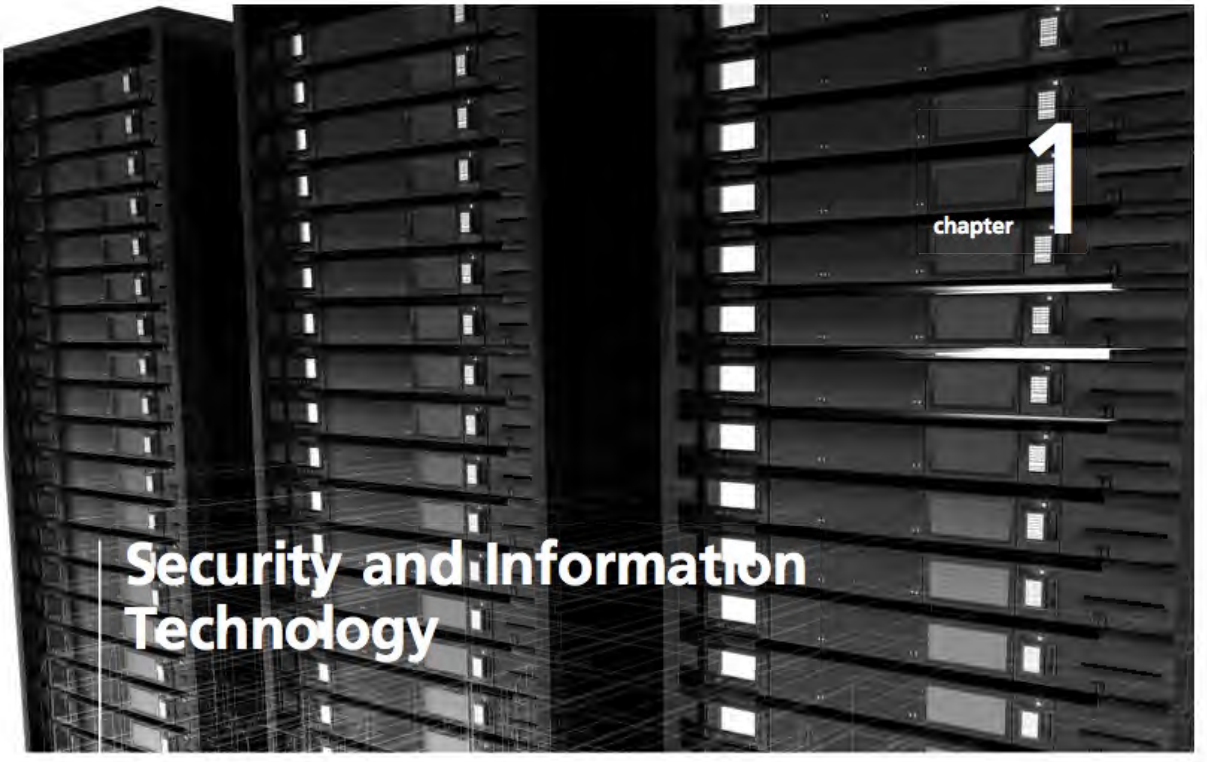

chapter 1

Security and Information Technology

After reading this chapter and completing the exercises, you will be able to:

- Define the nature of database and information systems security
- Identify the three main security objectives when protecting information systems
- Identify security threats
- Define and identify the characteristics of viruses and how they infiltrate systems
- Identify specific types of operational security and describe how to implement them
- Describe the information security life cycle
- Describe the multilayered nature of security architecture

Security In Your World

Joseph Anthony awakens on Monday morning refreshed and ready to begin his five-day workweek. On a typical weekday morning, Joe eats breakfast, watches the news, and gets dressed before walking out the door. Little did he know that this particular day would soon begin one of the worst cases of "Monday mishaps" ever experienced in the history of all Mondays.

As Joseph sits up in bed, he realizes that his alarm did not go off and he has no power at all. Remembering that there was quite a large storm in his area overnight, he assumes this is the reason for the power outage and does not give it a second thought. He glances at his watch sitting on the nightstand and notices that he is running a bit late, so he jumps out of bed, scurries to get dressed, and heads for the door.

With no power to make himself breakfast, he decides to stop at the local café to grab a doughnut and a coffee on his way to work. He orders his breakfast and swipes his debit card to pay and be on his way. To his dismay, the employee informs him that his card has been denied and that he needs to pay with cash. Insisting that the machine is incorrect, Joe asks the employee to try it again, but to no avail.

Concerned about his bank account, Joe runs back to his car and pulls out his cell phone to call his bank, yet his cell phone seems to have been deactivated. Frustrated and hoping that this was due to the storm the night before, Joseph proceeds to work and plans to call the bank from there.

Joe arrives at work, parks his car, and locates the key card that he uses to enter his employer's secure building. He swipes his card into the door's scanning device and is denied access to enter the building. Luckily, he sees a colleague heading for the door and so he asks her to let him in.

Late for work and irritated, Joe makes his way to his desk and sits down at his computer. He picks up the phone to call the bank while attempting to log into his computer. With the phone on his ear, he tries his Windows password and is denied access. He tries again and again, but eventually his repeated attempts lock the system. At this time, the bank representative answers the call and Joe explains the difficulty that he had with his debit card that morning. The bank teller asks him for his account number in hopes of identifying the cause of his problems, but when he provides her with the information, she informs him that no such account exists at their bank! They try using his Social Security number and his address, but nothing is found and so the teller recommends that Joe drive to the bank in person.

Now angry at the chain of events that have occurred thus far, Joe hangs up the phone to call his company's IT department in hopes that they can reset his password so that he can log into his computer and e-mail his boss to inform her that he must

(continued)

leave work to run to the bank. An IT representative answers the phone and asks Joe for his employee ID number. After a few minutes have passed, the IT representative explains that Joe doesn't have a user account recorded in the system. Just when Joe thought things couldn't get any worse, he was told the most frightening news he had heard this whole horrific day. According to the employee records database, Joe is not even listed as an employee of the company! Is this all a dream? Has Joe lost his mind?

We have seen this scenario in movies and on TV, and although this seems a bit extreme, it is not as far-fetched as one might imagine. In today's digital society, our identity exists as a compilation of various types of accounts, and almost every bit of personal information is stored in some type of database. Our bank accounts, medical accounts, employment records, utility accounts, phone records, car registration, income tax information, and even our Social Security numbers are all examples of important information that is stored within our current databases. The loss of our modern databases would result in the loss of our identity similar to what Joe has experienced. The obliteration of our major accounts would result in the eradication of our existence in today's digitally reliant world.

There are two types of events that are most likely to cause a scenario similar to the one described above: cyberattacks and natural disasters. This chapter will cover the measures that can be taken to detect and diminish the probability of both. Database security is defined and our enemies are identified. There is no such thing as guaranteed security, but armed with the knowledge provided in this chapter, major risks can be minimized.

Why Database Security?

Although databases can be configured within an organization to provide only local data storage and retrieval, most databases today provide access that spans several networks, and across the world. More time is spent communicating through computer networks than any other medium available and most of the transactions conducted online involve some type of database. Through e-mails, instant messages, text messages, and tweets, individuals are able to conduct business, monitor finances, maintain friendships, nurture relationships, and supervise their children. Society has progressed from one that accepts technology to a society that relies on technology to thrive.

This progression in technology has attracted an exponentially growing poll of network and database users, each outfitted with personal and business-critical information that is available through public networks. For intruders seeking destruction and monetary gain, this is a very enticing notion and has resulted in an unprecedented number of database intrusions throughout the entire globe.

Although the majority of our security efforts seem to be geared toward protecting individual and business assets, there is much more than personal identities and business finances at stake.

Advancements in technology have enabled individuals to use network databases to preserve their most critical infrastructures. Water supplies, electricity grids, and gas and oil production processes depend on a computer network to thrive. A breach to one of these data storage structures could have a disastrous impact on our livelihood. The view of a network intruder has transformed beyond the lonely hacker residing in the basement of his parent's home to a collection of hacking brigades professionally trained by the militaries of nations around the globe.

Database security has become an issue of global importance, and as networks become more sophisticated, so do intruders. Although we can never be 100% secure, we need to make every effort to minimize risks and to stay one step ahead of intruders, because the impact of a breach could be catastrophic on so many levels.

A Secure Data Environment

Deploying multiple layers of security within critical database environments is the most effective approach to minimizing the risk of a data breach. It is quite a simple concept to comprehend: if multiple layers of security are applied to a data storage environment, then intruders will have a more difficult time accessing the data. In multilayer-secured environments, an intruder who might compromise the first layer will have to find a way to bypass the second and even a third in order to obtain access, making intrusion more complex and time-consuming.

Consider a scenario in which a database administrator wants to protect his network from malicious e-mail attachments. He develops a training to teach users about the dangers of e-mail, hoping to educate them to identify the signs (such as file extensions) of dangerous attachments. If this is the only measure taken to ensure that attachments do not pose a threat to a network, then one forgetful user can cause major damage to a system.

If a second layer is added to this strategy, such as the implementation of a filter placed on the exchange server to block and quarantine certain well-known malicious e-mail attachments, the risk of a security leak is lessened. In this scenario, the attachment must fool the exchange server by changing its filenames *and* a forgetful user must download the attachment from the e-mail account.

Even a third layer can by applied, such as a firewall that is configured to deny certain types of traffic from entering the network, further lessening the risk. For a breach to occur now, the firewall, exchange servers, and user all must be fooled into allowing the attachment to intrude upon the network. Therefore, the more security layers that we can apply, the more secure our environment will be.

There are three main layers of security that need to be addressed in order to achieve a multilayered-secure data storage environment: database security, computer security, and network security. Each of these layers offers an intruder a potential way to enter the system, so to effectively secure a database, one must secure the database environment as well as the database itself.

Database Security Database security is a set of established procedures, standards, policies, and tools that is used to protect data from theft, misuse, and unwanted intrusions, activities, and attacks. Database security deals with the permission and access to the data structure and the data contained within it. Tools used to secure the database are typically included and configured within the installed database software packages, but the abilities of these packages vary by vendor (e.g., Oracle, MySQL, Microsoft SQL Server). Database security will be covered in more detail throughout the book, but the most common features made available by vendors to secure

the database include database-level access control, database-level authentication, and data storage encryption. Granularity varies from vendor to vendor and some vendors also offer additional software packages customized to secure their particular database management application.

Computer Security Although database security deals directly with the security assurance of the data structure and its contents, when considered independently without addressing its environment, database security is irrelevant. The security of the layers of the database environment includes the hardware and software upon which the database is installed, and is just as important as the security of the data structure itself. **Computer security** is a set of established procedures, standards, policies, and tools that are used to protect a computer from theft, misuse, and unwanted intrusions, activities, and attacks. Computer security is typically defined by the operating system used on the computer. Security features are available within the operating system and can be expanded upon using third-party applications. Common computer security features include operating system-level access control, operating system-level authentication, application security, and hardware and software monitors and logs. The features vary by vendor and operating system edition. For example, Microsoft Windows Server 2008 offers many more security features than Microsoft Windows Server 2003.

Network Security From securing the entrance doors of a building to applying firewalls to block traffic coming into the network, each and every security effort plays an important role in the overall security of a database. Network security is the outermost layer of the database and arguably the biggest security concern. Consider the potential danger for a man who secures the title to his car in a locked glove compartment, but never locks his car doors. An administrator who does not secure their network environment is essentially leaving the "car unlocked" for all to explore. **Network security** is a set of established procedures, standards, policies, and tools that are used to protect a network from theft, misuse, and unwanted intrusions, activities, and attacks. Achieving a reasonably secure network requires a combination of hardware and software devices that may include firewalls, antivirus programs, network monitors, intrusion detection systems, proxy servers, and authentication servers.

Database Security Objectives

Security measures should keep information private from an outside view, be steadfast in their efforts to maintain the consistency of data, and ensure that resources remain at a high degree of availability. The key to achieving an effective data security architecture relies in an organization's efforts to maintain the confidentiality, integrity, and availability of its environment. As illustrated in Figure 1-1, these terms are commonly known as a security model called the C.I.A. Triangle because of the three entities that connect to form one concept.

Figure 1-1 C.I.A. Triangle
© Cengage Learning 2012

Confidentiality Confidentiality refers to the efforts taken through policy, procedure, and design in order to create and maintain the privacy and discretion of information and systems. For a system to provide confidentiality, it needs to do two things:

- Ensure that information maintains its privacy by limiting authorized access to resources
- Block unauthorized access to resources

The confidentiality of resources on a database system is protected through the use of authentication and access controls. For example, an administrator can use a person's login information to restrict that person's access within a database or a database environment and therefore maintain confidentiality.

Confidentiality is not always left to the administrator's own discretion. Many of the measures that we take to ensure confidentiality are required by federal and state laws. Depending on the nature of the organization, there are state and federal laws that set the rules for privacy and confidentiality. These laws require network administrators to uphold a certain level of security to ensure the privacy of information. For example, the **Health Insurance Portability and Accountability Act (HIPAA)** defines strict laws for health institutions throughout the United States. HIPAA laws ensure the security and privacy of patient records by dictating the way files are accessed, stored, and transmitted on a network.

Confidentiality is an important goal to achieve within security efforts. A breach in confidentiality could result in a number of disastrous effects. Some of the major repercussions include identities being stolen, business's trade secrets being exposed, disaster cleanup expenses incurred, tarnished reputations, and, as mentioned earlier, even critical infrastructure failures.

Integrity Integrity refers to the efforts taken through policy, procedure, and design in order to create and maintain reliable, consistent, and complete information and systems.

Integrity within a database refers to the reliability, accuracy, and consistency of the data stored within and retrieved from the database. A database's integrity is protected by preventing both unauthorized and authorized modifications, whether accidental or deliberate, that might cause the database storage or retrieval to be unreliable and inconsistent. Integrity is the most difficult item to measure because the corrupted data are not necessarily missing, just modified.

A number of checks and balances are necessary to find changes or flaws that exist throughout a database. This process is also known as *auditing* and involves an audit professional who looks for discrepancies within the system by checking data against older backed-up versions of that data. Database auditing will be discussed in further detail later in the book, but as you can imagine, the more complex the database, the greater the auditing task. Integrity is a very important characteristic of a database, and if unsuccessfully implemented can result in system failures, unreliable data, flawed programs, and poor performance.

Availability Availability refers to the efforts taken through policy, procedures, and design to maintain the accessibility of resources on a network or within a database. These resources include, but are not limited to, data, applications, other databases, computers, servers, applications, files, drives, shares, and network access.

In order to protect our resources, we must identify those things that pose a threat to the availability of our databases, assess the level of threat that they pose, and plan an intervention accordingly. Common potential threats include technical failures (e.g., a

defective or broken device, a flawed program or piece of software), natural disasters (e.g., floods or fires), intrusions (e.g., viruses, Trojans, and worms), and users (e.g., accidental or intentional harm).

Unlike confidentiality and integrity, a business cannot operate without availability, because not having access to its most critical tools leaves it disabled and unable to complete the simplest of tasks.

There are many who take the availability of resources for granted. They don't realize the impact that a missing file or application can have on productivity until it is no longer available. Even with the best plans in place, one cannot assure 100% availability of all network resources all of the time. What is certain is that the longer a network or data resource is unavailable, the greater the loss that will incur. Proper identification and planning is the key to keeping a network available and a business thriving.

Who Are We Securing Ourselves Against?

Despite the expensive tools that can be purchased to aid in the protection of our data, we are defenseless against dangerous intruders without a complete understanding of those things that pose a threat.

It can be said that an uninformed security professional is similar to a sight-impaired watchdog. Both are insecure and eager to protect, having all of the tools necessary to keep the environment safe, yet they lack the ability to make the distinction between an intruder and a visitor. As a result, they blindly attack everything that enters their environment. Databases that are too restrictive are just as ineffective as those that are too accessible. Databases that are given too much access can lead to issues with security, integrity, and privacy, while those given little to no access can lead to frustrated and ineffective users. A healthy balance requires an understanding of what we need to protect ourselves against. There is a common misconception that those who pose the greatest threat to our assets are those on the outside attempting to break in. As you will learn from this section, there are a greater number of threats on the inside of a network, ready and able to destroy the resources that you work so hard to protect.

As you review the following material about the dangers that pose threats, keep in mind that the most fundamental goal for an intruder is to obtain access to the network. With access on a private network, intruders have the ability to explore and further exploit. Having access to a network provides intruders with the potential to locate and access the critical database systems. Each and every threat that is discussed in this chapter can potentially lead to the loss of the confidentiality, integrity, and availability of our database system.

Hackers

Although the term **hacker** has taken on negative connotations over the years, by true definition, it refers to those who have mastered the firmware and software of modern computer systems, and enjoy the exploration and analysis of network security with no intent to intrude or cause harm. A **cracker** refers to those individuals who break into our networks without authorization with the hope of destroying and/or stealing information. The mass media has

played an enormous role in the confusion of these two terms, so they are often inaccurately used interchangeably.

Despite the general public's perception of the term *hacker*, learning to define these terms correctly is important in understanding the threats to our systems. To further clarify the difference between hackers and crackers, a new classifying system has been created and is currently in use today. This system has introduced new terminology to provide clear definitions that accommodate the changes and growth within this field, as well as naming these groups of individuals based on their motivation for exploration of networks. Table 1-1 contains information regarding the different types of online intruders.

Intruder type:	Definition:	Example:
White hat	Ethical hacker: a hacker that uses extensive experience and knowledge to test systems and provide security consultation to others; white hats pose no threat to our network systems	A security consultant hired by an organization to use different methodology to attempt to hack into their system with the intention of testing the security of that environment
Grey hat	An individual or groups of individuals that waver between the classification of a hacker and a cracker; grey hats sometimes act in goodwill and other times in malice	An individual breaks into the *Wall Street Journal*'s network and leaves notes within the system's database to alert the network team of the vulnerabilities that exist; on a different occasion, this same individual breaks into *The Boston Globe*'s human resource system to obtain sensitive employee information for personal gain
Black hat	Someone who breaks into computer networks without authorization and with malicious intent; black hats are responsible for the theft and destruction that affect our systems	An individual breaks into Walmart's point of sales system to obtain credit card information from consumers in hopes of financial gain
Hactivist	Refers to hackers and crackers who use their extensive experience and skill to use networks to share their ideologies regarding controversial social, political, and economic topics; hactivism can be malicious in nature due to the methods by which hactivists attempt to place further attention and emphasis on their cause	A group of political extremists use their computer know-how to hijack MSNBC's Web site, altering the site to display messages that disparage mainstream media
Script kiddie	Refers to an amateur cracker that uses programs and scripts written by other people to infringe upon a computer network system's integrity; script kiddies are especially inexperienced, and attacks are often experimental in nature	An individual searches for and downloads a cracking tool found online and uses it to haphazardly gain access into an organization's network and steals information

Table 1-1 Types of online intruders

Social Engineers

Social engineers use human interaction to manipulate people into gaining access to systems, unauthorized areas, and confidential information. They often build trust with an authorized user of a network, and through deception and trickery they convince these people to break normal security policies. One of the most common ways that intruders gain access to systems is by asking others for their passwords. By nature, human beings desire to help one another. This is one of many characteristics that facilitate the success of social engineers.

Imagine that you are working in a building that requires employees to use special key cards to enter through the main doors. Returning from lunch one day, you are walking toward the entrance and you notice a well-dressed man carrying a number of files, his briefcase, and a bag from the local fast food restaurant walking directly behind you. As you approach the door and swipe your card, you hear the man clearly struggling to locate his access key card. Do you leave him standing outside, or do you hold the door for him out of politeness and a desire to help one of your fellow human beings?

Computer Users

Well over half of the security breaches that occur on a network involve our own network users. Lack of education and a total disregard of policy are two major contributors to outside intrusions. Users are the weakest link within an organization, and the best security architecture in the world can't save us from a haphazard user. The circumstances that provide potential for user error are endless, so it is important to learn as much as you can about the users on your system and educate the users through continuing education. Here is a list of the most common errors made by users on a network:

- *Poor habits*—leaving computers unlocked and unattended while using the restroom, attending meetings, going to lunch, or visiting colleagues.
- *Password error*—choosing easy-to-guess passwords; writing passwords down on sticky notes or in notebooks and storing them in plain sight on desks, under keyboards, or on top of monitors.
- *Disregard for company policy*—visiting unauthorized Web sites and downloading unauthorized software in the process; attaching unauthorized equipment to their PCs, like USB (Universal Serial Bus) devices and external hard drives; logging into the company remotely using unapproved personal laptops and computers.
- *Opening unknown e-mails*—viewing risky attachments containing games, greeting cards, pictures, and macro files.
- *Inappropriate disclosure*—giving out information over the phone and falling prey to social engineering.
- *Procrastination*—failing to report computer or network issues in a timely manner.

Computer-literate users can be just as dangerous (if not more dangerous) as those users who are new to network and computer systems. These users generally take more risks and have less regard for security policy. They change settings and configurations, disable programs, and find shortcuts to basic security measures. These types of users tend to install more sophisticated programs, causing network-wide compatibility issues and opening doors for intruders to enter the network. Unfortunately, due to fear of exposure, these types of users are also less likely to call technical support when things go wrong. This can potentially

cause major problems that are extremely difficult to detect and are detrimental to the integrity of a network.

Another type of user that we need to be aware of is the angry and disgruntled employee. Having a computer-literate disgruntled employee on your network can pose as great a threat as unknowingly hiring a security cracker. These users are more likely to intentionally cause destruction and abuse their files and access rights. These users pose a great threat because they already know the organization of the network, are aware of how to find confidential information, and have most likely already gained trust within a circle of their peers. These intelligent disgruntled workers pose a threat to the confidentiality, integrity, and availability of the database and network as a whole because they can attack from the inside, and defense is much more difficult at this level.

Network and Database Administrators

The network and database administrators and their team of specialists are not often viewed as threats to the network that they run, yet several problems can be caused at this level of administration. This department creates, maintains, manages, and monitors the entire database and/or network architecture, leaving quite a bit of room for error. As diligent and hard-working as they may be, they are only human, and they do make mistakes. Unfortunately, the mistakes made at this level are almost certain to have consequences for the integrity, availability, and reliability of the network.

Depending on the size of an organization, a great amount of change can occur over the period of a year. As employees are hired, fired, retired, promoted, and demoted, networks and databases must accommodate and adapt. When employees are hired or fired, computers and user accounts are added and removed from the network and permissions are added and removed from the database. As business is flourishing, data within the databases are changing as well, so the network and database that you have today may not be the same that you will have in a year.

This dynamic and ever-changing nature of the data environment can cause security flaws to be created in places where once no flaws existed. Therefore, it is important that virtually every component on the network is audited and reassessed on a regular basis, searching for potential, newly created security concerns and need to be frequent, comprehensive, and complete.

Overlooking security flaws within a system can cause a great deal of havoc and unfortunately it happens more often than administrators would like to admit. One prominent mistake that network administrators make much too often is forgetting to remove a user's rights and account credentials from the database environment. This is a common mistake that can have devastating results. Imagine the consequences of leaving network access available to a disgruntled exemployee who was recently fired from an organization.

So, although we rely on our network administrators to keep our environment confidential, reliable, and secure, even they can contribute to the security issues within a network.

The Internet

The environments to which we are exposed can be as threatening to our assets as the people within them. Currently, there are approximately 2 billion users and 100 million Web sites residing on the Internet. Over 75% of people within the United States have

Internet access and most of these people use the Internet to socialize, learn, and buy or sell goods.

The popularity of online education has increased tremendously over the last five years and Web sites like Twitter and Facebook are recruiting Internet users by the millions. We are locating lost family members, reuniting with old friends, finding soul mates, and developing and maintaining friendships. We have redefined the term socialization. Although many are undecided about the impact that technology is having on our culture, there is no denying that virtual has gone viral.

Sales and marketing groups have also learned that there is great potential within online markets. Billions and billions of dollars in transactions are being made online each year. Local companies are now able to reach customers on the other side of the globe, and for many products, online marketing techniques have surpassed our most lucrative traditional ones.

Yes, the use of the Internet has offered great advantages to both individuals and businesses around the world; it seems that everyone is benefiting from our virtual society. Unfortunately, with the good comes an equal amount of bad. The threats posed on the Internet are greater than ever before as intruders become more sophisticated and their numbers continue to increase. Unfortunately, the nature of the Internet makes it a main attraction for thousands of cybercriminals each year. Activities conducted online leave you susceptible to hacking and make you vulnerable to the well over 600,000 viruses that are scanning our networks today. Businesses, information, and assets are greatly threatened by the Internet and our social interactions are resulting in the greatest number of identity thefts in our recorded history.

Listed on the following pages are the most common user tools available on the Internet, along with the potential threats that they pose.

Web Pages Surfing the Web and/or Web browsing in and of itself can be a dangerous activity that poses great threat to your data. **Web pages** are documents containing a specific programming language, such as HyperText Markup Language (HTML) or JAVA, etc., that are designed to be viewable on the World Wide Web. The code contained in a Web page document usually has two purposes:

- To inform the Web browser how to display the document
- To inform the Web browser the way that it and the Web document should react to certain user responses (e.g., clicking a button, submitting a form, clicking a link etc.).

Anything that involves programmed code can be manipulated.

Some Web sites are hacked into, also known as **hijacking**, and rewritten to react differently to users than the original Web site designer intended. These hacked sites can be rewritten to distribute malicious code upon user activity (such as downloading), or to redirect the user to a site that was built by a hacker. The intent here is to spread embedded malicious code.

Malware is an abbreviation for the term malicious software. **Malicious software** is programming code written and used by unauthorized intruders to perform a certain task on a computer.

These tasks are often intended to be harmful and destructive. Malware can be written using virtually any programming language available, so the only limitation to the impact that malware can have is an intruder's creativity.

Malware will be discussed in detail later in the chapter.

Consider a user surfing the Web who encounters a site that generates a Windows alert, or a pop-up window, which requires the user to click OK or some other labeled button in order to continue surfing. This simple window can be an invitation to disaster, programmed to initiate a response and intended to cause harm. An attacker can program the button so that when it is clicked, the Web site either begins attaching malware to the user's PC, or redirects the user to a hacker-owned Web site in hopes of obtaining the user's personal information.

Hackers build some Web sites made to look identical to other popular sites in hopes of drawing in a user. The legitimate site's address can be cloned and used to even further convince the user that the fake site is real. This is called **spoofing**. A spoofed Uniform Resource Locator (URL) is used to fool a user into believing that the site is a legitimate or well-known site, such as Yahoo! or Google.

Spoofed URLs and fake Web sites are often created to trick users into providing sensitive information like passwords or account numbers, and they are very effective tools. Spoofing a URL can be as easy as registering a domain name that is a slightly misspelled company's URL (e.g., Gogle, Yaho). After all, the user has no indication that the site is not real.

These Web sites can be accidentally stumbled upon by a user while surfing the Web, yet some Web sites are more vulnerable than others and have similar characteristics. Table 1-2 contains characteristics of Web sites that attract intruders and malicious doers.

Characteristic	Example	Reasoning
Illegal and of moral suspect	Darknets, pornography sites, and warez sites	The people who run these sites are less likely to report an incident of hacking
Social sites	MySpace, Facebook, Blogger, and Twitter	These sites are often populated with millions of inexperienced users; inexperienced users are more likely to make errors in judgment in terms of security, offering intruders a large number of potential hosts to spread or store viruses
Newsgroups and technical forums	Usenet and BinSearch	These sites are often perceived as credible knowledge bases, so users are more likely to click links provided in the forums

Table 1-2 Common characteristics for dangerous Web sites

Web Browsers All network communication is based on the same basic principles. A request to obtain a resource is sent from a source machine or application (client), to a destination machine or application (server). The request is received and processed and the client's privileges are checked to determine allowable permissions for the requested resource. Once approved, the requested resource is packaged and sent from the server to the client. Each step is handled using a set of standards or protocols that determine the who, what, when, where, and how of the communication process.

A Web browser is an application that acts as the interface for the Internet, providing a way for people to interact with and view pages on the World Wide Web. Web browsers are responsible for aiding the process of sending and receiving user requests. Web browsers pose another threat to users, because just like a Web page, they have built-in programming languages that can be manipulated. As noted earlier, anything with programming language can be attacked.

One very important job of a browser is to forward requests made by users from the source machine to the destination machine or application server. User requests occur when a user types a URL or address (e.g., *http://www.yahoo.com*) into the address field of a browser or when a button is activated on the form. When a user types an address, she is essentially requesting permission to view the Web page residing at that address. The browser reads this address, and from the information given by the user, the browser sends the request to view the page to the main Web server that holds the Web page. URLs can be manipulated and intruders can use them to obtain access to database information. Appending malicious code onto a database-directed URL is one way that intruders manipulate or trick a database into sending confidential information. This is called SQL injections, and will be discussed in much further detail in Chapters 7 and 8.

When a user fills out a form on the Web and activates a submit or send button, the browser sends the request to a separate server that contains information about what to do next. Typically, forms available on the Web are intended for communication with a database of some kind. Either a person is filling out the form to sign up for an account online (input data into a database), or to search for specific products on a Web site store (retrieve information from a database). These forms can also be using SQL injections to retrieve confidential information or perform unauthorized manipulation of the database. Again, SQL injections are discussed in further detail in Chapters 7 and 8.

The **hypertext transfer protocol (HTTP)** portion of a Web address informs the browser what protocol is used to send the request for the Web site. In our example, the Web server is told to use HTTP protocol, or port 80 (the specific port used for HTTP to send the request forward). HTTP is also used to determine how Web form information is transferred to the application server, or to the database HTTP to be configured to send all form-related data (the information that the user input into a Web form) to the database by appending the input to the URL. If this configuration is set, the user's form information can be intercepted and used by intruders to log into a database. For example, let's say that you have the capability to log into your school to check your grades online. Once you enter the address for your school (e.g., the URL), the browser sends the request to your school's Web server and returns to you a form that instructs you to insert your username and password. If you were to log in, you essentially send a request to your school's database server, asking it to retrieve the grade information for the username and password that you input into the form. If HTTP is configured to append the

information that you input (username and password) onto the URL, this information can be intercepted by an intruder before it reaches the application server. In this case, you might provide the intruder with a username and password to the school's database and enable him to send unauthorized requests using your credentials.

Besides the SQL injections, there are other ways the URLs can be manipulated for the purpose of obtaining access to a network. The Web address, *www.yahoo.com*, tells the browser where to find the main server to send this request. Before the request can be sent, the URL needs to be changed into an Internet Protocol (IP) address to make it routable across the network. The browser needs to first send the address to the Domain Name Server (DNS), which holds a database of domain names and their respective IP addresses. The DNS looks up the IP address for *yahoo.com* and sends this IP address to the Web server. The Web server can now send the request to the main server holding Yahoo's main Web page.

It is possible for the DNS server to be attacked, an intrusion called **DNS poisoning**. When DNS poisoning occurs, a cracker gains control over the DNS server and changes the domain name's respective IP address, redirecting requests to sites that the cracker has built and maintains. These fake sites will look and feel identical to the Web sites the user has attempted to access. Therefore, the user is fooled into providing their **personal identifiable information (PII)**, which is personal information that identifies a person or entity, and may include information such as names, passwords, Social Security numbers, etc. These fake sites may also hold malware, which the user unknowingly downloads just as described in the preceding spoofing example. The whole process takes place without the user suspecting a thing.

A Web browser also contains menus which allow organizations or individuals to customize and configure security and personal settings. These settings can be compromised using malware and cracking techniques, as well. An example of a browser setting that can be changed by a user is a home page setting. A browser's home page setting allows a user to customize the main **startup page** (the Web site that is displayed when the browser is started). In a home environment, a user can set his browser startup page to be directed to his favorite news Web site or search engine. Within organizations, this setting is often configured to be directed to the company's Web site.

Common Web Browsers Some common Web browsers include Internet Explorer, Firefox, Netscape Navigator, and Opera. The most common of these Web browsers is Microsoft Internet Explorer (IE). IE comes installed on Windows machines by default.

A good rule of thumb to consider with security and the Internet is that the more popular a system is, the more likely it is that it will be violated. Most crackers want to do as much damage as possible, so they attempt to crack into systems and sites that attract the most people. The more people there are who use a system, the more available hosts there are to carry the virus. Therefore, unless Microsoft changes their default Web browser, IE will be more vulnerable to attacks than any other Web browser. IE has been the victim of many attacks throughout the history of the Internet. These attacks have been accomplished by crackers finding security weaknesses and vulnerabilities within the application. Each time a browser is attacked, Microsoft locates the weakness and creates an update to the browser to patch that vulnerability.

Patches will be discussed later in the chapter.

These patches are normally included as part of a Windows Update, yet in large and enterprise organizations where patches are centrally managed, it is the administrator's responsibility to install or apply the most recent patch. If an old version of the Web browser exists, or updates to the Web browser are not completed on a regular basis, the Web browser can be left vulnerable to attacks. Web browsers are also vulnerable to hijacking as well. This is also the case for database systems. If a database administrator does not update the database software as new updates are made available, the system can be vulnerable to security breaches.

Misleading Applications

The fear of the many viruses and threats floating through cyberspace makes users vulnerable to fearmongering and phishing techniques, designed to trick users into downloading malicious code. **Misleading applications** are applications that deceive users into believing that their computer's security has been breached, therefore tricking the user into downloading and purchasing rogue antivirus tools to remove the bogus breach. The application often is very believable, and may look like common spyware and antivirus removal tools.

For example, the user performs some activity, such as clicking the link or installing the software, and once downloaded, the misleading application will hijack the PC and obtain personal information, destroy files, and collect data. Misleading applications are found throughout the Internet and are most often found in earlier shareware sites, social sites, and newsgroups. These applications can also hide in banners and advertisements, as well as in programming code of legitimate sites.

The victim essentially pays to have malware installed onto their PC, which robs them of their PII, retrieving the malware that was installed, thus compromising the reliability of their system. All of this is done without the user's knowledge, and they are left with a false sense of security that potentially leads to future threats.

E-Mails

Electronic mail has become one of the most common forms of communication in today's society. E-mail is used to send critical, sensitive, personal, and business information daily. E-mail has become vital to the success of businesses, and yet it poses the biggest threat to a network and the database environment. E-mail provides crackers with a simple channel of attack, and is the most common way that malicious code gains access into a business. Along with the basic vulnerabilities that come with sending and receiving e-mail, Web e-mail accounts carry the vulnerabilities that come with browsers. The most common threats to e-mail are attachments, phishing, and HTML code attacks.

Attachments Attachments are the largest threats to an e-mail system. Although we can train ourselves and our users to carefully consider attachments, crackers make it very difficult to identify a fake attachment. Crackers can send an endless array of malicious programs

through attachments provided in an e-mail, yet in order for these intrusive programs to be effective, the user must download and initiate them. Crackers often use attachment names and/or file extensions in an attempt to gain trust.

Intruders understand that attachments coming from strange e-mail accounts are not likely to be opened, and file extensions of common virus types are likely to be blocked, so they use techniques like e-mail spoofing and strategic filenaming, such as pjb.jpg.exe, to help their cause. Spoofing e-mail addresses, like spoofed URLs discussed earlier, fools a user into believing that the sender is a friend or a colleague.

Spoofing an e-mail address can be as easy as using a stolen e-mail address from the "from" and "reply" fields of an e-mail account. Like URLs, spoofed addresses are very effective. Imagine that a user receives an e-mail that she thinks is from a colleague, and this e-mail contains an attachment with a filename related to the nature of her business; the user is very likely to open the attachment.

The cracker only needs to fool one user into opening an unknown attachment for an entire system to erupt in chaos. Imagine this scenario: A cracker sends an e-mail containing a malicious attachment to 1,000 people. One of the 1,000 people opens this attachment, initiating code that sends a copy of their contact list back to the cracker and begins to attack the user's machine. Most important, the virus has gained access to the network, allowing it to propagate, affecting the network and the database environment, and affecting each and every machine (including the servers) along the way. With each machine affected, the intruder receives a contact list of e-mail addresses. Now the intruder has broken the integrity of the network and potentially the database environment, obtaining potentially thousands of e-mail addresses to spoof, to gain trust of other users, improving the possibility that the next attack will result in many more opened attachments. This is a dangerous, never-ending cycle.

Phishing Phishing is the attempt to obtain PII from people through the use of spoofed e-mail addresses and URLs. Phishing relies on spoofing to gain trust. While spoofing is the act of cloning accounts, URLs, and IP addresses, phishing is the act of trying to fish information out of people.

Imagine receiving an e-mail from your company's IT group informing you that your account password was about to expire unless you changed it immediately by using the link provided in the e-mail. Without making a call to the IT group, you might fall prey to giving a stranger your system username and password. This can be disastrous for your organization as well as your professional career.

Phishing can also include the act of convincing a user to click a link, therefore redirecting the user to a cracker-owned site where attacks can then ensue. A very effective phishing technique used to redirect a user is through the use of fake holiday and birthday card e-mails. Sending cards through e-mail has become a grand gesture around the office. When an individual sends another individual an electronic greeting card, an e-mail is normally sent to the recipient that provides them a link to click to view the greeting card. These electronic card e-mails can be spoofed, and phishing can convince users to click a link provided in the e-mail. Instead of receiving a birthday wish, the recipient of the bogus card would be led to a cracker-built Web site where exploitation is initiated.

Web-embedded HTML Today, it is not necessary to use Webmail to send and receive HTML-based messages. E-mail clients, even local, have advanced from simple text files to colorful formatted documents that hackers use to exploit the programming code. HTML allows e-mail to be formatted using font, colors and styles, as well as many other formatting features available in a word-processing application. It can also include scripting languages such as JavaScript and active content such as ActiveX controls.

As said previously, anything that contains programming code can be manipulated. These e-mails have been and continue to be exploited by viruses, because malicious software can be created using scripting language and active content. In cases like these, in which malicious programs can be embedded directly into the e-mail's programming code, users do not have to initiate the virus, so they cannot be blocked as easily as attachments. Intruders do not have to rely on users to download risky attachments or click unfamiliar links. Users only have to read their e-mail for the malware to attack. As they open and read their mail, the malicious code is obtaining information, corrupting files, and forwarding e-mail addresses to the individual who created the virus.

Instant Messages

Instant messaging has become more than just a way to connect and socialize with a network of friends and family members. It is used much more frequently throughout the culture of organizations to help employees meet business goals, and it provides a quick and convenient way for employees to communicate, share files, and even hold brief chat sessions. It is a great tool to increase a team's productivity and solidarity as a whole.

Instant messaging is also being implemented to improve customer service and build client relationships. It provides a way for customers to immediately chat with product representatives, to ask product questions, receive technical support, or finalize business transactions.

The evolution of instant messaging has made this technology much more attractive to crackers who are looking to obtain sensitive information and break into confidential database environments, and it provides a viable option due to its insecure and vulnerable nature. Instant messaging does not encrypt data on either file transfer or peer dialog. In fact, it circumvents the security architecture put into place to protect such resources. Instant messaging provides an ideal environment for phishing that involves spoofed buddy names and redirection techniques to lead users into the traps of the cracker's custom-built malicious code distribution sites.

Tweets

Twitter.com is a Web site that provides members with a miniature blog-like service where a person can post small messages (140 characters maximum) onto the Twitter site for friends and family members to see. These messages are most often updates of a person's activities or status throughout the day. A person can post a tweet using a cell phone, a computer, or any device that provides access to the Internet.

Tweeting has become quite popular over the last few years, and as with any other popular social site, Twitter is a haven for intruders looking to invade. Like instant messages, images and links can be included within a tweeted message, posing great threat to any unsuspecting Twitter member. Twitter accounts are quickly falling prey to phishing, spoofing, and redirection techniques. As was expected by security professionals throughout the world, Twitter is now the cracker's new playground.

Malware

As mentioned earlier in the chapter, malware is capable of performing harmful and destructive tasks on the computers of those who fall victim to it. Because it can be written using so many different programming languages, an active imagination is all a cracker needs to wreak havoc on scores of users. Falling under the umbrella of malware are:

- Computer viruses
- Worms
- Trojans
- Spyware
- Adware
- Bots

Computer Viruses

Computer viruses are a form of malware intended to spread from one computer to another without detection. Viruses vary in degrees of danger; while some viruses only cause annoying disturbances to a computer system, others can copy sensitive information, corrupt files, delete data, or destroy entire systems. There are currently about 600,000 viruses floating around on our computer networks. The most common channel for virus transmission is e-mail.

Before we explore the different types of viruses, it is necessary to first identify the characteristics often found within malicious code. These characteristics are often built into the code of viruses, worms, and Trojans, and act as defense strategies to avoid detection and removal. Common characteristics and defense strategies include:

- *Self-encryption*—**Encryption** is the transformation of data by using sophisticated algorithms in an attempt to make the data unrecognizable. Viruses are often recognized by their **signature**, which is a pattern of characters that is identified for a specific family of viruses. Most antivirus programs use character string recognition scanning techniques to identify virus signatures on a network, so encryption can be a very effective method for viruses to avoid exposure. If a virus is able to disguise the way it appears to a network (pattern of numbers or characters), then antivirus programs searching for common characteristics of this virus will have a very difficult time recognizing the virus.
- *Stealth*—Every program that is installed onto a system makes changes to that system in some way or another. Whether these changes involve the use of memory or an increase in file size, once a program enters a system, a change is inevitable. Antivirus programs regularly monitor these changes by requesting information from the operating system (OS) looking for unauthorized altered system files, suspect registry entries, etc. If changes are found, antivirus programs look to attack. In order for a virus to remain successfully hidden within a system, it needs to cover its tracks by concealing any changes that it makes. This is what stealth code does to avoid detection. Stealth involves the interception of the requests from the antivirus programs and answers them instead of the OS. The malicious program provides information to the antivirus program, making it appear as though there haven't been any changes made to the system. This misleads

the antivirus program into believing that the system is clear of viruses. When users are unaware of their presence, viruses with stealth defenses can remain on a system corrupting files or collecting sensitive information for quite some time.

- *Polymorphism*—As mentioned before, antivirus programs look for signatures and patterns found within data. **Polymorphism** refers to the incidence of changing forms, or self-modification. Viruses that use polymorphism as a defense take encryption to another level by changing forms subsequent to each infection. This means that code that is polymorphic changes its signature each time it infects a file, and that each file infected is infected by a different copy of the original code. The dynamic and ever-changing nature of this code makes it the most serious and difficult threat to detect.
- *Residence*—The general term for a virus that requires users to initiate it by downloading a program or opening up an e-mail attachment is **nonresident virus**. Nonresident viruses cannot affect a system unless users make them active. Once active, these viruses attack software that resides on the hard drive. A virus that installs itself or takes residence directly in the main system memory of a computer is known as a **resident virus**. Unlike nonresident viruses, resident viruses are viruses that do not need users to make them active. Because these viruses reside within random access memory (RAM), they attack any and all programs that become active on the computer system. Resident viruses take advantage of the multitasking feature of a computer system, and therefore can infect programs at a much faster rate. These types of viruses usually result in the need to reinstall virtually every piece of software on the computer. Resident viruses are much more common because they are much more effective.

Classes of Viruses Just as with defense strategies, there are certain categories of viruses that can apply to some or many other types of viruses. This section explores these categories by defining the components of both a logic and time bomb virus, and describes the characteristics of spyware and adware.

- *Logic bombs and time bombs*—**Time and logic bombs** are general terms for viruses that do not become active until predetermined specific conditions are met. These viruses can lie dormant and undetected for many months before the effects of the virus begin to appear. A time bomb or time-delayed virus involves conditions in which the variables are times, days, or specific dates. For example, a time-delayed virus may be written to corrupt certain systems at regular intervals (e.g., every other Tuesday), or once on a specified date. The variables and conditions are predetermined and written within the code of the virus. A logic bomb has an endless amount of possible variables. These conditions normally depend on the environment in which the logic bomb resides. For example, a logic bomb might be set to corrupt data or systems only when a specific user logs into a network, or only if a specific name appears within a database query. These viruses are often thorough and purposeful in intent. Historically, these viruses have been written by disgruntled employees or people seeking revenge on a specific organization or individual.
- *Spyware*—**Spyware** is a general term for any software that intentionally monitors and records a user's computer and/or Internet activities. Spyware gathers sensitive information about users or businesses, compiles this information into some type of document, or **transmission packet**, and then forwards this information back to the

originator or creator for use as the creator sees fit. Spyware is most often downloaded and/or installed accidently from users surfing the Internet, yet it can be manually installed on an individual's computer as well.

- *Adware*—**Adware** is a general term for software that uses typical malware intrusion techniques to obtain marketing data or advertise a product or service. There are a few different types of advertisement software that fit the definition of adware. One form of adware is a software program that collects information for advertising and market research purposes through using attractive pieces of software that are offered to computer users at free or a reduced cost. Some adware, once installed, automatically generates advertisements in the form of pop-ups and Web site redirects. For example, a Web site might offer a free calendar and meeting-organizing software that, once downloaded, gathers information about the user's Internet activities for purposes of market research or personalized advertising. This type of adware can also be considered spyware because the software gathers information about your activities without your explicit knowledge. Another more invasive form of adware automatically generates advertisements on a person's PC once the free software is installed and used. The advertisements can be displayed using pop-ups and Web site redirects. This form of adware can take up a great amount of resources on the network or a computer system, causing the computer and/or Internet to become slow, and in some cases unusable. The effect on the computer and the network is dependent on the amount, frequency, and origin of the advertisement displays. Although some adware can cause slow networks and computers, adware by definition is not designed with the intent to harm a person's PC, a network, or a database environment.

Virus Types Viruses come in many forms and are often written to attack specific vulnerabilities, objects, and locations. The following section will review the different types of viruses that exist, as well as explore the vulnerabilities and objects of which they take advantage.

- *Boot Sector Viruses*—A boot sector virus loads itself into the boot sector of a computer's hard disk drive. The **boot sector** of a hard disk drive is an area of the drive that contains records necessary to the boot process of a computer. If the boot sector is compromised, then so is the entire boot process. Boot sector viruses normally infiltrate a system via an infected floppy disk left in a floppy disk drive. Due to advancements in technology, most new computer systems do not include floppy disk drives as a default firmware device, so boot sector viruses are not as common as they were in the past.
- *Macro Viruses*—A **macro** is a small program that enables users to automate a large number of repeated processes within a document. Macros are found within many Office-related applications, such as word-processing and spreadsheet-rendering programs. Macro viruses can either be attached to a macro or can replace a macro within a document. Macro viruses run automatically when the document containing the infected macro is opened or closed, and because people are often more comfortable opening and sharing documents of this type, macro viruses spread pretty quickly and easily, most often through e-mail attachments. Once the virus is initiated, it affects all documents of the same type on that computer or network drive. The affected documents include those that have already been saved, and any documents that are saved in the future. These viruses are easy to write and can be quite dangerous. They corrupt

data, install external software, and in some cases, like the infamous Melissa virus of 1999, they can take down entire networks of computer systems.

- *File-infected viruses*—Most file-infected viruses come from users on the Internet (e.g., downloaded e-mail attachments, instant messaging file transfers, and downloaded programs). Unlike macro viruses, which attack the data within a file, a file-infected virus will attach itself to an executable file that requires a user to run it before it can propagate and corrupt the system. File-infected viruses can begin as nonresident viruses that are downloaded from the Internet and require a user's action to become active. Once file infectors become active, they become resident viruses by copying themselves to the system memory. As with resident viruses, the effects are devastating to a computer system because the virus stays in memory.
- *Multipartite viruses*—A multipartite virus combines the characteristics of a boot sector virus with those of a file-infected virus. A multipartite virus can be obtained and activated in the same way a file-infected virus can be obtained. The difference is that the multipartite virus also infects the boot record of the boot sector on the hard disk drive, so that at the next boot, the virus will be distributed throughout the entire system.

Worms

Worms can be defined as self-replicating malware that are able to harness the power of networks and use this power in their attacks against the networks. Worms and viruses share a number of the same characteristics, yet, unlike viruses, worms do not need users to travel from one computer to another. In addition, worms propagate across networks, while viruses primarily propagate across computer systems.

A worm takes control of one computer, using weaknesses and vulnerabilities that were learned while maintaining control over its previous target computer. A worm uses information from the computer it is currently victimizing to look for vulnerabilities in nearby systems. Once these vulnerabilities are identified, a worm attacks these vulnerabilities and takes control of the next target computer. A worm repeats this pattern as it travels across networks, destroying systems along the way.

Elements of a Worm's Travel All worms, regardless of the type, share common elements that make up the self-replication process. As illustrated in Figure 1-2, the common process of a worm's travel across a network is as follows.

Figure 1-2 Elements of a worm's travel

This example refers to a worm traveling across the Internet, yet the same concepts apply to all other modes of travel.

1. *Find a weak target*—The enormity of the Internet leaves room for much vulnerability. The first element of a worm's travel across a network is to identify and take advantage of these vulnerabilities in order to find an open door onto the network and get closer to a potential database target. A worm looks for vulnerabilities such as those discussed earlier in the chapter (e.g., e-mail, file sharing sites, and insecure passwords) and then uses its knowledgeable code to obtain its first victim.
2. *Take control of the machine*—Once a worm gains access to its identified target through the use of e-mail, cracking passwords, or utilizing insecure file-sharing technologies, the worm installs itself onto the system. At this point, infection of the system begins. The level of harm that a worm can cause depends on the directives that the creator has included as part of its payload. The worm's **payload** is the component of the worm that contains a list of action commands, which the worm will follow for each machine that it encounters. These instructions can include steps for deleting files from a machine, opening back doors into a system, or forcing **denial of service (DoS) attacks** across a network. DoS attacks are concerted efforts made by malware to keep system resources busy, halting normal functionality. A **back door** is a path created to enable unauthorized access that evades all system and environmental security measures. Opening a back door into a system allows an intruder unauthorized and undetected access to a system or its environment.
3. *Interrogate the machine*—After the damage to the first target is complete, the worm hidden within the system begins searching for a new target. The worm looks for a new target by interrogating the system in which it currently resides. E-mail addresses and network configurations are pulled in search of a nearby computer to exploit. DNS queries are made to the nearest DNS servers, looking to obtain the IP addresses of nearby machines. Much information is discovered during this process.
4. *Testing a new target*—Once IP addresses, e-mail addresses, and network neighbor information is collected, the worm sends a series of packets to nearby machines to test their vulnerability and the effectiveness of the techniques that the worm used to obtain control of the current PC. Based on these tests, a new target is identified and the process starts all over again.

Types of Worms There are several types of worms that are categorized by the modes of travel by which these worms spread. Table 1-3 identifies and gives a brief description.

Trojan Viruses

A **Trojan**, or Trojan horse, is a form of malware that disguises itself and its harmful code. Trojans often hide within enticing programs such as software updates, games, and movies. Unlike viruses and worms, Trojans cannot replicate. Their primary purpose is to gain access into your system in order to obtain sensitive information, destroy important files, or to create opportunities for downloading and installing bigger and better threats, such as bots, onto your systems.

Types of Trojans Just as with viruses and worms, there are several varieties of Trojans. Trojans are categorized by their purposes. In this section, we will examine the different types of Trojans that exist and explore the damage that they have the potential to cause.

Worm types	Description
E-mail worm	Propagate from e-mail to e-mail using messages that contain worm-infected attachments or links that redirect users to worm-infected Web sites
Instant Messaging worm	Travel from messenger to messenger by sending links that redirect users to worm-infected Web sites; these links are often sent using a target's entire buddy list
Internet worm	Travel across the Internet using Internet scans and information found within a target (see example in the section titled "Elements of a Worm's Travel")
IRC (Internet relay chat) worm	Travel from chat to chat by sending worm-infected files and redirect links to worm-infected Web sites
File-sharing network worm	Travel from file-sharing network to file-sharing network by making copies of itself and placing them in a shared folder with an appropriate name

Table 1-3 Types of worms

- ***Remote access and administration Trojan (RAT)***—Provides remote access capability to the cracker from whom the virus originated. Remote access provides a user's complete control and access to your computer from a remote location. Remote access also gives a cracker the ability to record content (such as video from your Webcam and voice from a microphone), which is a cracker's own personal reality show—your life at his fingertips. Remote access is potentially the most dangerous capability a cracker can have.
- ***Data-sending Trojan***—Obtains sensitive data from your computer and transmits it back to a cracker. Key loggers are the most often used data-sending Trojans. A key logger is an application that logs your keystrokes in an attempt to retrieve sensitive data. Key loggers can remain on your system unnoticed for months, silently recording every key that you punch into your keyboard. These records are then sent via e-mail to the cracker for review at previously specified intervals. As with other malware, the most common way that Trojans spread is through e-mail.
- ***Destructive Trojan***—As its name suggests, a destructive Trojan is installed on your computer with the intent to destroy your system as a whole. Destructive Trojans are more like viruses than any other Trojans. They randomly delete files and folders and corrupt the registry. If left undetected for too long, this type of Trojan will leave you with a corrupt OS and an inoperable PC.
- ***Proxy Trojan***—Enables a cracker to use someone else's computer to access the Internet in order to keep her identity hidden. Using your computer to access the Internet means that the attacker will be using your IP address to commit cybercrimes. Your IP address is registered to your home address through your ISP (Internet service provider) and is used to identify you on the Internet. Therefore, any crime committed with your IP address could potentially result in you being investigated for identity theft or other cybercrimes.
- ***File transfer protocol (FTP) Trojan***—Allows the attacker to use someone else's computer as an FTP server. Installing this Trojan onto your computer would enable the

intruder to download files from his PC to yours, which could provide another avenue for more installation of malware. It also can enable the intruder to download files from your PC to his. Establishing your PC as an FTP server could also lead to the attacker storing illegal or pirated material (e.g., software, music, movies) on your computer, from which other Internet users could download. Again, this situation could potentially result in you being investigated for the illegal server.

Bots

Bots, also called software robots, are so named due to their ability to perform a large array of automated tasks for an intruder at a remote location. These tasks range in severity from spamming a system to initiating DoS attacks on systems. A DoS attack can slow down or completely shut down a database system or network of systems by flooding and overwhelming them with requests. For example, think about what would happen if there were one hundred applications running on your PC simultaneously. The PC would slow down tremendously due to the RAM overload. This is one strategy that a bot uses to compromise a system.

Bots can be hidden in games and other enticing programs downloaded by unsuspecting users, e-mailed from one infected machine to another, downloaded from infected Web sites, and/or can break into a person's computer through vulnerabilities found in the security architecture.

Bots allow intruders to gain access and full control of a number of computer systems from a distance, without an administrator or antivirus program ever becoming wise to the attack. Removal and detection of a bot often requires the use of a special antivirus program created specifically to look for bots. Because of the complexity of a bot, the detection of one bot normally requires the reinstallation of the whole system. You can never be completely sure that all bots have been removed from a system. Even with these specialized programs, bots are often not detected because:

- Not all bots are alike.
- Bots know how to update themselves, so new versions are created daily.
- Like Trojans, bots have the ability to hide and disguise themselves, and are aware of all the necessary steps to avoid being detected.
- Bots run in the background, virtually in silence.

Bots are only one component in a larger scheme to destroy networks. Once a computer is under the control of a bot, it becomes a part of a network of bots, called a **botnet**. This botnet is controlled by an individual often referred to as a **botmaster**. A botmaster accumulates a number of bots and then rents these botnets to other intruders and cybercriminals for the purpose of spamming, phishing, and other more serious types of cybercrime.

Security Architecture: A Never-ending Cycle

There is a great amount of uncertainty in the security field. Creating a security architecture that effectively ensures the confidentiality, integrity, and availability of database environments is no easy task. In considering the steps to achieving security goals, one message should remain in the forefront of our minds: security is never 100%, and we can never be 100% secure.

The techniques that are used to attack databases and other systems are developed using the same technology that is used to protect these systems. This means that as security systems become more sophisticated, viruses, Trojans, and worms become more sophisticated as well, and as technology becomes more advanced, so do intruders. With a virus population of approximately 600,000 and growing, there is no time to rest. By the time you reach a level of security where you feel comfortable, several new intrusions will have been developed and the process starts all over again. In this section, we will review the process of creating and maintaining security architecture.

Phase 1: Assessment and Analysis

Assessing and analyzing an organization's data security needs involves the identification of vulnerabilities, threats, and assets that exist within an environment's devices, resources, and vendor relationships. A security audit must be thorough and exhaustive, searching for every type of potential threat that may exist within the database environment. These threats can range from social engineering gaps to external firewall faults. They can be present within any of the computer, network, and database layers, so all types of security should be addressed.

An audit can be a difficult task for a group of security professionals. The process can take quite a bit of time and resources to complete, so audits can be conducted either internally or by paying a third-party company, such as a group of white hats.

By identifying risks, defining the likelihood of a threat to an asset, and determining the cost of a breached or lost asset, you can prioritize and plan reasonable measures to counteract these threats. Security measures that are created to counteract risks found on a network or database system should never exceed the value of the assets that they are protecting. Questions that are often asked during the assessment and analysis phase are:

- What are the devices and resources within the database environment?
- What type of threats exist within the database environment?
- What are the assets that need protection?
- What value do the assets have?
- What cost would the threats incur?
- What is the likelihood that each threat will occur?
- What level of security is needed for each threat?

Steps that are often taken to complete a risk assessment include:

1. Create a list of all devices and resources within a database environment.
2. Identify the vulnerabilities and the assets involved with each resource and device.
3. Define the value of these assets as well as the cost of any damage from the threats.
4. Using the information from Step two, as well as an understanding of the likelihood of each threat, create security measures to counteract these threats.
5. Prioritize your security measures.

Phase 2: Design and Modeling

The design and modeling phase involves the creation of policies and prototype security architecture that fit the needs of a business. A model for security will rely strictly on the results of the assessment and analysis phase. The prioritized lists of threats that are

created dictate how the model is developed and what policies are put into place. In the design and modeling phase, security policies and procedures are created, necessary firmware and software changes are defined, and security tools or applications that are used to minimize risk are identified.

The entire organization must be included in this process. From senior management to human resources to network users, all should be made aware of the security efforts taking place. Involving the entire organization in this process will ensure that policies are correctly focused and realistic for both user and business needs. Questions that are often asked during the design and modeling phase are:

- What policies need to be put into place to meet the security goals?
- What firmware changes need to take place to minimize vulnerabilities and support policies and procedures?
- What software is put into place to minimize vulnerabilities and support security policies and procedures?
- What are the steps for the implementation of the plan?
- What additional staff training will be necessary?
- How will success and failure be determined?
- What is the communication plan?
- How will the communication and training be delivered?
- How will firmware and software be tested?

Steps that are often taken to complete a risk assessment include:

1. Define the policies and procedures that need to be put into place.
2. Define the firmware and software changes that support the policies defined in Step one.
3. Identify the implementation plan.
4. Create baselines to determine success and failure.
5. Define a plan for user training and awareness.

Phase 3: Deployment

During the deployment phase, the security policies, firmware, and tools defined in previous phases are put into place. These security measures are deployed using the steps that were defined in the design and modeling phase. Deployment usually occurs in a test form first. A test environment is often created to simulate the environment in which deployment will take place. Firmware and software is purchased and also tested to ensure that unforeseen variables do not affect the overall deployment and security goals. Changes to user training and awareness are put into place in this phase as well. Questions that are often asked during the deployment phase are:

- Have user training adjustments been well accepted?
- Are all firmware and software tests successful?
- Are the security measures ready for full deployment into the active environment?

Steps that are often taken to complete a risk assessment include:

1. Adjust user training and awareness based on user acceptance.
2. Test firmware and software changes in a controlled simulation environment.
3. Deploy changes as defined by the deployment plan.

Phase 4: Management and Support

The management and support phase involves the ongoing support, maintenance, and assessment of the security architecture that was deployed in the previous phase. During this phase, performance of the security system is monitored, and any failures or breaches would result in the reevaluation of the security architecture. Security policies can go through minor changes, yet too many minor changes or a failure in a system should initiate the need to repeat the entire process from the beginning. Questions that are often asked during the management and support phase are:

- Is the security plan protecting the intended assets?
- Are enough time and resources invested in high-priority threats and assets?
- What impact do the security measures have on the users' ability to complete their tasks?
- What impact do the security measures have on the network's ability to complete a function normally?
- Are minor revisions needed?
- Are the security measures out of date?
- Have breaches increased?
- Is it time to reassess the environment?

Steps that are often taken to complete a risk assessment include:

1. Monitor performance of security architecture as well as user security awareness and training.
2. Make minor policy revisions as necessary.
3. Identify the need for a reassessment and initiate the start of the security life cycle.

Global Policies for the Database Environment

Operational information security ensures the secure operation of an organization through the development and reliability of an environment's policies and procedures. It focuses on security policies, change management, update management, and disaster recovery plans and is a necessary component for maintaining the database environment.

Security Policies

Security policies define the overall goal of security, identify the scope of what to secure, and define the roles and responsibilities of people within the organization. In addition, they identify specific communication processes and discus the enforcement of the policies. These policies

are vital to the success of a security architecture. Without a security policy in place, employees are forced to make up their own rules for security-related decisions, resulting in disastrous effects within a database environment. A security policy's effectiveness can affect how quickly a company will recover from a threat to a system.

The amount of loss incurred from a security breach can also be dependent on the effectiveness of the security policy. Security plans should be written by a committee of individuals, all whom have a stake in the success of the business. Security policies should not be created by IT professionals alone. The more invested the stakeholders are in this process, the more effective the policy will be. A security policy must include the following information:

- *Define the overall goals of the policy*—The goals of a security policy should show a direct relationship to the overall business goals.
- *Provide the scale of the security policy*—The policy should define which data, people, departments, facilities, and technology are included and protected by the policy. Assets should also be clearly identified and included in the scope of the security policy.
- *Define the roles and responsibilities of all employees*—The policy should identify the roles of those involved in maintaining the security of the environment. These roles should include the identification of a security team, decision makers and policy enforcers, and user responsibility in keeping the network secure.
- *Identify processes*—This component of the policy should include processes for prevention, detection, and reaction of security threats that include, but are not limited to, securing, updating, maintaining, managing, and monitoring a network. This section should include instructions for both accidental and purposeful attacks.
- *Handle noncompliance*—The policy should include the consequences for not complying with the security policy. What disciplinary actions will take place? Disciplinary actions should be based on the severity of the situation, yet need to be flexible enough to be applied consistently to everyone in the organization. Consequences should not be based on the role an employee plays within an organization (e.g., senior managers should face the same consequences as administrative assistants).

Once these components are defined and the security policy is complete, a plan for communication of the policy should be created. Security policies can be communicated using a regular training regimen, or delivered to employees individually. Either way, security plan awareness should be conducted regularly. The plan for dissemination should include a plan as to how new employees will receive the information and how updates to the plan will be communicated. Another process that should be considered after the security plan is complete is the process of assessment and revision. The security plan should be regularly assessed and revised based on changes made to business goals.

Update and Upgrade Management

Defining procedures for updating a database environment's software and firmware is as important to a security system's architecture as the security plan itself. **Update management policies** should include procedures for patch updates, software upgrades, OS upgrades, and firmware changes. An update or change to a system should never be done without careful planning and thoughtful consideration.

The first question that must be answered prior to a system update is: Is the update necessary? To help answer this question, we must first distinguish between an update and an upgrade.

An **update** is a change to a system that is added to software or firmware that is already installed on a network. This can include the database application, database server, client applications, and client machines. Updates are often minor changes made to software or firmware to slightly improve upon the functionality of a system or to help ensure that your current software or firmware maintains compatibility within a database environment. They are often small changes that are easy to apply and easy to reverse. They are normally distributed from a vendor free of charge. An update is represented as going from version 1.0 to version 1.1. Updates may be necessary if they provide fixes to the current version of the software or firmware residing on your database, if there are known security vulnerabilities within the software or firmware on your system, or are necessary to maintain compatibility throughout an environment.

On the other hand, upgrades are usually larger, more intrusive changes. **Upgrades** are normally replacements for older versions of software or firmware. They are more difficult to apply and even more difficult to reverse. Upgrades normally need to be purchased. An upgrade is represented as going from version 1.0 to version 2.0. One could be necessary when older versions are no longer being supported by the vendor, if they provide a significant amount of improvements to a system, or if the older version no longer fits an organization's needs. Unless absolutely necessary, an upgrade should not be applied to a database or its environment immediately after release. Upgrades are often complete overhauls of previous versions of software and firmware. They often do not become stable for months, and, in some occasions as with operating systems, years after the upgrade's release.

Vendors of software and firmware cannot fully test their systems against all components that might be found within a database environment; therefore, upgrades are often released to the public with bugs or glitches that are not found and fixed until after some time. Bugs and vulnerabilities of new programs are often discovered through reports made by companies and individuals who purchased the software early. It is good practice to wait until a software upgrade becomes stable before applying one to a database or its environment.

The next question in determining whether to apply an update or an upgrade to your system is: What are the possible repercussions of the install? Understanding the possible compatibility issues or negative consequences that can result from a change made to your environment can be vital to maintaining the integrity and availability of the resources. Unfortunately, software vendors are not eager to share the possible issues that might arise from updating or upgrading their product, so this information can be difficult to find. Here are a few suggestions to find out what known complications and compatibility issues exist with a software or firmware upgrade or update:

- Check the vendor's help and troubleshooting pages.
- Search technical support forums, newsgroups, and blogs.
- Ask your peers and colleagues who may have recently installed the product.
- Check within your own database manual for software and firmware specifications to ensure compatibility.

You must be relentless in your search to ensure that you are not putting your database environment and resources in jeopardy. If possible, create a test database environment in which

to install your upgrade. Test database environments, although not foolproof, can often give you clues as to what issues you may encounter. The time and effort that you put forth early in this process will pay off greatly in the long run by possibly saving you time, money, and many late hours at the office.

Prior to making any changes within the database environment, a recovery and restore plan should be put into place including failsafe options, such as reversal and backup. Instructions are often included with updates to allow an administrator to reverse an update in the event that the installation fails or the database rejects the new software. So before applying an update, check to be sure that an update has a reversal technique, and locate and read the reversal instructions so you are prepared for problems that may occur.

To add extra redundancies, back up your files. In case the reversal does not work, a backup can save your database settings. In addition, if at all possible, schedule the update during off-hours so as not to interfere with your users' work hours. Finally, after an update has been applied, it is important to document the changes that were made. Sometimes negative effects do not appear right away. Without a document showing recent changes, it can be quite difficult to pinpoint the problem and restore your system.

Types of Updates and Upgrades There are different types of updates and upgrades that can be performed within an environment. Understanding the different types of updates and upgrades is vital to the success of an update and upgrade management policy. This section focuses on patches, software upgrades, and operating system upgrades to determine the level of commitment that is required to complete each one.

A **patch** is a small program that is used to fix or update software programs or firmware devices. A patch is often created as a response to a newly discovered vulnerability found within a program. Because it is impossible for software vendors to test their software in every single environment, they are forced to release programs to the public that contain certain vulnerabilities, glitches, and compatibility issues that can result in a security breach.

Vendors learn about these vulnerabilities as well as other software issues from organizations who install and report them early on. Once a vulnerability or compatibility issue is found, a vendor will develop a patch to fix the problem and fill in security holes. If a user does not update or patch their system, the system will remain vulnerable. A system that has not been updated allows malware to take advantage of these vulnerabilities to intrude upon a database environment.

Therefore, it is a best practice to develop a strategy for receiving notification, managing installation, and documenting or tracking patches that are applied on your system's software and firmware. Some vendors' software programs include a tool that automatically searches for and installs updates and patches online (e.g., Windows Automatic Updates). Some vendors send e-mail alerts and text messages to inform their clients of critical or recommended software patches available, while other vendors require administrators (or the people that they have designated within the security plan) to search a company's Web site looking for patches on their own.

Often, software or firmware vendors will combine a number of patches to create a new version of the software, called a **software upgrade**. As mentioned earlier, upgrades are more intrusive than updates because they involve changes to the system, so careful, thoughtful planning is necessary to avoid availability and reliability issues.

An **OS upgrade** is accomplished by installing a new version into a host or a server. They are the most significant and risky upgrades that can be installed onto a network. These upgrades often involve radical changes to both clients and servers. Database server upgrades affect every single client within the environment and if the clients are Web applications or Web forms online, they can impact the external user's ability to access the database. Therefore, without proper planning, scheduling, research, and testing, an OS upgrade can result in a great amount of wasted time and money for a business. A failed OS upgrade can potentially leave networks and resources completely unavailable for days at a time. Imagine the loss for a company like Walmart if their point of sales servers (register software) stopped working for an entire day!

Just as with most projects, proper planning is critical to the success of the upgrade and requires a strong understanding of the current environment, future objectives, and the beginning and end points. Servers maintain all of the structure and the resources available within the database environment. They determine the logical structure of the data within the database, the user rights and access with the database, and the general resource availability of the database to internal and external clients. Therefore, a plan should carefully consider the impact that the new database system will have on the environment. The research required to do this is quite a task within itself.

The sheer scale of a server OS upgrade makes choosing the right time to schedule it extremely important. As with any smaller software upgrades, scheduling the deployment after business hours is ideal, yet with a server OS upgrade, the time of year is also critical. Because our servers essentially maintain every detail of our database as a resource, a great deal of budget resources and time are needed. Therefore, even a successful upgrade can negatively impact a business if it is scheduled during busy seasons.

Just as with smaller software upgrades, a test environment in which the main elements of the database environment can be tested and configured will help you identify potential threats that the upgrade will pose. Although test environments do not provide a foolproof way to maintain a secure network, they do provide a good deal of help and can be the step that separates success from failure when upgrading a system.

Backup Management Plan

A **backup** is an intentional copy of your data, program files, and system configurations, and is used to archive and store information. A backup can be used to replace or restore your files and systems after a network failure or malware attacks. A **backup management plan** is a process developed to ensure the safety of the data on a network. Without a backup management plan in place, the risk of losing valuable data and network resources is immense.

Backup Solutions There are many backup management solutions available today, and choosing the solution that best fits your data and business goals is important. In customizing a backup management plan that works best for your organization, several questions must be answered:

- *What type of media should I use?*—One of the first decisions that needs to be made once a backup management plan is created is what type of media will be used to store your backup data. There are different types of media available for saving your data, and choosing the right one for your business will depend on the cost, compatibility with the environment, labor, and potential for growth. Table 1-4 illustrates the types of backup media available today.

Type	Storage size and technology	Effort
CDs (compact discs), CD-RW (compact disc-rewritable) DVD (digital versatile discs) and DVDs	Use optical digitized data; lasers burn image onto the disc CDs and DVDs can be written onto one or both sides The size of the disc depends on the type of CD or DVD, but ranges from 700 MB to 17 GBs of storage	Require a computer with a suitable CD or DVD drive; process can be automated only to a certain extent; CDs and DVDs must be changed often and require a fair amount of supervision
Tape backup cassettes	Small cassette tapes Use magnetic tape to store information onto a cassette Can store up to 366 PB (petabytes) of information	Process is fully automated; removing and changing tapes require supervision Require special backup server and software
External drives, hard drives, jump drives	Thumb drives, external hard drives, USB, PCMCIA; FireWire uses flash memory technology Can range from 1 GB (gigabyte) of information to 1 TB (terabyte) of information	Process has very little automation Requires very little supervision

Table 1-4 Media storage types

- *Where is the backup to be placed?*—Backups should be saved in areas of the network other than the original data location to avoid both the original and copy being destroyed at the same time. Backups should never rely on the same system as the original data. It is not uncommon to move daily backup media off-site in order to save it. For example, once a tape or CD backup has completed for the day, the administrator drives the media to a local bank's safety deposit box to ensure the safety and security of the data. Many third-party data storage companies offer online storage and backup of your organization's data as well. Third-party systems require the creation of a secure connection, but it is available.
- *What should be backed up?*—Involves assessing and mapping data throughout the environment, as well as the identification of critical information, and should ensure that critical data receives backup priority.
- *How often should information be saved?*—The frequency with which data should be saved should correlate with the importance of the data being saved. In other words, the more valuable and critical the data, the more frequent should be the backup of that data.
- *What time of day or night should backup occur?*—Backup should occur during downtimes and/or the end of the work day. Backup should not be completed while information is being saved and updated.
- *What type of backup should be completed?*—There are different types of backups. A **full backup** will back up all of your information, regardless of its critical nature, age, and prior backup activity. An **incremental backup** conducts a backup on only the data that has changed since the last full or incremental backup. As the incremental backup completes, it flags data that is being updated so that it is not included in the next incremental backup. A **differential backup** will save only the data that has changed since the last backup was complete (full, incremental, or differential). As the differential backup completes, it does not flag any data, so that the data is stored again when the next incremental or full backup occurs.

The Disaster Plan

A **disaster plan** is a plan developed to ensure the quick reinstatement of a network that has fallen victim to a human or naturally caused disaster (e.g., earthquakes, fires, floods, snow, ice, hurricanes, tornadoes, explosions, chemical fires, hazardous spills, smoke, water, solar flares, and human error).

The goal of a disaster plan is to ensure the speedy reinstatement of the most critical aspects of the business. Disaster plans should not focus on minor outages, but on major disturbances that could potentially cause devastating results for an organization.

A disaster plan should include the contact names and phone numbers for emergency coordinators that will execute the disaster recovery response, as well as roles and responsibilities of emergency response staff.

Details of the network backups should be included to ensure that the most recent accurate data can be restored and these details should include information about the data and servers that are being backed up, how frequent these backups occur, where backups are stored, and information on how these backups can be recovered.

Any and all agreements made with national services carriers should also be outlined at both the local and regional level for cases when the regional service carriers are hit by the disaster as well.

Communication strategies should be defined for communication to employees and customers before, during, and after the disaster. Communication strategies should include means for communicating in scenarios where regular modes of communication are not available.

Information related to any contracts that are currently in place with organizations that specialize in disaster recovery services and provide off-site help should also be included.

Third-party disaster recovery organizations offer cold sites, warm sites, and hot sites that offer solutions to concerns about disaster recovery.

A **cold site** is a facility that provides the basic necessities for rebuilding your network. A contract that involves a cold site would promise the use of a facility that provides water, power, air conditioning, or heat. A cold site only offers the environment necessary to rebuild the network, and does not include the computer's software or firmware. This is the least expensive agreement that can be made. Rebuilding at a cold site would involve restructuring a network from the ground up.

A **warm site** is an agreement that promises the existence of a facility that contains the basic environmental concerns, as well as computers, connection firmware, and software devices necessary to rebuild a network system. Applications are not included. Rebuilding at a warm site would involve restorating all applications from backups available. A warm site provides the technical infrastructure, and the customer rebuilds the logical infrastructure.

A **hot site** is an exact replica of an organization's network, in other words, a mirror site. A hot site promises that the vendor will assume all responsibility for ensuring that the network is readily available in the event of a disaster. Hot sites are very expensive, but for critical environments they are well worth the financial cost.

Shared site agreements are arrangements between companies with similar, if not identical, data centers; there is compatibility in hardware and software that enables companies who

agree to a shared site to back up each other's data. If a disaster occurs and a shared site agreement is in place, the affected organization would move to the unaffected organization's building and continue business there. This is a great alternative to a hot site, yet it is very difficult to find a company with a close enough match to enter into this agreement. Shared sites are usually different locations of the same company.

Chapter Summary

- Database security refers to the efforts taken through policy, procedure, and design to achieve the absence of threat or harm within our database systems.
- The key to achieving effective database security involves achieving confidentiality, integrity, and availability. It is important for security professionals to have experience and remain up to date in order to identify threats and vulnerabilities to the system they are protecting.
- Some potential threats to database systems include crackers, social engineers, computer users, network administrators, malware, Web browsers, and e-mail.
- Malware is an umbrella term for malicious software and can come in many forms, such as computer viruses, worms, Trojans, spyware, adware, crimeware, and bots.
- A virus is a program that spreads from one computer to another without detection, either through network connections or storage devices. There are over 600,000 viruses currently residing on our networks, and that number grows daily. They use different defense strategies to remain undetectable, and have common traits that make them extremely resilient. These strategies include self-encryption, stealth, polymorphism, and residence.
- There are different classes of viruses. For example, logic and time-dependent viruses initiate when certain variables are met. Other viruses, like spyware, only focus on gathering sensitive information. An adware virus's main purpose is marketing by obtaining potential customer information and/or excessively promoting products.
- Unlike viruses, worms do not need users. They self-replicate by harnessing the power of the networks and using this power to attack those networks.
- All worms share common elements in replicating themselves across the network. First, they find a weak target, using vulnerabilities found on a network. Next, they take control of the target and pull all viable sensitive information from it. Finally, they test the network looking for new vulnerabilities, and, therefore, targets.
- A Trojan horse is a form of malware that disguises itself in its harmful code. Trojans may come in the form of music, software, or movies, using this disguise to infiltrate a system in order to corrupt, transfer files, and log user activities.
- Bots, or software robots, have the ability to perform a large array of automated tasks for a botmaster at a remote location. Bots are difficult to detect because they self-update, morph exceptionally, and run virtually in silence. Often it requires a special program to remove a bot from a system.
- Security is never 100% and we are never 100% secure. Security is a never-ending cycle of assessing a network, designing security policies, deploying security architecture, and testing security performance.

- Operational information security ensures secure operation within an organization through the development of policies and procedures. It focuses on policies that record and define changes, updates, and failsafe measures that are made within a database environment and it provides comprehensive disaster recovery plans.
- A disaster plan ensures the quick reinstatement of a network that has fallen to a disaster. Such plans define the dynamics of hardware, software, and user correlation throughout the organization's environment and make provisions for restoring the network in jeopardy.
- Security must never be one-dimensional. Building a secure architecture requires a multifaceted, strategic approach. Careful attention to structural layers must be considered in order to build an exceptional and reliable infrastructure.

Key Terms

adware A general term for software that uses typical malware intrusion techniques to obtain marketing data or advertise a product or service.

availability The efforts taken through policy, procedures, and design in order to create and maintain the accessibility of resources within a database environment.

back door A method created during the programming of a worm in which access is gained into a system by avoiding normal security, which gives the creator of the worm undetected access into the system.

backup An intentional copy of data, program files, and system configurations that is used to archive and store information in the event of network failure or malware attacks.

backup management plan A process developed to ensure the safety of the data on a network.

black hat Someone who breaks into computer networks without authorization and with malicious intent.

boot sector An area of the hard disk that contains records necessary to the boot process of a computer.

boot sector virus Malware that infiltrates a system by loading itself onto the boot sector of a hard disk via an infected floppy disk left in a floppy disk drive.

bot (software robot) A form of malware that has the ability to perform a large array of automated tasks for an intruder at a remote location, ranging in severity from spamming a system to initiating DoS attacks on systems.

botmaster An individual who controls a network of bots and who accumulates a number of bots and then rents these botnets to other intruders and cybercriminals for the purpose of spamming, phishing, and other more serious types of cybercrime.

botnet A network of bots.

cold site A facility that provides the basic necessities for rebuilding a network. A contract that involves a cold site would promise the use of a facility that provides water, power, air conditioning, or heat.

computer security A set of established procedures, standards, policies, and tools that are used to protect a computer from theft, misuse, and unwanted intrusions, activities, and attacks.

computer virus A form of malware intended to spread from one computer to another without detection.

confidentiality The efforts taken through policy, procedure, and design in order to create and maintain the privacy and discretion of information and systems.

cracker An individual who breaks into our networks without authorization with hopes to destroy and/or steal information.

database security A set of established procedures, standards, policies, and tools that are used to protect data from theft, misuse, and unwanted intrusions, activities, and attacks.

data sending Trojan Malware that obtains sensitive data from your computer and transmits it back to a cracker.

denial of service (DoS) attack A concerted effort made by malware to keep system resources, such as Internet sites, from functioning correctly.

destructive Trojan Malware that is installed on a computer with the intent to destroy a system as a whole by randomly deleting files and folders and corrupting the registry.

differential backup An intentional copy of data, program files, and system configurations that only saves the data that has changed since the last backup was complete.

disaster plan A plan developed to ensure the quick reinstatement of a network that has fallen victim to a human or naturally caused disaster.

DNS poisoning An intrusion where a cracker gains control over the DNS server and changes the domain name's respective IP address, redirecting requests to sites that the cracker has built and maintains.

encryption The transformation of data by using sophisticated algorithms in an attempt to make the data unrecognizable.

file-infected virus A form of malware that will attach itself to an executable file that requires a user to run before it can propagate and corrupt the system.

file transfer protocol (FTP) Trojan Malware that allows the attacker to use someone else's computer as an FTP server.

full backup An intentional copy of data, program files, and system configurations that stores all information, regardless of its critical nature, age, and prior backup activity.

grey hat An individual or groups of individuals who waver between the classification of a hacker and a cracker, and who either act in goodwill or in malice.

hacker Someone who has mastered the hardware and software of modern computer systems and enjoys the exploration and analysis of network security with no intent to intrude or cause harm.

hactivist Hackers and crackers who use their extensive experience and skill to use networks to share their ideologies regarding controversial social, political, and economic topics.

Health Insurance Portability and Accountability Act (HIPAA) Strict laws for health institutions throughout the United States that ensure the security and privacy of patient records by dictating the way in which files are accessed, stored, and transmitted on a network.

hijacking A process in which Web sites are hacked into and rewritten to react differently to users than how the original Web site designer intended.

hot site An exact replica of an organization's network, or a mirror site, that promises the vendor will assume all responsibility for ensuring that the network is readily available in the event of a disaster.

hypertext transfer protocol (HTTP) The portion of an Internet address that informs the browser what protocol is used to send the request for a particular Web site.

incremental backup An intentional copy of data, program files, and system configurations that is conducted on only the data that has changed since the last full or incremental backup.

integrity Efforts taken through policy, procedure, and design in order to create and maintain reliable, consistent, and complete information and systems.

logic bomb Malware that can lie dormant until a specific predetermined variable is met, whose variables typically depend on the environment in which it resides.

macro A small program that enables users to automate a large number of repeated processes within a document.

macro virus Malware that can either be attached to a macro, or can replace a macro within a document, and that runs automatically when the document containing the infected macro is opened or closed.

malicious software A programming code written and used by unauthorized intruders to perform a certain task on a computer.

malware An abbreviation for the term malicious software.

misleading applications Applications that deceive users into believing that their computer's security has been breached, therefore tricking the user into downloading and purchasing rogue antivirus tools to remove the bogus breach.

multipartite virus A form of malware that combines the characteristics of a boot sector virus with those of a file-infected virus.

network security A set of established procedures, standards, policies, and tools that are used to protect data from theft, misuse, and unwanted intrusions, activities, and attacks.

nonresident virus The general term for malware that requires users to initiate it by downloading a program or opening up an e-mail attachment.

operational information security Ensures the secure operation of an organization through the development and reliability of an environment's policies and procedures that focus on security policies, change management, update management, and disaster recovery plans.

OS upgrade Installing a new version of an operating system onto a host or a server.

patch A small program that is used to fix or update software programs or hardware devices.

payload The component of a worm that holds all of the instructions on how to affect each computer that it encounters.

personal identifiable information (PII) Personal information that identifies a person.

phishing The attempt to obtain PII from people through the use of spoofed e-mail addresses and URLs.

polymorphism The incidence of changing forms, or self-modification.

proxy Trojan Malware that enables a cracker to use someone else's computer to access the Internet in order to keep their identity hidden.

remote access and administration Trojan (RAT) Malware that provides remote access capability to the cracker from whom the virus originated, who in turn is provided complete control of and access to someone else's computer from a remote location.

resident virus Malware that installs itself or takes residence directly in the main system memory of a computer.

script kiddie Amateur crackers that use programs and scripts written by other people to infringe upon a computer network system's integrity.

security policy A document that defines the overall goal of security and identifies the scope of what to secure, as well as the roles and responsibilities of the people within the organization.

shared site agreement An arrangement between companies with similar, if not identical, data centers.

signature A pattern of characters that is identifed for a specific family of viruses.

social engineer An individual who uses human interaction to manipulate people into gaining access to systems, unauthorized areas, and confidential information.

software upgrade A combination of a number of software or hardware packages that creates a new version of software.

spoofing A process that involves hackers building Web sites to look identical to other popular sites in hopes of drawing in a user.

spyware A general term for any software that intentionally monitors and records a user's computer and/or Internet activities.

startup page The Web site that is displayed when the Web browser is started.

time bomb (time-delayed virus) Malware that can lie dormant until a specific variable is met, such as times, days, or specific days that are predetermined and written within its code.

transmission packet Sensitive information about users or businesses compiled by spyware that is sent back to its original creators for use as they see fit.

Trojan (Trojan horse) Malware that disguises itself and its harmful code and often hides within enticing programs such as software updates, games, and movies.

update A change to a system that is added to software or firmware that is already installed on a network.

update management policy A document that includes procedures for patch updates, software upgrades, OS upgrades, and firmware changes.

upgrade Replacements for older versions of software or firmware.

warm site A facility that contains the basic environmental concerns, as well as computers, connection firmware, and software devices necessary to rebuild a network system.

Web browser An application that acts as a user interface of the Internet, allowing users to interact and view Web pages on the World Wide Web.

Web page A document containing a specific programming language (e.g., HTML or JAVA) that is designed to be viewable on the World Wide Web.

white hat An ethical hacker; hackers who use their extensive experience and knowledge to test systems and provide security consultation to others.

worm Self-replicating malware that is able to harness the power of networks and use this power in its attacks against them.

Review Questions

1. Identify the three main items that are utilized in achieving security objectives in order to protect our database systems.
2. Identify and define three objectives that are key to achieving effective security architecture.
3. List and define the different classifications created to clarify the difference between hackers and crackers.
4. List six common errors that users make on a network. Give examples of each.
5. Identify three ways that the Internet can be used as a tool to compromise information security.
6. List the destructive tactics that uneducated computer users can run into when using e-mail.
7. Define the following: computer viruses, worms, Trojans, spyware, adware, and bots.
8. List and define each phase in the process of creating and maintaining a security architecture.
9. List and describe the information that should be included in a security policy.
10. Explain the difference between an update and an upgrade.
11. List six questions you should ask when creating a backup management plan.
12. Which backup media would be most appropriate for a large enterprise or network?
13. Identify and explain the four options available for restoring your network in the event of disaster.
14. Explain the multilayered nature of security.
15. Identify the five layers of security and give examples of each.

Case Projects

Case Project 1-1: Locating Recent Vulnerabilities

Use one or more of the links provided below and write a paper that includes the top ten most current exploits and vulnerabilities identified. Include a description of these vulnerabilities as well as any known fixes or counterattacks.

- *http://www.microsoft.com/technet/security/advisory/default.mspx*
- *http://www.us-cert.gov/*
- *http://www.exploit-db.com/*

Case Project 1-2: Legal Privacy Compliancy

HIPAA is one example of a well-known act passed by Congress that has identified compliancy and privacy laws for certain types of organizations. Choose at least two other acts from the list provided and discuss the rules for privacy

and confidentiality that are set forth in these acts. Include at least one policy, procedure, and design strategy that an organization can take to comply with each act discussed.

- Payment Card Industry (PCI) Data Security Standard (DSS)
- The Health Information Technology for Economic and Clinical Health (HITECH)
- The Sarbanes-Oxley Act (SOX)
- Among other changes to financial laws, the Gramm-Leach-Bliley Act
- The Basel II
- ISO 17799 Compliance
- FFIEC Compliance
- CA SB 1386 Compliance
- DCID 6/3 Compliance
- DoD 8100.2 Compliance
- FISMA Compliance
- NISPOM Compliance

Case Project 1-3: Understanding the Risks That Users Pose

Organizations take great measures to ensure the secure design of their organization architecture. Write a paper that discusses the ways users can harm this architecture if they are not properly trained. Include in your paper the policies and procedures that you will recommend to ensure that users will support the security of your network.

Hands-On Projects

Hands-On Project 1-1: Assessing and Prioritizing Risks

You have been hired as the security professional in your current work or school environment. Your department has experienced a recent breach within its database architecture. Your manager feels it is time to reassess the network as a first step in a phased approach to hardening the network security. You have been delegated to lead this assessment. Research your organization or school to determine its database architecture. Write a paper describing your assessment process. Include the following information in your paper:

- Define who would be included in the assessment of the database environment.
- Identify the assets that you are protecting.
- Define the threats you are protecting yourself against.
- Using the following table, assign a risk value to each threat, based on its likelihood and cost to the company.
- Prioritize your threats based on their risk value (5 being the highest risk and 1 being the lowest).

Risk values 5 = high, 1 = low

	High cost	Medium cost	Low cost
High probability	5	4	3
Medium probability	4	3	2
Low probability	3	2	1

Hands-On Project 1-2: Designing Your Security Defense

You have been hired as the security professional in your current work or school environment. Your department has experienced a recent breach within its database architecture. Your manager has delegated you to lead an effort to eliminate threats within your database environment.

Based on the results that you found in your assessment, five priority threats have been identified. Define each threat identified in Hands-On Project 1-1 and indicate the policy, procedure, and design changes that need to be taken to ensure the security of your network. Ensure that your measures follow the multilayered approach to security by using the following table:

Layer	Computer layer measures	Network layer measures	Database layer measures
Threat 1			
Threat 2			
Threat 3			
Threat 4			
Threat 5			

Hands-On Project 1-3: Implementing a Strategy

You have been hired as the security professional in your current work or school environment. Your department has experienced a recent breach within its database architecture. Your manager has delegated you to lead an effort to eliminate threats from the database environment. Based on the results from your assessment, five priority threats have been identified. Changes in policy, procedure, and design need to take effect in order to ensure protection from these threats. Using the five priority threats you identified in Hands-On Project 1-1, create a plan for implementation. Your plan for implementation should include the following:

- The phases in which you intend to implement the project
- The cost of the implementation, including a list of all new firmware and software to be purchased as well as training costs

- The plan for communicating these changes to network users
- The plan for training network users on changes, if necessary
- The time frame for implementations
- The time of day that the implementations will occur
- The testing strategy