No book on computer incidents or incident response would be complete without a discussion of the various types of attacks that may be launched. This chapter provides that discussion, building on the terminology presented in Chapter 3. As this book is not intended to be an in-depth technical guide, such attacks will be described only at a high level. The chapter begins by examining some of the major consequences that can result from an attack, and then looks at some specific vectors used to launch such attacks. It ends with an in-depth discussion of malicious logic, especially the computer virus. Steps that can help organizations reduce their potential for virus infections will also be presented as part of the malicious logic discussion.

**NOTE**  A more thorough discussion of computer viruses and other such code is presented because many managers find themselves dealing with this sort of incident on a frequent basis. Adding to the interest is the level of attention given to the topic of computer viruses by the media over recent years.Viruses are not necessarily the most significant threat that a CIRT will address. In reality, they are only one of many problems that an organization might encounter and must be prepared to handle.
Computer security professionals from various organizations have worked together to establish a list of the most commonly exploited vulnerabilities. This list is maintained by the SysAdmin, Audit, Network, Security (SANS) Institute and maintained at its Web site (http://www.sans.org). The development of the list included input from the FBI, CERT CC, and many other organizations and individuals. The original list was posted in 2000 and included the top 10 most frequently exploited Internet vulnerabilities. The list expanded to the top 20 vulnerabilities in 2001. The top 20 list is actually a combination of two top 10 lists—the top 10 vulnerabilities associated with UNIX and the top 10 vulnerabilities associated with Windows operating systems. Despite the best efforts of many security professionals, this combined list still accounts for the most frequently exploited vulnerabilities.

Anyone charged with securing an information infrastructure should begin by using the SANS list as a checklist for possible vulnerabilities and proceed from there. In some cases, it may be quite difficult to totally prevent the vulnerability from being present in an infrastructure. For example, poor password choices are a vulnerability that is frequently exploited to gain unauthorized access to a system, yet passwords are needed in most cases as one level of security. Nevertheless, some specific steps may be taken to lessen the exposure of the potential vulnerability, such as conducting frequent password scans, conducting password training for end users, and invoking automated password rules. Similarly, sendmail is a mail server application used with many UNIX installations that has had numerous vulnerabilities identified over the years. Ensuring that the most recent version of sendmail is loaded and used on the system can help prevent some of the problems associated with the program. The same holds true for patching other operating systems as vulnerabilities are discovered and versions of the software are updated or patched.
Consequences of Computer Attacks

Computer Intrusion, Unauthorized Access, or Compromise

A compromise occurs when someone gains unauthorized access to a computer resource. Typically, we tend to think of people outside an organization when discussing compromises. In reality, a compromise can also be accomplished by an authorized user of a system or an internal employee. For example, if a person is authorized to use a system and has set privileges at an end-user level, bypassing security countermeasures to increase that privilege level should be considered an unauthorized access or compromise. Additionally, if an internal employee gains access to a system that he or she is not authorized to use, that breach should be considered a compromise.

System compromise can be achieved through both physical and electronic means. Access through physical means would be accomplished if an unauthorized person gained physical access to a system and was able to log on. The following scenarios provide examples of how such an event might happen:

1. Sam, an authorized user who happens to be a system administrator, is logged onto the computer system. While working on the computer, he realizes that he needs a tape that is stored down the hall and walks out of the room to get it. Unfortunately, Sam is in a hurry and forgets to lock his screen before leaving the room. Fred is a curious employee whose own office is currently being renovated so he is sharing the office space with Sam. Seeing that Sam forgot to lock his system, Fred decides to increase his user privileges to do some corporate snooping. He sits down at Sam’s computer and quickly changes his privilege level to that of system administrator. He returns the computer to the screen that was up when Sam left the room and goes back to his own desk. Sam returns with the tape and continues to work, with Fred now able to perform tasks as a system administrator. This increased access will most likely allow Fred to view files that he was previously unable to see.

2. Sherry is a receptionist for Company X and is very diligent about watching who comes into the office space. In fact, no one is able to
get into the offices without first going by her. Sherry's managers like to make administrative tasks as simple as possible for their employees, so a pool of computer resources are maintained in a room behind Sherry's office. Because Sherry is so good at her job and sits close to this resource pool, Company X does not see the need to require passwords on the computer systems located in the room. After all, it's there to provide greater ease of use for the employees.

Sherry takes her annual vacation. The boss hires a temporary employee to fill in as receptionist while she is away.

Tom works for Company X's main competitor and is determined to get access to its computer system to see the firm's latest developments. He disguises himself as a copier repairman and appears at the office shortly after noon during the week that Sherry is on vacation. He gets past the temporary employee to "repair the copier," which happens to be in the same room as the pooled computer. Most of the regular employees are at lunch at this time, so Tom has ready access to the computer system for at least 30 minutes. During that period, he is able to access some files that he saves to a diskette, which he then slips into his repair bag. Tom informs the temporary employee that the copier is repaired and walks out of the office with his findings.

3. Emily has had some problems with her supervisor and is very concerned that her personnel file contains some negative remarks. One night she notices the door to the human resources department has been left open and decides to investigate her personnel file. Emily obtains a boot diskette from her desk and inserts it into the A drive before turning on the system. After the system boots, she is able to search the contents of the system's hard drive to look for personnel files and anything else she is able to access.

Requiring employees to wear identification badges and teaching them to challenge unknown personnel who are walking alone through a space will help to protect against a compromise occurring from an external resource through physical means. Ensuring that security countermeasures (e.g., passwords) are used and educating personnel on the importance of locking their screens will help to safeguard against the internal compromise threat. Additionally, systems should
be programmed to lock or initiate a screen saver after a short period of being idle.

Although physical access is one avenue used to get into a computer system, many more compromises occur through electronic means. A wide variety of methods are used to gain access to a computer system, and many excellent resources outline these means. The “Attack Vectors” section later in this chapter describes some specific attacks or avenues used to gain access electronically to a system.

These methods provide a small sampling of the attack avenues that may be used to gain unauthorized access to a system. Entire books have been written on such attack methods, so this chapter is not intended to provide a comprehensive examination of all avenues. Rather, the intent is to highlight some methods to demonstrate that threats may come through a multitude of means and from many directions.

**Denial-of-Service Attacks**

A denial-of-service attack accomplishes exactly what its name implies—it denies authorized personnel the use of a computer resource or resources. It is normally accomplished by flooding the target with signals so that its bandwidth is so consumed it cannot take further instructions. Several types of denial-of-service attacks may be used, some of which have familiar names such as LAND and smurf. (Smurf will be described in more detail later as an example of a denial-of-service attack.)

**Port Scans or Probes**

Some organizations do not recognize port scans or probes as incidents. However, experience working in an incident response team will reveal that many successful attacks (compromise or denial of service) are often preceded by some type of scanning activity. Rules must be set to not track too granular a level of probing, yet allow the recognition of this type of reconnaissance when it occurs.

Network scanning tools are frequently used to conduct a scan of a network. By scanning the network, the intruder is trying to glean as much information as possible that may be used to gain access. For
example, determining the type and version of the operating system used on a network will give an intruder hints about the associated vulnerabilities that may be tried. Most network scans will be readily apparent, as they will come from the same or a handful of IP addresses. This information can easily be identified and even protected against with most intrusion detection systems. More skilled attackers may change the scanning parameters to lessen their chances of being detected. Similarities in the ports scanned or timing of probes can help with the detection of these tests.

**Attack Vectors**

The following subsections describe some specific attack avenues that may be used to gain unauthorized access to a system, deny the use of information system resources, or conduct reconnaissance on the network. As noted previously, the avenues described here represent merely a sampling of possible attack vectors. Numerous other books provide more in-depth reviews of various types of attacks, and the reader is encouraged to continue research into a larger sample of attacks.

**The Human Factor**

*Default Accounts and Passwords*

Most vendors ship hardware and software products with default accounts and passwords already set. Before computer security started gaining more attention over recent years, these default accounts and passwords were used by vendors to help troubleshoot or fix problems when they arose. However, these settings soon became known to less helpful people and provided an easy route into computer systems. Although many vendors have tried to strengthen the security in their products over recent years, many continue to ship products with such accounts and passwords enabled. Before placing any system into production, all default accounts should be disabled or have their associated passwords changed.
The passwords selected and used by authorized end users represent another major avenue for unauthorized access to systems. Numerous programs are available that may be used to crack or decrypt passwords from the password file. Typically, an intruder will obtain a copy of the password file, run one of these programs on it long enough to identify a couple of login accounts and corresponding passwords, and then use these account login IDs and passwords to further their exploration of a system.

L0phtCrack is a well-known program that is used by many security professionals during penetration tests or vulnerability assessments. If permitted to run long enough, L0phtCrack will eventually reveal all passwords in the file. It does so not through a decryption algorithm, but rather through a pattern matching scheme. LC4 is the latest version of L0phtCrack and is designed to crack passwords on Windows systems. (More information on L0phtCrack and LC4 can be obtained from http://www.atstake.com.)

Table 4–1 provides examples of other password cracking software that may be used to reveal passwords. Most of the tools listed in the table as being able to crack Windows NT passwords can be used on all versions of Windows platforms (e.g., XP, NT, 2000).

Another popular method for obtaining passwords is through a brute-force attack. This type of attack involves simply guessing the user’s password. This guessing of passwords may be done either manually or by using an automated tool that attempts to guess the password by cycling through every possible combination of passwords. (This is actually how L0phtCrack identifies passwords through its pattern matching capability.) Educating end users on good password selection techniques, requiring passwords to be changed periodically, and not permitting the reuse of passwords are all measures that can reduce the likelihood of a successful brute-force attack. The stronger the password, the more likely the entity trying to guess the password will give up and move onto another target. The Tips for Selecting Good Passwords sidebar lists a few recommendations to consider in selecting good passwords and may be used as a starting point to help address this topic in training programs.
# Table 4-1  Examples of Password Cracking Software

<table>
<thead>
<tr>
<th>Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gammaprog</td>
<td>Cracks passwords of Web-based e-mail (e.g., Yahoo!, Hotmail) and POP.</td>
</tr>
<tr>
<td>Hypnopaedia</td>
<td>Remotely connects to a POP3 mail server and attempts to guess the password of a specific account.</td>
</tr>
<tr>
<td>John the Ripper</td>
<td>Popular UNIX-based password cracker that can attack UNIX, Kerberos, Andrew File System (AFS), and Windows NT passwords.</td>
</tr>
<tr>
<td>Mssqlpwd</td>
<td>Cracks Microsoft SQL Server passwords.</td>
</tr>
<tr>
<td>PalmCrack</td>
<td>Runs on Palm PDA and can crack UNIX, NT, and some Cisco router passwords.</td>
</tr>
<tr>
<td>PGPPASS</td>
<td>Cracks PGP key rings. Note that the Caligula virus is designed to steal PGP key rings by FTPing them from the victim’s PC to a server on the Internet.</td>
</tr>
<tr>
<td>Slurpie</td>
<td>UNIX password cracker designed to work in a distributed mode, so the cracking task can be shared by multiple machines (the same concept followed by the commercial cracking tool from AccessData).</td>
</tr>
<tr>
<td>Webcracker</td>
<td>Remote brute-force tool to crack password-protected Web sites.</td>
</tr>
<tr>
<td>Aimpw</td>
<td>AOL Instant Messenger Password Cracker.</td>
</tr>
<tr>
<td>!Bios</td>
<td>Decrypts the passwords used in common BIOSs, including those from IBM, American Megatrends, Award, and Phoenix.</td>
</tr>
<tr>
<td>Zipcrack &amp; PkCrack</td>
<td>Cracks encrypted ZIP and PKZIP files.</td>
</tr>
<tr>
<td>Cain</td>
<td>Anyone with access to Windows 9x can use Cain to locate and retrieve cached passwords in the Registry or in .pwl files. It typically recovers passwords for logon, local and remote shares, screen savers, and dial-up access. An associated agent called ABLE, when placed on a victim’s machine, can be remotely accessed to collect passwords.</td>
</tr>
</tbody>
</table>

Tips for Selecting Good Passwords

**DO**

Use a phrase to create a strong, easy-to-remember password. For example:

- Create a phonetic sentence using the pronounced sounds of letters, numbers, and special characters.
  
  I10D24GET   “I tend to forget”
  RULOSTIM    “Are you lost? I am!”

- Use short, unrelated words with numbers and special characters.
  
  GO2SEPT     ROSE49SPORT

- Use the first letter from each word in a phrase.
  
  4s&7yaoff   “Four score and seven years ago our forefathers”

Use more than just A to Z.

Use mixed-case and embedded numbers.

Use a minimum of seven characters.

Use punctuation, spaces, and symbols where possible (not all systems support them).

**DO NOT USE**

- Words found in the dictionary
- Common acronyms or words associated with the work environment
- Proper names
- Foreign languages
- Consecutive keys on the keyboard
• All the same character (e.g., 1111111)
• Personal numbers (e.g., Social Security number, phone number)
• Words that are easily personally identifiable (e.g., pets' names, children’s names, license plate numbers, addresses)

OTHER TIPS
• Do not write passwords down.
• Do not tell anyone your password.
• Be alert—watch for shoulder surfers.
• Change passwords often.
• Use unique passwords (do not reuse passwords).
• Do not use the same password for every system.

Social Engineering
A very popular method to bypass or exceed normal computer access is social engineering. Social engineering occurs when an intruder successfully convinces someone with authorized access to divulge information that may be used to gain access to a system. This can be accomplished without the authorized user even realizing it is being done. The intruder may lie about his or her identity, position, location, or other factors to convince an employee or authorized user that access should be granted.

A person posing as a member of the information technology department and requesting an authorized user to provide a password to conduct system checks is just one example of social engineering. Although this approach may not seem to be an avenue that is easily used, it is surprising how often it succeeds. For this reason, many Internet service providers post warnings on their Web sites to not divulge passwords when requested. Addressing the threat of social engineering and posting security alerts within an organization when such threats are being targeted toward a group are the best tactics to prevent a successful compromise via this route.

Wireless Attacks
Many organizations are now using wireless connectivity to communicate within their environment. Unfortunately, the same wireless
capabilities may be used by unauthorized users to gain access to the systems if proper security is not implemented. Numerous consulting firms now conduct “war driving” exercises to check for unsecured wireless access points that may be used as points of access into a system. (War driving refers to scanning for wireless access points that may not be authorized, just as war dialing refers to the search for modems.) Numerous cases have been documented where systems were able to be accessed via the wireless access points. For example, if two companies share a building and both have wireless networks, without proper controls employees from one company may be able to access the other company’s network. There have also been cases where access was obtained from the parking lot of a company with a wireless network. The best way to protect a wireless network is to include a security architect as part of the initial design of the system to address security concerns.

TCP/IP Design Limitations

**SYN Flooding**

A network connection is normally accomplished through a series of signals and responses. First, the client wishing to connect to the server sends a SYN message to indicate a connection is needed. Next, the server responds with a SYN-ACK signal, recognizing or acknowledging the request. Finally, the client responds back to the server with an ACK signal and the connection is opened, enabling data to flow between the two.

In a SYN flood attack, numerous SYN signals are sent to the server with addresses that are actually unavailable. The server will allocate records for these connections as if they were legitimate requests. As the number of “false” requests increases, the server becomes tied up in dealing with the “false” requests and is no longer available to accept legitimate connection requests.

**Smurf**

Although many readers will tend to think of a little blue cartoon guy when hearing this term, smurf is actually a type of denial-of-service attack that has been around for many years. A smurf attack relies on
routers being configured to allow directed broadcasts and requires an intermediary site that may also become a victim of the attack. Basically, the attacker sends a “ping” or Internet Control Message Protocol (ICMP) echo request packet to the broadcast address of a network. The source address of the ping request is modified so that it appears to come from a specific host that is the victim of the attack. When the broadcast address receives the request, it forwards the packet to every live host on that network, which may include hundreds or even thousands of systems. Each host replies to the ping request with an ICMP echo reply. The hundreds or even thousands of replies are all sent back to the source address of the victim, potentially overwhelming it with network traffic and creating a denial of service. Note that the smurf attack is not a design limitation of the TCP/IP protocol suite, but rather an abuse of behavior that is normal and expected within the protocol.

Coding Oversight

Multiple vulnerabilities exist in various platforms due to oversight or holes in the program code that intruders can discover and use to their advantage. For example, sendmail in UNIX programs has experienced several problems over the years that have been manipulated to gain unauthorized access to systems. This is one reason that new versions of sendmail (and other programs) have been released over time.

The best safeguard to prevent coding problems from being used as an attack avenue for a system is to monitor vendor advisories and alerts for problems as they are discovered. Patches and updates should be implemented as appropriate. If the vendor provides an MD5 checksum for the patch, the patch’s or program’s checksum should be checked to ensure that it matches the one provided by the vendor prior to installing the program. In a few cases, patches that were made available to address a vulnerability have been compromised and altered so that personnel downloading the patch actually installed a Trojan horse version that presented new problems. Check-
ing the MD5 checksum helps to prevent the installation of invalid patches.

**Buffer Overflows**

A buffer is an area of the computer’s memory used to temporarily store data. Typically, a buffer of a set size is created with enough memory for the needs of the program. However, when a larger-than-normal amount of data is sent to the buffer, the data is routinely appended at the end of the buffer, possibly overwriting other instructions or memory. This may cause data to be lost or corrupted, or even cause the program to crash. The use of buffer overflows to create a denial of service has increased over the past four to five years. In some cases, malicious commands may be embedded in the overflow data, then executed on the computer when the program freezes or crashes.

**Common Oversights in Web Common Gateway Interface (CGI) and Network Interfaces**

Many Web applications have been written to accept unfiltered user input from any source and act on that input. As a result, the Web server can be tricked into doing something that is neither intended nor expected by the authorized users. For example, Common Gateway Interface (CGI) programs are used with Web servers to provide input from a Web page that may be beyond HTML’s ability to manipulate. Some problems have arisen with some of these programs accepting input without question and passing it to a shell program for execution. As a result, intruders have been able to gain unauthorized access to systems as well as “hack” multiple Web pages.

One notable CGI program that has been used to gain access to Web servers is called phf. Multiple teams have provided advisories on the exploits associated with phf, and Web servers should be checked to ensure that proper safeguards are in place to prevent this program from being manipulated. Ways to manipulate this program have also been detailed in hacker publications and on hacker Web sites.

The vulnerabilities associated with CGI programs have been known for several years, yet some Web pages still fall prey to this sort of attack. Other, more recent tactics used to trick or manipulate Web
servers, Web server applications, or the underlying operating systems include the following:

- Entering malformed URL requests and HTTP headers
- Creating a buffer overflow by exceeding the expected length of input in form fields
- Embedding metacharacters in URLs or form input data
- SQL injection, where malicious SQL commands are issued.

### Malicious Logic

Malicious logic—viruses, worms, Trojan horses, and logic bombs—robs its victims of productivity and jeopardizes every organization’s information security and infrastructure. These computer programs, which are written with spiteful intent, perform unauthorized routines to damage and destroy data or to degrade system performance. Although the level and type of damage varies, the impact can be enormous, with the greatest costs usually tallied in the form of person-hours expended on recovery. There is virtually no way to keep an organization completely free from malicious logic, but there are steps everyone can take to proactively address the problem. First, one must understand the types of codes and recommended courses of action.

### The Computer Virus

The form of malicious logic that is familiar to most people is the computer virus. A virus is a self-replicating program whose purpose is to propagate to as many different places as possible. Viruses accomplish this goal by modifying other programs to include copies of themselves through an (unknowing) act of a user. Although viruses have received much attention from the media in recent years, they have actually been in existence since 1980. In his thesis written in 1984, Fred Cohen termed the phrase “computer virus” because of its similarity to a biologic virus, which also replicates. Table 4–2 compares computer and biologic viruses.

A virus is normally named based on some function that it performs or a name that is used in its “signature.” For example, Wazzu is
a macro virus that first appeared in the late 1990s. When the virus launched its payload, it would randomly insert the term “Wazzu” throughout the contents of an infected Microsoft Word document.

Frequently, virus writers will change an existing virus slightly
and re-release the modified code to create a new strain of the virus. These new strains are usually identified with the original virus name and an alphabetic extension. For example, the Wazzu macro virus was modified several times to create new strains of the virus denoted as Wazzu.A, Wazzu.B, and so on.

New strains of existing viruses account for the population explosion among computer viruses over the years. In 1992 and 1993, the average number of new computer viruses per month ranged from 100 to 150.1 Most of these “new” viruses were not actually brand-new viruses, but rather new strains of existing viruses. In January 1997, the average number of new discoveries was 200 per month, with a total of more than 8500 known computer viruses. The numbers have continued to rise with time. In October 1999, Symantec released an updated signature file for its Norton antivirus software that increased the number of detected viruses by 646 in just four days! As of March 2003, the Norton antivirus software included signatures to detect more than 63,000 viruses.

**Virus Types**

There are five main types of computer viruses: file infectors, partition-sector viruses, boot-sector viruses, companion viruses, and macro viruses. The type of virus is determined by how the code propagates.

- **File infectors** replicate by inserting themselves into executable files. Usually, the virus code is written at the beginning of a targeted file, so it is executed immediately, or appended to the end. If the code is written somewhere other than the beginning of the file, the start-up sequence of instructions in the file is normally modified to transfer program control to the malevolent code.
- **Partition-sector infectors** contaminate the partition record of a hard disk. First, the virus copies and stores the whole partition sector somewhere else on the hard disk. Then the virus copies itself to the partition sector and executes when the computer is booted from the hard drive.

---

• **Boot-sector viruses** infect both floppy diskettes and hard disks by changing or replacing the boot-sector program with a copy of itself. This type of virus is most frequently spread by forgotten floppy diskettes left in disk drives. It can be considered a special case of the file infector virus, which infects only the boot-sector program.

• **A companion virus** searches for executable programs with the .exe or .bat extension and creates a copy of itself with the same name but a .com extension. The virus is activated when a user tries to execute the legitimate file from the command line. After the virus completes its routine, control returns to the original file. Companion viruses tend to spread only within a computer system and are normally easy to spot through both manual and automatic means.

• **Macro viruses** were discovered in the summer of 1995. Instead of infecting an executable program file or disk boot sector, this type of virus infects an application’s document files. The virus contains a set of application-specific macro commands that automatically execute in an unsolicited manner and then spread to the application’s documents.

In addition to these types of viruses, two other terms frequently used to describe viruses are *multipartite* and *polymorphic*. A multipartite virus combines two or more different infection methods, and a polymorphic virus changes itself with each virus infection. The polymorphic virus normally includes some sort of “mutation engine” to change itself during an infection in an attempt to elude detection.

Symptoms indicating the presence of a computer virus vary. Some of the typical signs indicating a virus may be present include the following:

• Display of an unusual message
• Missing files
• Files of increased size
• A slowdown in the operation of the computer
• A sudden lack of disk space
• Inability to access the disk drive

These symptoms may be attributed to other problems as well, so one should not assume the problem is automatically a virus without undertaking further investigation.
Important Steps to Remain Virus-Free

Regardless of the type of virus, some basic steps can go a long way toward ensuring that your computer system remains virus-free:

- Document and share standard antivirus procedures.
- Train users to consistently follow standards.
- Conduct regular scans with antivirus software.
- Update antivirus software as new signature files become available from software vendors.
- Use antivirus software at various levels within the information infrastructure.
- Keep computer users informed of the latest threats.
- Remind end users not to open unknown e-mail attachments without first scanning for viral code.
- Do not leave diskettes in floppy drives when computers are turned off.
- In Microsoft Word applications, the global template, typically a file called Normal.dot, should be scanned for viruses and set to read only.
- Any auto-run and auto-open features on e-mail programs and browsers should be disabled.
- The CMOS boot sequence could be changed to start with the C drive first, then the A drive.

Ultimately, a particularly nasty virus may escape detection and cause the loss of critical data on workstations or file servers. In those situations, a conscientious file backup program is absolutely essential. Even if the virus code is backed up on tape or other media, system administrators can update virus definitions to remove it. The code will not run until the operating system is running again, so restoration of data onto a new PC updated with antivirus protection becomes possible.

Other Forms of Malicious Logic

The following descriptions provide a breakdown of the other main types of malicious logic. As the writers of these types of code build off
of one another’s knowledge, the programs that result from their aggregate work is beginning to cross or blur some of these definitional boundaries. In some cases, it may be very difficult to determine whether the code is a virus, a worm, or a Trojan horse. The following definitions are the commonly accepted terms as of the writing of this book:

- **A worm** is “an independent program that reproduces by copying itself from one system to another, usually over a network. Like a virus, a worm may damage data directly or it may degrade system performance by tying up resources and even shutting down a network.”
  
  The worm written and released by Robert Morris, Jr., in late 1988 and LoveLetter are examples.

- **A Trojan horse** is a program or routine concealed in software that appears to be harmless. Trojan horses are not viruses and do not replicate like viruses, but they may contain or include a worm or virus as part of the package. The most common way to remove a Trojan horse is to simply delete the identified Trojan application. Trojan horses are quickly becoming one of the top security threats to computer systems. Back Orifice, Back Orifice 2000 (or BO2K), and Netbus are all examples of Trojan horses.

- **A logic bomb** is a software program that is triggered by a timing device (e.g., a date or event) to launch its payload. The payload may release a virus or worm, or perform some other type of attack. This logic bomb is a popular device among disgruntled employees.

Most antivirus software vendors try to include worms and Trojan horses in their signature strings, but their success rates for detection and eradication vary greatly. System administrators cannot assume the antivirus software will detect and solve all malicious code problems.

---

Virus Hoaxes and Urban Legends

Any discussion of malicious logic should address the topic of virus hoaxes and urban legends. Although these problems do not actually involve malicious code, they are a major nuisance for system administrators and can present potential security risks. For example, the forwarding of messages to multiple recipients has been known to shut down e-mail servers. This problem should be addressed through end-user education programs.

“A virus’s true prey is not the computer, but the good will and ignorance of the users.” Similarly, virus hoaxes and urban legends prey on users to spread fear throughout the Internet. A virus hoax is an e-mail warning of some new virus rumored to be in circulation on the Internet. Some of the warnings are very well written and convince well-intentioned users to forward them to others. Unfortunately, mass-forwarding of the false e-mail spreads panic and productivity suffers (not to mention the time wasted in debunking false allegations, reading bogus postings, and tying up resources with multiple forwards of group mail).

Urban legends are very similar to virus hoaxes, except they forward a warning about some other major event, problem, or impending catastrophe. The story that aired on Dateline in late 1999 about the use of suntan lotion causing blindness in children is an excellent example of an urban legend, as well as the e-mail stating that Congress is planning to pass legislation that will tax all e-mail. Several Internet resources may be used to confirm whether a story is a virus hoax or urban legend. One of the best sites is http://www.vmyths.com/. If the warning directs the recipient to “forward the e-mail to everyone you know,” it’s a pretty safe bet that it is a hoax.

The bottom line—do not e-mail virus hoaxes or urban legends. If in doubt, check with a reliable source regarding their validity before taking any further action. The safest action is to press the Delete key. Some organizations have experienced serious system performance degradation when well-intentioned users forwarded warnings to multiple sites. It is far better to leave the warnings to the experts than to

---

try and spread the “word,” which merely adds to the hype and may hurt one’s reputation in the process.

In some cases, well-intentioned people have caused denial-of-service attacks by responding to hoaxes received via the Internet. In the late 1990s, a hoax circulated on the Internet concerning a young girl who was dying from cancer. Her dying wish was to raise money for cancer research. If the recipient responded by e-mailing the American Cancer Society, the sender promised to donate $0.02 toward cancer research. The response from well-intentioned recipients was so great that it overloaded the systems at the American Cancer Society, which suffered a denial-of-service attack.

Computer viruses rob workers of productivity, redirect the attention of system administrators from more severe security threats, and can jeopardize the security of the organization’s information infrastructure. Yet, there is virtually no way to keep an organization completely free from the viral code. To keep the problem in perspective, an organization must take steps to proactively address the problem.

The first step is to increase the overall awareness level of all end users. Addressing topics such as the importance of the macro warning box in Microsoft Word applications and the need to scan attachments for malicious code prior to opening them can go a long way toward eliminating large-scale infections. Increased awareness will also help eliminate problems attributed to the virus hoaxes.

In conjunction with awareness education, implementing antivirus software throughout the enterprise with automatic updates of signature files remains the best course of action developed to combat viruses to date. The signature file updates may be programmed to occur during off-peak hours if desired. Whenever possible, antivirus software and signature file updates should be implemented in a way that cannot be bypassed by end users.

Finally, a check of configuration settings should be made to ensure that auto-open features are disabled and global templates are set to read-only access. These two changes alone may help reduce the penetration achieved by the next rapidly spreading virus. By implementing a proactive antivirus program, the effects of malicious code can be greatly minimized.
Summary

This chapter presented an overview of the major categories of computer attacks as well as some specific attack vectors. Although the attacks described are in no way inclusive of all the attacks known, the descriptions should provide some insight into some of the many “needles in a haystack” that the incident response team must be prepared to detect and manage. The chapter concluded with a discussion of malicious logic infections, particularly viruses. Several pointers for protecting a computer or network from a virus infection were presented.

Incident response teams must be prepared to deal with any type of attack that may be launched against the operating systems used in their infrastructure. Not every form of attack may be used on every operating system. For example, UNIX users should ensure that they are using the latest version of sendmail to prevent problems associated with that program. The best advice for any incident response team about identifying and knowing the specifics of attacks is to first take a look at the SANS top 20 list and see how it applies to your infrastructure.

One of the biggest complaints voiced by system administrators is that there are too many vulnerabilities to keep up with. Table 4–3 shows the number of vulnerabilities reported to CERT CC and the number of advisories the center issued regarding the most serious vulnerabilities. Although the problem of so many vulnerabilities cannot easily be reduced, the best way to avoid becoming overwhelmed is to always remain alert to the advisories and bulletins issued. The CERT CC advisories serve as a good starting point. Vendor Web sites and bulletins should also be monitored as applicable for the systems in one’s infrastructure. Some consulting firms and managed security service providers have begun offering services in recent years that focus on vulnerability management as well. These services may provide a good avenue for keeping current with the latest announcements, but they should not stand alone in addressing the vulnerabilities in an organization’s infrastructure. The warnings or advice presented must be coupled with knowledge of the internal environment to identify vulnerable platforms and decide what and when to patch. The best advice to remember about monitoring vulnerabilities
is to rely on multiple sources and not keep “all of the eggs in one basket.” Relying heavily on only one source will eventually catch the organization off-guard.

Chapters 3 and 4 focused on the corners of the incident response puzzle by discussing the importance of terminology and identifying numerous types of attacks. Although this book is not intended to be an in-depth technical review, the information presented in this chapter should give some idea of the threats that an incident response team must be prepared to address. We now return to putting the puzzle together, by identifying the other considerations that should be taken into account during the initial formation of a team. Let’s continue by discussing operational and resource issues.

Table 4-3  Number of Vulnerabilities and Number of More Significant Problems Reported Annually to CERT CC that Resulted in Advisories

<table>
<thead>
<tr>
<th>Year</th>
<th>Vulnerabilities Reported</th>
<th>Advisories Issued</th>
<th>Vendor Bulletins Issued</th>
</tr>
</thead>
<tbody>
<tr>
<td>1995</td>
<td>171</td>
<td>18</td>
<td>10</td>
</tr>
<tr>
<td>1996</td>
<td>345</td>
<td>27</td>
<td>20</td>
</tr>
<tr>
<td>1997</td>
<td>311</td>
<td>28</td>
<td>16</td>
</tr>
<tr>
<td>1998</td>
<td>262</td>
<td>13</td>
<td>13</td>
</tr>
<tr>
<td>1999</td>
<td>417</td>
<td>17</td>
<td></td>
</tr>
<tr>
<td>2000</td>
<td>1,090</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td>2001</td>
<td>2,437</td>
<td>37</td>
<td></td>
</tr>
<tr>
<td>2002</td>
<td>4,129</td>
<td>37</td>
<td></td>
</tr>
</tbody>
</table>
