CHAPTER 10

ELECTRONIC COMMERCE SECURITY

LEARNING OBJECTIVES

In this chapter, you will learn about:

- Online security issues
- Security for client computers
- Security for the communication channels between computers
- Security for server computers
- Organizations that promote computer, network, and Internet security

INTRODUCTION

In 2002, the U.S. Congress held hearings to review the federal government's computer security status. The results were not encouraging. The **General Accounting Office (GAO)** summarized its previous two years' work in reviewing security at 24 government agencies. According to the GAO, 16 of those agencies had failed completely in their computer security efforts, and all 24 had at least one major security weakness.

Most of the security problems identified by the GAO did not involve sophisticated technological issues, nor did they require large amounts of money to resolve. The most prevalent security weaknesses stemmed from inadequate employee training and awareness and failure to keep software updated with the

latest security patches available. The most common problem was failure to enforce basic standards for access control, such as rotating passwords periodically and having employees maintain the confidentiality of their passwords.

In many of the agencies, readily available security patches for well-known vulnerabilities had not been applied to system software. The GAO noted that more than 90 percent of all successful attacks on U.S. government agency systems had exploited known vulnerabilities for which a patch was available but had not been installed. The GAO concluded that by simply adhering to their own existing policies, these agencies could improve their level of computer security significantly. In many cases, the agencies had not made any person responsible for monitoring vulnerabilities and for ensuring that available solutions were applied. The GAO report emphasized that this state of affairs was unacceptable, especially in the wake of the terrorist attacks of September 11, 2001.

When businesses began using computers 50 years ago, security was accomplished by using physical controls over access to the computers. Alarmed doors and windows, guards, security badges to admit people to sensitive areas, and surveillance cameras were the tools used to secure computers. Back then, interactions between people and computers were limited to terminals (which had no internal processing capabilities) connected directly to large mainframe computers. There were no other connections to computers, and there were very few networks of computers (and those few networks did not extend outside the organization in which they existed). Computer security meant dealing with the few people who had access to terminals or physical access to the computer room. In many computer installations of the day, people ran programs by submitting decks of punched cards that were fed into card readers. The card readers translated the punched holes in the cards into electrical impulses that were processed by the computer. The computer printed out the results when it was finished running the program. When program submitters returned to the computer operations center (often the next day; computers were not very fast then), they could pick up the printouts and reclaim their punched card decks from the input/output clerk. Security was a pretty simple matter.

Both the population of computer users and the methods to access computing resources have increased tremendously since those early years of computing. Millions of people now have access to computing power over both private and public networks that connect millions of computers. It is no longer a simple matter to determine who is using a computing resource. A user in South Africa could be using a computer in California. New security tools and methods have evolved and are employed today to protect computers and the electronic assets they store. The transmission of valuable information, such as electronic

receipts, purchase orders, payment data, and order confirmations, has drastically increased the need for security and new automatic methods to deal with security threats.

Data security measures date back to the time of the Roman Empire, when Julius Caesar coded information to prevent enemies from reading secret war and defense plans carried by his Roman legions. Many modern electronic security techniques were developed for wartime use. The U.S. Department of Defense was the main driving force behind early security requirements and more recent advances. In the late 1970s, the Defense Department formed a committee to develop computer security guidelines for handling classified information on computers. The result of that committee's work was *Trusted Computer System Evaluation Criteria*, known in defense circles as the "Orange Book" because its cover was orange. It spelled out rules for mandatory access control—the separation of confidential, secret, and top secret information—and established criteria for certification levels for computers ranging from D (not trusted to handle multiple levels of classified documents at once) to A1 (the most trustworthy level).

This early security work has been helpful because it provided a basis for electronic commerce security research. This research today provides commercial security products and practical security techniques. This early work also helped current security efforts by developing formal approaches to security analysis and evaluation, including the explicit evaluation and management of risk.

ONLINE SECURITY ISSUES OVERVIEW

In the early days of the Internet, one of its most popular uses was electronic mail. Despite e-mail's popularity, people have often worried that a business rival might intercept e-mail messages for competitive gain. Another fear was that employees' nonbusiness correspondence might be read by their supervisors, with negative repercussions. These were significant and realistic concerns.

Today, the stakes are much higher. The consequences of a competitor having unauthorized access to messages and digital intelligence are now far more serious than in the past. Electronic commerce, in particular, makes security a concern for all users. A typical worry of Web shoppers is that their credit card numbers might be exposed to millions of people as the information travels across the Internet. Recent surveys show that more than 80 percent of all Internet users have at least "some concern" about the security of their credit card numbers in electronic commerce transactions. This echoes the fear shoppers have expressed for many years about credit card purchases over the phone.

Consumers are now more comfortable giving their credit card numbers and other information over the phone, but many of those same people fear providing that same information on a Web site. As you learned in Chapter 7, people are concerned about personal information they provide to companies over the Internet. Increasingly, people doubt that these companies have the willingness and the ability to keep customers' personal information confidential. This chapter examines security in the context of electronic commerce, presenting an introduction to important security problems and some solutions to those problems.

Computer security is the protection of assets from unauthorized access, use, alteration, or destruction. There are two general types of security: physical and logical. **Physical security** includes tangible protection devices, such as alarms, guards, fireproof doors, security fences, safes or vaults, and bombproof buildings. Protection of assets using nonphysical means is called **logical security**. Any act or object that poses a danger to computer assets is known as a **threat**.

Managing Risk

Countermeasure is the general name for a procedure, either physical or logical, that recognizes, reduces, or eliminates a threat. The extent and expense of countermeasures can vary, depending on the importance of the asset at risk. Threats that are deemed low risk and unlikely to occur can be ignored when the cost to protect against the threat exceeds the value of the protected asset. For example, it would make sense to protect from tornadoes a computer network in Oklahoma City, where there is significant and regular tornado activity, but not to protect a similar network in Los Angeles, where tornadoes are rare. The risk management model shown in Figure 10-1 illustrates four general actions that an organization could take, depending on the impact (cost) and the probability of the physical threat. In this model, a tornado in Oklahoma would be in quadrant II, whereas a tornado in Southern California would be in quadrant IV.

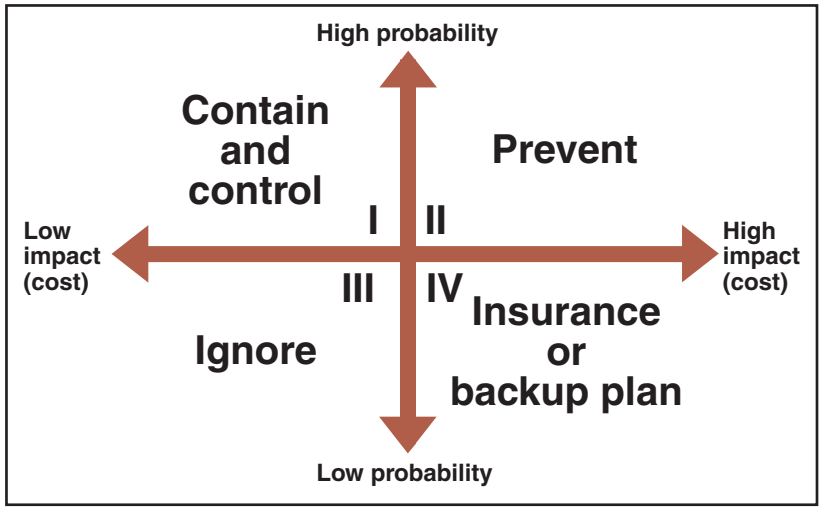

FIGURE 10-1 Risk management model

The same sort of risk management model applies to protecting Internet and electronic commerce assets from both physical and electronic threats. Examples of the latter include impostors, eavesdroppers, and thieves. An **eavesdropper**, in this context, is a person or device that can listen in on and copy Internet transmissions. People who write programs or manipulate technologies to obtain unauthorized access to computers and networks are called **crackers** or **hackers**.

A cracker is a technologically skilled person who uses their skills to obtain unauthorized entry into computers or network systems—usually with the intent of stealing information or damaging the information, the system's software, or even the system's hardware. Originally, the term hacker was used to describe a dedicated programmer who enjoyed writing complex code that tested the limits of technology. Although the term hacker is still used in a positive way—even as a compliment—by computer professionals (who make a strong distinction between the terms hacker and cracker), the media and the general public usually use the term to describe those who use their skills for ill purposes. Some IT people also use the terms **white hat hacker** and **black hat hacker** to make the distinction between good hackers and bad hackers.

To implement a good security scheme, organizations must identify risks, determine how to protect threatened assets, and calculate how much to spend to protect those assets. In this chapter, the primary focus in risk management protection is on the central issues of identifying the threats and determining the ways to protect assets from those threats, rather than on the protection costs or value of assets.

Computer Security Classifications

Computer security is generally classified into three categories: secrecy, integrity, and necessity (also known as denial of service). **Secrecy** refers to protecting against unauthorized data disclosure and ensuring the authenticity of the data source. **Integrity** refers to preventing unauthorized data modification. **Necessity** refers to preventing data delays or denials (removal). Secrecy is the best known of the computer security categories. Every month, newspapers report on break-ins to government computers or theft of stolen credit card numbers that are used to order goods and services. Integrity threats are reported less frequently and, thus, may be less familiar to the public. For example, an integrity violation occurs when an Internet e-mail message is intercepted and its contents are changed before it is forwarded to its original destination. In this type of integrity violation, which is called a **man-in-the-middle exploit**, the contents of the e-mail are often changed in a way that negates the message's original meaning. Necessity violations take several forms, and they occur relatively frequently. Delaying a message or completely destroying it can have grave consequences. Suppose that a message sent at 10:00 a.m. to an online stockbroker includes an order to purchase 1000 shares of IBM at market price. If the stockbroker does not receive the message (because an attacker delays it) until 2:30 p.m. and IBM's stock price has increased by $3, the buyer loses $3000.

Security Policy and Integrated Security

Any organization concerned about protecting its electronic commerce assets should have a **security policy** in place. A security policy is a written statement describing which assets to protect and why they are being protected, who is responsible for that protection, and which behaviors are acceptable and which are not. The policy primarily addresses physical security, network security, access authorizations, virus protection, and disaster recovery. The policy develops over time and is a living document that the company and security officer must review and update at regular intervals.

Both defense and commercial security guidelines state that organizations must protect assets from unauthorized disclosure, modification, or destruction. However, military

security policy differs from commercial policy because military applications stress separation of multiple levels of security. Corporate information is usually classified as either "public" or "company confidential." The typical security policy concerning confidential company information is straightforward: Do not reveal company confidential information to anyone outside the company.

The first step an organization must take in creating a security policy is to determine which assets to protect from which threats. For example, a company that stores its customers' credit card numbers might decide that those numbers are an asset that must be protected from eavesdroppers. Then, the organization must determine who should have access to various parts of the system. Next, the organization determines what resources are available to protect the assets identified. Using the information it has acquired, the organization develops a written security policy. Finally, the organization commits resources to building or buying software, hardware, and physical barriers that implement the security policy. For example, if a security policy disallows any unauthorized access to customer information, including credit card numbers and credit history, then the organization must either create or purchase software that guarantees end-to-end secrecy for electronic commerce customers.

A comprehensive plan for security should protect a system's privacy, integrity, and availability (necessity), and authenticate users. When these goals are used to create a security policy for an electronic commerce operation, they should be selected to satisfy the list of requirements shown in Figure 10-2. These requirements provide a minimum level of acceptable security for most electronic commerce operations.

Requirement	Meaning
Secrecy	Prevent unauthorized persons from reading messages and business plans, obtaining credit card numbers, or deriving other confidential information.
Integrity	Enclose information in a digital envelope so that the computer can automatically detect messages that have been altered in transit.
Availability	Provide delivery assurance for each message segment so that messages or message segments cannot be lost undetectably.
Key management	Provide secure distribution and management of keys needed to provide secure communications.
Nonrepudiation	Provide undeniable, end-to-end proof of each message's origin and recipient.
Authentication	Securely identify clients and servers with digital signatures and certificates.

FIGURE 10-2 Requirements for secure electronic commerce

The Network Security Library, which is sponsored by GFI Software (a company that sells security and messaging software), is a good source for information about security policies. The Network Security Library includes a number of white papers that provide guidance on how to craft a workable security policy. **Information Security Policy World** is another Web site that provides information about security policy matters.

Although absolute security is difficult to achieve, organizations can create enough barriers to deter most intentional violators. With good planning, organizations can also reduce the impact of natural disasters or terrorist acts. Integrated security means having all security measures working together to prevent unauthorized disclosure, destruction, or modification of assets. A security policy covers many security concerns that must be addressed by a comprehensive and integrated security plan. Specific elements of a security policy address the following points:

- *Authentication*: Who is trying to access the electronic commerce site?
- *Access control*: Who is allowed to log on to and access the electronic commerce site?
- *Secrecy*: Who is permitted to view selected information?
- *Data integrity*: Who is allowed to change data?
- *Audit*: Who or what causes specific events to occur, and when?

In this chapter, you will explore these security policy issues with a focus on how they apply to electronic commerce in particular. The electronic commerce security topics in this chapter are organized to follow the transaction processing flow, beginning with the consumer and ending with the Web server (or servers) at the electronic commerce site. Each logical link in the process includes assets that must be protected to ensure security: client computers, the communication channel on which the messages travel, and the Web servers, including any other computers connected to the Web servers.

SECURITY FOR CLIENT COMPUTERS

Client computers, usually PCs, must be protected from threats that originate in software and data that are downloaded to the client computer from the Internet. In this section, you will learn that active content delivered over the Internet in dynamic Web pages can be harmful. Another threat to client computers can arise when a malevolent server site masquerades as a legitimate Web site. Users and their client computers can be duped into revealing information to those Web sites. This section explains these threats, describes how they work, and outlines some protection mechanisms that can prevent or reduce the threats they pose to client computers.

Cookies

The Internet provides a type of connection between Web clients and servers called a stateless connection. In a **stateless connection**, each transmission of information is independent; that is, no continuous connection (also called an **open session**) is maintained between any client and server on the Internet. Earlier in this book, you learned that cookies are small text files that Web servers place on Web client computers to identify returning visitors. Cookies also allow Web servers to maintain continuing open sessions with Web clients. An open session is necessary to do a number of things that are important in

online business activity. For example, shopping cart and payment processing software both need an open session to work properly. Early in the history of the Web, cookies were devised as a way to maintain an open session despite the stateless nature of Internet connections. Thus, cookies were invented to solve the stateless connection problem by saving information about a Web user from one set of server-client message exchanges to another.

There are two ways of categorizing cookies: by time duration and by source. The two kinds of time duration cookie categories include **session cookies**, which exist until the Web client ends the connection (or "session"), and **persistent cookies**, which remain on the client computer indefinitely. Electronic commerce sites use both kinds of cookies. For example, a session cookie might contain information about a particular shopping visit and a persistent cookie might contain login information that can help the Web site recognize visitors when they return to the site on subsequent visits. Each time a browser moves to a different part of a merchant's Web site, the merchant's Web server asks the visitor's computer to send back any cookies that the Web server stored previously on the visitor's computer.

Another way of categorizing cookies is by their source. Cookies can be placed on the client computer by the Web server site, in which case they are called **first-party cookies**, or they can be placed by a different Web site, in which case they are called **third-party cookies**. A third-party cookie originates on a Web site other than the site being visited. These third-party Web sites usually provide advertising or other content that appears on the Web site being viewed. The third-party Web site providing the advertising is often interested in tracking responses to their ads by visitors who have already seen the ads on other sites. If the advertising Web site places its ads on a large number of Web sites, it can use persistent third-party cookies to track visitors from one site to another. Earlier in this book, you learned about DoubleClick and similar online ad placement services that perform this function.

The most complete way for Web site visitors to protect themselves from revealing private information or being tracked by cookies is to disable cookies entirely. The problem with this approach is that useful cookies are blocked along with the others, requiring visitors to enter information each time they revisit a Web site. The full resources of some sites are not available to visitors unless their browsers are set to allow cookies. For example, most distance learning software used by schools to deliver online courses does not work properly in student Web browsers unless cookies are enabled.

Web users can accumulate large numbers of cookies as they browse the Internet. Most Web browsers have settings that allow the user to refuse only third-party cookies or to review each cookie before it is accepted. Some browsers, such as **Netscape Navigator**, **Mozilla**, **Mozilla Firefox**, and **Opera**, provide comprehensive cookie management functions. Figure 10-3 shows the dialog box that can be used to manage stored cookies in the Mozilla Firefox Web browser.

Another approach is to use one of the many third-party programs, called **cookie blockers**, that prevent cookie storage selectively. Some of these programs, such as **WebWasher**, plug into a browser and allow users to block cookies from the Web servers that load advertising banners into Web pages. Other cookie blocking programs, such as **Cookie Pal**, allow cookies to be filtered by Internet (IP) address, allowing in the "good"

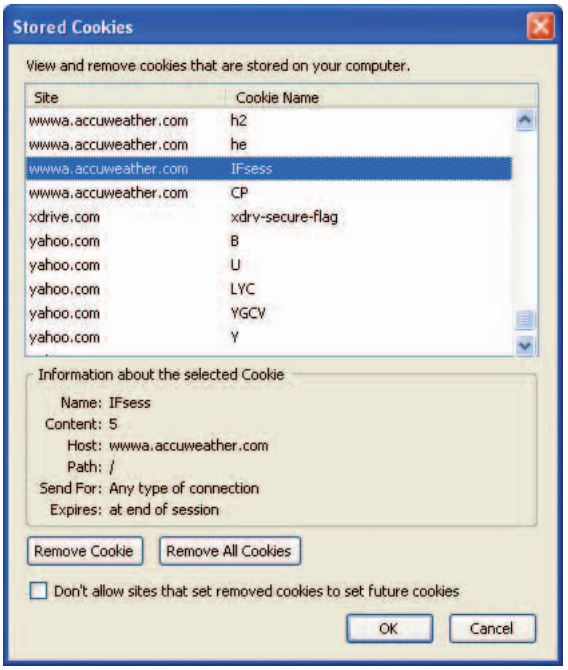

FIGURE 10-3 Mozilla Firefox dialog box for managing stored cookies

cookies and denying storage to all others. **Cookie Crusher** is another program that controls cookies before they are stored on a user's hard drive.

WebSideStory provides software that Web site managers can use to analyze Internet traffic at their sites. The company also sells a reporting service to Web sites that provides information about who visits their sites and what sites the visitors came from. WebSideStory's HitBox software collects and warehouses data from Web site visitors remotely, securely, and anonymously. The company does allow Web site visitors to opt out of these cookies. Figure 10-4 shows the WebSideStory Privacy Center Web page.

FIGURE 10-4 WebSideStory Privacy Center Web Page

Web Bugs

Some advertisers send images (from their third-party servers) that are included on Web pages, but are too small to be visible. A **Web bug** is a tiny graphic that a third-party Web site places on another site's Web page. When a site visitor loads the Web page, the Web bug is delivered by the third-party site, which can then place a cookie on the visitor's computer. A Web bug's only purpose is to provide a way for a third-party Web site (the identity of which is unknown to the visitor) to place cookies from that third-party site on the visitor's computer. The Internet advertising community sometimes calls Web bugs "clear GIFs" or "1-by-1 GIFs" because the graphics can be created in the GIF format with a color value of "transparent" and can be as small as 1 pixel by 1 pixel.

Active Content

Until the debut of executable Web content, Web pages could do little more than display content and provide links to related pages with additional information. The widespread use of active content has changed the situation. **Active content** refers to programs that are

embedded transparently in Web pages and that cause action to occur. For example, active content can display moving graphics, download and play audio, or implement Web-based spreadsheet programs. Active content is used in electronic commerce to place items into a shopping cart and compute a total invoice amount, including sales tax, handling, and shipping costs. Developers use active content because it extends the functionality of HTML and moves some data processing chores from the busy server machine to the user's client computer. Unfortunately, because active content elements are programs that run on the client computer, active content can damage the client computer. Thus, active content can pose a threat to the security of client computers.

Active content is provided in several forms. The best-known active content forms are cookies, Java applets, JavaScript, VBScript, and ActiveX controls. Other ways to provide Web active content include graphics, Web browser plug-ins, and e-mail attachments.

JavaScript and VBScript are **scripting languages**; they provide scripts, or commands, that are executed. An **applet** is a small application program. Applets typically run within the Web browser. Active content is launched in a Web browser automatically when that browser loads a Web page containing active content. The applet downloads automatically with the page and begins running. Depending on how the browser's security settings are configured, the browser might open a warning dialog box, such as the one shown in Figure 10-5, announcing the active content and asking the user for permission to open that content.

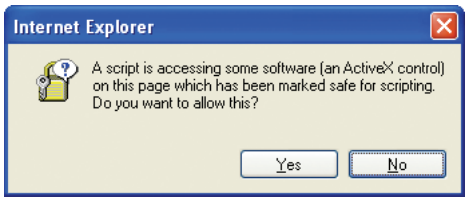

FIGURE 10-5 Dialog box asking for permission to open active content on a Web page

Because active content modules are embedded in Web pages, they can be completely transparent to anyone browsing a page containing them. Crackers intent on doing mischief to client computers can embed malicious active content in these seemingly innocuous Web pages. This delivery technique is called a Trojan horse. A **Trojan horse** is a program hidden inside another program or Web page that masks its true purpose. The Trojan horse could snoop around a client computer and send back private information to a cooperating Web server—a secrecy violation. The program could alter or erase information on a client computer—an integrity violation. Zombies are equally threatening. A **zombie** is a Trojan horse that secretly takes over another computer for the purpose of launching attacks on other computers. The computers running the zombie are also sometimes called zombies. Zombie attacks can be very difficult to trace to their creators.

Java Applets

Java is a programming language developed by Sun Microsystems that is used widely in Web pages to provide active content. The Web server sends the Java applets along with Web pages requested by the Web client. In most cases, the Java applet's operation will be visible to the site visitor; however, it is possible for a Java applet to perform functions that would not be noticed by the site visitor. The client computer then runs the programs within its Web browser. Java can also run outside the confines of a Web browser. Java is platform independent; that is, it can run on many different computers. This "develop once, deploy everywhere" feature reduces development costs because only one program needs to be developed for all operating systems.

Java adds functionality to business applications and can handle transactions and a wide variety of actions on the client computer. That relieves an otherwise busy server-side program from handling thousands of transactions simultaneously. Once downloaded, embedded Java code can run on a client's computer, which means that security violations can occur. To counter this possibility, a security model called the **Java sandbox** has been developed. The Java sandbox confines Java applet actions to a set of rules defined by the security model. These rules apply to all untrusted Java applets. **Untrusted Java applets** are those that have not been established as secure. When Java applets are run within the constraints of the sandbox, they do not have full access to the client system. For example, Java applets operating in the sandbox cannot perform file input, output, or delete operations. This prevents secrecy (disclosure) and integrity (deletion or modification) violations. You can follow the Online Companion link to the **Java Security Page** maintained by the Center for Education and Research in Information and Assurance (CERIAS) to learn more about Java applet security.

JavaScript

JavaScript is a scripting language developed by Netscape to enable Web page designers to build active content. Despite the similar-sounding names, JavaScript is based only loosely on Sun's Java programming language. Supported by popular Web browsers, JavaScript shares many of the structures of the full Java language. When a user downloads a Web page with embedded JavaScript code, it executes on the user's (client) computer.

Like other active content vehicles, JavaScript can be used for attacks by executing code that destroys the client's hard disk, discloses the e-mail stored in client mailboxes, or sends sensitive information to the attacker's Web server. JavaScript code can also record the URLs of Web pages a user visits and capture information entered into Web forms. For example, if a user enters credit card numbers while reserving a rental car, a JavaScript program could copy the credit card number. JavaScript programs, unlike Java applets, do not operate under the restrictions of the Java sandbox security model.

Unlike Java applets, a JavaScript program cannot commence execution on its own. To run an ill-intentioned JavaScript program, a user must start the program. For example, a site with a retirement income calculator might require a visitor to click a button to see a retirement income projection. Once the user clicks the button, the JavaScript program starts and does its work.

ActiveX Controls

An **ActiveX** control is an object that contains programs and properties that Web designers place on Web pages to perform particular tasks. ActiveX components can be constructed using many different programming languages, but the most common are C++ and Visual Basic. Unlike Java or JavaScript code, ActiveX controls run only on computers with Windows operating systems.

When a Windows-based Web browser downloads a Web page containing an embedded ActiveX control, the control is executed on the client computer. Other ActiveX controls include Web-enabled calendar controls and Web games. The **ActiveX** page at Download.com contains a comprehensive list of ActiveX controls.

The security danger with ActiveX controls is that once they are downloaded, they execute like any other program on a client computer. They have full access to all system resources, including operating system code. An ill-intentioned ActiveX control could reformat a user's hard disk, rename or delete files, send e-mails to all the people listed in the user's address book, or simply shut down the computer. Because ActiveX controls have full access to client computers, they can cause secrecy, integrity, or necessity violations. The actions of ActiveX controls cannot be halted once they begin execution. Most Web browsers can be configured to provide a notice when the user is about to download an ActiveX control. Figure 10-6 shows an example of the warning issued when Internet Explorer detects an ActiveX control.

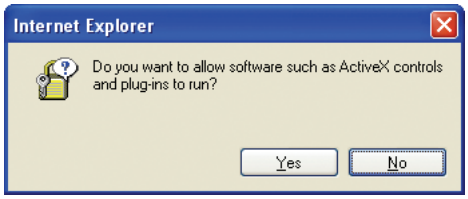

FIGURE 10-6 Internet Explorer ActiveX Control warning message

Graphics and Plug-Ins

Graphics, browser plug-ins, and e-mail attachments can harbor executable content. Some graphics file formats have been designed specifically to contain instructions on how to render a graphic. That means that any Web page containing such a graphic could be a threat because the code embedded in the graphic could cause harm to a client computer. Similarly, browser **plug-ins**, which are programs that enhance the capabilities of browsers, handle Web content that a browser cannot handle. Plug-ins are normally beneficial and perform tasks for a browser, such as playing audio clips, displaying movies, or animating graphics. Apple's QuickTime, for example, is a plug-in that downloads and plays movies stored in a special format.

Plug-ins can also pose security threats to a client computer. Users download these plug-in programs and install them so their browsers can display content that cannot be included in HTML tags. Popular plug-ins include Macromedia's Flash Player and Shockwave Player, Apple's QuickTime Player, and RealNetworks' RealPlayer.

In 1999, *The New York Times* revealed that RealNetworks had been using its RealPlayer plug-in to gather information surreptitiously from users. Downloaded and installed easily from the Internet, RealPlayer was recording user information such as the RealPlayer user's name, e-mail address, country, ZIP code, computer operating system, and other details. RealPlayer used the Internet connection to send the information it had gathered back to RealNetworks. Soon after the discovery, and after considerable public embarrassment, RealNetworks issued a statement that a software patch was available for all current users. The patch prevents the RealNetworks software from collecting and transmitting user information.

Many plug-ins execute commands buried within the media being manipulated. This opens the door to the possibility that someone intent on doing harm could embed commands within a seemingly innocuous video or audio clip. The ill-intentioned commands hidden within the object that the plug-in is interpreting could damage a client computer by erasing some (or all) of its files.

Viruses, Worms, and Antivirus Software

The potential dangers lurking in e-mail attachments get a lot of news coverage and are the most familiar to the general population. E-mail attachments provide a convenient way to send nontext information over a text-only system—electronic mail. Attachments can contain word-processing files, spreadsheets, databases, images, or virtually any other information you can imagine. Most programs, including Web browser e-mail programs, display attachments by automatically executing an associated program; for example, the recipient's Excel program reads an attached Excel workbook file and opens it, or Word opens and displays a Word document. Although this activity itself does not cause damage, Word and Excel macro viruses inside the loaded files can damage a client computer and reveal confidential information when those files are opened.

A virus is software that attaches itself to another program and can cause damage when the host program is activated. A worm is a type of virus that replicates itself on the computers that it infects. Worms can spread quickly through the Internet. A **macro virus** is a type of virus that is coded as a small program, called a macro, and is embedded in a file. You have probably read about or have personally experienced recent examples of e-mail attachment-borne virus attacks.

E-mail attachments containing viruses and other malicious software are reported daily. Some of the most famous in recent years include the ILOVEYOU virus, also known as the "love bug," and its variants. The ILOVEYOU virus was eventually traced to a 23-year-old computer science student who lived in the Philippines. The virus spread through the Internet with amazing speed as an e-mail message. It infected the computer of anyone who opened the e-mail attachment and clogged e-mail systems with thousands of copies of the useless e-mail message. The virus spread quickly because it automatically sent itself to as many as 300 addresses stored in a computer's Microsoft Outlook address book. Besides replicating itself explosively through e-mail, the virus caused other harm, destroying digital music and photo files stored on the target computers. The ILOVEYOU virus also searched for other users' passwords and forwarded that information to the original perpetrator. Within days, the virus spread to 40 million computers in more than 20 countries and caused an estimated $9 billion in damages—most of it in lost worker productivity.

In 2001, the incidences of virus and worm attacks increased. With more than 40,000 reported security violations occurring that year, the parade of attacks included Code Red and Nimda virus-worm combinations, each affecting millions of computers and costing billions of dollars to clean up. Both Code Red and Nimda are examples of a **multivector virus**, so called because they can enter a computer system in several different ways (vectors). Even though Microsoft issued security patches that should have stopped the Code Red virus-worm, it continued to propagate throughout the Internet in 2002. Both the original Code Red virus and a variant called Code Red 2 infected thousands of new computers during the year.

New virus-worm combinations also appeared in 2002 and 2003, including a version of the Code Red virus called Bugbear. Bugbear was spread through Microsoft Outlook e-mail clients. The person receiving the e-mail did not even have to click on an attachment to run the malicious code—Bugbear started itself through a security loophole in the connection between Outlook and the Internet Explorer browser. Of course, Microsoft issued a security patch for the browser, but many users did not install the patch (or, in many cases, even know about it). When launched, Bugbear first checked to see if the computer was running antivirus software. **Antivirus software** detects viruses and worms and either deletes them or isolates them on the client computer so they cannot run. If antivirus software existed on the system, Bugbear attempted to destroy it. Then it installed a Trojan horse program on the computer that let attackers access the computer through the Internet and upload or download files at will. (Bugbear was difficult to eliminate from an infected computer because it gave its own files a randomly generated name; thus, the virus files had different names on every infected computer.) Bugbear would then send out e-mail messages with attachments that would infect the recipients. It did not create its own e-mail messages, but took previously sent e-mail messages that were on the computer and resent them to different addresses. This often fooled recipients because the e-mail messages had subject headers that seemed normal and did not hint that the e-mail might contain a virus. Figure 10-7 summarizes some of the major viruses, worms, and Trojan horses that have plagued Internet users over the years.

Symantec and **McAfee**, among other companies, keep track of viruses and sell antivirus software. You can follow the links in the Online Companion to those companies to find descriptions of thousands of viruses. Antivirus software is only effective if the antivirus data files are kept current. The data files contain virus-identifying information that is used to detect viruses on a client computer. Because people generate new viruses by the hundreds every month, users must be vigilant and update their antivirus data files regularly so that the newest viruses are recognized and eliminated. Some Web e-mail systems, such as Yahoo! Mail, let users scan attachments using antivirus software before downloading e-mail. In these cases, the antivirus software is run by the Web site and the user does not need to take any action to keep the software updated.

Year	Name	Type	Description
1986	Brain	Virus	Written in Pakistan, this virus infects floppy disks used in personal computers at that time. It consumes empty space on the disks, preventing them from being used to store data or programs.
1988	Internet Worm	Worm	Robert Morris, Jr., a graduate student at Cornell University, wrote this experimental, self-replicating, self-propagating program and released it onto the Internet. It replicated faster than he had anticipated, crashing computers at universities, military sites, and medical research facilities throughout the world.
1991	Tequila	Virus	Tequila writes itself to a computer's hard disk and runs any time the computer is started. It also infects programs when they are executed. Tequila originated in Switzerland and was mostly transmitted through Internet downloads.
1992	Michelangelo	Trojan Horse	Set to activate on March 6 (Michelangelo's birthday), this Trojan Horse overwrites large portions of the infected computer's hard disk.
1993	SatanBug	Virus	Infects programs when they run, causing them to fail or perform incorrectly. SatanBug was designed to interfere with antivirus programs so they cannot detect it.
1996	Concept	Virus Worm	One of the first viruses to be written in Microsoft Word's macro language, Concept travels with infected Word document files. When an infected document is opened, Concept places macros in Word's default document template, which infects any new Word document created on that computer.
1999	Melissa	Virus Worm	Melissa is a Microsoft Word macro virus that spreads by e-mailing itself automatically from one user to another. It inserts comments from "The Simpsons" television show and confidential information from the infected computer. Melissa spread throughout the world in a few hours. Many large companies were inundated by Melissa. For example, Microsoft closed down its e-mail servers to prevent the spread of this virus within the company.
2000	ILOVEYOU	Virus Worm	Arrives attached to an e-mail message with the subject line "ILOVEYOU" and infects any computer on which the attachment is opened. It sends itself to addresses in any Microsoft Outlook address book it finds on the infected computer. The virus destroys music and photo files stored on the infected computers. When it was launched, it clogged e-mail servers in many large organizations and slowed down the operation of the entire Internet.
2001	Code Red	Virus Worm Trojan Horse	Code Red can infect Web servers and personal computers. It defaces Web pages and can be transmitted from Web servers to personal computers. It can give hackers control over Web server computers. Code Red can reinstall itself from hidden files after it is removed.

FIGURE 10-7 Major viruses, worms, and Trojan horses

Year	Name	Type	Description
2001	Nimda	Virus Worm	Nimda modifies Web documents and certain programs on the infected computer. It also creates multiple copies of itself using various file names. It can be transmitted by e-mail, a LAN, or from a Web server to a Web client.
2002	BugBear	Virus Worm Trojan Horse	BugBear is spread through e-mail and through local area networks. It identifies antivirus software and attempts to disable it. BugBear can log keystrokes and store them for later transmission through a Trojan Horse program that it installs on the infected computer.This program gives hackers access to the computer and allows file uploads and downloads.
2002	Klez	Virus Worm	Klez is transmitted as an e-mail attachment and overwrites files, creates hidden copies of the original files, and attempts to disable antivirus software.
2003	Slammer	Worm	Slammer's primary purpose was to demonstrate how rapidly a worm could be transmitted on the Internet. It infected 75,000 computers in its first ten minutes of propagation.
2003	Sobig	Trojan Horse	Sobig turns infected computers into spam relay points. Sobig transmits mass e-mails with copies of itself to potential victims.
2004	MyDoom	Worm Trojan Horse	MyDoom turns the infected computer into a zombie that will participate in a denial of service attack on a specific company's Web site.
2004	Sasser	Virus Worm	Written by a German high school student, Sasser finds computers with a specific security flaw and then infects them. The infected computers are slowed by the virus, often to the point that they must be rebooted.
2005	Zotob	Worm Trojan Horse	Zotob peforms port scans and infects computers that appear to have a specific security flaw. Once installed on a target computer, Zotob can log keystrokes, capture screens, and steal authentication credentials and CD software keys. Infected computers can also be used as zombies for mass mailing or attacking other computers.

FIGURE 10-7 Major viruses, worms, and Trojan horses (continued)

LEARNING FROM FAILURES

MICROSOFT INTERNET INFORMATION SERVER

As you learned in Chapter 8, Internet Information Server (IIS) is Microsoft's Web server software. Microsoft supplies versions of the IIS software with its Windows server operating systems that are suitable for use in operating electronic commerce Web sites.

In August 2001, Microsoft faced an uncomfortable situation that many U.S. manufacturing companies have experienced with recalled, defective products—Microsoft executives stood by at a news conference while a U.S. government official announced to gathered reporters that there was a serious flaw in a Microsoft product. The director of the FBI's National Infrastructure Protection Center was warning reporters that the Code Red worm, which was spreading through the Internet for the third time in as many weeks, was a serious threat to the continued operation of the Internet. A **worm** is a type of virus that replicates itself on the computers that it infects.

The Code Red worm exploits a vulnerability in the Microsoft IIS Web server software. When the worm was first identified, Microsoft rapidly made a patch available on its Web site. Microsoft also announced that Web server installations that had kept current with all of the updates and patches that Microsoft had issued would not be subject to attack by the worm.

Many Microsoft customers were outraged by these statements, noting that Microsoft had issued more than 40 software patches in the first half of 2001 and 100 or more patches in each of several prior years. IIS users complained that keeping the software current was virtually impossible and called for Microsoft to deliver software that was more secure when first installed.

Many IIS users began to consider switching to other Web server software. Gartner, Inc., a major IT consulting firm, recommended to its clients that they seriously consider alternatives to IIS for their critical Web server installations. Many industry observers and software engineers agree that Microsoft was a victim of its own success. It had created a very popular and complex piece of software. It is extremely difficult to ensure that no bugs exist in complex software products, and the popularity of the software made it an attractive target for crackers—one worm could bring down many of the servers operating on the Internet. These two factors, plus the likelihood that many IIS servers would not have all of the available security upgrades installed, combined to make it an irresistible target for a worm creator.

Microsoft has struggled to gain the confidence of large corporate IT departments. The company has worked hard in recent years to establish the reputation of its operating system software as reliable and trustworthy. The Code Red worm attack on its Web server software was a major setback in its reputation-building effort. You can review the **Microsoft Security Pages** through the link in the Online Companion to see how Microsoft is still trying to establish that its software is secure in the face of continuing cracker and virus-writer attacks that are both regular and frequent.

Digital Certificates

One way to control threats from active content is to use digital certificates. A **digital certificate** or digital ID is an attachment to an e-mail message or a program embedded in a

Web page that verifies that the sender or Web site is who or what it claims to be. In addition, the digital certificate contains a means to send an encrypted message—encoded so others cannot read it—to the entity that sent the original Web page or e-mail message. In the case of a downloaded program containing a digital certificate, the encrypted message identifies the software publisher (ensuring that the identity of the software publisher matches the certificate) and indicates whether the certificate has expired or is still valid. The digital certificate is a **signed** message or code. Signed code or messages serve the same function as a photo on a driver's license or passport. They provide proof that the holder is the person identified by the certificate. Just like a passport, a certificate does not imply anything about either the usefulness or quality of the downloaded program. The certificate only supplies a level of assurance that the software is genuine. The idea behind certificates is that if the user trusts the software developer, signed software can be trusted because, as proven by the certificate, it came from that trusted developer.

Digital certificates are used for many different types of online transactions, including electronic commerce, electronic mail, and electronic funds transfers. A digital ID verifies a Web site to a shopper and, optionally, identifies a shopper to a Web site. Web browsers or e-mail programs exchange digital certificates automatically and invisibly when requested to validate the identity of each party involved in a transaction.

Figure 10-8 displays the digital certificate owned by Amazon.com. Whenever a browser indicates that it has established secure communication with a Web site; that is, when a lock appears in the browser's status line, the user can double-click the lock (the exact procedure varies somewhat from browser to browser) to display the Web site's digital certificate.

A digital certificate for software is an assurance that the software was created by a specific company. The certificate does not attest to the quality of the software, just to the identity of the company that published it. Digital certificates are issued by a **certification authority (CA)**. A CA can issue digital certificates to organizations or individuals. A CA requires entities applying for digital certificates to supply appropriate proof of identity. Once the CA is satisfied, it issues a certificate. Then, the CA signs the certificate, and its stamp of approval is affixed in the form of a public encryption key, which "unlocks" the certificate for anyone who receives the certificate attached to the publisher's code.

Digital certificates cannot be forged easily. A digital certificate includes six main elements, including:

- Certificate owner's identifying information, such as name, organization, address, and so on
- Certificate owner's public key (you will learn more about public and private keys later in this chapter)
- Dates between which the certificate is valid
- Serial number of the certificate
- Name of the certificate issuer
- Digital signature of the certificate issuer

A **key** is simply a number—usually a long binary number—that is used with the encryption algorithm to "lock" the characters of the message being protected so that they are undecipherable without the key. Longer keys usually provide significantly better protection than shorter keys. In effect, the CA is guaranteeing that the individual or organization that presents the certificate is who or what it claims to be.

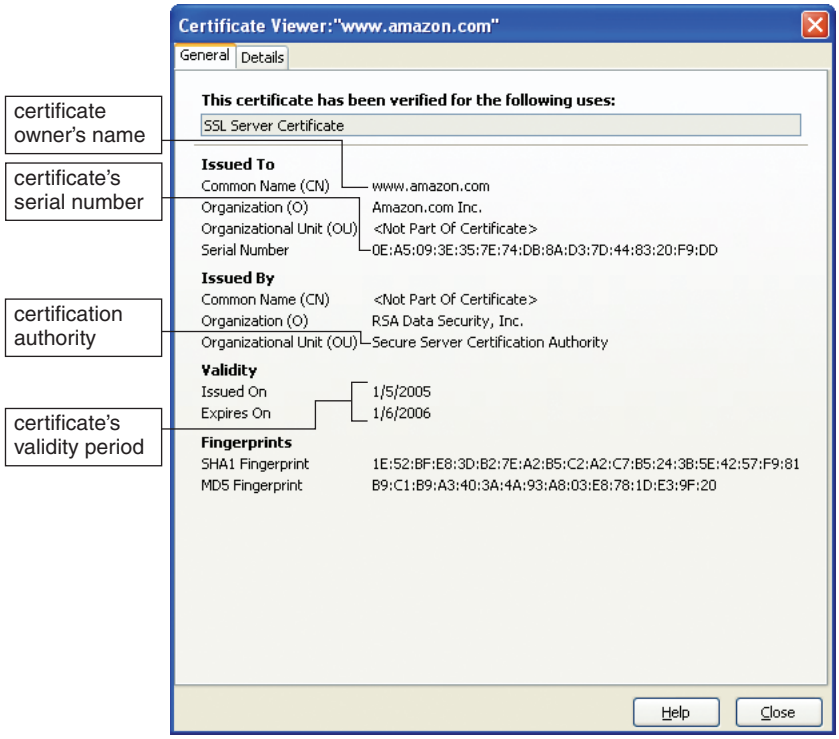

FIGURE 10-8 Amazon.com's digital certificate

Identification requirements vary from one CA to another. One CA might require a driver's license for individuals' certificates; others might require a notarized form or fingerprints. CAs usually publish identification requirements so that any Web user or site accepting certificates from each CA understands the stringency of that CA's validation procedures. There are only a small number of CAs because the certificates issued are only as trustworthy as the CA itself, and only a few companies have decided to build the reputation needed to be a successful seller of digital certificates. Two of the most commonly used CAs are **Thawte** and **VeriSign**, but other companies such as **Entrust** and **Equifax Secure** also offer CA services. The digital certificate for Amazon.com (information about this certificate appears in the dialog box shown in Figure 10-8) was issued by VeriSign. As you examine the certificates of various Web sites, you will notice that many of them indicate that the issuer is "RSA Data Security," which is the division of VeriSign that issues many of its digital certificates.

Certificates are classified as low, medium, or high assurance, based largely on the identification requirements imposed on certificate seekers. The fees charged by CAs vary with the level of assurance provided; higher levels of assurance are more expensive. For example, VeriSign provides certificate issuing and revocation services and offers several classes of certificates—from Class 1 through Class 4—that are differentiated by assurance level, which is the confidence level one can assume based on the process the CA uses to verify the owner's identity. Class 1 certificates are the lowest level and bind e-mail addresses

and associated public keys. Class 4 certificates apply to servers and the server organizations. Requirements for Class 4 certificates are significantly greater than those for Class 1. VeriSign's Class 4 certificate, for example, offers assurance of the individual's identity and that person's relationship to the specified company or organization.

Digital certificates expire after a period of time (often one year). This built-in limit provides protection for both users and businesses. Limited-duration certificates guarantee that businesses and individuals must submit their credentials for reevaluation periodically. The expiration date appears in the certificate itself and in the dialog boxes that browsers display when a Web page or applet that has a digital certificate is about to be opened. Certificates become invalid on their expiration dates or when they are intentionally revoked by the CA. If the CA determines that a Web site has begun delivering malicious code, it will refuse to issue new certificates to that site and revoke any existing certificates it might already have obtained.

Steganography

The term **steganography** describes the process of hiding information (a command, for example) within another piece of information. This information can be used for malicious purposes. Frequently, computer files contain redundant or insignificant information that can be replaced with other information. This other information resides in the background and is undetectable by anyone without the correct decoding software. Steganography provides a way of hiding an encrypted file within another file so that a casual observer cannot detect that there is anything of importance in the container file. In this two-step process, encrypting the file protects it from being read, and steganography makes it invisible.

Many security analysts believe that the terrorist organization Al Qaeda used steganography to hide attack orders and other messages in images that its confederates posted on Web sites. Messages hidden using steganography are extremely difficult to detect. This fact, combined with the fact that there are millions of images on the Web, makes the use of steganography by global terrorist organizations a deep concern of governments and security professionals. The Online Companion includes a link to a site with more information about **Steganography and Digital Watermarking**.

Physical Security for Clients

In the past, physical security was a major concern for large computers that ran important business functions such as payroll or billing; however, as networks (including intranets and the Internet) have made it possible to control important business functions from client computers, concerns about physical security for client computers have become greater. Many of the physical security measures used today are the same as those used in the early days of computing; however, some interesting new technologies have been implemented as well.

Devices that read fingerprints are now available for personal computers. These devices, which cost less than $200, provide a much stronger protection than traditional password approaches. In addition to fingerprint readers, companies can use other biometric security devices that are more accurate and, of course, cost more. A **biometric security device** is one that uses an element of a person's biological makeup to perform the

identification. These devices include writing pads that detect the form and pressure of a person writing a signature, eye scanners that read the pattern of blood vessels in a person's retina or the color levels in a person's iris, and scanners that read the palm of a person's hand (rather than just one fingerprint) or that read the pattern of veins on the back of a person's hand.

COMMUNICATION CHANNEL SECURITY

The Internet serves as the electronic connection between buyers (in most cases, clients) and sellers (in most cases, servers). The most important thing to remember as you learn about communication channel security is that the Internet was not designed to be secure. Although the Internet has its roots in a military network, that network was not designed to include any significant security features. It was designed to provide redundancy in case one or more communications lines were cut. In other words, the goal of the Internet's packet-switching design was to provide multiple alternative paths on which critical military information could travel. The military always sends sensitive information in an encrypted form so that the content of messages traveling over any network—even if intercepted—remain secret. The security of messages traversing the military predecessors to the Internet was provided by software that operated independently of the network to encrypt messages. As the Internet developed, it did so without any significant security features that became a part of the network itself.

Today, the Internet remains largely unchanged from its original, insecure state. Message packets on the Internet travel an unplanned path from a source node to a destination node. A packet passes through a number of intermediate computers on the network before reaching its final destination. The path can vary each time a packet is sent between the same source and destination points. Because users cannot control the path and do not know where their packets have been, it is possible that an intermediary can read the packets, alter them, or even delete them. That is, any message traveling on the Internet is subject to secrecy, integrity, and necessity threats. This section describes these problems in more detail and outlines several solutions for those problems.

Secrecy Threats

Secrecy is the security threat that is most frequently mentioned in articles and the popular media. Closely linked to secrecy is privacy, which also receives a great deal of attention. Secrecy and privacy, though similar, are different issues. Secrecy is the prevention of unauthorized information disclosure. **Privacy** is the protection of individual rights to nondisclosure. The **Privacy Council**, which helps businesses implement smart privacy and data practices, created an extensive Web site surrounding privacy—covering both business and legal issues. Secrecy is a technical issue requiring sophisticated physical and logical mechanisms, whereas privacy protection is a legal matter. A classic example of the difference between secrecy and privacy is e-mail.

A company might protect its e-mail messages against secrecy violations by using encryption (you will learn more about encryption later in this chapter). In encryption, a message is encoded into an unintelligible form that only the proper recipient can convert back into the original message. Secrecy countermeasures protect outgoing messages. E-mail privacy issues address whether company supervisors should be permitted to read

employees' messages randomly. Disputes in this area center around who owns the e-mail messages: the company, or the employees who sent them. The focus in this section is on secrecy, preventing unauthorized persons from reading information they should not be reading.

One significant threat to electronic commerce is theft of sensitive or personal information, including credit card numbers, names, addresses, and personal preferences. This kind of theft can occur any time anyone submits information over the Internet because it is easy for an ill-intentioned person to record information packets (a secrecy violation) from the Internet for later examination. The same problems can occur in e-mail transmissions. Software applications called **sniffer programs** provide the means to record information that passes through a computer or router that is handling Internet traffic. Using a sniffer program is analogous to tapping a telephone line and recording a conversation. Sniffer programs can read e-mail messages and unencrypted Web client-server message traffic such as user logins, passwords, and credit card numbers.

Periodically, security experts find electronic holes, called **backdoors**, in electronic commerce software. These can be left open accidentally by the software developer, or they can be left open intentionally. Either way, content is exposed to secrecy threats. A backdoor allows anyone with knowledge of the existence of the backdoor to cause damage by observing transactions, deleting data, or stealing data. In 2000, the Cart32 shopping cart software made by McMurtrey/Whitaker & Associates was found to have a backdoor through which credit card numbers could be obtained by anyone with a backdoor password. The company quickly supplied a patch to eliminate the backdoor. Although the backdoor resulted from a programming error and not from intentional efforts, the consequences were serious for merchants that used the software—their customers' credit card numbers were available to hackers around the world.

Credit card number theft is an obvious problem, but proprietary corporate product information or prerelease data sheets mailed to corporate branches can be intercepted and passed along easily, too. Confidential information can be considerably more valuable than information about credit cards, which usually have spending limits. Stolen corporate information can be worth millions of dollars.

Here is an example of how an online eavesdropper might obtain confidential information. Suppose a user logs on to a Web site that contains a form with text boxes for name, address, and e-mail address. When the user fills out those text boxes and clicks the Submit button, the information is sent to the Web server for processing. Some Web servers obtain and track that data by collecting the text box responses and placing them at the end of the server's URL (which appears in the address box of the user's Web browser). This long URL (with the text box responses appended) is included in all HTTP request and response messages that travel between the user's browser and the server.

So far, no violations have occurred. Suppose, however, that the user decides not to wait for a response from the server. Instead, the user visits another Web site. The server at this second Web site might be set up to collect Web demographics. If it is, it logs the URL from which the user just came by capturing it from the HTTP request message that the browser sends. Web sites use this URL logging technique for the completely legitimate purpose of identifying sources of customer traffic. However, any employee at the second site who has access to the server log can read the part of the URL that includes the information entered into those text boxes on the first site, thus obtaining that user's confidential information.

Web users continually reveal information about themselves when they use the Web. This information includes IP addresses and the type of browser being used. Such data exposure is a secrecy breach. Several Web sites offer an anonymous browser service that hides personal information from sites visited. One of these sites, **Anonymizer**, provides a measure of secrecy to Web surfers who use the site as a portal (the beginning site from which they visit other sites). Anonymizer places its address on the front end of any URLs that the user visits. This shield reveals only the Anonymizer Web site URL to other Web sites that the user visits. This can make anonymous Web surfing possible, but tedious, because each URL that the user wants to visit must be typed in the text box on the Anonymizer home page. To make the process easier, Anonymizer and other companies provide browser plug-in software that users can download and install for an annual subscription fee. Figure 10-9 shows Anonymizer's home page.

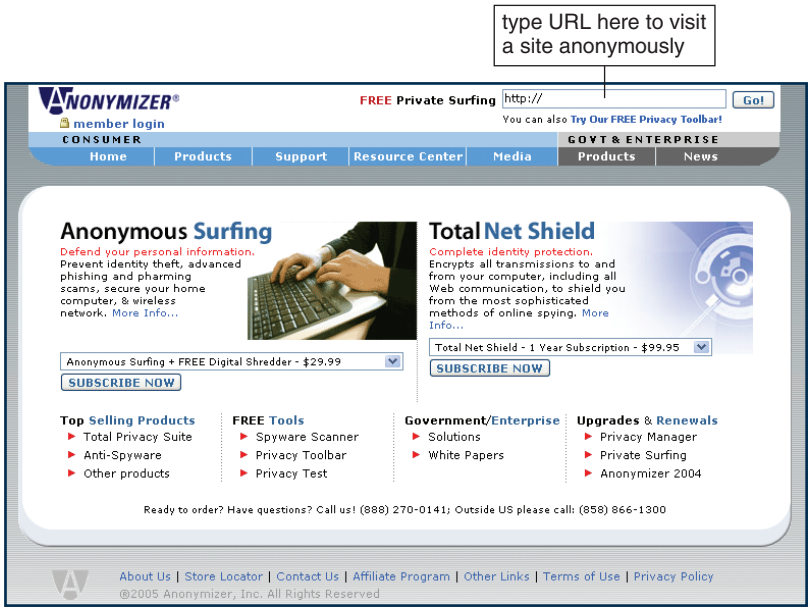

FIGURE 10-9 Anonymizer home page

Integrity Threats

An integrity threat, also known as **active wiretapping**, exists when an unauthorized party can alter a message stream of information. Unprotected banking transactions, such as deposit amounts transmitted over the Internet, are subject to integrity violations. Of course, an integrity violation implies a secrecy violation because an intruder who alters information can read and interpret that information. Unlike secrecy threats, where a viewer simply sees information he or she should not, integrity threats can cause a change in the actions a person or corporation takes because a mission-critical transmission has been altered.

Cybervandalism is an example of an integrity violation. **Cybervandalism** is the electronic defacing of an existing Web site's page. The electronic equivalent of destroying property or placing graffiti on objects, cybervandalism occurs whenever someone replaces a Web site's regular content with his or her own content. Recently, several cases of Web page defacing involved vandals replacing business content with pornographic material and other offensive content.

Masquerading or **spoofing**—pretending to be someone you are not, or representing a Web site as an original when it is a fake—is one means of disrupting Web sites. **Domain name servers (DNSs)** are the computers on the Internet that maintain directories that link domain names to IP addresses. Perpetrators can use a security hole in the software that runs on some of these computers to substitute the addresses of their Web sites in place of the real ones to spoof Web site visitors.

For example, a hacker could create a fictitious Web site masquerading as www.widgets.com by exploiting a DNS security hole that substitutes his or her fake IP address for Widgets.com's real IP address. All subsequent visits to Widgets.com would be redirected to the fictitious site. There, the hacker could alter any orders to change the number of widgets ordered and redirect shipment of those products to another address. The integrity attack consists of altering an order and passing it to the real company's Web server. The Web server is unaware of the integrity attack and simply verifies the consumer's credit card number and passes on the order for fulfillment. Major electronic commerce sites that have been the victims of masquerading attacks in recent years include Amazon.com, AOL, eBay, and PayPal. Some of these schemes combine spam with spoofing. The perpetrator sends millions of spam e-mails that appear to be from a respectable company. The e-mails contain a link to a Web page that is designed to look exactly like the company's site. The victim is encouraged to enter username, password, and sometimes even credit card information. These exploits, which capture confidential customer information, are called **phishing expeditions**. The most common victims of phishing expeditions are users of online banking and payment system (such as PayPal) Web sites. You will learn more about the phishing problem and the measures banks and other companies are taking to combat it in Chapter 11.

Necessity Threats

The purpose of a **necessity threat**, also known by other names such as a delay, denial, or denial-of-service (DoS) threat, is to disrupt normal computer processing, or deny processing entirely. A computer that has experienced a necessity threat slows processing to an intolerably slow speed. For example, if the processing speed of a single ATM transaction slows from one or two seconds to 30 seconds, users will abandon ATMs entirely. Similarly, slowing any Internet service drives customers to competitors' Web or commerce sites—possibly discouraging them from ever returning to the original commerce site. In other words, slower processing can render a service unusable or unattractive. For example, an online newspaper that reports three-day-old news is worth very little.

DoS attacks remove information altogether, or delete information from a transmission or file. One documented denial attack caused selected PCs that have Quicken (an accounting program) installed to divert money to the perpetrator's bank account. The denial attack denied money from its rightful owners. In another famous DoS attack against high-profile electronic commerce sites such as Amazon.com and Yahoo!, the attackers

used zombie computers to send a flood of data packets to the sites. This overwhelmed the sites' servers and choked off legitimate customers' access. Prior to the attack, perpetrators located vulnerable computers and loaded them with the software that attacked the commerce sites. The Internet Worm attack of 1998, which disabled thousands of computer systems that were connected to the Internet, was the first recorded example of a DoS attack.

Threats to the Physical Security of Internet Communications Channels

The Internet was designed from its inception to withstand attacks on its physical communication links. Recall from Chapter 2 that the main purpose of the U.S. government research project that led to the development of the Internet was to provide an attack-resistant technology for coordinating military operations. Thus, the Internet's packet-based network design precludes it from being shut down by an attack on a single communications link on that network.

However, an individual user's Internet service can be interrupted by destruction of that user's link to the Internet. Few individual users have multiple connections to an ISP. However, larger companies and organizations (and ISPs themselves) often do have more than one link to the main backbone of the Internet. Typically, each link is purchased from a different network access provider. If one link becomes overloaded or unavailable, the service provider can switch traffic to another network access provider's link to keep the company, organization, or ISP (and its customers) connected to the Internet.

Threats to Wireless Networks

As you learned in Chapter 2, networks can use wireless access points (WAPs) to provide network connections to computers and other mobile devices within a range of several hundred feet. If not protected, a wireless network allows anyone within that range to log in and have access to any resources connected to that network. Such resources might include any data stored on any computer connected to the network, networked printers, messages sent on the network, and, if the network is connected to the Internet, free access to the Internet. The security of the connection depends on the Wireless Encryption Protocol (WEP), which is a set of rules for encrypting transmissions from the wireless devices to the WAPs.

Companies that have large wireless networks are usually careful to turn on WEP in devices, but smaller companies and individuals who have installed wireless networks in their homes often do not turn on the WEP security feature. Many WAPs are shipped to buyers with a default login and password already set. Companies that install these WAPs sometimes fail to change that login and password. This has given rise to a new avenue of entry into networks.

In some cities that have large concentrations of wireless networks, attackers drive around in cars using their wireless-equipped laptop computers to search for accessible networks. These attackers are called **wardrivers**. When wardrivers find an open network (or a WAP that has a common default login and password), they sometimes place a chalk mark on the building so that other attackers will know that an easily entered wireless network is nearby. This practice is called **warchalking**. Some warchalkers have even created Web sites that include maps of wireless access locations in major cities around the world. Companies can avoid becoming targets by simply turning on WEP in their access

points and changing the logins and passwords to something other than the manufacturers' default settings.

In 2002, Best Buy was using wireless point-of-sale (POS) terminals in some of its 1900 stores. The wireless POS terminals could be moved easily from one area of the store to another, and they helped Best Buy handle large customer flows better than it could using only fixed POS terminals. Unfortunately, Best Buy failed to enable WEP on these terminals. A customer who had just purchased a wireless card for his laptop decided to launch a sniffer utility program on the laptop in his car in the parking lot. The customer was able to intercept data from the POS terminals, including transaction details and what he said looked like credit card numbers. Best Buy stopped using the wireless POS terminals when the story appeared on several Web sites and newswire services.

Encryption Solutions

Encryption is the coding of information by using a mathematically based program and a secret key to produce a string of characters that is unintelligible. The science that studies encryption is called **cryptography**, which comes from a combination of the two Greek words *krypto* and *grapho*, which mean "secret" and "writing," respectively. That is, cryptography is the science of creating messages that only the sender and receiver can read.

Cryptography is different from steganography, which makes text undetectable to the naked eye. Cryptography does not hide text; it converts it to other text that is visible, but does not appear to have any meaning. What an unauthorized reader sees is a string of random text characters, numbers, and punctuation.

Encryption Algorithms

The program that transforms normal text, called **plain text**, into **cipher text** (the unintelligible string of characters) is called an **encryption program**. The logic behind an encryption program that includes the mathematics used to do the transformation from plain text to cipher text is called an **encryption algorithm**. There are a number of different encryption algorithms in use today. Some have been developed by the U.S. government and others have been developed by IBM and other commercial enterprises. You can learn more about the development of encryption algorithms, including an evaluation of currently available algorithms, by consulting a Web security textbook (see, for example, the Mackey reference in the For Further Study and Research section at the end of this chapter).

Messages are encrypted just before they are sent over a network or the Internet. Upon arrival, each message is decoded, or **decrypted**, using a **decryption program**—a type of encryption-reversing procedure. Encryption algorithms are considered so vitally important to preserving security within the United States that the National Security Agency has control over their dissemination. Some encryption algorithms are considered so important that the U.S. government has banned publication of details about them. Currently, it is illegal for U.S. companies to export some of these encryption algorithms. Web pages containing software whose distribution is restricted include warnings about U.S. export laws. The Freedom Forum Online contains a number of articles on lawsuits and legislation surrounding encryption export laws. Critics consider publication restrictions a freedom of speech issue. If you are interested in reading more about the latest arguments in the ongoing debates over freedom of speech and export law, search the **Freedom Forum** using

the keyword "encryption" as the search term.

One property of encryption algorithms is that someone can know the details of the algorithm and still not be able to decipher the encrypted message without knowing the key that the algorithm used to encrypt the message. The resistance of an encrypted message to attack attempts depends on the size (in bits) of the key used in the encryption procedure. A 40-bit key is currently considered to provide a minimal level of security. Longer keys, such as 128-bit keys, provide much more secure encryption. A sufficiently long key can help make the security unbreakable.

The type of key and associated encryption program used to lock a message, or otherwise manipulate it, subdivides encryption into three functions:

- Hash coding
- Asymmetric encryption
- Symmetric encryption

Hash Coding

Hash coding is a process that uses a **hash algorithm** to calculate a number, called a **hash value**, from a message of any length. It is a fingerprint for the message because it is almost certain to be unique for each message. Good hash algorithms are designed so that the probability of two different messages resulting in the same hash value, which would create a **collision**, is extremely small. Hash coding is a particularly convenient way to tell whether a message has been altered in transit because its original hash value and the hash value computed by the receiver will not match after a message is altered.

Asymmetric Encryption

Asymmetric encryption, or **public-key encryption**, encodes messages by using two mathematically related numeric keys. In 1977, Ronald Rivest, Adi Shamir, and Leonard Adleman invented the RSA Public Key Cryptosystem while they were professors at MIT. Their invention revolutionized the way sensitive information is exchanged. In their system, one key of the pair, called a **public key**, is freely distributed to the public at large—to anyone interested in communicating securely with the holder of both keys. The public key is used to encrypt messages using one of several different encryption algorithms. The second key—called a **private key**—belongs to the key owner, who keeps the key secret. The owner uses the private key to decrypt all messages received.

Here is an overview of how an asymmetric encryption system works: If Herb wants to send a message to Allison, he obtains Allison's public key from any of several well-known public places. Then, he encrypts his message to Allison using her public key. Once the message is encrypted, only Allison can read the message by decrypting it with her private key. Because the keys are unique, only one secret key can open the message encrypted with a corresponding public key, and vice versa. Reversing the process, Allison can send a private message to Herb using Herb's public key to encrypt the message. When he receives Allison's message, Herb uses his private key to decrypt the message and then read it. If they are sending e-mail to one another, the message is secret only while in transit. Once a message is downloaded from the mail server and decoded, it is stored in plain text on the recipient's machine for all to view.

One of the most popular technologies used to implement public-key encryption today is called **Pretty Good Privacy (PGP)**. PGP was invented in 1991 by Phil Zimmerman, who charged businesses for use of PGP, but allowed individuals to use PGP at no cost. PGP is a set of software tools that can use several different encryption algorithms to perform public-key encryption. The PGP business was purchased by Network Associates in 1997 and sold back to the product's developers, who formed PGP Corporation in 2002. Today, individuals can download free versions of PGP for personal use from the **PGP Corporation** site and from the **PGP International** site. Individuals can use PGP to encrypt their e-mail messages to protect them from being read if they are intercepted on the Internet. The PGP Corporation site sells licenses to businesses that want to use the technology to protect business communication activities.

Symmetric Encryption

Symmetric encryption, also known as **private-key encryption**, encodes a message with one of several available algorithms that use a single numeric key, such as 456839420783, to encode and decode data. Because the same key is used, both the message sender and the message receiver must know the key. Encoding and decoding messages using symmetric encryption is very fast and efficient. However, the key must be guarded. If the key is made public, then all messages sent previously using that key are vulnerable, and both the sender and receiver must use new keys for future communication.

It can be difficult to distribute new keys to authorized parties while maintaining security and control over the keys. The catch is that to transmit *anything* privately, it must be encrypted. This includes the new, secret key. Another significant problem with private keys is that they do not scale well in large environments such as the Internet. Each pair of users on the Internet who wants to share information privately must have their own private key. That results in a huge number of key-pair combinations, similar to a telephone system of private lines without switching stations. Enabling 12 people to have a private key pair between all pairs (or private telephone lines between each pair) would require 66 private keys. In general, n individual Internet clients require $(n(n\text{-}1))/2$ private key pairs.

In secure environments such as the defense sector, using private-key encryption is simpler, and it is the prevalent method to encode sensitive data. Distribution of classified information and encryption keys is straightforward in the defense sector. It requires guards (two-person control) and secret transportation plans. The **Data Encryption Standard (DES)** is a set of encryption algorithms adopted by the U.S. government for encrypting sensitive or commercial information. It is the most widely used private-key encryption system. However, the DES private-key size is increased periodically because individuals are using increasingly fast computers to break messages encoded with shorter keys. In 1999, for example, the Electronic Frontier Foundation's Deep Crack key breaker used 100,000 PCs on the Internet to break a DES-encrypted test message in under 23 hours (see the **EFF DES Cracker Project** for more information).

Today, the U.S. government uses a stronger version of the Data Encryption Standard, called **Triple Data Encryption Standard (Triple DES** or **3DES)**. Triple DES offers good protection because it cannot be cracked even with today's supercomputers. Experts expect that it will continue to be extremely difficult to crack for the next several years. However, the U.S. government's **National Institute of Standards and Technology** (NIST) has developed a new encryption standard designed to keep government information secure.

The new standard is called the **Advanced Encryption Standard (AES)**. In February 2001, the NIST announced that the four-year development process had been successful and that two cryptography researchers from Belgium had created the algorithm chosen for AES. The algorithm's name is Rijndael (pronounced "rain doll"); you can learn more about the development process and the algorithm at the NIST's **AES Algorithm (Rijndael)** Web site.

Comparing Asymmetric and Symmetric Encryption Systems

Public-key (asymmetric) systems provide several advantages over private-key (symmetric) encryption methods. First, the combination of keys required to provide private messages between enormous numbers of people is small. If *n* people want to share secret information with one another, then only *n* unique public-key pairs are required—far fewer than an equivalent private-key system. Second, key distribution is not a problem. Each person's public key can be posted anywhere and does not require any special handling to distribute. Third, public-key systems make implementation of digital signatures possible. This means that an electronic document can be signed and sent to any recipient with nonrepudiation. That is, with public-key techniques, it is not possible for anyone other than the signer to produce the signature electronically; in addition, the signer cannot later deny signing the electronic document.

Public-key systems have disadvantages. One disadvantage is that public-key encryption and decryption are significantly slower than private-key systems. This extra time can add up quickly as individuals and organizations conduct commerce on the Internet. Public-key systems do not replace private-key systems, but serve as a complement to them. Public-key systems are used to transmit private keys to Internet participants so that additional, more efficient communication can occur in a secure Internet session. Figure 10-10 shows a graphical representation of the hashing, private-key, and public-key encryption methods: Figure 10-10a shows hash coding; Figure 10-10b depicts private-key encryption; and Figure 10-10c illustrates public-key encryption.

Several encryption algorithms exist that can be used with secure Web servers. The U.S. government approves the use of several of these inside the United States. Electronic commerce Web servers can accommodate most of these algorithms because they must be able to communicate with a wide variety of Web browsers.

The **Secure Sockets Layer (SSL)** system developed by Netscape Communications and the Secure Hypertext Transfer Protocol (S-HTTP) developed by CommerceNet are two protocols that provide secure information transfer through the Internet. SSL and S-HTTP allow both the client and server computers to manage encryption and decryption activities between each other during a secure Web session.

SSL and S-HTTP have different goals. SSL secures connections between two computers, and S-HTTP sends *individual* messages securely. Encryption of outgoing messages and decryption of incoming messages happens automatically and transparently with both SSL and S-HTTP.

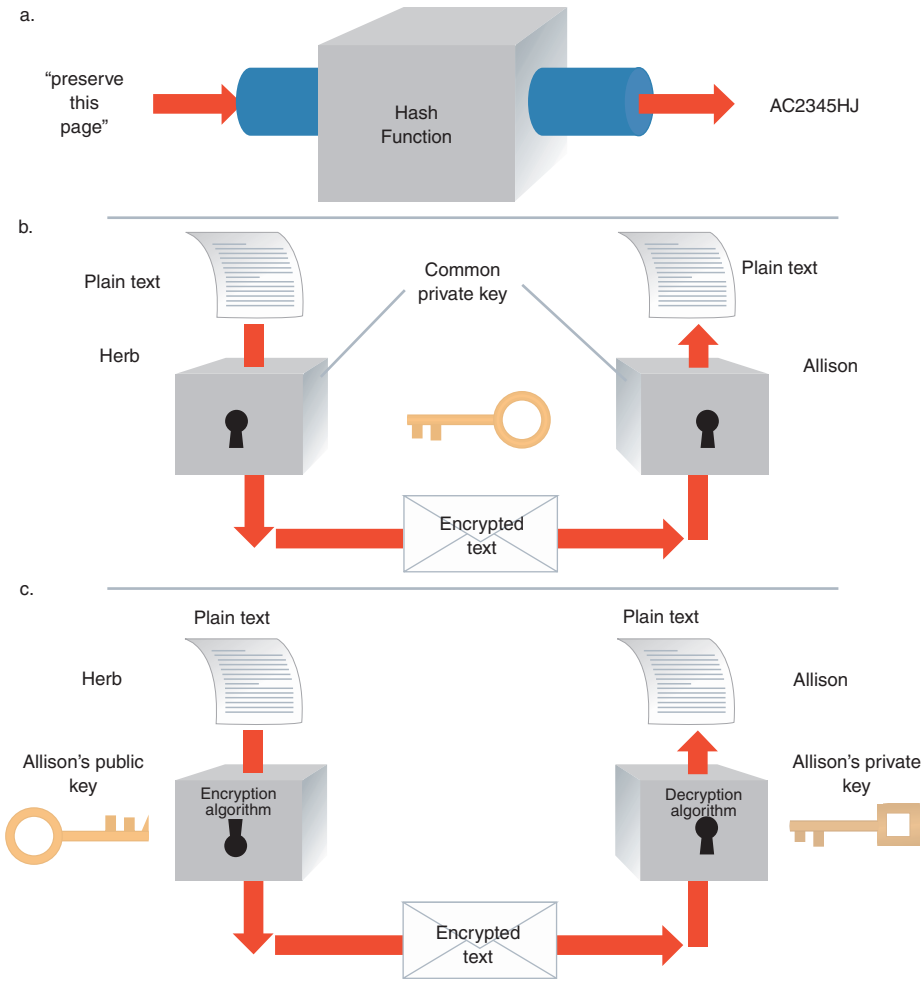

FIGURE 10-10 (a) hash coding, (b) private-key, and (c) public-key encryption

Secure Sockets Layer (SSL) Protocol

SSL provides a security "handshake" in which the client and server computers exchange a brief burst of messages. In those messages, the level of security to be used for exchange of digital certificates and other tasks is agreed upon. Each computer identifies the other. After identification, SSL encrypts and decrypts information flowing between the two computers. This means that information in both the HTTP request and any HTTP response is encrypted. Encrypted information includes the URL the client is requesting, any forms containing information the user has completed (which might include a credit card number), and HTTP access authorization data, such as usernames and passwords. In short, *all* communication between SSL-enabled clients and servers is encoded. When SSL encodes everything flowing between the client and server, an eavesdropper receives only unintelligible information.

SSL can secure many different types of communication between computers in addition to HTTP. For example, SSL can secure FTP sessions, enabling private downloading and uploading of sensitive documents, spreadsheets, and other electronic data. SSL can secure Telnet sessions in which remote computer users can log on to corporate host machines and send their passwords and usernames. The protocol that implements SSL is HTTPS. By preceding the URL with the protocol name HTTPS, the client is signifying that it would like to establish a secure connection with the remote server.

Secure Sockets Layer allows the length of the private session key generated by every encrypted transaction to be set at a variety of bit lengths (such as 40-bit, 56-bit, 128-bit, and 168-bit). A **session key** is a key used by an encryption algorithm to create cipher text from plain text during a single secure session. The longer the key, the more resistant the encryption is to attack. A Web browser that has entered into an SSL session indicates that it is in an encrypted session (most browsers use an icon in the browser status bar). Once the session is ended, the session key is discarded permanently and not reused for subsequent secure sessions.

Here is how SSL works with an exchange between a client and an electronic commerce site: Remember that SSL has to authenticate the commerce site and encrypt any transmissions between the two computers. When a client browser sends a request message to a server's secure Web site, the server sends a hello request to the browser (client). The browser responds with a client hello. The exchange of these greetings, or the handshake, allows the two computers to determine the compression and encryption standards that they both support.

Next, the browser asks the server for a digital certificate—proof of identity. In response, the server sends to the browser a certificate signed by a recognized certification authority. The browser checks the serial number and certificate fingerprint on the server certificate against the public key of the CA stored within the browser. Once the CA's public key is verified, the endorsement is verified. That action authenticates the Web server.

Both the client and server agree that their exchanges should be kept secure because they involve transmitting credit card numbers, invoice numbers, and verification codes over the Internet. To implement secrecy, SSL uses public-key (asymmetric) encryption and private-key (symmetric) encryption. Although public-key encryption is handy, it is slow compared to private-key encryption. That is why SSL uses private-key encryption for nearly all its secure communications. Because it uses private-key encryption, SSL must have a way to get the key to both the client and server without exposing it to an eavesdropper. SSL accomplishes this by having the browser generate a private key for both to share. Then the browser encrypts the private key it has generated using the server's public key. The server's public key is stored in the digital certificate that the server sent to the browser during the authentication step. Once the key is encrypted, the browser sends it to the server. The server, in turn, decrypts the message with its private key and exposes the shared private key.

From this point on, public-key encryption is no longer used. Instead, only private-key encryption is used. All messages sent between the client and the server are encrypted with the shared private key, also known as the session key. When the session ends, the session key is discarded. A new connection between a client and a secure server starts the entire process all over again, beginning with the handshake between the client browser and the server. The client and server can agree to use 40-bit encryption or 128-bit encryption.

The client and server also agree on which specific encryption algorithm to use. Figure 10-11 illustrates the SSL handshake that occurs before a client and server exchange private-key encoded business information for the remainder of the secure session.

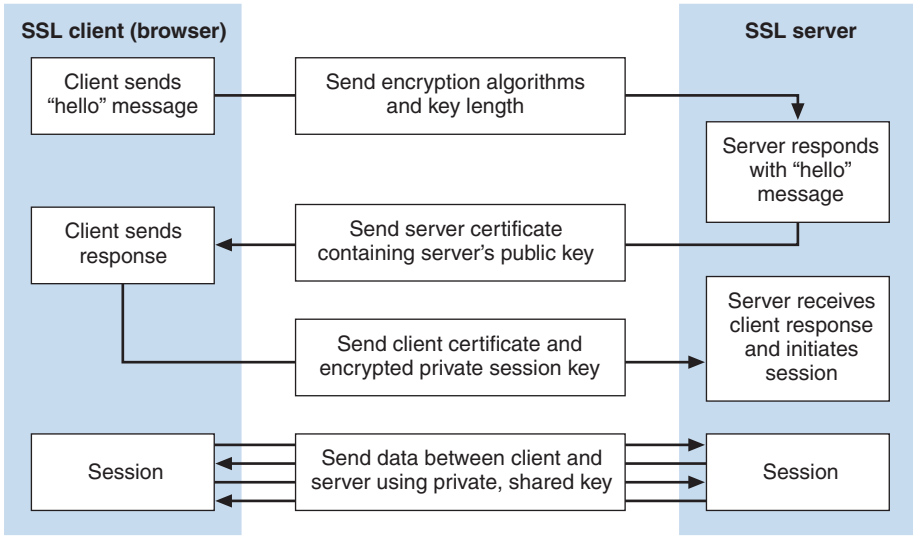

FIGURE 10-11 Establishing an SSL session

Secure HTTP (S-HTTP)

Secure HTTP (S-HTTP) is an extension to HTTP that provides a number of security features, including client and server authentication, spontaneous encryption, and request/response nonrepudiation. The protocol was developed by **CommerceNet**, a consortium of organizations interested in promoting electronic commerce. S-HTTP provides symmetric encryption for maintaining secret communications and public-key encryption to establish client/server authentication. Either the client or the server can use S-HTTP techniques separately. That is, a client browser may require security through the use of a private (symmetric) key, whereas the server may require client authentication by using public-key techniques.

The details of S-HTTP security are conducted during the initial negotiation session between the client and server. Either the client or the server can specify that a particular security feature be required, optional, or refused. When one party stipulates that a particular security feature be required, the client or server continues the connection only if the other party (client or server) agrees to enforce the specified security. Otherwise, no secure connection is established. Suppose the client browser specifies that encryption is required to render all communications secret. In such a situation, the transactions of a high-fashion clothing designer purchasing silk from a Far East textile house will remain confidential. Eavesdropping competitors cannot learn which fabrics are featured next season. On the other hand, the textile mill may insist that integrity be enforced so that

quantities and prices quoted to the purchaser remain intact. In addition, the textile mill may want assurances that the purchaser is who he or she claims to be, not an imposter. A form of nonrepudiation, this security property provides positive confirmation of an offer by a client and makes it impossible for the client to deny ever having made the offer.

S-HTTP differs from SSL in the way it establishes a secure session. SSL carries out a client/server handshake exchange to set up a secure communication, but S-HTTP sets up security details with special packet headers that are exchanged in S-HTTP. The headers define the type of security techniques, including the use of private-key encryption, server authentication, client authentication, and message integrity. Header exchanges also stipulate which specific algorithms each side supports, whether the client or the server (or both) supports the algorithm, and whether the security technique (for example, secrecy) is required, optional, or refused. Once the client and server agree to security implementations enforced between them, all subsequent messages between them during that session are wrapped in a secure container, sometimes called an envelope. A **secure envelope** encapsulates a message and provides secrecy, integrity, and client/server authentication. In other words, it is a complete package. With it, all messages traveling on the network or Internet are encrypted so that they cannot be read. Messages cannot be altered undetectably because integrity mechanisms provide a detection code that signals a message has been altered. Clients and servers are authenticated with digital certificates issued by a recognized certification authority. The secure envelope includes all of these security features. S-HTTP is no longer used by many Web sites. SSL has become a more generally accepted standard for establishing secure communication links between Web clients and Web servers.

You have learned how encryption provides message secrecy and confidentiality, and you have learned how digital certificates serve to authenticate a server to a client, and vice versa. However, you have not learned how to implement message integrity. The methods that allow you to ensure that an interloper does not change a message in transit appear in the next section.

Ensuring Transaction Integrity with Hash Functions

Electronic commerce ultimately involves a client browser sending payment information, order information, and payment instructions to the Web server and that server responding with a confirmation of the order details. If an Internet interloper alters any of the order information in transit, harmful consequences can result. For instance, the perpetrator could alter the shipment address so that he or she receives the merchandise instead of the original customer. This is an example of an **integrity violation**, which occurs whenever a message is altered while in transit between the sender and receiver.

Although it is difficult and expensive to *prevent* a perpetrator from altering a message, there are security techniques that allow the receiver to *detect* when a message has been altered. When the receiver—a Web server, for example—receives a damaged message, the receiver simply asks the sender to retransmit the message. Apart from being annoying, a damaged message harms no one as long as both parties are aware of the alteration. Harm occurs when unauthorized message changes go undetected by the message's sender and receiver.

A combination of techniques creates messages that are both tamperproof and authenticated. Additionally, those techniques provide the property of nonrepudiation —making it impossible for message creators to claim that the message was not theirs or that

they did not send it. To eliminate fraud and abuse caused by messages being altered, two separate algorithms are applied to a message. First, a hash algorithm is applied to the message. Hash algorithms are **one-way functions**, meaning that there is no way to transform the hash value back to the original message. This approach is acceptable because a hash value is compared only with another hash value to see if there is a match—the original, prehash values are never compared with one another.

All encryption programs convert text into a **message digest**, which is a small integer number that summarizes the encrypted information. A hash algorithm uses no secret key; the message digest it produces cannot be inverted to produce the original information; the algorithm and information about how it works are publicly available; and finally, hash collisions are nearly impossible. Once the hash function computes a message's hash value, that value is appended to the message. Suppose the message is a purchase order containing the customer's address and payment information. When the merchant receives the purchase order and attached message digest, he or she calculates a message digest value for the message (exclusive of the original attached message digest). If the message digest value that the merchant calculates matches the message digest attached to the message, the merchant then knows the message is unaltered—that is, no interloper altered the amount or the shipping address information. Had someone altered the information, then the merchant's software would compute a message digest value different from the message digest that the client calculated and sent along with the purchase order.

Ensuring Transaction Integrity with Digital Signatures

Hash functions are not a complete solution. Because the hash algorithm is public and (by design) widely known, anyone could intercept a purchase order, alter the shipping address and quantity ordered, re-create the message digest, and send the message and new message digest on to the merchant. Upon receipt, the merchant would calculate the message digest value and confirm that the two message digest values match. The merchant is fooled into concluding that the message is unadulterated and genuine. To prevent this type of fraud, the sender encrypts the message digest using his or her private key.

An encrypted message digest (message hash value) is called a **digital signature**. A purchase order accompanied by a digital signature provides the merchant with positive identification of the sender and assures the merchant that the message was not altered. Because the message digest is encrypted using a public key, only the owner of the public/private key pair could have encrypted the message digest. Thus, when the merchant decrypts the message with the user's public key and subsequently calculates a matching message digest value, the result is proof that the sender is authentic. Furthermore, matching hash values prove that only the sender could have authored the message (nonrepudiation) because only his or her private key would yield an encrypted message that could be decrypted successfully by an associated public key. This solves the spoofing problem.

If necessary, both parties can agree to provide transaction secrecy in addition to the integrity, nonrepudiation, and authentication that the digital signature provides. Simply encrypting the entire string—digital signature and message—guarantees message secrecy. Used together, public-key encryption, message digests, and digital signatures provide a high level of security for Internet transactions. Figure 10-12 illustrates how a digital signature and a signed message are created and sent.

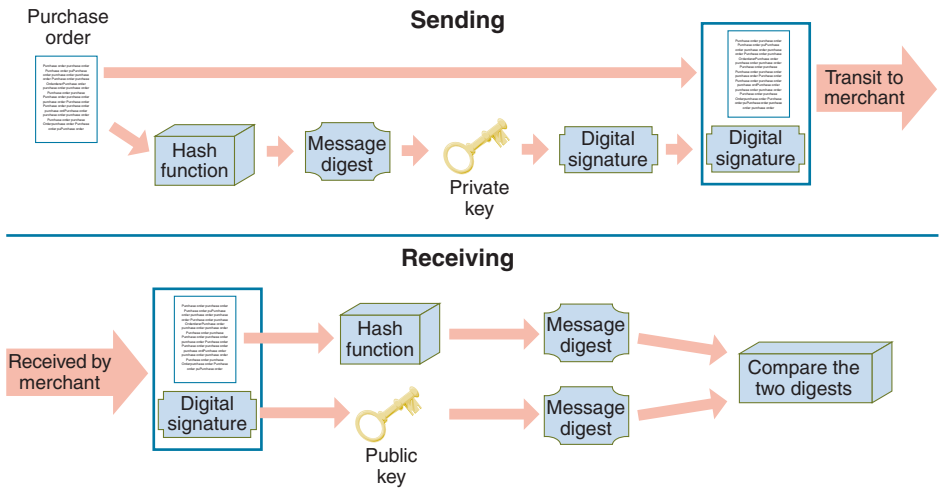

FIGURE 10-12 Sending and receiving a digitally signed message

In 2000, U.S. President Bill Clinton signed a bill that gave digital signatures the same legal status as traditional signatures. Clinton first signed the paper version of the new digital signature legislation with a pen. Then, he signed the electronic version of the bill with a smart card (you will learn about smart cards in Chapter 11) containing his digital signature. After doing so, the name "Bill Clinton" appeared on the screen under the text of the new law entitled *Electronic Signatures in Global and National Commerce Act*. People can now electronically sign all sorts of legal documents, such as online car lease agreements, loan papers, and purchase orders.

The European Union followed closely on the heels of the U.S. legislation and required all of its member countries to enact digital signature laws by mid-2001. Most Canadian provinces had also enacted digital signature legislation by the end of 2001. Other countries have passed or are working toward passing laws that enable the use of digital signatures.

Guaranteeing Transaction Delivery

As you learned earlier in this chapter, denial or delay-of-service attacks remove or absorb resources. Neither encryption nor a digital signature protects information packets from theft or slowdown. However, the Transmission Control Protocol (TCP) half of the TCP/IP pair is responsible for end-to-end control of packets. When it reassembles packets at the destination in the correct order, it handles all the details when packets do not appear. Among TCP's duties are to request that the client computer resend data when packets seem to be missing. That is, no special computer security protocol beyond TCP/IP is required as a countermeasure against denial attacks. TCP/IP builds checks into the data so that it can tell when data packets are altered, inadvertently or otherwise.

SECURITY FOR SERVER COMPUTERS

The server is the third link in the client-Internet-server electronic commerce path between the user and a Web server. Servers have vulnerabilities that can be exploited by anyone determined to cause destruction or acquire information illegally. One entry point is the Web server and its software. Other entry points are any back-end programs containing data, such as a database and the server on which it runs. Although no system is completely safe, the Web server administrator's job is to make sure that security policies are documented and considered in every part of the electronic commerce operation.

Web Server Threats

Web server software, as you learned in Chapter 8, is designed to deliver Web pages by responding to HTTP requests. Although Web server software is not inherently high-risk software, it has been designed with Web service and convenience as the main design goals. The more complex the software, the greater the probability that it contains coding errors or security weaknesses.

A Web server can compromise secrecy if it allows automatic directory listings. The secrecy violation occurs when the contents of a server's folder names are revealed to a Web browser. This happens frequently and is caused when a user enters a URL, such as http://www.somecompany.com/FAQ/, and expects to see the default page in the FAQ directory. The default Web page that the server normally displays is named index.htm or index.html. If that file is not in the directory, a Web server that allows automatic directory listings displays all of the file and folder names in that directory. Then, visitors can click folder names at random and open folders that might otherwise be off limits. Careful site administrators turn off this folder name display feature. If a user attempts to browse a folder where protections prevent browsing, the Web server issues a warning message stating that the directory is not available.

Web servers can compromise security by requiring users to enter a username and password. The username and password can be subsequently revealed when the user visits multiple pages within the same Web server's protected area if the server requires that users reestablish their usernames and passwords for each protected page they visit. This repeated information requirement is necessary because the Web is stateless—it cannot remember what happened during the last transaction. The most convenient way to remember a username and password is to store the user's confidential information in a cookie on his or her computer. That way, the Web server can request confirmation of the data by requesting that the computer send a cookie. Although cookies are not inherently unsafe, a Web server should not ask a Web browser to transmit a cookie in unencrypted form. The **W3C Security FAQ** provides additional information about server security.

One of the most sensitive files on a Web server is the file that holds Web server username and password pairs. If that file is compromised, an intruder can enter privileged areas masquerading as someone else. Such an intruder can obtain usernames and passwords if that information is readily available and not encrypted. Most Web servers store user authentication information in encrypted form.

The passwords that users select can be a threat. Users sometimes select passwords that are guessed easily, such as mother's maiden name, name of a child, a telephone number, or some easily obtained identification number, such as a Social Security number. **Dictionary**

attack programs cycle through an electronic dictionary, trying every word in the book as a password. Users' passwords, once broken, may provide an opening for illegal entry into a server that can remain undetected for a long time. To prevent dictionary attacks, many organizations use a dictionary check as a preventive measure in their password assignment software. When a user selects a new password, the password assignment software checks the password against its dictionary and, if it finds a match, refuses to allow the use of that password. An organization's password assignment software dictionary typically includes common words, names (including common pet names), acronyms that are commonly used in the organization, and words or characters (including numbers) that have some meaning for the user requesting the password (for example, employees might be prohibited from using their employee numbers as passwords).

Database Threats

Electronic commerce systems store user data and retrieve product information from databases connected to the Web server. Besides storing product information, databases connected to the Web contain valuable and private information that could damage a company irreparably if disclosed or altered. Most large-scale database systems include security features that rely on usernames and passwords. Once a user is authenticated, select portions of the database become available to that user. However, some databases either store username/password pairs in an unencrypted table, or they fail to enforce security altogether and rely on the Web server to enforce security. If unauthorized users obtain user authentication information, they can masquerade as legitimate database users and reveal or download confidential and potentially valuable information. Trojan horse programs hidden within the database system can also reveal information by changing the access rights of various user groups. A Trojan horse can even remove access controls within a database, giving all users complete access to the data—including intruders.

Other Programming Threats

Web server threats can arise from programs executed by the server. Java or C++ programs that are passed to Web servers by a client, or that reside on a server, frequently make use of a buffer. A **buffer** is an area of memory set aside to hold data read from a file or database. A buffer is necessary whenever any input or output operation takes place because a computer can process file information much faster than the information can be read from input devices or written to output devices. Programs filling buffers can malfunction and overfill the buffer, spilling the excess data outside the designated buffer memory area. This is called a **buffer overrun** or **buffer overflow** error. Usually, this occurs because the program contains an error or bug that causes the overflow. Sometimes, however, the buffer overflow is intentional. The Internet Worm of 1988 was such a program. It caused an overflow condition that eventually consumed all resources until the affected computer could no longer function.

A more insidious version of a buffer overflow attack writes instructions into critical memory locations so that when the intruder program has completed its work of overwriting buffers, the Web server resumes execution by loading internal registers with the address of the main attacking program's code. This type of attack can open the Web server to severe damage because the resumed program—which is now the attacker program—may regain control of the computer, exposing its files to disclosure and destruction by the

attacking program. **The Red Hat Linux Buffer Overflow Attacks Web Page** describes the buffer vulnerabilities of Web servers that run on the Linux operating system. Good programming practices can reduce the potential damage from buffer overflows and some computers include hardware that works with the operating system to limit the effects of buffer overflows that are intentionally programmed to create damage.

A similar attack, one in which excessive data is sent to a server, can occur on mail servers. Called a **mail bomb**, the attack occurs when hundreds or even thousands of people each send a message to a particular address. The attack might be launched by a large team of well-organized hackers, but more likely the attack is launched by one or a few hackers who have gained control over others' computers using a Trojan horse virus or some other method of turning those computers into zombies. The accumulated mail received by the target of the mail bomb exceeds the allowed e-mail size limit and can cause e-mail systems to malfunction. Although it is fairly easy to track the people responsible for the attack, it is debilitating nonetheless.

Threats to the Physical Security of Web Servers

Web servers and the computers that are networked closely to them, such as the database servers and application servers used to supply content and transaction-processing capabilities to electronic commerce Web sites, must be protected from physical harm. For many companies, these computers have become repositories of important data (information about customers, products, sales, purchases, and payments). They have also become important parts of the revenue-generating function in many businesses. As key physical resources, these computers and related equipment warrant high levels of protection against threats to their physical security.

As you learned in Chapter 8, many companies use CSPs to host Web sites. Even large companies that own servers and have IT staff to maintain those servers often put the computers in a CSP facility. The security that CSPs maintain over their physical premises (see earlier section on Threats to the Physical Security of Internet Communications Channels) is, in many cases, stronger than the security that a company could provide for computers maintained at its own location.

Companies can take additional steps to protect their Web servers. Many companies maintain backup copies of server contents at a remote location. If the Web server operation is critical to the continuation of the business, a company can maintain a duplicate of the entire Web server physical facility at a remote location. In the case of a natural disaster or a terrorist attack, the Web operations can be switched over in a matter of seconds to the backup location. Examples of mission-critical Web servers that would warrant such a comprehensive (and expensive) level of physical security include airline reservation systems, stock brokerage firm trading systems, and bank account clearing systems.

Some companies rely on their service providers to help with Web server security. Major service providers that offer managed services, such as **Level 3**, **PSINet**, and **Verio Security Services**, often include Web server security as an add-on service. Other companies hire smaller, specialized security service providers to handle security (see Learning From Failures—Pilot Network Services to learn more about one alternative to this approach). Having a service provider handle security usually adds an additional $1000 to $3000 per month to the bandwidth charges. The specialized security firms often charge two to three times more than that for their services.

LEARNING FROM FAILURES

PILOT NETWORK SERVICES

Pilot Network Services began operations in 1993, at the dawn of commercial use of the Internet. Its goal was to build a network that would be secure for electronic commerce activities. It built a network that included its own carefully monitored connections to the Internet and a database of attack signatures. Attack signatures are descriptions of the Internet traffic characteristics that indicate a cracker attack on a Web server. Pilot, as a firm specializing in security services, built an excellent collection of attack signatures and kept it updated much better than other firms that were not security specialists.

Pilot maintained the Web servers for many of its clients, and it used versions of the operating systems and Web server software that it had customized to be especially resistant to attacks. Pilot's engineers meticulously applied patches for all known points of access to the software and worked to identify new, as yet unknown, points of vulnerability—for which they immediately created and applied protective patches. For customers hosting their own servers, Pilot provided the Internet connection through its own secure network. The router between the client's network and Pilot's network and the operating system running the Pilot network were customized to eliminate any known security loopholes.

Pilot had 24/7 monitoring of its network by computer security experts, in addition to the network technicians that any other Web hosting company would provide as part of a managed services offering. Because it offered high-quality services, its fees were considerably higher than the security service charges imposed by other service providers. Typical charges were $6000 per month for the basic connection, plus $4000 per month for each Web server.

Even at these high prices, Pilot had many fans among the Fortune 500. Pilot never had more than 300 customers, but it monitored more than 70,000 individual networks for a customer list that included General Electric, PeopleSoft, Sovereign Bancorp, The Washington Post Company, and many other major accounts. By 1999, Pilot appeared to be doing well. Its revenue had increased more than 80 percent over 1998. News releases were issued regularly announcing new customers.

In late 2000, Pilot's stock price began to fall, along with the stock prices of many companies in Internet-related businesses. Although Pilot's sales were growing, its costs were escalating at an even more rapid rate. The company had never reported a profit, and its annual losses had increased to $21.7 million in 2000. Pilot executives assured its customers that the company was financially sound, but the ability of companies in Internet-related businesses to survive on the promise of future earnings had disappeared. Pilot's ability to raise the cash it needed to continue operating had vanished.

continued

In early 2001, some Pilot customers noticed that the service was failing. Phone calls and e-mails were not being returned quickly. On the afternoon of April 25, 2001, Pilot employees received four e-mails. The first explained that telephones would be disconnected that evening. The second asked all employees to turn in their mobile phones and pagers. The third announced that the chief financial officer had resigned. The final e-mail stated that all employees were out of a job as of 4:30 p.m.

Pilot's clients, many of which found out about the collapse from the Pilot employees who had been servicing their accounts, were in serious trouble. Connections to the Internet vanished with no warning. The companies that had used Pilot to host entire Web operations were in an even worse situation. A group of Pilot customers convinced AT&T (the provider of Pilot's Internet connections) to continue to carry traffic from Pilot, even though Pilot had not paid AT&T. Providian Financial, a major bank holding company and credit card processor, sent its own employees into Pilot operations centers to keep Providian's Web servers operating. Other Pilot customers that were Providian's competitors protested loudly. Most Pilot customers were concerned that their Web servers were suddenly open and vulnerable to attack.

Several of Pilot's competitors tried to raise funding to take over the business, but all of those attempts failed, and on May 9, 2001—two weeks after the collapse—AT&T cut Internet service and Pilot was liquidated. Pilot's former customers were scrambling to hire security staff, find alternative hosting firms, or join forces with other companies to keep their electronic commerce sites operating. The lesson from this failure is that security is a critical part of an electronic commerce operation. It should be handled with the same care that a company would use to protect any physical asset. If any part of the security function is handed over to another company, that company's condition becomes an important concern and must be monitored carefully.

Access Control and Authentication

Access control and authentication refers to controlling who and what has access to the Web server. Most people who work with Web servers in electronic commerce environments do not sit at a keyboard connected to the server. Instead, they access the server from a client computer. Recall that authentication is verification of the identity of the entity requesting access to the computer. Just as users can authenticate servers with which they are interacting, servers can authenticate individual users. When a server requires positive identification of a user, it requests that the client send a certificate.

The server can authenticate a user in several ways. First, the certificate represents the user's admittance voucher. If the server cannot decrypt the user's digital signature contained in the certificate using the user's public key, then the certificate did not come from the true owner. Otherwise, the server is certain that the certificate came from the owner. This procedure prevents fraudulent certificates of "admission" to a secure server. Second, the server checks the timestamp on the certificate to ensure that the certificate has not expired. A server will reject an expired certificate and provide no further service. Third, a server can use a callback system in which the user's client computer name and address are checked against a list of usernames and assigned client computer addresses. Such a

system works especially well in an intranet where usernames and client computers are controlled closely and assigned systematically. On the Internet, a callback system is more difficult to manage—particularly if client users are mobile and work from different locations. It is easy to see how certificates issued by trusted CAs play a central role in authenticating client computers and their users. Certificates provide attribution —irrefutable evidence of identity—if a security breach occurs.

Usernames and passwords can also provide some element of protection. To authenticate users using passwords and usernames, the server must acquire and store a database containing rightful users' passwords and usernames. Many Web server systems store usernames and passwords in a file. Large electronic commerce sites usually keep username/password combinations in a separate database with built-in security features.

The easiest way to store passwords is to maintain usernames in plain text and encrypt passwords using a one-way encryption algorithm. With the plain text username and encrypted password stored, the system can validate users when they log on by checking the usernames they enter against the list of usernames stored in the database. The password that a user enters when he or she logs on to a system is encrypted. Then the resulting encrypted password from the user is checked against the encrypted password stored in the database. If the two encrypted versions of the password match for the given user, the login is accepted. That is why even a system administrator cannot tell you what your forgotten password is on most systems. Instead, the administrator must assign a new temporary password that the user can change to another password. Passwords are not immune to discovery, and a person truly intent on stealing a password can often figure out a way to do so.

Note that the site visitor can save his or her username and password as a cookie on the client computer, which allows access to subscription areas of the site without entering the username and password on subsequent site visits. The trouble with that system of cookies is that the information might be stored on the client computer in plain text. If the cookie contains login and password information, then that information is visible to anyone who has access to the user's computer.

Web servers often provide access control list security to restrict file access to selected users. An **access control list (ACL)** is a list or database of files and other resources and the usernames of people who can access the files and other resources. Each file has its own access control list. When a client computer requests Web server access to a file or document that has been configured to require an access check, the Web server checks the resource's ACL file to determine if the user is allowed to access that file. This system is especially convenient to restrict access of files on an intranet server so that individuals can only access selected files on a need-to-know basis. The Web server can exercise fine control over resources by further subdividing file access into the activities of read, write, or execute. For example, some users may be permitted to read the corporate employee handbook, but not allowed to update or write to the file. Only the human resources (HR) manager would have write access to the employee handbook, and that access privilege is stored along with the HR manager's ID and password in an ACL.

Firewalls

A **firewall** is software or a hardware and software combination that is installed in a network to control the packet traffic moving through it. Most organizations place a firewall at the Internet entry point of their networks. The firewall provides a defense between a network

and the Internet or between a network and any other network that could pose a threat. Firewalls have the following characteristics:

- All traffic from inside to outside and from outside to inside the network must pass through it.
- Only authorized traffic, as defined by the local security policy, is allowed to pass through it.
- The firewall itself is immune to penetration.

Those networks inside the firewall are often called **trusted**, whereas networks outside the firewall are called **untrusted**. Acting as a filter, firewalls permit selected messages to flow into and out of the protected network. For example, one security policy a firewall might enforce is to allow all HTTP (Web) traffic to pass back and forth, but disallow FTP or Telnet requests either into or out of the protected network. Ideally, firewall protection should prevent access to networks inside the firewall by unauthorized users, and thus prevent access to sensitive information. Simultaneously, a firewall should not obstruct legitimate users. Authorized employees outside the firewall ought to have access to firewall-protected networks and data files. Firewalls can separate corporate networks from one another and prevent personnel in one division from accessing information from another division of the same company. Using firewalls to segment a corporate network into secure zones serves as a coarse need-to-know filter.

Large organizations that have multiple sites and many locations must install a firewall at each location that has an external connection to the Internet. Such a system ensures an unbroken security perimeter that is effective for the entire corporation. In addition, each firewall in the organization must follow the same security policy. Otherwise, one firewall might permit one type of transaction to flow into the corporate network that another excludes. The result is an unwanted access that is permitted throughout the corporation because one firewall left a small security door open to the entire network.

Firewalls should be stripped of any unnecessary software. Because the firewall computer is used only as a firewall and not as a general-purpose computing machine, only essential operating system software and firewall-specific protection software should remain on the computer. Having fewer software programs on the system should reduce the chances for malevolent software security breaches. Access to a firewall should be restricted to a console physically connected directly to the firewall machine. Otherwise, remote administration of the firewall must be provided, which opens up the possibility of a break in the firewall by an imposter remotely accessing the firewall along the same path that an administrator would use.

Firewalls are classified into the following categories: packet filter, gateway server, and proxy server. **Packet-filter firewalls** examine all data flowing back and forth between the trusted network (within the firewall) and the Internet. Packet filtering examines the source and destination addresses and ports of incoming packets and denies or permits entrance to the packets based on a preprogrammed set of rules.

Gateway servers are firewalls that filter traffic based on the application requested. Gateway servers limit access to specific applications such as Telnet, FTP, and HTTP. Application gateways arbitrate traffic between the inside network and the outside network. In contrast to a packet-filter technique, an application-level firewall filters requests and logs

them at the application level, rather than at the lower IP level. A gateway firewall provides a central point where all requests can be classified, logged, and later analyzed. An example is a gateway-level policy that permits incoming FTP requests, but blocks outgoing FTP requests. That policy prevents employees inside a firewall from downloading potentially dangerous programs from the outside.

Proxy server firewalls are firewalls that communicate with the Internet on the private network's behalf. When a browser is configured to use a proxy server firewall, the firewall passes the browser request to the Internet. When the Internet sends back a response, the proxy server relays it back to the browser. Proxy servers are also used to serve as a huge cache for Web pages.

One problem faced by companies that have employees working from home is that the location of computers outside the traditional boundaries of the company's physical site expands the number of computers that must be protected by the firewall. This **perimeter expansion** problem is particularly troublesome for companies that have salespeople using laptop computers to access confidential company information from all types of networks at customer locations, vendor locations, and even public locations, such as airports.

Another problem faced by organizations connected to the Internet is that their servers are under almost constant attack. Crackers spend a great deal of time and energy on attempts to enter the servers of organizations. Some of these crackers use automated programs to continually attempt to gain access to servers. Organizations often install intrusion detection systems as part of their firewalls. **Intrusion detection systems** are designed to monitor attempts to login to servers and analyze those attempts for patterns that might indicate a cracker's attack is underway. Once the intrusion detection system identifies an attack, it can block further attempts that originate from the same IP address until the organization's security staff can examine and analyze the access attempts and determine whether they are an attack.

In addition to firewalls installed on organizations' networks, it is possible to install software-only firewalls on individual client computers. These firewalls are often called **personal firewalls**. The use of personal firewalls, such as **ZoneAlarm**, has become an important tool in the protection of expanded network perimeters for many companies. Many home computer users are installing personal firewalls on their home networks. You can learn more about firewall protection for your home computer at the **Gibson Research Shields Up!** Web site.

ORGANIZATIONS THAT PROMOTE COMPUTER SECURITY

Following the occurrence of the Internet Worm of 1988, a number of organizations were formed to share information about threats to computer systems. These organizations are devoted to the principle that sharing information about attacks and defenses for those attacks can help everyone create better computer security. Some of the organizations began at universities; others were launched by government agencies. In this section, you will learn about some of these organizations and their resources.

CERT

In 1988, a group of researchers met to study the infamous Internet Worm attack soon after it occurred. They wanted to understand how worms worked and how to prevent damage from future attacks of this type. The National Computer Security Center, part of the National Security Agency, initiated a series of meetings to figure out how to respond to future security breaks that might affect thousands of people. Soon after that meeting of security experts in 1988, the U.S. government created the Computer Emergency Response Team and housed it at Carnegie Mellon University in Pittsburgh. The organization is now operated as part of the federally funded Software Engineering Institute at Carnegie Mellon, and it has changed its legal name from the Computer Emergency Response Team (which had been abbreviated to "CERT" by most people who wrote and talked about it) to **CERT**. CERT still maintains an effective and quick communications infrastructure among security experts so that security incidents can be avoided or handled quickly.

Today, CERT responds to thousands of security incidents each year and provides a wealth of information to help Internet users and companies become more knowledgeable about security risks. CERT posts alerts to inform the Internet community about security events, and it is regarded as a primary authoritative source for information about viruses, worms, and other types of attacks.

Other Organizations

CERT is the most prominent of these organizations and has formed relationships, such as the **Internet Security Alliance**, with other industry associations. However, CERT is not the only computer security resource. In 1989, one year after CERT was formed, a cooperative research and educational organization called the Systems Administrator, Audit, Network, and Security Institute was launched. Now known as the **SANS Institute**, this organization includes more than 150,000 members who work in computer security consulting firms and information technology departments of companies as auditors, systems administrators, and network administrators.

Many SANS education and research efforts yield resources such as news releases, research reports, security alerts, and white papers that are available on the Web site at no cost. SANS also sells publications to generate funds that it uses for research and educational programs. The SANS Institute operates the **SANS Internet Storm Center**, a Web site that provides current information on the location and intensity of computer attacks throughout the world. Purdue University's Center for Education and Research in Information Assurance and Security **(CERIAS)** is a center for multidisciplinary research and education in information security. The CERIAS Web site provides resources in computer, network, and communications security and includes a section on information assurance. The **Center for Internet Security** is a not-for-profit cooperative organization devoted to helping companies that operate electronic commerce Web sites reduce the risk of disruptions from technical failures or deliberate attacks on their computer systems. It also provides information to auditors who review such systems and to insurance companies that provide coverage for companies who operate such systems. **Microsoft Security Research Group** is a privately sponsored site that offers free information about computer security issues. For current information about computer security, you can visit **CSO Online**, which carries

articles that have appeared in *CSO Magazine* along with other news items related to computer security.

The U.S. government has several Web sites devoted to security enhancement efforts. The **U.S. Department of Justice's Cybercrime** site offers information about computer crimes and intellectual property violations. The U.S. Department of Homeland Security operates the **National Infrastructure Protection Center** (NIPC) Web site, which provides information about threats to U.S. infrastructure, including its computing infrastructure.

Computer Forensics and Ethical Hacking

A small group of firms, endorsed by corporations and security organizations, have the unlikely job of breaking into client computers. Called **computer forensics experts** or **ethical hackers**, these computer sleuths are hired to probe PCs and locate information that can be used in legal proceedings. The field of **computer forensics** is responsible for the collection, preservation, and analysis of computer-related evidence. Ethical hackers are often hired by companies to test computer security safeguards. Links to the Web sites of several companies that offer computer forensics and ethical hacking services are included in the Additional Resources section of the Online Companion for this chapter.

Summary

Electronic commerce is vulnerable to a wide range of security threats. Attacks against electronic commerce systems can disclose or manipulate proprietary information. The three general assets that companies engaging in electronic commerce must protect are client computers, computer communication channels, and Web servers. Key security provisions in each of these parts of the Web client-Internet-Web server linkage are secrecy, integrity, and available service. Threats to commerce can occur anywhere in the commerce chain. News accounts of virus attacks have kept Web users aware of the security risks to client computers. Antivirus software is also an important element in the protection of client computers. More subtle threats are delivered as client-side applets. Java, JavaScript, and ActiveX controls run on client machines and have the potential to breach security. Cookies, if not controlled and used properly, can present threats to client computers.

Communication channels, in general, and the Internet, in particular, are especially vulnerable to attacks. The Internet is a vast network and because no control exists over the nodes through which Internet traffic passes, information sent through the Internet is vulnerable to unauthorized disclosure. This can lead to disclosure of private information, alteration of critical business documents, and theft or loss of important business messages. Encryption provides secrecy, and several forms of encryption are available that use hash functions or other more complex algorithms. They include private-key and public-key techniques. Although public-key encryption eliminates the problem of sharing a secret key, it is much slower than private-key encryption. Private-key encryption is used during most commerce sessions because it is fast and efficient. Integrity protections ensure that messages between clients and servers are not altered. Digital certificates provide both integrity controls and user authentication. A trusted third party such as a certification authority can provide digital certificates to users and organizations. Several Internet protocols, including Secure Sockets Layer and Secure HTTP, use encryption to provide secure Internet transmission capabilities. As wireless networks have grown to become important parts of the data communication infrastructure, security concerns have increased. Although many wireless networks (especially home networks) are installed without security features, wireless encryption methods that make them more secure are available. Most wireless networks installed in businesses today do have wireless encryption.

Web servers are susceptible to security threats. Programs that run on servers have the potential to damage databases, abnormally terminate server software, or make subtle changes in proprietary information. Attacks can come from within the server in the form of programs, or they can come from outside the server. One type of external attack can occur when a message overflows a server's internal storage region and overwrites crucial information. Overwritten information is replaced with either data or instructions that cause other programs on the server to execute. Backup copies of servers provide redundancy in the case of a physical threat to a server. The Web server must be protected from both physical threats and Internet-based attacks on its software. Protections for the server include access control and authentication, provided by username and password login procedures and client certificates. Firewalls can be used to separate trusted inside computer networks and clients from untrusted outside networks, including other divisions of a company's enterprise network system and the Internet.

A number of organizations have been formed to share information about computer security threats and defenses. CERT, the SANS Institute, and similar organizations address security outbreaks by linking knowledgeable security experts. When large security outbreaks occur, the members of these organizations join together and discuss methods to locate and eliminate the threat. Computer forensics firms that undertake attacks against their clients' computers can play an important role in helping identify security weaknesses.

Key Terms

Access control list (ACL)
Active content
Active wiretapping
ActiveX
Advanced Encryption Standard (AES)
Antivirus software
Applet
Asymmetric encryption
Backdoor
Biometric security device
Black hat hacker
Buffer
Buffer overrun (buffer overflow)
Certification authority (CA)
Cipher text
Collision
Computer forensics
Computer forensics expert
Computer security
Cookie blocker
Countermeasure
Cracker
Cryptography
Cybervandalism
Data Encryption Standard (DES)
Decrypted
Decryption program
Dictionary attack program
Digital certificate (digital ID)
Digital signature
Domain name server (DNS)
Eavesdropper
Encryption
Encryption algorithm
Encryption program
Ethical hacker
Firewall
First-party cookies
Gateway server
Hacker
Hash algorithm
Hash coding
Hash value
Integrity
Integrity violation
Intrusion detection system
Java sandbox
JavaScript
Key
Logical security
Macro virus
Mail bomb
Man-in-the-middle exploit
Masquerading (spoofing)
Message digest
Multivector virus
Necessity
Necessity threat (delay, denial, or denial-of-service threat)
One-way function
Open session
Packet-filter firewall
Perimeter expansion
Persistent cookie

Personal firewall
Phishing expeditions
Physical security
Plain text
Plug-ins
Pretty Good Privacy (PGP)
Privacy
Private key
Private-key encryption
Proxy server firewall
Public key
Public-key encryption
Scripting language
Secrecy
Secure envelope
Secure Sockets Layer
Security policy
Session cookie
Session key
Signed (message or code)
Sniffer program
Stateless connection
Steganography
Symmetric encryption
Third-party cookies
Threat
Triple Data Encryption Standard (Triple DES, 3DES)
Trojan horse
Trusted (network)
Untrusted (network)
Untrusted Java applet
Warchalking
Wardrivers
Web bug
White hat hacker
Worm
Zombie

Review Questions

RQ1. In about 200 words, explain why Web sites use cookies. In your answer, discuss the reasons that cookies were first devised and explain where cookies are stored. You can use the links in the Online Companion to help with your research.

RQ2. In about 100 words, describe steganography and explain its connection to the topic of online security. You can use the links in the Online Companion to help with your research.

RQ3. In about 200 words, explain the differences between public-key encryption and private-key encryption. List advantages and disadvantages of each encryption method. Explain which method you would use for e-mail sent from a field sales office to corporate headquarters. Assume that the e-mail regularly includes highly confidential information about upcoming sales opportunities.

RQ4. In about 300 words, describe the security threats that a company will face when it implements a wireless network. Assume that the company occupies the six middle floors in a 12-story office building that is located in a downtown business area between two other buildings of similar height. Briefly explain how the company could reduce the risks it faces.

RQ5. Consider the reasons that programs such as Java applets that run on client machines are considered security threats. In about 200 words, explain how these programs could breach security and compare the security risks posed by JavaScript programs to the risks posed by Java applets.

RQ6. Write a 200-word description of computer forensics in general and ethical hacking in particular. In your essay, describe at least one real situation in which computer forensic experts or ethical hackers used their talents to help a company overcome a security weakness.

Exercises

E1. Brought Back Bugs is a used Volkswagen dealer in Lincoln, Nebraska. The dealership hired you to update its Web site. One of the requirements is that the site must display a few banner advertisements showing the week's specials. You decide that active content would be the best technology to automate the placement and rotation of the advertisement. You are also considering using active content to make the content of the banner ad more interesting. You decide to investigate Java, JavaScript, Jscript, and Java applets as alternatives. Use the Online Companion and Web search engines to learn more about these alternatives, and write a 300-word summary that describes each and evaluates its use for automating the rotation and placement of banner ads on the Brought Back Bugs Web site.

E2. You are the administrator of a Web server for an electronic commerce site. The site receives about 12,000 visitors per day, maintains a product catalog of about 4000 items, and processes about 2000 sales per day. The average sale amount is $87. The site accepts four major credit cards and it outsources its payment processing for all of the credit cards to another company. In about 200 words, describe the types of threats that could be launched against your Web server, given the types of activity (catalog presentation, order entry, and payment processing) it handles and the volume of those activities. Consult sources on the Internet or in your library to help you complete this exercise.

E3. Write a 300-word paper in which you evaluate the CERT organization. Include information about when it was founded, what groups or people are members, and where it is headquartered. Include in your discussion at least three current security alerts, specifying the name of the virus or attack program, the date the alert was posted, and two sentences about each reported security alert. Use Internet search engines and the CERT Web site to help you locate information.

Cases

C1. Bibliofind

Bibliofind was founded in 1996 as one of the first Web sites to specialize in hard-to-find and collectible books. The site featured a powerful search engine for used and rare books. The search engine's database was populated with the results of Bibliofind's daily surveys of a worldwide network of suppliers. Registered site visitors could specify the title for which they were searching, a price range, and whether they were seeking a first edition. The site also allowed visitors to build a wish list that would trigger an e-mail when a specific book on the list became available.

Bibliofind had developed a large customer list, an excellent reputation, and a solid network of rare book dealers, all of which made the company an attractive acquisition for other online bookstores. In 1999, Amazon.com bought Bibliofind, but Bibliofind continued to operate its own Web site and conduct its business as it had before the acquisition.

In 2001, Bibliofind's Web site was hacked. The cracker had gained access to the company's Web server and replaced the company's Web pages with defaced versions. Bibliofind shut down its Web site for several days and undertook a complete review of its Web site's security. When the company's IT staff examined the server logs carefully, they found that the Web page hacking was only the tip of the iceberg. Entries in the logs showed that attackers had been accessing Bibliofind's computers for more than four months. Even worse, some of the crackers had been able to go through the Web servers to gain access to the computers that held Bibliofind customer information, including names, addresses, and credit card numbers. That information had been stored in plain text files on Bibliofind's transaction servers.

Bibliofind called in state and federal law enforcement officials to investigate the hacking incidents and sent an e-mail notification to the 98,000 customers whose private information might have been obtained by the crackers. The investigation did not result in any arrests, nor did it determine the identity of the intruders. Many of Bibliofind's customers were very upset when they learned what had happened.

A month after the hacking incident, Amazon.com moved Bibliofind into its zShops online mall. As an Amazon zShop, Bibliofind could process its transactions through Amazon's system and no longer needed to maintain private information about its customers on its computers. Eventually, Bibliofind was closed down. A successful business had been seriously damaged because it failed to maintain adequate security over the customer information it had gathered.

Required:

1. In about 300 words, explain how Bibliofind might have used firewalls to prevent the intruders from gaining access to its transaction servers. Be specific about where the firewalls should have been placed in the network and what kinds of rules they should have used to filter network traffic at each point.
2. In about 200 words, explain how encryption might have helped prevent or lessen the effects of Bibliofind's security breach.
3. In 2003, the State of California enacted a law that requires companies to inform customers whose private information might have been exposed during a security breach like the one that Bibliofind experienced. While the legislation was being debated, businesses argued that the law would encourage nuisance lawsuits. In about 300 words, present arguments for and against this type of legislation.

Note: Your instructor might assign you to a group to complete this case, and might ask you to prepare a formal presentation of your results to your class.

C2. Wilderness Trailhead

Wilderness Trailhead, Inc. (WTI) is a retailer that offers hiking, rock-climbing, and survival gear for sale on its Web site. WTI targets the serious outdoor enthusiast and offers high-quality equipment at competitive prices. The company has been in business on the Web for five years. It has grown rapidly and has been profitable since its first year of operations. WTI offers about 1200 different items for sale and has about 1000 visitors per day at its Web site. Because the company's offerings are specialized and high quality, its average transaction size is much higher than other outdoor equipment stores. WTI makes about 200 sales each day on its site, with an average transaction value of $372.

WTI sells products primarily through its Web site (it does have a small retail outlet store for discontinued items in Bellingham, Washington) to customers in the United States and Canada. WTI ships orders from its two warehouses—one in Vancouver, British Columbia, and a second in Shoreline, Washington.

WTI accepts four major credit cards and processes its own credit card transactions. It stores records of all transactions on a database server that shares a small room with the Web server computer at WTI's main offices in a small industrial park just outside of Bellingham. Harry Bogdosian, the manager of IT for WTI, has become increasingly concerned about the security of the company's Web and database servers as the company has grown.

Required:

1. WTI faces certain risks that arise from its storage of customer credit card numbers on its database server. List at least four specific threats to the database server's security, and identify defenses, deterrents, or countermeasures that might reduce or eliminate the potential damage that could be caused by those threats.
2. Write a security policy for the operation of the WTI database server. Be sure to consider the threats that exist because that server stores customer credit card numbers. You can use the links included in the Online Companion under the heading "Computer Security Policy Resources" to help you as you write this policy.
3. WTI is considering moving its existing Web and database server computers to a CSP in a co-location arrangement. Prepare a two-page outline of the security features that WTI should ask a CSP to provide as part of this co-location service.

Note: Your instructor might assign you to a group to complete this case, and might ask you to prepare a formal presentation of your results to your class.

C3. Materials Equipment

You are an information technology (IT) consultant to Materials Equipment, Inc. (MEI), a major industrial equipment distributor. Its products include materials-handling machinery for assembly lines and product-packaging areas, hydraulic equipment (for moving fluids), hoses, hose fittings, and similar items. MEI has been in business for more than 70 years and sells more than $200 million worth of parts and equipment each year to its 3000 customers. MEI's customers are located all over the world, but most are in the United States, Mexico, Malaysia, China, and Singapore.

Joe Andrejewski, MEI's director of sales, has retained you to help him with a new marketing idea. He has read about other companies that have created Web portal sites for customers, and he is interested in developing a portal site that MEI could operate with three other companies that sell products (such as bearings, seals, hoses, and hose fittings) and services (design, layout, and installation of materials-handling equipment) that are complementary to MEI products. The portal would provide MEI customers with a Web site at which they could buy MEI products, buy the products and services of the three MEI strategic partners, and obtain information about current trends in industrial equipment technologies and the application of those technologies. The portal site would also include a used equipment area in which MEI customers could list equipment for sale. Joe believes that giving customers a convenient way to liquidate old equipment will make it easier for his sales representatives to sell new equipment to those customers.

Joe has put together an internal team to examine the feasibility of the portal site, including key employees from MEI's Sales, Finance, Product Engineering, and IT Services departments. The team has identified several security issues that they want to resolve before they take the portal idea much further. Joe would like you to help the team understand two security technologies—digital certificates and encryption—and how these techniques might be used in MEI's proposed portal site.

Required:

1. Prepare two briefing reports of about 700 words each for the MEI portal team—one about digital certificates and one about encryption. Each report should explain the technology and describe one or two common applications.
2. Assume that the MEI portal project is approved and implemented. Further assume that MEI has decided to require each customer that participates in the portal to obtain a digital certificate. Write a memo of about 500 words addressed to potential participants (MEI customers) in which you explain why they must obtain a digital certificate as a condition of participation.

Note: Your instructor might assign you to a group to complete this case, and might ask you to prepare a formal presentation of your results to your class.

For Further Study and Research

Alexander, S. 2000. "Viruses, Worms, Trojan Horses and Zombies," *Computerworld*, 34(18), May 1, 74.

Anderson, R. and F. Petitcolas. 1998. "On the Limits of Steganography," *IEEE Journal of Selected Areas in Communications*, 16(4), May, 474–481.

Austin, R. and C. Darby. 2003. "The Myth of Secure Computing," *Harvard Business Review*, 81(6), June, 120–126.

Bank, D. and R. Richmond. 2005 "Where the Dangers Are: The Threats to Information Security That Keep the Experts Up at Night," *The Wall Street Journal*, July 18, R1.

Betts, M. 2000. "Digital Signatures Law to Speed Online B-to-B Deals," *Computerworld*, 34(26), June 26, 8.

Cohen, A. 2001. "When Terror Hides Online," *Time*, November 12, 65.

Colkin, E., A. Gilbert, G. Hulme, M. McGee, and J. Rendleman. 2001. "IT Security and the Law," *Information Week*, November 26, 22–24.

Connell, S. 2004. "Security Lapses, Lost Equipment Expose Students to Possible ID Theft Loss," *The Los Angeles Times*, August 29, B4.

Costanzo, C. 2003. "Dealing with Phishing and Spoofing," *American Banker*, 168(184), September 24, 10.

Creighton, D. 2004. "Chronology of Virus Attacks," *The Wall Street Journal*, May 13. (http://online.wsj.com/article/0,,SB108362410782000798,00.html)

Dacey, R. 2001. *Information Security: IRS Electronic Filing Systems* (GAO-01-306). Washington, D.C.: U.S. General Accounting Office.

DeMaria, M. 2001. "Symantec Firewall/VPN Devices Secure the Small Office at the Right Price," *Network Computing*, 12(24), November 26, 30–31.

DoD Directive 5215.1 CSC-STD-001-83. 1983. *Department of Defense Trusted Computer System Evaluation Criteria* (the "Orange Book"), Washington, D.C.

Dunleavey, M. 2005. "Don't Let Data Theft Happen to You," *The New York Times*, July 2, C7.
Erlanger, L. 2002. "Defensive Strategies," *PC Magazine*, 21(19), November 5, 70–72.
Evers, J. 2001. "Hackers Get Credit Card Data from Amazon's Bibliofind," *PC World*, March 6. (http://www.pcworld.com/news/article/0,aid,43582,00.asp)
Files, J. 2005. "For Fourth Time, Judge Seeks to Shield Indian Data," *The New York Times*, October 25, A17.
Gallagher, S. 2002. "Best Buy: May Day Mayday for Security," *Baseline*, June 7. (http://www.baselinemag.com/article2/0,3959,687,00.asp)
Garfinkel, S. and G. Spafford. 2002. *Web Security, Privacy, & Commerce*. Cambridge, MA: O'Reilly.
Glass, B. and D. Fisher. 2004. "Biometrics Security," *PC Magazine*, 23(1), January 20, 66.
Gonsalves, C. 2005. "Computing Insecurity," *eWeek*, May 23, 32.
Gurley, J. 2001. "From Wired To Wiretapped," *Fortune*, 144(7), October 15, 214–215.
Hancock, B. 2001. "Terrorism and Steganography: Shaken, Not Stirred," *Computers & Security*, 20(2) 110–111.
Harrison, A. 2000. "Advanced Encryption Standard," Computerworld, 34(22), May 29, 57.
Hayes, F. 2002. "Thanks, Warchalkers," *Computerworld*, 36(35), August 26, 56.
Information Technology Association of America (ITAA). 2000. *Intellectual Property Protection in Cyberspace*. Arlington, VA: ITAA.
Katzenheisser, S. and F. Petitcolas (eds.). 1999. *Information Hiding Techniques for Steganography and Digital Watermarking*, Norwood. MA: Artech House.
Krim, J. 2003. "WiFi Is Open, Free and Vulnerable to Hackers: Safeguarding Wireless Networks Too Much Trouble for Many Users," *The Washington Post*, July 27, A1.
Krim, J. 2003. "Microsoft Critic Forced Out, Firm Does Business With Software Giant," *The Washington Post*, September 26, E1.
Kuchinskas, S. 2003. "Lack of Trust Could Impact E-Commerce Sales," *E-Commerce Guide*, December 3. (http://ecommerce.internet.com/news/news/article/0,10375_3115741,00.html)
Kutter, M. 1998. "Watermarking Resistance to Translation, Rotation, and Scaling," *Proceedings of SPIE: Multimedia Systems and Applications*, Vol. 3528, November 1–6, 423–431.
Lohmeyer, D., J. McCrory, and S. Pogreb. 2002. "Managing Information Security," *The McKinsey Quarterly*, June, 12–15.
Mackey, D. 2003. *Web Security for Network and System Administrators*. Boston: Course Technology.
Manes, S. 2001. "Security, Microsoft Style: No Safety Net?" *PC World*, 19(11), November, 210.
Maney, K. 2001. "Osama's Messages Could Be Hiding in Plain Sight," *USA Today*, December 19, 6B.
McCracken, H. 2004. "Microsoft's Security Problem—and Ours," *PC World*, 22(1), January, 25.
McCullagh, D. 2001. "'Secure' U.S. Site Wasn't Very," *Wired News*, July 6. (http://www.wired.com/news/privacy/0,1848,45031,00.html)
Nerney, C. 2003. "Get It Right, Redmond," *Internet News*, May 12. (http://www.internetnews.com/commentary/ article.php/2205081)
Nielsen, J. 2004. "User Education Is Not the Answer to Security Problems," *Alertbox*, October 25. (http://www.useit.com/alertbox/20041025.html)
Null, C. 2000. "Name Grab," *PC Computing*, 13(4), April, 40–42.
Oppliger, R. 1997. "Internet Security: Firewalls and Beyond," *Communications of the ACM*, 40(5), May, 92–102.
Palmer, C. 2001. "Ethical Hacking," *IBM Systems Journal*, 40(3), 769–780.

Petreley, N. 2001. "The Cost of Free IIS," *Computerworld*, 35(43), October 22, 49.
Piazza, P. 2003. "Phishing for Trouble," *Security Management*, 47(12), December, 32–33.
Pleas, K. 1999. "Certificates, Keys, and Security," *PC Magazine*, 18(8), April 20, 227–230.
Powell, T. 2004. "Quick Tips for Web Application Security," *Network World*, 21(20), May 17, 50–51.
Regan, K. 2001. "Hack Victim Bibliofind to Move to Amazon," *E-Commerce Times*, April 6. (http://www.ecommercetimes.com/story/8768.html)
Rivest, R. 1992. *The MD5 Message-Digest Algorithm*, IETF RFC 1321.
Rosencrance, L. 2004. "Federal Audit Raises Doubts About IRS Security System," *Computerworld*, 38(36), September 6, 9.
Rothstein, P. 2001. "Disaster Recovery: September 11 Changes Everything," *Information Security*, 4(11), 48–49.
Rubenking, N. 2002. "Securing Web Services," *PC Magazine*, 21(17), October 1, IP01–04.
Rutrell, Y. 2001. "So Many Patches, So Little Time," *InternetWeek*, October 8, 1–2.
Saita, A. 2001. "Deep Digital Cover," *Information Security*, 4(10), October, 22.

Schwartz, J. 2002. "13,000 Credit Reports Stolen by Hacker," *The New York Times*, May 17. (http://www.nytimes.com/2002/05/17/technology/17IDEN.html)
Security Management, 2002. "Government Infosec Gets Failing Grade,"46(2), February, 34–35.
Shipley, G. 2001. "Growing Up with a Little Help from the Worm," *Network Computing*, 12(20), October 1, 39.
Shively, G. 2002. *Network InSecurity*. Newport Beach, CA: PivX Solutions.
Skoudis, E. 2002. "Infosec's Worst Nightmares: The Five Past Attacks That Haunt Us, the Five Fears That Trouble Us," *Information Security*, November. (http://www.infosecuritymag.com/2002/nov/nightmares.shtml)
Skoudis, E. 2005. "Five Malicious Code Myths and How To Protect Yourself in 2005," *SearchSecurity.com*, January 4. (http://searchsecurity.techtarget.com/tip/1,289483,sid14_gci1041736,00.html)
Sterling, B. 2001. "Steganography Goes Digital," *The New York Times*, December 9, 102.
Thompson, H. and J. Whittaker. 2002. "Testing for Software Security," *Dr. Dobb's Journal*, November, 24–32.
Tippett, P. 2001. "The Crypto Myth," *Information Security*, 4(5), May, 38–40.
U.S. National Institute of Standards and Technology. 1993. *Data Encryption Standard (DES): Federal Information Processing Standards Publication 46–2*. Gaithersburg, MD: U.S. Computer Systems Laboratory.
Verton, D. 2001. "Microsoft in Hot Seat After Code Red," *Computerworld*, 35(32), August 6, 1–2.
Verton, D. 2002. "Mapping of Wireless Networks Could Pose Enterprise Risk," *Computerworld*, August 14. (http://computerworld.com/securitytopics/security/story/0,10801,73479,00.html)
Verton, D. 2002. "Fixes Named Along With Top 20 Holes," *Computerworld*, 36(41), October 7, 1–2.
Vijayan, J. 2001. "Corporations Left Hanging as Security Outsourcer Shuts Doors," *Computerworld*, 35(18), April 30, 13.
Vijayan, J. 2005. "Companies Scramble to Bolster Online Security," *Computerworld*, 39(10), March 7, 1, 61.
Yeh, W-H. and J-J. Hwang. 2001. "Hiding Digital Information Using a Novel System Scheme," *Computers & Security*, 20(6), 533–538.
Zetter, K. 2001. "Holey Software!" *PC World*, 19(11), November, 135–140.