Complete coverage of all exam objectives

Ideal as both a study tool and an on-the-job reference

Filled with practice exam questions and in-depth explanations

WM. ARTHUR CONKLIN, Ph.D.  GREGORY WHITE, Ph.D.
CompTIA Security+, CISSP®
Malware and Indicators of Compromise

In this chapter, you will

- Examine the types of malware
- Understand the different types of malicious software that exist, including viruses, worms, Trojan horses, logic bombs, and rootkits
- Learn how artifacts called indicators of compromise can tell you if a system has been attacked

There are various forms of malicious software, software that is designed to compromise an end system, leaving it vulnerable to attack. In this chapter we examine the various types of malware (malicious software) and indicators of compromise that demonstrate a system has been attacked.

Certification Objective: This chapter covers CompTIA Security+ exam objective 1.1. Given a scenario, analyze indicators of compromise and determine the type of malware. This is a performance-based question testable objective, which means expect a question in which one must employ the knowledge based on a scenario. The best answer to a question will depend upon details in the scenario, not just the question. The question may also involve tasks other than just picking the best answer from a list. Instead, choices, such as order things on a diagram, rank order answers, match two columns of items, may be found.

Malware

Malware refers to software that has been designed for some nefarious purpose. Such software can be designed to cause damage to a system, such as by deleting all files, or it can be designed to create a backdoor in the system to grant access to unauthorized individuals. Generally the installation of malware is done so that it is not obvious to the authorized users. Several different types of malicious software can be used, such as viruses, Trojan
In May of 2017, a crypto-worm form of malware, WannaCry, was released, resulting in a ransomware attack that swept across many government computers in Europe, including medical devices in England’s National Health Service (NHS). This ransomware created havoc by exploiting a vulnerability in Microsoft Windows systems that was exposed by the group known as Shadow Brokers.

**Ransomware**

Ransomware is a form of malware that performs some action and extracts ransom from a user. A current ransomware threat, first appearing in 2015, is Cryptolocker. Cryptolocker is a Trojan horse that will encrypt certain files using RSA public key encryption. When the user attempts to get the files, they are provided with a message instructing them how to purchase the decryption key. Because the system is using 2048-bit RSA encryption, brute force decryption is out of the realm of recovery options. RSA encryption is covered in more detail in Chapter 27. The system is highly automated and users have a short time window to get the private key. Failure to get the key will result in the loss of the data.

**EXAM TIP** Crypto-malware and ransomware are both new to the Security+ objectives. Adding these attack vectors and how to differentiate them from other attacks to your knowledgebase will be useful for the exam.

**Worm**

It was once easy to distinguish between a worm and a virus. Recently, with the introduction of new breeds of sophisticated malicious code, the distinction has blurred. *Worms* are pieces of code that attempt to penetrate networks and computer systems. Once a penetration occurs, the worm will create a new copy of itself on the penetrated system. Reproduction of a worm thus does not rely on the attachment of the virus to another piece of code or to a file, which is the definition of a virus.

Viruses were generally thought of as a system-based problem, and worms were network-based. If the malicious code is sent throughout a network, it may subsequently be called a worm. The important distinction, however, is whether the code has the ability to infect something else (a virus) or if it can "survive" on its own (a worm).

Some examples of worms that have had high profiles include the Sobig worm of 2003, the SQL Slammer worm of 2003, the 2001 attacks of Code Red and Nimba, and the 2005 Zotob worm that took down CNN Live. Nimba was particularly impressive in that it used five different methods to spread: via e-mail, via open network shares, by accessing infected websites, using the directory traversal vulnerability of Microsoft IIS 4.0/5.0, and most impressively through the use of backdoors left by Code Red II and hardened worms.

**EXAM TIP** Worms act like a virus but also have the ability to travel without human action.
Trojan

A Trojan horse, or simply Trojan, is a piece of software that appears to do one thing (and may, in fact, actually do that thing) but hides some other functionality. The analogy to the famous story of antiquity is very accurate. In the original case, the object appeared to be a large wooden horse, and in fact it was. At the same time, it hid something much more sinister and dangerous to the occupants of the city of Troy. As long as the horse was left outside the city walls, it could cause no damage to the inhabitants. It had to be taken in by the inhabitants, and it was inside that the hidden purpose was activated. A computer Trojan works in much the same way. Unlike a virus, which reproduces by attaching itself to other files or programs, a Trojan is a stand-alone program that must be copied and installed by the user—it must be "brought inside" the system by an authorized user. The challenge for the attacker is enticing the user to copy and run the program. This generally means that the program must be disguised as something that the user would want to run—a special utility or game, for example. Once it has been copied and is inside the system, the Trojan will perform its hidden purpose with the user often still unaware of its true nature.

A good example of a Trojan is Back Orifice (BO), originally created in 1999 and now offered in several versions. BO can be attached to a number of types of programs. Once it is attached, and once an infected file is run, BO will create a way for unauthorized individuals to take over the system remotely, as if they were sitting at the console. BO is designed to work with Windows-based systems. Many Trojans communicate to the outside through a port that the Trojan opens, and this is one of the ways Trojans can be detected.

**EXAM TIP** Ensure you understand the differences between viruses, worms, Trojans, and various other types of threats for the exam.

Rootkit

Rootkits are a form of malware that is specifically designed to modify the operation of the operating system in some fashion to facilitate nonstandard functionality. The history of rootkits goes back to the beginning of the UNIX operating system, where rootkits were sets of modified administrative tools. Originally designed to allow a program to take greater control over operating system function when it fails or becomes unresponsive, the technique has evolved and is used in a variety of ways. One high-profile case occurred at Sony BMG Corporation, where rootkit technology was used to provide copy protection technology on some of the company’s CDs. Two major issues led to this being a complete debacle for Sony: first, the software modified systems without the users’ approval; and second, the software opened a security hole on Windows-based systems, creating an exploitable vulnerability at the rootkit level. This led the Sony case to be labeled as malware, which is the most common use of rootkits.

A rootkit can do many things—in fact, it can do virtually anything that the operating system does. Rootkits modify the operating system kernel and supporting functions, changing the nature of the system’s operation. Rootkits are designed to avoid, either by subversion or evasion, the security functions of the operating system to avoid detection. Rootkits act as a form of malware that can change thread priorities to boost an application’s performance, perform keylogging, act as a sniffer, hide other files from other applications, or create backdoors in the authentication system. The use of rootkit functionality to hide other processes and files enables an attacker to use a portion of a computer without the user or other applications knowing what is happening. This hides exploit code from antivirus and anti-spyware programs, acting as a cloak of invisibility.

Rootkits can load before the operating system loads, acting as a virtualization layer, as in SisVirt and Blue Pill. Rootkits can exist in firmware, and these have been demonstrated in both video cards and expansion cards. Rootkits can exist as loadable library modules, effectively changing portions of the operating system outside the kernel. Further information on specific rootkits in the wild can be found at www.antirootkit.com.

**EXAM TIP** Five types of rootkits exist: firmware, virtual, kernel, library, and application level.

Once a rootkit is detected, it needs to be removed and cleaned up. Because of rootkits’ invasive nature, and the fact that many aspects of rootkits are not easily detectable, most system administrators don’t even attempt to clean up or remove a rootkit. It is far easier to use a previously captured clean system image and reimage the machine than to attempt to determine the depth and breadth of the damage and attempt to fix individual files.

Keylogger

As the name suggests, a keylogger is a piece of software that logs all of the keystrokes that a user enters. Keyloggers in their own respect are not necessarily evil, for you could consider Microsoft Word to be a keylogger. What makes a keylogger a malicious piece of software is when its operation is 1) unknown to the user, and 2) not under the user’s control. Keyloggers have been marketed for a variety of uses, from surveillance over your children’s activity, to that of a spouse, to maintaining records of what has been done on a machine. Malicious keyloggers have several specific characteristics: they are frequently hidden from the user’s view, even when you look at task managers and they are used against the end-user’s interests. Hackers use keyloggers to obtain passwords and other sensitive pieces of information, enabling them to use these secrets to act as the user without the user’s consent. Keylogger functionality has even been found in legitimate programs, where keystrokes are recorded for “legitimate” purposes and then are stored in a fashion that enables unauthorized users to steal the data.

Adware

The business of software distribution requires a form of revenue stream to support the cost of development and distribution. One form of revenue stream is advertising. Software that is supported by advertising is called adware. Adware comes in many different forms. With legitimate adware, the user is aware of the advertising and agrees to the
This frequently involves the use of specially configured communication protocols that are set up upon initial infection of the target computer. This backdoor into the target machine can allow an attacker unfettered access, including the ability to monitor user behavior, change computer settings, browse and copy files, access connected systems, and more. RATs are commonly employed by the more skilled threat actors, although there are RATs that are easy enough for even beginners to employ.

A RAT should be considered as another form of malware, but rather than just being a program, it has an operator behind it, guiding it to do even more persistent damage. RATs can be delivered via phishing e-mails, watering holes, or any of a myriad of other malware infection vectors. RATs typically involve the creation of hidden file structures on a system and are vulnerable to detection by modern anti-malware programs. There are several major families of RATs, but an exhaustive list would be long and ever increasing. When facing a more skilled adversary, it is not uncommon to find RAT packages that have been modified for specific use, such as the program used in the Ukraine electric grid attack in 2015.

Logic Bomb

Logic bombs, unlike viruses and Trojans, are a type of malicious software that is deliberately installed, generally by an authorized user. A logic bomb is a piece of code that sits dormant for a period of time until some event or date invokes its malicious payload. An example of a logic bomb might be a program that is set to load and run automatically, and that periodically checks an organization’s payroll or personnel database for a specific employee. If the employee is not found, the malicious payload executes, deleting vital corporate files.

If the event is a specific date or time, the program will be referred to as a time bomb. In one famous example of a time bomb, a disgruntled employee left a time bomb in place for two weeks before being fired from his job. Two weeks later, thousands of client records were deleted. Police were eventually able to track the malicious code to the disgruntled ex-employee, who was prosecuted for his actions. He had hoped that the two weeks that had passed since his dismissal would have caused investigators to assume he could not have been the individual who had caused the deletion of the records.

Logic bombs are difficult to detect because they are often installed by authorized users and, in particular, have been installed by administrators who are also often responsible for security. This demonstrates the need for a separation of duties and a periodic review of all programs and services that are running on a system. It also illustrates the need to maintain an active backup program so that if your organization loses critical files to this sort of malicious code, it loses only transactions that occurred since the most recent backup, resulting in no permanent loss of data.

Backdoor

Backdoors were originally (and sometimes still are) nothing more than methods used by software developers to ensure that they could gain access to an application even if something were to happen in the future to prevent normal access methods. An example would
be a hard-coded password that could be used to gain access to the program in the event that administrators forgot their own system password. The obvious problem with this sort of backdoor (also sometimes referred to as a trapdoor) is that, since it is hard-coded, it cannot be removed. Should an attacker learn of the backdoor, all systems running that software would be vulnerable to attack.

The term backdoor is also, and more commonly, used to refer to programs that attackers install after gaining unauthorized access to a system to ensure that they can continue to have unrestricted access to the system, even if their initial access method is discovered and blocked. Backdoors can also be installed by authorized individuals inadvertently, should they run software that contains a Trojan horse (introduced earlier). Common backdoors include NetBus and Back Orifice. Both of these, if running on your system, can allow an attacker remote access to your system—access that allows them to perform any function on your system. A variation on the backdoor is the rootkit, discussed in a previous section, which is established not to gain root access but rather to ensure continued root access.

**Exam Tip** The Security+ exam objectives include the ability to compare and contrast different forms of attacks, including keyloggers, adware, spyware, bots, RATS, logic bombs, and backdoors. To prepare for the exam, you should understand the differences between these attacks.

**Indicators of Compromise**

Indicators of compromise (IOC) are just as the name suggests: indications that a system has been compromised by unauthorized activity. When a threat actor makes changes to a system, either by direct action, malware, or other exploit, forensic artifacts are left behind in the system. IOC are as bread crumbs for investigators, providing little clues that can help identify the presence of an attack on a system. The challenge is in looking for, collecting, and analyzing these bits of information and then determining what they mean for a given system. This is one of the primary tasks for an incident responder. Gathering and processing these disparate pieces of data and creating a meaningful picture of the current state of a system.

Fortunately, there are tools to aid the investigator in this task. Tools such as YARA can take a set of signatures (also called IOC) and then scan a system for them, determining whether or not a specific threshold is met indicating a particular infection. Although the specific list will vary based on the system and the specific threat that one is looking for, a common set of IOC that firms should monitor include:

- Unusual outbound network traffic
- Anomalies in privileged user account activity
- Geographical irregularities in network traffic

- Account login red flags
- Increases in database read volumes
- HTML response sizes
- Large numbers of requests for the same file
- Mismatched port-application traffic, including encrypted traffic on plain ports
- Suspicious registry or system file changes
- Unusual DNS requests
- Unexpected patching of systems
- Mobile device profile changes
- Bundles of data in the wrong place
- Web traffic with nonhuman behavior
- Signs of DDoS activity, even if temporary

No single compromise will exhibit everything on this list, but monitoring these items will tend to catch most compromises, because at some point in the compromise lifecycle, every compromise will exhibit one or more of the preceding behaviors. Then, once detected, a responder can zero in on the information and fully document the nature and scope of the problem.

As with many other sophisticated systems, IOC have developed their own internal languages, protocols, and tools. Two major, independent systems for communicating IOC information exist:

- **OpenIOC** Originally developed by Mandiant (acquired by Fireeye) to facilitate information of IOC data. Mandiant subsequently made OpenIOC open source.
- **STIX/TAXII/CybOX** MITRE designed Structured Threat Information Expression (STIX), Trusted Automated Exchange of Indicator Information (TAXII), and Cyber Observable Expression (CybOX) to specifically facilitate automated information sharing between organizations.

**Chapter Review**

This chapter examined the types of malware commonly found in today's environment, including viruses, polymorphic malware, ransomware, worms, Trojans, keyloggers, rootkits, and more. The chapter then looked at systems including RATS, logic bombs, and backdoors. The chapter concluded with an examination of the topic of indicators of compromise, examining this as a means of determining a past or active infection.
Questions
To help you prepare further for the CompTIA Security+ exam, and to test your level of preparedness, answer the following questions and then check your answers against the correct answers at the end of the chapter.

1. A disgruntled administrator is fired for negligence at your organization. Thirty days later, your organization's internal file server and backup server crash at exactly the same time. Examining the servers, it appears that critical operating system files were deleted from both systems. If the disgruntled administrator was responsible for administering those servers during her employment, this is most likely an example of what kind of malware?
   A. Crypto-malware
   B. Trojan
   C. Worm
   D. Logic bomb

2. A desktop system on your network has been compromised. Despite loading different operating systems using different media on the same desktop, attackers appear to have access to that system every time it is powered up and placed on the network. This could be an example of what type of rootkit?
   A. Application
   B. Kernel
   C. Firmware
   D. Virtual

3. A colleague has been urging you to download a new animated screensaver he has been using for several weeks. While he is showing you the program, the cursor on his screen moves on its own and a command prompt window opens and quickly closes. You can tell what if anything was displayed in that command prompt window. Your colleague says "It's been doing that for a while, but it's no big deal." Based on what you've seen, you suspect the animated screensaver is really what type of malware?
   A. A worm
   B. A Trojan
   C. Ransomware
   D. Adware

4. Several desktops in your organization are displaying a red screen with the message: "Your files have been encrypted. Pay 1 bitcoin to recover them." These desktops have most likely been affected by what type of malware?
   A. Zorab worm
   B. Adware
   C. Ransomware
   D. Rootkit

5. While port scanning your network for unauthorized systems, you notice one of your file servers has TCP port 31337 open. When you connect to the port with netcat, you see a prompt that reads: "Enter password for access." Your server may be infected with what type of malware?
   A. Virus
   B. Cryptolocker
   C. Backdoor
   D. Spyware

6. A user in your organization is having issues with her laptop. Every time she opens a web browser, she sees different pop-ups every few minutes. It doesn't seem to matter which websites are being visited—the pop-ups still appear. What type of malware does this sound like?
   A. Adware
   B. Virus
   C. Ransomware
   D. BitLocker

7. Your organization is struggling to contain a recent outbreak of malware. On some of the PCs, your antivirus solution is able to detect and clean the malware. On other PCs exhibiting the exact same symptoms, your antivirus solution reports the system is "clean." These PCs are all running the same operating system and same antivirus software. What might be happening?
   A. Your firewall rules are allowing attackers to backdoor those PCs.
   B. The antivirus solution is reporting false negatives on some of the PCs.
   C. The antivirus solution isn't properly licensed on all systems.
   D. Your systems are infected with polymorphic malware.

8. Malware engineers sometimes take steps to prevent reverse engineering of their code. A virus, such as Zeus, that uses encryption to resist reverse engineering attempts is what type of malware?
   A. Armored virus
   B. Rootkit
   C. RAT
   D. Cryptolocker
9. A colleague can’t open any Word document he has stored on his local system. When you force open one of the documents to analyze it, you see nothing but seemingly random characters. There’s no visible sign the file is still a Word document. Regardless of what you use to view or open the Word documents, you don’t see anything but random characters. Your colleague was most likely a victim of what type of malware?
   A. Virus  
   B. Crypto-malware  
   C. RAT  
   D. Backdoor

10. An employee at your organization is concerned because her ex-spouse “seems to know everything she does.” She tells you her ex keeps accessing her e-mail and social media accounts even after she has changed her passwords multiple times. She is using a laptop at home that was a gift from her ex. Based on what you’ve been told, you suspect the laptop has what type of malware loaded on it?
    A. Adware  
    B. Keylogger  
    C. Logic bomb  
    D. Ransomware

11. Users at your organization are complaining about slow systems. Examining several of them, you see that CPU utilization is extremely high and a process called “brimine” is running on each of the affected systems. You also notice each of the affected systems is communicating with an IP address outside your country via UDP port 43232. If you disconnect the network connections on the affected systems, the CPU utilization drops significantly. Based on what you’ve observed, you suspect these systems are infected with what type of malware?
    A. Adware  
    B. Bot  
    C. CryptoLocker  
    D. Armored Virus

12. A piece of malware is infecting the desktops in your organization. Every hour, more systems are infected. The infections are happening in different departments and in cases where the users don’t share any files, programs, or even e-mails. What type of malware can cause this type of infection?
    A. Virus  
    B. RAT  
    C. BitLocker  
    D. Worm

13. Which of the following could be an indicator of compromise?
   A. Unusual outbound network traffic  
   B. Increased number of logins  
   C. Large numbers of requests for the same file  
   D. All of the above

14. You notice some unusual network traffic and discover several systems in your organization are communicating with a rather dubious "market research" company on a regular basis. When you investigate further you discover that users of the affected systems all installed the same piece of freeware. What might be happening on your network?
   A. These users unwittingly installed spyware.  
   B. These systems are all infected with ransomware.  
   C. This could be normal behavior and nothing to worry about.  
   D. These systems are infected with logic bombs.

15. Which of the following are characteristics of remote-access Trojans?
   A. They can be deployed through malware such as worms.  
   B. They allow attacks to connect to the system remotely.  
   C. They give attackers the ability to modify files and change settings.  
   D. All of the above

**Answers**

1. D. As both servers crashed at exactly the same time, this is most likely a logic bomb. A logic bomb is a piece of code that sits dormant for a period of time until some event or date invokes its malicious payload—in this case, 30 days after the disgruntled employee was fired.

2. C. This is most likely a firmware rootkit, possibly in the video card or expansion card. In the given scenario, the rootkit has to reside outside of the operating system and applications loaded on that system.

3. B. The animated screensaver is most likely a Trojan. The software appears to do one thing, but contains hidden, additional functionality. Your colleague brought the Trojan "inside the walls" when he downloaded and installed the software on his desktop.

4. C. This is quite clearly ransomware. The malware has encrypted files on the affected systems and is demanding payment for recovery of the files.

5. C. This prompt most likely belongs to a backdoor—an alternate way of accessing the system. The TCP service is listening for incoming connections and prompts for a password when connections are established. Providing the correct password would grant command-line access to the system.
6. A. This is classic adware behavior. Unwanted pop-ups that appear during browsing sessions regardless of the website being viewed are very typical of adware.

7. D. This is most likely an infection with polymorphic malware. Polymorphic malware is designed to change its own code on a regular basis, but retain the same functionality. The changes in code are designed to mask the malware from signature-based detection. The "clean" PCs in this example are still infected, but with a variant of the malware that no longer matches any signature in the antivirus solution.

8. A. An armored virus is a piece of malware specifically designed to resist reverse engineering attempts. Zeus user encryption in its attempts to prevent security researchers from learning how it works, how it communicates, and so on.

9. B. If specific file types are no longer usable and seem to be nothing but strings of random characters, it's likely your colleague was a victim of crypto-malware. Crypto-malware encrypts files on a system to make them unusable to anyone without the decryption key.

10. B. This is most likely a keylogger, a piece of software that records all keystrokes entered by the user. If the ex was able to access the logs generated by the keylogger, he would be able to see the new passwords for e-mail and social media accounts as they were being changed.

11. B. These systems are most likely infected with a bot and are now part of a botnet. The systems are running an unknown/ unauthorized process and communicating with an external IP address on UDP port 43252. These are all classic signs of bot and botnet activity.

12. D. This infection pattern is typical of a worm. Worms are self-propagating and don't require any human interaction to spread to additional systems.

13. D. Unusual network traffic, additional logins, and large numbers of requests for the same file are all potential indicators of compromise. Individually, they could be considered suspicious, but seen together and affecting the same system would definitely warrant a deeper inspection of that system.

14. A. If all the users installed the same piece of freeware, it is likely they are all infected with spyware. Spyware records and reports user behavior and can do everything from recording keystrokes to monitoring web usage. Spyware is often bundled with freeware.

15. D. All of these are characteristics of remote-access Trojans (RATs). RATs are often deployed through other malware, allow remote access to the affected system, and give the attacker the ability to manipulate and modify the affected system.

**Attacks**

In this chapter, you will:

- Learn how to compare and contrast different types of attacks
- Learn about the different types of attacks, including social engineering, application/service attacks, wireless attacks, and cryptographic attacks

Attacks can be made against virtually any layer or level of software, from network protocols to applications. When an attacker finds a vulnerability in a system, he exploits the weakness to attack the system. The effect of an attack depends on the attacker's intent and can result in a wide range of effects, from minor to severe. An attack on a system might not be visible on that system because the attack is actually occurring on a different system, and the data the attacker will manipulate on the second system is obtained by attacking the first system. Attacks can be against the user, as in social engineering, or against the application, the network, or the cryptographic elements being employed in a system. This chapter compares and contrasts these types of attacks.

Although hackers and viruses receive the most attention in the news, they are not the only methods used to attack computer systems and networks. This chapter addresses many different ways computers and networks are attacked on a daily basis. Each type of attack threatens at least one of the three security requirements: confidentiality, integrity, and availability (the CIA of security).

From a high-level standpoint, attacks on computer systems and networks can be grouped into two broad categories: attacks on specific software (such as an application or the operating system) and attacks on a specific protocol or service. Attacks on a specific application or operating system are generally possible because of an oversight in the code (and possibly in the testing of that code) or because of a flaw, or bug, in the code (again indicating a lack of thorough testing). Attacks on specific protocols or services are attempts either to take advantage of a specific feature of the protocol or service or to use the protocol or service in a manner for which it was not intended. This chapter discusses various forms of attacks of which security professionals need to be aware.

**Certification Objective** This chapter covers CompTIA Security+ exam objective 1.2,

Compare and contrast types of attacks.
Social Engineering Methods

Social engineering is an attack against a user, and typically involves some form of social interaction. The weakness that is being exploited in the attack is not necessarily one of technical knowledge, or even security awareness. Social engineering at its heart involves building a rapport with a user, and then exploiting their trust or vulnerability.

One method of social engineering is to present the user with a situation that appears to be legitimate, but is in fact a lie. This can be done by pretending to be an official from a government agency, or a financial institution, or a company that the user is familiar with. The attacker may use various tactics to gain the user's trust, such as pretending to have a warrant, or pretending to be a friend of a friend.

The best defense against social engineering attacks is a comprehensive training and awareness program that includes social engineering training, as well as other forms of security awareness training. This includes educating users about how to recognize and respond to social engineering attacks, as well as other types of cyber attacks.

Phishing

Phishing is a type of social engineering in which an attacker attempts to obtain sensitive information from users by masquerading as a trusted entity. The attacker may use e-mail, instant messaging, or social media to contact the user and ask for sensitive information such as passwords, credit card numbers, or personal identification numbers.

Spear Phishing

Spear phishing is a type of phishing attack in which the attacker targets a specific group of users, such as employees of a particular company. The attacker may use information gathered from public sources to create a more personalized attack, making it more likely to succeed.

Whaling

Whaling is a type of phishing attack that targets high-value targets, such as executives or senior management. The attacker may use social engineering techniques to gain access to the target's network and then attempt to steal sensitive information or gain control of the network.
the odds of success. Spear phishing is a common method used against whales, as it is designed to appear to be ordinary business for the target, being crafted to initiate a non-suspicious communication. Whales can be deceived in the same manner as any other person; the difference is that the target group is limited, hence an attacker cannot rely upon random returns from a wide population of targets.

**Vishing**

Vishing is a variation of phishing that uses voice communication technology to obtain the information the attacker is seeking. Vishing takes advantage of the trust that some people place in the telephone network. Users are unaware that attackers can spoof (simulate) calls from legitimate entities using Voice over IP (VoIP) technology. Voice messaging calls can also be compromised and used in these attempts. This is used to establish a form of trust that is then exploited by the attacker over the phone. Generally, the attackers are hoping to obtain credit card numbers or other information that can be used in identity theft. The user may receive an e-mail asking him or her to call a number that is answered by the user. The attacker will then verify that the message received was legitimate or report the vishing attempt.

**Tailgating**

Tailgating (or piggybacking) is the simple tactic of following closely behind a person who has just used their own access card or personal identification number (PIN) to gain physical access to a room or building. People are often in a hurry and will frequently not attempt to stop and question this characteristic behavior. The attacker will then gain access to the facility without having to know the access code or having to acquire an access card. It is similar to shoulder surfing in that it relies on the attacker taking advantage of an authorized user who is not following security procedures. Frequently the attacker may even start a conversation with the target before reaching the door so that the user may be more comfortable with allowing the individual in without challenging them. In this sense piggybacking is related to social engineering attacks.

Both the piggybacking and shoulder surfing attack techniques rely on the poor security practices of an authorized user in order to be successful. Thus, both techniques can be easily countered by training employees to use simple procedures to ensure nobody follows them too closely or is in a position to observe their actions. A more sophisticated countermeasure to piggybacking is a mantrap, which utilizes two doors to gain access to the facility. The second door does not open until the first one is closed, and the doors are closely spaced so that an enclosure is formed that only allows one individual through at a time.

**Impersonation**

Impersonation is a common social engineering technique and can be employed in many ways. It can occur in person, over a phone, or online. In the case of an impersonation attack, the attacker assumes a role that is recognized by the person being attacked, and in assuming that role, the attacker uses the potential victim's biases against their better judgment to follow procedures. Impersonations can occur in a variety of manners, from third parties to helpdesk operators, to vendors, or even online sources.

**Third-Party Authorization**

Using previously obtained information about a project, deadlines, bosses, and so on, the attacker arrives with 1) something the victim is quasi-expecting or would see as normal, 2) uses the guise of a project in trouble or some other situation where the attacker will be viewed as helpful or as someone not to upset, and 3) they name-drop "Mr. Big," who happens to be out of the office and unreachable at the moment, avoiding the reference check. And the attacker seldom asks for anything that on the face of it seems unreasonable, or is unlikely to be shared based on the circumstances. These actions can create the appearance of a third-party authorization, when in fact there is none.

**Help Desk/Tech Support**

Calls to or from helpdesk and tech support units can be used to elicit information. Posing as an employee, an attacker can get a password reset, information about some system, or other useful information. The call can go the other direction as well, where the social engineer is posing as the help desk or tech support. Then, by calling employees, the attacker can get information on system status and other interesting elements that they can use later.

**Contractors/Outside Parties**

It is common in many organizations to have outside contractors clean the building, water the plants, and do other routine chores. In many of these situations, without proper safeguards, an attacker can simply put on clothing that matches a contractor's uniform, show up to do the job at a slightly different time than it's usually done, and, if challenged, play on the sympathy of the workers by saying they are filling in for X or covering for Y. The attacker then roams the halls unnoticed because they blend in, all the while photographing desks and papers and looking for information.

**Online Attacks**

Impersonation can be employed in online attacks as well. In these cases, technology plays an intermediary role in the communication chain. Some older forms, such as pop-up windows, tend to be less effective today, because users are wary of them. Yet phishing attempts via e-mail and social media scams abound.

**Defenses**

In all of the cases of impersonation, the best defense is simple—have processes in place that require employees to ask to see a person's ID before engaging with them if the
employees do not personally know them. That includes challenging people such as delivery drivers and contract workers. Don’t let people in through the door, piggybacking, without checking their ID. If this is standard process, then no one becomes offended, and if someone takes offense, it becomes even more suspicious. Training and awareness do work, as proven by trends such as the diminished effectiveness of pop-up windows. But the key to this defense is to make the training periodic and to tailor it to what is currently being experienced, rather than a generic recitation of best practices.

**Exam Tip**
A training and awareness program is still the best defense against social engineering attacks.

**Dumpster Diving**
The process of going through a target’s trash in hopes of finding valuable information that might be useful in a penetration attempt is known in the security community as *dumpster diving*. One common place to find information, if the attacker is in the vicinity of the target, is in the target’s trash. The attacker might find little bits of information that could be useful for an attack. The tactic is not, however, unique to the computer community; it has been used for many years by others, such as identity thieves, private investigators, and law enforcement personnel, to obtain information about an individual or organization. If the attackers are very lucky, and the target’s security procedures are very poor, they may actually find user IDs and passwords.

An attacker may gather a variety of information that can be useful in a social engineering attack. In most locations, trash is no longer considered private property after it has been discarded (and even where dumpster diving is illegal, little enforcement occurs). An organization should have policies about discarding materials. Sensitive information should be shredded and the organization should consider securing the trash receptacle so that individuals can’t forage through it. People should also consider shredding personal or sensitive information that they wish to discard in their own trash. A reasonable quality shredder is inexpensive and well worth the price when compared with the potential loss that could occur as a result of identity theft.

**Shoulder Surfing**
_Shooulder surfing_ does not necessarily involve direct contact with the target, but instead involves the attacker directly observing the individual entering sensitive information on a form, keypad, or keyboard. The attacker may simply look over the shoulder of the user at work, for example, or may set up a camera or use binoculars to view the user entering sensitive data. The attacker can attempt to obtain information such as a PIN at an automated teller machine (ATM), an access control entry code at a secure gate or door, or a calling card or credit card number. Many locations now use a small shield to surround a keypad so that it is difficult to observe somebody as they enter information. More sophisticated systems can actually scramble the location of the numbers so that the top row at one time includes the numbers 1, 2, and 3 and the next time includes 4, 8, and 0.

While this makes it a bit slower for the user to enter information, it thwarts an attacker’s attempt to observe what numbers are pressed and then enter the same buttons/pattern, since the location of the numbers constantly changes.

**Hoax**
At first glance, it might seem that a hoax related to security would be considered a nuisance and not a real security issue. This might be the case for some hoaxes, especially those of the urban legend type, but the reality of the situation is that a hoax can be very damaging if it causes users to take some sort of action that weakens security. One real hoax, for example, described a new, highly destructive piece of malicious software. It instructed users to check for the existence of a certain file and to delete it if the file was found. In reality, the file mentioned was an important file used by the operating system, and deleting it caused problems the next time the system was booted. The damage caused by users modifying security settings can be serious. As with other forms of social engineering, training and awareness are the best and first line of defense for both users and administrators. Users should be trained to be suspicious of unusual e-mails and stories and should know who to contact in the organization to verify their validity if they are received. A hoax can also be used to help spread the hoax. Users need to be suspicious of any e-mail telling them to “spread the word.”

**Watering Hole Attack**
The most commonly recognized attack vectors are those that are direct to a target. Because of their incoming and direct nature, defenses are crafted to detect and defend against them. But what if the user “asked” for the attack by visiting a website? Just as a hunter waits near a watering hole for animals to come drink, attackers can plant malware at sites where users are likely to frequent. First identified by RSA, a _watering hole attack_ involves the infecting of a target website with malware. In some of the cases detected, the infection was constrained to a specific geographical area. These are not simple attacks, yet they can be very effective at delivering malware to specific groups of end users. Watering hole attacks are complex to achieve and appear to be backed by nation states and other resource attackers. In light of the stakes, the typical attack vector will be a zero day attack to further avoid detection.

**Social Engineering Principles**
Social engineering is very successful for two general reasons. The first is the basic desire of most people to be helpful. When somebody asks a question for which we know the answer, our normal response is not to be suspicious but rather to answer the question. The problem with this is that seemingly innocuous information can be used either directly in an attack or indirectly to build a bigger picture that an attacker can use to create an aura of authenticity during an attack—the more information an individual has about an organization, the easier it will be to convince others that he is part of the organization and has a right to even sensitive information.
The second reason that social engineering is successful is that individuals normally seek to avoid confrontation and trouble. If the attacker attempts to intimidate the target, threatening to call the target's supervisor because of a lack of help, the target may give in and provide the information to avoid confrontation.

Tools
The tools in a social engineer's toolbox are based on a knowledge of psychology and don't necessarily require a sophisticated knowledge of software or hardware. The social engineer will employ strategies aimed to exploit people's own biases and beliefs in a manner to momentarily deny them the service of good judgment and the use of standard procedures. Employing social engineering tools is second nature to a social engineer, and with skill they can switch these tools in and out in any particular circumstance, just as a plumber uses various hand tools and a system administrator uses OS commands to achieve complex tasks. When watching any of these professionals work, we marvel at how they wield their tools, and the same is true for social engineers—except their tools are more subtle, and the target is people and trust. The "techniques" that are commonly employed in social engineering attacks are described next.

NOTE A great video showing the use of several social engineering tools can be found at https://www.youtube.com/watch?v=uc7svV7K000 (This is how hackers hack you using simple social engineering). This video demonstrates the use of visting to steal someone's cell phone credentials.

Authority
The use of authority in social situations can lead to an environment where one party feels at risk in challenging another over an issue. If an attacker can convince a target that he has authority in a particular situation, he can entice the target to act in a particular manner or risk adverse consequences. In short, if you act like a boss when requesting something, people are less likely to withhold it.

The best defense against this and many social engineering attacks is a strong set of policies that has no exceptions. Much like security lines in the airport, when it comes to the point of screening, everyone gets screened, even flight crews, so there is no method of bypassing the critical step.

Intimidation
Intimidation can be either subtle, through perceived power, or more direct, through the use of communications that build an expectation of superiority.

Consensus
Consensus is a group-wide decision. It frequently comes not from a champion, but rather through rounds of group negotiation. These rounds can be manipulated to achieve desired outcomes. The social engineer simply motivates others to achieve her desired outcome.

Scarcity
If something is in short supply and is valued, then arriving with what is needed can bring rewards—and acceptance. "Only X widgets left at this price" is an example of this technique. Even if something is not scarce, implied scarcity, or implied future change in availability, can create a perception of scarcity. By giving the impression of scarcity, or short supply, of a desirable product, an attacker can motivate a target to make a decision quickly without deliberation.

Familiarity
People do things for people they like or feel connected to. Building this sense of familiarity and appeal can lead to misplaced trust. The social engineer can focus the conversation on familiar items, not the differences. Again, leading with persuasion that one has been there before and done something, even if they haven't, for perception will lead to the desired familiar feeling.

Trust
Trust is defined as having an understanding of how something will act under specific conditions. Social engineers can shape the perceptions of a target to where they will apply judgments to the trust equation and come to false conclusions. The whole objective of social engineering is not to force people to do things they would not do, but rather to give them a pathway that leads them to feel they are doing the correct thing in the moment.

Urgency
Time can be manipulated to drive a sense of urgency and prompt shortcuts that can lead to opportunities for interjection into processes. Limited-time offers should always be viewed as suspect. Perception is the key. Giving the target a reason to believe that they can take advantage of a time situation, whether it really is present or not, achieves the outcome of them acting in a desired manner.

Application/Service Attacks
In the beginning of the computer security era, most attacks were against the network and operating system layers because both had easily exploitable vulnerabilities and were relatively ubiquitous. As the networking companies and OS vendors cleaned up their code bases, exploiting these layers became much more difficult. Attackers shifted their focus to applications. The application layer was a much less homogenous target because there were many different applications, but the ubiquity of vulnerabilities made up for the
lower level of homogeneity. Certain desktop applications, like Adobe Flash and Acrobat, became frequent targets.

Application security controls and techniques are important in ensuring that the applications deployed are as secure as possible. Establishing the security of an application begins with secure coding techniques and then adding security controls to provide defense in depth. Using application hardening techniques and proper configuration and change controls provides a process-driven method to ensure continued security per a defined risk profile.

**DoS**

Denial-of-service (DoS) attacks can exploit a known vulnerability in a specific application or operating system, or they can attack features (or weaknesses) in specific protocols or services. In a DoS attack, the attacker attempts to deny authorized users access either to specific information or to the computer system or network itself. This can be accomplished by crashing the system—taking it offline—or by sending so many requests that the machine is overwhelmed.

The purpose of a DoS attack can be simply to prevent access to the target system, or the attack can be used in conjunction with other actions to gain unauthorized access to a computer or network. For example, a SYN flooding attack can be used to prevent service to a computer or network, or in conjunction with other actions, to gain unauthorized access to a computer or network.

SYN flooding is an example of a DoS attack that takes advantage of a common TCP/IP network error. The error is a function of TCP and can be exploited to establish a false connection between two systems. Under normal circumstances, the first system sends a SYN packet to the second system with which it wants to communicate. The second system responds with a SYN/ACK if it is able to accept the request. When the initial system receives the SYN/ACK from the second system, it responds with an ACK packet, and communication can then proceed. This process is shown in Figure 2-1.

In a SYN flooding attack, the attacker sends fake communication requests to the target system. Each of these requests is answered by the target system, which then waits for the third part of the handshake. Since the requests are fake (a nonexistent IP address is used in the requests, to the target system is responding to a target that doesn't exist), the target will wait for responses that never come, as shown in Figure 2-2. The target system will drop these connections after a specific time-out period, but if the attacker sends requests faster than the time-out period eliminates them, the system will quickly be filled with requests. The number of connections a system can support is finite, so when more requests come in than can be processed, the system will soon be reserving all its connections for fake requests. At this point, any further requests are simply dropped (ignored), and legitimate users who want to connect to the target system will not be able to do so, because use of the system has been denied to them.

Another simple DoS attack is the infamous ping of death (POD), and it illustrates the other type of attack—one targeted at a specific application or operating system, as opposed to SYN flooding, which targets a protocol. The POD attack, the attacker sends an Internet Control Message Protocol (ICMP) ping packet equal to, or exceeding, 64KB (which is to say, greater than 64 \times 1024 = 65,536 bytes). This type of packet should not occur naturally (there is no reason for a ping packet to be larger than 64KB). Certain systems are not able to handle this size of packet, and the system will hang or crash.

**DDoS**

DDoS attacks are conducted using a single attacking system. A DoS attack employing multiple attacking systems is known as a distributed denial-of-service (DDoS) attack. The goal of a DDoS attack is to deny the use of or access to a specific service or system. DDoS attacks were made famous in 2000 with the highly publicized attacks on eBay, CNN, Amazon, and Yahoo.

In a DDoS attack, service is denied by overwhelming the target with traffic from many different systems. A network of attack agents (sometimes called zombies) is created by the attacker, and upon receiving the attack command from the attacker, the attack agents commence sending a specific type of traffic against the target. If the attack traffic is large enough, even ordinary web traffic can overwhelm the largest of sites, such as the 400-Gbps CloudFlare attack in early 2014.

Creating a DDoS network is no simple task. The attack agents are not willing agents—they are systems that have been compromised and on which the DDoS attack software has been installed. To compromise these agents, the attacker has to have gained unauthorized access to the system or tricked authorized users to run a program that installed the attack software. The creation of the attack network may in fact be a multistep process in which the attacker first compromises a few systems that are then used as handlers or
masters, which in turn compromise other systems. Once the network has been created, the agents (zombies) wait for an attack message that will include data on the specific target before launching the attack. One important aspect of a DDoS attack is that with just a few messages to the agents, the attacker can have a flood of messages sent against the targeted system. Figure 2-3 illustrates a DDoS network with agents and handlers.

How can you stop or mitigate the effects of a DoS or DDoS attack? One important precaution is to ensure that you have applied the latest patches and updates to your systems and the applications running on them. Once a specific vulnerability is discovered, it is crucial to take action as soon as possible. Generally, does not take long before multiple exploits are written and released to take advantage of it. Generally, you will have a small window of opportunity in which to patch your system between the time the vulnerability is discovered and the time exploited becomes widely available. A vulnerability can also be exploited by hackers, and exploits provide the first clues to what a system has been compromised. Attackers can also reverse-engineer patches to learn what vulnerabilities have been patched, allowing them to attack unpatched systems.

Another approach involves changing the time-out option for TCP connections so that attacks such as the SYN flooding attack are more difficult to perform, because unused connections are dropped more quickly.

For DDoS attacks, much has been written about distributing your own load across several systems so that any attack against your system would have to target several hosts to be completely successful. While this is true, if large enough, DDoS networks are created (with tens of thousands of zombies, for example), any network, no matter how much the load is distributed, can be successfully attacked. Such an approach also involves additional costs to your organization to establish this distributed environment. Addressing the problem in this manner is actually an attempt to mitigate the effects of the attack, rather than preventing or stopping an attack.

To prevent a DDoS attack, you must either be able to intercept or block the attack messages or keep the DDoS network from being established in the first place. Tools have been developed that will scan your systems, searching for sleeping zombies waiting for an attack signal. Many of the current antivirus/spyware security suite tools will detect known zombie-type infections. The problem with this type of prevention approach, however, is that it is not something you can do to prevent an attack on your network—it is something you can do to keep your network from being used to attack other networks or systems. You have to rely on the community of network administrators to test their own systems to prevent attacks on yours.

A final option you should consider that will address several forms of DoS and DDoS attacks is to block ICMP packet at your border, since many attacks rely on ICMP. Carefully consider this approach before implementing it, however, because it will also prevent the use of some possibly useful troubleshooting tools.

**Man-in-the-Middle**

A man-in-the-middle attack, as the name implies, generally occurs when an attacker is able to place himself in the middle of two other hosts that are communicating. Ideally (from the attacker's perspective), this is done by ensuring that all communication going to or from the target host is routed through the attacker's host (which can be accomplished if the attacker can compromise the router for the target host). The attacker can then observe all traffic before relaying it, and can actually modify or block traffic. To the target host, it appears that communication is occurring normally, since all expected replies are received. Figure 2-4 illustrates this type of attack.

There are numerous methods of instantiating a man-in-the-middle attack. One of the common methods is via session hijacking, which can occur when information such as a cookie is stolen, allowing the attacker to impersonate the legitimate session. This attack can be a result of a cross-site scripting attack, which tricks a user into executing code resulting in cookie theft. The amount of information that can be obtained in a man-in-the-middle attack will be limited if the communication is encrypted. Even in this case, however, sensitive information can still be obtained, since knowing what communication is being conducted, and between which individuals, may in fact provide information that is valuable in certain circumstances.

**Buffer Overflow**

If there's one item that could be labeled as the "Most Wanted" in coding security, it would be the buffer overflow. The CERT Coordination Center (CERT/CC) at Carnegie Mellon University estimates that nearly half of all exploits of computer programs stem historically from some form of buffer overflow. Finding a vaccine to buffer overflows would stamp
out half of these security-related incidents, by type, and probably 90 percent by volume. The Morris finger worm in 1988 was an exploit of an overflow, as were more recent big-name events such as Code Red and Slammer. The generic classification of buffer overflows includes many variants, such as static buffer overruns, indexing errors, format string bugs, Unicode and ANSI buffer size mismatches, and heap overruns.

The concept behind these vulnerabilities is relatively simple. In a buffer overflow, the input buffer that is used to hold program input is overwritten with data that is larger than the buffer can hold. The root cause of this vulnerability is a mixture of two things: poor programming practice and programming language weaknesses. For example, what would happen if a program that asks for a 7- to 10-character phone number instead receives a string of 150 characters? Many programs will provide some error checking to ensure that this will not cause a problem. Some programs, however, cannot handle this error, and the extra characters continue to fill memory, overwriting other portions of the program. This can result in a number of problems, including causing the program to abort or the system to crash. Under certain circumstances, the program can execute a command supplied by the attacker. Buffer overflows typically inherit the level of privilege enjoyed by the program being exploited. This is why programs that use root-level access are so dangerous when exploited with a buffer overflow, as the code that will do this does so at root-level access.

Programming languages such as C were designed for space and performance constraints. Many functions in C, like gets(), are unsafe in that they will permit unsafe operations, such as unbounded string manipulation into fixed buffer locations. The C language also permits direct memory access via pointers, a functionality that provides a lot of programming power, but carries with it the burden of proper safeguards being provided by the programmer.

**EXAM TIP** Buffer overflows can occur in any code, and code that runs with privilege has an even greater risk profile. In 2014, a buffer overflow in the OpenSSL library, called Heartbleed, left hundreds of thousands of systems vulnerable and exposed critical data for tens to hundreds of million users worldwide.

Buffer overflows are input validation attacks, designed to take advantage of input routines that do not validate the length of inputs. Surprisingly simple to resolve, all that is required is the validation of all input lengths (input validation) prior to writing to memory. This can be done in a variety of manners, including the use of safe library functions for inputs. This is one of the vulnerabilities that has been shown to be solvable, and in fact the prevalence is declining substantially among major security-conscious software firms.

**Injection**

When user input is used without input validation, this results in an opportunity for an attacker to craft input to create specific events to occur when the input is parsed and used by an application. SQL injection attacks involve the manipulation of input resulting in a SQL statement that is different than intended by the designer. XML and LDAP injections are done in the same fashion. As SQL, XML, and LDAP are used to store data, this can give an attacker access to data against business rules. Command injection attacks can occur when input is used in a fashion that allows command-line manipulation. This can give an attacker command-line access at the privilege level of the application.

**Cross-Site Scripting**

Cross-site scripting (XSS) is one of the most common web attack methodologies. The cause of the vulnerability is weak user input validation. If input is not validated properly, an attacker can include a script in their input and have it rendered as part of the web process. There are several different types of XSS attacks, which are distinguished by the effect of the script:

- **Non-persistent XSS attack** The injected script is not persisted or stored, but rather is immediately executed and passed back via the web server.
- **Persistent XSS attack** The script is permanently stored on the web server or some back-end storage. This allows the script to be used against others who log in to the system.
- **DOM-based XSS attack** The script is executed in the browser via the Document Object Model (DOM) process as opposed to the web server.

Cross-site scripting attacks can result in a wide range of consequences, and in some cases, the list can be anything that a clever scripter can devise. Common uses that have been seen in the wild include:

- Theft of authentication information from a web application
- Session hijacking
- Deploying hostile content
- Changing user settings, including future users
- Impersonating a user
- Phishing or stealing sensitive information

Controls to defend against XSS attacks include the use of anti-XSS libraries to strip scripts from the input sequences. Various other ways to mitigate XSS attacks include limiting types of uploads and screening the size of uploads, whitelisting inputs, and so on, but attempting to remove scripts from inputs can be a tricky task. Well-designed anti-XSS input library functions have proven to be the best defense. Cross-site scripting vulnerabilities are easily tested for and should be a part of the test plan for every application. Testing a variety of encoded and unencoded inputs for scripting vulnerability is an essential test element.
Cross-Site Request Forgery

Cross-site request forgery (CSRF) attacks utilize unintended behaviors that are proper in defined use but are performed under circumstances outside the authorized use. This is an example of a "confused deputy" problem, a class of problems where one entity mistakenly performs an action on behalf of another. An XSRF attack relies upon several conditions that can make it effective. It is performed against sites that have an authenticated user and exploits it. In order to mitigate the attack, sites should have a way to authenticate a user and perform transactions, but do not validate the authentication in a transaction that appears to be authorized but in fact was not done by the user. There are many different mitigation techniques that can be employed, from limiting to cookie expiration, to managing specific elements of a web page like header checking. The strongest method is the use of random XSRF tokens in form submissions. Subsequent requests cannot work, as the token is not set in advance. Testing for XSRF takes a bit more planning than for other injection-type attacks, but this, too, can be accomplished as part of the design process.

Privilege Escalation

Cyberattacks are multipurpose processes. Most attacks begin at a privilege level associated with an ordinary user. From this level, the attacker exploits vulnerabilities that enable them to achieve root- or admin-level access. This step in the attack chain is called privilege escalation and is essential for many attack efforts.

There are a couple of ways to achieve privilege escalation. On a pathway is to use existing privilege and do an act that allows you to steal a better set of credentials. The use of sniffers to grab credentials, getting the SAM or etc/passwd file, is one method of obtaining "better" credentials. Another method is through vulnerabilities or weaknesses in processes that are running with escalated privileges. Injecting malicious code into these processes can also achieve escalated privilege.

EXAM TIP Blocking privilege escalation is an important defensive step in a system. This is the rationale behind Microsoft's recent reduction in processes and services that run in elevated mode. This greatly reduces the attack surface available for an attacker to perform this essential task.

ARP Poisoning

In moving packets between machines, a device sometimes needs to know where to send a packet using the MAC or layer 2 address. Address Resolution Protocol (ARP) handles this problem through four basic message types:

- ARP request "Who has this IP address?"
- ARP reply "I have that IP address; my MAC address is..."

These messages are exchanged in conjunction with a device's ARP table, where a form of short-term memory is associated with these data elements. The commands are used as a simple form of lookup. When a device sends an ARP request to the network, the reply is received and inserted into all devices that hear the reply. This facilitates efficient address lookups, but also makes the system subject to attack.

When the ARP table gets a reply, it automatically trusts the reply and updates the table. Some operating systems will even accept ARP reply data if they never heard the original request. There is no mechanism to verify the collectivity of the data received.

An attacker can send messages, corrupt the ARP table, and cause packets to be misrouted. This form of attack is called ARP poisoning and results in malicious address redirection. This can allow an attacker to inject themselves into the middle of a conversation between two machines, a man-in-the-middle attack.

Amplification

Certain types of attacks could be considered to be dependent upon volume, such as DoS and DDOS attacks. For these attacks, a sufficient volume of packets to overwhelm a host, typically a large server, they require more than a single home PC. Amplification is a trick where an attacker uses a specific protocol aspect to achieve what a single machine cannot by itself. As an example, consider the ICMP command ping. If you issue an ICMP ping command, the machine receiving it provides a ping reply packet. What if you were to send the ICMP request to a network address, in essence all active hosts within that network? You would all reply with a packet. Now, suppose that an attacker forges the requesting packet so that the reply address is a specific machine. The net effect is all of those machines will reply to the forged address—one machine, with an amplified response.

DNS Poisoning

The DNS system is used to convert a name into an IP address. There is not a single DNS system, but rather a hierarchy of DNS servers, from root servers on the backbone of the Internet, to copies at your ISP, your home router, and your local machine, each in the form of a DNS cache. To examine a DNS query for a specific address, you can use the nslookup command. Figure 2-5 shows a series of DNS queries executed on a Windows machine. In the first request, the DNS server was from an ISP; while on the second request, the DNS server was from a VPN connection. Between the two requests, the network connections were changed, resulting in different DNS lookups. The changing of where DNS is resolved can be a DNS poisoning attack. The challenge in detecting these attacks is knowing what the authoritative DNS entry should be, and detecting when it changes in an unauthorized fashion. Using a VPN can change a DNS source, and this may be desired, but unauthorized changes can be attacks.
At times, nslookup will return a nonauthoritative answer, as shown in Figure 2-6. This typically means the result is from a cache as opposed to a server that has an authoritative (that is, known to be current) answer.

There are other commands you can use to examine and manipulate the DNS cache on a system. In Windows, the ipconfig /displaydns command will show the current DNS cache on a machine. Figure 2-7 shows a small DNS cache. This cache was recently emptied using the ipconfig /flushdns command to make it fit on the screen.

Looking at DNS as a complete system shows that there are hierarchical levels from the top (root server) down to the cache in an individual machine. DNS poisoning can occur at any of these levels, with the effect of the poisoning growing wider the higher up it occurs. In 2010, a DNS poisoning event resulted in the “Great Firewall of China” censoring Internet traffic in the United States until caches were resolved.

DNS poisoning is a variant of a larger attack class referred to as DNS spoofing. In DNS spoofing, an attacker changes a DNS record through any of a multitude of means. There are many ways to perform DNS spoofing, a few of which include compromising a DNS server, the use of the Kaminsky attack, and the use of a false network node advertising a false DNS address. An attacker can even use DNS cache poisoning to result in DNS spoofing. By poisoning an upstream DNS cache, all of the downstream users will get spoofed DNS records.

Because of the importance of integrity on DNS requests and responses, a project has begun to secure the DNS infrastructure using digital signing of DNS records. This project, initiated by the U.S. government and called Domain Name System Security Extensions (DNSSEC), works by digitally signing records. This is done by adding records to the DNS system, a key and a signature attesting to the validity of the key. With this information, requestors can be assured that the information they receive is correct. It will take a substantial amount of time (years) for this new system to propagate through the entire DNS infrastructure, but in the end, the system will have much greater assurance.

**Domain Hijacking**

Domain hijacking is the act of changing the registration of a domain name without the permission of its original registrant. Technically a crime, this act can have devastating consequences because the DNS system will spread the false domain location far and wide automatically. The original owner can request it to be corrected, but this can take time.
**Man-in-the-Browser**

The man-in-the-browser (MitB) attack is a variant of a man-in-the-middle attack. In a MitB attack, the first element is a malware attack that places a Trojan element that can act as a proxy on the target machine. This malware changes browser behavior through browser helper objects or extensions. When a user connects to their bank, the malware recognizes the target (a financial transaction) and injects itself in the stream of the conversation. When the user approves a transfer of $150 to pay a utility bill, for example, the malware intercepts the user's keystrokes and modifies them to perform a different transaction. A famous example of a MitB attack was the financial malware, Zeus, which targeted financial transactions on users' machines, manipulating and changing them after the users had entered password credentials.

**Zero Day**

A zero day attack is one that uses a vulnerability for which there is no previous knowledge outside of the attacker, or at least not the software vendor. Zero day attacks are critical as there is no known defense to the vulnerability itself, leaving the only security solution to be secondary solutions, such as catching subsequent hacker activity. Zero day vulnerabilities are highly valued by attackers because they are almost sure bets when attacking a system. There is a market in zero day vulnerabilities, where hackers trade them. There is also an interesting question with respect to government collection of zero days, which they use for intelligence operations. Should governments keep secret libraries of zero days, or should they alert the software vendors, allowing patching and protection across the broader environment of systems?

**EXAM TIP** Zero day attacks are becoming more common as collections of zero day vulnerabilities are being released by hackers outing government collections of zero day techniques. Understanding this term and how it relates to risk in a system is important to differentiate this attack pattern, which is part of the "compare and contrast" aspect of the Security+ objective associated with understanding attacks.

**Replay**

A replay attack occurs when the attacker captures a portion of a communication between two parties and retransmits it at a later time. For example, an attacker might replay a series of commands and codes used in a financial transaction to cause the transaction to be conducted multiple times. Generally, replay attacks are associated with attempts to circumvent authentication mechanisms, such as the capturing and reuse of a certificate or ticket.

The best way to prevent replay attacks is with encryption, cryptographic authentication, and time stamps. If a portion of the certificate or ticket includes a date/time stamp or an expiration date/time, and this portion is also encrypted as part of the ticket or certificate, replaying it at a later time will prove useless, since it will be rejected as having expired.

**Pass the Hash**

**EXAM TIP** The best method for defending against replay attacks is through the use of encryption and short time frames for legal transactions. Encryption can protect the contents from being understood, and a short time frame for a transaction prevents subsequent use.

**Pass the Hash**

Pass the hash is a hacking technique where the attacker captures the hash used to authenticate a process. They can then use this hash by injecting it into a process in place of the password. This is a highly technical attack, targeting the Windows authentication process, injecting a copy of the password hash directly into the system. The attacker does not need to know the password, but instead can use a captured hash and inject it directly, which will verify correctly, granting access. As this is a very technically specific hack, tools have been developed to facilitate its operation.

**Hijacking and Related Attacks**

Hijacking is a form of attack where the attacker hijacks a user's experience, typically after the exchange of credentials, or in the background in a manner where the user is not even aware of the attack process.

**Clickjacking**

Clickjacking is an attack against the design element of a user interface. Clickjacking tricks a web browser user into clicking something different from what the user perceives, by means of malicious code in the web page. This malicious code may be a transparent overlay or other means of disguising rouge elements, but the net result is the user thinks they are clicking one thing but in reality are clicking the attacker's hidden control, causing the browser to execute the attacker's code. If the attacker modifies a page so that a transparent overlay with invisible clickable elements aligns with actual elements, then the code that runs when a click occurs can be the attacker's code.

**Session Hijacking**

TCP/IP hijacking and session hijacking are terms used to refer to the process of taking control of an already existing session between a client and a server. The advantage to an attacker of hijacking over attempting to penetrate a computer system or network is that the attacker doesn't have to circumvent any authentication mechanisms, since the user has already authenticated and established the session. Once the user has completed the authentication sequence, the attacker can then usurp the session and carry on as if the attacker, and not the user, had authenticated with the system. To prevent the user from noticing anything unusual, the attacker can decide to attack the user's system and perform a DoS attack on it, taking it down so that the user, and the system, will not notice the extra traffic that is taking place.

Hijack attacks generally are used against web and Telnet sessions. Sequence numbers as they apply to spoofing also apply to session hijacking, since the hijacker will need to provide the correct sequence numbers to continue the appropriate sessions.
URL Hijacking

URL hijacking is a generic name for a wide range of attacks that target the URL. The URL is the primary means by which a user receives web content. If the correct URL is used, you get the desired content. If the URL is tampered with or altered, you can get different content. There are a wide range of URL-based attacks, from malware manipulations, to typo squatting, to ad-based attacks that make the user think they are clicking the correct link. The net result is the same: the user thinks they are asking for content A, and they get B instead.

Typo Squatting

Typo squatting is an attack form that involves capitalizing upon common typo errors. If a user mistypes a URL, then the result should be a 404 error, or "resource not found." But if an attacker has registered the misspelled URL, then you would land on the attacker's page. This attack pattern is also referred to as URL hijacking, fake URL, or brandjacking if the objective is to deceive based on branding.

There are several reasons that an attacker will pursue this avenue of attack. The most obvious is one of a phishing attack. The fake site collects credentials, passing them on to the real site, and then steps out of the conversation to avoid detection once the credentials are obtained. It can also be used to plant drive-by malware on the victim machine. It can move the packets through an affiliate network, earning click-through revenue based on the typos. There are numerous other forms of attacks that can be perpetrated using a fake URL as a starting point.

Driver Manipulation

Drivers are pieces of software that sit between the operating system and a peripheral device. In one respect, drivers are a part of the OS, an extension. In another respect, drivers are code that is not part of the OS and is developed by firms other than the OS developer. Driver manipulation is the attack on a system by changing drivers, thus changing the behavior of the system. Drivers may not be as protected as other parts of the core system, yet they join it when invoked. This has led to drivers being signed and significantly tightening up the environment of drivers and ancillary programs.

Shimming

Shimming is a process of putting a layer of code between the driver and the OS. Shimming allows flexibility and portability, for it enables changes between different versions of an OS without modifying the original driver code. Shimming also represents a means by which malicious code can change a driver's behavior without changing the driver itself.

Refactoring

Refactoring is the process of restructuring existing computer code without changing its external behavior. Refactoring is done to improve nonfunctional attributes of the software, such as improving code readability and/or reducing complexity. Refactoring can uncover design flaws that lead to exploitable vulnerabilities, allowing these to be closed without changing the external behavior of the code. Refactoring is a means by which an attacker can add functionality to a drive, yet maintain its desired functionality. Although this goes against the original principle of refactoring, improving code efficiency, it speaks to the ingenuity of attackers.

Spoofing

Spoofing is nothing more than making data look like it comes from a different source. This is possible in TCP/IP because of the friendly assumptions behind the protocols. When the protocols were developed, it was assumed that individuals who had access to the network layer would be privileged users who could be trusted.

When a packet is sent from one system to another, it includes not only the destination IP address and port but the source IP address as well. You are supposed to fill in the source with your own address, but nothing stops you from filling in another system's address. This is one of the several forms of spoofing.

MAC Spoofing

MAC spoofing is the act of changing a MAC address to bypass security checks based on the MAC address. This can work when the return packets are being routed by IP address and can be correctly linked to the correct MAC address. Not all MAC spoofing is an attack; small firewall routers commonly have a MAC clone function by which the device can clone a MAC address to make it seem transparent to other devices such as the cable modem connection.

IP Address Spoofing

IP is designed to work so that the originators of any IP packet include their own IP address in the From portion of the packet. While this is the intent, nothing prevents a system from inserting a different address in the From portion of the packet. This is known as IP address spoofing. An IP address can be spoofed for several reasons.

Smurf Attack

In a specific DoS attack known as a Smurf attack, the attacker sends a spoofed packet to the broadcast address for a network, which distributes the packet to all systems on that network. In the Smurf attack, the packet sent by the attacker to the broadcast address is an echo request with the From address forged so that it appears that another system (the target system) has made the echo request. The normal response of a system to an echo request is an echo reply, and it is used in the ping utility to let a user know whether a remote system is reachable and is responding. In the Smurf attack, the request is sent to all systems on the network, so we will respond with an echo reply to the target system, as shown in Figure 2-8. The attacker has sent one packet and has been able to generate as many as 254 responses aimed at the target. Should the attacker send several of these spoofed requests, or send them to several different networks, the target can quickly become overwhelmed with the volume of echo replies it receives.
EXAM TIP  A Smurf attack allows an attacker to use a network structure to send large volumes of packets to a victim. By sending ICMP requests to a broadcast IP address, with the victim as the source address, the multitudes of replies will flood the victim system.

Spoofing and Trusted Relationships
Spoofing can also take advantage of a trusted relationship between two systems. If two systems are configured to accept the authentication accomplished by each other, an individual logged on to one system might not be forced to go through an authentication process again to access the other system. An attacker can take advantage of this arrangement by sending a packet to one system that appears to have come from a trusted system. Since the trusted relationship is in place, the targeted system may perform the requested task without authentication.

Since a reply will often be sent once a packet is received, the system that is being impersonated could interfere with the attack, since it would receive an acknowledgment for a request it never made. The attacker will often initially launch a DoS attack (such as a SYN flooding attack) to temporarily take out the spoofed system for the period of time that the attacker is exploiting the trusted relationship. Once the attack is completed, the DoS attack on the spoofed system would be terminated and the administrators, apart from having a temporarily nonresponsive system, possibly may never notice that the attack occurred. Figure 2-9 illustrates a spoofing attack that includes a SYN flooding attack.

Because of this type of attack, administrators are encouraged to strictly limit any trusted relationships between hosts. Firewalls should also be configured to discard any packets from outside of the firewall that have From addresses indicating they originated from inside the network (a situation that should not occur normally and that indicates spoofing is being attempted).

Spoofing and Sequence Numbers
How complicated the spoofing is depends heavily on several factors, including whether the traffic is encrypted and where the attacker is located relative to the target. Spoofing attacks from inside a network, for example, are much easier to perform than attacks from outside of the network, because the inside attacker can observe the traffic to and from the target and can do a better job of formulating the necessary packets.

Formulating the packets is more complicated for external attackers because a sequence number is associated with TCP packets. A sequence number is a 32-bit number established by the host that is incremented for each packet sent. Packets are not guaranteed to be received in order, and the sequence number can be used to help reorder packets as they are received and to refer to packets that may have been lost in transmission.

In the TCP three-way handshake, two sets of sequence numbers are created, as shown in Figure 2-10. The first system chooses a sequence number to send with the original SYN packet. The system receiving this SYN packet acknowledges with a SYN/ACK. It sends an acknowledgment number back, which is based on the first sequence number plus one (that is, it increments the sequence number sent to it by one). It then also creates its own sequence number and sends that along with it. The original system receives the SYN/ACK with the new sequence number. It increments the sequence number by one and uses it as the acknowledgment number in the ACK packet with which it responds.

The difference in the difficulty of attempting a spoofing attack from inside a network and from outside involves determining the sequence number. If the attacker is inside of the network and can observe the traffic with which the target host responds, the attacker can easily see the sequence number the system creates and can respond with the correct sequence number.
sequence number. If the attacker is external to the network and the sequence number the
target system generates is not observed, it is next to impossible for the attacker to provide
the final ACK with the correct sequence number. So the attacker has to guess what the
sequence number might be.

Sequence numbers are somewhat predictable. Sequence numbers for each session are
not started from the same number, so that different packets from different concurrent
connections will not have the same sequence numbers. Instead, the sequence number for
each new connection is incremented by some large number to keep the numbers from
being the same. The sequence number may also be incremented by some large number
every second (or some other time period). An external attacker has to determine what
values are used for these increments. The attacker can do this by attempting connections
at various time intervals to observe how the sequence numbers are incremented. Once the
pattern is determined, the attacker can attempt a legitimate connection to determine
the current value, and then immediately attempt the spoofed connection. The spoofed
connection sequence number should be the legitimate connection incremented by the
determined value or values.

EXAM TIP The Security+ exam objective for attacks (1.2) is to compare
and contrast different attack types. This means that you need to be able
to differentiate attacks based on a set of given symptoms and indications.
Learning how these attacks are performed, what they look like, and how to
recognize specific attacks is essential for the exam.

Wireless Attacks

Wireless is a common networking technology that has a substantial number of standards
and processes to connect users to networks via a radio signal, freeing machines from
wires. As in all software systems, wireless networking is a target for hackers. This is partly
because of the simple fact that wireless removes the physical barrier.

Replay

A replay attack in wireless uses the same principle as replay attacks presented earlier in
the "Application/Service Attacks" section. By repeating information, one can try to get
repeated behavior from a system. Because wireless systems are not constrained by wires,
attackers can copy traffic rather easily between endpoints and the wireless access point.
Replay protections are essential in wireless systems to prevent exploitation of the open
signal.

EXAM TIP The best method for defending against replay attacks is through
the use of encryption and short time frames for legal transactions.
Encryption can protect the contents from being understood, and a short
time frame for a transaction prevents subsequent use.

IV

The initialization vector (IV) is used in wireless systems as the randomization element
at the beginning of a connection. Attacks against the IV aim to determine it, thus finding
the repeating key sequence.

The IV is the primary reason for the weaknesses in WEP. The IV is sent in the plain-
text part of the message, and because the total key space is approximately 16 million
keys, the same key will be reused. Once the key has been repeated, an attacker has two
cipher-texts encrypted with the same key stream. This allows the attacker to examine
the ciphertext and retrieve the key. This attack can be improved by examining only
packets that have weak IVs, reducing the number of packets needed to crack the key.
Using only weak IV packets, the number of required captured packets is reduced to
around four or five million, which can take only a few hours to capture on a fairly busy
access point (AP). For a point of reference, this means that equipment with an adver-
tised WEP key of 128 bits can be cracked in less than a day, whereas to crack a normal
128-bit key would take roughly 2,000,000,000,000,000,000,000 years on a computer able
to attempt one trillion keys a second. AirSnort is a modified sniffing program that
takes advantage of this weakness to retrieve the WEP keys. The biggest weakness of
WEP is that the IV problem exists regardless of key length, because the IV always
remains at 24 bits.

Evil Twin

The evil twin attack is in essence an attack against the wireless protocol via substitute
hardware. This attack uses an access point owned by an attacker that usually has been
enhanced with higher-power and higher-gain antennas to look like a better connection
to the users and computers attaching to it. By getting users to connect through
the evil access point, attackers can more easily analyze traffic and perform man-in-
the-middle-type attacks. For simple denial of service, an attacker could use interfere-
ence to jam the wireless signal, not allowing any computer to connect to the access
point successfully.

Rogue AP

By setting up a rogue access point, or rogue AP, an attacker can attempt to get clients to
connect to it as if it were authorized and then simply authenticate to the real AP, a simple
way to have access to the network and the client's credentials. Rogue APs can act as a
man in the middle and easily steal users' credentials. Enterprises with wireless APs should
routinely scan for and remove rogue APs, as users have difficulty avoiding them.

Jamming

Jamming is a form of denial of service that specifically targets the radio spectrum aspect
of wireless. Just as other DoS attacks can manipulate things behind the scenes, so can
Jamming on a wireless AP, enabling things such as attachment to a rogue AP.
WPS

Wi-Fi Protected Setup (WPS) is a network security standard that was created to provide users with an easy method of configuring wireless networks. Designed for home networks and small business networks, this standard involves the use of an eight-digit PIN to configure wireless devices. WPS consists of a series of Extensible Authentication Protocol (EAP) messages and has been shown to be susceptible to a brute force attack. A successful attack can reveal the PIN and subsequently the WPA/WPA2 passphrase and allow unauthorized parties to gain access to the network. Currently, the only effective mitigation is to disable WPS.

Bluejacking

Bluejacking is a term used for the sending of unauthorized messages to another Bluetooth device. This involves sending a message as a phonebook contact:

Then the attacker sends the message to the possible recipient via Bluetooth. Originally, this involved sending text messages, but more recent phones can send images or audio as well. A popular variant of this is the transmission of “shock” images, featuring disturbing or crude photos. As Bluetooth is a short-range protocol, the attack and victim must be within roughly 10 yards of each other. The victim’s phone must also have Bluetooth enabled and must be in discoverable mode. On some early phones, this was the default configuration, and while it makes connecting external devices easier, it also allows attacks against the phone. If Bluetooth is turned off, or if the device is set to nondiscourable, bluejacking can be avoided.

Bluesnarfing

Bluesnarfing is similar to bluejacking in that it uses the same contact transmission protocol. The difference is that instead of sending an unsolicited message to the victim’s phone, the attacker copies off the victim’s information, which can include e-mails, contact lists, calendar, and anything else that exists on that device. More recent phones with media capabilities can be snarled for private photos and videos. Bluesnarfing used to require a laptop with a Bluetooth adapter, making it relatively easy to identify a possible attacker, but bluesnarfing applications are now available for mobile devices. Moreover, a combination of Bluetooth and Floozer is one such application that runs as a Java applet. The majority of Bluetooth phones need to be discoverable for the bluesnarf attack to work, but it does not necessarily need to be paired. In theory, an attacker can also brute-force the device’s unique 48-bit name. A program called RedFang attempts to perform this brute force attack by sending all possible names and seeing what gets a response. This approach was addressed in Bluetooth 1.2 with an anonymity mode.

EXAM TIP: The Security+ exam objective is to compare and contrast attacks, and in the case of bluejacking and bluesnarfing, these are both attacks against Bluetooth. They differ in that bluejacking is the sending of unauthorized data via Bluetooth, whereas bluesnarfing is the unauthorized taking of data over a Bluetooth channel. Understanding this difference is important.

RFID

Radio frequency identification (RFID) tags are used in a wide range of use cases. From tracking devices to keys, the unique serialization of these remotely sensible devices has made them useful in a wide range of applications. RFID tags come in several different forms and can be classified as either active or passive. Active tags have a power source, while passive tags utilize the RF energy transmitted to them for power. RFID tags are used as a means of identification, and have the advantage over bar codes that they do not need to be visible, just within radio wave range, typically centimeters to 200 meters depending upon tag type. RFID tags are used in a range of security situations including contactless identification systems such as smart cards.

RFID tags have multiple security concerns. First and foremost, because they are connected via RF energy, physical security is a challenge. Security is an important issue for RFID tag systems because they form a means of identification and there is a need for authentication and confidentiality of the data transfers. There are several standards associated with the RFID data flow, including ISO/IEC 18000 and ISO/IEC 29167 for cryptography methods to support confidentiality, untraceability, tag and reader authentication, and over-the-air privacy, while ISO/IEC 20248 specifies a digital signature data structure for use in RFID systems.
There are several different attack types that can be performed against RFID systems:

- Against the RFID devices themselves, the chips and readers
- Against the communication channel between the device and the reader
- Against the reader and back-end system

The last type is more of a standard IT/IS attack depending upon the interfaces used (web, database, etc.) and is not covered any further. Attacks against the communication channel are relatively easy because the radio frequencies are known and devices exist to interface with tags. Two main attacks are replay and eavesdropping. In a replay attack, the RFID information is recorded and then replayed later. In the case of an RFID-based access badge, it could be read in a restaurant from a distance and then replayed at the appropriate entry point to gain entry. In the case of eavesdropping, the data can be collected, monitoring the movement of tags for whatever purpose needed by an unauthorized party. Both of these attacks are easily defeated using the ISO/IEC security standards previously listed.

If eavesdropping is possible, then what about man-in-the-middle attacks? These are certainly possible, as they would be a combination of a sniffing (eavesdropping) action, followed by a replay (spoofing) attack. This leads to the question of to whether an RFID can be cloned. And again, the answer is yes, if the RFID information is not protected via a cryptographic component.

**NFC**

*Near field communication (NFC)* is a set of wireless technologies that enables smartphones and other devices to establish radio communication over a short proximity, typically a distance of 10 cm (3.9 in) or less. This technology did not see much use until recently when it started being employed to move data between cell phones and in mobile payment systems. Now that NFC has become the mainstream method of payments via mobile phones, it is becoming ubiquitous, and in many cases connected directly to financial information. The importance of understanding and protecting this communication channel is paramount.

**Disassociation**

Disassociation attacks against a wireless system are attacks designed to disassociate a host from the wireless access point, and from the wireless network. Disassociation attacks stem from the deauthentication frame that is in the IEEE 802.11 (Wi-Fi) standard. The deauthentication frame is designed as a tool to remove unauthorized stations from a Wi-Fi access point, but because of the design of the protocol, they can be implemented by virtually anyone. An attacker only needs to have the MAC address of the intended victim, which enables them to send a spoofed message to the access point, specifically spoofing the MAC address of the victim machine. This results in the disconnection of the victim machine, making this attack a form of denial of service.

Disassociation attacks are not typically used alone, but rather in concert with another attack objective. For instance, if you disassociate a connection and then sniff the reconnection, you can steal passwords. After disassociating a machine, the user attempting to reestablish a WPA or WPA2 session will need to repeat the WPA four-way handshake. This gives the hacker a chance to sniff the event, the first stop in gathering needed information for a brute force or dictionary-based WPA password-cracking attack. Forcing users to reconnect creates a chance to mount a man-in-the-middle attack against content provided during a connection. This has been used by the WiPhisher tool to collect passwords.

### Cryptographic Attacks

Attacks against the cryptographic system are referred to as *cryptographic attacks*. These attacks are designed to take advantage of two specific weaknesses. First, users widely view cryptography as magic, or otherwise incomprehensible "stuff," leading them to trust the results without valid reasons. Second, although understood by computer scientists, algorithmic weaknesses that can be exploited are frequently overlooked by developers.

### Birthday

The *birthday attack* is a special type of brute force attack that gets its name from something known as the birthday paradox, which states that in a group of at least 23 people, the chance that two individuals will have the same birthday is greater than 50 percent. Mathematically, we can use the equation $1 - e^{-k^2/2n}$ (with $k$ equaling the size of the set of possible values), and in the birthday paradox, $k$ would be equal to 365 (the number of possible birthdays). This same phenomenon applies to passwords, with $k$ (number of passwords) being quite a bit larger.

### Known Plaintext/Ciphertext

If an attacker has the original plaintext and ciphertext for a message, then they can determine the key used through brute force attempts targeting the key space. These *known plaintext/ciphertext* attacks can be difficult to mitigate, as some messages are particularly prone to this problem. For example, by having known messages, such as the German weather reports, during WWII, the Allies were able to use cryptanalysis techniques to eventually determine the Enigma machine rotor combinations, leading to the breakdown of that system. Modern cryptographic algorithms have protections included in the implementations to guard against this form of attack. One is the use of large keyspaces, making the brute force spanning of the keyspaces, or even a significant portion of it, no longer possible.

### Password Attacks

The most common form of authentication is the user ID and password combination. While it is not inherently a poor mechanism for authentication, the combination can be attacked in several ways. Too often, these attacks yield favorable results for the attackers, not as a result of a weakness in the scheme but usually due to the user not following good password procedures.
Poor Password Choices

The least technical of the various password-attack techniques consists of the attacker simply attempting to guess the password of an authorized user of the system or network. It is simply surprising how often this simple method works, and the reason it does is because people are not as security-conscious as they should be. Users often choose passwords that are not particularly strong, and these passwords are often stored in a straightforward manner. This makes it easy for the attacker to try various combinations until the correct password is found.

Rainbow Tables

Rainbow tables are precomputed tables of hash values associated with passwords. Using rainbow tables can dramatically reduce the time needed to crack a given password. The basic idea behind rainbow tables is that a hash function is used to transform the password into a hash value, which is then stored in a database. When the attacker wants to crack the password, they simply look up the hash value in the database to find the original password.

Dictionary

Another method of determining passwords is to use a password-cracking program that uses a dictionary of known passwords and attempts to match them against the target password. This method is often used in conjunction with rainbow tables, as the attacker can use the rainbow tables to obtain a list of hash values for known passwords, and then use the dictionary to try to match these hash values against the target password.

Brute Force

If the user has selected a password that is not found in a dictionary, even if various numbers or special characters are substituted for letters, the only way the password can be cracked is for an attacker to attempt a brute force attack, in which the password-cracking program attempts all possible password combinations.

Online vs. Offline

When the brute force attack occurs in real time against a system, it is frequently done by using a single account with multiple examples of passwords. The use of a dictionary attack is limited by the systems response time and bandwidth.

Hybrid Attack

A hybrid attack is an attack that combines the preceding methods. Most cracking tools have this option built in, first attempting a dictionary attack, and then moving to brute force methods.

Collision

A collision attack is where two different inputs yield the same output of a hash function. Through the manipulation of data, creating subtle changes that are not visible to the user yet create different versions of a digital file and the creation of many different versions,
then using the birthday attack to find a collision between any two of the many versions, an attacker has a chance to create a file with changed visible content but identical hashes.

**Downgrade**

As part of a Transport Layer Security/Secure Sockets Layer (TLS/SSL) setup, there is a specification of the cipher suite to be employed. This is done to enable the highest form of encryption that both server and browser can support. In a **downgrade attack**, the attacker takes advantage of a frequently employed principle to support backward compatibility to downgrade the security to a lower or nonexistent state.

**Replay**

Replay attacks work against cryptographic systems like they do against other systems. If an attacker can record a series of packets and then replay them, what was valid before may well be valid again. There is a wide range of defenses against replay attacks, and as such this should not be an issue. But developers that do not follow best practices can create implementations that lack replay protections, enabling this attack path to persist.

**Weak Implementations**

Weak implementations are another problem associated with backward compatibility. The best example of this is SSL. SSL, in all of its versions, has now fallen to attackers. TLS, an equivalent methodology that does not suffer these weaknesses, is the obvious solution, yet many websites still employ SSL. Cryptography has long been described as an arms race between attackers and defenders, with multiple versions and improvements over the years. Whenever an older version is allowed to continue operation, there is a risk associated with weaker implementations.

**Chapter Review**

This chapter examined the attack methods used by hackers. Four major categories of attack were covered: social engineering attacks against the people/user component, application/service attacks against specific types of components, wireless attacks against the network connection, and cryptographic attacks.

The social engineering section examined phishing/spam phishing, whaling, vishing, tailgating, impersonation, dumpster diving, shoulder surfing, hoaxes, and watering hole attacks and then described the tools and principles that make these attacks so successful.

The section on application/service attacks first examined DoS, DDoS, man-in-the-middle, buffer overflows, injections, cross-site scripting, cross-site request forgery, privilege escalation, and ARP poisoning. The section continued with amplification attacks, DNS poisoning, domain hijacking, man-in-the-browser attacks, zero days, exploits, and pass the hash methods. The section concluded with hijacking methods, driver manipulation, and a wide array of spoofing techniques.

The wireless section covered replay, IV, evil twin, and rogue IP attacks. It also covered jamming, attacking WPS, bluejacking, bluelnarthing, attacks on RFID and NFC, and disassociation attacks.

The chapter concluded with a section on cryptographic attacks, including birthday attacks, known plaintext/cipher text attacks, password attacks, dictionary attacks, brute force, and hybrid methods. Collision, downgrade, replay, and weak implementations were also covered.

What is important to remember is that this material is designed to assist you in understanding CompTIA exam objective 1.7: Compare and contrast types of attacks. Be prepared to differentiate between the types of attacks.

**Questions**

To help you prepare further for the CompTIA Security+ exam, and to test your level of preparedness, answer the following questions and then check your answers against the correct answers at the end of the chapter.

1. While waiting in the lobby of your building for a guest, you notice a man in a red shirt standing close to a locked door with a large box in his hands. He waits for someone else to come along and open the locked door, then proceeds to follow her inside. What type of social engineering attack have you just witnessed?

   A. Impersonation
   B. Phishing
   C. Boxing
   D. Tailgating

2. A user reports seeing “odd certificate warnings” on her web browser this morning whenever she visits Google. Looking at her browser, you see certificate warnings. Looking at the network traffic, you see all HTTP and HTTPS requests from that system are being routed to the same IP regardless of destination. Which of the following attack types are you seeing in this case?

   A. Phishing
   B. Man-in-the-middle
   C. Cryptolocker
   D. DDoS

3. Users are reporting the wireless network on one side of the building is broken. They can connect, but can't seem to get to the Internet. While investigating, you notice all of the affected users are connecting to an access point you don't recognize. These users have fallen victim to what type of attack?

   A. Rogue AP
   B. WPS
   C. Bluejacking
   D. Disassociation
4. When an attacker captures network traffic and retransmits it at a later time, what type of attack are they attempting?
   A. Denial of service attack
   B. Replay attack
   C. Bluejacking attack
   D. Man-in-the-middle attack

5. What type of attack involves an attacker putting a layer of code between an original device driver and the operating system?
   A. Refactoring
   B. Trojan horse
   C. Shimming
   D. Pass the hash

6. A colleague asks you for advice on why he can't log in to his Gmail account. Looking at his browser, you see he has typed www.gmail.com in the address bar. The screen looks very similar to the Gmail login screen. Your colleague has just fallen victim to what type of attack?
   A. Jamming
   B. Rainbow table
   C. Whale phishing
   D. Typo squatting

7. You've been asked to try and crack the password of a disgruntled user who was recently fired. Which of the following could help you crack that password in the least amount of time?
   A. Rainbow tables
   B. Brute force
   C. Dictionary
   D. Hybrid attack

8. You're sitting at the airport when your friend gets a message on her phone. In the text is a picture of a duck with the word "Pwned" as the caption. Your friend doesn't know who sent the message. Your friend is a victim of what type of attack?
   A. Snarfing
   B. Bluejacking
   C. Quacking
   D. Collision

9. All of the wireless users on the third floor of your building are reporting issues with the network. Every 15 minutes, their devices disconnect from the network. Within a minute or so they are able to reconnect. What type of attack is most likely underway in this situation?
   A. WPS attack
   B. Downgrade attack
   C. Brute force attack
   D. Disassociation attack

10. Your organization's web server was just compromised despite being protected by a firewall and IPS. The web server is fully patched and properly configured according to industry best practices. The IPS logs show no unusual activity, but your network traffic logs show an unusual connection from an IP address belonging to a university. What type of attack is most likely occurring?
    A. Cross-site scripting attack
    B. Authority attack
    C. Zero day attack
    D. URL hijacking attack

11. Your e-commerce site is crashing under an extremely high traffic volume. Looking at the traffic logs, you see tens of thousands of requests for the same URL coming from hundreds of different IP addresses around the world. What type of attack are you facing?
    A. DoS
    B. DDoS
    C. DNS poisoning
    D. Snarfing

12. A user wants to know if the network is down, because she is unable to connect to anything. While troubleshooting, you notice the MAC address for her default gateway doesn't match the MAC address of your organization's router. What type of attack has been used against this user?
    A. Consensus attack
    B. ARP poisoning
    C. Refactoring
    D. Smurf attack
13. A user in your organization contacts you to see if there's any update to the “account compromise” that happened last week. When you ask him to explain what he means, the user tells you he received a phone call earlier in the week from your department and was asked to verify his user id and password. The user says he gave the caller his user id and password. This user has fallen victim to what specific type of attack?
A. Spear phishing
B. Voishing
C. Phishing
D. Replication

14. Coming into your office, you overhear a conversation between two security guards. One guard is telling the other she caught several people digging through the trash behind the building early this morning. The security guard says the people claimed to be looking for aluminum cans, but only had a bag of papers—no cans. What type of attack has this security guard witnessed?
A. Spear phishing
B. Pharming
C. Dumpster diving
D. Combining

15. A user calls to report a problem with an application you support. The user says when she accidentally pasted an entire paragraph into an input field, the application crashed. You are able to consistently reproduce the results using the same method. What vulnerability might that user have accidentally discovered in that application?
A. Poison apple
B. Shoulder surfing
C. Smurfing
D. Buffer overflow

Answers
1. D. Tailing (or piggybacking) is the simple tactic of following closely behind a person who has just used their own access card, key, or PIN to gain physical access to a room or building. The large box clearly impedes the person in the red shirt’s ability to open the door, so they let someone else do it for them and follow them in.
2. B. This is most likely some type of man-in-the-middle attack. This attack method is usually done by routing all of the victim’s traffic to the attacker’s host, where the attacker can view it, modify it, or block it. The attacker inserts himself into the middle of his victim’s network communications.
3. A. This is a rogue AP attack. Attackers set up their own access points in an attempt to get wireless devices to connect to the rogue AP instead of the authorized access points.
4. B. A replay attack occurs when the attacker captures a portion of the communication between two parties and retransmits it at a later time. For example, an attacker might replay a series of commands and codes used in a financial transaction to cause the transaction to be conducted multiple times. Generally, replay attacks are associated with attempts to circumvent authentication mechanisms, such as the capturing and reuse of a certificate or ticket.
5. C. Skimming is the process of putting a layer of code between the device driver and the operating system.
6. D. Typo squatting capitalizes on common typing errors, such as gmai instead of gmail. The attacker registers a domain very similar to the real domain and attempts to collect credentials or other sensitive information from unsuspecting users.
7. A. Rainbow tables are precomputed tables or hash values associated with passwords. When used correctly in the right circumstances, they can dramatically reduce the amount of work needed to crack a given password.
8. B. This is most likely a bluejacking attack. If a victim’s phone has Bluetooth enabled and is in discoverable mode, it may be possible for an attacker to send unwanted texts, images, or audio to the victim’s phone.
9. D. Disassociation attacks against a wireless system are attacks designed to dissociate the victim from the wireless access point. If an attacker has a list of MAC addresses for the wireless devices, they can spoof deauthentication frames, causing the wireless devices to disconnect from the network.
10. C. If a “properly secured” and patched system is suddenly compromised, it is most likely the result of a zero day attack. A zero day attack is one that uses a vulnerability for which there is no previous knowledge outside of the attacker.
11. B. This is a DDoS attack. DDoS (or distributed denial of service) attacks attempt to overwhelm their targets with traffic from many different sources. Botnets are quite commonly used to launch DDoS attacks.
12. B. ARP poisoning is an attack that involves sending spoofed ARP or RARP responses to a victim in an attempt to alter the ARP table on the victim’s system. If successful, an ARP poisoning attack will replace one of more MAC addresses in the victim’s ARP table with the MAC address of the attacker’s spoofed responses.
13. B. Voishing is a social engineering attack that uses voice communication technology to obtain the information the attacker is seeking. Most often the attacker will call a victim and pretend to be someone else in an attempt to extract information from the victim.
14. C. Dumpster diving is the process of going through a target's trash in the hopes of finding valuable information such as user lists, directories, organization charts, network maps, passwords, and so on.

15. D. This user may have discovered a buffer overflow vulnerability in the application. A buffer overflow can occur when more input is supplied than the program is designed to process (for example, 150 characters supplied to a 10-character input field). If the application doesn't reject the additional input, the extra characters can continue to fill up memory and overwrite other portions of the program, causing instability or undesirable results.

---

**CHAPTER 3**

**Threat Actors**

In this chapter, you will
- Explain threat actor types
- Explain threat actor attributes
- Explore open source threat intelligence

This chapter examines the types and attributes of threat actors. Threat actors can take many forms and represent different levels of threats, so understanding the type and motivation of an actor can be critical for proper defensive measures. The chapter also explores open source intelligence as used to understand the current threat environment, and to lead the proper resource deployment of cyber defenses.

**Certification Objective** This chapter covers CompTIA Security+ exam objective 1.3, Explain threat actor types and attributes.

**Types of Actors**

The act of deliberately accessing computer systems and networks without authorization is generally referred to as hacking, with individuals who conduct this activity being referred to as hackers. The term "hacking" also applies to the act of exceeding one's authority in a system. This would include authorized users who attempt to gain access to files they aren't permitted to access or who attempts to obtain permissions that they have not been granted. While the act of breaking into computer systems and networks has been glorified in the media and movies, the physical act does not live up to the Hollywood hype. Intruders are, if nothing else, extremely patient, since the process to gain access to a system takes persistence and dogged determination. The attacker will conduct many pre-attack activities in order to obtain the information needed to determine which attack will most likely be successful. Typically, by the time an attack is launched, the attacker will have gathered enough information to be very confident that the attack will succeed.