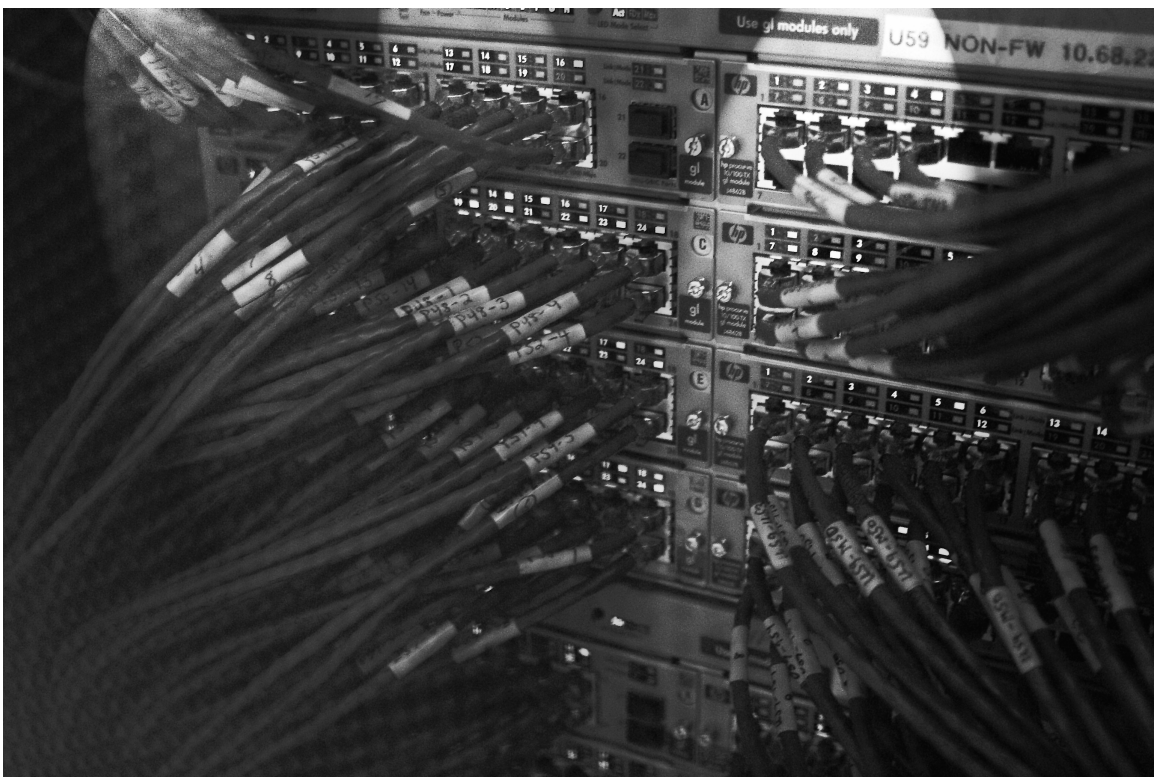
PART 4

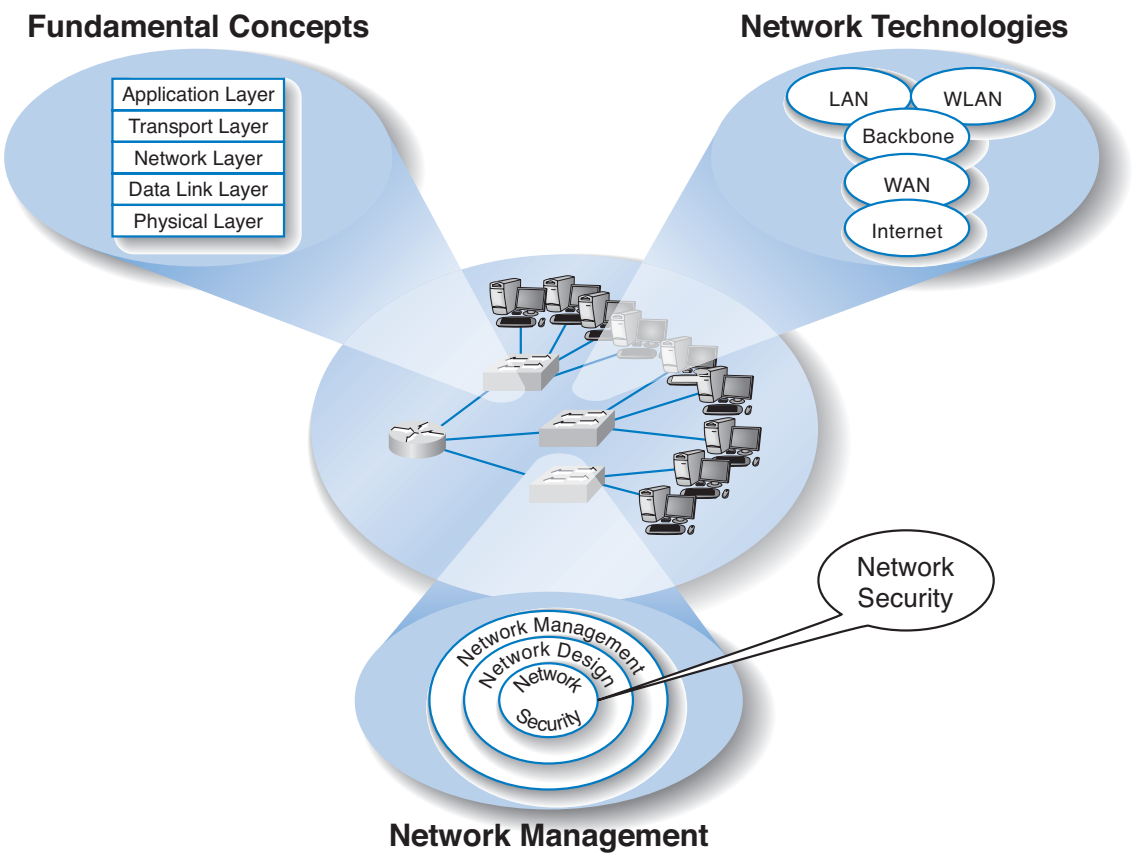
NETWORK MANAGEMENT



Courtesy Alan Dennis

CHAPTER 11

NETWORK SECURITY[1]



The Three Faces of Networking

[1]This chapter was written by Alan Dennis and Dwight Worker.

THIS CHAPTER describes why networks need security and how to provide it. The first step in any security plan is risk assessment, understanding the key assets that need protection, and assessing the risks to each. There are a variety of steps that can be taken to prevent, detect, and correct security problems due to disruptions, destruction, disaster, and unauthorized access.

OBJECTIVES

- Be familiar with the major threats to network security
- Be familiar with how to conduct a risk assessment
- Understand how to conduct business continuity planning
- Understand how to prevent intrusion

CHAPTER OUTLINE

INTRODUCTION
- Why Networks Need Security
- Types of Security Threats
- Network Controls

RISK ASSESSMENT
- Develop a Control Spreadsheet
- Identify and Document the Controls
- Evaluate the Network's Security

BUSINESS CONTINUITY PLANNING
- Preventing Disruption, Destruction, and Disaster
- Detecting Disruption, Destruction, and Disaster
- Correcting Disruption, Destruction, and Disaster

INTRUSION PREVENTION
- Preventing Intrusion
- Detecting Intrusion
- Correcting Intrusion

BEST PRACTICE RECOMMENDATIONS

IMPLICATIONS FOR MANAGEMENT

SUMMARY

INTRODUCTION

Business and government have always been concerned with physical and information security. They have protected physical assets with locks, barriers, guards, and the military since organized societies began. They have also guarded their plans and information with coding systems for at least 3,500 years. What has changed in the last 50 years is the introduction of computers and the Internet.

The rise of the Internet has completely redefined the nature of information security. Now companies face global threats to their networks, and, more importantly, to their data. Viruses and worms have long been a problem, but credit card theft and identity theft, two of the fastest growing crimes, pose immense liability to firms who fail to protect their customers' data. Laws have been slow to catch up, despite the fact that breaking into a computer in the United States—even without causing damage—is now a federal crime punishable by a fine and/or imprisonment. Nonetheless, we have a new kind of transborder cyber crime against which laws may apply but will be very difficult to enforce. The United States and Canada may extradite and allow prosecution of digital criminals operating within their borders, but investigating, enforcing, and prosecuting transnational cyber crime across different borders is much more challenging. And even when someone is caught they face lighter sentences than bank robbers.

Computer security has become increasingly important over the last 5 years with the passage of the Sarbanes-Oxley Act (SOX) and the Health Insurance Portability and Accountability Act (HIPAA). The number of Internet security incidents reported to the *Computer Emergency Response Team (CERT)* has doubled every year up until 2003, when CERT stopped keeping records because there were so many incidents that it was no longer meaningful to keep track.[2] CERT was established by the U.S. Department of Defense at Carnegie Mellon University with a mission to work with the Internet community to respond to computer security problems, raise awareness of computer security issues, and prevent security breaches.

Several other organizations monitor security threats. Postini, an e-mail software vendor, provides information on current virus, spam, and other threats. Figure 11.1 shows the current threats when I visited their site in 2006. About 70 percent of all e-mail sent worldwide was spam, and about 1 percent of all e-mail messages contained a virus.

Approximately 95% of the respondents to the 2005 Computer Security Institute/FBI Computer Crime and Security Survey reported that they had detected security breaches in the last 12 months. About 90% reported they suffered a measurable financial loss due to a security problem, with the average loss being about $200,000, which is significantly lower than in previous years. Experts estimate that worldwide annual losses due to security problems exceed $2 trillion.

Part of the reason for the increase in computer security problems is the increasing availability of sophisticated tools for breaking into networks. Five years ago, someone wanting to break into a network needed to have some expertise. Today, even inexperienced attackers can download tools from a Web site and immediately begin trying to break into networks.

[2]CERT maintains a Web site on security at www.cert.org. Another site for security information is www.infosyssec.net.

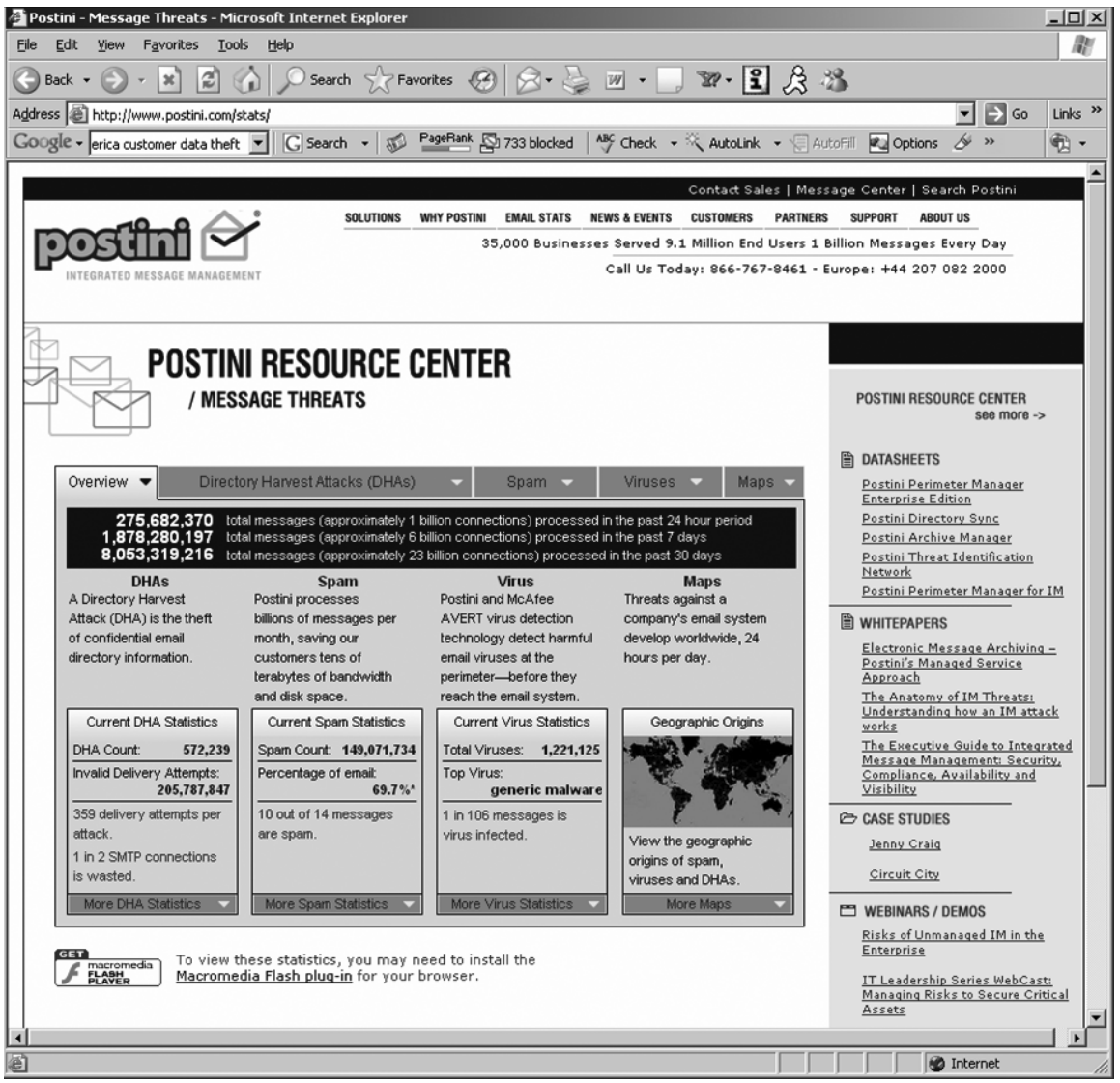


FIGURE 11.1 Current security threats. *Source:* www.postini.com/stats.

As a result, the cost of network security has increased. The CSI/FBI survey found that firms spent an average of about 5 percent of their total IT budget on network security. The average expenditure was about $250 per employee per year—and that's all employees in the organization not per IT employee, so that an organization with 100 employees spends an average of $250,000 per year on network security. About 25 percent of organizations had purchased insurance for security risks.

Why Networks Need Security

In recent years, organizations have become increasingly dependent on data communication networks for their daily business communications, database information retrieval, distributed data processing, and the internetworking of LANs. The rise of the Internet with opportunities to connect computers anywhere in the world has significantly increased the potential vulnerability of the organization's assets. Emphasis on network security also has increased as a result of well-publicized security break-ins and as government regulatory agencies have issued security-related pronouncements.

The losses associated with the security failures can be huge. An average loss of about $200,000 sounds large enough, but this is just the tip of the iceberg. The potential loss of consumer confidence from a well-publicized security break-in can cost much more in lost business. More important than these, however, are the potential losses from the disruption of application systems that run on computer networks. As organizations have come to depend upon computer systems, computer networks have become "mission-critical." Bank of America, one of the largest banks in the United States, estimates that it would cost the bank $50 million if its computer networks were unavailable for 24 hours. Other large organizations have produced similar estimates.

Protecting customer privacy and the risk of identity theft also drives the need for increased network security. In 1998, the European Union passed strong data privacy laws that fined companies for disclosing information about their customers. In the United States, organizations have begun complying with the data protection requirements of the HIPAA, and a California law providing fines up to $250,000 for each unauthorized disclosure of customer information (e.g., if someone were to steal 100 customer records, the fine could be $25 million).

As you might suspect, the value of the data stored on most organizations' networks and the value provided by the application systems in use far exceeds the cost of the networks themselves. For this reason, the primary goal of network security is to protect organizations' data and application software, not the networks themselves.

Types of Security Threats

For many people, security means preventing unauthorized access, such as preventing an attacker from breaking into your computer. Security is much more than that, however. There are three primary goals in providing security: confidentiality, integrity, and availability. *Confidentiality* refers to the protection of organizational data from unauthorized disclosure of customer and proprietary data. *Integrity* is the assurance that data have not been altered or destroyed. *Availability* means providing continuous operation of the organization's hardware and software so that staff, customers, and suppliers can be assured of no interruptions in service.

There are many potential threats to confidentiality, integrity, and availability. Figure 11.2 shows some threats to a computer center, the data communication circuits, and the attached computers. In general, security threats can be classified into two broad categories: ensuring business continuity and preventing unauthorized access.

Business continuity planning refers primarily to ensuring availability, with some aspects of data integrity. There are three main threats to business continuity. *Disruptions* are the loss of or reduction in network service. Disruptions may be minor and temporary. For

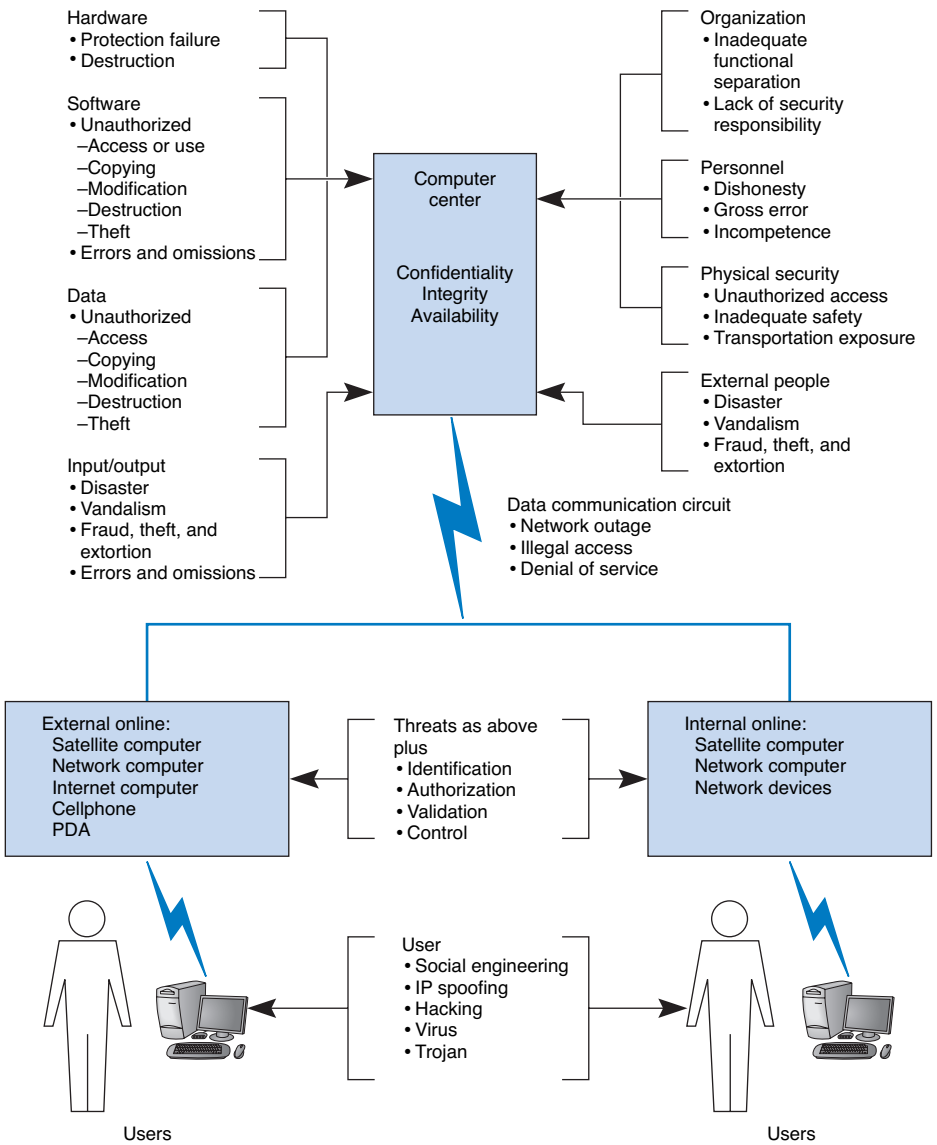


FIGURE 11.2 Some threats to a computer center, data communication circuits, and client computers.

example, a network switch might fail or a circuit may be cut causing part of the network to cease functioning until the failed component can be replaced. Some users may be affected, but others can continue to use the network. Some disruptions may also be caused by or result in the *destruction* of data. For example, a virus may destroy files, or the "crash" of a hard disk may cause files to be destroyed. Other disruptions may be cata-

MANAGEMENT FOCUS **11-1 CREDIT CARD DATA THEFT**

In May of 2005, hackers broke into a database operated by CardSystems Solutions and stole data on as many as 40 million MasterCard, Visa, and other credit cards users. The breach was the largest single data leak in history, affecting one out of every seven credit cards issued in the United States.

The breach was discovered by MasterCard's fraud department, who tracked the stolen data to its source at CardSystems, a third-party processor of credit data. CardSystems processes more than $15 billion in credit card transactions per year, mostly for over 100,000 small- and medium-sized businesses.

The intruder used the Internet to break into a server at a processing center in Tucson, Arizona, that hosted the credit card database. The intruder exploited a known security flaw in the server software.

SOURCE: Robert Lemos, "MasterCard Warns of Massive Credit-Card Breach," *SecurityFocus.com,* 2005-06-17; Paul F. Roberts "Major Card Vendors Stay Mum on Data Breach," *www.eweek.com,* June 20, 2005.

strophic. Natural (or man-made) *disasters* may occur that destroy host computers or large sections of the network. For example, hurricanes, fires, floods, earthquakes, mudslides, tornadoes, or terrorist attacks can destroy large parts of the buildings and networks in their path.

Intrusion (or unauthorized access) refers primarily to confidentiality, but also to integrity, as someone with unauthorized access may change important data. Intrusion is often viewed as external attackers gaining access to organizational data files and resources from across the Internet. However, almost half of all intrusion incidents involve employees. Intrusion may have only minor effects. A curious intruder may simply explore the system, gaining knowledge that has little value. A more serious intruder may be a competitor bent on industrial espionage who could attempt to gain access to information on products under development, or the details and price of a bid on a large contract, or a thief trying to steal customer credit card numbers or information to carry out identity theft. Worse still, the intruder could change files to commit fraud or theft or could destroy information to injure the organization.

Network Controls

Developing a secure network means developing *controls.* Controls are mechanisms that reduce or eliminate the threats to network security. There are three types of controls that *prevent, detect,* and *correct* whatever might happen to the organization because of threats facing its computer-based systems.

Preventive controls mitigate or stop a person from acting or an event from occurring. For example, a password can prevent illegal entry into the system, or a set of second circuits can prevent the network from crashing. Preventative controls also act as a deterrent by discouraging or restraining someone from acting or proceeding because of fear or doubt. For example, a guard or a security lock on a door may deter an attempt to gain illegal entry.

Detective controls reveal or discover unwanted events. For example, software that looks for illegal network entry or enabling can detect these problems. They also document an event, a situation, or a trespass, providing evidence for subsequent action against the individuals or organizations involved or enabling corrective action to be taken. For example, the same software that detects the problem must report it immediately so that someone or some automated process can take corrective action.

Corrective controls remedy an unwanted event or a trespass. Either computer programs or humans verify and check data to correct errors or fix a security breach so it will not recur in the future. They also can recover from network errors or disasters. For example, software can recover and restart the communication circuits automatically when there is a data communication failure.

The remainder of this chapter will discuss the various controls that can be used to prevent, detect, and correct threats. We also present a control spreadsheet and risk analysis methodology for identifying the threats and their associated controls. The control spreadsheet provides a network manager with a good view of the current threats and any controls that are in place to mitigate the occurrence of threats.

Nonetheless, it is important to remember that it is not enough just to establish a series of controls; someone or some department must be accountable for the control and security of the network. This includes being responsible for developing controls, monitoring their operation, and determining when they need to be updated or replaced.

Controls must be reviewed periodically to be sure that they are still useful and must be verified and tested. Verifying ensures that the control is present, and testing determines whether the control is working as originally specified.

It is also important to recognize that there may be occasions in which a person must temporarily override a control, for instance when the network or one of its software or hardware subsystems is not operating properly. Such overrides should be tightly controlled, and there should be a formal procedure to document this occurrence should it happen.

RISK ASSESSMENT

One key step in developing a secure network is to conduct a *risk assessment.* This assigns levels of risk to various threats to network security by comparing the nature of the threats to the controls designed to reduce them. It is done by developing a control spreadsheet and then rating the importance of each risk. This section provides a brief summary of the risk assessment process.[3]

Develop a Control Spreadsheet

To be sure that the data communication network and microcomputer workstations have the necessary controls and that these controls offer adequate protection, it is best to build a

[3]CERT has developed a detailed risk assessment procedure called OCTAVE[SM], which is available at www.cert.org/octave.

TECHNICAL FOCUS

11-1 Basic Control Principles of a Secure Network

- The less complex a control, the better.
- A control's cost should be equivalent to the identified risk. It often is not possible to ascertain the expected loss, so this is a subjective judgment in many cases.
- Preventing a security incident is always preferable to detecting and correcting it after it occurs.
- An adequate system of internal controls is one that provides "just enough" security to protect the network, taking into account both the risks and costs of the controls.
- Automated controls (computer-driven) always are more reliable than manual controls that depend on human interaction.
- Controls should apply to everyone, not just a few select individuals.
- When a control has an override mechanism, make sure that it is documented and that the override procedure has its own controls to avoid misuse.
- Institute the various security levels in an organization on the basis of "need to know." If you do not need to know, you do not need to access the network or the data.
- The control documentation should be confidential.
- Names, uses, and locations of network components should not be publicly available.
- Controls must be sufficient to ensure that the network can be audited, which usually means keeping historical transaction records.
- When designing controls, assume that you are operating in a hostile environment.
- Always convey an image of high security by providing education and training.
- Make sure the controls provide the proper separation of duties. This applies especially to those who design and install the controls and those who are responsible for everyday use and monitoring.
- It is desirable to implement entrapment controls in networks to identify attackers who gain illegal access.
- When a control fails, the network should default to a condition in which everyone is denied access. A period of failure is when the network is most vulnerable.
- Controls should still work even when only one part of a network fails. For example, if a backbone network fails, all local area networks connected to it should still be operational, with their own independent controls providing protection.
- Don't forget the LAN. Security and disaster recovery planning has traditionally focused on host mainframe computers and WANs. However, LANs now play an increasingly important role in most organizations but are often overlooked by central site network managers.
- Always assume your opponent is smarter than you.
- Always have insurance as the last resort should all controls fail.

control spreadsheet (Figure 11.3). Threats to the network are listed across the top, organized by business continuity (disruption, destruction, disaster) and intrusion, and the network assets down the side. The center of the spreadsheet incorporates all the controls that *currently* are in the network. This will become the benchmark upon which to base future security reviews.

Threats / Assets (with Priority)	Disruption, Destruction, Disaster					Intrusion		
	Fire	Flood	Power Loss	Circuit Failure	Virus	External Intruder	Internal Intruder	Eaves-drop
(92) Mail server								
(90) Web server								
(90) DNS server								
(50) Computers on sixth floor								
(50) Sixth-floor LAN circuits								
(80) Building A backbone								
(70) Router in building A								
(30) Network software								
(100) Client database								
(100) Financial database								

FIGURE 11.3 Sample control spreadsheet with some assets and threats. DNS = Domain Name Service; LAN = local area network.

Assets The first step is to identify the assets on the network. An *asset* is something of value and can be either hardware, software, data, or applications. Probably the most important asset on a network is the organization's data. For example, suppose someone destroyed a mainframe worth $10 million. The mainframe could be replaced simply by buying a new one. It would be expensive, but the problem would be solved in a few weeks. Now suppose someone destroyed all the student records at your university so that no one knows what courses anyone had taken or their grades. The cost would far exceed the cost of replacing a $10 million computer. The lawsuits alone would easily exceed $10 million, and the cost of staff to find and reenter paper records would be enormous and certainly would take more than a few weeks. Figure 11.4 summarizes some typical assets.

An important type of asset is the *mission-critical application,* which is an information system that is critical to the survival of the organization. It is an application that cannot be permitted to fail, and if it does fail, the network staff drops everything else to fix it. For example, for an Internet bank that has no brick-and-mortar branches, the Web site is a mission-critical application. If the Web site crashes, the bank cannot conduct business with its customers. Mission-critical applications are usually clearly identified, so their importance is not overlooked.

Once you have a list of assets, they should be evaluated based on their importance. There will rarely be enough time and money to protect all assets completely, so it is important to focus the organization's attention on the most important ones. Prioritizing asset importance is a business decision, not a technology decision, so it is critical that senior business managers be involved in this process.

Hardware	• Servers, such as mail servers, Web servers, DNS servers, DHCP servers, and LAN file servers • Client computers • Devices such as hubs, switches, and routers
Circuits	• Locally operated circuits such as LANs and backbones • Contracted circuits such as MAN and WAN circuits • Internet access circuits
Network software	• Server operating systems and system settings • Application software such as mail server and Web server software
Client software	• Operating systems and system settings • Application software such as word processors
Organizational data	• Databases with organizational records
Mission-critical applications	• For example, for an Internet bank, its Web site is mission-critical

FIGURE 11.4 Types of assets. DNS = Domain Name Service; DHCP = Dynamic Host Control Protocol; LAN = local area network; MAN = metropolitan area network; WAN = wide area network.

Threats A *threat* to the data communication network is any potential adverse occurrence that can do harm, interrupt the systems using the network, or cause a monetary loss to the organization. While threats may be listed in generic terms (e.g., theft of data, destruction of data), it is better to be specific and use actual data from the organization being assessed (e.g., theft of customer credit card numbers, destruction of the inventory database).

Once the threats are identified they can be ranked according to their probability of occurrence and the likely cost if the threat occurs. Figure 11.5 summarizes the most common threats and their likelihood of occurring, plus a typical cost estimate, based on several surveys (primarily the 2005 CSI/FBI Computer Crime and Security Survey, and the 2005 Secret Service/CSO/CERT E-Crime Survey). The actual probability of a threat to your organization and its costs depend upon your business. An Internet bank, for example, is more likely to be a target of fraud and to suffer a higher cost if it occurs than a restaurant with a simple Web site. Nonetheless, Figure 11.5 provides some general guidance.

From Figure 11.5 you can see that the most likely event is a virus infection, suffered by more than 80 percent of organizations each year. The average cost to clean up a virus that slips through the security system and infects an average number of computers is about $100,000 per virus. Depending upon your background, this was probably not the first security threat that came to mind; most people first think about unknown attackers breaking into a network across the Internet. This does happen, too; unauthorized access by an external hacker is experienced by about 42 percent of all organizations each year, with some experiencing an act of sabotage or vandalism. The average cost to recover after these attacks is $150,000.

Interestingly, companies suffer intrusion by their own employees about as often as by outsiders, although the dollar loss is usually less unless fraud or theft of information is

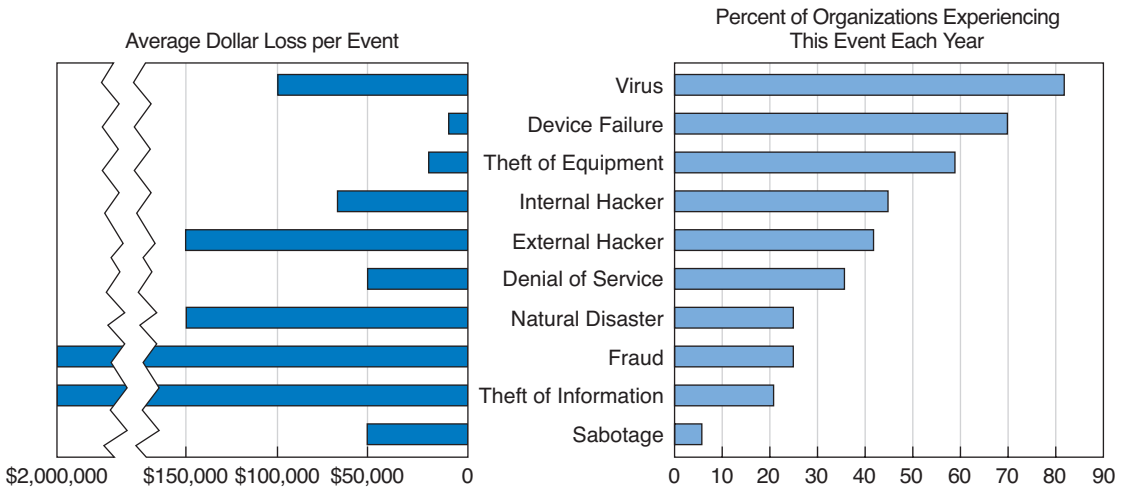


FIGURE 11.5 Likelihood and costs of common risks.

SOURCE: CSI/FBI Computer Crime and Security Survey, 2005 and SS/CSO/CERT E-Crime Survey, 2005.

involved. While few organizations experience fraud or theft of information from internal or external attackers, the cost to recover afterward can be very high, both in dollar cost and bad publicity. Several major companies have had their networks broken into and have had proprietary information such as customer credit card numbers stolen. Winning back customers whose credit card information was stolen can be an even greater challenge than fixing the security breach.

You will also see that device failure and computer equipment theft are common problems but usually result in low dollar losses compared to other security violations. Natural disasters (e.g., fire, flood) are also fairly common, experienced by 20 percent of organizations each year, and result in high dollar losses (about $150,000 per event).

Denial of service attacks, in which someone external to your organization blocks access to your networks, are also common (35 percent) and somewhat costly ($50,000 per event). Even temporary disruptions in service that cause no data loss can have significant costs. Estimating the cost of denial of service is very organization-specific; the cost of disruptions to a company that does a lot of e-commerce through a Web site is often measured in the millions.

Amazon.com, for example, has revenues of more than $10 million per hour, so if its Web site were unavailable for an hour or even part of an hour it would cost millions of dollars in lost revenue. Companies that do no e-commerce over the Web would have lower costs, but recent surveys suggest losses of $100,000–200,000 per hour are not uncommon for major disruptions of service. Even the disruption of a single LAN has cost implications; surveys suggest that most businesses estimate the cost of lost work at $1,000–5,000 per hour.

There are two "big picture" messages from Figure 11.5. First, the most common threat that has a fairly high cost is viruses. In fact, if we look at the relative probabilities of the different threats, we can see that the threats to business continuity (e.g., virus, device failure, theft of equipment, or natural disaster) have a greater chance of occurring than in-

trusion. Nonetheless, given the cost of fraud and theft of information, even a single event can have significant impact.[4]

The second important message is that the threat of intrusion from the outside intruder coming at you over the Internet has increased. For the past 25 years, more organizations reported encountering security breaches caused by employees than by outsiders. This has been true ever since the early 1980s when the FBI first began keeping computer crime statistics and security firms began conducting surveys of computer crime. However, in recent years, the number of external attacks has increased at a much greater rate while the number of internal attacks has stayed relatively constant. Even though some of this may be due to better internal security and better communications with employees to prevent security problems, much of it is simply due to an increase in activity by external attackers and the global reach of the Internet. Today, for the first time ever, external attackers pose as great a risk as internal employees.

Identify and Document the Controls

Once the specific assets and threats have been identified, you can begin working on the network *controls,* which mitigate or stop a threat, or protect an asset. During this step, you identify the existing controls and list them in the cell for each asset and threat.

Begin by considering the asset and the specific threat, and then describe each control that prevents, detects, or corrects that threat. The description of the control (and its role) is placed in a numerical list, and the control's number is placed in the cell. For example, assume 24 controls have been identified as being in use. Each one is described, named, and numbered consecutively. The numbered list of controls has no ranking attached to it: the first control is number 1 just because it is the first control identified.

Figure 11.6 shows a partially completed spreadsheet. The assets and their priority are listed as rows, with threats as columns. Each cell lists one or more controls that protect one asset against one threat. For example, in the first row, the mail server is currently protected from a fire threat by a Halon fire suppression system, and there is a disaster recovery plan in place. The placement of the mail server above ground level protects against flood, and the disaster recovery plan helps here too.

Evaluate the Network's Security

The last step in using a control spreadsheet is to evaluate the adequacy of the existing controls and the resulting degree of risk associated with each threat. Based on this assessment, priorities can be established to determine which threats must be addressed immediately. Assessment is done by reviewing each set of controls as it relates to each threat and network component. The objective of this step is to answer the specific question: are the controls adequate to effectively prevent, detect, and correct this specific threat?

The assessment can be done by the network manager, but it is better done by a team of experts chosen for their in-depth knowledge about the network and environment being

[4]We should point out, though, that the losses associated with computer fraud are small compared with other sources of fraud.

Threats / Assets (with Priority)	Disruption, Destruction, Disaster					Intrusion		
	Fire	Flood	Power Loss	Circuit Failure	Virus	External Intruder	Internal Intruder	Eavesdrop
(92) Mail server	1, 2	1, 3	4	5, 6	7, 8	9, 10, 11	9, 10	
(90) Web server	1, 2	1, 3	4	5, 6	7, 8	9, 10, 11	9, 10	
(90) DNS server	1, 2	1, 3	4	5, 6	7, 8	9, 10, 11	9, 10	
(50) Computers on sixth floor	1, 2	1, 3			7, 8	10, 11	10	
(50) Sixth-floor LAN circuits	1, 2	1, 3						
(80) Building A backbone	1, 2	1, 3		6				
(70) Router in building A	1, 2	1, 3				9	9	
(30) Network software					7, 8	9, 10, 11	9, 10	
(100) Client database					7, 8	9, 10, 11	9, 10	
(100) Financial database					7, 8	9, 10, 11	9, 10	

Controls

1. Disaster recovery plan
2. Halon fire system in server room; sprinklers in rest of building
3. Not on or below ground level
4. Uninterruptable power supply (UPS) on all major network servers
5. Contract guarantees from interexchange carriers
6. Extra backbone fiber cable laid in different conduits
7. Virus checking software present on the network
8. Extensive user training about viruses and reminders in monthly newsletter
9. Strong password software
10. Extensive user training about password security and reminders in monthly newsletter
11. Application-layer firewall

FIGURE 11.6 Sample control spreadsheet with some assets, threats, and controls. DNS = Domain Name Service; LAN = local area network.

reviewed. This team, known as the *Delphi team,* is composed of three to nine key people. Key managers should be team members because they deal with both the long-term and day-to-day operational aspects of the network. More important, their participation means the final results can be implemented quickly, without further justification, because they make the final decisions affecting the network.

BUSINESS CONTINUITY PLANNING

Business continuity means that the organization's data and applications will continue to operate even in the face of disruption, destruction, or disaster. A business continuity plan has two major parts: the development of controls that will prevent these events from hav-

MANAGEMENT FOCUS **11-2 ATTACK OF THE AUDITORS**

Security has become a major issue over the past few years. With the passage of HIPPA and the Sarbanes-Oxley Act, more and more regulations are addressing security. It takes years for most organizations to become compliant, because the rules are vague and there are many ways to meet the requirements.

"If you've implemented commonsense security, you're probably already in compliance from an IT standpoint," says Kim Keanini, Chief Technology Officer of nCricle, a security software firm. "Compliance from an auditing standpoint, however, is something else." Auditors require documentation. It is no longer sufficient to put key network controls in place; now you have to provide documented proof that a control is working, which usually requires event logs of transactions and thwarted attacks.

When it comes to security, Bill Randal, MIS Director of Red Robin Restaurants, can't stress the importance of documentation enough. "It's what the auditors are really looking for," he says. "They're not IT folks, so they're looking for documented processes they can track. At the start of our [security] compliance project, we literally stopped all other projects for other three weeks while we documented every security and auditing process we had in place."

Software vendors are scrambling to ensure than their security software not only performs the functions it is designed to do, but also to improve its ability to provide documentation for auditors.

SOURCE: Oliver Rist, "Attack of the Auditors," *InfoWorld*, March 21, 2005, pp. 34-40.

ing a major impact on the organization, and a disaster recovery plan that will enable the organization to recover if a disaster occurs. In this section, we discuss controls that attempt to prevent, detect, and correct these threats.[5]

Preventing Disruption, Destruction, and Disaster

The key principle in preventing disruption, destruction, and disaster—or at least reducing their impact—is *redundancy.* Redundant hardware that automatically recognizes failure and intervenes to replace the failed component can mask a failure that would otherwise result in a service disruption. Redundancy can be built into any network component.

Using Redundant Hardware The most common example of redundancy is an *uninterruptable power supply* (UPS), a separate battery-operated power supply unit that can supply power for minutes (or even hours) in the event of a power loss. The UPS is installed on the network server so that in the event of a power failure, the server continues to operate until power is restored or until the UPS battery becomes low. When the UPS battery begins to weaken, many UPSs can send a special message to the server enabling it to start a normal shutdown.

[5]There are many good business continuity planning sites such as www.disasterrecoveryworld.com.

You can also buy a special-purpose *fault-tolerant server* that contains many redundant components to prevent failure. One common strategy, *disk mirroring,* utilizes a second redundant disk for every disk on the server. Every data item written to the primary disk is automatically duplicated on the mirrored disk. If the primary disk fails, the mirrored disk automatically takes over, with no observable effects on any network applications. This concept can be extended to include disk controllers (called *disk duplexing*), so that even if the disk controller fails, the server continues to operate.

Redundancy can be applied to other network components as well. For example, additional client computers, circuits, or devices (e.g., routers, switches, multiplexers) can be installed to ensure that the network remains operational should any of these components fail. The last control point is the network personnel and equipment in the network control center, which oversees network management and operation, the test equipment, reports, documentation, and the like.

Preventing Natural Disaster Disasters are different. In this case, an entire site can be destroyed. Even if redundant components are present, often the scope of the loss is such that returning the network to operation is extremely difficult. The best solution is to have a completely redundant network that duplicates every network component but is in a separate location.

Generally speaking, preventing disasters is difficult. How do you prevent an earthquake? There are, however, some practical, commonsense steps that can be taken to prevent the full impact of disasters from affecting business continuity. The most fundamental principle is to store critical data in at least two different locations (ideally in different parts of the country or even different countries). By having critical data in two very different locations, you can eliminate the chance that a huge natural disaster can destroy all of your data in one stroke.

Other steps depend upon the type of disaster to be prevented. For example, to reduce the risks due to flood, key network components should not be located near rivers or oceans, or in the basement or ground floor of buildings. To reduce the risks from fire, Halon fire suppression systems should be installed in rooms containing important network equipment. To reduce the risks from terrorist attacks, the location of key network components should be kept secret and protected by security guards.

Preventing Theft In some cases, the disruption is intentional. One often-overlooked security risk is theft. Computers and network devices are commonplace items that are relatively expensive. There is a good secondhand market for such equipment, making them valuable to steal. Several industry sources estimate that about $1 billion is lost each year to theft of computers and related equipment. Any security plan should include an evaluation of ways to prevent someone from stealing equipment.

Preventing Viruses Special attention also must be paid to preventing computer *viruses.* Some are harmless and just cause nuisance messages, but others are serious such as destroying data. In most cases, disruptions or the destruction of data are local and affect only a small number of components (although the failure of one WAN or BN circuit may affect many computers). Such disruptions are usually fairly easy to deal with; the failed component is replaced or the virus is removed and the network continues to operate.

MANAGEMENT FOCUS **11-3 RECOVERING FROM KATRINA**

As Hurricane Katrina swept over New Orleans, Ochsner Hospital lost two of its three backup power generators knocking out air conditioning in the 95-degree heat. Fans were brought out to cool patients, but temperatures inside critical computer and networking equipment reached 150 degrees. Kurt Induni, the hospital's network manager, shut down part of the network and the mainframe with its critical patient records system to ensure they survived the storm. The hospital returned to paper-based record keeping, but Induni managed to keep e-mail alive, which became critical when the telephone system failed and a main fiber line was cut. E-mail through the hospital's T-3 line into Baton Rouge became the only reliable means of communication. After the storm, the mainframe was turned back on and the patient records were updated.

While Ochsner Hospital remained open, Kindred Hospital was forced to evacuate patients (under military protection from looters and snipers). The patients' files, all electronic, were simply transferred over the network to other hospitals with no worry about lost records, X-rays, CT scans, and such.

In contrast, the Louisiana court system learned a hard lesson. The court system is administered by each individual parish (i.e., county) and not every parish had a disaster recovery plan or even backups of key documents–many parishes still use old paper files that were destroyed by the storm. "We've got people in jails all over the state right now that have no paperwork and we have no way to offer them any kind of means for adjudication," says Freddie Manit, CIO for the Louisiana Ninth Judicial District Court. No paperwork means no prosecution, even for felons with long records, so many prisoners will simply be released. Sometimes losing data is not the worst thing that can happen.

SOURCE: Phil Hochmuth, "Weathering Katrina," *NetworkWorld*, September 19, 2005, pp. 1, 20; and M. K. McGee, "Storm Shows Benefits, Failures of Technology," *Informationweek*, September 15, 2005, p. 34.

Most viruses attach themselves to other programs or to special parts on disks. As those files execute or are accessed, the virus spreads. *Macro viruses,* viruses that are contained in documents e-mails, or spreadsheet files, can spread when an infected file is simply opened. Some viruses change their appearances as they spread, making detection more difficult.

A *worm* is special type of virus that spreads itself without human intervention. Many viruses attach themselves to a file and require a person to copy the file, but a worm copies itself from computer to computer. Worms spread when they install themselves on a computer and then send copies of themselves to other computers, sometimes by e-mail, sometimes via security holes in software. (Security holes are described later in this chapter.)

The best way to prevent the spread of viruses is to not copy or download files of unknown origin, or at least to check every file you do copy or download. Many antivirus software packages are available to check disks and files to ensure that they are virus-free. Always check all files for viruses before using them (even those from friends!). Researchers estimate that 10 new viruses are developed every day, so it is important to frequently update the virus information files that are provided by the antivirus software.

Preventing Denial-of-Service Attacks Another special case is the *denial-of-service attack (DoS).* With a DoS attack, an attacker attempts to disrupt the network by

flooding it with messages so that the network cannot process messages from normal users. The simplest approach is to flood a Web server, mail server, and so on with incoming messages. The server attempts to respond to these, but there are so many messages that it cannot.

One might expect that it would be possible to filter messages from one source IP so that if one user floods the network, the messages from this person can be filtered out before they reach the Web server being targeted. This could work, but most attackers use tools that enable them to put false source IP addresses on the incoming messages so that it is difficult to recognize a message as a real message or a DoS message.

A *distributed denial-of-service attack (DDoS)* is even more disruptive. With a DDoS attack, the attacker breaks into and takes control of many computers on the Internet (often several hundred to several thousand) and plants software on them called a *DDoS agent* (or sometimes a zombie or a bot). The attacker then uses software called a *DDoS handler* (sometimes called a botnet) to control the agents. The handler issues instructions to the computers under the attacker's control, which simultaneously begin sending messages to the target site. In this way, the target is deluged with messages from many different sources, making it harder to identify the DoS messages and greatly increasing the number of messages hitting the target (see Figure 11.7).

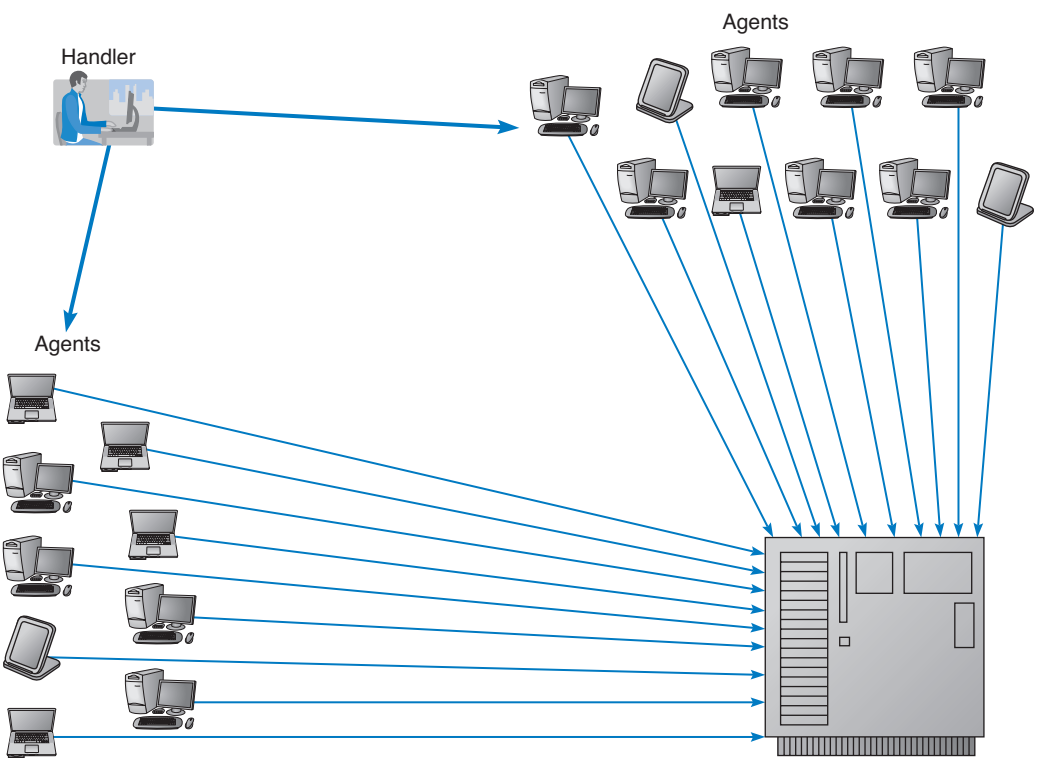


FIGURE 11.7 A distributed denial-of-service attack.

There are several approaches to preventing DoS and DDoS attacks from affecting the network. The first is to configure the main router that connects your network to the Internet (or the firewall, which will be discussed later in this chapter) to verify that the source address of all incoming messages is in a valid address range for that connection (called *traffic filtering*). For example, if an incoming message has a source address from inside your network, then it is obviously a false address. This ensures that only messages with valid addresses are permitted into the network, although it requires more processing in the router and thus slows incoming traffic.

A second approach is to configure the main router (or firewall) to limit the number of incoming packets that could be DoS/DDoS attack packets that it allows to enter the network, regardless of their source (called *traffic limiting*). Technical Focus box 11-2 describes some of the types of DoS/DDoS attacks and the packets used. Such packets have the same content as legitimate packets that should be permitted into the network. It is a flood of such packets that indicates a DoS/DDoS attack, so by discarding packets over a certain number that arrive each second, one can reduce the impact of the attack. The disadvantage is that during an attack, some valid packets from regular customers will be discarded so they will be unable to reach your network. Thus the network will continue to operate, but some customer packets (e.g., Web requests, e-mails) will be lost.

A third and more sophisticated approach is to use a special-purpose security device, called a *traffic anomaly detector*, that is installed in front of the main router (or firewall) to perform *traffic analysis*. This device monitors normal traffic patterns and learns what normal traffic looks like. Most DoS/DDoS attacks target a specific server or device so when the anomaly detector recognizes a sudden burst of abnormally high traffic destined for a specific server or device, it quarantines those incoming packets but allows normal traffic to flow through into the network. This results in minimal impact to the network as a whole. The anomaly detector re-routes the quarantined packets to a *traffic anomaly analyzer* (see Figure 11.8). The anomaly analyzer examines the quarantined traffic, attempts to recognize valid source addresses and "normal" traffic, and selects which of the quarantined packets to release into the network. The detector can also inform the router owned by the ISP that is sending the traffic into the organization's network to re-route the suspect traffic to the anomaly analyzer, thus avoiding the main circuit leading into the organization. This process is never perfect, but is significantly better than the other approaches.

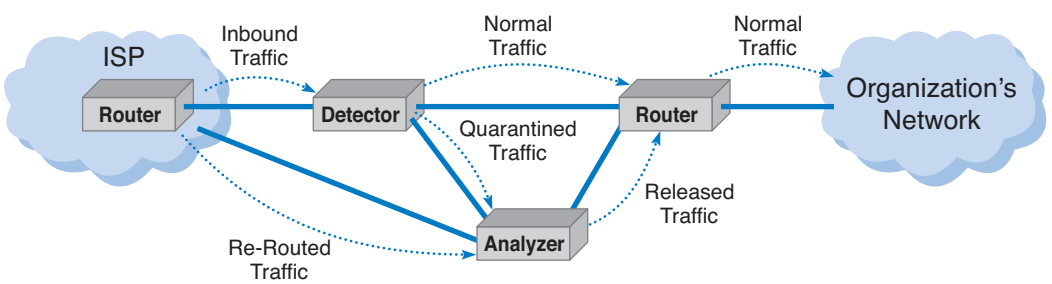


FIGURE 11.8 Traffic analysis reduces the impact of denial of service attacks.

TECHNICAL FOCUS

11-2 INSIDE A DoS ATTACK

A DoS attack typically involves the misuse of standard TCP/IP protocols or connection processes so that the target for the DoS attack responds in a way designed to create maximum trouble. Five common types of attacks include:

- **ICMP Attacks:** The network is flooded with ICMP echo requests (i.e., pings) that have a broadcast destination address and a faked source address of the intended target. Because it is a broadcast message, every computer on the network responds to the faked source address so that the target is overwhelmed by responses. Because there are often dozens of computers in the same broadcast domain, each message generates dozens of messages at the target.
- **UDP Attacks:** This attack is similar to an ICMP attack except that it uses UDP echo requests instead of ICMP echo requests.
- **TCP SYN Floods:** The target is swamped with repeated SYN requests to establish a TCP connection, but when the target responds (usually to a faked source address) there is no response. The target continues to allocate TCP control blocks, expects each of the requests to be completed, and gradually runs out of memory.
- **UNIX Process Table Attacks:** This is similar to a TCP SYN flood, but instead of TCP SYN packets, the target is swamped by UNIX open connection requests that are never completed. The target allocates open connections and gradually runs out of memory.
- **Finger of Death Attacks:** This is similar to the TCP SYN flood, but instead the target is swamped by finger requests that are never disconnected.
- **DNS Recursion Attacks:** The attacker sends DNS requests to DNS servers (often within the target's network), but spoofs the from address so the requests appear to come from the target computer which is overwhelmed by DNS responses. DNS responses are larger packets than ICMP, UDP, or SYN responses so the effects can be stronger

SOURCE: "Web Site Security and Denial of Service Protection," www.nwfusion.com.

Another possibility under discussion by the Internet community as a whole is to require Internet service providers (ISPs) to verify that all incoming messages they receive from their customers have valid source IP addresses. This would prevent the use of faked IP addresses and enable users to easily filter out DoS messages from a given address. It would make it virtually impossible for a DoS attack to succeed, and much harder for a DDoS attack to succeed. Because small- to medium-sized businesses often have poor security and become the unwilling accomplices in DDoS attacks, many ISPs are beginning to impose security restrictions on them, such as requiring firewalls to prevent unauthorized access (firewalls are discussed later in this chapter).

Detecting Disruption, Destruction, and Disaster

Major problems need to be recognized quickly. As we will discuss in Chapter 12, one function of network management software is to alert network managers to network problems so these can be corrected. Some intelligent network servers even can be programmed to send an alarm to a pager if necessary. The organization's disaster procedures should include notifying the network managers as soon as possible when a problem occurs.

MANAGEMENT FOCUS **11-4 A DDoS Attack Takes Down StormPay**

StormPay, an e-commerce payment processor, was taken down for several days by a DDoS attack. StormPay is used by many Web hosting companies to process payments. The attack occurred after StormPay froze the account of a controversial service that pays users to view Internet ads. The service was under investigation by the FBI and SEC for running a Ponzi scheme.

The attack was a DNS recursion attack (see Technical Focus 11-2) that sent bogus DNS requests and DNS responses into StormPay's network. About 120,000 computers (zombies) were used in the attack which flooded StormPay's network with 6 gigabits of data per second. After StormPay took action to bring its site back online, the attack switched to the ISPs that host StormPay's sites, which again took StormPay's site offline.

SOURCE: Rich Miller, "Payment Gateway StormPay Battling Sustained DDOS Attack, Netcraft.com, February 10, 2006; Jon Swartz, "Increasing Web Attacks Disrupt Commerce," *USAToday.com*, February 26, 2006.

Detecting minor disruptions and destruction can be more difficult. A network drive may develop bad spots that remain unnoticed unless the drive is routinely checked. Likewise, a network cable may be partially damaged by hungry squirrels, resulting in intermittent problems. These types of problems require ongoing monitoring. The network should routinely log fault information to enable network managers to recognize minor service problems before they become major ones. In addition, there should be a clear procedure by which network users can report problems.

Correcting Disruption, Destruction, and Disaster

Disaster Recovery Plan A critical element in correcting problems is the *disaster recovery plan,* which should address various levels of response to a number of possible disasters and should provide for partial or complete recovery of all data, application software, network components, and physical facilities. A complete disaster recovery plan covering all these areas is beyond the scope of this text. Figure 11.9 provides a summary of many key issues. A good example of a disaster recovery plan is MIT's business continuity plan at web.mit.edu/security/www/pubplan.htm.

The most important elements of the disaster recovery plan are *backup and recovery controls* that enable the organization to recover its data and restart its application software should some portion of the network fail. The simplest approach is to make backup copies of all organizational data and software routinely and to store these backup copies off-site. Most organizations make daily backups of all critical information, with less important information (e.g., e-mail files) backed up weekly. Backups used to be done on tapes that were physically shipped to an off-site location, but more and more, companies are using their WAN connections to transfer data to remote locations (it's faster and cheaper than moving tapes). Backups should always be encrypted (encryption is discussed later in the chapter) to ensure that no unauthorized users can access them.

Continuous data protection (CDP) is another option that firms are using in addition to or instead of regular backups. With CDP, copies of all data and transactions on

Elements of a Disaster Recovery Plan

A good disaster recovery plan should include:

- The name of the decision-making manager who is in charge of the disaster recovery operation. A second manager should be indicated in case the first manager is unavailable.
- Staff assignments and responsibilities during the disaster
- A preestablished list of priorities that states what is to be fixed first
- Location of alternative facilities operated by the company or a professional disaster recovery firm and procedures for switching operations to those facilities using backups of data and software
- Recovery procedures for the data communication facilities (backbone network, metropolitan area network, wide area network, and local area network), servers, and application systems. This includes information on the location of circuits and devices, whom to contact for information, and the support that can be expected from vendors, along with the name and telephone number of the person at each vendor to contact.
- Action to be taken in case of partial damage or threats such as bomb threats, fire, water or electrical damage, sabotage, civil disorders, and vendor failures
- Manual processes to be used until the network is functional
- Prodecures to ensure adequate updating and testing of the disaster recovery plan
- Storage of the data, software, and the disaster recovery plan itself in a safe area where they cannot be destroyed by a catastrophe. This area must be accessible, however, to those who need to use the plan.

FIGURE 11.9 Elements of a disaster recovery plan.

selected servers are written to CDP servers as the transaction occurs. CDP is more flexible than traditional backups that take snapshots of data at specific times, or disk mirroring, that duplicates the contents of a disk from second to second. CDP enables data to be stored miles from the originating server and time-stamps all transactions to enable organizations to restore data to any specific point in time. For example, suppose a virus brings down a server at 2:45 P.M. The network manager can restore the server to the state it was in at 2:30 P.M. and simply resume operations as though the virus had not hit.

Backups and CDP ensure that important data is safe, but they do not guarantee the data can be used. The disaster recovery plan should include a documented and tested approach to recovery. The recovery plan should have specific goals for different types of disasters. For example, if the main database server was destroyed, how long should it take the organization to have the software and data back in operation by using the backups? Conversely, if the main data center was completely destroyed, how long should it take? The answers to these questions have very different implications for costs. Having a spare network server or a server with extra capacity that can be used in the event of the loss of the primary server is one thing. Having a spare data center ready to operate within 12 hours (for example) is an entirely different proposition.

While many organizations have a disaster recovery plan, only a few test their plans. A *disaster recovery drill* is much like a fire drill in that it tests the disaster recovery plan and provides staff the opportunity to practice little-used skills to see what works and what doesn't work before a disaster happens and the staff must use the plan for real. Without

regular disaster recovery drills, the only time a plan is tested is when it must be used. For example, when an island-wide blackout shut down all power in Bermuda, the backup generator in the British Caymanian Insurance office automatically took over and kept the company operating. However, the key-card security system, which was not on the generator, shut down, locking out all employees and forcing them to spend the day at the beach. No one had thought about the security system and the plan had not been tested.

Disaster Recovery Outsourcing Most large organizations have a two-level disaster recovery plan. When they build networks they build enough capacity and have enough spare equipment to recover from a minor disaster such as loss of a major server or portion of the network (if any such disaster can truly be called minor). This is the first level. Building a network that has sufficient capacity to quickly recover from a major disaster such as the loss of an entire data center is beyond the resources of most firms.

MANAGEMENT FOCUS

11-5 Disaster Recovery Hits Home

"The building is on fire" were the first words she said as I answered the phone. It was just before noon and one of my students had called me from her office on the top floor of the business school at the University of Georgia. The roofing contractor had just started what would turn out to be the worst fire in the region in more than 20 years although we didn't know it then. I had enough time to gather up the really important things from my office on the ground floor (memorabilia, awards, and pictures from 10 years in academia) when the fire alarm went off. I didn't bother with the computer; all the files were backed up off-site.

Ten hours, 100 firefighters, and 1.5 million gallons of water later, the fire was out. Then our work began. The fire had completely destroyed the top floor of the building, including my 20-computer networking lab. Water had severely damaged the rest of the building, including my office, which, I learned later, had been flooded by almost 2 feet of water at the height of the fire. My computer, and virtually all the computers in the building, were damaged by the water and unusable.

My personal files were unaffected by the loss of the computer in my office; I simply used the backups and continued working—after making new backups and giving them to a friend to store at his house. The Web server I managed had been backed up to another server on the opposite side of campus 2 days before (on its usual weekly backup cycle), so we had lost only 2 days' worth of changes. In less than 24 hours, our Web site was operational; I had our server's files mounted on the university library's Web server and redirected the university's DNS server to route traffic from our old server address to our new temporary home.

Unfortunately, the rest of our network did not fare as well. Our primary Web server had been backed up to tape the night before and while the tapes were stored off-site, the tape drive was not; the tape drive was destroyed and no one else on campus had one that could read our tapes; it took 5 days to get a replacement and reestablish the Web site. Within 30 days we were operating from temporary offices with a new network, and 90 percent of the office computers and their data had been successfully recovered.

Living through a fire changes a person. I'm more careful now about backing up my files, and I move ever so much more quickly when a fire alarm sounds. But I still can't get used to the rust that is slowly growing on my "recovered" computer.

Therefore, most large organizations rely on professional disaster recovery firms to provide this second-level support for major disasters.

Many large firms outsource their disaster recovery efforts by hiring *disaster recovery firms* that provide a wide range of services. At the simplest, disaster recovery firms provide secure storage for backups. Full services include a complete networked data center that clients can use when they experience a disaster. Once a company declares a disaster, the disaster recovery firm immediately begins recovery operations using the backups stored on-site and can have the organization's entire data network back in operation on the disaster recovery firm's computer systems within hours. Full services are not cheap, but compared to the potential millions of dollars that can be lost per day from the inability to access critical data and application systems, these systems quickly pay for themselves in time of disaster.

INTRUSION PREVENTION

Intrusion is the second main type of security problem and the one that tends to receive the most attention. No one wants an intruder breaking into their network.

There are four types of intruders who attempt to gain unauthorized access to computer networks. The first are casual intruders who have only a limited knowledge of computer security. They simply cruise along the Internet trying to access any computer they come across. Their unsophisticated techniques are the equivalent of trying doorknobs, and, until recently, only those networks that left their front doors unlocked were at risk. Unfortunately, there are now a variety of hacking tools available on the Internet that enable even novices to launch sophisticated intrusion attempts. Novice attackers that use such tools are sometimes called *script kiddies.*

The second type of intruders are experts in security, but their motivation is the thrill of the hunt. They break into computer networks because they enjoy the challenge and enjoy showing off for friends or embarrassing the network owners. These intruders are called *hackers* and often have a strong philosophy against ownership of data and software. Most cause little damage and make little attempt to profit from their exploits, but those that do can cause major problems. Hackers that cause damage are often called *crackers.*

The third type of intruder is the most dangerous. They are professional hackers who break into corporate or government computers for specific purposes, such as espionage, fraud, or intentional destruction. The U.S. Department of Defense (DoD), which routinely monitors attacks against U.S. military targets, has until recently concluded that most attacks are individuals or small groups of hackers in the first two categories. While some of their attacks have been embarrassing (e.g., defacement of some military and intelligence Web sites), there have been no serious security risks. However, in the late 1990s the DoD noticed a small but growing set of intentional attacks that they classify as exercises, exploratory attacks designed to test the effectiveness of certain software attack weapons. Therefore, they established an *information warfare* program and a new organization responsible for coordinating the defense of military networks under the U.S. Space Command.

The fourth type of intruder is also very dangerous. These are organization employees who have legitimate access to the network, but who gain access to information they are not authorized to use. This information could be used for their own personnel gain,

sold to competitors, or fraudulently changed to give the employee extra income. Many security break-ins are caused by this type of intruder.

Preventing Intrusion

The key principle in preventing intrusion is to be *proactive.* This means routinely testing your security systems before an intruder does. Many steps can be taken to prevent intrusion or unauthorized access to organizational data and networks, but no network is completely safe. The best rule for high security is to do what the military does: do not keep extremely sensitive data online. Data that need special security are stored in computers isolated from other networks.

In the same way that a disaster recovery plan is critical to controlling risks due to disruption, destruction, and disaster, a *security policy* is critical to controlling risk due to intrusion. The security policy should clearly define the important assets to be safeguarded and the important controls needed to do that. It should have a section devoted to what employees should and should not do. It should contain a clear plan for routinely training employees—particularly end-users with little computer expertise—on key security rules and a clear plan for routinely testing and improving the security controls in place (Figure 11.10). A good set of examples and templates is available at www.sans.org/resources/policies.

In the sections below, we focus on the three main aspects of preventing intrusion: securing the network perimeter, securing the interior of the network, and authenticating users to make sure only valid users are allowed into network resources. Unfortunately, too

Elements of a Security Policy

A good security policy should include:

- The name of the decision-making manager who is in charge of security
- An incident reporting system and a rapid-response team to respond to security breaches in progress
- A risk assessment with priorities as to which assets are most important
- Effective controls placed at all major access points into the network to prevent or deter access by external agents
- Effective controls placed within the network to ensure that internal users cannot exceed their authorized access
- Use of minimum number of controls possible to reduce management time and to provide the least inconvenience to users
- An acceptable use policy that explains to users what they can and cannot do, including guidelines for accessing others' accounts, password security, e-mail rules, and so on
- A procedure for monitoring changes to important network components (e.g., routers, DNS servers)
- A plan to routinely train users regarding security policies and build awareness of security risks
- A plan to routinely test and update all security controls that includes monitoring of popular press and vendor reports of security holes
- An annual audit and review of the security practices

FIGURE 11.10 Elements of a security policy.

often companies focus on the first and the last and forget the middle—or do all three, but fail to implement controls to detect security breaches. Such networks are said to have *candy security*: "crunchy outside, soft and chewy inside."

Securing the Network Perimeter There are three basic access points into most organizational networks: from LANs inside the organization, from dial-up access through a modem, and from the Internet. Recent surveys suggest that the most common access point used by attackers is the Internet (about 90 percent of respondents to the CSI/FBI Computer Crime and Security Survey reported experiencing an attack from the Internet), followed by internal LANs (30 percent) and dial-up (20 percent). Naturally, most attacks from the Internet were launched by those external to the firm, while most internal attacks were launched by employees.

One important element of preventing unauthorized users from accessing an internal LAN is *physical security*: preventing outsiders from gaining access into the organization's offices, server room, or network equipment facilities. Both main and remote physical facilities should be secured adequately and have the proper controls. Good security requires implementing the proper access controls so that only authorized personnel can enter closed areas where servers and network equipment are located or access the network. The network components themselves also have a level of physical security. Computers can have locks on their power switches or passwords that disable the screen and keyboard.

In the previous section we discussed the importance of locating backups and servers at separate (off-site) locations. Some companies have also argued that by having many servers in different locations you can reduce your risk and improve business continuity. Does having many servers disperse risk, or does it increase the points of vulnerability? A clear disaster recovery plan with an off-site backup and server facility can disperse risk, like distributed server systems. Distributed servers offer many more physical vulnerabilities to an attacker: more machines to guard, upgrade, patch, and defend. Many times these dispersed machines are all part of the same logical domain, which means that breaking into one of them often can give the attacker access to the resources of the others. It is our feeling that a well backed-up, centralized data center can be made inherently more secure than a proliferated base of servers.

Proper security education, background checks, and the implementation of error and fraud controls are also very important. In many cases, the simplest means to gain access is to become employed as a janitor and access the network at night. In some ways this is easier than the previous methods because the intruder only has to insert a listening device or computer into the organization's network to record messages. Two areas are vulnerable to this type of unauthorized access: network cabling and network devices.

Network cables are the easiest target for eavesdropping because they often run long distances and usually are not regularly checked for tampering. The cables owned by the organization and installed within its facility are usually the first choice for *eavesdropping*. It is 100 times easier to tap a local cable than it is to tap an interexchange channel because it is extremely difficult to identify the specific circuits belonging to any one organization in a highly multiplexed switched interexchange circuit operated by a common carrier. Local cables should be secured behind walls and above ceilings, and telephone equipment and switching rooms (wiring closets) should be locked and their doors equipped with alarms. The primary goal is to control physical access by employees or vendors to the

TECHNICAL FOCUS

11-3 Data Security Requires Physical Security

The general consensus is that if someone can physically get to your server for some period of time, then all of your information on the computer (except perhaps strongly encrypted data) is available to the attacker.

With a Windows server, the attacker simply boots the computer from the CD drive with a Knoppix version of Linux. (Knoppix is Linux on a CD.) If the computer won't boot from the CD, the attacker simply changes the BIOS to make it boot from the CD. Knoppix finds all the drivers for the specific computer and gives you a Linux desktop that can fully read all of the NTFS or FAT32 files.

But what about Windows password access? Nothing to it. Knoppix completely bypasses it. The attacker can then read, copy, or transmit any of the files on the Windows machine. Similar attacks are also possible on a Linux or Unix server, but they are slightly more difficult.

connector cables and modems. This includes restricting their access to the wiring closets in which all the communication wires and cables are connected.

Certain types of cable can impair or increase security by making eavesdropping easier or more difficult. Obviously, any wireless network is at extreme risk for eavesdropping because anyone in the area of the transmission can easily install devices to monitor the radio or infrared signals. Conversely, fiber-optic cables are harder to tap, thus increasing security. Some companies offer armored cable that is virtually impossible to cut without special tools. Other cables have built-in alarm systems. The U.S. Air Force, for example, uses pressurized cables that are filled with gas. If the cable is cut, the gas escapes, pressure drops, and an alarm is sounded.

Network devices such as controllers, hubs, and bridges should be secured in a locked wiring closet. As discussed in Chapter 6, all messages within a given local area network are actually received by all computers on the LAN although they only process those messages addressed to them. It is rather simple to install a *sniffer program* that records all messages received for later (unauthorized) analysis. A computer with a sniffer program could then be plugged into an unattended hub or bridge to eavesdrop on all message traffic. A *secure hub* makes this type of eavesdropping more difficult by requiring a special authorization code to be entered before new computers can be added.

Dial-in security is important for any organization that permits staff members to access its network via modems. Some dial-up modem controls include changing the modem telephone numbers periodically and keeping telephone numbers confidential. In recent years, *automatic number identification* (ANI) has been used. The network manager can specify several telephone numbers authorized to access each account. When a user successfully logs on to an account, the source of the incoming phone call is identified using ANI and if it is one of the authorized numbers, the login is accepted; otherwise, the host computer or communications server disconnects the call. ANI does not work for users who frequently travel (e.g., sales representatives) because they often call from hotel rooms and have no knowledge of telephone numbers in advance.

With the increasing use of the Internet, it becomes important to prevent intrusion to the network from attackers on other networks. The obvious solution is to disconnect any

computer or network containing confidential information from the Internet, which is often not a practical solution. In many cases, organizations are disconnecting unneeded applications to improve security. For example, a Web server often does not need e-mail, so network managers often remove e-mail software to reduce the number of entry points that an attacker has into the network.

A *firewall* is commonly used to secure an organization's Internet connection. A firewall is a router or special-purpose computer that examines packets flowing into and out of a network and restricts access to the organization's network. The network is designed so that a firewall is placed on every network connection between the organization and the Internet (Figure 11.11). No access is permitted except through the firewall. Some firewalls have the ability to detect and prevent denial-of-service attacks, as well as unauthorized access attempts. Two commonly used types of firewalls are packet-level firewalls and application-level firewalls.

A *packet-level firewall* examines the source and destination address of every network packet that passes through it. It only allows packets into or out of the organization's networks that have acceptable source and destination addresses. In general, the addresses are examined only at the transport layer (TCP port id) and network layer (IP address). Each packet is examined individually, so the firewall has no knowledge of what the user is attempting to do. It simply chooses to permit entry or exit based on the contents of the packet itself. This type of firewall is the simplest and least secure because it does not monitor the contents of the packets or why they are being transmitted, and typically does not log the packets for later analysis.

Some packet-level firewalls are vulnerable to *IP spoofing*. The goal of an intruder using IP spoofing is to send packets to a target computer requesting certain privileges be granted to some users (e.g., setting up a new account for the intruder or changing the access permission or password for an existing account). Such a message would not be accepted by the target computer unless it can be fooled into believing that the request is genuine.

IP spoofing is done by changing the source address on incoming packets from their real IP address to an IP address inside the organization's network. Seeing a valid internal address, the firewall lets the packets through to their destination. The destination computer believes the packets are from a valid internal user and processes them. Typically, IP spoofing is more complex than this because such changes often require a dialogue between the computers. Since the target computer believes it is talking to an internal computer, it directs its messages to the internal computer, not to the intruder's computer. Intruders therefore have to guess at the nature and timing of these messages so that they can generate more spoofed messages that appear to be responses to the target computer's messages. In practice, expert hackers have enough knowledge to have a reasonable chance of getting this right.

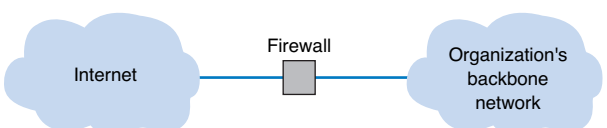


FIGURE 11.11 Using a firewall to protect networks.

Many firewalls have had their security strengthened as IP spoofing has become more common. For example, some firewalls automatically delete any packets arriving from the Internet that have internal source addresses. However, IP spoofing still remains a problem.

An *application-level firewall* acts as an intermediate host computer between the Internet and the rest of the organization's networks. These firewalls are generally more complicated to install and manage than packet-level ones, because they examine the contents of the application layer packet and search for known attacks (see security holes later in this chapter), as well as any rules programmed by the organization. Remember from Chapter 5 that TCP uses connection-oriented messaging in which a client first establishes a connection with a server before beginning to exchange data. Application-level firewalls use *stateful inspection*, which means that they monitor and record the status of each connection and can use this information in making decisions about what packets to discard as security threats. In some cases, special programming code must be written for the firewall to permit the use of application software unique to the organization.

Many application-level firewalls prohibit external users from uploading executable files. In this way, intruders (or authorized users) cannot modify any software unless they have physical access to the firewall. Some refuse changes to their software unless it is done by the vendor. Others also actively monitor their own software and automatically disable outside connections if they detect any changes.

Most firewalls today also perform *network address translation (NAT)*—translating between one set of private addresses inside a network and a set of public addresses outside

TECHNICAL FOCUS

11-4 How Packet-Level Firewalls Work

Remember from Chapter 5 that TCP/IP networks such as the Internet use TCP packets and IP packets. IP packets provide the source and destination IP addresses. TCP packets provide application layer port numbers that indicate the application layer software to which the packet should be sent. For example, the Web uses port 80, telnet uses port 23, and SMTP uses port 25.

Packet-level firewalls enable the network administrator to establish a series of rules in an *Access Control List* that define what packets should be allowed to pass through and what packets should be deleted. Suppose, for example, that the organization had a Web server with an IP address of 128.192.55.55 that was for internal use only. The administrator could define a rule on the firewall that instructed the firewall to delete any packet from the Internet that listed 128.192.55.55 as a destination. In this case, the firewall simply needs to examine the destination address.

Suppose, however, the organization had a Web server (128.192.44.44) and a mail server (128.192.44.45) that were intended to be available to Internet users. However, to prevent anyone on the Internet from making changes to the server, the organization wants to prevent any telnet, FTP, or other similar packets from reaching the servers. In this case, the administrator could define a rule that instructed the firewall to permit TCP packets with a destination port address of 80, a destination IP address of 128.192.44.44, and any source address to pass through (see Figure 11.12). A second rule could permit packets with a port of 25 and any source address to reach the mail server. A third rule would instruct the firewall to delete any packets with any other port number and destination IP address. If some one then tried to telnet to the Web server, the firewall would discard the packet.

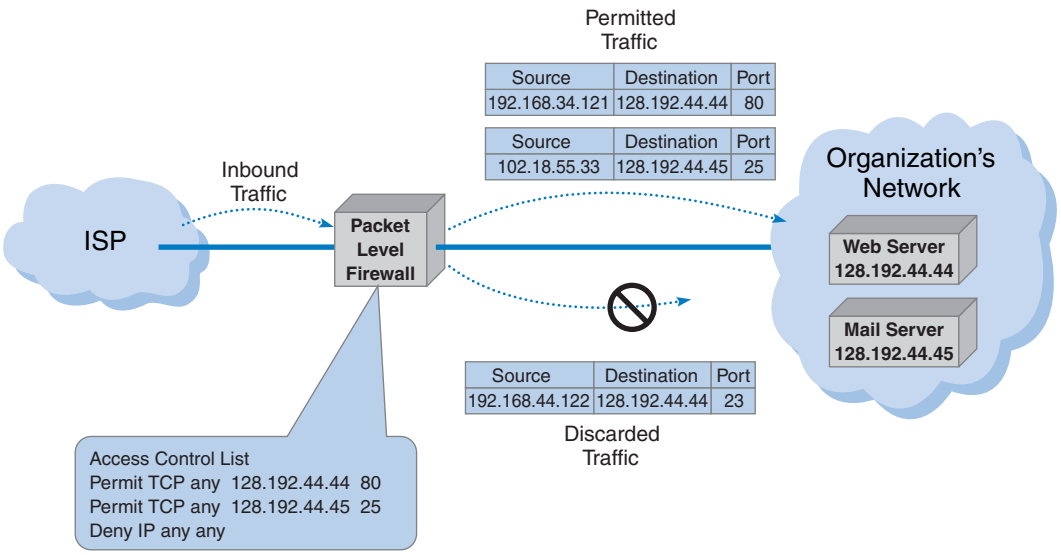


FIGURE 11.12 How packet level firewalls work.

the network. NAT is transparent in that no computer notices that it is being done. While NAT can be done for several reasons, the primary use today is for security.

The *NAT proxy server* uses an address table to translate the private IP addresses used inside the organization into proxy IP addresses used on the Internet. When a computer inside the organization accesses a computer on the Internet, the proxy server changes the source IP address in the outgoing IP packet to its own address. It also sets the source port number in the TCP packet to a unique number that it uses as an index into its address table to find the IP address of the actual sending computer in the organization's internal network. When the external computer responds to the request, it addresses the message to the proxy server's IP address. The proxy server receives the incoming message, and after ensuring the packet should be permitted inside, changes the destination IP address to the private IP address of the internal computer and changes the TCP port number to the correct port number before transmitting it on the internal network.

This way systems outside the organization never see the actual internal IP addresses, and thus they think there is only one computer on the internal network. Some organizations also increase security by using illegal internal addresses. For example, if the organization has been assigned the Internet 128.192.55.X address domain, the NAT proxy server would be assigned an address such as 128.192.55.1. Internal computers, however, would *not* be assigned addresses in the 128.192.55.X subnet. Instead, they would be assigned unauthorized Internet addresses such as 10.3.3.55 (addresses in the 10.X.X.X domain are not assigned to organizations but instead are reserved for use by private intranets). Since these internal addresses are never used on the Internet but are always converted by the proxy server, this poses no problems for the users. Even if attackers discover the actual internal IP address, it would be impossible for them to reach the internal

address from the Internet because the addresses could not be used to reach the organization's computers.[6]

NAT proxy servers work very well and are replacing traditional firewalls. They do, however, slow message transfer between internal networks and the Internet. They also require a separate DNS server for use by external users on the Internet and a separate internal DNS server for use on the internal networks. Many organizations use internal firewalls to prevent employees in one part of an organization from access to resources in a different part.

Many organizations use layers of NAT proxy servers and packet-level and application-level firewalls (Figure 11.13). Packet-level firewalls are used as an initial screen from the Internet into a network devoted solely to servers intended to provide public access (e.g., Web servers, public DNS servers). This network is sometimes called the *DMZ* (demilitarized zone) because it contains the organization's servers but does not provide complete security for them. This packet-level firewall will permit Web requests and similar access to the DMZ network servers but will deny FTP access to these servers from the Internet because no one except internal users should have the right to modify the servers. Each major portion of the organization's internal networks has its own proxy server to grant (or deny) access based on rules established by that part of the organization.

This figure also shows how a packet sent by a client computer inside one of the internal networks protected by a proxy server would flow through the network. The packet created by the client has the client's false source address and the source port number of the process on the client that generated the packet (an HTTP packet going to a Web server, as you can tell from the destination port address of 80). When the packet reaches the proxy server, the proxy server changes the source address on the IP packet to its own address and changes the source port number to an index it will use to identify the client computer's address and port number. The destination address and port number are unchanged. The proxy server then sends the packet on its way to the destination. When the destination Web server responds to this packet, it will respond using the proxy server's address and port number. When the proxy server receives the incoming packets it will use the destination port number to identify what IP address and port number to use inside the internal network, change the inbound packet's destination and port number, and send it into the internal network so it reaches the client computer.

Securing the Interior Even with physical security, firewalls, and NAT, a network may not be safe because of *security holes.* A security hole is simply a bug that permits unauthorized access. Many commonly used operating systems have major security holes well known to potential intruders. Many security holes have been documented and "patches" are available from vendors to fix them, but network managers may be unaware of all the holes or simply forget to update their systems with new patches regularly.

A complete discussion of security holes is beyond the scope of this book. Many security holes are highly technical; for example, sending a message designed to overflow a memory buffer, thereby placing a short command into a very specific memory area that

[6]Most routers and firewalls manufactured by Linksys (a manufacturer of networking equipment for home and small office use owned by Cisco) use NAT. Rather than setting the internal address to 10.x.x.x, Linksys sets them to 192.168.1.x, which is another subnet reserved for private intranets. If you have Linksys equipment with a NAT firewall, your internal IP address is likely to be 192.168.1.100.

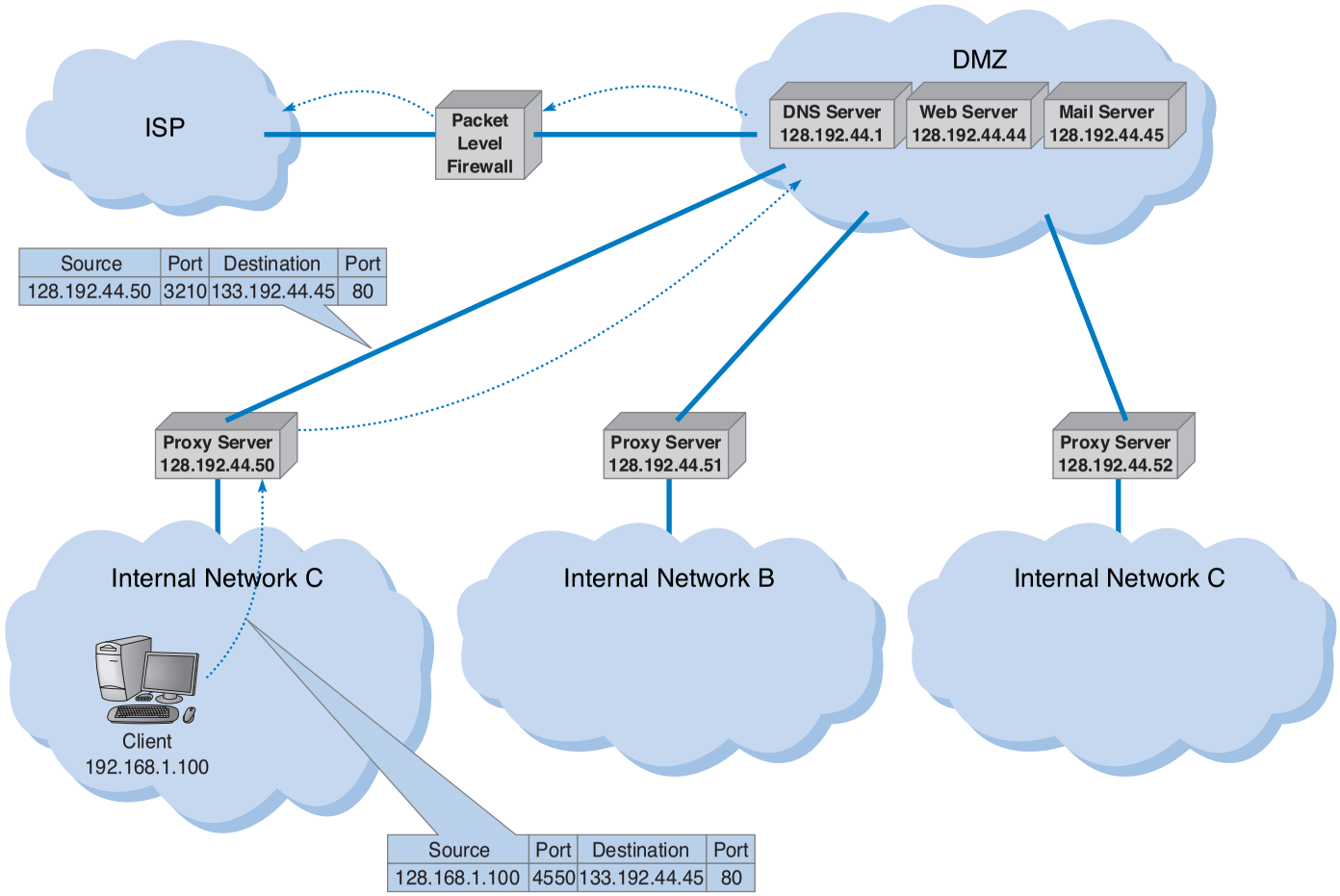


FIGURE 11.13 A typical network design using firewalls and proxy servers.

performs some function. Others are rather simple, but not obvious. For example, the attacker sends a message that lists the server's address as both the sender and the destination, so the server repeatedly sends messages to itself until it crashes.

Once a security hole is discovered, it is quickly circulated through the Internet. The race begins between hackers and security teams; hackers share their discovery with other hackers and security teams share the discovery with other security teams. CERT is the central clearinghouse for major Internet-related security holes, so the CERT team quickly responds to reports of new security problems and posts alerts and advisories on the Web and e-mails them to those who subscribe to its service. The developer of the software with the security hole usually works quickly to fix the security hole and produces a *patch* that corrects the hole. This patch is then shared with customers so they can download and apply it to their systems to prevent hackers from exploiting the hole to break in. The problem is that many network managers do not routinely respond to such security threats and immediately download and install the patch. Often it takes many months for patches to be distributed to most sites.[7] Do you regularly install all the Windows or Mac updates on your computer?

MANAGEMENT FOCUS

11-6 PATCH AND PRAY

In January 2003, the Slammer worm infected 90 percent of all vulnerable computers on the Internet in just 10 minutes after it was released. Slammer was stopped by ISPs that blocked port 1434, the one Slammer used to propagate itself.

When Slammer subsided, talk focused on patching. Those looking to cast blame cried a familiar refrain: if everyone had just patched their systems, Slammer wouldn't have happened. But that's not true; patching no longer works. Software today is massive (Windows contains over 45 million lines of code) and the rate of sloppy coding (10 to 20 errors per 1,000 lines of code) has led to thousands of vulnerabilities. There are simply too many patches coming too quickly.

Patch writing is usually assigned to entry-level programmers. They fix problems in a race with hackers trying to exploit them. From this patch factory comes a poorly written product that can break as much as it fixes. One patch, for example, worked fine for everyone except the unlucky users who happened to have a certain computer with outdated drivers, which the patch crashed. Sometimes if you just apply patches, you get nailed.

There are two emerging and opposite patch philosophies: either patch more or patch less. Patch-more adherents believe patching isn't the problem, but that manual patching is. Vendors in the patch-more school have created patch management software that automates the process of finding, downloading, and applying patches.

The patch-less school argues that historically only 2 percent of vulnerabilities have resulted in attacks. Therefore, most patches aren't worth applying; they're at best superfluous and, at worst, *add* significant additional risk. Instead, you should improve your security policy (e.g., turn off ports such as 1434 that aren't needed) and pay third parties to determine which patches are really necessary.

SOURCE: "Patch and Pray," www.csoonline.com/read/081303/patch.html, August 2003.

[7]For an example of one CERT advisory posted about problems with the most common DNS server software used on the Internet, see www.cert.org/advisories/CA-2001-02.html. The history in this advisory shows that it took about 8 months for the patch for the previous advisory in this family (issued in November 1999) to be installed on most DNS servers around the world. This site also has histories of more recent advisories.

Other security holes are not really holes but simply policies adopted by computer vendors that open the door for security problems, such as computer systems that come with a variety of preinstalled user accounts. These accounts and their initial passwords are well documented and known to all potential attackers. Network managers sometimes forget to change the passwords on these well-known accounts thus enabling an attacker to slip in.

The American government requires certain levels of security in the operating systems and network operating systems it uses for certain applications. The minimum level of security is C2. Most major operating systems (e.g., Windows) provide at least C2. Most widely used systems are striving to meet the requirements of much higher security levels such as B2. Very few systems meet the highest levels of security (A1 and A2).

There has been a long running debate about whether the Windows operating system is less secure than other operating systems such as Linux. Every new attack on Windows systems ignites the debate; Windows detractors repeat "I told you so" while Windows defenders state that this happens mostly because Windows is the obvious system to attack, and because of the hostility of the Windows detractors themselves.

There is a critical difference in what applications can do in Windows and in Linux. Linux (and its ancestor Unix) was first written as a multi-user operating system in which different users had different rights. Only some users were system administrators and had the rights to access and make changes to the critical parts of the operating system. All other users were barred from doing so.

In contrast, Windows (and its ancestor DOS) was first written as an operating system for a single personal computer, an environment in which the user was in complete control of the computer and could do anything he or she liked. As a result, Windows applications regularly access and make changes to critical parts of the operating system. There are advantages to this. Windows applications can do many powerful things without the user needing to understand them. These applications can be very rich in features, and more important, they can appear to the user to be very friendly and easy to use. Everything appears to run "out-of-the-box" without modification. Windows has built these features into the core of their systems. Any major rewrite of Windows to prevent this would most likely cause significant incompatibilities with all applications designed to run under previous versions of Windows. To many, this would be a high price to pay for some unseen benefits called "security."

But there is a price for this friendliness. Hostile applications can easily take over the computer and literally do whatever they want without the user knowing. Simply put, there is a tradeoff between ease of use and security. Increasing needs for security demand more checks and restrictions, which translates into less friendliness and fewer features. It may very well be that there is an inherent and permanent contradiction between the ease of use of a system and its security.

One important tool in gaining unauthorized access is a *Trojan horse*. Trojans are remote access management consoles (sometimes called *rootkits*) that enable users to access a computer and manage it from afar. If you see free software that will enable you to control your computer from anywhere, be careful; the software may also permit an attacker to control your computer from anywhere! Trojans are more often concealed in other software that unsuspecting users download over the Internet (their name alludes to the original Trojan horse). Music and video files shared on Internet music sites are common

TECHNICAL FOCUS

11-5 Exploiting a Security Hole

In order to exploit a security hole, the hacker has to know it's there. So how does a hacker find out? It's simple in the era of automated tools.

First, the hacker has to find the servers on a network. The hacker could start by using network scanning software to systematically probe every IP address on a network to find all the servers on the network. At this point, the hacker has narrowed the potential targets to a few servers.

Second, the hacker needs to learn what services are available on each server. To do this, he or she could use port scanning software to systematically probe every TCP/IP port on a given server. This would reveal which ports are in use and thus what services the server offers. For example, if the server has software that responds to port 80, it is a Web server, while if it responds to port 25, it is a mail server.

Third, the hacker would begin to seek out the exact software and version number of the server software providing each service. For example, suppose the hacker decides to target mail servers. There are a variety of tools that can probe the mail server software, and based on how the server software responds to certain messages, determine which manufacturer and version number of software is being used.

Finally, once the hacker knows which package and version number the server is using, the hacker uses tools designed to exploit the known security holes in the software. For example, some older mail server software packages do not require users to authenticate themselves (e.g., by a userid and password) before accepting SMTP packets for the mail server to forward. In this case, the hacker could create SMTP packets with fake source addresses and use the server to flood the Internet with spam (i.e., junk mail). In another case, a certain version of a well-known e-commerce package enabled users to pass operating system commands to the server simply by including a UNIX pipe symbol (|) and the command to the name of a file name to be uploaded; when the system opened the uploaded file, it also executed the command attached to it.

carriers of Trojans. When the user downloads and plays a music file, it plays normally and the attached Trojan software silently installs a small program that enables the attacker to take complete control of the user's computer, so the user is unaware that anything bad has happened. The attacker then simply connects to the user's computer and has the same access and controls as the user. Many Trojans are completely undetectable by the very best antivirus software.

One of the first major Trojans was Back Orifice, which aggressively attacked Windows servers. Back Orifice gave the attacker the same functions as the administrator of the infected server, and then some: complete file and network control, device and registry access, with packet and application redirection. It was every administrator's worst nightmare, and every attacker's dream.

More recently, Trojans have morphed into tools such as MoSucker and Optix Pro. These attack consoles now have one-button clicks to disable firewalls, antivirus software, and any other defensive process that might be running on the victim's computer. The attacker can choose what port the Trojan runs on, what it is named, and when it runs. They can listen in to a computer's microphone or look through an attached camera—even if the device appears to be off. Figure 11.14 shows a menu from one Trojan

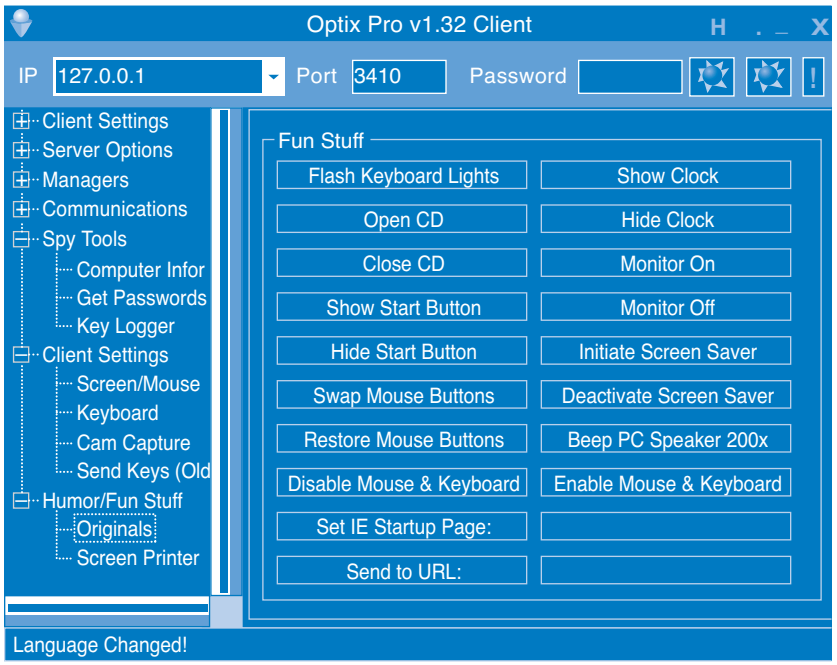


FIGURE 11.14 One menu on the control console for the Optix Pro Trojan.

that illustrates some of the "fun stuff" that an attacker can do, such as opening and closing the CD tray, beeping the speaker, or reversing the mouse buttons so that clicking on the left button actually sends a right click.

Not only have these tools become powerful, but they are also very easy to use—much easier to use than the necessary defensive countermeasures to protect oneself from them. And what does the near future hold for Trojans? We can easily envision Trojans that schedule themselves to run at, say 2:00 AM, choosing a random port, emailing the attacker that the machine is now "open for business" at port # NNNNN. The attackers can then step in, do whatever they want to do, run a script to erase most of their tracks, and then sign out and shut off the Trojan. Once the job is done, the Trojan could even erase itself from storage. Scary? Yes. And the future does not look better.

Spyware, adware, and DDoS agents are three types of Trojans. DDoS agents were discussed in the previous section. As the name suggests, spyware monitors what happens on the target computer. Spyware can record keystrokes that appear to be userids and passwords so the intruder can gain access to the user's account (e.g., bank accounts). Adware monitors user's actions and displays pop-up advertisements on the user's screen. For example, suppose you clicked on the Web site for an online retailer. Adware might pop-up a window for a competitor, or, worse still, redirect your browser to the competitor's Web site. Many anti-virus software package now routinely search for and remove spyware, adware, and other Trojans. Some firewall vendors are now adding anti-Trojan logic to their devices to block any transmissions from infected computers from entering or leaving their networks.

MANAGEMENT FOCUS

11-7 SONY'S SPYWARE

Sony BMG Entertainment, the music giant, included a spyware rootkit on audio CDs sold in the fall of 2005, including CDs by such artists as Celine Dion, Frank Sinatra, and Ricky Martin. The rootkit was automatically installed on any PC that played the infected CD. The rootkit was designed to track the behavior of users who might be illegally copying and distributing the music on the CD, with the goal of preventing illegal copies from being widely distributed.

Sony made two big mistakes. First, it failed to inform customers who purchased its CDs about the rootkit, so users unknowingly installed it. The rootkit used standard spyware techniques to conceal its existence to prevent users from discovering it. Second, Sony used a widely available rootkit, which meant that any knowledgeable user on the Internet could use the rootkit to take control of the infected computer. Several viruses have been written that exploit the rootkit and are now circulating on the Internet. The irony is that rootkit infringes on copyrights held by several open source projects, which means Sony was engaged in the very act it was trying to prevent: piracy.

When the rootkit was discovered, Sony was slow to apologize, slow to stop selling rootkit-infected CDs, and slow to help customers remove the rootkit. Several lawsuits have been filed in the United States and abroad seeking damages.

SOURCE: J.A. Halderman and E.W. Felton, "Lessons from the Sony CD DRM Episode," working paper, Princeton University, 2006; and "Sony Anti-Customer Technology Roundup and Time-Line,"*www.boingboing.net,* February 15, 2006.

One of the best ways to prevent intrusion is *encryption,* which is a means of disguising information by the use of mathematical rules known as *algorithms.*[8] Actually, *cryptography* is the more general and proper term. *En*cryption is the process of disguising information whereas *decryption* is the process of restoring it to readable form. When information is in readable form, it is called *plaintext*; when in encrypted form, it is called *ciphertext.* Encryption can be used to encrypt files on a computer or to encrypt communication between computers.[9]

There are two fundamentally different types of encryption: symmetric and asymmetric. With *symmetric encryption,* the key used to encrypt a message is the *same* as the one used to decrypt it. With *asymmetric encryption,* the key used to decrypt a message is *different* from the key used to encrypt it.

Symmetric encryption (also call single-key encryption) has two parts: the *algorithm* and the *key*, which personalizes the algorithm by making the transformation of data unique. Two pieces of identical information encrypted with the same algorithm but with different keys produce completely different ciphertexts. With symmetric encryption, the communicating parties must share the one key. If the algorithm is adequate and the key is kept secret, acquisition of the ciphertext by unauthorized personnel is of no consequence to the communicating parties.

Good encryption systems do not depend on keeping the algorithm secret. Only the keys need to be kept secret. The key is a relatively small numeric value (in terms of the

[8]For more information on cryptography, see the FAQ at www.rsasecurity.com.

[9]If you use Windows, you can encrypt files on your hard disk. Just use the Help facility and search on encryption to learn how.

MANAGEMENT FOCUS **11-8 TROJANS AT HOME**

It started with a routine phone call to technical support—one of our users had a software package that kept crashing. The network technician was sent to fix the problem but couldn't, so thoughts turned to a virus or Trojan. After an investigation, the security team found a remote FTP Trojan installed on the computer that was storing several gigabytes of cartoons and making them available across the Internet. The reason for crash was that the FTP server was an old version that was not compatible with the computer's operating system. The Trojan was removed and life went on.

Three months later the same problem occurred on a different computer. Because the previous Trojan had been logged, the network support staff quickly recognized it as a Trojan. The same hacker had returned, storing the same cartoons on a different computer. This triggered a complete investigation. All computers on our Business School network were scanned and we found 15 computers that contained the Trojan. We gathered forensic evidence to help identify the attacker (e.g., log files, registry entries) and filed an incident report with the University incident response team advising them to scan all computers on the university network immediately .

The next day, we found more computers containing the same FTP Trojan and the same cartoons. The attacker had come back overnight and taken control of more computers. This immediately escalated the problem. We cleaned some of the machines but left some available for use by the hacker to encourage him not to attack other computers. The network security manager replicated the software and used it to investigate how the Trojan worked. We determined that the software used a brute force attack to break the administrative password file on the standard image that we used in our computer labs. We changed the password and installed a security patch to our lab computer's standard configuration. We then upgraded all the lab computers and only then cleaned the remaining machines controlled by the attacker.

The attacker had also taken over many other computers on campus for the same purpose. With the forensic evidence that we and the university security incident response team had gathered, the case is now in court.

number of bits). The larger the key, the more secure the encryption because large "key space" protects the ciphertext against those who try to break it by *brute-force attacks*—which simply means trying every possible key.

There should be a large enough number of possible keys that an exhaustive brute-force attack would take inordinately long or would cost more than the value of the encrypted information.

Because the same key is used to encrypt and decrypt, symmetric encryption can cause problems with *key management*; keys must be shared among the senders and receivers very carefully. Before two computers in a network can communicate using encryption, both must have the same key. This means that both computers can then send and read any messages that use that key. Companies often do not want one company to be able to read messages they send to another company, so this means that there must be a separate key used for communication with each company. These keys must be recorded but kept secure so that they cannot be stolen. Because the algorithm is known publicly, the disclosure of the key means the total compromise of encrypted messages. Managing this system of keys can be challenging.

One commonly used symmetric encryption technique is the *Data Encryption Standard (DES)*, which was developed in the mid-1970s by the U.S. government in conjunction with IBM. DES is standardized by the National Institute of Standards and Technology (NIST). The most common form of DES uses a 56-bit key, which experts can break in less than a day (i.e., experts with the right tools can figure out what a message encrypted using DES says without knowing the key in less than 24 hours). DES is no longer recommended for data needing high security although some companies continue to use it for less important data.

Triple DES (3DES) is a newer standard that is harder to break. As the name suggests, it involves using DES three times, usually with three different keys to produce the encrypted text, which produces a stronger level of security because it has a total of 168 bits as the key (i.e., 3 times 56 bits). [10]

The NIST's new standard, called *Advanced Encryption Standard (AES)*, has replaced DES. AES has key sizes of 128, 192, and 256 bits. NIST estimates that, using the most advanced computers and techniques available today, it will require about 150 trillion years to crack AES by brute force. As computers and techniques improve, the time requirement will drop, but AES seems secure for the foreseeable future; the original DES lasted 20 years, so AES may have a similar life span.

Another commonly used symmetric encryption algorithm is *RC4,* developed by Ron Rivest of RSA Data Security, Inc. RC4 can use a key up to 256 bits long but most commonly uses a 40-bit key. It is faster to use than DES but suffers from the same problems from brute-force attacks: its 40-bit key can be broken by a determined attacker in a day or two.

Today, the United States government considers encryption to be a weapon and regulates its export in the same way it regulates the export of machine guns or bombs. Present rules prohibit the export of encryption techniques with keys longer than 64 bits without permission, although exports to Canada and the European Union are permitted, and American banks and Fortune 100 companies are now permitted to use more powerful encryption techniques in their foreign offices. This policy made sense when only American companies had the expertise to develop powerful encryption software. Today, however, many non-American companies are developing encryption software that is more powerful than American software that is limited only by these rules. Therefore, the American software industry is lobbying the government to change the rules so that they can successfully compete overseas.[11]

The most popular form of asymmetric encryption (also called *public key encryption*) is *RSA,* which was invented at MIT in 1977 by Rivest, Shamir, and Adleman, who founded RSA Data Security in 1982.[12] The patent expired in 2000, so many new

[10]There are several versions of 3DES. One version (called 3DES-EEE) simply encrypts the message three times with different keys as one would expect. Another version (3DES-EDE) encrypts with one key, decrypts with a second key (i.e., reverse encrypts), and then encrypts with a third key. There are other variants, as you can imagine.

[11]The rules have been changed several times in recent years, so for more recent information, see www.bxa.doc.gov/Encryption.

[12]Rivest, Shamir, and Adleman have traditionally been given credit as the original developers of public key encryption (based on theoretical work by Whitfield Diffie and Martin Hellman), but recently declassified material has revealed that public key encryption was actually first developed years earlier by Clifford Cocks based on theoretical work by James Ellis, both of whom were employees of a British spy agency.

TECHNICAL FOCUS

11-6 Open Source versus Closed Source Software

"A cryptographic system should still be secure if everything is known about it except its key. You should not base the security of your system upon its obscurity."—Auguste Kerckhoffs (1883).

Auguste Kerckhoffs was a Flemish cryptographer and linguist who studied military communications during the Franco-Prussian War. He observed that neither side could depend upon hiding their telegraph lines and equipment from the other side because the enemy would find the hidden telegraph lines and tap into the communications. One could not rely upon their system being obscure. In 1948, Claude Shannon of Bell Labs extended Kerckhoffs' Law when he said, "Always assume that the enemy knows your system." Cryptographers and military colleges teach Kerckhoffs' and Shannon's laws as fundamental rules in information security.

How does this apply to computer security? There are a few basics that we should understand first: programmers write their code in human-readable source code, which is then compiled to produce binary object code (i.e., zeros and ones); very few people can read binary code. For-profit developers do *not* release their source code when they sell software; they only release the binary object code. This *closed source* code is their proprietary "crown jewels," to be jealously guarded. In contrast, *open source* software is not-for-profit software in which the source code is provided along with the binary object code so that other developers can read the code and write new features or find and fix bugs.

So, does this mean that closed source is safer than open source because no one can see any bugs or security holes that might be hidden in the source code? No. With closed source, there is the temptation to use "security via obscurity." The history of security holes is that they become well known. Why? First, because there may be literally hundreds of people with access to the source code. Some of those people come and go. Some take the code with them. And some talk to others, who post it on the Internet.

And then there are the decompilers. A decompiler converts binary object code back into source code. Decompilers do not produce exact copies of the original source code, but they are getting better and better. With their use, attackers can better guess where the security holes are.

There is also a tendency within the closed source community to rely upon the source code being hidden as a line of defense. In effect, they drop their guard, falsely thinking that they are safe behind the obscurity of hidden code. The open source community has far more people able to examine the code than any closed source system. One of the tenets of the open source community is "No bug is too obscure or difficult for a million eyes."

Also, the motives of the developers are different. Open source coders generally do not write for profit. Closed source developers are inevitably writing for profit. With the profit motive comes more pressure to release software quickly to "beat the market." Rushing code to market is one of the surest ways of releasing flawed code. This pressure does not exist in the open source world since no one is going to make much money on it anyway.

Can there be secure closed source software? Yes. But the developers must be committed to security from the very beginning of development. By most reasonable measures, open source software has been and continues to be more secure than closed source software. This is what Auguste Kerckhoffs would have predicted.

companies have entered the market and public key software has dropped in price. The RSA technique forms the basis for today's *public key infrastructure (PKI)*.

Public key encryption is inherently different from symmetric single-key systems like DES. Because public key encryption is asymmetric, there are two keys. One key

(called the *public key*) is used to encrypt the message and a second, very different *private key* is used to decrypt the message. Keys are often 512 bits or 1,024 bits in length.

Public key systems are based on one-way functions. Even though you originally know both the contents of your message and the public encryption key, once it is encrypted by the one-way function, the message cannot be decrypted without the private key. One-way functions, which are relatively easy to calculate in one direction, are impossible to "uncalculate" in the reverse direction. Public key encryption is one of the most secure encryption techniques available, excluding special encryption techniques developed by national security agencies.

Public key encryption greatly reduces the key management problem. Each user has its public key that is used to encrypt messages sent to it. These public keys are widely publicized (e.g., listed in a telephone book-style directory)—that's why they're called "public" keys. In addition, each user has a private key that decrypts only the messages that were encrypted by its public key. This private key is kept secret (that's why it's called the "private" key). The net result is that if two parties wish to communicate with one another, there is no need to exchange keys beforehand. Each knows the other's public key from the listing in a public directory and can communicate encrypted information immediately. The key management problem is reduced to the on-site protection of the private key.

Figure 11.15 illustrates how this process works. All public keys are published in a directory. When Organization A wants to send an encrypted message to Organization B, it looks through the directory to find its public key. It then encrypts the message using B's public key. This encrypted message is then sent through the network to Organization B, which decrypts the message using its private key.

Public key encryption also permits the use of *digital signatures* through a process of *authentication.* When one user sends a message to another, it is difficult to legally prove who actually sent the message. Legal proof is important in many communications, such as bank transfers and buy/sell orders in currency and stock trading, which normally require legal signatures. Public key encryption algorithms are *invertable*, meaning that text encrypted with either key can be decrypted by the other. Normally, we encrypt with the public key and decrypt with the private key. However, it is possible to do the inverse: encrypt with the private key and decrypt with the public key. Since the private key is secret, only the real user could use it to encrypt a message. Thus, a digital signature or authentication sequence is used as a legal signature on many financial transactions. This signature is usually the name of the signing party plus other *key-contents* such as unique information from the message (e.g., date, time, or dollar amount). This signature and the other key-contents are encrypted by the sender using the private key. The receiver uses the sender's public key to decrypt the signature block and compares the result to the name and other key contents in the rest of the message to ensure a match.

Figure 11.16 illustrates how authentication can be combined with public encryption to provide a secure and authenticated transmission with a digital signature. The plaintext message is first encrypted using Organization A's private key and then encrypted using Organization's B public key. It is then transmitted to B. Organization B first decrypts the message using its private key. It sees that part of the message (the key-contents) is still in cyphertext, indicating it is an authenticated message. B then decrypts the key-contents part of the message using A's public key to produce the plaintext message. Since only A has the private key that matches A's public key, B can safely assume that A sent the message.

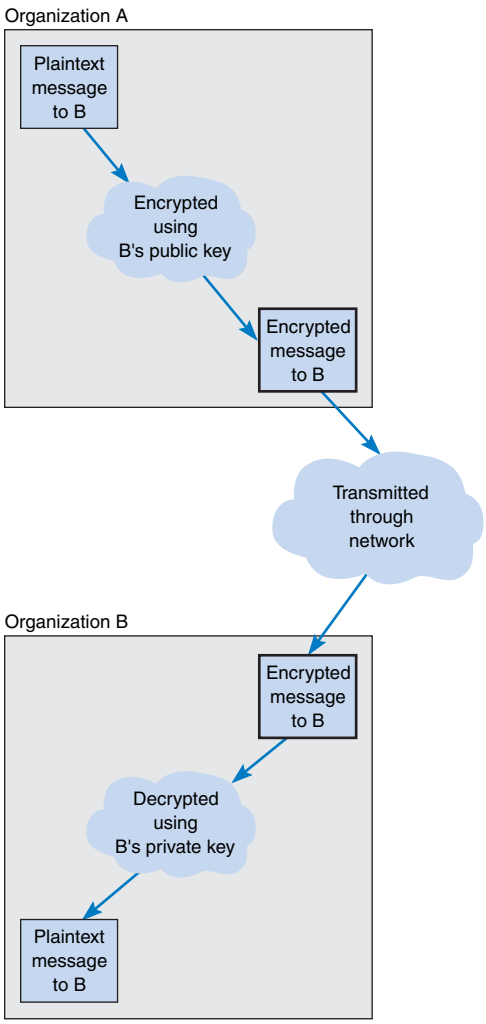


FIGURE 11.15 Secure transmission with public key encryption.

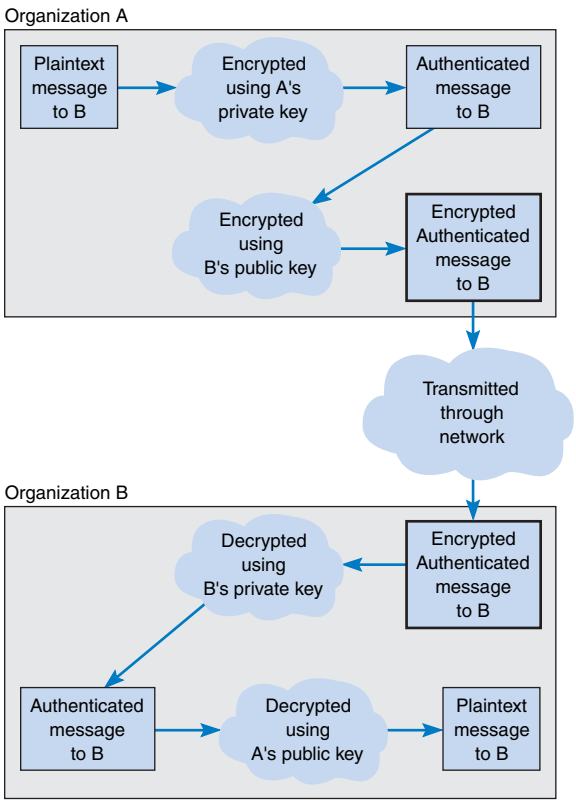


FIGURE 11.16 Authenticated and secure transmission with public key encryption.

The only problem with this approach lies in ensuring that the person or organization who sent the document with the correct private key is actually the person or organization they claim to be. Anyone can post a public key on the Internet, so there is no way of knowing for sure who they actually are. For example, it would be possible for someone to create a Web site and claim to be "Organization A" when in fact they are really someone else.

This is where the Internet's public key infrastructure (PKI) becomes important.[13] The PKI is a set of hardware, software, organizations, and polices designed to make pub-

[13]For more on the PKI, go to www.ietf.org and search on PKI.

lic key encryption work on the Internet. PKI begins with a *certificate authority (CA)*, which is a trusted organization that can vouch for the authenticity of the person or organization using authentication (e.g., VeriSign). A person wanting to use a CA registers with the CA and must provide some proof of identity. There are several levels of certification, ranging from a simple confirmation from a valid e-mail address to a complete police-style background check with an in-person interview. The CA issues a digital *certificate* that is the requestor's public key encrypted using the CA's private key as proof of identity. This certificate is then attached to the user's e-mail or Web transactions, in addition to the authentication information. The receiver then verifies the certificate by decrypting it with the CA's public key—and must also contact the CA to ensure that the user's certificate has not been revoked by the CA.

For higher security certifications, the CA requires that a unique "fingerprint" be issued by the CA for each message sent by the user. The user submits the message to the CA, who creates the unique fingerprint by combining the CA's private key with the message's authentication key contents. Because the user must obtain a unique fingerprint for each message, this ensures that the CA has not revoked the certificate between the time it was issued and the time the message was sent by the user.

Pretty Good Privacy (PGP) is a freeware public key encryption package developed by Philip Zimmermann that is often used to encrypt e-mail. Users post their public key on Web pages, for example, and anyone wishing to send them an encrypted message simply cuts and pastes the key off the Web page into the PGP software, which encrypts and sends the message.[14]

Secure Sockets Layer (SSL) is an encryption protocol widely used on the Web. SSL operates between the application layer software and the transport layer (in what the OSI model calls the presentation layer). SSL encrypts outbound packets coming out of the application layer before they reach the transport layer and decrypts inbound packets coming out of the transport layer before they reach the application layer. With SSL, the client and the server start with a handshake for PKI authentication and for the server to provide its public key and preferred encryption technique to the client (usually RC4, DES, 3DES, or AES). The client then generates a key for this encryption technique, which is sent to the server encrypted with the server's public key. The rest of the communication then uses this encryption technique and key.

IP Security Protocol (IPSec) is another widely used encryption protocol. IPSec differs from SSL in that SSL is focused on Web applications, while IPSec can be used with a much wider variety of application layer protocols. IPSec sits between IP at the network layer and TCP/UDP at the transport layer. IPSec can use a wide variety of encryption techniques so the first step is for the sender and receiver to establish the technique and key to be used. This is done using *Internet Key Exchange (IKE)*. Both parties generate a random key and send it to the other using an encrypted authenticated PKI process, and then put these two numbers together to produce the key.[15] The encryption technique is also

[14]For example, Cisco posts the public keys it uses for security incident reporting on its Web site; go to www.cisco.com and search on "security incident response." For more information on PGP, see www.pgpi.org and www.pgp.com.

[15]This is done using the Diffie-Hellman process; see the FAQ atwww.rsasecurity.com

negotiated between the two, often being 3DES. Once the keys and technique have been established, IPSec can begin transmitting data.

IPSec can operate in either transport mode or tunnel mode. In *transport mode,* IPSec encrypts just the IP payload, leaving the IP packet header unchanged so it can be easily routed through the Internet. In this case, IPSec adds an additional packet (either an Authentication Header [AH] or an Encapsulating Security Payload [ESP]) at the start of the IP packet that provides encryption information for the receiver.

In *tunnel mode,* IPSec encrypts the entire IP packet, and must therefore add an entirely new IP packet that contains the encrypted packet, as well as the IPSec AH or ESP packets. In tunnel mode, the newly added IP packet just identifies the IPSec encryption agent at the next destination, not the final destination; once the IPSec packet arrives at the encryption agent, the encrypted packet is decrypted and sent on its way. In tunnel mode, attackers can only learn the endpoints of the tunnel, not the ultimate source and destination of the packets.

Encryption is an important security control, whether it is used to secure backups, data inside the network, or user access from outside the network. However, encrypting data streams and stored data is processor intensive. You must decrypt every byte you read, and encrypt every byte you write. This uses up computer cycles, and lots of them. If you are storing data with encryption, you may have to boost processing and RAM requirements on your file servers.

Authenticating Users Once the network perimeter and the network interior have been secured, the next step is to develop a way to ensure that only authorized users are permitted into the network and into specific resources in the interior of the network. This is called *user authentication.*

The basis of user authentication is the *user profile* for each user's *account* that is assigned by the network manager. Each user's profile specifies what data and network resources he or she can access, and the type of access (read only, write, create, delete).

Gaining access to an account can be based on *something you know*, *something you have*, or *something you are.* The most common approach is *something you know*, usually a *password.* Before users can login, they need to enter a password. Unfortunately, passwords are often poorly chosen, enabling intruders to guess them and gain access.

Requiring passwords provides at best mid-level security (much like locking your doors when you leave the house); it won't stop the professional intruder, but it will slow amateurs. More and more systems are requiring users to enter a password in conjunction with *something they have*, such as a *smart card.* A smart card is a card about the size of a credit card that contains a small computer chip. This card can be read by a smart device and in order to gain access to the network, the user must present both the card and the password. Intruders must have access to both before they can break in. The best example of this is the automated teller machine (ATM) network operated by your bank. Before you can gain access to you account, you must have both your ATM card and the access number.

Another approach is to use *one-time passwords.* The user connects into the network as usual, and after the user's password is accepted, the system generates a one-time password. The user must enter this password to gain access, otherwise the connection is terminated. The user can receive this one-time password in a number of ways (e.g., via a pager). Other systems provide the user with a unique number that must be entered into a separate handheld device (called a *token* system), which in turn displays the password for

MANAGEMENT FOCUS **11-9 SELECTING PASSWORDS**

The key to users' accounts are passwords; each account has a unique password chosen by the user. The problem is that passwords are often chosen poorly and not changed regularly. Many network managers require users to change passwords periodically (e.g., every 90 days), but this does not ensure that users choose "good" passwords.

A good password is one that the user finds easy to remember, but is difficult for potential intruders to guess. Several studies have found that about three-quarters of passwords fall into one of four categories:

- Names of family members or pets
- Important numbers in the user's life (e.g., SSN or birthday)
- Words in a dictionary, whether an English or other language dictionary (e.g., cat, hunter, supercilious, gracias, ici)
- Keyboard patterns (e.g., QWERTY, ASDF)

The best advice is to avoid these categories because such passwords can be easily guessed.

Better choices are passwords that:

- Are meaningful to the user but no one else
- Are at least seven characters long
- Are made of two or more words that have several letters omitted (e.g., PPLEPI [apple pie]) or are the first letters of the words in phase that is not in common usage (e.g., no song lyrics) such as hapwicac (hot apple pie with ice cream and cheese)
- Include characters such as numbers or punctuation marks in the middle of the password (e.g., 1hapwic,&c for one hot apple pie with ice cream, and cheese)
- Include some uppercase and lowercase letters (e.g., 1HAPwic,&c)
- Substitute numbers for certain letters that are similar, such as using a 0 instead of an O, a 1 instead of an I, a 2 instead of a Z, a 3 instead of an E, and so on (e.g., 1HAPw1c,&c)

For more information, see www.securitystats.com/tools/password.asp.

the user to enter. Other systems use *time-based tokens* in which the one-time password is changed every 60 seconds. The user has a small device (often attached to a key chain) that is synchronized with the server and displays the one-time password. With any of these systems, an attacker must know the user's account name, password, and have access to the user's password device before he or she can login.

In high-security applications, a user may be required to present *something they are*, such as a finger, hand, or the retina of their eye for scanning by the system. These *biometric systems* scan the user to ensure that the user is the sole individual authorized to access the network account. While most biometric systems are developed for high-security users, several low-cost biometric systems are now on the market. The most popular biometric system is the fingerprint scanner. Several vendors sell devices the size of a mouse that can scan a user's fingerprint for less than $100. Other technologies include facial scans via small desktop video-conferencing cameras and retina scans by more sophisticated devices. While some banks have begun using fingerprint devices for customer access to their accounts over the Internet, such devices have not become widespread, which we find a bit puzzling. The fingerprint is unobtrusive and means users no longer have to remember arcane passwords.

User profiles can limit the allowable log-in days, time of day, physical locations, and the allowable number of incorrect log-in attempts. Some will also automatically log a user out if that person has not performed any network activity for a certain length of time (e.g., the user has gone to lunch and has forgotten to log off the network). Regular security checks throughout the day when the user is logged in can determine whether a user is still permitted access to the network. For example, the network manager might have disabled the user's profile while the user is logged in, or the user's account may have run out of funds.

Creating accounts and profiles is simple. When a new staff member joins an organization, that person is assigned a user account and profile. One security problem is the removal of user accounts when someone leaves an organization. Often, network managers are not informed of the departure and accounts remain in the system. For example, an examination of the user accounts at the University of Georgia found 30 percent belonged to staff members no longer employed by the university. If the staff member's departure was not friendly, there is a risk that he or she may attempt to access data and resources and use them for personal gain, or destroy them to "get back at" the organization. Many systems permit the network manager to assign expiration dates to user accounts to ensure that unused profiles are automatically deleted or deactivated, but these actions do not replace the need to notify network managers about an employee's departure as part of the standard Human Resources procedures.

TECHNICAL FOCUS 11-7 CRACKING A PASSWORD

To crack Windows passwords, you just need to get a copy of the SAM file in the WINNT directory, which contains all the Windows passwords in an encrypted format. If you have physical access to the computer, that's sufficient. If not, you might be able to hack in over the network. Then, you just need to use a Windows-based cracking tool such as LophtCrack. Depending upon the difficulty of the password, the time needed to crack the password via brute force could take minutes or up to a day.

Or that's the way it used to be. Recently the *Cryptography and Security Lab* in Switzerland developed a new password-cracking tool that relies upon very large amounts of RAM. It then does indexed searches of possible passwords that are already in memory. This tool can cut cracking times to less than 1/10 of the time of previous tools. Keep adding RAM and mHertz and you could reduce the crack times to 1/100 that of the older cracking tools. This means that if you can get your hands on the Windows-encrypted password file, then the game *is over.* It can literally crack complex passwords in Windows in seconds.

It's different for Linux, Unix, or Apple computers. These systems insert a 12-bit random "salt" to the password, which means that cracking their passwords will take 4,096 (2^12) times longer to do. That margin is probably sufficient for now, until the next generation of cracking tools comes along. Maybe.

So what can we say from all of this? That you are 4,096 times safer with Linux? Well, not necessarily. But what we may be able to say is that strong password protection, by itself, is an oxymoron. We must combine it with other methods of security to have reasonable confidence in the system.

One long-standing problem has been that users are often assigned user profiles and passwords on several different computers. Each time a user wants to access a new server, he or she must supply his or her password. This is cumbersome for the users, and even worse for the network manager who must manage all the separate accounts for all the users.

More and more organizations are adopting *network authentication* (also called central authentication, single sign-on, or directory services), in which a login server is used to authenticate the user. Instead of logging into a file server or application server, the user logs into the *authentication server.* This server checks the userid and password against its database and if the user is an authorized user, issues a *certificate* (also called credentials). Whenever the user attempts to access a restricted service or resource that requires a userid and password, the user is challenged and his or her software presents the certificate to the authentication server (which is revalidated by the authentication server at the time). If the authentication server validates the certificate, then the service or resource lets the user in. In this way, the user no longer needs to enter his or her password to be authenticated to each new resource or service he or she uses. This also ensures that the user does not accidentally give out his or her password to an unauthorized service—it provides mutual authentication of both the user and the service or resource. The most commonly used authentication protocol is *Kerberos*, developed at MIT (see web.mit.edu/kerberos/www).

While many systems use only one authentication server, it is possible to establish a series of authentication servers for different parts of the organization. Each server authenticates clients in its domain but can also pass authentication credentials to authentication servers in other domains.

Social Engineering One of the most common ways for attackers to break into a system, even master hackers, is through *social engineering,* which refers to breaking security simply by asking. For example, attackers routinely phone unsuspecting users and, imitating someone such as a technician or senior manager, ask for a password. Unfortunately, too many users want to be helpful and simply provide the requested information. At first, it seems ridiculous to believe that someone would give their password to a complete stranger, but a skilled social engineer is like a good con artist: he—and most social engineers are men—can manipulate people.[16]

Most security experts no longer test for social engineering attacks; they know from experience that social engineering will eventually succeed in any organization and therefore assume that attackers can gain access at will to normal user accounts. Training end users not to divulge passwords may not eliminate social engineering attacks, but it may reduce their effectiveness so that hackers give up and move on to easier targets. Acting out social engineering skits in front of users often works very well; when employees see how they can be manipulated into giving out private information, it becomes more memorable and they tend to become much more careful.

Phishing is a very common type of social engineering. The attacker simply sends an e-mail to millions of users telling them that their bank account has been shut down due to an unauthorized access attempt and that they need to reactivate it by logging in. The e-mail

[16]For more information about social engineering and many good examples, see *The Art of Deception* by Kevin Mitnick and William Simon.

TECHNICAL FOCUS

11-8 INSIDE KERBEROS

Kerberos, the most commonly used authentication protocol, uses symmetric encryption. When you login to a Windows network that uses active directory services, the Kerberos client software in your computer sends a request to the Windows Domain Controller (i.e., the authentication server or the ticket-granting service [TGS] of the Key Distribution Center [KDC], in Kerberos terminology). The request contains the userid and preauthentication data (e.g., a time and date stamp) that have been encrypted using the user's password as the encryption key.

The KDC checks its database for the user id and uses the password associated with that user id to decrypt the preauthentication data. If the preauthentication data are correct after decrypting with the user's password, then the KDC accepts the login. The KDC generates a unique session key (SK1), which will be used to encrypt all further communication between the client computer and the KDC until the user logs off. The SK1 is generated separately for each user and is different each and every time the user logs in. The KDC encrypts the SK1 using the user's password and sends it to the user's client computer. The client receives the SK1 and decrypts it using the user's password.

The KDC also creates a Ticket-Granting Ticket (TGT). The TGT includes the SK1, plus some other information (e.g., the user computer's address). The KDC encrypts the TGT using the KDC's unique key and sends it to the client computer as well (encrypted with SK1, of course, because all communications between the client and the server are encrypted with SK1). The client decrypts the transmission to receive the TGT, but because the client does not know the KDC key, it cannot decrypt the *contents* of the TGT. From now until the user logs off, the user does not need to provide his or her password again; the Kerberos client software will use the TGT to gain access to all servers that require a password.

When the user accesses a restricted server that requires a password, the user's Kerberos client sends the TGT to the KDC (remember that all communications between the client and the server are encrypted with the SK1 until the user logs off). If the TGT is validated, the KDC sends the client a service ticket (ST) for the desired server and a new session key (SK2) that the client will use to communicate with the new server, both of which have been encrypted using SK1. The ST contains authentication information and the SK2, both of which have been encrypted using a key known only to the KDC and the server. The client presents the ST to the server, which decrypts it using the KDC key to find the authentication information and the SK2 to be used with the client. The server then sends the client a date time stamp packet that has been encrypted with the SK2. This process authenticates the client to the server, and also authenticates the server to the client. Both now communicate using SK2.

contains a link that directs the user to a fake Web site that appears to be the bank's Web site. After the user logs into the fake site, the attacker has the user's userid and password and can break into his or her account at will. Clever variants on this include an e-mail informing you that a new user has been added to your paypal account, stating that the IRS has issued you a refund and you need to verify your social security number, or offering a mortgage at very low rate for which you need to provide your social security number and credit number.

Detecting Intrusion

The previous section focused on preventing intrusion. While one hopes that these techniques are successful, the possibility of a security break-in still remains. Therefore, networks often need an *intrusion prevention system (IPS)*.

MANAGEMENT FOCUS **11-10 SOCIAL ENGINEERING WINS AGAIN**

Danny had collected all the information he needed to steal the plans for the new product. He knew the project manager's name (Bob Billings), phone number, department name, office number, computer user id, and employee number, as well as the project manager's boss's name. These had come from the company Web site and a series of innocuous phone calls to helpful receptionists. He had also tricked the project manager into giving him his password, but that hadn't worked because the company used one-time passwords using a time-based token system called Secure ID. So, after getting the phone number of the computer operations room from another helpful receptionist, all he needed was a snowstorm.

Late one Friday night, a huge storm hit and covered the roads with ice. The next morning, Danny called the computer operations room:

Danny: "Hi, this is Bob Billings in the Communications Group. I left my Secure ID in my desk and I need it to do some work this weekend. There's no way I can get into the office this morning. Could you go down to my office and get it for me? And then read my code to me so I can login?"

Operations: "Sorry, I can't leave the Operations Center."

Danny: "Do you have a Secure ID yourself?"

Operations: "There's one here we keep for emergencies."

Danny: "Listen. Can you do me a big favor? Could you let me borrow your Secure ID? Just until it's safe to drive in?"

Operations: "Who are you again?"

Danny: "Bob Billings. I work for Ed Trenton."

Operations: "Yeah, I know him."

Danny: "My office is on the second floor (2202B). Next to Roy Tucker. It'd be easier if you could just get my Secure ID out of my desk. I think it's in the upper left drawer." (Danny knew the guy wouldn't want to walk to a distant part of the building and search someone else's office.)

Operations: "I'll have to talk to my boss."

After a pause, the operations technician came back on and asked Danny to call his manager on his cell phone. After talking with the manager and providing some basic information to "prove" he was Bob Billings, Danny kept asking about having the Operations technician go to "his" office.

Finally, the manager decided to let Danny use the Secure ID in the Operations Center. The manager called the technician and gave permission for him to tell "Bob" the one-time password displayed on their Secure ID any time he called that weekend. Danny was in.

SOURCE: Kevin Mitnick and William Simon, *The Art of Deception,* John Wiley and Sons, 2002.

There are three general types of IPSs, and many network managers choose to install all three. The first type is a *network-based IPS.* With a network-based IPS, an *IPS sensor* is placed on key network circuits. An IPS sensor is simply a device running a special operating system that monitors all network packets on that circuit and reports intrusions to an *IPS management console.* The second type of IPS is the *host-based IPS,* which, as the name suggests, is a software package installed on a host or server. The host-based IPS monitors activity on the server and reports intrusions to the IPS management console. An *application-based IPS* is a specialized form of host-based IPS that just monitors one application on the server, often a Web server.

There are two fundamental techniques that these three types of IPSs can use to determine that an intrusion is in progress; most IPSs use both techniques. The first technique is *misuse detection,* which compares monitored activities with signatures of known

attacks. Whenever an attack signature is recognized, the IPS issues an alert and discards the suspicious packets. The problem, of course, is keeping the database of attack signatures up to date as new attacks are invented.

The second fundamental technique is *anomaly detection,* which works well in stable networks by comparing monitored activities with the "normal" set of activities. When a major deviation is detected (e.g., a sudden flood of ICMP ping packets, an unusual number of failed logins to the network manager's account), the IPS issues an alert and discards the suspicious packets. The problem, of course, is false alarms when situations occur that produce valid network traffic that is different from normal (e.g., on a heavy trading day on Wall Street, E-trade receives a larger than normal volume of messages).

IPSs are often used in conjunction with other security tools such as firewalls (Figure 11.17). In fact, some firewalls are now including IPS functions. One problem is that the IPS and its sensors and management console are a prime target for attackers. Whatever IPS is used, it must be very secure against attack. Some organizations deploy redundant IPSs from different vendors (e.g., a network-based IPS from one vendor and a host-based IPS from another) in order to decrease the chance that the IPS can be hacked.

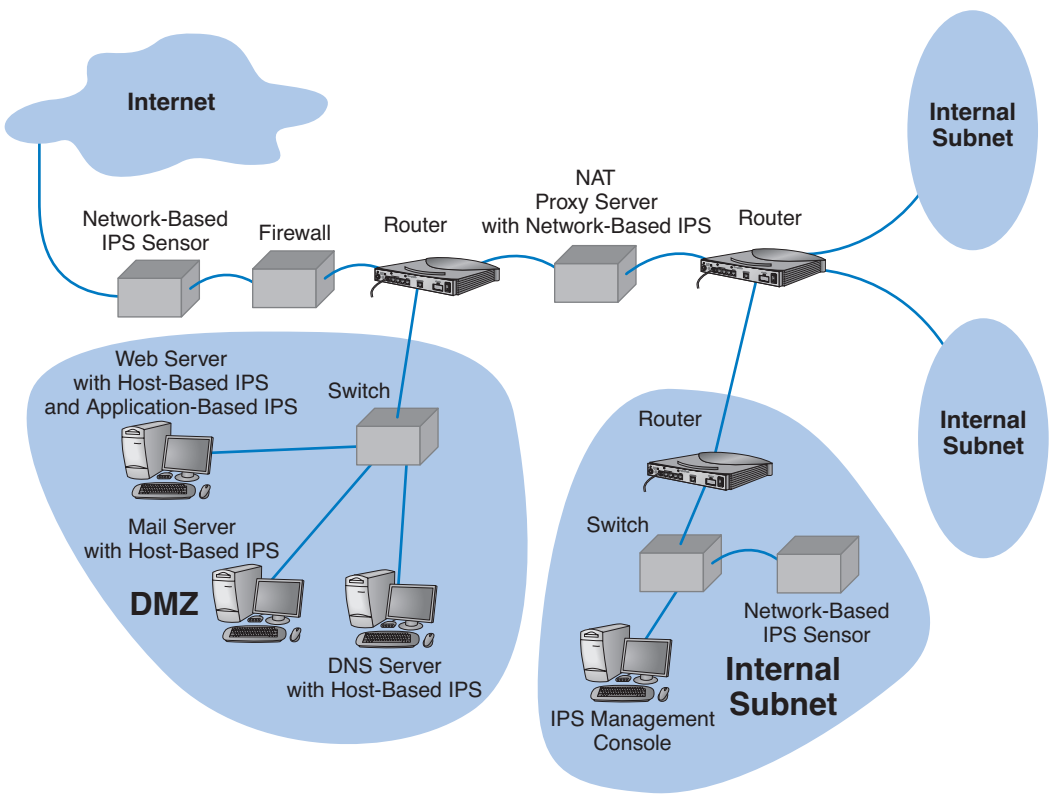


FIGURE 11.17 Intrusion prevention system (IPS). DMZ = demilitarized zone; DNS = Domain Name Service; NAT = network address translation.

TECHNICAL FOCUS 11-9 INTRUSION DETECTION GETS ACTIVE

Worms have been responsible for some of the most costly virus infections because they spread much more quickly than traditional viruses. The "Code Red" worm, for example, spread by using a security hole in Microsoft's IIS Web server software. By sending an HTTP request that is too large for the server's incoming message buffer, the server can be tricked into running operating system commands contained in the HTTP request. The commands imbedded in the request install the worm, which then attempts to infect other computers by sending the same HTTP request to more computers.

A new IPS freely available on the Internet has developed a way to trap worms that minimizes or prevents their spread. Called LaBrea for the LaBrea Tarpits in California that trapped hundreds of dinosaurs, the tool traps the connection requests that many worms use when they spread. LaBrea is the first of a new breed of IDSs that detect and attempt to disable the intrusion.

When Code Red and similar worms attempt to spread, they send HTTP requests containing the worm addressed to all IP addresses they can think of (e.g., if they have infected a company with an IP range of 128.196.x.x, they first try 128.196.1.1, then 128.196.1.2, then 128.196.1.3, and so on). In most cases, there are no Web servers on most of these addresses, so the worm ends up trying to reach computers that do not exist. When the worm sends an HTTP request, the TCP software on the infected computer first sends a TCP open connection request to a selected IP address before the HTTP request is sent (see the TCP/IP example in Chapter 5). The TCP request eventually reaches the router that is the gateway into the TCP/IP subnet that would have a Web server with the IP address if the computer existed. If there is no server with the requested IP address, the router doesn't have an Ethernet address that matches the IP address in its memory, and thus the router broadcasts an ARP, requesting that the computer with that IP address send its Ethernet address to the router. Of course, no computer will respond because there is no computer with that IP address. ARP is a tenacious protocol. Because it expects that there really is a computer with that IP address, the router will issue the ARP many times without getting an answer before it gives up and returns the message to the sender as undeliverable.

This is where LaBrea steps in. After hearing several ARP requests for the same IP address go unanswered, LaBrea will issue an ARP response to the router, giving its computer's Ethernet address as the one that matches the phantom IP address. From this point forward, all messages targeted at the phantom IP address will be delivered to the LaBrea software. When LaBrea receives the TCP open connection request that precedes the HTTP request containing the worm, LaBrea will accept the open connection but not acknowledge the TCP segment in the normal way. TCP is also a tenacious protocol, which means that the TCP software at the infected machine will keep trying to send data, but will never quite succeed because LaBrea never responds properly. LaBrea will also try to trick the sending computer's TCP software into accepting a "persistent connection," which means that the connection will not be closed until the receiver (i.e., the LaBrea software) closes it—which, of course, it will never do.

By holding the connection open, the LaBrea software prevents the worm from moving onto the next IP address in its sequence, or at least significantly delays its movement to the next IP address. And of course, the next false IP address that the worm tries will again be met by the LaBrea software.

Because LaBrea holds connections open indefinitely, it becomes much easier to contact the owners of the infected computer and enable them to identify and fix the problem. LaBrea will respond to all requests, not just HTTP requests, so it is able to capture and hold open connections from port scanning software often used by hackers—which again makes it possible to trace them more easily.

Correcting Intrusion

While IPS monitoring is important, it has little value unless there is a clear plan for responding to a security breach in progress. Every organization should have a clear response planned if a break-in is discovered. Many large organizations have emergency response "SWAT" teams ready to be called into action if a problem is discovered. The best example is CERT, which is the Internet's emergency response team. CERT has helped many organizations establish such teams.

Responding to an intrusion can be more complicated than it at first seems. For example, suppose the IPS detects a DoS attack from a certain IP address. The immediate reaction could be to discard all packets from that IP address; however, in the age of IP spoofing, the attacker could fake the address of your best customer and trick you into discarding packets from it.

Once an intrusion has been detected, the first step is to identify how the intruder gained unauthorized access and prevent others from breaking in the same way. Some organizations will simply choose to close the door on the attacker and fix the security problem. Other organizations may take a more aggressive response by logging the intruder's activities and working with police to catch the individuals involved. Once identified, the attacker will be charged with criminal activities and/or sued in civil court.

A whole new area called *computer forensics* has recently opened up. Computer forensics is the use of computer analysis techniques to gather evidence for criminal and/or civil trials. The basic steps of computer forensics are similar to those of traditional forensics, but the techniques are different. First, identify potential evidence. Second, preserve evidence by making backup copies and use those copies for all analysis. Third, analyze the evidence. Finally, prepare a detailed legal report for use in prosecutions. While companies are sometimes tempted to launch counterattacks (or counterhacks) against intruders, this is illegal.

Some organizations have taken their own steps to snare intruders by using *entrapment* techniques. The objective is to divert the attacker's attention from the real network to an attractive server that contains only fake information. This server is often called a *honey pot*. The honey pot server contains highly interesting, fake information available only through illegal intrusion to "bait" the intruder. The honey pot server has sophisticated tracking software to monitor access to this information that allows the organization and law enforcement officials to trace and legally document the intruder's actions. Possession of this information then becomes final legal proof of the intrusion.

BEST PRACTICE RECOMMENDATIONS

This chapter provides numerous suggestions on business continuity planning and intrusion prevention. Good security starts with a clear disaster recovery plan and a solid security policy. Probably the best security investment is user training: training individual users on data recovery and ways to defeat social engineering. But this doesn't mean that technologies aren't needed either.

Figure 11.18 shows the most commonly used security controls. Most organizations now routinely use antivirus software, firewalls, physical security, intrusion detection, and encryption.

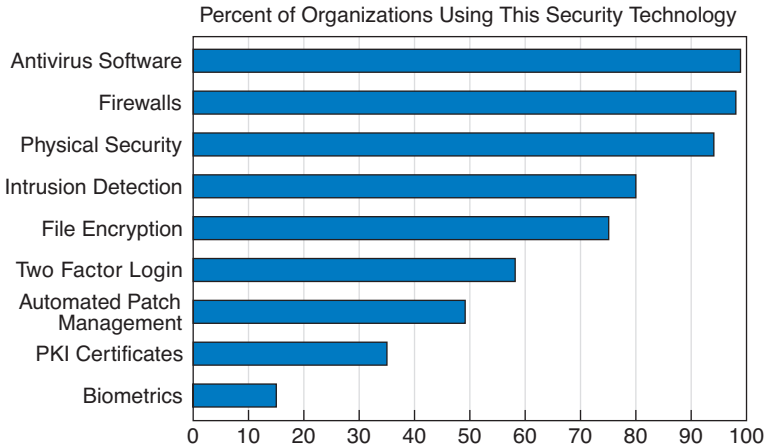


FIGURE 11.18 Percent of organizations using certain security technologies. PKI = public key infrastructure.

SOURCE: CSI/FBI Computer Crime and Security Survey, 2005 and SS/CSO/CERT E-Crime Survey, 2005.

Even so, rarely does a week pass without a new warning of a major vulnerability. Leave a server unattended for two weeks, and you may find that you have five critical patches to install.

People are now asking, "Will it end?" Is (in)security just a permanent part of the information systems landscape? In a way, yes. The growth of information systems, along with the new and dangerous ability to reach into them from around the world, has created new opportunities for criminals. Mix the possibilities of stealing valuable, marketable information with the low possibilities for getting caught and punished, and we would expect increasing numbers of attacks.

Perhaps the question should be: Does it have to be this bad? Unquestionably, we could be protecting ourselves better. We could better enforce security policies and restrict access. But all of this has a cost. Attackers are writing and distributing a new generation of attack tools right before us—tools that are very powerful, more difficult to detect, and very easy to use. Usually such tools are much easier to use than their defensive countermeasures.

The attackers have another advantage, too. Whereas the defenders have to protect *all* vulnerable points *all the time* in order to be safe, the attacker just has to break into *one place one time* to be successful.

So what may we expect in the future in "secure" organizational environments? We would expect to see strong *desktop management,* including the use of thin clients (perhaps even network PCs that lack hard disks). Centralized desktop management, in which individual users are not permitted to change the settings on their computers with regular reimaging of computers to prevent Trojans and viruses and to install the most recent security patches. All external software downloads will likely be prohibited.

Continuous content filtering, in which all incoming packets (e.g., Web, e-mail) are scanned, may become common, thus significantly slowing down the network. All server

files and communications with client computers would be encrypted, further slowing down transmissions.

Finally, all written security policies would be rigorously enforced. Violations of security policies might even become a "capital offense" (i.e., meaning one violation and you are fired).

We may look forlornly back to the early days of the Internet when we could "do anything" as its Golden Days.

IMPLICATIONS FOR MANAGEMENT

Network security was once an esoteric field of interest to only a few dedicated professionals. Today, it is the fastest-growing area in networking. The cost of network security will continue to increase as the tools available to network attackers become more sophisticated, as organizations rely more and more on networks for critical business operations, and as information warfare perpetrated by nations or terrorists becomes more common. As the cost of networking technology decreases, the cost of staff and

A DAY IN THE LIFE: NETWORK SECURITY MANAGER

"Managing security is a combination of detective work and prognostication about the future."

A network security manager spends much of his or her time doing three major things. First, much time is spent looking outside the organization by reading and researching potential security holes and new attacks because the technology and attack opportunities change so fast. It is important to understand new attack threats, new scripting tools used to create viruses, remote access Trojans and other harmful software, and the general direction in which the hacking community is moving. Much important information is contained at Web sites such as those maintained by CERT (www.cert.org) and SANS (www.sans.org). This information is used to create new versions of standard computer images that are more robust in defeating attacks, and to develop recommendations for the installation of application security patches. It also means that he or she must update the organization's written security policies and inform users of any changes.

Second, the network security manager looks inward toward the networks he or she is responsible for. He or she must check the vulnerability of those networks by thinking like a hacker to understand how the networks may be susceptible to attack, which often means scanning for open ports and unguarded parts of the networks and looking for computers that have not been updated with the latest security patches. It also means looking for symptoms of compromised machines such as new patterns of network activity or unknown services that have been recently opened on a computer.

Third, the network security manager must respond to security incidents. This usually means "firefighting"—quickly responding to any security breach, identifying the cause, collecting forensic evidence for use in court, and fixing the computer or software application that has been compromised.

With thanks to Kenn Crook

networking technologies providing security will become an increasingly larger proportion of an organization's networking budget. As organizations and governments see this, there will be a call for tougher laws and better investigation and prosecution of network attackers.

Security tools available to organizations will continue to increase in sophistication and the use of encryption will become widespread in most organizations. There will be an ongoing "arms race" between security officers in organizations and attackers. Software security will become an important factor in selecting operating systems, networking software, and application software. Those companies that provide more secure software will see a steady increase in market share while those that don't will gradually lose ground.

SUMMARY

Types of Security Threats In general, network security threats can be classified into one of two categories: (1) business continuity and (2) unauthorized access. Disruptions are usually minor and temporary. Some disruptions may also be caused by or result in the destruction of data. Natural (or man-made) disasters may occur that destroy host computers or large sections of the network. Unauthorized access refers to intruders (external attackers or organizational employees) gaining unauthorized access to files. The intruder may gain knowledge, change files to commit fraud or theft, or destroy information to injure the organization.

Risk Assessment Developing a secure network means developing controls that reduce or eliminate threats to the network. Controls prevent, detect, and correct whatever might happen to the organization when its computer-based systems are threatened. The first step in developing a secure network is to conduct a risk assessment. This is done by identifying the key assets and threats and comparing the nature of the threats to the controls designed to protect the assets. A control spreadsheet lists the assets, threats, and controls that a network manager uses to assess the level of risk.

Business Continuity Planning The key principle in controlling these threats—or at least reducing their impact—is redundancy. Redundant hardware that automatically recognizes failure and intervenes to replace the failed component can mask a failure that would otherwise result in a service disruption. Special attention needs to be given to preventing computer viruses and denial-of-service attacks. Generally speaking, preventing disasters is difficult, so the best option is a well-designed disaster recovery plan that includes backups and sometimes a professional disaster recovery firm.

Intrusion Prevention The key principle in intrusion prevention is to be proactive in routinely testing and upgrading security controls. Intruders are both organization employees and external attackers. There are four general ways to prevent intrusion: developing a strong security policy, securing the network perimeter (physical security, firewalls, network address translation, and dial-in security), securing the network interior (security holes, preventing remote access Trojans, and encryption), and authenticating users (something they know, something they have, something they are, and guarding against social engineering). The best approach in detecting intrusion is using an intrusion prevention system to monitor for known attacks and/or to look for anything out of the ordinary.

KEY TERMS

access control list
account
Advanced Encryption Standard (AES)
adware
anomaly detection
application-based IPS
application-level firewall
asset
asymmetric encryption
authentication
authentication server
automatic number identification (ANI)
backup controls
biometric system
block cipher
brute-force attack
business continuity planning
candy security
certificate
certificate authority (CA)
ciphertext
closed source
Computer Emergency Response Team (CERT)
computer forensics
continuous data protection (CDP)
control principles
control spreadsheet
controls
cracker
Data Encryption Standard (DES)
DDoS agent
DDoS handler
decryption
Delphi team
denial-of-service (DoS) attack
desktop management
disaster recovery drill
disaster recovery firm
disaster recovery plan
disk mirroring
distributed denial-of-service (DDoS) attack
eavesdropping
encryption
entrapment
fault-tolerant server
firewall
hacker
honey pot
host-based IPS
IPS management console
IPS sensor
information warfare
Internet Key Exchange (IKE)
intrusion prevention system (IPS)
IP Security Protocol (IPSec)
IP spoofing
IPSec transport mode
IPSec tunnel mode
Kerberos
key
key escrow
key management
mission-critical application
misuse detection
NAT proxy server
network address translation (NAT)
network authentication
network-based IPS
one-time password
open source
packet-level firewall
password
patch
phishing
physical security
plaintext
Pretty Good Privacy (PGP)
private key
public key
public key encryption
public key infrastructure (PKI)
RC4
recovery controls
redundancy
risk assessment
rootkit
RSA
script kiddies
secure hub
security hole
security policy
smart card
sniffer program
social engineering
something you are
something you have
something you know
spyware
symmetric encryption
threat
time-based token
token
traffic analysis
traffic anomoly analyzer
traffic anomoly detector
traffic filtering
traffic limiting
triple DES (3DES)
Trojan horse
uninterruptible power supply (UPS)
user profile
user authentication
virus
worm

QUESTIONS

1. What factors have brought increased emphasis on network security?
2. Briefly outline the steps required to complete a risk assessment.
3. Name at least six assets that should have controls in a data communication network.
4. What are some of the criteria that can be used to rank security risks?
5. What are the most common security threats? What are the most critical? Why?
6. Explain the primary principle of business continuity planning.
7. What is the purpose of a disaster recovery plan? What are five major elements of a typical disaster recovery plan?
8. What is a computer virus? What is a worm?

9. How can one reduce the risk of natural disaster?
10. Explain how a denial-of-service attack works.
11. How does a denial-of-service attack differ from a distributed denial-of-service attack?
12. What is a disaster recovery firm? When and why would you establish a contract with them?
13. Explain the primary principle of controlling unauthorized access.
14. People who attempt unauthorized access can be classified into four different categories. Describe them.
15. There are many components in a typical security policy. Describe three important components.
16. What are the three major aspects of controlling unauthorized access (not counting the security policy)?
17. How do you secure the network perimeter?
18. What is physical security and why is it important?
19. What is eavesdropping in a computer security sense?
20. What is a sniffer?
21. How do you secure dial-in access?
22. Describe how an ANI modem works.
23. What is a firewall?
24. How do the different types of firewalls work?
25. What is IP spoofing?
26. What is a NAT proxy server and how does it work?
27. What is a security hole and how do you fix it?
28. Explain how a Trojan horse works.
29. Compare and contrast symmetric and asymmetric encryption.
30. Describe how symmetric encryption and decryption work.
31. Describe how asymmetric encryption and decryption work.
32. What is key management?
33. How does DES differ from 3DES? From RC4? From AES?
34. Compare and contrast DES and public key encryption.
35. Explain how authentication works.
36. What is PKI and why is it important?
37. What is a certificate authority?
38. How does PGP differ from SSL?
39. How does SSL differ from IPSec?
40. Compare and contrast IPSec tunnel mode and IPSec transfer mode.
41. What are the three major ways of authenticating users? What are the pros and cons of each approach?
42. What are the different types of one-time passwords and how do they work?
43. Explain how a biometric system can improve security. What are the problems with it?
44. Why is the management of user profiles an important aspect of a security policy?
45. How does network authentication work and why is it useful?
46. What is social engineering? Why does it work so well?
47. What techniques can be used to reduce the chance that social engineering will be successful?
48. What is an intrusion detection system?
49. Compare and contrast a network-based IPS, a host-based IPS, and an application-based IPS.
50. How does IPS anomaly detection differ from misuse detection?
51. What is computer forensics?
52. What is a honey pot?
53. What is desktop management?
54. A few security consultants have said that broadband and wireless technologies are their best friends. Explain.
55. Most hackers start their careers breaking into computer systems as teenagers. What can we as a community of computer professionals do to reduce the temptation to become a hacker?
56. Some experts argue that CERT's posting of security holes on its Web site causes more security break-ins than it prevents and should be stopped. What are the pros and cons on both sides of this argument? Do you think CERT should continue to post security holes?
57. What is one of the major risks of downloading unauthorized copies of music files from the Internet (aside from the risk of jail, fines, and lawsuits)?
58. Suppose you started working as a network manager at a medium-sized firm with an Internet presence, and discovered that the previous network manager had done a terrible job of network security. Which *four* security controls would be your *first* priority? Why?
59. How can we reduce the number of viruses that are created every month?
60. While it is important to protect all servers, some servers are more important than others. What server(s) are the most important to protect and why?

EXERCISES

11-1. Conduct a risk assessment of your organization's networks. Some information may be confidential, so report what you can.

11-2. Investigate and report on the activities of CERT (the Computer Emergency Response Team).

11-3. Investigate the capabilities and costs of a disaster recovery service.

11-4. Investigate the capabilities and costs of a firewall.

11-5. Investigate the capabilities and costs of an intrusion detection system.

11-6. Investigate the capabilities and costs of an encryption package.

MINICASES

I. Belmont State Bank

Belmont State Bank is a large bank with hundreds of branches that are connected to a central computer system. Some branches are connected over dedicated circuits and others use the dial-up telephone network. Each branch has a variety of client computers and ATMs connected to a server. The server stores the branch's daily transaction data and transmits it several times during the day to the central computer system. Tellers at each branch use a four-digit numeric password, and each teller's computer is transaction-coded to accept only its authorized transactions. Perform a risk assessment.

II. Western Bank

Western Bank is a small, family-owned bank with six branches spread over the county. It has decided to move onto the Internet with a Web site that permits customers to access their accounts and pay bills. Design the key security hardware and software the bank should use.

III. Classic Catalog Company, Part 1

Classic Catalog Company runs a small but rapidly growing catalog sales business. It outsourced its Web operations to a local ISP for several years but as sales over the Web have become a larger portion of its business, it has decided to move its Web site onto its own internal computer systems. It has also decided to undertake a major upgrade of its own internal networks. The company has two buildings, an office complex, and a warehouse. The two-story office building has 60 computers. The first floor has 40 computers, 30 of which are devoted to telephone sales. The warehouse, located 400 feet across the company's parking lot from the office building, has about 100,000 square feet, all on one floor. The warehouse has 15 computers in the shipping department located at one end of the warehouse. The company is about to experiment with using wireless handheld computers to help employees more quickly locate and pick products for customer orders. Based on traffic projections for the coming year, the company plans to use a T1 connection from its office to its ISP. It has three servers: the main Web server, an e-mail server, and an internal application server for its application systems (e.g., orders, payroll). Perform a risk assessment.

IV. Classic Catalog Company, Part 2

Read Minicase III above. Outline a brief business continuity plan including controls to reduce the risks in advance as well as a disaster recovery plan.

(*continued*)

MINI CASES (*continued*)

V. Classic Catalog Company, Part 3

Read Minicase III above. Outline a brief security policy and the controls you would implement to control unauthorized access.

VI. Classic Catalog Company, Part 4

Read Minicase III above. Reread Management Focus box 11-6. What patching policy would you recommend for Classic Catalog?

VII. Personal Security

Conduct a risk assessment and develop a business continuity plan and security policy for the computer(s) you own.

CASE STUDY

NEXT-DAY AIR SERVICE

See the Web site.

HANDS-ON ACTIVITY

Securing Your Computer

This chapter has focused on security, including risk analysis, business continuity, and intrusion prevention. At first glance, you may think security applies to corporate networks, not your network. However, if you have a LAN at your house or apartment, or even if you just own a desktop or laptop computer, security should be one of your concerns. There are so many potential threats to your business continuity—which might be your education—and to intrusion into your computer(s) that you need to take action.

You should perform your own risk analysis, but this section provides a brief summary of some simple actions you should take that will greatly increase your security. Do this this week; don't procrastinate. Our focus is on Windows security, because most readers of this book use Windows computers, but the same advice (but different commands) applies to Apple computers.

Business Continuity

If you run your own business, then ensuring business continuity should be a major focus of your efforts. But even if you are "just" an employee or a student, business continuity is important. What would happen if your hard disk failed just before the due date for a major report?

1. The first and most important security action you can take is to configure Windows to perform automatic updates. This will ensure you have the latest patches and updates installed.
2. The second most important action is to buy and install antivirus software such as that from McAfee or Symantec. Be sure to configure it for regular updates too. If you perform just these two actions, you will be relatively secure from viruses, but you should scan your system for viruses on a regular basis, such as the first of every month.

3. Spyware is another threat. You should buy and install antispyware software that provides the same protection that antivirus software does for viruses. Good packages include McAfee antispyware software and Spybot. Be sure to configure this software for regular updates and scan your system on a regular basis.
4. One of the largest sources of viruses, spyware, and adware is free software and music/video files downloaded from the Internet. Simply put, don't download any file unless it is from a trusted vendor or distributor of software and files.
5. Develop a disaster recovery plan. You should plan today for what you would do if your computer was destroyed. What files would you need? If there are any important files that you wouldn't want to lose (e.g., reports you're working on, key data, or precious photos), you should develop a backup and recovery plan for them. The simplest is to copy the files to a shared directory on another computer on your LAN. But this won't enable you to recover the files if your apartment or house was destroyed by fire, for example (see Management Focus 11-5). A better plan is to copy your files to a network site at your university or business at the end of each day (think CDP on the cheap). If you don't have such a site, buy a large USB drive, copy your files to it, and store it off-site in your office or at a friend's house. A plan is only good if it is followed, so your data should be regularly backed up, such as doing so the first of every month.

Intrusion Prevention

With the increase of Internet-based attacks, everyone's computer is at greater risk for intrusion, not just the computers of prominent organizations. There are a few common-sense steps you can take to prevent intrusion.

1. Think good physical security. Always turn off your computer when you are finished using it. A computer that is off cannot be attacked, either over the Internet or from someone walking by your desk.
2. Windows has the ability to have multiple user accounts. The default accounts are Administrator and Guest. You should disable the Guest account and to change the name of the administrator account so that any intruders attacking the computer will have to guess the user names as well as the passwords. It's also a good idea to create an account other than the administrator account that you can use on a day-to-day basis. The administrator account should only be used when you are installing software or changing configurations that require administrator privileges on your computer. You can manage these user accounts from the Control Panel, User Accounts. Be sure to add passwords that are secure, but easy to remember for all the accounts that you use.
3. Turn on the Windows Firewall. Use Control Panel, Security Center to examine your security settings, including the "firewall" built into Windows. The firewall is software that prevents other computers from accessing your computer. You can turn it on and examine the settings. The default settings are usually adequate, but you may want to make changes. Click on Internet Options. This will enable you to configure the firewall for four different types of site: the Internet, you local intranet (i.e., LAN), trusted sites (that have a valid PKI certificate), and restricted sites (that are sites of known hackers). Figure 11.19 shows some of the different security settings.
4. Disable unneeded services. Windows was designed to support as many applications as the developers could think of. Many of these services are not needed by most users, and unfortunately, some have become targets of intruders. For example, Windows is a Telnet server (see Chapter 2) so that anyone with a Telnet client can connect to your computer and issue operating system commands. The Telnet server is usually turned off by the person who installed Windows on your computer, but it is safer to make sure.
   a. Right click on My Computer and select Manage
   b. Click on Services and Applications and then click on Services
   c. You should see a screen like that in Figure 11.20. Make sure the Telnet service says "Disabled." If it doesn't, right click on it, Select Properties, and change the Startup Type to Disabled.
   d. Three other services that should be set to disabled are Messenger (don't worry, this is *not* any type of Instant Messenger), Remote Registry, and Routing and Remote Access.
5. If you have a LAN in your apartment or house, be sure the router connecting you to the Internet is a NAT proxy server. This will prevent many intruders from attacking your computers. The Disable WAN connections option on my router permits me to deny any TCP request from the Internet side of the router—that is, my client computer can establish outgoing TCP connections, but no one on the Internet can establish a TCP connection to a computer in my LAN.

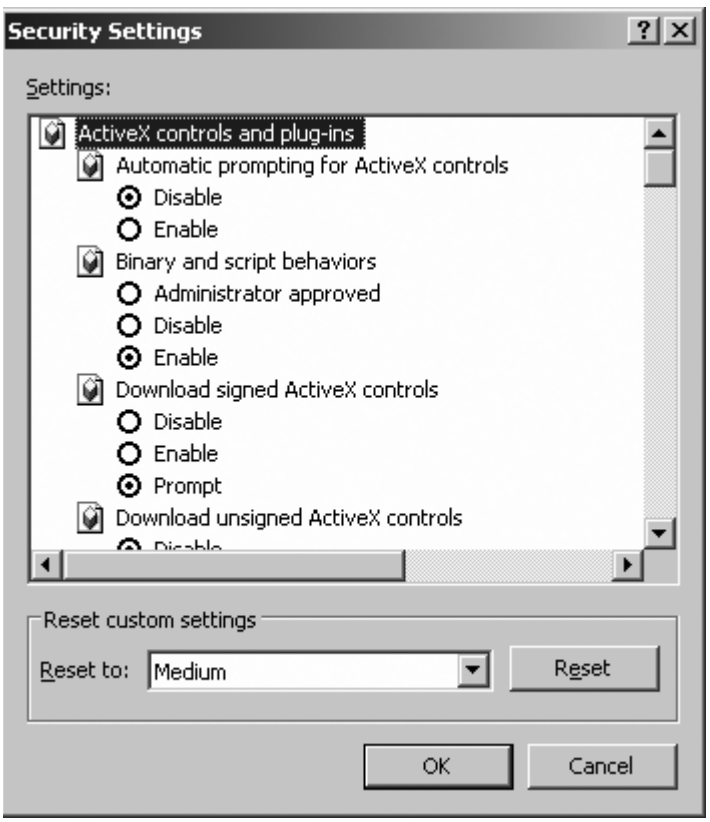


FIGURE 11.19 Security controls in Windows.

6. In Chapter 6, we described how to share files on your LAN. If you don't need to share files right now, this capability should be turned off. See Chapter 6 for more details.
7. Avoid phishing attacks. A recent analysis of e-mail found that 70 percent of all e-mail was spam and phishing attacks. That's right, "real" e-mail is outnumbered more than two-to-one by fake e-mail. Do not *ever* click on a link in an e-mail. No exceptions. *Never* click an e-mail link. Even if you are a valued customer, have been offered a chance to participate in a survey, or receive a low cost mortgage. Even if the e-mail appears to be from a well-known firm. Let us say that again: *Never* click an e-mail link. If you want to visit a Web site mentioned in an e-mail, open a new browser window and manually type the correct address. Figure 11.21 shows a recent phishing attack I received. Looks real, doesn't it? I particularly enjoyed the parts that talk about spotting and avoiding fraudulent e-mails. If I had clicked on the link, it would have taken me to a Web site owned by a Singaporean company.

Finally, you may want to have you computer scanned for vulnerabilities. Symantec, the antivirus software maker, has a free Web site that will scan your computer and list its strengths and weaknesses: scan.symantec.com. You can also see statistics from the results of scanning millions of computers. The day I scanned my computer, almost 20 percent of the computers scanned were at risk of intrusion, 10 percent failed the Windows update check, and more than 30 percent failed the Trojan and antivirus test.

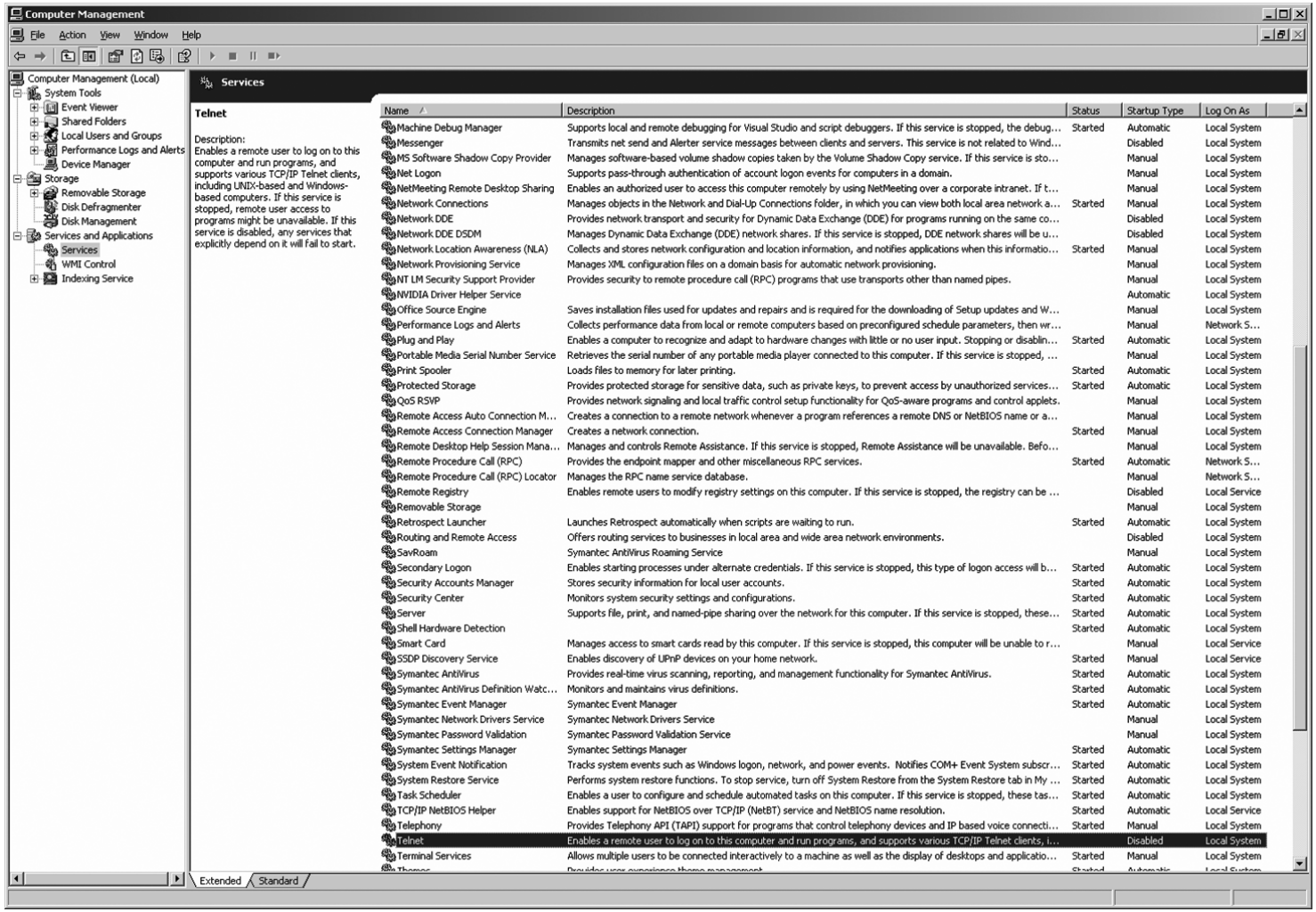


FIGURE 11.20 Windows services management.

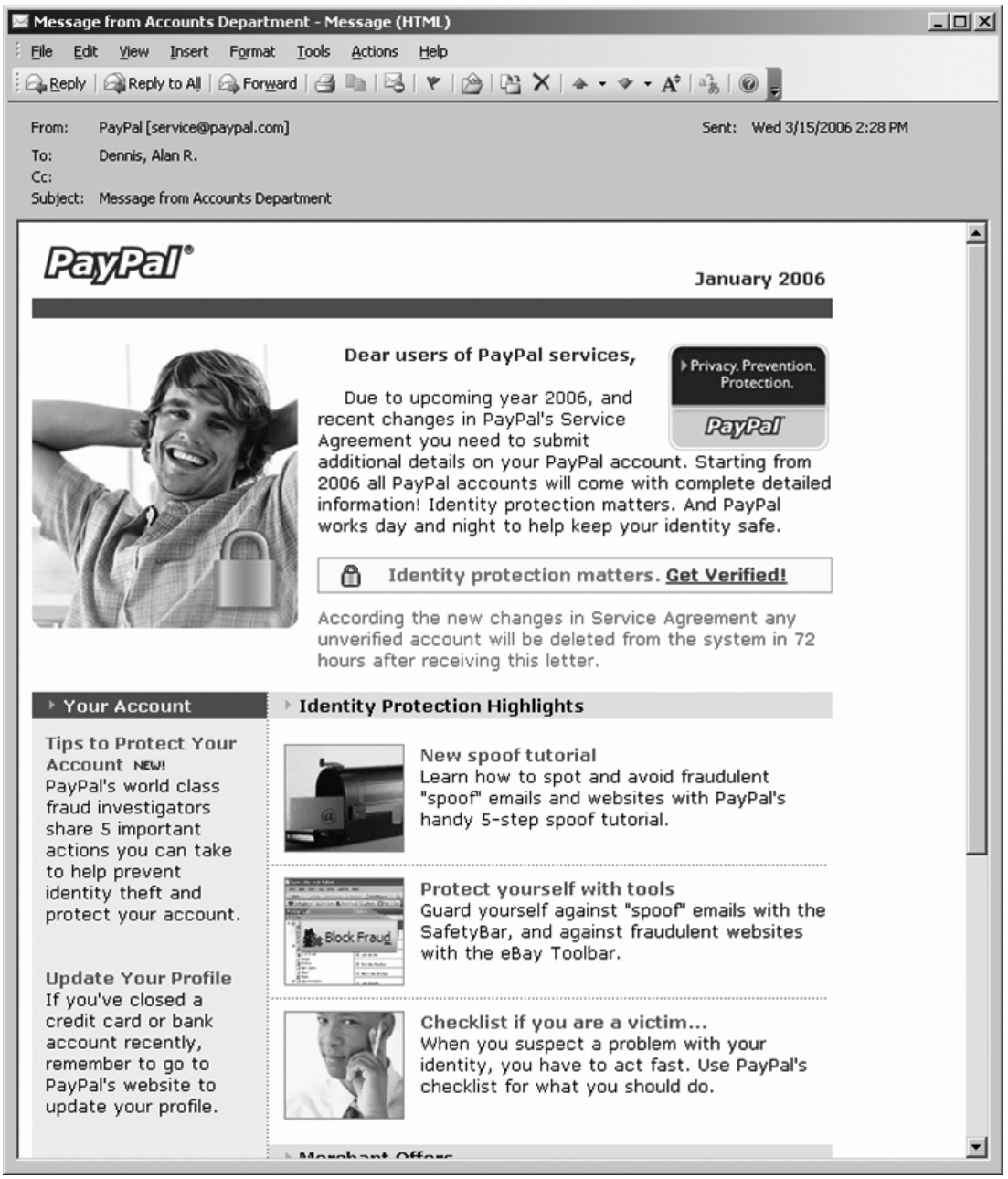


FIGURE 11.21 Phishing attack.