Method–Opportunity–Motive

the 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 how it 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 to 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 they present a challenge and a successful attack shows the attacker's prowess). Other systems are attacked because they are easy to attack. And some systems are attacked at random simply because they are there.

**Method, opportunity, and motive are all necessary for an attack to succeed; deny any of these and the attack will fail.**

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.

## 1.4 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.

**Vulnerabilities are weaknesses that can allow harm to occur.**

Security analysts speak of a system's **attack surface**, which is the system's full set of vulnerabilities—actual and potential. Thus, the attack surface includes physical hazards, malicious attacks by outsiders, stealth data theft by insiders, mistakes, and impersonations. Although such attacks range from easy to highly improbable, analysts must consider all possibilities.

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

## 1.5 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.

**Security professionals balance the cost and effectiveness of controls with the likelihood and severity of harm.**

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 1-12 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 too, such as walls and fences
  - locks

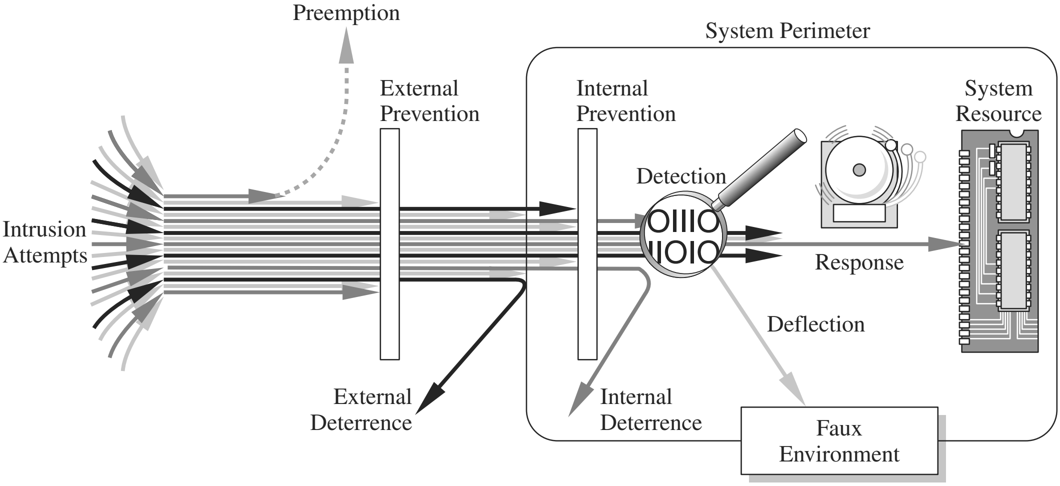


**FIGURE 1-12** Effects of Controls

  - (human) guards
  - sprinklers and other fire extinguishers
- **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
  - program or operating system access controls
  - 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 1-13, 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.

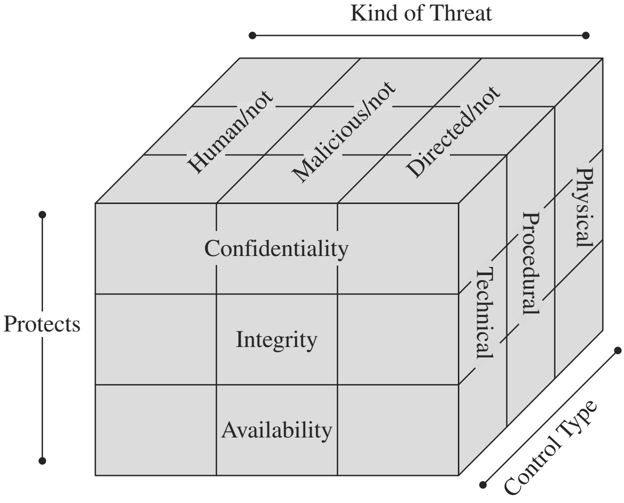


**FIGURE 1-13** Types of Countermeasures

## 1.6 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 this chapter we have explained the following computer security 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 have 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.

## 1.7 WHAT'S NEXT?

The rest of this book is organized around the major aspects or pieces of computer security. As you have certainly seen in almost daily news reports, computer security incidents abound. The nature of news is that failures are often reported, but seldom successes. You almost never read a story about hackers who tried to break into the computing system of a bank but were foiled because the bank had installed strong, layered defenses. In fact, attacks repelled far outnumber those that succeed, but such good situations do not make interesting news items.

Still, we do not want to begin with examples in which security controls failed. Instead, in Chapter 2 we begin by giving you descriptions of three powerful and widely used security protection methods. We call these three our security toolkit, in part because they are effective but also because they are applicable. We refer to these techniques in probably every other chapter of this book, so we want not only to give them a prominent position up front but also to help lodge them in your brain. Our three featured tools are identification and authentication, access control, and encryption.

After presenting these three basic tools we lead into domains in which computer security applies. We begin with the simplest computer situations, individual programs, and explore the problems and protections of computer code in Chapter 3. We also consider malicious code, such as viruses and Trojan horses (defining those terms along with other types of harmful programs). As you will see in other ways, there is no magic that can make bad programs secure or turn programmers into protection gurus. We do, however, point out some vulnerabilities that show up in computer code and describe ways to counter those weaknesses, both during program development and as a program executes.

Modern computing involves networking, especially using the Internet. We focus first on how networked computing affects individuals, primarily through browsers and other basic network interactions such as email. In Chapter 4 we look at how users can be tricked by skillful writers of malicious code. These attacks tend to affect the protection of confidentiality of users' data and integrity of their programs.

Chapter 5 covers operating systems, continuing our path of moving away from things the user can see and affect directly. We see what protections operating systems can provide to users' programs and data, most often against attacks on confidentiality or integrity. We also see how the strength of operating systems can be undermined by attacks, called rootkits, that directly target operating systems and render them unable to protect themselves or their users.

In Chapter 6 we return to networks, this time looking at the whole network and its impact on users' abilities to communicate data securely across the network. We also study a type of attack called denial of service, just what its name implies, that is the first major example of a failure of availability.

We consider data, databases, and data mining in Chapter 7. The interesting cases involve large databases in which confidentiality of individuals' private data is an objective. Integrity of the data in the databases is also a significant concern.

In Chapter 8 we move even further from the individual user and study cloud computing, a technology becoming quite popular. Companies are finding it convenient and cost effective to store data "in the cloud," and individuals are doing the same to have shared access to things such as music and photos. There are security risks involved in this movement, however.

You may have noticed our structure: We organize our presentation from the user outward through programs, browsers, operating systems, networks, and the cloud, a progression from close to distant. In Chapter 9 we return to the user for a different reason: We consider privacy, a property closely related to confidentiality. Our treatment here is independent of where the data are: on an individual computer, a network, or a database. Privacy is a property we as humans deserve, and computer security can help preserve it, as we present in that chapter.

In Chapter 10 we look at several topics of management of computing as related to security. Security incidents occur, and computing installations need to be ready to respond, whether the cause is a hacker attack, software catastrophe, or fire. Managers also have to decide what controls to employ, because countermeasures cost money that must be spent wisely. Computer security protection is hard to evaluate: When it works you do not know it does. Performing risk analysis and building a case for security are important management tasks.

Some security protections are beyond the scope an individual can address. Organized crime from foreign countries is something governments must deal with, through a legal system. In Chapter 11 we consider laws affecting computer security. We also look at ethical standards, what is "right" in computing.

In Chapter 12 we return to cryptography, which we introduced in Chapter 2. Cryptography merits courses and textbooks of its own, and the topic is detailed enough that most of the real work in the field is done at the graduate level and beyond. We use Chapter 2 to introduce the concepts enough to be able to apply them. In Chapter 12 we expand upon that introduction and peek at some of the formal and mathematical underpinnings of cryptography.

Finally, in Chapter 13 we raise four topic areas. These are domains with an important need for computer security, although the areas are evolving so rapidly that computer security may not be addressed as fully as it should. These areas are the so-called Internet of Things (the interconnection of network-enabled devices from toasters to automobiles and insulin pumps), computer security economics, electronic voting, and computer-assisted terrorism and warfare.

We trust this organization will help you to appreciate the richness of an important field that touches many of the things we depend on.

## 1.8 EXERCISES

1. Distinguish between vulnerability, threat, and control.
2. Theft usually results in some kind of harm. For example, if someone steals your car, you may suffer financial loss, inconvenience (by losing your mode of transportation), and emotional upset (because of invasion of your personal property and space). List three kinds of harm a company might experience from theft of computer equipment.
3. List at least three kinds of harm a company could experience from electronic espionage or unauthorized viewing of confidential company materials.
4. List at least three kinds of damage a company could suffer when the integrity of a program or company data is compromised.
5. List at least three kinds of harm a company could encounter from loss of service, that is, failure of availability. List the product or capability to which access is lost, and explain how this loss hurts the company.
6. Describe a situation in which you have experienced harm as a consequence of a failure of computer security. Was the failure malicious or not? Did the attack target you specifically or was it general and you were the unfortunate victim?
7. Describe two examples of vulnerabilities in automobiles for which auto manufacturers have instituted controls. Tell why you think these controls are effective, somewhat effective, or ineffective.
8. One control against accidental software deletion is to save all old versions of a program. Of course, this control is prohibitively expensive in terms of cost of storage. Suggest a less costly control against accidental software deletion. Is your control effective against all possible causes of software deletion? If not, what threats does it not cover?
9. On your personal computer, who can install programs? Who can change operating system data? Who can replace portions of the operating system? Can any of these actions be performed remotely?
10. Suppose a program to print paychecks secretly leaks a list of names of employees earning more than a certain amount each month. What controls could be instituted to limit the vulnerability of this leakage?
11. Preserving confidentiality, integrity, and availability of data is a restatement of the concern over interruption, interception, modification, and fabrication. How do the first three concepts relate to the last four? That is, is any of the four equivalent to one or more of the three? Is one of the three encompassed by one or more of the four?
12. Do you think attempting to break in to (that is, obtain access to or use of) a computing system without authorization should be illegal? Why or why not?

**13.** Describe an example (other than the ones mentioned in this chapter) of data whose confidentiality has a short timeliness, say, a day or less. Describe an example of data whose confidentiality has a timeliness of more than a year.

**14.** Do you currently use any computer security control measures? If so, what? Against what attacks are you trying to protect?

**15.** Describe an example in which absolute denial of service to a user (that is, the user gets no response from the computer) is a serious problem to that user. Describe another example where 10 percent denial of service to a user (that is, the user's computation progresses, but at a rate 10 percent slower than normal) is a serious problem to that user. Could access by unauthorized people to a computing system result in a 10 percent denial of service to the legitimate users? How?

**16.** When you say that software is of high quality, what do you mean? How does security fit in your definition of quality? For example, can an application be insecure and still be "good"?

**17.** Developers often think of software quality in terms of faults and failures. Faults are problems (for example, loops that never terminate or misplaced commas in statements) that developers can see by looking at the code. Failures are problems, such as a system crash or the invocation of the wrong function, that are visible to the user. Thus, faults can exist in programs but never become failures, because the conditions under which a fault becomes a failure are never reached. How do software vulnerabilities fit into this scheme of faults and failures? Is every fault a vulnerability? Is every vulnerability a fault?

**18.** Consider a program to display on your website your city's current time and temperature. Who might want to attack your program? What types of harm might they want to cause? What kinds of vulnerabilities might they exploit to cause harm?

**19.** Consider a program that allows consumers to order products from the web. Who might want to attack the program? What types of harm might they want to cause? What kinds of vulnerabilities might they exploit to cause harm?

**20.** Consider a program to accept and tabulate votes in an election. Who might want to attack the program? What types of harm might they want to cause? What kinds of vulnerabilities might they exploit to cause harm?

**21.** Consider a program that allows a surgeon in one city to assist in an operation on a patient in another city via an Internet connection. Who might want to attack the program? What types of harm might they want to cause? What kinds of vulnerabilities might they exploit to cause harm?