

ATME COLLEGE OF ENGINEERING

13th KM Stone, Bannur Road, Mysore - 560 028



A T M E

College of Engineering

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
- CYBER SECURITY



LECTURE NOTES

COURSE: Hackers Technique, Tools and Incident Handling

COURSE CODE: BCY515C

SEMESTER: V

COURSE COORDINATOR: Ms. Darshini Y

(ACADEMIC YEAR 2026)

INSTITUTIONAL MISSION AND VISION

Objectives

- To provide quality education and groom top-notch professionals, entrepreneurs and leaders for different fields of engineering, technology and management.
- To open a Training-R & D-Design-Consultancy cell in each department, gradually introduce doctoral and postdoctoral programs, encourage basic & applied research in areas of social relevance, and develop the institute as a center of excellence.
- To develop academic, professional and financial alliances with the industry as well as the academia at national and transnational levels.
- To develop academic, professional and financial alliances with the industry as well as the academia at national and transnational levels.
- To cultivate strong community relationships and involve the students and the staff in local community service.
- To constantly enhance the value of the educational inputs with the participation of students, faculty, parents and industry.

Vision

Development of academically excellent, culturally vibrant, socially responsible and globally competent human resources.

Mission

- To keep pace with advancements in knowledge and make the students competitive and capable at the global level.
- To create an environment for the students to acquire the right physical, intellectual, emotional and moral foundations and shine as torch bearers of tomorrow's society.
- To strive to attain ever-higher benchmarks of educational excellence

MODULE 4

Chapter 1. Malware, Worms, and Viruses

ONE OF THE PROBLEMS in the technology business that has grown considerably over the years is the issue of **malware**. Malware in all its forms has moved from being one of a simple annoyance to one of downright maliciousness. Software in this category has evolved to the point of being dangerous, as it now can steal passwords, personal information, and plenty of other information from an unsuspecting user.

Malware is nothing new, even though the term may be. The problem has existed for years under different names such as viruses, worms, adware, scareware, and spyware. But it has become easier to spread because of the convenient distribution channel the Internet offers, as well as the increasingly clever social-engineering methods the creators of this type of software employ. Making the problem of malware even larger is the complexity of modern software, lack of security, known vulnerabilities, and users' lax attitude toward security updates and patches.

Malware or malicious code is not going to decline; in fact, the opposite is true. One type of malware, Trojans with keyloggers, saw an increase of roughly 250 percent between January 2004 and May 2006, and such a trend represents just one category. Some types of malware have seen even larger increases.

It is with these points in mind that this chapter will examine the problem of malware, trends, and how to deal with the increasingly serious threat this type of software poses.

Chapter 10 Topics

This chapter covers the following topics and concepts:

- What malware is
- What viruses are and how they function
- What worms are and how they function
- What spyware is
- What adware is
- What scareware is

Chapter 10 Goals

When you complete this chapter, you will be able to:

MODULE 4

- List the common types of malware found in the wild
- Describe the threats posed by malware
- Describe the characteristics of malware
- Describe the threats posed by viruses
- Identify the different characteristics of malware
- Identify removal techniques and mitigation techniques for malware

Malware

The term *malware* is often tossed around, but what exactly does it mean? Malware refers to software that performs any action or activity without the knowledge or consent of the system's owner. But the definition of malware can be expanded to include any software that is inherently hostile, intrusive, or annoying in its operation.

NOTE

Malware is a contraction for the term *malicious software*, which gives a much more accurate picture of the goal of this class of software.

In the past, malware was designed to infect and disrupt, disable, or even destroy systems and applications. In some cases this disruption went one step further and used an infected system as a weapon to disable or disrupt other systems. In recent years the nature of malware has changed with the software seeking to remain out of sight in an effort to evade detection and removal by the system owner for as long as possible. All the while, the malware is resident on a system taking up resources and power for whatever purpose the attacking or infecting party may have in mind.

NOTE

If the definition of malware is limited to just software that performs actions without the user's knowledge or consent, this could include a large amount of software on the average system. It is also important to classify as malware software that is hostile in nature.

In the present day malware has changed in nature dramatically with the criminal element realizing the advantages of using it for more malicious purposes. In the past it was not uncommon for malware to be written as a prank or to annoy the victim, but times have changed. Malware in the current day has been adopted by criminals for a wide array of purposes to capture information about the victim or commit other

MODULE 4

acts. As technology has evolved, so has malware—from the annoying to the downright malicious.

FYI

Increasing amounts of malware have shown up over the past decade with the goal of making some sort of financial gain for their creators. In the 1990s the idea of financial gain from such software started in the form of dialers that would use a computer's modem to call up numbers such as adult services or other types to generate revenue. Over the last few years the tactics have changed, however, with malware tracking a person's actions all the way to targeting ads and other items on a victim's system.

The term *malware* used to cover just viruses, worms, Trojans, and other similar software that performed no useful function or carried out malicious activities. Malware has evolved to include new forms, such as spyware, adware, and scareware. Software that used to just dial up systems or be annoying now redirects browsers, targets search engine results, or even displays advertisements on a system.

Another aspect of malware that has emerged is its use to steal information. Malware programs have been known to install what is known as a keylogger on a system. The intention here is to capture keystrokes when entered with the intention of gathering information such as credit card numbers, bank account numbers, or other similar information. For example, malware has been used to steal information from those engaging in online gaming to obtain players' game account information.

FYI

Malware doesn't necessarily hide from the user in every case; it depends on the intended purpose of the creator. In some cases, spyware creators have stated their intentions outright by presenting **end user license agreements (EULAs)** to the victim. Because most users never read EULAs and the document looks legitimate, they tend to install the software without realizing that the document may clear the attacker of responsibility.

FYI

The popular online game by Activision Blizzard known as World of Warcraft (WoW) has been a target of multiple keyloggers since its debut. The intention with most keyloggers that have targeted this game has been to capture what is known as

MODULE 4

an Authentication Code, used to authenticate user accounts. When a victim is infected, the code is intercepted when entered and a false code is sent to the WoW servers. The attackers get the real code at this point and can now log onto the account directly while the victim is left out in the cold.

Malware's Legality

Malware has tested and defined legal boundaries since it came into being. Lawmakers have passed statues specifically to deal with the problem. Malware initially was perceived as being harmless, relegated to the status of a prank. But times changed—a more serious look at the problem of malware became necessary. Over the past few years the problems malicious code poses have been addressed technologically. In addition, new legal remedies have emerged in several countries.

In the United States several laws have been introduced since the 1980s. Some of the more notable ones include:

- **The Computer Fraud and Abuse Act 1986**—This law was originally passed to address federal computer-related offenses and the cracking of computer systems. The act applies to cases that involve federal interests, or situations involving federal government computers or those of financial institutions. Additionally the law covers computer crime that crosses state lines or jurisdictions.
- **The Patriot Act**—This expanded on the powers already included in the Computer Fraud and Abuse Act. The law:
 - Provides penalties of up to 10 years for a first offense and 20 years for a second offense
 - Assesses damages over the course of a year to multiple systems to determine if such damages are more than \$5,000 total

FYI

In 2009 Canada enacted the Electronic Commerce Protection Act (ECPA), which was designed to meet the problem of malware head-on. The ECPA has several provisions for both spam and malware designed to limit the proliferation of the software both inside and outside Canada. The act introduces some steep fines of up to \$10 million for an organization and \$1 million for an individual for those installing unauthorized software on a system.

MODULE 4

- Increases punishment for any violation that involves systems that process information relating to the justice system or military
- Covers damage to foreign computers involved in US interstate commerce
- Includes, in calculating damages, the time and money spent investigating a crime
- Makes selling computer systems infected with malware a federal offense.

Each country has approached the problem of malware a little differently, with penalties ranging from jail time to potentially steep fines for violators. In the United States, states such as California, West Virginia, and a host of others have put in place laws designed to punish malware perpetrators. While the laws have different penalties designed to address malware's effects, it has yet to be seen what the effects of these laws will be.

Types of Malware

While the term *malware* may refer to any software that fits the definition, it is also important to understand the specifics and significance of each piece of software under the malware banner. A broad range of software types and categories exists, some of which have been around for a long time. Malware includes the following:

- Viruses
- Worms
- Spyware
- Adware
- Scareware
- Trojan horses
- Rootkits

The latter two will be discussed in the next chapter.

Malware's Targets

A quick review of the targets of malware authors gives a good taste of why the problem is so serious:

- **Credit card data**—Credit card data and personal information is a tempting and all too common target. Upon obtaining this information an attacker can

MODULE 4

go on a shopping spree, purchasing any type of product or service: Web services, games, merchandise, or other products.

- **Passwords**—Passwords are another attractive target for attackers. The compromise of this sort of information can be devastating to the victim. Most individuals will reuse passwords over and over again, and stealing a person's password can easily open many doors to the attacker. Stealing passwords can allow a hacker to read passwords from a system that includes everything from e-mail and Internet accounts to banking passwords.
- **Insider information**—Confidential or insider information is another target for an attacker. An attacker may very well use malware to gain such information from an organization to gain a competitive or financial benefit.
- **Data storage**—In some cases a system infected with malware may find itself a point for storing data without the owners' knowledge. Uploading data to an infected system can turn that system into a server hosting any type of content. This has included illegal music or movies, pirated software, pornography, financial data, or even child pornography.

Viruses and How They Function

A virus is one of the oldest pieces of software that fits under the definition of malware. It may also be one of the most frequently misunderstood. The term *virus* is frequently used to refer to all types of malware.

Before getting too far into a discussion of viruses it is important to make clear first what a virus actually is and the behaviors viruses exhibit. A virus is a piece of code or software that spreads from system to system by attaching itself to other files. When the file is accessed, the virus is activated. Once activated, the code carries out whatever attack or action the author wishes to execute, such as corrupting data or destroying it outright.

Viruses have a long history, one that shows how this form of malware adapted and evolved as technology and detective techniques improved. Let's examine the "back story" of viruses, how they have changed with the times, and how this affects you as a security professional.

Viruses: A History

As stated earlier, viruses are nothing new; the first viruses debuted in the "wild" roughly 40 years ago as research projects. They have evolved dramatically since then into the malicious weapons they are today.

MODULE 4

NOTE

A second piece of code, known as the Reaper, was specifically designed to remove the Creeper from circulation.

The first recognized virus was created as a proof-of-concept application designed in 1971 to demonstrate what was known as a *mobile application*. In practice the Creeper virus, as it was known, spread from system to system by locating a new system while resident on another. When a new system was found the virus would copy itself and delete itself off the old one. Additionally the Creeper virus would print out a message on an infected machine that stated "I'm the Creeper, catch me if you can." In practice the virus was harmless and was not that advanced compared with modern examples.

NOTE

The term *virus* was not coined until the 1980s, so the negative term was not applied to these early examples.

In the mid-1970s a new feature was introduced in the Wabbit virus. The Wabbit virus represented a change in tactics in that it demonstrated one of the features associated with modern day viruses—replication. The virus replicated on the same computer over and over again until the system was overrun and eventually crashed.

In 1982 the first virus seen outside academia debuted in the form of the ElkCloner virus. This piece of malware debuted another feature of later viruses—the ability to spread rapidly and remain in the computer's memory to cause further infection. Once resident in memory, it would infect floppy disks placed into the system later, as many later viruses would do.

NOTE

The ElkCloner virus was developed by Rich Skrenta when he was all of 15 years old. He developed the virus to have fun with friends who no longer trusted floppies that he gave them. He came up with the novel concept of infecting floppies with a memory-resident program.

Four short years later, the first PC-compatible virus debuted. The viruses prior to this point were Apple II types or designed for specific research networks. In 1986 the first of what was known as **boot sector** viruses debuted, demonstrating a

MODULE 4

technique later seen on a much wider scale. This type of virus infected the boot sector of a drive and would spread its infection when the system was going through its boot process.

NOTE

The first logic bomb most individuals heard of was the Michelangelo virus, designed to infect on the famous painter's birthday. In reality the virus was a great non-event—it was detected very early and eradicated before it could cause any serious damage.

The first of what would later be called logic bombs debuted in 1987: the Jerusalem virus. This virus was designed to cause damage only on a certain date—in this case, Friday the 13th. The virus was so named because of its initial discovery in Jerusalem.

Multipartite viruses made their appearance in 1989 in the Ghostball virus. This virus was designed to cause damage using multiple methods and components, all of which had to be neutralized and removed to clear out the virus effectively. Polymorphic viruses first appeared in 1992 as a way to evade early virus-detection techniques. Polymorphic viruses are designed to change their code and "shape" to avoid detection by virus scanners, which would look for a specific virus code and not the new version.

Fast-forward to 2008 and MoccmeX. MoccmeX was shipped on digital photo frames manufactured in China. When the virus infected a system, its firewall and antivirus software were disabled; then the virus would attempt to steal online-game passwords.

Modern viruses and virus writers have gotten much more creative in their efforts and in some cases are financed by criminal organizations to build their software.

Types of Viruses

So you can see that not all viruses are the same; there are several variations of viruses, each of which is dangerous in its own way. Understanding each type of virus can give you a better idea of how to thwart them and address the threats they pose.

FYI

On October 29, 2008, a logic bomb was discovered at Fannie Mae, the Federal National Mortgage Association, in the United States. The bomb was created and installed by Rajendrasinh Makwana, an IT contractor who worked in Fannie Mae's

MODULE 4

Urbana, Maryland, facility. As designed, the bomb was to activate on January 31, 2009. If successful, it would have wiped all of Fannie Mae's more than 4,000 servers.

Makwana, upset that he had been terminated, planted the bomb before his network access was terminated. He was indicted in a Maryland court on January 27, 2009, for unauthorized computer access.

Logic Bombs

A logic bomb is a piece of code or software designed to lie in wait on a system until a specified event occurs. When the event occurs the bomb "goes off" and carries out its destructive behavior as the creator intended. While the options are literally endless as far as what a logic bomb can do, the common use of this type of device is to destroy data or systems.

Logic bombs have been notoriously difficult to detect because of their very nature of being "harmless" until they activate. Malware of this type is simply dormant until whatever it is designed to look for happens. What can activate this software is known as a positive or negative trigger event coded in by the creator. A positive trigger is a mechanism that looks for an event to occur such as a date. A negative trigger, on the other hand, is designed to monitor an action; when such action does not occur it goes off. An example would be if a user does not log on for some period. This process of "hiding" until an event occurs or does not occur makes this particular type of malware dangerous.

As a security professional you will have to be extra vigilant to detect logic bombs before they do damage. Traditionally the two most likely ways to detect this type of device are by accident or after the fact. In the first method, an IT worker just happens to stumble upon the device by sheer "dumb luck" and deactivates the bomb. In the second method, the device "detonates" and then the cleanup begins. The best detection and prevention methods are to be vigilant, to limit access of employees to only what is necessary, and to restrict access where possible.

Polymorphic Viruses

The polymorphic virus is unique because of its ability to change its "shape" to evade antivirus programs and therefore detection. In practice this type of malware possesses code that allows it to hide and mutate itself in random ways that prevent detection. This technique debuted in the late 1980s as a method to avoid the detection techniques of the time.

MODULE 4

Polymorphic viruses employ a series of techniques to change or mutate, these methods include:

- **Polymorphic engines**—Designed to alter or mutate the device's design while keeping the payload, the part that does the damage, intact
- **Encryption**—Used to scramble or hide the damaging payload, keeping antivirus engines from detecting it

When in action, polymorphic viruses rewrite or change themselves upon every execution. The extent of the change is determined by the creator of the virus and can include simple rewrite to changes in encryption routines or alteration of code.

Modern antivirus software is much better equipped to deal with the problems polymorphic viruses pose. Techniques to detect these types of viruses include decryption of the virus and statistical analysis and heuristics designed to reveal the software's behavior.

Multipartite Viruses

The term multipartite refers to a virus that infects using multiple attack vectors, including the boot sector and executable files on the hard drive. What makes these types of viruses dangerous and powerful weapons is that to stop them, you must totally remove all their parts. If any part of the virus is not eradicated from the infected system, it can re-infect the system.

Multipartite viruses represent a problem because they can reside in different locations and carry out different activities. This class of virus has two parts, a boot infector and a file infector. If the boot infector is removed the file infector will re-infect the computer. Conversely, if the file infector is removed the boot sector will re-infect the computer.

Macro Viruses

Macro viruses are a class of virus that infects and operates through the use of a macro language. A macro language is a programming language built into applications such as Microsoft Office in the form of Visual Basic for Applications (VBA). It is designed to automate repetitive tasks. Macro viruses have been very effective because users have lacked the protection or knowledge to counteract them.

NOTE

MODULE 4

After the initial outbreaks of macro viruses, Microsoft introduced the ability to disable macros. In Office 2010 macros are disabled by default.

Macro viruses can be implemented in different ways, usually by being embedded into a file or spread via e-mail. The initial infections spread quite quickly because earlier applications would run the macro when a file was opened or when an e-mail was viewed. Since the debut of these viruses, most modern applications disable the macro feature or ask users whether they want to run macros.

Hoaxes

A hoax is not a true virus. But no discussion of viruses is complete without mentioning the hoax virus. Hoax viruses are those designed to make the user take action even though no infection or threat exists. The following example is an e-mail that actually is a hoax "virus."

PLEASE FORWARD THIS WARNING AMONG FRIENDS, FAMILY AND CONTACTS: You should be alert during the next days: Do not open any message with an attached file called "Invitation" regardless of who sent it. It is a virus that opens an Olympic Torch which "burns" the whole hard disc C of your computer. This virus will be received from someone who has your e-mail address in his/her contact list. That is why you should send this e-mail to all your contacts. It is better to receive this message 25 times than to receive the virus and open it. If you receive a mail called "Invitation," though sent by a friend, do not open it and shut down your computer immediately. This is the worst virus announced by CNN; it has been classified by Microsoft as the most destructive virus ever. This virus was discovered by McAfee yesterday, and there is no repair yet for this kind of virus. This virus simply destroys the Zero Sector of the Hard Disc, where the vital information is kept. SEND THIS E-MAIL TO EVERYONE YOU KNOW, COPY THIS E-MAIL AND SEND IT TO YOUR FRIENDS AND REMEMBER: IF YOU SEND IT TO THEM, YOU WILL BENEFIT ALL OF US.

Here's another example:

All,

There's a new virus which was found recently which will erase the whole C drive. If you get a mail with the subject "Economic Slow Down in US" please delete that mail right away. Otherwise it will erase the whole C drive. As soon as you open it, it says, "Your system will restart now ... Do you want to continue?". Even if you click on

MODULE 4

NO, your system will be shut down and will never boot again. It already caused major damage in the US and few other parts of the world. The remedy for this has not yet been discovered.

Please make sure you have backed up any local hard drive files adequately—network, floppy, etc.

In both cases a simple search of Google or discussion with the IT department of a company will reveal these to be hoaxes; however, in many cases the recipients of these messages panic and forward them on, causing further panic.

Prevention Techniques

Viruses have been in the computer and network business almost as long as the business itself has been around. A wide variety of techniques and tools have evolved to deal with the threat.

Education

Knowledge is half the battle. Getting system owners to understand how not to get infected or spread viruses is a huge element in stopping the problem. Users should be instructed on proper procedures to stop the spread of virus code. Such tips should generally include:

- Don't allow employees to bring media from home
- Instruct users not to download files except from known and trusted sources
- Don't allow workers to install software without permission from the company IT department
- Inform IT or security of strange system behaviors or virus notifications
- Ban flash drives
- Ban portable hard drives
- Limit the use of administrative accounts

Antivirus

The next line of defense is the antivirus software that is designed to stop the spread and activity of viruses. Antiviruses are designed to run in the background on a system, staying vigilant for activity that suggests viruses and stopping or shutting it down. Antiviruses are effective tools, but they can be so only if they are kept up to date. Antiviruses rely on a database of signatures that lets them know what to look

MODULE 4

for and remove. Because new viruses are released each day, if you neglect this database it becomes much more likely a virus will get through.

Because there is a wide range of viruses and other malicious code, an antivirus must be able to detect more than a simple virus. Good antivirus software can detect viruses, worms, Trojans, phishing attacks, and, in some cases, spyware.

Antiviruses tend to use one of two different methods. The first is the *suspicious behavior method*. Antivirus programs use this to monitor the behavior of applications on a system. This approach is widely used as it can detect suspicious behavior in existing programs, as well as detecting suspicious behavior that indicates a new virus may be attempting to infect your system.

The second method is dictionary-based detection. This method will scan applications and other files when they have access to your system. This advantage of this method is that it can detect a virus almost immediately instead of letting it run and detecting the behavior later. The downside is that the method can detect only viruses that it knows about—if you neglect to update the software it cannot detect new viruses.

Applying Updates

Another detail that you cannot overlook is applying patches on systems and software when they become available. Vendors of operating systems and applications such as Microsoft regularly release patches designed to close holes and address vulnerabilities on systems that viruses could exploit. Missing a patch or update can easily mean the difference between avoiding a problem and having your system crippled.

NOTE

Microsoft is one of many software vendors that have made a point of regularly addressing security issues via patches. In Microsoft's case a monthly event known as Patch Tuesday is specifically geared toward addressing security issues.

Worms and How They Function

Worms are a different type of malware altogether. Viruses require user intervention for their infection to take place—such as the opening of a file or the booting of a computer. In the case of worms, however, no user action is required. A worm is a self-replicating piece of software that combines the convenience of computer

MODULE 4

networks with the power of malware. Worms also differ from viruses in that viruses require a host program to stay resident. A worm does not require this and is actually self-contained. Worms also can cause substantially more harm than a virus, which is typically limited to corrupting data and applications.

NOTE

Worms can cause alterations to or corruption of data on a system, but can also cause damage indirectly by replicating at a rapid rate, clogging networks with traffic they cannot handle.

An earlier chapter mentioned the earliest recognized worm, known now as the Morris worm. This worm exhibited some of the traits associated with modern-day worms, particularly the ability to rapidly replicate. At the time the Morris worm was unleashed, the Internet was small compared with today, but the effect was no less devastating. The worm replicated so rapidly and so aggressively that networks were clogged with traffic and brought down. Estimates at the time placed the damage from the outbreak at \$10 million (not adjusted for inflation).

NOTE

The fallout from the Morris worm is still debated today, with damage estimates ranging up to \$100 million and several thousand computers or more infected. While the numbers can be argued, what cannot be is the impact of the infection. People realized that worms posed a threat and that tougher laws on cybercrime were needed.

One worm that caused widespread damage was the SQL Slammer or Slammer worm. The Slammer worm was responsible for widespread slowdowns and denials of service on the Internet. It was designed to exploit a known buffer overflow in Microsoft's SQL Server and SQL Server Desktop Engine products. Even though Microsoft had released a software patch six months before the actual infection, many had neglected to install the patch, and therefore the vulnerability still existed on many systems. As a result, in the early morning hours of January 25, 2003, the worm became active and in less than 10 minutes had infected 75,000 machines.

How Worms Work

Worms are relatively simple in design and function, but very dangerous due to the speed and effectiveness with which they spread. Most worms share certain

MODULE 4

characteristics, which help define how they work and what they can do. The characteristics are as follows:

- Do not need a host program to function
- Do not require user intervention
- Replicate rapidly
- Consume bandwidth and resources

Worms can also perform some other functions, including:

- Transmit information from a victim system
- Carry a payload such as a virus

Examining these characteristics a bit more in detail will help you understand how a worm works and the challenges worms pose to a security professional. In fact, worms differ from viruses in two key ways:

- A worm can be considered a special type of malware that can replicate and consume memory, but not attach to other programs.
- A worm spreads through infected networks automatically, while a virus does not.

One of the main characteristics of worms is that they do not need a host program to function, unlike their fellow malware viruses. Worms are designed to function by leveraging vulnerabilities on a target system that is generally unknown or unpatched. Once a worm locates one of these vulnerabilities, it infects the system and then uses the system to spread and infect other systems. A worm performs all these functions by using the system's own processes to do its job, but does not require any host program to run before starting the initial process.

Another characteristic that differentiates worms from other malware is their ability to run without user intervention. Viruses, for example, require a host program to be executed for the infection to begin; worms simply need the vulnerability to exist in order for the process to take place. In the case of worms, just having a system turned on and connected to the Internet is enough to make it a target. Combine this with the vulnerabilities and the danger is obvious.

NOTE

MODULE 4

The Slammer worm doubled the number of infected machines every 8.5 seconds, much faster than previous worms. Slammer boasted an infection rate that was 250 times as fast as Code Red, which had come only two years earlier.

Since Day 1, worms have possessed a feature that makes them a dangerous force to deal with—their ability to replicate very rapidly. One of the features of the Morris worm that even its creator did not expect was that it replicated so rapidly that it choked up networks and shut them down quite effectively. This feature has been a characteristic of worms ever since. Worms can replicate so quickly that their creators are frequently caught off guard. This replication is made possible by a number of factors, including poorly maintained systems, networked systems, and the number of systems linked via the Internet.

LIGHT SIDE VERSUS DARK SIDE

Some worms have been created for benign purposes. One such family of worms is the Nachi family. Nachi was designed to locate systems that had certain vulnerabilities not patched by the system owner. It would then download the appropriate patches to fix the problem.

Such worms introduced several questions. Among them was, if a worm has benign purposes in mind, is it OK? This question has compelling arguments on both sides.

Probably the most visible or dramatic feature of worms is their consumption of resources, which shows up as a side effect. Mix into this equation of speed and replication the number of computers on the Internet, and you have a situation that leads to bandwidth resources being consumed on a huge scale. Worms such as Slammer caused massive slowdowns on the Internet due to the scans it sent out looking for vulnerable systems and the way it moved its payload around. Additionally, the worm consumed resources on infected systems as it replicated off the system, using system resources to do so.

NOTE

One of the earliest warning signs of worms is the unexplained slowdown of a system even after repeated reboots or other checks. While not always a sign of a worm, it is one of the red flags that the system owner should investigate.

In recent years some new characteristics have been added to the behaviors of worms, one of which is the ability to carry a payload. While traditionally worms have not

MODULE 4

directly damaged systems, worms that carry payloads can do all sorts of mischief. One of the more creative uses of worms has been to perform "cryptoviral extortion." The worm drops off a payload that looks for specific file types (such as .doc files) and encrypts them. Once this has taken place, the worm leaves a message for the user offering to reveal the encryption key after the user pays a certain amount of money.

Stopping Worms

At the core of the worm problem is operating systems that have overlooked or unpatched vulnerabilities. Vendors such as Microsoft have made concerted efforts to release patches regularly to address issues in their operating systems—including vulnerabilities that worms could use to spread. The problem becomes one of knowing patches are available for a system and applying them. This problem becomes even bigger when you realize that worms aren't restricted just to corporate systems—they can also hit home users, who are more likely to miss patches. In some cases, patches are not yet released for a vulnerability. This leads to what is called a *zero-day exploit*, in which a hole can be exploited immediately.

NOTE

Several worms such as Code Red, Nimda, Blaster, and Slammer are still alive and well on the Internet today, although at levels well below their initial outbreak. These worms, some of which are nine years old, still infect systems. The main reason? System owners that have neglected to patch their systems, either out of ignorance or laziness.

The Power of Education

Much as with viruses, education is key to stopping worms. Worms are frequently spread via e-mail applications by e-mails bearing the name ILOVEYOU, for example. These prey on a user's curiosity—the user opens the e-mail and unknowingly runs the worm in the background. Add in attacks such as phishing, which further pique a user's curiosity, and you have a problem that only education can address.

NOTE

MODULE 4

The old saying "An ounce of prevention is worth a pound of cure" applies to virus and worm prevention, as it is vastly easier to stop the problem before it starts than to try to remedy it after the fact.

Antivirus and Firewalls

One of the primary lines of defense against worms is reputable antivirus and anti-spyware applications. Having an antivirus application on a system helps prevent a worm infection—but only if it is kept up to date. Modern and up-to-date antivirus applications can easily stop most worms when they appear.

Another way to stop worms is the firewall. The firewall is a valuable tool as it can block the scans to and from a system that worms use both to spread the infection and to deliver it from an infected system to other systems. Most modern operating systems such as Microsoft's Windows 7 include this feature as part of the core system.

Spyware

Spyware is software designed to collect and report information on a user's activities without the user's knowledge or consent. Spyware can collect any type of information about the user that the author wishes to gather, such as:

- Browsing habits
- Keystrokes
- Software usage
- General computer usage

Spyware has been used to gather information for any reason that its author deems useful. The information collected has been used to target ads, generate revenue for the author, steal personal information, or steal data from an infected system. In some cases, spyware has gone beyond simple information collection to altering a system's behavior to be more along the lines of the author's wishes. Additionally, spyware has been known to act as a precursor to further attacks or infection. It can be used to download and install software designed to perform other tasks.

Methods of Infection

MODULE 4

Spyware can be placed on a system by a number of different methods, each of which is effective in its own way. When the software is installed, it typically remains hidden and proceeds to carry out its task. Delivery methods for spyware include:

- **Peer-to-peer networks (P2P)**—This delivery mechanism has become very popular because of the increased number of individuals using these networks to obtain free software.
- **Instant messaging (IM)**—Delivering malicious software via IM is easy and because IM software has never had much in the way of security controls.
- **Internet Relay Chat (IRC)**—IRC is a commonly used mechanism to deliver messages and software because of its widespread use and the ability to entice new users to download software.
- **E-mail attachments**—With the rise of e-mail as a communication medium, the practice of using it to distribute malware has also risen.
- **Physical access**—Once an attacker gains physical access, it becomes relatively easy to install the spyware and compromise the system.
- **Browser defects**—With many users forgetting or not choosing to update their browsers as soon as updates are released, distribution of spyware becomes easier.
- **Freeware**—Downloading software free from unknown or untrusted sources can mean that you may have downloaded something nastier, such as spyware.

One of the more common ways to install software on a system is through Web browsing. When a user visits a given Web site, the spyware is downloaded and installed using scripting or some other means. Spyware installed in this manner is quite common as Web browsers lend themselves to this process—they are frequently unpatched, do not have upgrades applied, or are incorrectly configured. In most cases users do not use the most basic security precautions that come with a browser, in some cases overriding them to get a better browsing experience or to see fewer popups or prompts.

NOTE

In some articles and publications, this installation method is referred to as drive-by downloads.

FYI

In Windows Vista one of the much-maligned features was known as the UAC or User Account Control. One thing this feature was designed to prevent is software

MODULE 4

installing or other activity happening without a user's knowledge. Because some users hated the change in behavior between Vista and Windows XP, they shut off this feature to stop the nag screen. But this also disabled protection in Internet Explorer designed to offer more security, including against spyware.



Figure 10-1. Installation options.

Bundling with Software

Another common way to place software on a user's system is via installation of other software that the user intentionally installs. In these cases, a user downloads a legitimate piece of software from a Web site and then proceeds to install it. During the installation process the user is prompted to install additional software before proceeding. In most cases users believe that they can't install the software they want without accepting it. Or they simply click the "Next" button and don't pay attention. Other ways to get spyware on a system during installation are strategically placed checkboxes that install spyware-type applications by default. Such a dialog is shown in [Figure 10-1](#).

Adware

MODULE 4

You will frequently find **adware** in the same machines infected with spyware. Adware is software specifically designed to display ads your system in the form of popups or nag screens. When this class of software is deployed with spyware, the effect can be quite dramatic, as you will be bombarded with ads specifically targeted to you and your search habits.

In a number of situations, adware is installed on victims' systems because it's been bundled with software that they wish to install. In these situations, when adware is installed it can monitor the usage of the software it was installed with or it can monitor a wide range of other activities. When a piece of adware is installed on a system, the goals can be very different from those of spyware or other types of malware. In the early days of adware, it was not uncommon for adware to be installed because developers wanted to make more money from their software than they otherwise could. When such software is installed, you will typically not notice until you are presented with ads or other types of prompts.

FYI

It is not unheard of for versions of software in which developers have embedded adware to be re-released by the pirate software community without the adware in place. One such example is the file sharing software Kazaa. Kazaa had a version that included spyware/adware in it as part of the normal free installation. However, this software was cracked and released without the adware in place. Of course, this raises the question: What did the pirates include?

In other cases, adware is not hidden from the user; it is much more obvious. Some developers will offer different versions of their software, one with ads and one without. Users wishing to get the software free must tolerate the annoyance of ads. Users wishing to avoid ads must pay for the privilege.

NOTE

It is common for developers of so-called freeware to include adware as part of their product. In fact, some well-known software such as Google Earth bundles other software with it, such as browsers or other products. Most manufacturers of this type of software justify their actions as a way to provide the software free or at low cost.

Scareware

MODULE 4

Scareware is a type of malware designed to trick victims into purchasing and downloading useless and potentially dangerous software.

Scareware generates authentic looking popups and other ads on a system to make users think something bad has happened or will happen. For example, a common tactic is to display a popup on-screen that appears to initiate a virus scan. It inevitably locates a "virus" and then presents you with an offer to purchase software that removes it. In most cases this software is worthless or actually installs something else that performs other nasty actions, such as those connected to spyware. Users who fall for this scam typically find themselves at the very least out some amount of money—not to mention that whatever they installed may have damaged their system.

NOTE

This type of software has become more common over the last few years as users have become more savvy, and malware authors have had to change their tactics. Enticing users to click on realistic dialogs and presenting real-looking error messages can be powerful ways to place illicit software on a user's system.

What makes this software even worse is that it frequently employs techniques that outright frighten system users. In addition to generating large numbers of bogus error messages, this class of malware may also generate real-looking dialogs such as those seen in Windows. When you click on these "dialogs" to close them, they may actually be installing the software.

When executed, some scareware will go one step further, even weakening existing system security. Scareware has been known to install on a system and specifically hunt down and disable protective software such as firewalls and antiviruses. Even worse, some of this software will even prevent updates from the system vendor, meaning that security holes and defects may no longer be fixed.

Removing scareware can be a daunting task, because it disables legitimate software that protects the system. In some cases, the system may be so compromised that all Internet activity and other update systems may error out, preventing you from making any changes.

Current tactics have evolved even further to include extortion. Recent tactics have included installing software on a system that hunts for certain file types (i.e., Word documents) that it encrypts. It then offers to decrypt them only if the user pays up.

MODULE 4

CHAPTER SUMMARY

Malware has increased in power and aggressiveness over the past few years to the point where a security professional cannot overlook or ignore the threat. Malware has taken many forms and has moved from being a simple annoyance to being criminal mischief. Software in this category has evolved dramatically to the point of being extremely malicious. Malware can now steal passwords, personal information, and plenty of other information from an unsuspecting user.

The modern concept of malware first came into being in the 1980s and 1990s. Terms such as viruses, worms, adware, scareware, and spyware have become more common in popular usage. In the past, malware was just annoying. But it has become easier to spread because of the convenient distribution channel the Internet offers, as well as the increasingly clever social engineering methods the creators of this type of software employ. Making the problem of malware even worse is the complexity of modern software, frequent lack of security, known vulnerabilities, and the lax attitude many users have toward applying security updates and patches.

New types of malware have included increasingly common scareware. Software in this category is designed to scare you into installing the package. When you do, it takes over the system and disables protective mechanisms or other items.

KEY CONCEPTS AND TERMS

- **Adware**
- **Boot sector**
- **End user license agreement (EULA)**
- **Malware**
- **Scareware**
- **Worms**

CHAPTER 10 ASSESSMENT

1. Viruses do not require a host program.
 1. True
 2. False
2. Worms are designed to replicate repeatedly.
 1. True
 2. False
3. _____ is designed to intimidate users.

MODULE 4

1. Adware
 2. Viruses
 3. Scareware
 4. Worms
4. Which is used to intercept user information?
1. Adware
 2. Scareware
 3. Spyware
 4. Viruses
5. _____ is known to disable protective mechanisms on a system such as antiviruses, antispyware, and firewalls, and to report on a user's activities.
1. Adware
 2. Scareware
 3. Spyware
 4. A virus
6. Which of the following is a characteristic of adware?
1. Gathering information
 2. Displaying popups
 3. Intimidating users
 4. Replicating
7. Prevention of viruses and malware includes _____.
1. Popup blockers
 2. Antivirus
 3. Buffer overflows
 4. All of the above
8. _____ is a powerful preventative measure to stopping viruses.
9. Which of the following can limit the impact of worms?
1. Antiviruses, firewalls, patches
 2. Anti-spyware, firewalls, patches
 3. Anti-wormware, firewalls, patches
 4. Anti-malware
10. _____ attach(es) to files.
1. Viruses
 2. Worms
 3. Adware
 4. Spyware
11. Multipartite viruses come in encrypted form.
1. True
 2. False

MODULE 4

12. _____ record(s) a user's typing.
1. Spyware
 2. Virsues
 3. Adware
 4. Malware
13. _____ are configured to go off at a certain date, time, or when a specific event occurs.
14. Scareware is harmless.
1. True
 2. False

table of contentssearch

Settings

9. Web and Database Attacks

10. Malware, Worms, and Viruses

queue

Hacker Techniques, Tools, and Incident Handling

11. Trojans and Backdoors

Table of contents collapsed

MODULE 4

Chapter 2.

Sniffers, Session Hijacking, and Denial of Service Attacks

THIS CHAPTER FOCUSES ON three broad types of network attacks: sniffers, session hijacking, and denial of service (DoS) attacks. Each of these is a dangerous tool in the hands of a skilled attacker, so you must have a thorough understanding of each one.

The first discussion in this chapter is on the topic of sniffing, or observing communications on the network in either a passive or an active mode. With sniffing you can see what is being transmitted on the network unprotected and potentially intercept sensitive information to use against the network or system owner. Sniffers are designed to go after and compromise the confidentiality of data as it flows across the network, capturing this data, and putting it in the hands of an unauthorized party.

An extension or upgrade to sniffing is the session hijack, which is a more aggressive and powerful weapon in the hacker's arsenal. A session hijack involves taking over an existing authenticated session and using it to monitor or manipulate the traffic and potentially execute commands on a system remotely. In its most advanced stages, session hijacking directly affects and attacks the integrity of information in an organization. Attackers using this technique can modify information at will as they have the credentials of the victim and whatever they have access to.

Denial of service (DoS) is the third type of attack covered in this chapter. It generally involves one computer targeting another, seeking to shut it down and deny legitimate use of its services. A distributed denial of service attack (DDoS) involves hundreds or even thousands of systems seeking to shut down a targeted system or a network. Such large-scale attacks are typically accomplished with the aid of botnets—networks of infected systems conscripted to do hackers' dirty work for them.

Chapter 12 Topics

This chapter covers the following topics and concepts:

- What sniffers are
- What session hijacking is
- What denial of service (DoS) is
- What distributed denial of service (DoS) attacks are
- What botnets are

Chapter 12 Goals

When you complete this chapter, you will be able to:

- Describe the value of sniffers
- Describe the purpose of session hijacking

MODULE 4

- Describe the process of DoS attacks
- Describe botnets
- List the capabilities of sniffers
- Describe the process of session hijacking
- Describe the features of a DoS attack

Sniffers

A sniffer is a valuable piece of software or a dangerous piece of software, depending on who is using the application. Before getting into a discussion of sniffers, it is necessary to understand what the program actually does. The simple definition of sniffers is that they are an application or device that is designed to capture, or "sniff," network traffic as it moves across the network itself. In the context of this book, sniffers are a technology used to steal or observe information that you may not otherwise have access to. A sniffer can give an attacker access to a large amount of information, including e-mail passwords, Web passwords, File Transfer Protocol (FTP) credentials, e-mail contents, and transferred files.

NOTE

Like most technologies, sniffers are not inherently bad or evil—it all depends on the intent of the user of the technology. Sniffers in the hands of a network administrator can be used to diagnose network problems and uncover design problems in the network.

Sniffers rely on the inherent insecurity in networks and the protocols that are in use on them. Recall that the Transmission Control Protocol/Internet Protocol (TCP/IP) suite was designed for a more trusting time, and therefore the protocols do not offer much in the way of security. Several protocols lend themselves to easy sniffing:

- **Telnet**—Keystrokes, such as those including usernames and passwords, that can be easily sniffed.
- **Hypertext Transfer Protocol (HTTP)**—Designed to send information in the clear without any protection and as such, a good target for sniffing
- **Simple Mail Transfer Protocol (SMTP)**—Commonly used in the transfer of e-mail, the protocol is simple and efficient, but it does not include any protection against sniffing.
- **Network News Transfer Protocol (NNTP)**—All communication is sent in the clear, including passwords and data.
- **Post Office Protocol (POP)**—Designed to retrieve e-mail from servers, but again does not include protection against sniffing as passwords and usernames can be intercepted
- **File Transfer Protocol (FTP)**—A protocol designed to send and receive files; all transmissions are sent in the clear in this protocol.
- **Internet Message Access Protocol (IMAP)**—Similar to SMTP in function and lack of protection

Sniffers are a powerful part of the security professional's toolkit, offering the ability to peek into the traffic that is on the network and observe the communications that are taking place. How does

MODULE 4

a sniffer get this ability? Typically a computer system can see only the communications that are specifically addressed to it or from it, but a sniffer possesses the ability to see all communications, whether they are addressed to the listening station or not. This ability is made possible by switching the network card into **promiscuous mode**. Promiscuous mode is the ability of the network card to see all traffic and not just the traffic specifically addressed to it. Of course, the traffic that a station can see varies depending on the network design, as you can't sniff what you can't see. There are two types of sniffing that can be used to observe traffic: passive and active. **Passive sniffing** takes place on networks such as those that have a **hub** as the connectivity device. With a hub in place, all stations are on the same **collision domain**, so all traffic can be seen by all other stations. In networks that have connectivity hardware that is smarter or more advanced, such as those with a **switch**, **active sniffing** is needed. For example, when a switch is in use, if traffic is not destined for a specific port, it isn't even sent to the port; therefore, there is nothing to observe.

In the Open Systems Interconnection (OSI) reference model, the sniffer functions at the data link layer. This layer is low in the hierarchy of layers, so not much "intelligence" is present (meaning that little filtering or refinement of the data is occurring). A sniffer is able to capture any and all data that happens to pass by on the wire, which even includes data that would otherwise be hidden by activities occurring at higher layers.

FYI

Before sniffing on any network, make sure you have permission from the network owner. Sniffing traffic on networks when you do not have permission to do so on can lead to serious problems up to and including legal repercussions.

According to Title 18, Section 2511 of the U.S. Code, which covers electronic crimes including those that would fall under the term "sniffing," the act of sniffing would be defined as

"Interception and disclosure of wire, oral, or electronic communications prohibited

(a) intentionally intercepts, endeavors to intercept, or procures any other person to intercept or endeavor to intercept, any wire, oral, or electronic communication."

Penalties for engaging in this activity can be anything from fines to civil and criminal penalties.

NOTE

Understanding the OSI reference model is an essential skill, and you should make sure to spend time reviewing and understanding the model well.

Passive Sniffing

Passive sniffing works when the traffic you wish to observe and the station that will do the sniffing are in the same collision domain. Passive sniffing works when a device known as a hub is in use. This is the key feature that makes this setup work. Think of the way a hub functions: traffic that is

MODULE 4

sent to one port on a hub is automatically sent to all ports on the hub. Because any station can transmit at any time, collisions can and do happen and can lead to a collision domain. When this type of situation exists, it is possible to listen in on traffic on the network quite easily because every station shares the same logical transmission area. What thwarts passive sniffing is a switch that separates the networks into multiple collision domains, therefore creating a situation in which stations do not transmit in the same logical area. Basically, passive sniffing is effective when the observer and the victim exist so that each can see each other's actions.

FYI

Sniffing may sound like a formidable threat to the security of information, and it definitely can be, but it can have its impact blunted to a certain degree. The answer is to use encryption for data in transit, specifically data that is of an extra-sensitive nature. The rise in usage of protocols such as Secure Sockets Layer (SSL), Internet Protocol Security (IPSec), Secure Shell (SSH), and others has made passive sniffing much less effective. Of course, you should always remember that encryption can protect information, but use it only when necessary to avoid overburdening processors on the sending and receiving systems.

The key to getting the most from passive sniffing is to plan carefully. Look for those locations on the network that will act as chokepoints for traffic, or those locations that the traffic that you are looking for will pass. Placing a sniffer on a collision domain different from the one that is to be observed will not yield the results that you desire, so placement must always be considered.

Some points to remember about passive sniffing:

- Passive sniffing is difficult to detect because the attacker does not broadcast anything on the network as a practice.
- Passive sniffing takes place and is effective when a hub is present.
- Passive sniffing can be done very simply. It can be as simple as an attacker plugging into a network hub and loading a sniffer.

Active Sniffing

So what happens if a network is broken into different collision domains using the power of switches? It would seem in these situations that the target is out of reach, but this problem can be overcome with the power of active sniffing. Because a switch limits the traffic, a sniffer can see the traffic that is specifically addressed to a system. Active sniffing is necessary to see the traffic that is not addressed to that system.

Active sniffing involves sniffing when a switch is present on the network. This technique is employed in environments where sniffing using passive methods would be ineffectual due to the presence of switches. Active sniffing requires the introduction of traffic onto the network and as such can be detected relatively easily.

MODULE 4

In order to use active sniffing, an understanding of two techniques is necessary, both of which are used to get around the limitations that switches put in place. These techniques are known as media access control (MAC) flooding and Address Resolution Protocol (ARP) poisoning, both of which are valuable tools in your arsenal.

MAC Flooding

The first technique to bypass switches is MAC flooding: the ability to overwhelm the switch with traffic designed to cause it to fail. A closer look at this attack reveals how it succeeds in its task of causing the switch to fail. Switches contain some amount of memory (known as **content addressable memory**, or **CAM**) onboard that is used to build what is called a **lookup table**, which is then used to track which MAC addresses are present on which ports on the switch. This memory allows a lookup to be performed to let the switch get traffic to the correct port and host as intended. This lookup table is built by the switch during normal operation and resides in the CAM. The goal of MAC flooding is to exploit a design defect or oversight in some switches, which is that they have only a limited amount of memory. An attacker can flood this memory with information in the form of MAC addresses and fill it up quickly until it cannot hold any more information. In the event that this memory fills up, some switches will enter a **fail-open** state.

FYI

Both MAC flooding and ARP poisoning generate some level of activity on the network and possibly on the clients themselves. This is the drawback of active sniffing: the introduction of traffic onto the network, and the fact that your presence is now detectable by anyone or anything that may be looking. Passive sniffing has the advantage of being much stealthier, as the presence of the sniffer is not as obvious due to the lack of broadcast information.

When a switch enters this fail-open state, the switch now becomes functionally a hub, and you are back to where you started with passive sniffing. By performing this attack on a switched network with a vulnerable switch, it is possible to attain a state where traffic that might not otherwise be sniffed now can be. Of course, you don't get something for nothing; in this case, the amount of traffic that is introduced on the network can make sniffing impossible, as well as send up a huge red flag to anyone or anything that may be watching for traffic anomalies.

MAC flooding involves overwhelming or flooding the switch with a high volume of requests. This technique overwhelms the memory on the switch used to map MAC addresses to ports. MAC flooding is performed by sending enough traffic through the switch that the memory and switch cannot keep up. Once CAM is overwhelmed, the switch acts like a hub.

To make this attack easy there are a diverse set of tools available for the security professional and hacker:

- **EtherFlood**—This utility has the ability to clog a switch and network with Ethernet frames with bogus, randomized hardware addresses. By flooding the network with such frames, the net effect is what is expected with MAC flooding: a switch that fails over to hub behavior.

MODULE 4

- **SMAC**—A MAC spoofing utility that is designed to change the MAC address of a system to one that the attacker specifies.
 - In modern operating systems from Windows XP forward, and in most Linux variants, this utility is not even necessary because the MAC address can be changed in the graphical user interface (GUI) or at the command line using tools bundled with the operating system (OS) itself.
- **Macof**—Designed to function like EtherFlood and overwhelm the network with bogus or false MAC addresses to cause the switch to fail to hub behavior
- **Technetium MAC Address Changer**—Designed to function much like SMAC, in that it can change the MAC address of a system to one the user desires instead

Address Resolution Protocol (ARP) Poisoning

The other method of bypassing a switch to perform sniffing is via **Address Resolution Protocol (ARP) poisoning**. Here are some key points:

- Address Resolution Protocol (ARP) is a protocol defined at the network layer which is used to resolve an IP address to a physical or MAC address.
- In order to locate a physical address, the requesting host will broadcast an ARP request to the network.
- The host that has the IP address that is sought after will return its corresponding physical address.

NOTE

If you are still unclear about the ARP process, refer to [Chapter 2](#) and the discussion on ARP and the OSI reference model.

- ARP resolves logical addresses to the physical address of an interface.
- ARP packets can be spoofed or custom crafted to redirect traffic to another system such as the attacker's.
- ARP poisoning can be used to intercept and redirect traffic between two systems on the network.
- MAC flooding can clog and overwhelm a switch's CAM, forcing it into what is known as forwarding mode.

MODULE 4

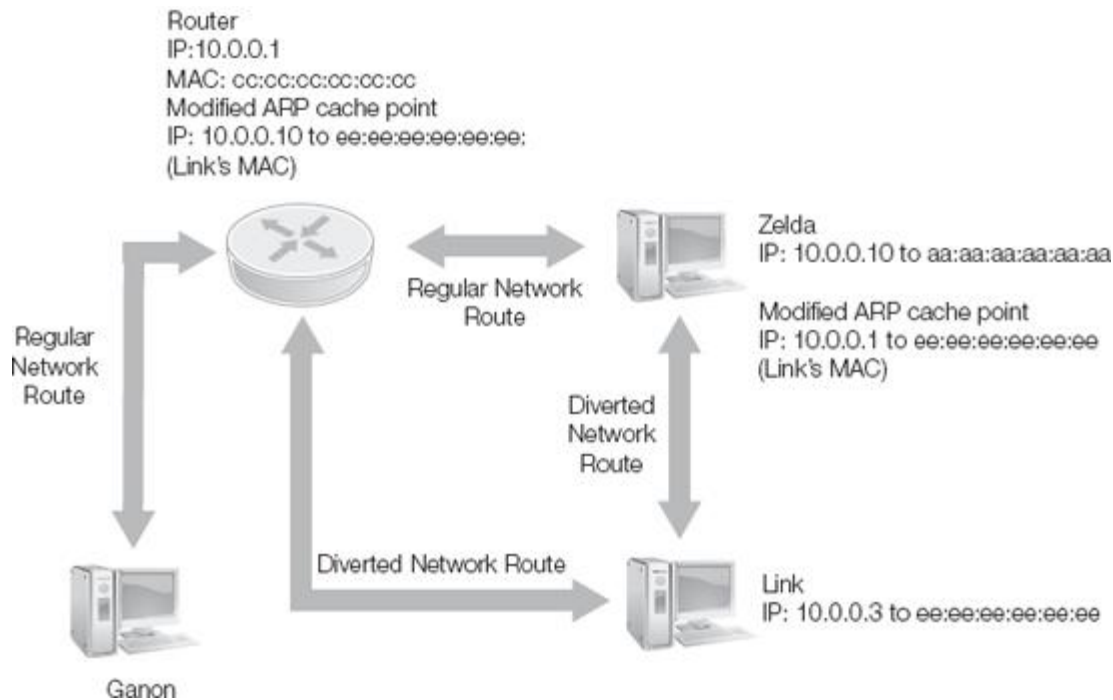


Figure 12-1. ARP poisoning in practice.

With knowledge of the ARP process in hand, it is very easy to understand the mechanics of ARP poisoning or ARP spoofing. ARP poisoning works by sending out bogus ARP requests to any requesting device and the switch. The idea is to force traffic to a location other than the intended target and therefore sniff what is being sent and received. When the bogus requests are sent out, the switch stores them. Other clients will then automatically send traffic to the new target, as they will check their cache first where the bogus entry has been stored.

[Figure 12-1](#) illustrates ARP poisoning in practice.

Here are the steps in the process:

1. Attackers send out a broadcast stating that a given IP address (such as a router or gateway) maps to their own MAC address.
2. A victim on the network initiates a communication that requires exiting the network or subnet.
3. When the traffic is transmitted, the ARP mapping shows that the router's IP address maps to a specific MAC address, so traffic is forwarded to the attacker instead.
4. To complete the sequence and avoid arousing suspicion, the attacker forwards traffic to the real destination (in this case, the router).

NOTE

Not forwarding traffic on to the original destination would arouse suspicion that would tip off the network administrator to the attacker's presence.

MODULE 4

Here are some points to remember about ARP poisoning:

- Anyone can download malicious software used to run ARP spoofing attacks from the Internet.
- Attackers can use bogus ARP messages to redirect traffic.
- It is possible to run DoS attacks with this technique.
- It can be used to intercept and read data.
- It can be used to intercept credentials such as usernames and passwords.
- It can be used to alter data in transmission.
- It can be used to tap voice over IP (VoIP) phone calls.

Several utilities in your security professional toolbox are specifically designed to carry out ARP spoofing, no matter what your OS of choice may be. The following list details some of the options available to you:

- **Arpspoof**—Designed to redirect traffic in the form of packets from a victim's system. Performs redirection by forging ARP replies. This utility is part of the popular Dsniff suite of utilities.
- **Cain**—The "Swiss army knife" of tools; can perform ARP poisoning, enumeration of Windows systems, sniffing, and password cracking
- **Ettercap**—An old but very capable protocol analyzer that can perform ARP poisoning, passive sniffing, protocol decoding, and as a packet capture
- **Internal Revenue Service (IRS)**—Not a port scanner; it is a "valid source IP address" scanner for a given service. Combines ARP poisoning and half-scan processes and attempts TCP connections to a specific victim.
- **ARPWorks**—Utility for creating customized packets over the network that perform the ARP announce feature
- **Nemesis**—Can perform some ARP spoofing

Sniffing Tools

Several very capable sniffing tools are available, including the popular ones in the following list:

- **Wireshark**—One of the most widely known and used packet sniffers. Offers a tremendous number of features designed to assist in the dissection and analysis of traffic. Wireshark is the successor to the Ethereal packet sniffer.
- **Tcpdump**—A well-known command line packet analyzer. Provides the ability to intercept and observe TCP/IP and other packets during transmission over the network.
- **Windump**—A port of the popular Linux packet sniffer known as TCPdump, which is a command line tool that is great for displaying header information. TCPdump is available at <http://www.tcpdump.org>.
- **Omnipeek**—Manufactured by Wildpackets, Omnipeek is a commercial product that is the evolution of the product Etherpeek.

MODULE 4

- **Dsniff**—A suite of tools designed to perform sniffing with different protocols with the intent of intercepting and revealing passwords. Dsniff is designed for UNIX and Linux platforms and does not have a complete equivalent on the Windows platform.
- **Etherape**—A Linux/UNIX tool that is designed to graphically display the connections incoming and outgoing from a system
- **MSN Sniffer**—A sniffing utility specifically designed for sniffing traffic generated by the MSN messenger application
- **Netwitness Nextgen**—A hardware-based sniffer, plus other features, designed to monitor and analyze all traffic on a network; a popular tool in use by the FBI and other law enforcement agencies

What Can Be Sniffed?

With this powerful technique, an attacker can reveal a wealth of information that can be used against you as a defender, but this information does not have to be accessible to an attacker, because it takes just a little care to take the teeth from these attacks. In this section you will learn some of the techniques that can be used to limit or block the effects of sniffing.

NOTE

Not all traffic needs to be protected, and it may not even be feasible to do so. Remember that all extra countermeasures that are deployed are extra devices and processes to support and are extra overhead

To defeat sniffing, a number of countermeasures can be employed, including the following:

- **Encryption**—Protecting traffic from being sniffed can be as simple as making it undecipherable to those not having the key. Encrypting select data through the use of technologies such as IPSec, SSL, virtual private networks (VPNs), and other related techniques can be a simple but effective way of thwarting sniffing. The downside here is that the process of encryption costs in processor power and performance.
- **Static ARP entries**—Configuring a device with the MAC addresses of the devices that may use it can block a number of attacks, but can be difficult to manage.
- **Port security**—Switches have the ability to be programmed to allow only specific MAC addresses to send and receive data on each port.

When considering network security and thwarting the power of sniffing, you should consider which protective measures are appropriate and which are not. In the case of encryption, for example, not all traffic needs to be encrypted because not all network traffic is of a sensitive nature. Always consider the exact nature of the traffic, too. Remember, just because you can do something does not mean you should.

Session Hijacking

MODULE 4

The next type of attack that can be used to alter and interrupt communications on a network is the technique known as **session hijacking**. Hijacking a session falls under the category of active attacks in that you must directly and somewhat aggressively interact with the network and the victims on it. Hijacking builds on the techniques discussed in our previous section of sniffing and raises the stakes by taking over the communication between two parties. Once attackers decide to undertake a session hijacking, they will be actively injecting packets into the network with the goal of disrupting and taking over an existing session on the network. Ultimately the session hijack will attempt to take over a session that is already authenticated to a resource to be attacked.

Here's a high-level view of what session hijacking looks like:

1. Insert yourself between Party A and Party B.
2. Monitor the flow of packets using sniffing techniques.
3. Analyze and predict the sequence number of the packets.
4. Sever the connection between the two parties.
5. Seize control of the session.
6. Perform packet injection into the network.

To summarize, session hijacking is the process of taking over an already established session between two parties. Some points to remember about session hijacking:

- TCP session hijacking is in process when an attacker seizes control of an existing TCP session between two systems.
- Session hijacking takes place after the authentication process that occurs at the beginning of a session. Once this process has been undertaken, the session can be hijacked, and access to the authenticated resources can take place.
- Session hijacking relies on a basic understanding of how messages and their associated packets flow over the Internet.

Session hijacking, much like sniffing, has two forms: active and passive. Each form of session hijacking has its advantages and disadvantages that make it an attractive option to the attacker. Let's compare and contrast the two to see what they offer an attacker.

- **Active session hijacking**—Active attacks are effective and useful to the attacker because they allow the attacker to search for and take over a session at will. In **active session hijacking**, the attacker will search for and take over a session and then interact with the remaining party as if the attacker were the party that has been disconnected. The attacker assumes the role of the party he has displaced, in other words.
- **Passive session hijacking**—Passive attacks are different in that the attacker locates and hijacks a session of interest, but does not interact with the remaining party. Instead, in **passive session hijacking**, attackers switch to an observation type mode where they record and analyze the traffic as it moves. Passive hijacking is functionally no different from sniffing.

Identifying an Active Session

MODULE 4

Earlier, when sniffing was discussed, the process was that of observing traffic on the network. Session hijacking builds on this process and refines it. Session hijacking adds the goal of not only observing the traffic and sessions currently active on the network but also taking over one of these sessions that has authenticated access to the resource you want to interact with. For a session hijack to be successful, the attacker must locate and identify a suitable session for hijacking. It sounds like a simple process until factors such as different network segments, switches, and encryption come into play. If you factor in the very real issue of having to uncover sequence numbers on packets in order to properly take control of a session, the challenges mount significantly. But they are not insurmountable. Remember that while the challenges are not small, what is on the line is the ability to interact with and execute commands against authenticated resources.

NOTE

Session hijacking builds on the techniques and lessons learned in passive and active sniffing so you may want to review those lessons again if you are not completely clear on them. Session hijacking takes sniffing and moves these lessons to the next level where you move from listening to interacting, which is more aggressive by nature.

Consider some of the challenges standing in the way of successful session hijacking:

- **Sequence numbers**—Every packet has a unique 32-bit number embedded into its header that identifies it and how it should be reassembled with its fellow packets to regenerate the original message.
- **Network segments**—When the attacker and victims are on the same network segment or on a network that uses a hub, observing traffic works like basic sniffing. However, if the victim and the attacker are on two different network segments separated by a switch, it becomes more difficult to carry out an attack, and techniques akin to the active sniffing techniques are needed.

Take a look at the sequence number problem. Let's review the steps involved in session hijacking once again:

1. Insert yourself between Party A and Party B.
2. Monitor the flow of packets using sniffing techniques.
3. Analyze and predict the sequence number of the packets.
4. Sever the connection between the two parties.
5. Seize control of the session.
6. Perform packet injection into the network.

NOTE

In the past, some operating systems did allow for the methodical and mathematical creation of sequence numbers. This was possible because these operating systems implemented very predictable sets of sequence numbers. Most operating systems now avoid this by randomly generating sequence numbers as a security measure.

MODULE 4

Look at Step 3—this step is easy on a network on which you can see both parties. On these types of networks you can sniff the traffic passively and read the sequence numbers off of the packets themselves. On a switched network, it becomes much more of an issue because you cannot see the other party(ies) so you must use techniques to guess the sequence number correctly (you can't just stumble in with whatever number you want). In this situation, you will send several packets to the victim or target in order to solicit a response with the sequence numbers on it.

Sequence numbers are a cornerstone of TCP that makes a number of features that you may take for granted possible. In TCP every piece or byte of data must have a sequence number assigned to it to track the data, assemble it with its fellow packets, and perform flow control. So where and when do the sequence numbers get assigned? During the three-way handshake, which is illustrated in [Figure 12-2](#).

FYI

Some facts about sequence numbers:

- Sequence numbers are a 32-bit counter. The possible combinations can be more than 4 billion
- Sequence numbers are used to tell the receiving machine what order the packets should go in when they are received.
- An attacker must successfully guess the sequence numbers in order to hijack a session.

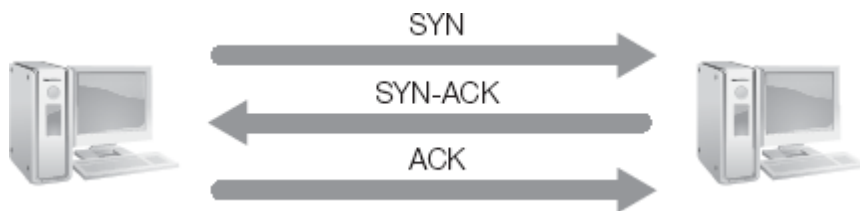


Figure 12-2. Three-way handshake.

Here are some points to bear in mind about sequence number prediction:

- When a client transmits a SYN packet to a server the response will be a SYN-ACK. This SYN-ACK will be responded to with an ACK.
 - During this handshake, the starting sequence number will be assigned using a random method if the operating system supports this function.
- If this sequence number is predictable, the attacker will initiate the connection to the server with a legitimate address and then open up a second connect from a forged address.

Once an attacker has determined the correct sequence numbers, the next move is to inject packets into the network. Of course, this is easier said than done, and just injecting packets into the network is not useful in every case because a few details must be in place first. Consider the two extremes of the session: the beginning and the end. At the beginning of the session, the process of authentication takes place, and injecting packets into the network and taking over the session here

MODULE 4

would be worthless if done prior to the authentication process (after all, you want an authenticated session). On the other hand, injecting packets too late, such as when the session is getting torn down or closed, will mean that the session you want to hijack is no longer present.

NOTE

You must wait for authentication to take place prior to taking over a session because without doing so you don't have trust, and in this case the system you are trying to interact with has no knowledge of you.

With the proper sequence numbers predicted and known the attack can move to the next phase which is to unplug one of the parties, such as a server if one is present. The goal at this stage is to knock out or remove one of the parties from the communication in order to get them out of the way. The removal can be performed by any method the attacker chooses, from a simple DoS to sending a connection reset request to the victim.

Seizing Control of a Session

At this point, the attacker now has control of a session and can move toward carrying out dirty work, whatever it may be. The trick for the attacker is to keep the session maintained and active because as long as this connection is maintained and kept alive, the attacker has an authenticated connection to their intended target.

Session Hijacking Tools

In order to perform session hijacking you can use a number of different tools, each having its own advantages and disadvantages. Each of the tools on this list has seen widespread use by hackers and will offer you the ability to perform session hijacking quite easily. Each of these tools is essentially a packet sniffer with the enhanced capability needed to perform session hijacking.

- **Ettercap**—An old-school tool that has the advantage of being multiplatform so you can learn how to use it on one platform and move those skills over easily to another platform such as Mac OS X. Ettercap possesses robust capabilities that enable it to perform its duties quite well. Included in this functionality is the ability to perform man-in-the-middle attacks, ARP spoofing, and session hijacking.
- **Hunt**—This is a commonly used tool for performing session hijacking; in fact, it is the first one most hackers and security professionals are introduced to. This software has the ability to observe and hijack a session between two parties, and also has the ability to fire off TCP resets to shut down a victim system. This software package is designed to work on Ethernet-based networks and can work in both passive and active modes.
- **IP Watcher**—This utility is a commercial-grade tool (read: you have to pay for it) that can perform session hijacking and monitor connections so you can choose the session you wish to take over.
- **T-Sight**—Another commercial offering that can hijack TCP sessions on a network much like IP watcher

MODULE 4

- **Remote TCP Reset**—Is designed to find and reset an existing TCP connection

Thwarting Session Hijacking Attacks

Session hijacking is dangerous. But you can limit its impact to a great degree through the proper application of your two best lines of defense: being proactive and looking for the signs of an attack. One of your tools for this is something you read about earlier: encryption. After all, it is hard for troublemakers to hijack a session if they can't see what is being transmitted. Other measures you can use include configuring routers to block spoofed traffic from outside the protected network. Additionally, you can use counter-measures such as an intrusion detection system (IDS) that can watch for suspicious activity and alert you to it, or even actively block this traffic automatically.

Denial of Service (DoS) Attacks

An older type of attack that still plagues the Internet and the computer systems attached to it is the DoS, which is a threat against one of the core tenets of security: availability. This makes sense when you consider that a DoS is designed to target a service or resource, and deny access to it by legitimate users. In this section, you will take a look at this simple form of hacking: what it can do as well as how it works.

A DoS functions by tying up valuable resources that could be used to service legitimate needs and users. In essence, a DoS functions like this: Imagine someone calling your cell phone over and over again; at some point they call often enough that no one else could call you nor could you call out. At that point you would become the victim of a DoS. Translate this scenario into the world of computer networks, and you have a situation where availability of a service is similarly threatened.

NOTE

DoS attacks are commonly used by those who fall in the category of script kiddies due to the relative simplicity of the attack. Don't be lulled into a false sense of security, however, as more advanced hackers have been known to use this attack as a last resort (as a way of shutting down a service that they were unable to get access to).

DoS attacks used to be used to annoy and irritate a victim, but over the past few years these attacks have evolved into something much more ominous: a means to extort money and commit other crimes. For example, a criminal may contact a victim and ask for protection money to prevent any unfortunate "accidents" from happening.

NOTE

The use of DoS to extort money has increased over the past few years as criminals have become more adept at using technology.

To summarize, the main points of a DoS action are to:

MODULE 4

- Deny the use of a system or service through the systematic overloading of its resources. An attacker is seeking a result in which the system becomes unstable, substantially slower, or overwhelmed to the point it cannot process any more requests.
- Be carried out when an attacker fails at other attempts to access the system and just decides to shut down a system in retaliation

Categories of DoS Attacks

DoS attacks are not all the same. They can be broken down into three broad categories based on how they carry out their goal of denying the service to legitimate uses and users:

- Consumption of bandwidth
- Consumption of resources
- Exploitation of programming defects

Consumption of Bandwidth

Bandwidth exhaustion is one of the more common attacks to be observed in the wild. This type of attack is in effect when the network bandwidth flowing to and from a machine is consumed to the point of exhaustion. It may seem to some that the solution here would be to add enough bandwidth that it cannot be easily exhausted, but the keyword is "easily" exhausted—it does not matter how much bandwidth is allocated to a system; it is still a finite amount. In fact, an attacker does not have to completely exhaust bandwidth to and from a system, but rather use up so much of it that performance becomes unacceptable to users. So the attacker's goal is to consume enough bandwidth to make the service unusable.

Some well-known forms of attacks in this category include:

- **Smurf**—Through the exploitation of the Internet Control Message Protocol (ICMP) and spoofed packets to the broadcast address of a network, the attacker can generate a torrent of traffic from the sheer number of systems that may reply.
- **Fraggle**—This type of attack is similar to the smurf attack with the difference being what it uses to consume bandwidth. In the case of fraggle attacks, bandwidth is consumed through the use of User Datagram Protocol (UDP) packets instead.
- **Chargen**—This protocol was originally designed for testing and evaluation purposes, but it can be used to perform a DoS by generating traffic rapidly. By doing so, chargen can consume the bandwidth on a network rapidly, at which point a DoS will have occurred.

Consumption of Resources

Much like bandwidth consumption, the goal of resource consumption-based attacks is to eat up a limited resource. However, unlike bandwidth consumption, the goal is not shared among multiple systems; instead it is targeting the resources on a single system. When an attack of this nature is carried out, a service or an entire system may become overloaded to the point where it slows, locks, or crashes.

MODULE 4

This type of attack can vary in how it is approached; the following list is some of the more common forms of this attack:

- **SYN flood**—This type of attack uses forged packets with the SYN flag set. When the victim receives enough of the packets, the result is an overwhelmed system as the SYN flood consumes connection resources to the point where no resources are available for legitimate connections.
- **ICMP flood**—This type of attack comes in two variants: smurf attack and ping flood.
 - **Smurf attack**—Carried out when a large amount of traffic is directed to the broadcast address of a network instead of to a specific system. By sending traffic to a broadcast address of a network, the request is sent to all hosts on the network, which respond in turn. However, because the attacker will take the extra step of configuring the packet with the intended victim as the source, all the hosts on the network will respond to the victim instead of to the attack. The result is that a flood of traffic overwhelms the victim causing a DoS.
 - **Ping flood**—Carried out by sending a large amount of ping packets to the victim with the intent of overwhelming the victim. This attack is incredibly simple, requiring only basic knowledge of the ping command, the victim's IP, and more bandwidth than the victim. In Windows, the command to pull off such an attack would be:

```
ping -t <victim IP address>
```

- **Teardrop attack**—In this type of attack, the attacker manipulates IP packet fragments in such a way that when reassembled by the victim, a crash occurs. This process involves having fragments reassembled in illegal ways or having fragments reassembled into larger packets than the victim can process.
- **Reflected attack**—This type of attack is carried out by spoofing or forging the source address of packets or requests and sending them to numerous systems, which in turn respond to the request. This type of attack is a scaled-up version of what happens in the ping flood attack.

Exploitation of Programming Defects

Consuming bandwidth isn't the only way to carry out a DoS attack on a system. Another is to exploit known weaknesses in the system's design. Vulnerabilities of this type may have been exposed due to flaws in the system's design that were inadvertently put in place by the programmers or developers of the system.

The following list has some of the more common methods of exploiting programming defects:

- **Ping of death (PoD)**—This type of attack preys upon the inability of some systems to handle oversized packets. An attacker sends them out in fragments; when these fragments reach the system they are reassembled by the victim, and when the "magic size" of the 65,536 bytes allowed by the IP protocol is reached, some systems will crash or become victim to a buffer overflow.

MODULE 4

- **Teardrop**—This attack succeeds by exploiting a different weakness in the way packets are processed by a system. In this type of attack, the packets are sent in a malformed state with their offset values adjusted so they overlap, which is illegal. When a system that does not know how to deal with this issue is targeted, a crash or lock may result.
- **Land**—In this type of attack, a packet is sent to a victim system with the same source and destination address and port. The result of this action is that systems that do not know how to process this crash or lock up.

NOTE

All these attacks have been around for years and so you would expect systems to be designed to be less susceptible to them. However, this is not the case. It has been discovered time and time again that modern systems from all vendors can be vulnerable to these attacks if they are not patched and managed correctly.

Tools for DoS

There are plenty of tools available to the hacker to perform a DoS attack, including:

- **Jolt2**—A piece of software designed to flood a system with incorrectly formatted packets
- **Targa**—This software is designed to attempt different types of attacks and has eight different variations to choose from.
- **Crazy Pinger**—This software is designed to send ping packets of varying sizes and other parameters to a victim.

NOTE

Some of these tools have been known to appear on systems seemingly inexplicably, which may be a sign of a system that has become part of a botnet, which will be discussed later in this chapter.

FYI

Do not be confused—DoS and DDoS attacks are as similar as they are indeed different. The two share some traits, but vary in others. The two attacks both seek to overwhelm a victim with requests designed to lock up, slow down, or crash a system. The difference is in implementation as DoS is generally one system attacking another, and DDoS is many systems attacking another. It could be said the difference is scale.

Distributed Denial of Service (DDoS) Attacks

A distributed denial of service (DDoS) attack is a powerful tool for those who know how to use it. Security professionals have developed techniques to prevent these attacks, but hackers keep developing new methods of carrying them out.

MODULE 4

Some Characteristics of DDoS Attacks

As you can readily imagine, a distributed attack, involving many compromised machines, is a more devastatingly effective way to commit a denial of service attack than simply using one machine to attack another. Here are some specifics you should know:

- Attacks of this type are characterized by being very large, using hundreds or thousands of systems to conduct the attack.
- DDoS has two types of victims; namely, primary and secondary. The former is the recipient of the actual attack; the latter are the systems used to launch the attack itself.
- The attack can be very difficult if not impossible to track back to its true source because of the sheer number of systems involved.
- Defense is extremely difficult due to the number of attackers. Configuring a router or firewall to block a small number of single IP addresses is child's play. Larger numbers of attackers are nearly impossible to block.
- Impact of this attack is increased over standard DoS because many hosts are involved in the attack, multiplying the attack's strength and power.

A DDoS is an "upgraded" and advanced version of the DoS. The DDoS has the same goal as the DoS, which is to shut a system down by consuming resources, but does so through sheer force of numbers. This type of attack generally tends to occur in two waves designed to position and carry out the attack.

In the first wave, the attack is staged, and the targets that will be the "foot soldiers" are infected with the implements that will be used to attack the final victim. Targets for infection in this phase include systems that have high-speed connections, poorly defended home and business networks, and poorly patched systems. What is infecting these systems can and will vary, but it could include software such as the ones mentioned previously for a traditional DoS.

Wave 2 is the attack itself. Foot soldiers form the army of systems that will collectively attack a designated target. These infected systems can number in the thousands, hundreds of thousands, or even millions awaiting the instruction that will turn their collective attention toward a target (these infected systems are called "zombies"). These are the steps of the attack itself:

- Construct a piece of malware that will transmit packets to a target network/Web site.
- Convert a predefined number of computers to drones.
- Initiate the attack by sending signals to the drones to attack a specific target.
- Have drones initiate an attack against a target until they are shut down or disinfected.

NOTE

The infected systems are not always referred to as "zombies"; they are sometimes called "bots" (short for robots) or, like the Borg in Star Trek, "drones." Whatever you call them, the goal is the same: to target a system and steamroll it with traffic.

MODULE 4

A DDoS attack like this sounds simple, but in practice it is not, because it takes quite a bit of planning and knowledge to set up, not to mention a good amount of patience. To set this type of attack up, two components are needed: a software component and a hardware component.

On the software side, two items are needed to make the attack happen:

- **Client-side software**—This is the software that ultimately will be used to send command and control requests to launch an attack against the target. This software will be used by the attacker to initiate the opening stages of the attack.
- **Daemon software**—This software is resident on the infected systems or bots. This software is installed on a victim and then waits for instructions to be received. If you have software of this type installed, you are the one actually attacking a system.

The second requirement that is essential is the hardware; more specifically, these are the systems that will be components of the attack:

- **Master or control system**—The system responsible for sending out the initial messages to start the attack; also the system that has the client software present and installed
- **Zombie**—The system that is the one carrying out the attack against the victim. The number of zombies can vary wildly in number.
- **Target**—The system that is the actual victim or recipient of the attack

You may be wondering whether, all things considered, a DDoS is unstoppable.

DDoS attacks rely on locating and using vulnerable hosts that are connected to the Internet. These systems are then targeted for these known vulnerabilities and taken over. Once the attack is initiated and the command sent out to the attackers, the DDoS is nearly impossible to stop.

Routers and firewalls may be configured to block the attack, but the attack can overwhelm these devices and shut down the connection anyway. The sheer volume of attackers involved in DDoS attacks makes them difficult to stop.

Tools for DDoS

To initiate a DDoS requires the proper tools, and there are a number available. The tool or tools you use will ultimately depend on what your preferences are as well as other factors such as platform, but the following list is a sampling of these tools:

- **Tribal Flood Network (TFN)**—TFN can launch ICMP, Smurf, UDP, and SYN flood attacks at will against an unsuspecting victim. TFN has the distinction of being the first publicly available DDoS tool.
- **Trinoo**—Trinoo can claim to be the first widely used DDoS application largely because it is easy to use and has the ability to command and control many systems to launch an attack.
- **Stacheldraht**—The best of both worlds is available in this tool, which offers features that are seen both in Trinoo and TFN. Stacheldraht uses TCP and ICMP to send commands and

MODULE 4

control its agents in order to attack. This software also includes what could be considered advanced features in the form of encrypted communication from client to handlers.

- **TFN2K**—An upgrade to TFN, it provides some more advanced features including spoofing of packets and port configuration options. As opposed to TFN, this software does include encryption features, but not as strong as those of Stacheldraht.
- **WinTrinoo**—This software is a Windows port of Trinoo and has the ability to use Windows clients as drones.
- **Shaft**—This works much the same way as Trinoo, but includes the ability for the client to configure the size of the flooding packets and the duration of attack.
- **MStream**—This utilizes spoofed TCP packets to attack a designated victim.
- **Trinity**—This performs several DDoS functions, including fraggle, fragment, SYN, RST, ACK, and others.

Botnets

An advanced type of attack mechanism is a **botnet**, which consists of systems that are infected with software such as those used in DDoS attacks. When enough of these systems are infected, and a critical mass has been reached, it is possible to use these machines to do tremendous damage to a victim. Botnets can stretch from one side of the globe to another and be used to attack a system or carry out a number of other tasks.

Botnets can perform several attacks, including:

- **DDoS**—This construct makes sense as an attack method based on the way a DDoS works and the number of systems that can be infected.
- **Sending**—Botnets have been used to transmit spam and other bogus information on behalf of their owner.
- **Stealing information**—Attacks have also been carried out with botnets to steal information from unsuspecting users' systems.
- **Clickfraud**—This attack is where the attacker infects a large number of systems with the idea that they will use the infected systems to click on ads on their behalf, generating revenue for themselves.

NOTE

Remember that a botnet can easily number into the hundreds of thousands or millions of systems, stretching from one end of the globe to another. With these kinds of numbers, the attacks noted here take on a new meaning and destructive capability.

A "bot" is a type of malware that allows an attacker to take control over an affected computer. Also known as "Web robots," bots are usually part of a network of infected machines known as a "botnet," which is typically made up of victim machines that stretch across the globe.

FYI

MODULE 4

The following is a clipping from an FBI news briefing:

... the Department of Justice and FBI announced the results of an ongoing cyber crime initiative to disrupt and dismantle "bot-herders" and elevate the public's cybersecurity awareness of botnets. OPERATION BOT ROAST is a national initiative. Ongoing investigations have identified over 1 million victim computer IP addresses." <http://www.fbi.gov/pressrel/pressrel07/botnet061307.htm>

CHAPTER SUMMARY

This chapter focused on three types of network attacks: sniffing, session hijacking, and DoS attacks. Each of these attacks represents a powerful weapon in the hands of a skilled attacker.

Sniffing is the process of capturing and analyzing traffic in an effort to observe information that is confidential. Sniffing can be performed on just about any network, but the technique may require that you adapt based on how the network operates. In networks with a hub, you can easily sniff using any packet sniffer and starting the process. On networks that use switches, however, it is different as the switch prevents you from seeing what is on a different collision domain. On networks where switching is used, you will have to use techniques such as MAC flooding and ARP spoofing to bypass the switch prior to sniffing.

Moving beyond or building upon the techniques that were introduced in sniffing is the session hijack, which is an aggressive and powerful weapon in the hacker's arsenal. A session hijack takes over an existing authenticated session and uses it to monitor or manipulate the traffic, and even execute commands on a system remotely. Session hijacking in its most advanced stages directly affects and attacks the integrity of information in an organization. An attacker using this technique can modify information at will as they have the credentials of the victim and whatever the victim has access to.

DoS attacks were discussed and you learned how these attacks are used to shut down and deny legitimate access to and usage of services to users. A DoS is used to target a service or system and prevent it from being used for legitimate uses for as long as the attacker wishes. Under the right conditions, a DoS directly attacks the confidentiality and integrity of data that users have been granted the right to use.

KEY CONCEPTS AND TERMS

- **Active session hijacking**
- **Active sniffing**
- **Address Resolution Protocol (ARP) poisoning**
- **Botnet**
- **Collision domain**
- **Content addressable memory (CAM)**
- **Fail-open**
- **Hub**

MODULE 4

- **Lookup table**
- **Passive session hijacking**
- **Passive sniffing**
- **Promiscuous mode**
- **Session hijacking**
- **Switch**

CHAPTER 12 ASSESSMENT

1. A DoS is meant to deny a service from legitimate usage.
 1. True
 2. False
2. Sniffers can be used to:
 1. Decrypt information
 2. Capture information
 3. Hijack communications
 4. Security enforcement
3. Session hijacking is used to capture traffic.
 1. True
 2. False
4. Session hijacking is used to take over an authenticated session.
 1. True
 2. False
5. Active sniffing is used when switches are present.
 1. True
 2. False
6. _____ is used to overwhelm a service.
7. _____ is used to flood a switch with bogus MAC addresses.
8. _____ is used to fake a MAC address.
 1. Spoofing
 2. Flooding
 3. Poisoning
 4. Hijacking
9. What type of device can have its memory filled up when MAC flooding is used?
 1. Hub
 2. Switch
 3. Router
 4. Gateway
10. What technique is used when traffic is captured on a network with hubs?
 1. Active sniffing
 2. Passive sniffing
 3. MAC Flooding
 4. Ether flooding

table of contentssearch

MODULE 4

Settings

11. Trojans and Backdoors

12. Sniffers, Session Hijacking, and Denial of Service Attacks
queue

Hacker Techniques, Tools, and Incident Handling

13. Linux, Live CDs, and Automated Assessment Tools

Table of contents collapsed