Chapter 5

Malware

Chapter Objectives

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

- Understand viruses (worms) and how they propagate, including the Sobig and Sasser types
- Have a working knowledge of several specific virus outbreaks
- Understand how virus scanners operate
- Understand what a Trojan horse is and how it operates
- Have a working knowledge of several specific Trojan horse attacks
- Grasp the concept behind the buffer-overflow attack
- Have a better understanding of spyware and how it enters a system
- Defend against each of these attacks through sound practices, antivirus software, and antispyware software

Introduction

In Chapter 4, "Denial of Service Attacks," we examined the denial of service attack. It is a very common attack and one that can easily be perpetrated. In this chapter, you will continue your examination of security threats by learning about several other types of attacks. First, you will learn about virus outbreaks. Our discussion will focus on information about how and why virus attacks work, including their deployment through Trojan horses. This chapter is not a "how to create your own virus" tutorial, but rather an introduction to the concepts underlying these attacks as well as an examination of some specific case studies.

This chapter will also explore buffer-overflow attacks, spyware, and several other forms of malware. Each of these brings a unique approach to an attack, and each needs to be considered when defending

a system. Your ability to defend against such attacks will be enhanced by expanding your knowledge of how they work. In the exercises at the end of the chapter, you will have the opportunity to research preventive methods for viruses and to try out antivirus methods from McAfee and Norton.

Viruses

By definition, a computer virus is a program that self-replicates. Generally, a virus will also have some other unpleasant function, but the self-replication and rapid spread are the hallmarks of a virus. Often this growth, in and of itself, can be a problem for an infected network. The last chapter discussed the MyDoom virus and the effects of its rapid, high-volume scanning. Any rapidly spreading virus can reduce the functionality and responsiveness of a network. Simply by exceeding the traffic load that a network was designed to carry, the network may be rendered temporarily nonfunctional. The infamous I Love You virus actually had no negative payload, but the sheer volume of emails it generated bogged down many networks.

How a Virus Spreads

A virus will usually spread primarily in one of two ways. The first is to simply scan your computer for connections to a network and then copy itself to other machines on the network to which your computer has access. This is actually the most efficient way for a virus to spread. However, this method requires more programming skill than other methods. The more common method is to read your email address book and email itself to everyone in your address book. Programming this is a trivial task, which explains why it is so common.

The latter method is, by far, the most common method for virus propagation, and Microsoft Outlook may be the one email program most often hit with such virus attacks. The reason is not so much a security flaw in Outlook as it is the ease of working with Outlook. All Microsoft Office products are made so that a legitimate programmer who is writing software for a business can access many of the application's internal objects and thereby easily create applications that integrate the applications within the Microsoft Office suite. For example, a programmer could write an application that would access a Word document, import an Excel spreadsheet, and then use Outlook to automatically email the resulting document to interested parties. Microsoft has done a good job of making this process very easy, for it usually takes a minimum amount of programming to accomplish these tasks. Using Outlook, it takes less than five lines of code to reference Outlook and send out an email. This means a program can literally cause Outlook itself to send emails, unbeknownst to the user. There are numerous code examples on the Internet that show exactly how to do this, free for the taking. For this reason, it does not take a very skilled programmer to be able to access your Outlook address book and automatically send emails. Essentially, the ease of programming Outlook is why there are so many virus attacks that target Outlook.

While the overwhelming majority of virus attacks spread by attaching themselves to the victim's existing email software, some recent virus outbreaks have used other methods for propagation, such

as their own internal email engine. Another virus propagation method is to simply copy itself across a network. Virus outbreaks that spread via multiple routes are becoming more common.

The method of delivering a payload can be rather simplistic and rely more on end-user negligence than on the skill of the virus writer. Enticing users to go to websites or open files they should not is a common method for delivering a virus and one that requires no programming skill at all. Regardless of the way a virus arrives at your doorstep, once it is on your system, it will attempt to spread and, in many cases, will also attempt to cause some harm to your system. Once a virus is on your system, it can do anything that any legitimate program can do. That means it could potentially delete files, change system settings, or cause other harm.

Types of Viruses

There are many different types of viruses. In this section we will briefly look at some of the major virus types. Viruses can be classified by either their method for propagation or their activities on the target computers.

- **Macro:** Macro viruses infect the macros in office documents. Many office products, including Microsoft Office, allow users to write mini-programs called macros. These macros can also be written as a virus. A macro virus is written into a macro in some business application. For example, Microsoft Office allows users to write macros to automate some tasks. Microsoft Outlook is designed so that a programmer can write scripts using a subset of the Visual Basic programming language, called Visual Basic for Applications (VBA). This scripting language is, in fact, built into all Microsoft Office products. Programmers can also use the closely related VBScript language. Both languages are quite easy to learn. If such a script is attached to an email and the recipient is using Outlook, then the script can execute. That execution can do any number of things, including scanning the address book, looking for addresses, sending out email, deleting email, and more.
- **Multi-partite:** Multi-partite viruses attack the computer in multiple ways for example, infecting the boot sector of the hard disk and one or more files.
- **Memory Resident:** A memory-resident virus installs itself and then remains in RAM from the time the computer is booted up to when it is shut down.
- **Armored:** An armored virus uses techniques that make it hard to analyze. Code confusion is one such method. The code is written such that if the virus is disassembled, the code won't be easily followed. Compressed code is another method for armoring the virus.
- **Sparse infector:** A sparse infector virus attempts to elude detection by performing its malicious activities only sporadically. With a sparse infector virus, the user will see symptoms for a short period, then no symptoms for a time. In some cases the sparse infector targets a specific program but the virus only executes every 10th time or 20th time that target program executes. Or a sparse infector may have a burst of activity and then lie dormant for a period of time. There are a number of variations on the theme, but the basic principle is the same: to reduce the frequency of attack and thus reduce the chances for detection.

- **Polymorphic:** A polymorphic virus literally changes its form from time to time to avoid detection by antivirus software. A more advanced form of this is called the Metamorphic virus; it can completely change itself.

Virus Examples

The threat from virus attacks cannot be overstated. While there are many web pages that give virus information, in my opinion, there are only a handful of web pages that consistently give the latest, most reliable, most detailed information on virus outbreaks. Any security professional will want to consult these sites on a regular basis. You can read more about any virus, past or current, at the following websites:

- http://www.pctools.com/security-news/top-10-computer-viruses/
- https://www.us-cert.gov/publications/virus-basics
- http://www.techrepublic.com/pictures/the-18-scariest-computer-viruses-of-all-time/

The following sections will look at several real-world virus outbreaks. We will examine very recent viruses as well as some examples from 10 or more years in the past. This should give you a fairly complete overview of how viruses behave in the real world.

Rombertik

Rombertik wreaked havoc in 2015. This malware uses the browser to read user credentials to websites. It is most often sent as an attachment to an email. Perhaps even worse, in some situations Rombertik will either overwrite the master boot record on the hard drive, making the machine unbootable, or begin encrypting files in the user's home directory.

Gameover ZeuS

Gameover ZeuS is a virus that creates a peer-to-peer botnet. Essentially, it establishes encrypted communication between infected computers and the command and control computer, allowing the attacker to control the various infected computers. In 2014 the U.S. Department of Justice was able to temporarily shut down communication with the command and control computers; then in 2015 the FBI announced a reward of $3 million for information leading to the capture of Evgeniy Bogachev for his alleged involvement with Gameover ZeuS.

A command and control computer is the computer used in a botnet to control the other computers. These are the central nodes from which a botnet will be managed.

CryptoLocker and CryptoWall

One of the most widely known examples of ransomeware is the infamous CryptoLocker, first discovered in 2013. CryptoLocker utilized asymmetric encryption to lock the user's files. Several varieties of CryptoLocker have been detected.

CryptoWall is a variant of CryptoLocker first found in August 2014. It looked and behaved much like CryptoLocker. In addition to encrypting sensitive files, it would communicate with a command and control server and even take a screenshot of the infected machine. By March 2015 a variation of CryptoWall had been discovered that is bundled with the spyware TSPY_FAREIT.YOI and actually steals credentials from the infected system, in addition to holding files for ransom.

FakeAV

This virus first appeared in July 2012. It affected Windows systems ranging from Windows 95 to Windows 7 and Windows server 2003. This was a fake antivirus (thus the name FakeAV) that would pop up fake virus warnings. This was not the first such fake antivirus malware, but it was one of the more recent ones.

MacDefender

This virus is very interesting for multiple reasons. First, it specifically targets Macintosh computers. Most experts have long agreed that Apple products remained relatively virus free simply because their products did not have enough market share to attract the attention of virus writers. It has long been suspected that if Apple garnered a greater market share, it would also begin to get more virus attacks. That has proven to be true.

This virus was first seen in the early months of 2011. It is embedded in some web pages, and when a user visits those web pages, she is given a fake virus scan that tells her she has a virus and it needs to be fixed. The "fix" is actually downloading a virus. The point of the virus is to get end users to purchase the MacDefender "antivirus" product. This is the second reason this case is noteworthy. Fake antivirus attacks, also known as scareware, have been becoming increasingly common.

Troj/Invo-Zip

This particular worm is a classic worm/Trojan horse that was first reported in mid-2010. It is transmitted as a zip file attached to an email. The email claims that the zip file contains data related to an invoice, tax issue, or similar urgent paperwork. This is a classic example of attempting to entice the recipient to open the attachment. And in this case, the recipients most likely to be enticed would be businesspeople.

If the recipient does open the attachment, then he will have installed spyware on his machine that would first disable the firewall and then start attempting to capture information including financial data. It even takes screenshots of the user's desktop.

W32/Netsky-P

This worm was first found in 2006 and was still going around in 2011. It is a fairly typical virus in that it spreads primarily through email, but it also uses file sharing utilities to copy itself. It copies itself to various directories and shared folders. In one interesting twist, it attempts to copy itself to

C:\WINDOWS\FVProtect.exe. The name would make many people (including otherwise technically savvy people) think this program was actually part of some antivirus utility. It also copies itself to C:\WINDOWS\userconfig9x.dll. Again, it would appear to be a system file, thus making people less likely to delete it.

This is also a classic worm/virus in that the email it sends has a fairly generic title and content that attempts to get the recipient to open the attachment. For example, the body of the message might say something like, "Please see the attached file for details" or "Your file is attached."

The Sobig Virus

This is obviously not a recent virus, as it was first found in 2003. However, it is an excellent virus to study because it received the most media attention and perhaps caused the most harm in 2003. The first interesting thing to study about this virus was how it utilized a multimodal approach to spreading. This means that it used more than one mechanism to spread and infect new machines. It would copy itself to any shared drives on your network and would email itself out to everyone in your address book. For these reasons, this virus was particularly virulent, which is also why it is important to study.

FYI: Virulent Virus

The term *virulent* means essentially the same thing in reference to a computer virus as it does to a biological virus. It is a measure of how rapidly the infection spreads and how easily it infects new targets.

In the case of Sobig, if one person on a network was unfortunate enough to open an email containing the virus, not only would his machine be infected, but so would every shared drive on that network to which this person had access. However, Sobig, like most email-distributed virus attacks, had tell-tale signs in the email subject or title that could be used to identify the email as one infected by a virus. The email would have some enticing title such as "here is the sample" or "the document" to encourage you to be curious enough to open the attached file. The virus would then copy itself into the Windows System directory.

This particular virus spread so far and infected so many networks that the multiple copying of the virus alone was enough to bring some networks to a standstill. This virus did not destroy files or damage the system, but it generated a great deal of traffic that bogged down the networks infected by it. The virus itself was of moderate sophistication. Once it was out, however, many variants began to spring up, further complicating the situation. One of the effects of some variants of Sobig was to download a file from the Internet that would then cause printing problems. Some network printers would just start printing junk. The Sobig.E variant would even write to the Windows Registry, causing itself to be in the computer startup (F-Secure, 2003). These complex characteristics indicate that the creator knew how to access the Windows Registry, access shared drives, alter the Windows startup, and access Outlook.

This brings up the issue of virus variants and how they occur. In the case of a biological virus, mutations in the genetic code cause new virus strains to appear, and the pressures of natural selection allow some of

these strains to evolve into entirely new species of viruses. Obviously, the biological method is not what occurs with a computer virus. With a computer virus, what occurs is that some intrepid programmer with malicious intent will get a copy of a virus (perhaps her own machine becomes infected) and will then reverse-engineer it. Since many virus attacks are in the form of a script attached to an email, unlike traditionally compiled programs, the source code of these attacks is readily readable and alterable. The programmer in question then simply takes the original virus code, introduces some change, and rereleases the variant. Frequently, the people who are caught for virus creation are actually the developers of the variants who lacked the skill of the original virus writer and therefore were easily caught.

The Mimail Virus

This is another older virus that is still worth studying. The Mimail virus did not receive as much media attention as Sobig, but it had its intriguing characteristics. This virus not only collected email addresses from your address book, but also from other documents on your machine (Gudmundsson, 2004). Thus, if you had a Word document on your hard drive and an email address was in that document, Mimail would find it. This strategy meant that Mimail would spread farther than many other viruses. Mimail had its own built-in email engine, so it did not have to "piggy back" off your email client. It could spread regardless of what email software you used.

These two variations from most virus attacks made Mimail interesting to people who study computer viruses. There are a variety of techniques that allow you to programmatically open and process files on your computer; however, most virus attacks do not employ them. The scanning of the document for email addresses indicates a certain level of skill and creativity on the part of the virus writer. In this author's opinion, Mimail was not the work of an amateur, but rather a person with professional-level programming skill.

The Bagle Virus

This is the last of the "historical viruses" that we will examine. It is noteworthy in that it combined email attachments along with the fake virus warning. The Bagle virus began to spread rapidly in the fourth quarter of 2003. The email it sent claimed to be from your system administrator. It would tell you that your email account had been infected by a virus and that you should open the attached file to get instructions. Once you opened the attached file, your system was infected. This virus was particularly interesting for several reasons. To begin with, it spread both through email and copying itself to shared folders. Second, it could scan files on your PC looking for email addresses. Finally, it would disable processes used by antivirus scanners. In biological terms, this virus took out your computer's "immune system." The disabling of virus scanners was a new twist that indicated at least moderate programming skills on the part of the virus creator.

A Nonvirus Virus

Another new type of virus has been gaining popularity in the past few years, and that is the "nonvirus virus" or, put simply, a hoax. Rather than actually writing a virus, a hacker sends an email to every

address he has. The email claims to be from some well-known antivirus center and warns of a new virus that is circulating. The email instructs people to delete some file from their computer to get rid of the virus. The file, however, is not really a virus but part of a computer's system. The jdbgmgr.exe virus hoax used this scheme (Rhode Island Soft Systems, Inc., 2003). It encouraged the reader to delete a file that was actually needed by the system. Surprisingly, a number of people followed this advice and not only deleted the file, but promptly emailed their friends and colleagues to warn them to delete the file from their machines.

FYI: The Morris Internet Worm

The Morris worm was one of the first computer worms ever to be distributed over the Internet. And it was certainly the first to gain any significant media attention.

Robert Tappan Morris, Jr., then a student at Cornell University, wrote this worm and launched it from an MIT system on November 2, 1988. Morris did not actually intend to cause damage with the worm. Instead, he wanted the worm to reveal bugs in the programs it exploited in order to spread. However, bugs in the code allowed an individual computer to be infected multiple times, and the worm became a menace. Each additional "infection" spawned a new process on the infected computer. At a certain point, the high number of processes running on an infected machine slowed down the computer to the point of being unusable. At least 6,000 UNIX machines were infected with this worm.

Morris was convicted of violating the 1986 Computer Fraud and Abuse Act and was sentenced to a $10,000 fine, 3 years probation, and 400 hours of community service. But perhaps the greatest impact of this worm was that it led to the creation of the Computer Emergency Response Team (CERT). CERT is an organization hosted at Carnegie Mellon University (www.cert.org/) that is a repository for security bulletins, information, and guidelines. CERT is a source that any security professional should be familiar with.

Flame

No modern discussion of viruses would be complete without a discussion of Flame. This virus, which first appeared in 2012, targeted Windows operating systems. The first item that makes this virus notable is that it was specifically designed by the U.S. government for espionage. It was discovered in May 2012 at several locations, including Iranian government sites. Flame is spyware that can monitor network traffic and take screenshots of the infected system.

Rules for Avoiding Viruses

You should notice a common theme with all virus attacks (except the hoax), which is that they want you to open some type of attachment. The most common way for a virus to spread is as an email attachment. This realization leads to some simple rules that will drastically reduce the odds of becoming infected with a virus.

- Use a virus scanner. McAfee and Norton (explored in the exercises at the end of this chapter) are the two most widely accepted and used virus scanners. However, Kaspersky and AVG are also good, reputable choices. Each costs about $30 per year to keep your virus scanner updated. Do it. Each antivirus has its proponents and detractors—I won't delve into the opinions on which is better. For most users, any of the four major antivirus programs would be effective. I rotate which one I use periodically, just so I can stay familiar with all of them.
- If you are not sure about an attachment, do not open it.
- You might even exchange a code word with friends and colleagues. Tell them that if they wish to send you an attachment, they should put the code word in the title of the message. Without seeing the code word, you will not open any attachment.
- Do not believe "security alerts" that are sent to you. Microsoft does not send out alerts in this manner. Check the Microsoft website regularly, as well as one of the antivirus websites previously mentioned.

These rules will not make your system 100% virus proof, but they will go a long way toward protecting your system.

Trojan Horses

Recall from earlier chapters that *Trojan horse* is a term for a program that looks benign but actually has a malicious purpose. We have already seen viruses that are delivered via a Trojan horse. You might receive or download a program that appears to be a harmless business utility or game. More likely, the Trojan horse is just a script attached to a benign-looking email. When you run the program or open the attachment, it does something else other than or in addition to what you thought it would. It might

- Download harmful software from a website.
- Install a key logger or other spyware on your machine.
- Delete files.
- Open a backdoor for a hacker to use.

It is common to find combination virus plus Trojan horse attacks. In those scenarios, the Trojan horse spreads like a virus. The MyDoom virus opened a port on your machine that a later virus, doomjuice, would exploit, thus making MyDoom a combination virus and Trojan horse.

A Trojan horse could also be crafted especially for an individual. If a hacker wished to spy on a certain individual, such as the company accountant, he could craft a program specifically to attract that person's attention. For example, if he knew the accountant was an avid golfer, he could write a program that computed handicap and listed best golf courses. He would post that program on a free web server. He would then email a number of people, including the accountant, telling them about the free software.

The software, once installed, could check the name of the currently logged-on person. If the logged-on name matched the accountant's name, the software could then go out, unknown to the user, and download a key logger or other monitoring application. If the software did not damage files or replicate itself, then it would probably go undetected for quite a long time. There have been a number of Trojan horses through the years. One of the earliest and most widely known was Back Orifice.

FYI: Virus or Worm?

As noted in the previous chapter, there is disagreement among the experts as to the distinction between a virus and a worm. Some experts would call MyDoom (as well as Sasser, which will be discussed later) a worm because it spread without human intervention. However, I would define a virus as any file that can self-replicate, and a worm as any program that can propagate without human interference. This is also the most common definition you will find among security experts.

Such a program could be within the skill set of virtually any moderately competent programmer. This is one reason that many organizations have rules against downloading *any* software onto company machines. I am unaware of any actual incident of a Trojan horse being custom tailored in this fashion. However, it is important to remember that those creating virus attacks tend to be innovative people.

It is also important to note that creating a Trojan horse does not require programming skill. There are free tools on the Internet, such as EliteWrapper, that allow someone to combine two programs, one hidden and one not. So one could easily take a virus and combine it with, for example, a poker game. The end user would only see the poker game, but when it was run it would launch the virus.

Another scenario to consider is one that would be quite devastating. Without divulging programming details, the basic premise will be outlined here to illustrate the grave dangers of Trojan horses. Imagine a small application that displays a series of unflattering pictures of Osama Bin Laden. This application would probably be popular with many people in the United States, particularly people in the military, intelligence community, or defense-related industries. Now assume that this application simply sits dormant on the machine for a period of time. It need not replicate like a virus because the computer user will probably send it to many of his associates. On a certain date and time, the software connects to any drive it can, including network drives, and begins deleting all files. If such a Trojan horse were released "in the wild," within 30 days it would probably be shipped to thousands, perhaps millions, of people. Imagine the devastation when thousands of computers begin deleting files and folders.

This scenario is mentioned precisely to frighten you a little. Computer users, including professionals who should know better, routinely download all sorts of things from the Internet, such as amusing flash videos and cute games. Every time an employee downloads something of this nature, there is the chance of downloading a Trojan horse. One need not be a statistician to realize that if employees continue that practice long enough, they will eventually download a Trojan horse onto a company machine. If they do, hopefully the virus will not be as vicious as the theoretical one just outlined here.

Because Trojan horses are usually installed by users themselves, the security countermeasure for this attack is to prevent downloads and installations by end users. From a law enforcement perspective, the

investigation of a crime involving a Trojan horse would involve a forensic scan of the computer hard drive, looking for the Trojan horse itself.

There are a number of tools, some free for download, that will help a person create a Trojan horse. One that I use in my penetration testing classes is eLiTeWrap. It is easy to use. Essentially, it can bind any two programs together. Using a tool such as this one, anyone can bind a virus or spyware to an innocuous program such as a shareware poker game. This would lead to a large number of people downloading what they believe is a free game and unknowingly installing malware on their own system.

The eLiTeWrap tool is a command line tool, but it is very easy to use. Just follow these steps:

1. Enter the file you want to run that is visible.
2. Enter the operation:
    - 1—Pack only
    - 2—Pack and execute, visible, asynchronously
    - 3—Pack and execute, hidden, asynchronously
    - 4—Pack and execute, visible, synchronously
    - 5—Pack and execute, hidden, synchronously
    - 6—Execute only, visible, asynchronously
    - 7—Execute only, hidden, asynchronously
    - 8—Execute only, visible, synchronously
    - 9—Execute only, hidden, synchronously
3. Enter the command line.
4. Enter the second file (the item you are surreptitiously installing).
5. Enter the operation.
6. When done with files, press Enter.

In Figure 5.1 you can see a demonstration that is appropriate for a classroom laboratory. In this example, two innocuous programs are combined into one Trojan horse. The programs chosen are simple Windows utilities that won't harm the computer. However, it illustrates how easy it would be to combine legitimate programs with malware, to deliver them to the target computer.

This is provided as an illustration of how easy it is to create a Trojan horse, not an encouragement for you to do so. It is important to understand just how easy this process is so you can understand the prevalence of malware. Any attachment or download should be treated with significant suspicion.

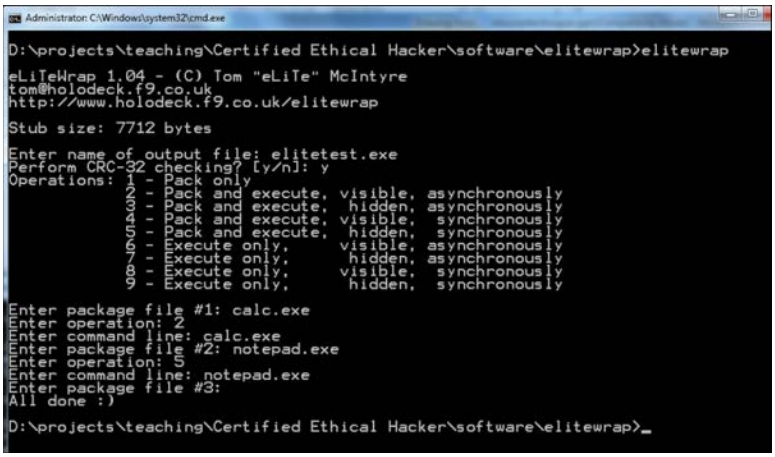

FIGURE 5.1 eLiTeWrap.

The Buffer-Overflow Attack

You have become knowledgeable about a number of ways to attack a target system: denial of service, virus, and Trojan horse. While these attacks are probably the most common, they are not the only methods. Another method of attacking a system is called a buffer-overflow (or buffer-overrun) attack. A buffer-overflow attack happens when someone tries to put more data in a buffer than it was designed to hold (searchSecurity.com, 2004a). Any program that communicates with the Internet or a private network must take in some data. This data is stored, at least temporarily, in a space in memory called a *buffer*. If the programmer who wrote the application was careful, when you try to place too much information into a buffer, that information is then either simply truncated or outright rejected. Given the number of applications that might be running on a target system and the number of buffers in each application, the chances of having at least one buffer that was not written properly are significant enough to cause any prudent person some concern.

Someone who is moderately skilled in programming can write a program that purposefully writes more into the buffer than it can hold. For example, if the buffer can hold 1024 bytes of data and you try to fill it with 2048 bytes, the extra 1024 bytes is then simply loaded into memory. If that extra data is actually a malicious program, then it has just been loaded into memory and is thus now running on the target system. Or, perhaps the perpetrator simply wants to flood the target machine's memory, thus overwriting other items that are currently in memory and causing them to crash. Either way, the buffer overflow is a very serious attack.

Fortunately, buffer-overflow attacks are a bit harder to execute than a DoS or simple Microsoft Outlook script virus. To create a buffer-overflow attack, you must have a good working knowledge of some programming language (C or C++ is often chosen) and understand the target operating system/application well enough to know whether it has a buffer overflow weakness and how that weakness might be exploited.

It must be noted that modern operating systems and web servers are not generally susceptible to common buffer overflow attacks. Windows 95 was quite susceptible, but it has been many years since a Windows operating system was susceptible. Certainly Windows 7, 8, or 10 cannot be compromised with this type of buffer overflow. However, the same cannot be necessarily said for all the custom applications developed to run on various systems. It is always possible that an Internet-enabled application, including but not limited to web applications, might be susceptible to this attack.

Essentially, this vulnerability only exists if programmers fail to program correctly. If all programs truncate extra data, then a buffer overflow cannot be executed on that system. However, if the program does not check the boundaries of variables and arrays and allows excess data to be loaded, then that system is vulnerable to a buffer overflow.

The Sasser Virus/Buffer Overflow

This is an older attack but one that demonstrates the use of a buffer-overflow attack. Sasser is a combination attack in that the virus (or worm) spreads by exploiting a buffer overrun.

The Sasser virus spreads by exploiting a known flaw in a Windows system program. Sasser copies itself to the Windows directory as avserve.exe and creates a Registry key to load itself at startup. In that way, once your machine is infected, you will start the virus every time you start the machine. This virus scans random IP addresses, listening on successive TCP ports starting at 1068 for exploitable systems—that is, systems that have not been patched to fix this flaw. When one is found, the worm exploits the vulnerable system by overflowing a buffer in LSASS.EXE, which is a file that is part of the Windows operating system. That executable is a built-in system file and is part of Windows. Sasser also acts as an FTP server on TCP port 5554, and it creates a remote shell on TCP port 9996. Next, Sasser creates an FTP script named cmd.ftp on the remote host and executes it. This FTP script instructs the target victim to download and execute the worm from the infected host. The infected host accepts this FTP traffic on TCP port 5554. The computer also creates a file named win.log on the C: drive. This file contains the IP address of the localhost. Copies of the virus are created in the Windows System directory as #_up.exe. Examples are shown here:

- c:\WINDOWS\system32\12553_up.exe
- c:\WINDOWS\system32\17923_up.exe
- c:\WINDOWS\system32\29679_up.exe

A side effect of this virus is that it causes your machine to reboot. A machine that is repeatedly rebooting without any other known cause may well be infected with the Sasser virus.

This is another case in which the infection can easily be prevented by several means. First, if you update your systems on a regular basis, your systems should not be vulnerable to this flaw. Second, if your network's routers or firewall block traffic on the ports mentioned (9996 and 5554), you will then prevent most of Sasser's damage. Your firewall should only allow traffic on specified ports; all

other ports should be shut down. In short, if you as the network administrator are aware of security issues and are taking prudent steps to protect the network, your network will be safe. The fact that so many networks were affected by this virus should indicate that not enough administrators are properly trained in computer security.

Spyware

In Chapter 1, "Introduction to Computer Security," spyware was mentioned as one of the threats to computer security. Using spyware, however, requires a great deal more technical knowledge on the part of the perpetrator than some other forms of malware. The perpetrator must be able to develop spyware for the particular situation or customize existing spyware for his needs. He must then be able to get the spyware on the target machine.

Spyware can be as simple as a cookie used by a website to record a few brief facts about your visit to that website, or it could be of a more insidious type, such as a key logger. Recall from Chapter 1 that key loggers are programs that record every keystroke you make on your keyboard; this spyware then logs your keystrokes to the spy's file. The most common use of a key logger is to capture usernames and passwords. However, this method can capture every username and password you enter and every document you type, as well as anything else you might type. This data can be stored in a small file hidden on your machine for later extraction or sent out in TCP packets to some predetermined address. In some cases, the software is even set to wait until after hours to upload this data to some server or to use your own email software to send the data to an anonymous email address. There are also some key loggers that take periodic screenshots from your machine, revealing anything that is open on your computer. Whatever the specific mode of operation, spyware is software that literally spies on your activities on a particular computer.

Legal Uses of Spyware

There are some perfectly legal uses for spyware. Some employers have embraced such spyware as a means of monitoring employee use of company technology. Many companies have elected to monitor phone, email, or web traffic within the organization. Keep in mind that the computer, network, and phone systems are the property of the company or organization, not of the employee. These technologies are supposedly only used for work purposes; therefore, company monitoring might not constitute an invasion of privacy. While courts have upheld this monitoring as a company's right, it is critical to consult an attorney before initiating this level of employee monitoring as well as to consider the potential negative impact on employee morale.

Parents can also elect to use this type of software on their home computer to monitor the activities of their children on the Internet. The goal is usually a laudable application—protecting their children from online predators. Yet, as with employees in a company, the practice may illicit a strong negative reaction from the parties being spied upon (namely, their children). Parents have to weigh the risk to their children versus what might be viewed as a breach of trust.

How Is Spyware Delivered to a Target System?

Clearly, spyware programs can track all activity on a computer, and that information can be retrieved by another party via a number of different methods. The real question is this: How does spyware get onto a computer system in the first place? The most common method is a Trojan horse. It is also possible that when you visit a certain website spyware may download in the background while you are simply perusing the website. Of course, if an employer (or parent) is installing the spyware, it can then be installed noncovertly in the same way that an organization would install any other application.

Obtaining Spyware Software

Given the many other utilities and tools that have been mentioned as available from the Internet, you probably will not be surprised to learn that you can obtain many spyware products for free, or at very low cost, on the Internet. You can check the Counterexploitation (www.cexx.org) website, shown in Figure 5.2, for a lengthy list of known spyware products circulating on the Internet and for information about methods you can use to remove them. The SpywareGuide website (SpywareGuide, 2004) (www.spywareguide.com) lists spyware that you can get right off the Internet should you feel some compelling reason to spy on someone's computer activities. Figure 5.3 shows the categories of malware that are available from this site. Several key logger applications are listed on this site, as shown in Figure 5.4. These applications include well-known key loggers such as Absolute Keylogger, Tiny Keylogger, and TypO. Most can be downloaded for free or for a nominal charge from the Internet.

FIGURE 5.2 Counterexploitation website.

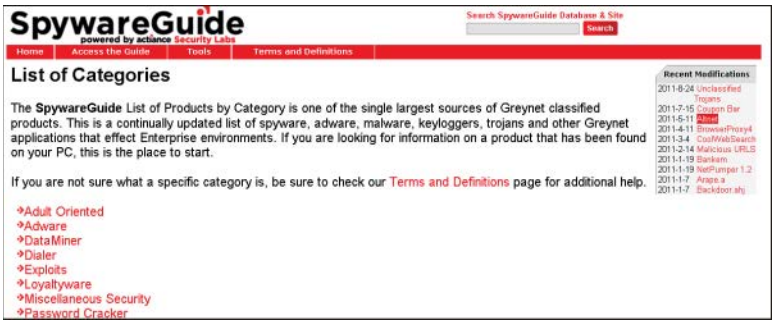

FIGURE 5.3 Malware categories at the SpywareGuide website.

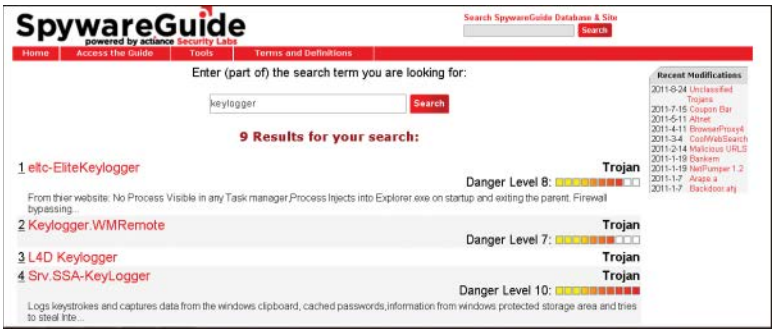

FIGURE 5.4 List of key loggers available through the SpywareGuide website.

Some well-known Trojan horses are also listed at this site (as shown in Figure 5.5), such as the 2nd Thought application that downloads to a person's PC and then blasts it with advertisements. This particular piece of spyware is one that downloads to your PC when you visit certain websites. It is benign in that it causes no direct harm to your system or files, nor does it gather sensitive information from your PC. However, it is incredibly annoying as it inundates your machine with unwanted ads. This sort of software is often referred to as adware. Frequently, these ads cannot be stopped by normal protective pop-up blockers because the pop-up windows are not generated by a website that you visit but rather by some rogue software running on your machine. Pop-up blockers only work to stop sites you visit from opening new windows. Websites use well-known scripting techniques to cause your browser to open a window, and pop-up blockers recognize these techniques and prevent the ad window from opening. However, if the adware launches a new browser instance, it bypasses the pop-up blocker's function.

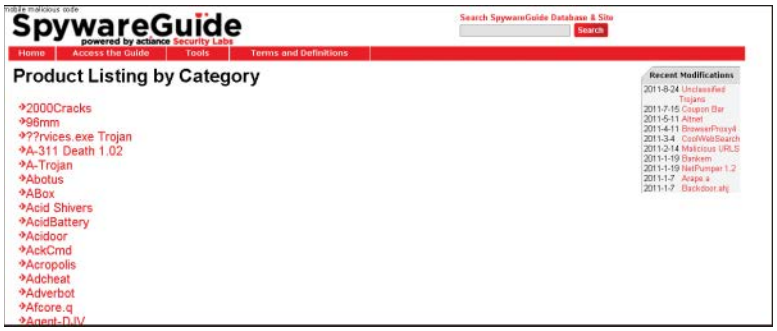

FIGURE 5.5 Trojan horses available at the SpywareGuide website.

Other Forms of Malware

In this and preceding chapters, the most prominent forms of malware have been discussed. There are, however, many other forms of attack. It is beyond the scope of this book to explore each of these, but you should be aware of the existence of these other forms of malware. Simply being aware can go a long way toward enabling you to defend your system efficiently. This section will touch upon just a few other forms of malware. You should reference the websites discussed in the end-of-chapter exercises and projects often so that you can stay up-to-date with all current forms of attack and defenses.

Rootkit

A *rootkit* is a collection of tools that a hacker uses to mask her intrusion and obtain administrator-level access to a computer or computer network. The intruder installs a rootkit on a computer after first obtaining user-level access, either by exploiting a known vulnerability or cracking a password. The rootkit then collects user IDs and passwords to other machines on the network, thus giving the hacker root or privileged access.

A rootkit may consist of utilities that also

- Monitor traffic and keystrokes
- Create a back-door into the system for the hacker's use
- Alter log files
- Attack other machines on the network
- Alter existing system tools to circumvent detection

The presence of a rootkit on a network was first documented in the early 1990s. At that time, Sun and Linux operating systems were the primary targets for a hacker looking to install a rootkit. Today, rootkits are available for a number of operating systems and are increasingly difficult to detect on any network (searchSecurity.com, 2004b).

Malicious Web-Based Code

A *malicious web-based code*, also known as a *web-based mobile code*, simply refers to a code that is portable to all operating systems or platforms such as HTTP, Java, and so on. The "malicious" part implies that is it a virus, worm, Trojan horse, or some other form of malware. Simply put, the malicious code does not care what the operating system may be or what browser is in use. It infects them all blindly (Yakabovicz, 2003).

Where do these codes come from, and how are they spread? The first generation of the Internet was mostly indexed text files. However, as the Internet has grown into a graphical, multimedia user experience, programmers have created scripting languages and new application technologies to enable a more interactive experience. As with any new technology, programs written with scripting languages run the gamut from useful to poorly crafted to outright dangerous.

Technologies such as Java and ActiveX enable these buggy or untrustworthy programs to move to and execute on user workstations. (Other technologies that can enable malicious code are executables, JavaScript, Visual Basic Script, and plug-ins.) The Web acts to increase the mobility of code without differentiating between program quality, integrity, or reliability. Using available tools, it is quite simple to "drag and drop" code into documents that are subsequently placed on web servers and made available to employees throughout the organization or individuals across the Internet. If this code is maliciously programmed or just improperly tested, it can cause serious damage.

Not surprisingly, hackers have used these very useful tools to steal, alter, and erase data files as well as gain unauthorized access to corporate networks. A malicious code attack can penetrate corporate networks and systems from a variety of access points, including websites, HTML content in email messages, or corporate intranets.

Today, with billions of Internet users, new malicious code attacks can spread almost instantly through corporations. The majority of damage caused by malicious code happens in the first hours after a first-strike attack occurs—before there is time for countermeasures. The costs of network downtime or theft of IP make malicious code a top priority (finjan software, 2004).

Logic Bombs

A *logic bomb* is a type of malware that executes its malicious purpose when a specific criteria is met. The most common factor is date/time. For example, a logic bomb might delete files on a certain date/time. An example is the case of Roger Duronio. In June 2006, Roger Duronio, a system administrator for UBS, was charged with using a logic bomb to damage the company's computer network. His plan was to drive the company stock down due to damage from the logic bomb, so he was charged with securities fraud. Duronio was later convicted and sentenced to 8 years and 1 month in prison and ordered to pay $3.1 million restitution to UBS.

Another good example of a logic bomb is the case of Michael Lauffenburger. In June 1992, Lauffenburger, who was an employee of defense contractor General Dynamics, was arrested for inserting a logic bomb into the company's systems. This logic bomb was designed to delete sensitive project data. Lauffenburger hoped the cause of the missing data would go unnoticed, and he could return as

a consultant to "fix" the problem. Fortunately, another employee of General Dynamics uncovered the logic bomb before it was triggered, and a thorough investigation ensued. Lauffenburger was charged with computer tampering and attempted fraud. While statute allowed for fines of up to $500,000 as well as incarceration, he was only fined $5,000 and given no jail time.

Another example occurred at the mortgage company Fannie Mae. On October 29, 2008, a logic bomb was discovered in the company's systems.[1] This logic bomb had been planted by a former contractor, Rajendrainh Makwana, who had been terminated. The bomb was set to activate on January 31, 2009, and completely wipe all of the company's servers. Makwana was indicted in a Maryland court on January 27, 2009 for unauthorized computer access. On December 17, 2010 he was convicted and sentenced to 41 months in prison, followed by 3 years of probation after release. What is most interesting about this case is that Makwana planted the logic bomb between the time he was terminated and the time the network administrators cancelled his network access. This illustrates the importance of ensuring that the accounts of former employees are deactivated immediately when their employment is terminated. That applies whether it is an involuntary termination, retirement, or voluntary quit.

Spam

Spam is something most readers are probably familiar with. *Spam* is unwanted and unsolicited email that is sent out to multiple parties. Often it is used for marketing purposes, but it can be used for much more malicious goals. For example, spam is a common way to spread a virus or worm. Spam is also used to send emails enticing recipients to visit phishing websites in order to steal the recipient's identity. Essentially, spam is, at best, an annoyance and, at worst, a vehicle for spyware, viruses, worms, and phishing attacks.

Advanced Persistent Threats

Advanced persistent threats, often abbreviated APTs, is a relatively new term for a continuous process of attacking. It can involve hacking, social engineering, malware, or combinations of attacks. The issue is that the attack must be relatively sophisticated, thus the term *advanced*, and it must be ongoing, thus the term *persistent*.

The security firm Mandiant tracked several APTs over a period of 7 years, all originating in China—specifically, Shanghai and the Pudong region. These APTs were simply named APT1, APT2, and so on.

The attacks were linked to the UNIT 61398 of China's Military. The Chinese government regards this unit's activities as classified, but it appears that offensive cyber warfare is one of its tasks. Just one of the APTs from this group compromised 141 companies in 20 different industries. APT1 was able to maintain access to victim networks for an average of 365 days, and in one case for 1,764 days. APT1 is responsible for stealing 6.5 terabytes of information from a single organization over a 10-month time frame. We will discuss the Chinese attack in more detail in Chapter 12 as part of our discussion of cyber terrorism and information warfare.

1. https://www.fbi.gov/baltimore/press-releases/2010/ba121710.htm

Detecting and Eliminating Viruses and Spyware

Once you understand the nature of malware, and just how devastating it can be, the next logical topic is how to detect and remove malware.

Antivirus Software

In this chapter and throughout this book, the need for running virus-scanning software has been discussed. It is prudent at this point to provide you with some details on how virus scanners work and information on the major virus-scanning software packages. This information should help you better understand how a virus scanner might protect your system and help you make intelligent decisions regarding the purchase and deployment of some antivirus solution.

A virus scanner can work in one of two ways. The first is to look for a signature (or pattern) that matches a known virus. This is why it is important to keep your virus software updated so that you have the most recent list of signatures with which to work.

The other way in which a virus scanner might check a given PC is to look at the behavior of an executable. If that program behaves in a way consistent with virus activity, the virus scanner may flag it as a virus. Such activity could include the following:

- Attempting to copy itself
- Attempting to access the address book of the system's email program
- Attempting to change Registry settings in Windows

Figure 5.6 shows the Norton AntiVirus software in action. You can see that the virus definitions are up-to-date, virus scanning is enabled, auto-protection is enabled, and the Internet worm protection is enabled as well. The other popular virus scanners have many of the same features.

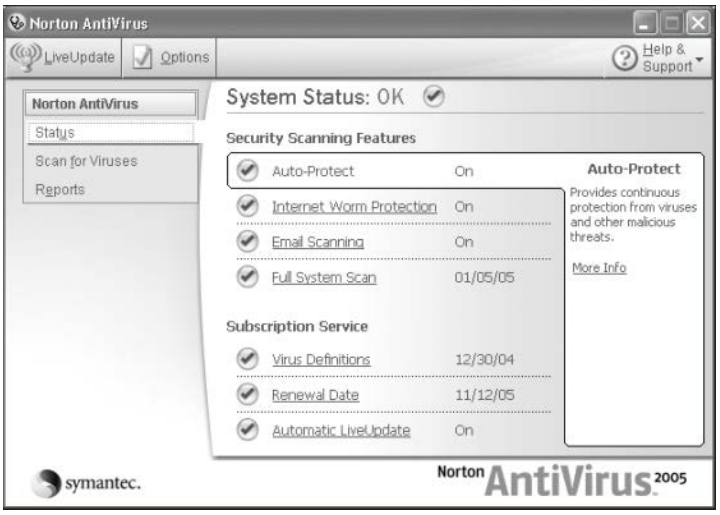

FIGURE 5.6 Norton AntiVirus interface.

Most antivirus software today offer additional features. Some of these features include warning the user of known phishing websites, detecting spyware as well as viruses, and even detecting likely phishing attempts. Any modern antivirus should be a comprehensive package, protecting against a variety of attacks, rather than just stopping viruses.

Antispyware Software

Fortunately, just as there are many different spyware applications available, there are likewise many different software applications on the market that are designed specifically to detect and remove spyware. These applications are also usually available at extremely low cost. You can often get a free trial version to use for a limited time so that you can make a more intelligent purchasing decision. Of course, the most prudent course of action you can take to avoid getting spyware on your machine is to never download anything from the Internet that does not come from a very well-known and trusted website. However, in an organizational environment, you cannot simply rely on your employees to do the right thing. It is prudent as the company's computer security expert to take steps yourself to prevent the employees from compromising your system security.

One of the better known and more widely used antispyware applications includes Spector Pro from www.spectorsoft.com. Many of these types of applications can be obtained for anywhere from $20 to $50, and many offer a free trial version. Most modern antivirus software either includes antispyware, or it can be added as an option.

Remediation Steps

Obviously, the first step is running up-to-date antivirus software. I don't endorse any particular antivirus program. McAfee, Norton, AVG, Kaspersky, and MalwareBytes are all reputable products. I do, however, recommend that if you are using both host-based antivirus and network antivirus you use products from two different vendors. Then what one misses, the other is likely to catch.

In addition to running antimalware software, there are specific steps security professionals can take to mitigate the risk from malware. Notice the goal is to mitigate the threat of malware. It is impossible to completely eliminate such threats, but you can reduce both the frequency of attacks and the severity of damage.

The first step is to educate end users. End users must be made aware of the prevalence of malware. The goal is for all users on your network to be suspicious of attachments and of downloads. Certainly, there are business needs that require the use of attached documents. But you can educate users on a few simple questions to ask themselves before opening any attachment:

- Was this attachment expected? Attachments from people you do not know or from whom you did not expect any attachment must always be treated as potential malware.
- Is the email specific, such as "These are the third-quarter sales reports we discussed in our meeting yesterday"? That indicates this is likely to be a real email with a real document attached. But generic-sounding emails such as, "Here is your document" or emails that try

to convince the user that he must urgently open the attachment should be suspected of being malware.

- When in doubt, ask technical support. If you have significant doubts about the authenticity of an email attachment, don't open it. Ask technical support.

These simple steps would eliminate many malware outbreaks. There is another, more complicated step, but it would render your computers virtually immune to malware. First set up a virtual machine on your computer. There are a variety of virtual machine applications, such as VMware. Oracle Box is free. Then install into that virtual machine (VM) some operating system other than what the host computer runs. So if your host is Windows 10, then the VM should run Linux. If your host is Macintosh, then run Windows in the VM. Now if you surf the Internet inside the VM, it is almost impossible for you to get a virus on the host machine. To date, no virus has jumped the VM/host barrier, and no virus infects multiple operating systems. It should be noted that Java viruses or certain web page malware can infect browsers on different systems. However, as a general rule, a virus written for Windows won't affect a Macintosh computer and vice versa.

Summary

Clearly, there are a number of ways to attack a target system: by denial of service, virus/worm, Trojan horse, buffer-overflow attacks, and spyware. Each type of attack comes in many distinct variations. It should be obvious by this point that securing your system is absolutely critical. In the upcoming exercises, you will try out the antivirus programs by Norton and McAfee. There are so many ways for a hacker to attack a system that securing your system can be a rather complex task.

Another theme that is driven home throughout this chapter is that many, if not most, attacks are preventable. The exercises ahead will give you practice in figuring out how to prevent the Sasser and Sobig viruses. In most cases, prompt and regular patching of the system, use of antivirus tools, and blocking unneeded ports would prevent the attack. The fact that so many systems do get infected is an indication of the very real problem of network professionals who are not skilled in computer security.

Test Your Skills

MULTIPLE CHOICE QUESTIONS

1. Which of the following is the best definition of virus?
   - **A.** Program that causes harm on your computer
   - **B.** Program used in a DoS attack
   - **C.** Program that slows down networks
   - **D.** Program that self-replicates
2. What is the most common damage caused by virus attacks?
   - **A.** Slowing down networks by the virus traffic
   - **B.** Deleting files
   - **C.** Changing the Windows Registry
   - **D.** Corrupting the operating system
3. What is the most common way for a virus to spread?
   - **A.** By copying to shared folders
   - **B.** By email attachment
   - **C.** By FTP
   - **D.** By downloading from a website

4. Which of the following is the primary reason that Microsoft Outlook is so often a target for virus attacks?

   A. Many hackers dislike Microsoft.
   B. Outlook copies virus files faster.
   C. It is easy to write programs that access Outlook's inner mechanisms.
   D. Outlook is more common than other email systems.

5. Which of the following virus attacks used a multimodal approach?

   A. Slammer virus
   B. Mimail virus
   C. Sobig virus
   D. Bagle virus

6. What factor about the Sobig virus made it most intriguing to security experts?

   A. It spread in multiple ways.
   B. It deleted critical system files.
   C. It was difficult to protect against.
   D. It was very sophisticated.

7. What was most interesting to security experts about the Mimail virus?

   A. It spread more rapidly than other virus attacks.
   B. It spread in multiple ways.
   C. It grabbed email addresses from documents on the hard drive.
   D. It deleted critical system files.

8. Which of the following reasons most likely made the Bagle virus spread so rapidly?

   A. The email containing it claimed to be from the system administrator.
   B. It copied itself across the network.
   C. It was a sophisticated virus.
   D. It was particularly virulent.

9. What made the Bagle virus so dangerous?

   A. It changed Windows Registry settings.
   B. It disabled antivirus software.
   C. It deleted key system files.
   D. It corrupted the operating system.

10. Which of the following is a way that any person can use to protect against virus attacks?

- **A.** Set up a firewall.
- **B.** Use encrypted transmissions.
- **C.** Use secure email software.
- **D.** Never open unknown email attachments.

11. Which of the following is the safest way to send and receive attachments?

- **A.** Use a code word indicating the attachment is legitimate.
- **B.** Only send spreadsheet attachments.
- **C.** Use encryption.
- **D.** Use virus scanners before opening attachments.

12. Which of the following is true regarding emailed security alerts?

- **A.** You must follow them.
- **B.** Most companies do not send alerts via email.
- **C.** You can trust attachments on security alerts.
- **D.** Most companies send alerts via email.

13. Which of the following is something a Trojan horse might do?

- **A.** Open a backdoor for malicious software.
- **B.** Change your memory configuration.
- **C.** Change ports on your computer.
- **D.** Alter your IP address.

14. What is a buffer-overflow attack?

- **A.** Overflowing a port with too many packets
- **B.** Putting more email in an email system than it can hold
- **C.** Overflowing the system
- **D.** Putting more data in a buffer than it can hold

15. What virus exploited buffer overflows?

- **A.** Sobig virus
- **B.** Mimail virus
- **C.** Sasser virus
- **D.** Bagle virus

16. What can you do with a firewall to help protect against virus attacks?

    A. There is nothing you can do on the firewall to stop virus attacks.
    B. Shut down all unneeded ports.
    C. Close all incoming ports.
    D. None of the above.

17. A key logger is what type of malware?

    A. Virus
    B. Buffer overflow
    C. Trojan horse
    D. Spyware

18. Which of the following is a step that all computer users should take to protect against virus attacks?

    A. Purchase and configure a firewall.
    B. Shut down all incoming ports.
    C. Use nonstandard email clients.
    D. Install and use antivirus software.

19. What is the primary way a virus scanner works?

    A. By comparing files against a list of known virus profiles
    B. By blocking files that copy themselves
    C. By blocking all unknown files
    D. By looking at files for virus-like behavior

20. What other way can a virus scanner work?

    A. By comparing files against a list of known virus profiles
    B. By blocking files that copy themselves
    C. By blocking all unknown files
    D. By looking at files for virus-like behavior

EXERCISES

EXERCISE 5.1: Using Norton Antivirus

1. Go to the Norton AntiVirus website (www.symantec.com/downloads) and download the trial version of its software.
2. Install and run its software.
3. Carefully study the application, noting features that you like and dislike.

EXERCISE 5.2: Using McAfee Antivirus

1. Go to the McAfee antivirus website (http://us.mcafee.com/root/package.asp?pkgid=100&cid=9901) and download the trial version of its software.
2. Install and run its software.
3. Carefully study the application, noting features you like and dislike.

EXERCISE 5.3: Preventing Sasser

1. Using resources on the Web or in journals, carefully research the Sasser virus. You may find that www.f-secure.com and Symantec's Security Response center at https://www.symantec.com/security_response/ helpful in this exercise.
2. Write a brief essay about how Sasser spread, what damage it caused, and what steps could be taken to prevent it.

EXERCISE 5.4: Preventing Sobig

1. Using resources on the Web or in journals, carefully research the Sobig virus. You may find that www.f-secure.com and Symantec's virus information center at www.sarc.com/avcenter/ are helpful in this exercise.
2. Write a brief essay about how Sobig spread, what damage it caused, and what steps could be taken to prevent it.

EXERCISE 5.5: Learning About Current Virus Attacks

1. Using resources on the Web or in journals, find a virus that has been spreading in the last 90 days. You may find that www.f-secure.com and Symantec's virus information center at www.sarc.com/avcenter/ are helpful in this exercise.
2. Write a brief essay about how the virus spread, what damage it caused, and what steps could be taken to prevent it.

PROJECTS

PROJECT 5.1: Antivirus Policies

This activity can also work as a group project.

Considering what you have learned in this chapter and in previous chapters, as well as using outside resources, write an antivirus policy for a small business or school. Your policy should include technical recommendations as well as procedural guidelines. You may choose to consult existing antivirus policy guidelines that you find on the Web to give you some ideas.

However, you should not simply copy their antivirus policies. Rather, you should come up with your own.

PROJECT 5.2: The Worst Virus Attacks

Using resources on the Web, books, or journals, find a virus outbreak that you consider to have been the worst in history. Write a brief paper describing this attack, and explain why you think it is the worst. Was it widely spread? How quickly did it spread? What damage did it do?

PROJECT 5.3: Why Write a Virus?

A number of hypotheses have been formed regarding why people write a virus. These hypotheses range from the frankly conspiratorial to the academically psychological. Taking whatever position you feel is most likely, write a paper explaining why you think people take the time and effort to write a virus.

Case Study

Chiao Chien manages IT security for a school. Given the wide range of people who use the school's computers, it is difficult for Chien to prevent virus attacks. Chien has a reasonably good budget and has installed antivirus software on every machine. He also has a firewall that has all unneeded ports blocked, and there is a school policy prohibiting the downloading of any software from the Web. Consider the following questions:

- How secure do you think Chien's network is from virus attacks?
- What areas has Chien not secured?
- What recommendations would you make to Chien?

Chapter 6

Techniques Used by Hackers

Chapter Objectives

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

- Understand the basic methodology used by hackers
- Be familiar with some of the basic hacking tools
- Understand the hacking mentality

Introduction

In the preceding five chapters, we have explored computer security and various security breaches. In this chapter we will be exploring the techniques that hackers use to commit computer crimes. Before we go any further, it is important that you realize that many hackers are not criminals. A hacker is a person who wants to understand a system, often by probing its weaknesses. There are even hackers that work for organizations, testing the organization's system security. This is called *penetration testing*. This is also often referred to as *white hat hacking*. There are several certifications for penetration testing:

> Offensive Security has several penetration testing certifications: https://www.offensive-security.com/information-security-certifications/
>
> SANS Institute has a penetration testing certification: http://pen-testing.sans.org/certification
>
> The EC Council (www.eccouncil.org) has a certification for this: the Certified Ethical Hacker.
>
> And there is the Professional Penetration Tester certification: www.professionalpentester.com/

There is also a magazine for such people called *2600* (www.2600.com). Many computer security professionals attempt to learn hacking techniques either to enhance their security capabilities or to

simply satisfy their curiosity. The techniques themselves are not criminal. However, there are people who use hacking techniques to breach systems to steal data, damage systems, or commit other cyber crimes. These people are usually referred to as *black hat hackers* or *crackers*.

The techniques presented in this chapter are not only presented to give the reader an understanding of how black hat hackers work, but also provide a method whereby a network administrator can perform a penetration test on his own network. By attempting some of these techniques on your network, you can assess your vulnerability. It should be pointed out that you should only do this once you are very comfortable with the techniques in this chapter, and only with permission from senior management.

Basic Terminology

Before we can delve into the world of hacking, we need to discuss the basic terminology used in this community. We have already introduced you to the term *white hat hacker*, one who uses hacking techniques for legal/ethical purposes. And we have discussed the term *black hat hacker* or *cracker*, one who uses hacking techniques for illegal techniques.

There are a few other terms you should be familiar with. A *gray hat hacker* is one who was previously a black hat hacker and turned into a white hat hacker (basically, a former criminal now turned ethical). With the proliferation of tools on the Internet, there are also a lot of people who download some tools (we will examine some of these in this chapter) and perform some cyber attack, without really understanding it. These people are termed *script kiddies* (also sometimes spelled kiddy's). Another important term is *phreaking*. This refers to hacking into phone systems. Hacking of phone systems actually predates hacking computer systems.

The Reconnaissance Phase

Any intelligent/experienced hacker is going to attempt to find out information about a target before actually attempting an attack. Just as a bank robber would want to know about a bank's alarm systems, number of guards, police response time, and so on, a black hat hacker will want to know about your system's security. What may surprise you is how much information can be found easily on the Internet, without even attaching to the target system.

Passive Scanning Techniques

One of the easiest things a hacker can do is check the target organization's websites. It is common for businesses to put information up that can be very useful to an attacker. For example, let's assume company XYZ lists John Doe as its IT manager. An enterprising hacker scans bulletin boards and discussion groups for references to John Doe at XYZ. That attacker might find information useful in spear phishing attacks (phishing targeted at a specific individual or group of individuals), or the attacker might find information useful in social engineering. For example, a number of former employees complain that John Doe is demanding and quick to fire people. Then an enterprising hacker could call someone at

XYZ claiming to be working for John Doe. The hacker claims he is trying to log on remotely to that person's computer to update that person's system. After a few moments, the hacker tells the person he forgot the password John Doe gave him and is very concerned he will get fired if he doesn't complete this assignment; then he asks that person for his password. The information the attacker gleaned from the Web gave him enough information to make this social engineering attack plausible.

It is also possible for an attacker to scan bulletin boards, chat rooms, discussion groups, and more looking for questions from IT staff at the target organization. For example, if an administrator posts in a discussion group asking about a particular server problem, this can give the attacker valuable information about that target network.

Another way attackers can use the Web to find out information about a target is through job ads. For example, if a company routinely advertises for ASP.NET developers and never for PHP or Perl, then it is likely that the company's web applications are developed with ASP.NET running on a Windows web server (Internet Information Services). This can allow the attacker to focus only on a small group of possible attacks (those against ASP.NET/Windows). Other information can be garnered from job ads. For example, if a small company, less than 200 employees, has an advertisement for a network administrator twice a year, it is more likely that the last administrator is no longer with them. A small company would not need multiple administrators. Knowing the current administrator is new means she is not as familiar with her own systems. Also, if this trend of advertising for new administrators extends over a couple of years, it means the company has high turnover and there is some problem the attacker may be able to exploit.

There are also specific websites that provide information an attacker may find useful. The first such website is netcraft.com, shown in Figure 6.1. This website provides information about websites. For example, you can find out what kind of server a site is running, and in some cases how long it has been since it was last rebooted.

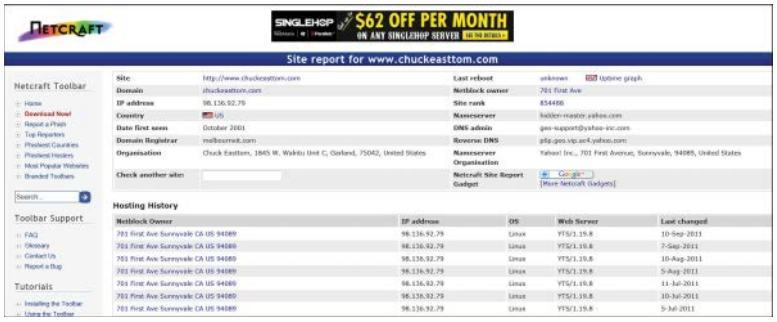

FIGURE 6.1 www.netcraft.com.

Another site that can be useful for attackers is https://archive.org. This site, shown in Figure 6.2, archives older versions of websites. The server scours the Web archiving sites. The frequency with which a site is archived depends on its popularity.

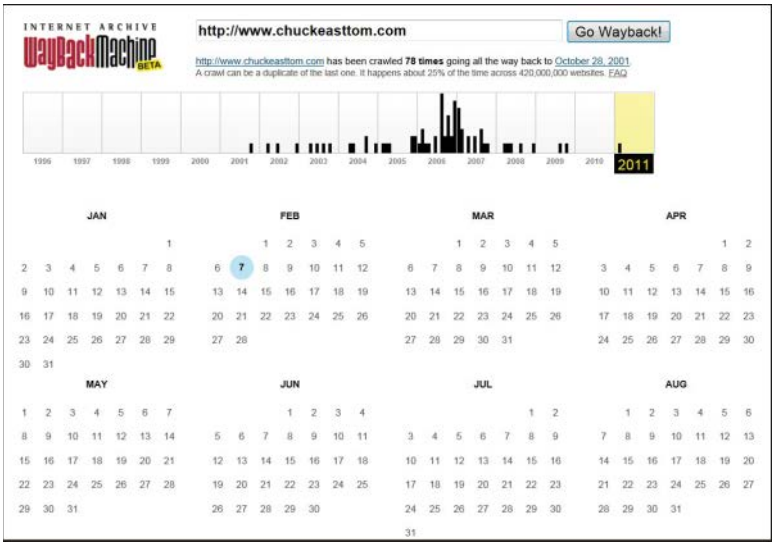

FIGURE 6.2 www.archive.org.

Active Scanning Techniques

The previously mentioned techniques are all considered passive, as they do not require the attacker to connect to the target system. Since the attacker is not actually connecting to the target system, it is impossible for an intrusion detection system (IDS) to detect the scan. Active scans are far more reliable but may be detected by the target system. There are a few types of active scans.

Port Scanning

Port scanning is the process of attempting to contact each network port on the target system and see which ones are open. There are 1,024 well-known ports that are usually associated with specific services. For example, port 161 is associated with Simple Network Management Protocol. If an attacker detects port 161 open on the target system, he might decide to try SNMP-related attacks. Even more information can be derived from a port scan. For example, ports 137, 138, and 139 are all associated with NetBIOS, a very old Windows method of network communication, not used in Windows anymore. However, NetBIOS is often used for systems where Windows machines need to communicate with Linux machines. So discovering those ports open reveals something about the target network.

A simple Google search for *port scanner* will reveal a host of well-known, widely used, and often free port scanners. However, the most popular port scanner in the hacking and security community is the free tool Nmap (https://nmap.org/). There is a Windows version of it, shown in Figure 6.3.

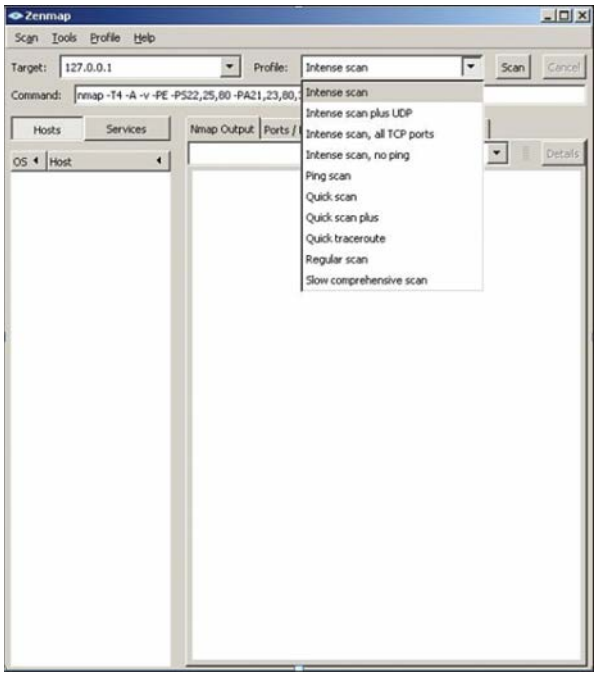

FIGURE 6.3 Nmap GUI

Nmap allows you to customize your scan making it more or less stealthy, and targeting certain systems. The most common types of scans are listed here:

- **Ping scan:** This scan simply sends a ping to the target port. Many network administrators block incoming ICMP packets for the purpose of stopping ping scans.
- **Connect scan:** This is the most reliable scan, but also the most likely to be detected. With this type of scan a complete connection is made with the target system.
- **SYN scan:** This scan is very stealthy. Most systems accept SYN (Synchronize) requests. This scan is similar to the SYN flood DoS attack described in Chapter 4, "Denial of Service Attacks." In this scan you send a SYN packet but never respond when the system sends a SYN/ACK. However, unlike the DoS SYN flood, you only send one packet per port. This is also called the *half-open scan*.
- **FIN scan:** This scan has the FIN flag, or connection finished flag set. This is also not an unusual packet for systems to receive, so it is considered stealthy.

Each of these scans provokes a different response on the target machine and thus provides different information to the port scanner:

- With a FIN scan or an XMAS scan, if the target port is closed, the system sends back an RST flag packet (RST means reset). If it is open, there is no response.

- With a SYN scan, if the port is closed, the response is an RST; if it is open, the response is a SYN/ACK.
- ACK scans and NULL scans only work on UNIX systems.

Nmap also lets you set a number of flags (either with the command-line version of Nmap or the Windows version) that customize your scan. The allowed flags are listed here:

-O	Detects operating system
-sP	Is a ping scan
-sT	TCP connect scan
-sS	SYN scan
-sF	FIN scan
-sX	Xmas tree scan
-sN	NULL scan
-sU	UDP scan
-sO	Protocol scan
-sA	ACK scan
-sW	Windows scan
-sR	RPC scan
-sL	List/DNS scan
-sI	Idle scan
-Po	Don't ping
-PT	TCP ping
-PS	SYN ping
-PI	ICMP ping
-PB	TCP and ICMP ping
-PM	ICMP netmask
-oN	Normal output
-oX	XML output
-oG	Greppable output
-oA	All output
-T	Timing

-T 0 Paranoid

-T 1 Sneaking

-T 2 Polite

-T 3 Normal

-T 4 Aggressive

-T 5 Insane

As you can see there are a number of options available to an attacker using Nmap. One can spend a lot of time just learning Nmap. There are, of course, a number of other port scanning tools. We have focused on Nmap because it is free and it is so widely used. It also is prominently figured on the EC Council Certified Ethical Hacker certification, GPEN (from SANS), and the Professional Penetration Tester Certification.

The settings are, for the most part, self-explanatory. Perhaps the timing warrants a bit more discussion. Timing involves how quickly to send scanning packets. Essentially, the faster you send packets, the more likely the scan is to be detected.

Here is the most basic Nmap scan:

```
nmap 192.168.1.1
```

Scan a range of IP addresses:

```
nmap 192.168.1.1-20
```

Scan to detect operating system, use TCP scan, and use sneaky speed:

```
nmap -O -PT -T1 192.168.1.1
```

Vulnerability Assessment

Vulnerability assessment is checking a system to see if it is vulnerable to specific attacks. These tools can be used by attackers to assess your system; however, they are designed to allow you to assess your system. These tools are not particularly stealthy and thus will probably be detected by an intrusion detection system. In fact, vulnerability assessment tools are commonly used by network administrators to test their own networks. These tools will be covered in Chapter 11, "Network Scanning and Vulnerability Scanning."

Enumeration

Another technique that is popular before the actual attack is enumeration. Enumeration is simply the process of finding out what is on the target system. If the target is an entire network, then the attacker is trying to find out what servers, computers, and printers are on that network. If the target is a specific computer, then the attacker is trying to find out what users and shared folders exist on that system.

A simple Google search will help you find a number of enumeration tools. One of the easiest to use is Cain and Abel, shown in Figure 6.4.

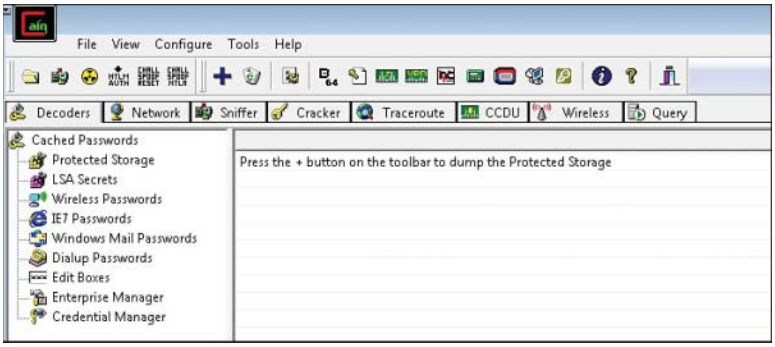

FIGURE 6.4 Cain and Abel.

Cain and Abel can do a lot more than just enumeration, but for our purposes that is what we are focusing on here. For enumeration, simply click on the Network tab and you will find all machines connected to the network you are on. This obviously requires some level of access before you can enumerate the target network.

A few other enumeration tools that are popular with hackers and can easily be found on the Internet are

- Sid2User
- Cheops (Linux only)
- UserInfo
- UserDump
- DumpSec
- Netcat
- NBTDump

This is not an exhaustive list, but it includes some of the most widely used enumeration tools.

To defend against scanning, you should use the following techniques:

- Be careful how much information you put on the Internet about your organization and its network.
- Make it a company policy that technical personnel who use bulletin boards, chat rooms, and so on for technical data must not use their real name or reveal the company's name.

- Use an IDS that detects many scans.
- Block incoming Internet Control Message Protocol (ICMP) packets.

These won't make scanning and reconnaissance impossible on your system, but they will make certain the attacker gathers significantly less information.

Actual Attacks

Now that we have discussed how attackers scan a target system, let's look at a few attacks that are commonly used. Obviously this won't be an exhaustive list, but it will provide you some insight into the attack methodologies used. In Chapter 4 we discussed denial of service (DoS) attacks and some tools used to cause these attacks. In this section we will look at other sorts of attacks and the techniques and tools used to make them happen.

SQL Script Injection

This may be the most popular attack on websites. In recent years, more websites have taken steps to ameliorate the dangers of this attack; unfortunately, all too many websites are susceptible. This attack is based on passing structured query language commands to a web application and getting the website to execute them.

Before we can discuss SQL injection, we must talk about SQL and relational databases. Relational databases are based on relations between various tables. The structure includes tables, primary and foreign keys, and relations. A basic description can be summarized with the following points:

- Each row represents a single entity.
- Each column represents a single attribute.
- Each record is identified by a unique number called a *primary key*.
- Tables are related by foreign keys. A *foreign key* is a primary key in another table.

You can see these relations in Figure 6.5.

All relational databases use Structured Query Language (SQL). SQL uses commands such as `SELECT`, `UPDATE`, `DELETE`, `INSERT`, `WHERE`, and others. At least the basic queries are very easy to understand and interpret.

The way the most basic SQL injection works is this. Many websites/applications have a page where users enter their username and password. That username and password will have to be checked against some database to see if they are valid. Regardless of the type of database (Oracle, SQL Server, MySQL), all databases speak SQL. SQL looks and functions a great deal like English. For example, to check a username and password, you might want to query the database and see if there is any entry in the users table that matches that username and password that was entered. If there is, then you have a match. The SQL statement might look something like this:

```
'SELECT * FROM tblUsers WHERE USERNAME = 'jdoe' AND PASSWORD = 'letmein'
```

PK	LNAME	FNAME	JobCode	Hire Date
1	Smith	Jane	2	1/10/2010
2	Perez	Juan	2	1/14/2011
3	Kent	Clark	1	3/2/2005
4	Euler	Leonard	3	3/5/2009
5	Plank	Max	3	4/2/2012

PK	Job Name	Min Edu.	Min Salary	Max Salary
1	Super Hero	None	100,000	1,000,000
2	Programmer	BA/BS	70,000	95,000
3	Math/Scientist	Ph.D.	80,000	110,000
4	Manager	BA/BS	140,000	220,000

FIGURE 6.5 Relations.

The problem with this, while it is valid SQL, is that we have hard coded the username and password. For a real website, we would have to take whatever the user entered into the username field and password field and check that. This can be easily done (regardless of what programming or scripting language the website is programmed in). It would look something like this:

```
'SELECT * FROM tblUsers WHERE USERNAME = '" + txtUsername.Text +' AND PASSWORD = '" +
txtPassword.Text +"'" .
```

If you enter username `'jdoe'` and password `'letmein'`, this code produces the following SQL command:

```
SELECT * FROM tblUsers WHERE USERNAME = 'jdoe' AND PASSWORD = 'letmein'
```

Now if there is a username jdoe in tblUsers, and the password for it is letmein, then this user will be logged on. If not, then an error will occur.

SQL injection works by putting some SQL into the username and password block that is always true. For example, suppose you enter `'OR X=X'` into the username and password boxes. This will cause the program to create this query:

```
SELECT * FROM tblUsers WHERE USERNAME = ''OR X=X' AND PASSWORD = ''OR X=X'
```

Notice that we start with a single quotation mark (`'`) before the `OR X=X`. This is to close the open quote the attacker knows must be in the code. And if you see `''`, that essentially is a blank or null. So what we are telling the database is to log us in if the username is blank, or if `X=X`, and if the password is blank, or if `X=X`. If you think about this for a second, you will see that X always equals x, so this will always be true.

There is no significance to `'OR X=X'`; it is simply a statement that will always be true. Attackers try other similar statements, such as the following:

' or `'a' ='a`

' or `'1' ='1`

' or (`1=1`)

This is only one example of SQL injection; there are other methods, but this is the most common. The defense against this attack is to filter all user input before processing it. This is often referred to as *input validation*. This prevents an attacker from entering SQL commands rather than a username and password. Unfortunately, many sites do not filter user input and are still vulnerable to this attack.

The example given here is the most basic version of SQL injection. You can do far more with SQL injection. The attacker is limited only by her knowledge of SQL and the target database system.

Remember that earlier in the text when we first, briefly, mentioned SQL injection, it was suggested that filtering input would prevent this. For example, the programmer creating the website should write the code to first check for any common SQL injection symbols such as the single quote (‘), percent sign (%), equal sign (=), or ampersand (&), and if those are found, stop processing and log an error. This would prevent many SQL injection attacks. There are methods to circumvent these security measures, but implementing them would stop many SQL injection attacks.

Cross-Site Scripting

With cross-site scripting, an attacker injects client-side script into web pages viewed by other users. The key is that the attacker enters scripts into an area that other users interact with. When they go to that part of the site, the attacker’s script is executed rather than the intended website functionality. For example, assume a shopping site allows users to review products. Rather than typing in a review, the attacker types in JavaScript that redirects the user to a phishing website. When another user views that “review,” the script will execute and take him to the new site. Again, this can be prevented by simply filtering all user input. As of this writing, all the major online shopping portals, such as Amazon.com, do filter input and are not susceptible to this attack. However, many smaller sites are still susceptible.

This attack, as well as SQL injection, illustrates why it is critical that all IT personnel be familiar with security, not just security administrators. If more web developers were more familiar with security, these two attacks would not be widespread.

Password Cracking

Doing password cracking is easiest when one can actually get physical access to a machine. This is not as difficult as it sounds. Many organizations (such as universities) have kiosk machines where someone can use the system with minimal/guest privileges. A skilled hacker can use this access to gain further access.

OphCrack

A very popular tool for cracking Windows passwords is OphCrack. OphCrack can be downloaded from http://ophcrack.sourceforge.net. It is based on an understanding of how Windows passwords work. Windows passwords are stored in a hash file in one of the system directories, usually C:\WINDOWS\system32\config\ in a SAM file. SAM is an acronym for Security Accounts Manager. The passwords are stored as a hash. (Hashes will be discussed in detail in Chapter 8, "Encryption.") What Windows does is hash the password you type in and compare it to the hash found in the SAM file. If there is a match, then you are logged in. To prevent someone from copying the SAM file and taking it off to try to brute force it, as soon as Windows begins the boot process, the operating system locks the SAM file. What OphCrack does is boot to Linux and then get the SAM file and look up the hashed passwords in a large table of hashed values it has, searching for a match. If it finds one, then the matching text in that table of hashed values is the password. You can see OphCrack in Figure 6.6.

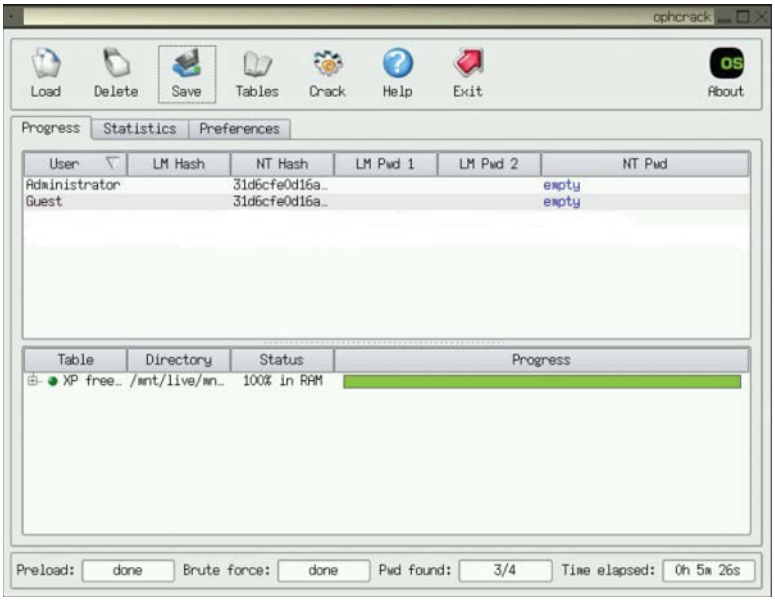

FIGURE 6.6 OphCrack.

This tool is remarkably easy to use. Just put the OphCrack CD into the machine and reboot. During the boot process you can press F12 for a boot menu and tell the system to boot from CD. You will then start OphCrack. It should be noted that longer passwords (as of this writing, longer than 10 characters) are usually not crackable by OphCrack.

Assuming OphCrack is successful (it isn't always), what can the attacker do with this? At best she simply got the local machine admin account, and not a domain account. Well, this can be used to then

gain domain access. One simple technique is to create a script that will in turn create a domain admin account. The script is simple:

```
net user /domain /add localaccountname password
net group /domain "Domain Admins" /add localaccount
```

Obviously, if the attacker executes this script it will not work. One must be a domain admin for it to work. So the attacker saves this script to the All Users startup folder. The next time a domain admin logs on to this system, the script will successfully execute. But the attacker may not want to wait until that happens. In order to speed up the process, the attacker causes some minor problem in the system (changes settings, alters configuration, and so on). In many organizations, the tech support personnel are in the domain admins group. When a tech support person logs on to the system to correct the problem, the script will successfully run.

Malware Creation

In this section we will briefly discuss how easy it is to create malware. Previously in Chapter 5, "Malware," you saw the tool eLiTeWrap. In this section you will see the methods used to actually create viruses. This is not in any way an encouragement for you to create such viruses. It is meant to educate you on why such malware is so common.

For many years, one needed significant programming skills in order to create a virus. However, in recent years there have been a number of tools developed that create viruses. These tools allow the end user to click a few buttons and create a virus. This is one reason viruses are becoming so prevalent. One such tool is the TeraBIT Virus Maker, shown in Figure 6.7.

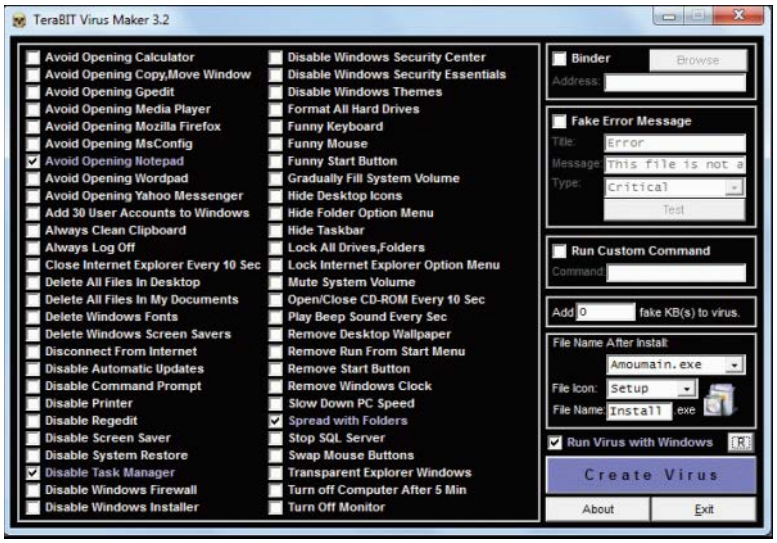

FIGURE 6.7 TeraBIT Virus Maker.

Tools like this make it very easy for even a novice to create a virus. When tools like this become prevalent, tools that automate some specific computer attack, then one can expect a great many more such attacks.

You can easily see from the options that TeraBIT Virus Maker can create some rather damaging malware. It is important to realize that this is only one option that a malware creator has. There are a number of tools on the Internet that help create viruses. There are even ransomware development kits.

In addition to these tools, there are websites that contain catalogs of malware code. Anyone with only moderate programming skills can download the code for a virus and modify that malware for his specific needs. You can think of this as a sort of cyber weapons proliferation.

This proliferation of cyber weapons is the primary reason for this section in this chapter. It is critical that security professionals, or aspiring security professionals, be aware of just how easy it is to create a virus. This means we should reasonably expect to see more viruses as time goes on. Of course, there are still custom written viruses, and these are in fact the most effective form of malware. But the proliferation of tools and source code means that even those with only minimal technical skills can create viruses.

Windows Hacking Techniques

Given the ubiquitous nature of Microsoft Windows, it should be no surprise that there are a wide range of attacks specifically aimed at that operating system. In this section, we will briefly look at some of these.

Pass the Hash

We will examine cryptographic hashes at some length in Chapter 8. For now just accept that many systems store passwords as a cryptographic hash. This is done because it is impossible to "unhash" something.

The pass the hash attack essentially realizes that the hash cannot be reversed; rather than trying to find out what the password is, the attacker just sends over the hash. If the attacker can obtain a valid username and user password hashes values (just the hash—the attacker does not know the actual password), then the hacker can use that hash, without ever knowing the actual password.

Windows applications ask users to type in their passwords; then they in turn hash them. Often this can be done with an API like LsaLogonUser, converting the password to either an LM hash or an NT hash. Pass the hash skips around the application and just sends the hash.

Net User Script

This particular exploit first requires access to the target machine with at least guest-level privileges. It is based on the fact that many organizations put the technical support personnel in the domain admin's group.

The attacker writes the following two-line script (obviously the word *localaccountname* is replaced with an actual local account name.):

```
net user /domain /add localaccountname password
net group /domain "Domain Admins" /add Domain
```

Save that script in the All Users startup folder. The next time someone with domain admin privileges logs on to the machine, it will execute and that *localaccountname* will now be a domain admin. The only problem is that it may be quite some time before someone with such privileges logs onto that machine. To make this happen, the attacker will cause a problem with the system that would necessitate technical support fixing it, such as by disabling the network card. The next user to log in will not be able to access the network or Internet and will call technical support. There is a reasonably high chance that the person in technical support is a member of the domain administrators group. When that person logs on to the computer to fix the problem, unbeknownst to her the script will execute.

This particular exploit illustrates two different security issues. The first is the concept of least privileges. This means each user has the minimum privileges to do his job. This concept was discussed briefly in Chapter 1, "Introduction to Computer Security." Therefore, technical support personnel should not be in the domain admin group.

The second issue is that access to any of your machines should be controlled. This exploit only requires that the attacker have guest-level access and then only for a few minutes. From that minimum access, a skilled attacker can move forward and acquire domain admin privileges.

Login as System

This particular attack requires physical access to one machine on your network. It does not require domain or even computer login credentials. To understand this attack, think about the last time you logged into any Windows computer, even a Windows server. Next to the login text boxes (Username and Password), there is an accessibility button that allows you to launch various tools to aid those users with disabilities. For example, you can launch the magnifier class in order to magnify text.

In this attack, the perpetrator will boot the system to any Linux live CD. Then, using the FDISK utility, the attacker will locate the Windows partition. Navigating to the Windows\System32 directory, the attacker can first take magnify.exe and make a backup, perhaps naming the backup magnify.bak. Then she can take command.exe (the command prompt) and rename it magnify.exe.

Now the attacker reboots to Windows. When the login screen appears, the perpetrator clicks Accessibility and then Magnify. Since command.exe was renamed to magnify.exe, this will actually launch the command prompt. No user has logged in yet, so the command prompt will have system privileges. At this point the attacker is only limited by her knowledge of commands executed from the command prompt.

This particular attack illustrates the need for physical security. If an attacker can get even 10 minutes alone with your Windows computer, she will likely find a way to breach the network.

Penetration Testing

As was mentioned at the beginning of this chapter, these techniques can also be conducted as part of a penetration test. However, a penetration test is not simply the random application of a variety of hacking techniques. Usually a penetration test is done along with or subsequent to a vulnerability assessment. We will discuss vulnerability assessments in detail in Chapter 11.

The process is a methodical probing of a target network in order to identify weaknesses in the network. The theory behind penetration testing is that the only way to objectively determine the security level of a given network is to have a competent penetration tester attempt to breach security. There are a variety of standards that one can use to guide a penetration test.

NIST 800-115

NIST 800-115 is the National Institute of Standards and Technology guideline for security assessments for Federal Information Systems. Assessments include penetration tests. NIST 800-115 describes security assessments and has four phases:

- **Planning:** During this phase the tester needs to set specific testing goals. Often these will be related to previous risk assessment evaluations of the target network.
- **Discovery:** This phase involves using a variety of tools, including port scanners, vulnerability scanners, and manual techniques to identify or discover any issues with the target network.
- **Attack:** Now the attacker can attempt to compromise the target network by exploiting the vulnerabilities found during the discovery phase. It is in this phase that the penetration tester applies the hacking techniques we have discussed in this chapter.
- **Reporting:** The final step is to prepare a detailed report and to deliver it to the person who hired the penetration tester. The report should provide details on what vulnerabilities were exploited, how they were exploited, and what remediation steps are recommended.

Even though this approach has only four phases, these are rather broad phases that include many substeps. It is not necessary for our purposes to delve into all the details of NIST 800-115. However, these broad steps provide a framework for penetration testing, even without delving into the details. Notice that there are two steps prior to the attack phase. Planning and discovery are critical, and you will see similar items in other penetration testing standards.

National Security Agency Information Assessment Methodology

The National Security Agency (NSA) has primary responsibility for information security throughout the United States Federal government. For this reason, it formulated a methodology to be applied to any information systems assessment to include security audits, vulnerability tests, and penetration tests. That methodology is briefly described here:

- Pre-Assessment
  - Determine and manage the customer's expectations.
  - Gain an understanding of the organization's information criticality.
  - Determine customer's goals and objectives.
  - Determine the system boundaries.
  - Coordinate with customer.
  - Request documentation.
- On-Site Assessment
  - Conduct opening meeting.
  - Gather and validate system information (via interview, system demonstration, and document review).
  - Analyze assessment information.
  - Develop initial recommendations.
  - Present out-brief.
- Post-Assessment
  - Give additional review of documentation.
  - Get help understanding what you learned.
  - Report coordination (and writing).

This particular summary of steps is interesting. Let's begin with the pre-assessment. Managing customer expectations is a critical step. It is important that the customer know what a penetration test can and cannot do. Notice that this phase is all about deciding what will be done and what is expected.

The on-site assessment includes the process of examining the system and culminates in an out briefing to let the customer know the essence of what you found. Then it culminates with a report that is written and delivered in the third phase. It is also interesting to notice that in the final phase there is the substep to get additional expertise. If your penetration test or security audit found items that are outside your expertise, then it is wise to consult with an expert in that area.

PCI Penetration Testing Standard

The Payment Card Industry Data Security Standards (PCI DSS) are standards used by companies that process credit cards. We will look at PCI standards in general in Chapter 10, "Security Policies." In this section we will briefly examine the penetration testing portion of those standards. PCI DSS

Requirement 11.3.4 mandates penetration testing to validate that segmentation controls and methods are operational, effective, and isolate all out-of-scope systems from systems in the cardholder data environment.

PCI standards recommend testing a separate environment, not on the live production environment during normal business hours.

It is recommended that pen testing include social engineering tests.

Per PCI DSS Requirements 11.3.1 and 11.3.2, penetration testing must be performed at least annually and after any significant change—for example, infrastructure or application upgrade or modification—or new system component installations. As with the previous models we examined, PCI DSS has some specific steps:

- **Pre-engagement:** Defining scope, documents, rules of engagement, success criteria, and review of past issues
- **The actual penetration test:** Where you apply the hacking techniques
- **Post-Engagement:** Reporting and recommending remediation steps

It is not critical that you memorize these standards. The point is to understand that hacking techniques are utilized in penetration testing, but that penetration testing is more than just random attempts to hack the target network. It is a methodical approach to verifying the security of a target network that happens to include real hacking techniques.

Summary

In this chapter we have examined just a few techniques hackers utilize. But these techniques and tools have illustrated the need for a variety of security measures. The scanning techniques illustrate the need for blocking certain traffic at the firewall and for running an IDS. The SQL injection attack demonstrates why security must be part of application development. And the OphCrack tool illustrates why physical security is important and why the principle of least privileges is important. Putting tech support staff into the domain admins group violates the concept of least privileges and makes the privilege escalation script possible.

Test Your Skills

MULTIPLE CHOICE QUESTIONS

1. SQL injection is based on what?
   - **A.** Having database admin privileges
   - **B.** Creating an SQL statement that is always true
   - **C.** Creating an SQL statement that will force access
   - **D.** Understanding web programming
2. Which of the following is a vulnerability scanner specifically for Windows systems?
   - **A.** Nmap
   - **B.** OphCrack
   - **C.** Nessus
   - **D.** MBSA
3. How can you prevent cross-site scripting?
   - **A.** Filter user input.
   - **B.** Use an IDS.
   - **C.** Use a firewall.
   - **D.** It cannot be prevented.
4. What is an advantage of using Nessus? Use your favorite search engine to research Nessus to answer this question.
   - **A.** It is free for businesses.
   - **B.** It has a wide range of vulnerabilities it can check for.
   - **C.** It is designed for Windows systems.
   - **D.** It includes an IDS.

5. OphCrack depends on the attacker doing what?
   - **A.** Getting physical access to the machine
   - **B.** Getting domain admin privileges
   - **C.** Using social engineering
   - **D.** Using a scanning tool

6. If you wish to view items that have been removed from a website, what is the best way to do that?
   - **A.** Use Nessus.
   - **B.** Use Nmap.
   - **C.** Use www.netcraft.com.
   - **D.** Use www.archive.org.

7. Which of the following is a popular port scanner?
   - **A.** Nessus
   - **B.** OphCrack
   - **C.** MBSA
   - **D.** Nmap

8. Blocking incoming ICMP packets will prevent what type of scan?
   - **A.** SYN
   - **B.** Ping
   - **C.** FIN
   - **D.** Stealth

9. A person who uses hacking techniques for illegal activities is referred to as what?
   - **A.** A hacker
   - **B.** A gray hat hacker
   - **C.** A phreaker
   - **D.** A cracker

10. A person who hacks into phone systems is referred to as what?
    - **A.** A hacker
    - **B.** A gray hat hacker
    - **C.** A phreaker
    - **D.** A cracker

11. A person who uses tools to hack without understanding the underlying technology is called what?
    - **A.** A script kiddy
    - **B.** A gray hat hacker
    - **C.** A novice
    - **D.** A white hat hacker
12. Trying to list all the servers on a network is referred to as what?
    - **A.** Port scanning
    - **B.** Enumeration
    - **C.** Vulnerability scanning
    - **D.** Scouting
13. Which of the following is a popular enumeration tool?
    - **A.** Nessus
    - **B.** Nmap
    - **C.** MBSA
    - **D.** Cheops
14. Which of the following is considered the most stealthy port scan?
    - **A.** SYN
    - **B.** Connect
    - **C.** Ping
    - **D.** Nmap
15. What is the most stealthy way to find out what type of server a website is running?
    - **A.** Use Nmap.
    - **B.** Use Cain and Abel.
    - **C.** Use www.netcraft.com.
    - **D.** Use www.archive.org.

EXERCISES

EXERCISE 6.1: Using www.archive.org

This exercise gives you practice in using www.archive.org. Go to www.archive.org and pull up at least two previous versions of your college/university's website. What information can you find that is no longer on the website?

EXERCISE 6.2: Using Nmap

This exercise introduces you to the Nmap tool. You should download and install Nmap. Then run at least three different scans on either your own computer or on a designated lab computer.

While it is not illegal to scan a computer, it may violate some security policies for some colleges and universities. Make certain you only scan a designated lab computer.

EXERCISE 6.3: Using OphCrack

Download OphCrack to a CD. Then reboot your own machine to OphCrack and attempt to crack your own local passwords.

It is critical that you only use this on your own machine or a designated lab machine. Using it on other machines will probably violate security policies at your college/university/company.

EXERCISE 6.4: Using Netcraft.com

Visit www.netcraft.com and do a search on at least three different websites of your choosing. Note what information you are able to gather about the website.

PROJECTS

PROJECT 6.1: Passive Reconnaissance

Select a local organization and conduct a passive reconnaissance. This should include searching job boards, the organization's own website, user groups/bulletin boards, social networking sites, www.archive.org, and more. Gather as much information about the target network as you can.

PROJECT 6.2: Port Scanners

Use your favorite search engine to locate at least two other port scanners. Download and install them, and then try them on your own machine or a designated lab computer. Compare and contrast these tools to Nmap. Are they easier to use? More informative?

PROJECT 6.3: MBSA

Download and install MBSA and run a vulnerability scan on your own computer or on a designated lab computer. What problems did you find? Was the tool easy to use?

Case Study

Jane is a hacker intent on breaking into the XYZ Corporation. She uses a variety of passive reconnaissance techniques and gathers extensive information about the company. Jane finds out what model routers are being used from network administrator questions/comments in user groups. She finds a complete list of the IT staff and their phone numbers from a personnel directory on the company website. She also was able to find out what services are running by using a port scan.

From this scenario, consider the following questions:

1. What reasonable steps could the company have taken to prevent Jane from finding out about company hardware, like router models?
2. What steps should the company take to prevent or at least reduce the efficacy of port scans?

This page intentionally left blank