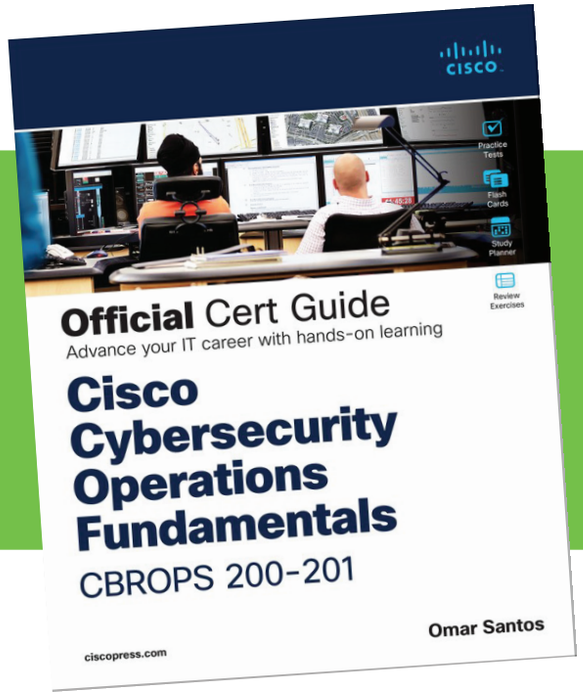
## Sample Chapter

# Cisco Cybersecurity Associate CBROPS 200-201 Official Cert Guide by Omar Santos




Special Offer: Save 40% on Cisco Cybersecurity Associate CBROPS 200-201 Official Cert Guide book or Premium Edition eBook. Use code **CBROPS40** during checkout to apply discount. Shop now at ciscopress.com/CBROPS.

* Discount code CBROPS40 confers a 40% discount off the list price of Cisco Cybersecurity Associate CBROPS 200-201 Official Cert Guide book or Premium Edition eBook when purchased on ciscopress.com. Discount code may not be combined with any other offer and is not redeemable for cash. Offer subject to change.

CHAPTER 4

# Types of Attacks and Vulnerabilities

**This chapter covers the following topics:**

Types of Attacks

Types of Vulnerabilities

The sophistication of cybersecurity attacks is increasing every day. In addition, there are numerous types of cybersecurity attacks and vulnerabilities. This chapter covers the most common.

## "Do I Know This Already?" Quiz

The "Do I Know This Already?" quiz allows you to assess whether you should read this entire chapter thoroughly or jump to the "Exam Preparation Tasks" section. If you are in doubt about your answers to these questions or your own assessment of your knowledge of the topics, read the entire chapter. Table 4-1 lists the major headings in this chapter and their corresponding "Do I Know This Already?" quiz questions. You can find the answers in Appendix A, "Answers to the 'Do I Know This Already?' Quizzes and Review Questions."

**Table 4-1** "Do I Know This Already?" Foundation Topics Section-to-Question Mapping

| Foundation Topics Section | Questions |
|---|---|
| Types of Attacks | 1–5 |
| Types of Vulnerabilities | 6–8 |

**CAUTION** The goal of self-assessment is to gauge your mastery of the topics in this chapter. If you do not know the answer to a question or are only partially sure of the answer, you should mark that question as wrong for purposes of the self-assessment. Giving yourself credit for an answer you correctly guess skews your self-assessment results and might provide you with a false sense of security.

1. Which of the following are examples of vulnerability and port scanners? (Select all that apply.)
   - **a.** SuperScan
   - **b.** nmap
   - **c.** Nexpose
   - **d.** Nessus

2. How do UDP scans work?
   a. By establishing a three-way handshake.
   b. By sending SYN packets to see what ports are open.
   c. By relying on ICMP "port unreachable" messages to determine whether a port is open. When the scanner sends a UDP packet and the port is not open on the victim's system, that system will respond with an ICMP "port unreachable" message.
   d. By sending ICMP "port unreachable" messages to the victim.
3. What is a phishing attack?
   a. A phishing attack is the act of incorporating malicious ads on trusted websites, which results in users' browsers being inadvertently redirected to sites hosting malware.
   b. A phishing attack uses SQL injection vulnerabilities to execute malicious code.
   c. This is a type of denial-of-service (DoS) attack where the attacker sends numerous phishing requests to the victim.
   d. This is a type of attack where the attacker sends an email and often presents a link that looks like a valid, trusted resource to a user. After clicking it, the user is prompted to disclose confidential information such as username and password.
4. What is a backdoor?
   a. A backdoor is a social engineering attack to get access back to the victim.
   b. A backdoor is a privilege escalation attack designed to get access from the victim.
   c. A backdoor is an application or code used by an attacker either to allow future access or to collect information to use in further attacks.
   d. A backdoor is malware installed using man-in-the-middle attacks.
5. What is an amplification attack?
   a. An amplification attack is a form of directed DDoS attack in which the attacker's packets are sent at a much faster rate than the victim's packets.
   b. An amplification attack is a form of reflected attack in which the response traffic (sent by the unwitting participant) is made up of packets that are much larger than those that were initially sent by the attacker (spoofing the victim).
   c. An amplification attack is a type of man-in-the-middle attack.
   d. An amplification attack is a type of data exfiltration attack.
6. What is a buffer overflow?
   a. In a buffer overflow, a program or software cannot write data in a buffer, causing the application to crash.
   b. In a buffer overflow, a program or software sends the contents of the buffer to an attacker.
   c. In a buffer overflow, an attacker overflows a program with numerous packets to cause a denial-of-service condition.
   d. In a buffer overflow, a program or software puts more data in a buffer than it can hold, or a program tries to put data in a memory location past a buffer.

7. What is a cross-site scripting (XSS) vulnerability?
   a. A type of web application vulnerability where malicious scripts are injected into legitimate and trusted websites
   b. A type of cross-domain hijack vulnerability
   c. A type of vulnerability that leverages the crossing of scripts in an application
   d. A type of cross-site request forgery (CSRF) vulnerability that is used to steal information from the network
8. What is a SQL injection vulnerability?
   a. A type of vulnerability where an attacker can insert or "inject" a SQL query via the input data from the client to the application or database
   b. A type of vulnerability where an attacker can "inject" a new password to a SQL server or the client
   c. A type of DoS vulnerability that can cause a SQL server to crash
   d. A type of privilege escalation vulnerability aimed at SQL servers

# Foundation Topics

## Types of Attacks

As you probably already know, most attackers do not want to be discovered, so they use a variety of techniques to remain in the shadows when attempting to compromise a network. The following sections describe the most common types of attacks carried out by threat actors.

### Reconnaissance Attacks

Reconnaissance attacks include the discovery process used to find information about the network, users, and victims. They could include scans of the network to find out which IP addresses respond and further scans to see which ports on the devices at these IP addresses are open. This is usually the first step taken to discover what is on the network and to determine what vulnerabilities to exploit.

Key Topic

Reconnaissance can be passive or active. Passive reconnaissance can be carried out by an attacker just researching information about the victim's public records, social media sites, and other technical information, such as DNS, whois, and sites such as Shodan (www.shodan.io). The attacker can use tools such as Maltego, Recon-ng, TheHarvester, Spiderfoot, and many others to accelerate this "research."

For instance, the Shodan search engine is a powerful database of prescanned networked devices connected to the Internet. It consists of scan results including banners collected from port scans of public IP addresses, with fingerprints of services like Telnet, FTP, HTTP, and other applications.

Shodan creates risk by providing both attackers and defenders a prescanned inventory of devices connected to public IP addresses on the Internet. For example, when a new

vulnerability is discovered and published, an attacker can quickly and easily search Shodan for vulnerable devices and then launch an attack. Attackers can also search the Shodan database for devices with poor configurations or other weaknesses, all without actively scanning.

Using Shodan search filters, a user can really narrow down search results, by country code or CIDR netblock, for example. Shodan application programming interfaces (APIs) and some basic scripting can enable many search queries and subsequent actions (for example, a weekly query of newly discovered IPs scanned by Shodan on a CIDR netblock that runs automatically and is emailed to the security team).

Remember that public IP addresses are constantly probed and scanned already. By using Shodan, you are not scanning because Shodan has already scanned these IPs. Shodan is a tool, and it can be used for good or evil. To mitigate risk, you can take tangible steps like registering for a free Shodan account, searching for your organization's public IPs, and informing the right network and security people of the risks of your organization's Shodan exposure. You can learn more at www.shodan.io.



Figure 4-1 shows an example of a query performed at the Shodan website to search for all known devices connected to the Internet with Telnet enabled.

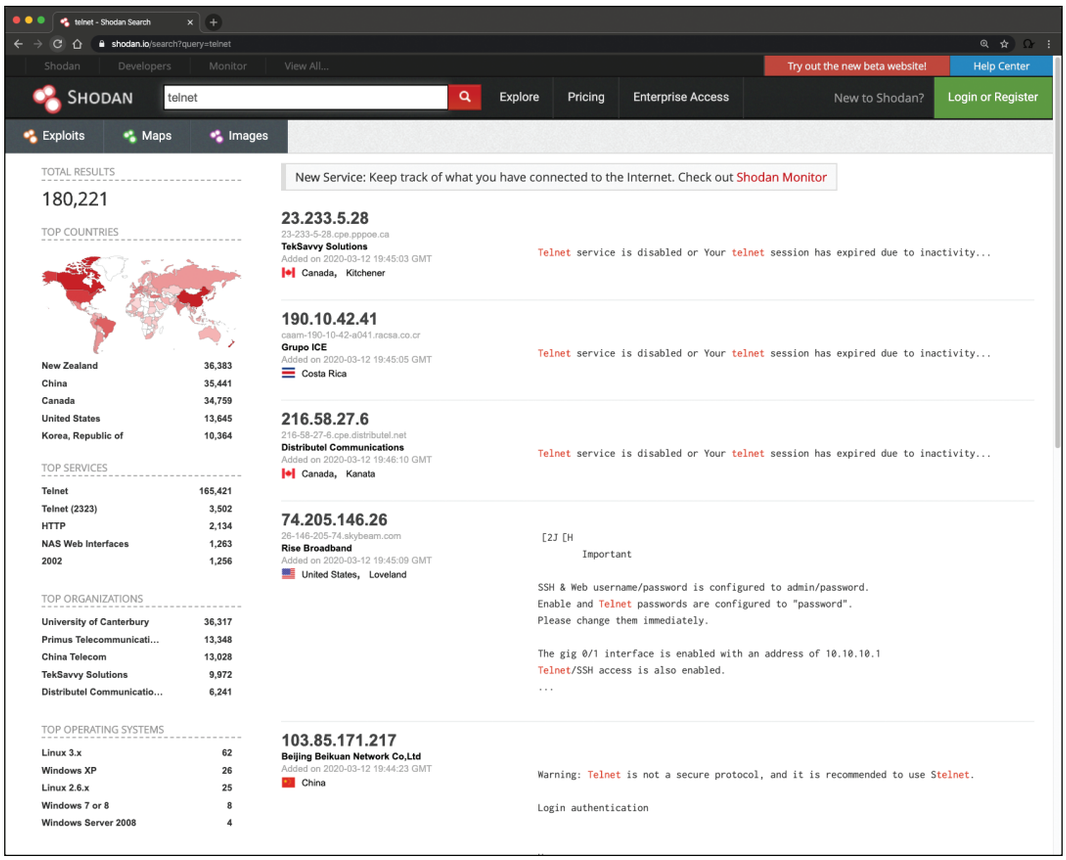



**Figure 4-1** *Shodan Search Engine Results Example*

Key Topic

**TIP** Open-source intelligence (OSINT) gathering is a method of gathering publicly available intelligence sources to collect and analyze information about a target. Open-source intelligence is "open source" because collecting the information does not require any type of covert method. Typically, the information can be found on the Internet. The larger the online presence of the target, the more information that will be available. This type of collection can often start with a simple Google search, which can reveal a significant amount of information about a target. It will at least give you enough information to know what direction to go with your information-gathering process. The following sections look at some of the sources that can be used for OSINT gathering. Several examples of tools and methodologies for OSINT and passive reconnaissance are included at my GitHub repository for your reference: https://github.com/The-Art-of-Hacking/h4cker/tree/master/osint.

Active reconnaissance is carried out by tools called *scanners*. The following are a few commercial and open-source application, port, and vulnerability scanners:

- AppScan by IBM
- Burp Suite Professional by PortSwigger
- Hailstorm by Cenzic
- N-Stalker by N-Stalker
- Nessus by Tenable Network Security
- NetSparker by Mavituna Security
- NeXpose by Rapid7
- nmap (open-source port scanner)
- nikto (open-source web application scanner)
- OWASP Zed Attack Proxy (open-source web application scanner, proxy, and attack platform maintained by the Open Web Application Security Project [OWASP])
- Qualys
- Retina Web Security Scanner by eEye Digital Security
- Sentinel by WhiteHat
- Veracode Web Application Security by Veracode
- VUPEN Web Application Security Scanner by VUPEN Security
- WebApp360 by nCircle

**TIP** Be aware that attacks are launched not only from individuals outside your company; they are also launched from people and devices inside your company, maliciously and otherwise, who have current, legitimate user accounts. This vector is of particular concern these days with the proliferation of organizations allowing employees to use their personal devices—known as bring your own device (BYOD)—to seamlessly access data, applications, and devices on the corporate networks. Perhaps the user is curious, or maybe a backdoor is installed on the computer on which the user is logged in. In either case, it is important to implement a security policy that takes nothing for granted and to be prepared to mitigate risk at several levels.

There are different types of port- and network-scanning techniques. The following are the most common:



Key Topic

- **Basic port scan:** This type of scan involves scanning a predetermined TCP/UDP port by sending a specifically configured packet that contains the port number of the port that was selected. This is typically used to determine what ports are "open" or available in a given system.
- **TCP scan:** This is a TCP-based scan of a series of ports on a machine to determine port availability. If a port on the machine is listening, the TCP "connect" is successful in reaching that specific port. Earlier, you learned that nmap is an open-source scanner; nmap refers to TCP scans as "connect scans," which is named after the UNIX **connect()** system call. If the scanner finds that a port is open, the victim operating system completes the TCP three-way handshake. In some cases, the port scanner will close the connection to avoid a denial-of-service condition.

  TCP SYN scan is one of the most common types of TCP scanning, and it is also referred to as "half-open scanning" because it never actually opens a full TCP connection. The scanner sends a SYN packet, and if the target responds with a SYN-ACK packet, the scanner typically responds with an RST packet.

  Another TCP scan type is TCP ACK. This type of scan does not exactly determine whether the TCP port is open or closed; instead, it checks whether the port is filtered or unfiltered. TCP ACK scans are typically used when trying to see if a firewall is deployed and its rule sets. There are also TCP FIN packets that in some cases can bypass legacy firewalls because closed ports may cause a system to reply to a FIN packet with a corresponding RST packet due to the nature of TCP.
- **UDP scan:** Because UDP is a connectionless protocol and does not have a three-way handshake like TCP, the UDP scans have to rely on ICMP "port unreachable" messages to determine if the port is open. When the scanner sends a UDP packet and the port is not open on the victim, the victim's system will respond with an ICMP "port unreachable" message. This type of scanning will be affected by firewalls and ICMP rate limiting.
- **Strobe scan:** Typically, attackers use this type of scan to find the ports that they already know how to exploit. Strobe scans execute on a more confined level.
- **Stealth scan:** This type of scan is designed to go undetected by network auditing tools.

Example 4-1 shows a basic nmap scan against a Linux machine (172.18.104.139).

**Example 4-1** *Nmap Scanner Example*

```
bash-3.2$ sudo nmap -sS 172.18.104.139
Password: ****************
Starting Nmap 7.12 ( https://nmap.org ) at 2016-09-06 11:13 EDT
Nmap scan report for 172.18.104.139
Host is up (0.024s latency).
Not shown: 995 closed ports
PORT    STATE SERVICE
22/tcp  open  ssh
25/tcp  open  smtp
80/tcp  open  http
110/tcp open  pop3
143/tcp open  imap
Nmap done: 1 IP address (1 host up) scanned in 1.26 seconds
```

In Example 4-1, the host (172.18.104.139) is listening to TCP ports 22, 25, 80, 110, and 143.

Example 4-2 shows how to perform a "ping sweep" using nmap to see what systems are present in a given subnet (in this example, 172.18.104.129/29).

**Example 4-2** *Nmap Ping Sweep Example*

```
bash-3.2$ nmap -sP 172.18.104.129/29
Starting Nmap 7.12 ( https://nmap.org ) at 2016-09-06 11:22 EDT
Nmap scan report for 172.18.104.129
Host is up (0.0071s latency).
Nmap scan report for 172.18.104.130
Host is up (0.0076s latency).
Nmap scan report for 172.18.104.132
Host is up (0.0076s latency).
Nmap scan report for 172.18.104.133
Host is up (0.0079s latency).
Nmap scan report for 172.18.104.134
Host is up (0.0074s latency).
Nmap scan report for 172.18.104.135
Host is up (0.011s latency).
Nmap done: 8 IP addresses (6 hosts up) scanned in 3.75 seconds
```

The following are some of the more popular port-scanning techniques:

- **TCP Full Connect scan:** This type of scan is the most reliable although it is also the most detectable. It is easily logged and detected because a full connection is established. Open ports reply with a SYN/ACK, and closed ports respond with an RST/ACK.
- **TCP SYN scan:** This type of scan is known as half open because a full TCP three-way connection is not established. This scan was originally developed to be stealthy and

evade intrusion detection systems (IDSs) although most now detect it. Open ports reply with a SYN/ACK, and closed ports respond with an RST/ACK.

- **TCP FIN scan:** Forget trying to set up a connection; this technique jumps straight to the shutdown. This type of scan sends a FIN packet to the target port. An open port should return no response. Closed ports should send back an RST/ACK. This technique is usually effective only on UNIX devices or those compliant to RFC 793.
- **TCP NULL scan:** Sure, there should be some type of flag in the packet, but a NULL scan sends a packet with no flags set. If the OS has implemented TCP per RFC 793, open ports send no reply, whereas closed ports return an RST.
- **TCP ACK scan:** This scan attempts to determine access control list (ACL) rule sets or identify if a firewall or simply stateless inspection is being used. A stateful firewall should return no response. If an ICMP destination is unreachable, and a "communication administratively prohibited" message is returned, the port is considered to be filtered. If an RST is returned, no firewall is present.
- **TCP XMAS scan:** Sorry, there are no Christmas presents here, just a port scan that has toggled on the FIN, URG, and PSH flags. Open ports should provide no response. Closed ports should return an RST. Systems must be designed per RFC 793 for this scan to work, as is common for Linux. It does not work against Windows computers.
- **ACK scan:** This scan sends an ACK probe with random sequence numbers. ICMP type 3 code 13 responses may mean that stateless firewalls are being used, and an RST can mean that the port is not filtered.
- **FTP Bounce scan:** This type of scan uses an FTP server to bounce packets and make the scan harder to trace.
- **RPC scan:** This scan attempts to determine whether open ports are RPC ports.
- **Window scan:** Similar to an ACK scan, this scan can sometimes determine open ports. It does so by examining the TCP window size of returned RST packets. On some systems, open ports return a positive window size, and closed ones return a zero window size.
- **UDP scan:** UDP is unlike TCP. TCP is built on robust connections, but UDP is based on speed. With TCP, the hacker can manipulate flags in an attempt to generate a TCP response or an error message from ICMP. UDP does not have flags, nor does it typically issue responses. Some protocols use UDP, such as the Internet Key Exchange (IKE) protocol and DNS, where a host may issue a response (UDP packet) back to the originator. However, most other UDP implementations do not reply back with another UDP message because UDP is a connectionless protocol. It's a fire-and-forget protocol! The most you can hope for is a response from ICMP. If the port is closed, ICMP attempts to send an "ICMP type 3 code 3 port unreachable" message to the source of the UDP scan. But if the network is blocking ICMP, no error message is returned. Therefore, the response to the scans might simply be no response. If you are planning on doing UDP scans, plan for unreliable results.
- **ICMP scan:** These scans are typically used for "ping sweeps" to discover what devices may be in the network, as you saw in Example 4-2.

**TIP** Additional examples and details about all the different nmap scanner options can be obtained at my GitHub repository at http://h4cker.org/nmap.

## Social Engineering

Social engineering attacks leverage the weakest link, which is the human user. If the attacker can get the user to reveal information, it is much easier for the attacker to cause harm rather than use some other method of reconnaissance. This could be done through email or misdirection of web pages, which results in the user clicking something that leads to the attacker gaining information. Social engineering can also be done in person by an insider or outside entity or over the phone.

A primary example is attackers leveraging normal user behavior. Suppose you are a security professional who is in charge of the network firewalls and other security infrastructure equipment in your company. An attacker could post a job offer for a lucrative position and make it very attractive to you, the victim. Say that the job description lists benefits and compensation far beyond what you are already making at your company. You decide to apply for the position. The criminal (attacker) then schedules an interview with you. Because you are likely to show off your skills and work, the attacker may ask you how you configured the firewalls and other network infrastructure devices for your company. You might disclose information about the firewalls used in your network, how you configured them, how they were designed, and so on. This disclosure gives the attacker a lot of knowledge about the organization without even performing any type of scanning or reconnaissance on the network.

Other social engineering techniques include the following:

Key Topic

- **Phishing:** The attacker presents a link that looks like a valid, trusted resource to a user. When the user clicks it, he or she is prompted to disclose confidential information such as username and password.
- **Spear phishing:** This is a special class of phishing. It is a phishing attack that is constructed in a specific way and directly targeted at specific individuals or companies. The attacker studies a victim and the victim's organization to be able to make emails look legitimate and perhaps make them appear to come from trusted users within the corporation.
- **Pharming:** *Pharming* is the term used to describe a threat actor redirecting a victim from a valid website or resource to a malicious one that could be made to appear as the valid site to the user. From there, an attempt is made to extract confidential information from the user or to install malware in the victim's system. Pharming can be done by altering the host file on a victim's system, through DNS poisoning, or by exploiting a vulnerability in a DNS server.
- **Malvertising:** This is the act of incorporating malicious ads on trusted websites, which results in users' browsers being inadvertently redirected to sites hosting malware.
- **SMS phishing:** Because phishing has been an effective tactic for threat actors, they have found ways other than using email to fool their victims into following malicious

links or activating malware from emails. A number of phishing campaigns have used Short Message Service (SMS) to send malware or malicious links to mobile devices. One example of SMS phishing is the bitcoin-related SMS scams that have surfaced in recent years. Numerous victims have received messages instructing them to click links to confirm their accounts and claim bitcoins. When users click such a link, they might be fooled into entering sensitive information on that attacker's site.

- **Voice phishing (or vishing):** *Vishing* is the name for a social engineering attack carried out over a phone conversation. The attacker persuades users to reveal private, personal, and financial information or information about another person or a company. Voice phishing is typically used to steal credit card numbers or other information used in identity theft schemes. Attackers might impersonate and spoof caller ID to obfuscate themselves when performing voice phishing attacks.

- **Whaling:** Whaling is similar to phishing and spear phishing; however, with whaling, the attack is targeted at high-profile business executives and key individuals in a corporation. So, what is the difference between whaling and spear phishing? Like threat actors conducting spear phishing attacks, threat actors conducting whaling attacks also create emails and web pages to serve malware or collect sensitive information; however, the whaling attackers' emails and pages have a more official or serious look and feel. Whaling emails are designed to look like critical business emails or something from someone who has legitimate authority, either externally or even internally in the company itself. In whaling attacks, web pages are designed to specifically address high-profile victims. In a regular phishing attack, the email might be a faked warning from a bank or service provider. In whaling attacks, the email or a web page would be created with a more serious executive-level form. The content is created to target an upper manager, such as the CEO, or an individual who might have credentials for valuable accounts within the organization. In summary, a whaling attack takes additional steps to target and entice higher-profile victims. The main goal in whaling attacks is to steal sensitive information or compromise the victim's system and then target other key high-profile victims.

- **Elicitation, interrogation, and impersonation (Pretexting):** How someone influences, interrogates, and impersonates others are key components of social engineering. In short, elicitation is the act of gaining knowledge or information from people. In most cases, an attacker gets information from the victim without directly asking for that particular information. How an attacker interrogates and interacts with a victim is crucial for the success of the social engineering campaign. An interrogator can ask good open-ended questions to learn about an individual's viewpoints, values, and goals. The interrogator can then use any information revealed to continue to gather additional information or to obtain information from another victim. It is also possible for an interrogator to use closed-ended questions to get more control of the conversation, to lead the conversation, or to stop the conversation. Asking too many questions can cause the victim to shut down the interaction, and asking too few questions might seem awkward. Successful social engineering interrogators use a narrowing approach in their questioning to gain the most information from the victim. With pretexting

(or impersonation) an attacker presents as someone else to gain access to information. In some cases, it can be very simple, such as quickly pretending to be someone else within an organization; in other cases, it can involve creating a whole new identity and then using that identity to manipulate the receipt of information. Social engineers might use pretexting to impersonate individuals in certain jobs and roles, even if they do not have experience in those jobs or roles.

A security-aware culture must include ongoing training that consistently informs employees about the latest security threats, as well as policies and procedures that reflect the overall vision and mission of corporate information security. This emphasis on security helps employees understand the potential risk of social engineering threats, how they can prevent successful attacks, and why their role within the security culture is vital to corporate health. Security-aware employees are better prepared to recognize and avoid rapidly changing and increasingly sophisticated social engineering attacks and are more willing to take ownership of security responsibilities.

Official security policies and procedures take the guesswork out of operations and help employees make the right security decisions. Such policies include the following:

- **Password management:** Guidelines such as the number and type of characters that each password must include, how often a password must be changed, and even a simple declaration that employees should not disclose passwords to anyone (even if they believe they are speaking with someone at the corporate help desk) will help secure information assets.
- **Multifactor authentication (MFA):** Authentication for high-risk network services such as critical systems, web applications, and VPNs should use multifactor authentication rather than fixed passwords.
- **Antimalware defenses:** Multiple layers of antivirus defenses, such as at mail gateways and end-user desktops, can minimize the threat of phishing and other social engineering attacks.
- **Change management:** A documented change management process is more secure than an ad hoc process, which is more easily exploited by an attacker who claims to be in a crisis.
- **Information classification:** A classification policy should clearly describe what information is considered sensitive and how to label and handle it.
- **Document handling and destruction:** Sensitive documents and media must be securely disposed of and not simply thrown out with the regular office trash.
- **Physical security:** The organization should have effective physical security controls such as visitor logs, escort requirements, and background checks.


## Privilege Escalation Attacks

Privilege escalation is a type of attack and also a type of vulnerability. Privilege escalation is the process of taking some level of access (whether authorized or not) and achieving an even greater level of access (elevating the user's privileges). An example is an attacker who gains

user-mode access to a firewall, router, or server and then uses a brute-force attack against the system that provides administrative access. Privilege escalation can occur because a bug, misconfiguration, or vulnerability in an application or operating system enables a hacker to gain access to resources that normally would have been protected from an average user. The end result of privilege escalation is that the application performs actions that are running within a higher security context than intended by the designer, and the hacker is granted full access and control.


## Backdoors

When threat actors gain access to a system, they usually want future access as well, and they want it to be easy. The attackers can install a backdoor application to either allow future access or collect information to use in further attacks.

Many backdoors are installed by users clicking something without realizing that the link they clicked or the file they opened is a threat. Backdoors can also be implemented as a result of a virus, worm, or malware.




## Buffer Overflows and Code Execution

When threat actors gain access to a system, they also might be able to take several actions. The type of action depends on the level of access the threat actor has, or can achieve, and is based on permissions granted to the account compromised by the attacker. One of the most devastating actions available to an attacker is the ability to execute code within a device. Code execution could result in an adverse impact to the confidentiality, integrity, and availability of the system or network. Remote code execution (RCE) allows attackers to fully compromise the confidentiality, integrity, and availability of a system remotely (network hops away from the victim).

Buffer overflows can lead to code execution. Buffer overflows are categorized into two types: heap and stack. A heap is a memory space that is dynamically allocated. A buffer is a temporary data storage area whose length is defined in the program that creates it or by the operating system. Heap-based buffer overflows are different from stack-based buffer overflows in that the stack-based buffer overflow depends on overflowing a fixed-length buffer. A heap overflow is a type of buffer overflow that occurs in the heap data area and attempts to overwrite internal structures such as linked list pointers.

Buffers have a finite amount of space allocated for any one task. For example, if you allocate a 14-character buffer and then attempt to stuff 32 characters into it, you're going to have a real problem. Ideally, programs should be written to check that you cannot stuff more than 14 characters into the buffer; however, this type of error checking does not always occur. Error checking is really nothing more than making sure that buffers receive the type and amount of information required.

A buffer overflow vulnerability typically involves many memory manipulation functions in languages such as C and C++, where the program does not perform bounds checking and can easily overwrite the allocated bounds of such buffers. A perfect example is a **strncpy()** function, which can cause vulnerabilities when used incorrectly.

Let's look at Figure 4-2, where the sample code shows a buffer that includes a small chunk of data (HELLO WORLD).

```
struct my_struct {
        char my_buffer[14];
        struct my_struct*next_struct;
};
```

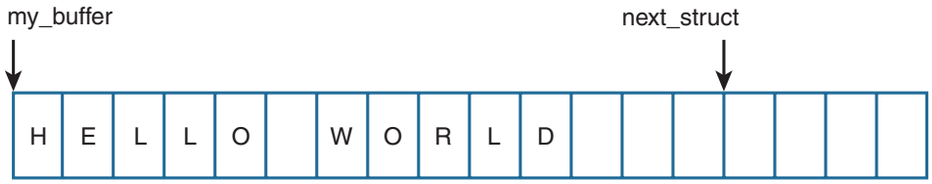



**Figure 4-2** *A Buffer Example*

An attacker can take advantage of this vulnerability and send data that can put data in a memory location past that buffer, as shown in Figure 4-3.

```
struct my_struct{
        char my_buffer[14];
        struct my_struct*next_struct;
};
```

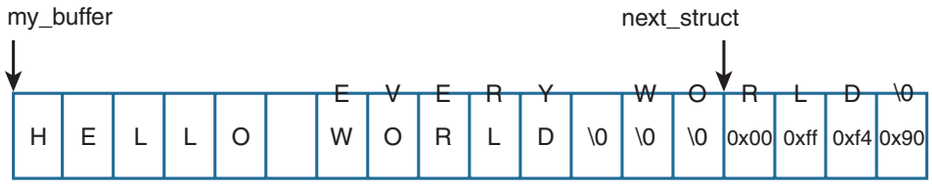



**Figure 4-3** *A Buffer Overflow*

In Figure 4-3, the attacker sent data (EVERY WORLD) that was more than the buffer could hold, causing it to subsequently write to the adjacent memory location. This simplistic example represents how an attacker could then write instructions to the system and potentially cause a local or remote code execution. In several of these attacks, the attacker writes "shellcode" to invoke instructions and manipulate the system.

The easiest way to prevent buffer overflows is to stop accepting data when the buffer is filled. This task can be accomplished by adding boundary protection. C programs are especially susceptible to buffer overflow attacks because C has many functions that do not properly check for boundaries.

A "return-to-libc" (or ret2libc) attack typically starts with a buffer overflow. In this type of attack, a subroutine return address on a call stack is replaced by an address of a subroutine that is already present in the executable memory of the process. This is done to potentially bypass the no-execute (NX) bit feature and allow attackers to inject their own code.

Operating systems that support nonexecutable stack help protect against code execution after a buffer overflow vulnerability is exploited. On the other hand, a nonexecutable stack cannot prevent a ret2libc attack because in this attack only existing executable code is used.

Another technique, called *stack-smashing protection*, can prevent or obstruct code execution exploitation because it can detect the corruption of the stack and can potentially flush out the compromised segment.

A technique called *ASCII armoring* can be used to mitigate ret2libc attacks. When you implement ASCII armoring, the address of every system library (such as libc) contains a NULL byte (0x00) that you insert in the first 0x01010101 bytes of memory. This is typically a few pages more than 16 MB and is called the ASCII armor region because every address up to (but not including) this value contains at least one NULL byte. When this methodology is implemented, an attacker cannot place code containing those addresses using string manipulation functions such as **strcpy()**.

Of course, this technique doesn't protect the system if the attacker finds a way to overflow NULL bytes into the stack. A better approach is to use the address space layout randomization (ASLR) technique, which mitigates the attack on 64-bit systems. When you implement ASLR, the memory locations of functions are random. ASLR is not very effective in 32-bit systems, though, because only 16 bits are available for randomization, and an attacker can defeat such a system by using brute-force attacks.



Key Topic

## Man-in-the Middle Attacks

A man-in-the-middle attack results when attackers place themselves in line between two devices that are communicating, with the intent of performing reconnaissance or manipulating the data as it moves between the devices. This can happen at Layer 2 or Layer 3. The main purpose is eavesdropping, so an attacker can see all the traffic.

If this happens at Layer 2, the attacker spoofs Layer 2 MAC addresses to make the devices on a LAN believe that the Layer 2 address of the attacker is the Layer 2 address of its default gateway. This is called *ARP poisoning*. Frames that are supposed to go to the default gateway are forwarded by the switch to the Layer 2 address of the attacker on the same network. As a courtesy, the attacker can forward the frames to the correct destination so that the client will have the connectivity needed, and the attacker now sees all the data between the two devices. To mitigate this risk, you could use techniques such as dynamic Address Resolution Protocol (ARP) inspection (DAI) on switches to prevent spoofing of the Layer 2 addresses.

The attacker could also implement the attack by placing a switch into the network and manipulating the Spanning Tree Protocol (STP) to become the root switch (and thus gain the ability to see any traffic that needs to be sent through the root switch).

A man-in-the-middle attack can occur at Layer 3 by placing a rogue router on the network and then tricking the other routers into believing that this new router has a better path. This could cause network traffic to flow through the rogue router and again allow the attacker to steal network data. You can mitigate attacks such as these in various ways, including using routing authentication protocols and filtering information from being advertised or learned on specific interfaces.

A man-in-the-middle attack can occur by compromising the victim's machine and installing malware that can intercept the packets sent by the victim and sending them to the attacker. This type of malware can capture packets before they are encrypted if the victim is using SSL/TLS/HTTPS or any other mechanism.

To safeguard data in motion, one of the best things you can do is to use encryption for the confidentiality of the data in transit. If you use plaintext protocols for management, such as Telnet or HTTP, an attacker who has implemented a man-in-the-middle attack can see the contents of your cleartext data packets and, as a result, will see everything that goes across his or her device, including usernames and passwords that are used. Using management protocols that have encryption built in, such as Secure Shell (SSH) and Hypertext Transfer Protocol Secure (HTTPS), is considered a best practice, and using VPN protection for cleartext sensitive data is also considered a best practice.

## Denial-of-Service Attacks

Denial-of-service (DoS) and distributed DoS (DDoS) attacks have been around for quite some time now, but there has been heightened awareness of them over the past few years.

DDoS attacks can generally be divided into the following three categories:


- Direct DDoS attacks
- Reflected
- Amplification DDoS attacks

## Direct DDoS

Direct DDoS attacks occur when the source of the attack generates the packets, regardless of protocol, application, and so on, that are sent directly to the victim of the attack.

Figure 4-4 illustrates a direct DDoS attack.

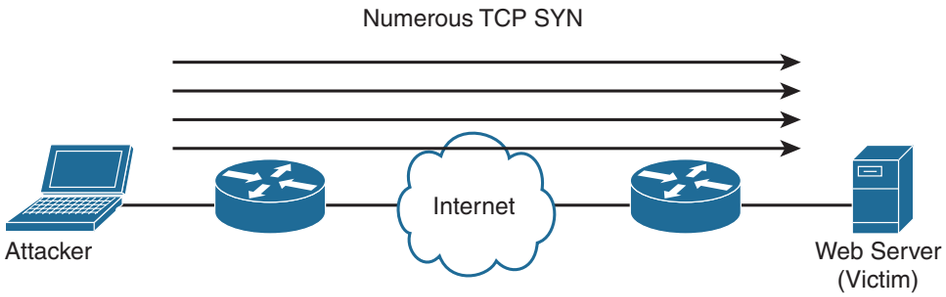



**Figure 4-4** *Direct DDoS Attack*

In Figure 4-4, the attacker launches a direct DoS to a web server (the victim) by sending numerous TCP SYN packets. This type of attack is aimed at flooding the victim with an overwhelming number of packets, oversaturating its connection bandwidth or depleting the target's system resources. This type of attack is also known as a SYN flood attack.

Cyber criminals also can use DDoS attacks to produce added costs to the victim when the victim is using cloud services. In most cases, when you use a cloud service such as Amazon Web Services (AWS), you pay per usage. Attackers can launch DDoS to cause you to pay more for usage and resources.

Another type of DoS is caused by exploiting vulnerabilities such as buffer overflows to cause a server or even network infrastructure device to crash, subsequently causing a denial-of-service condition.

## Botnets Participating in DDoS Attacks

Key Topic

Many attackers use botnets to launch DDoS attacks. A *botnet* is a collection of compromised machines that the attacker can manipulate from a command and control (C2 or CnC) system to participate in a DDoS, send spam emails, and perform other illicit activities. Figure 4-5 shows how an attacker uses a botnet to launch a DDoS attack.

In Figure 4-5, the attacker sends instructions to the command and control server; subsequently, the command and control server sends instructions to the bots within the botnet to launch the DDoS attack against the victim.

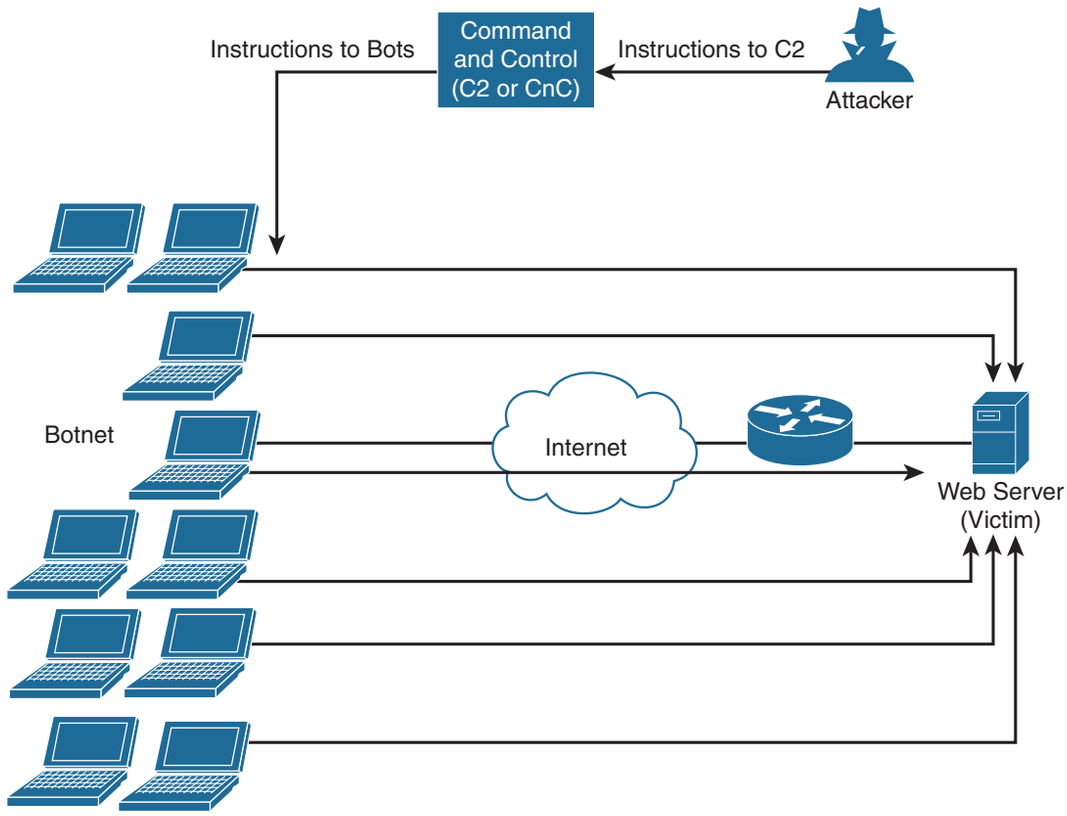



**Figure 4-5** *Botnets and a DDoS Attack*

## Reflected DDoS Attacks

Key Topic

Figure 4-6 illustrates what a reflected DDoS attack is. Reflected DDoS attacks occur when the sources of the attack are sent spoofed packets that appear to be from the victim, and then the "sources" of the attack become unwitting participants in the DDoS attacks by sending the response traffic back to the intended victim (in this example the "source" is the laptop illustrated in Figure 4-6). UDP is often used as the transport mechanism because it is more easily spoofed due to the lack of a three-way handshake. For example, if the attacker (A) decides to attack a victim (V), the attacker will send packets (for example, Network Time Protocol [NTP] requests) to a source (S) that thinks these packets are legitimate. The source then responds to the NTP requests by sending the responses to the victim, who was never expecting these NTP packets from the source (see Figure 4-6).

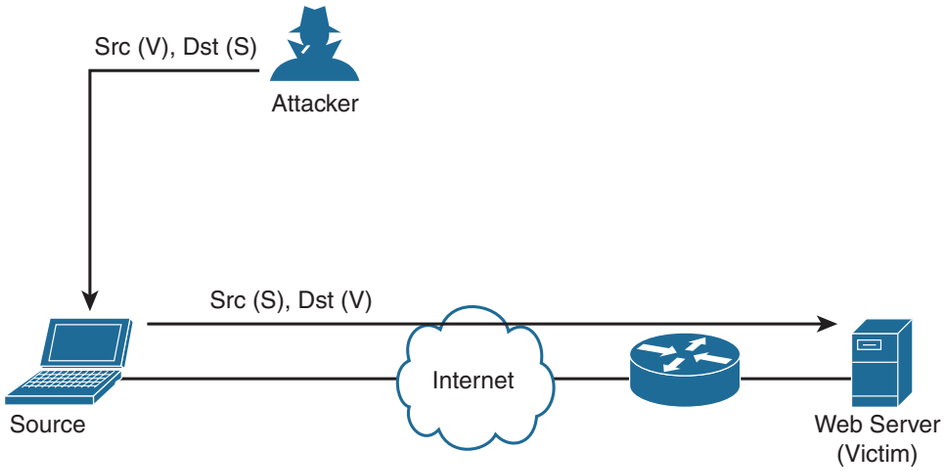



**Figure 4-6** *Reflected DDoS Attacks*

Key Topic

An amplification attack is a form of reflected attack in which the response traffic (sent by the unwitting participant) is made up of packets that are much larger than those that were initially sent by the attacker (spoofing the victim). An example occurs when DNS queries are sent and the DNS responses are much larger in packet size than the initial query packets. The end result is that the victim's machine gets flooded by large packets for which it never actually issued queries.

Key Topic

## Attack Methods for Data Exfiltration

There are many different attack methods for data exfiltration. One of the most popular is to use DNS tunneling. Cisco is seeing this method used more and more for malware-based data exfiltration out of enterprise networks.

Attackers can encapsulate chunks of data into DNS packets to steal sensitive information such as personal identifiable information (PII), credit card numbers, and much more. The following are examples of DNS tunneling tools used by attackers to exfiltrate data:

- **DNS2TCP:** Uses the KEY, TXT DNS record types. More information can be found at www.aldeid.com/wiki/Dns2tcp.
- **DNScat-P:** Uses the A and CNAME DNS record types. More information can be found at http://tadek.pietraszek.org/projects/DNScat/.
- **Iodine Protocol v5.00:** Uses the NULL DNS record type. More information can be found at http://code.kryo.se/iodine/.
- **Iodine Protocol v5.02:** Uses the A, CNAME, MX, NULL, SRV, and TXT DNS record types. More information can be found at http://code.kryo.se/iodine/.
- **OzymanDNS:** Uses the A and TXT DNS record types. More information can be found at http://dankaminsky.com/2004/07/29/51/.
- **SplitBrain:** Uses the A and TXT DNS record types. More information can be found at www.splitbrain.org/blog/2008-11/02-dns_tunneling_made_simple.

- **TCP-Over-DNS:** Uses the CNAME and TXT DNS record types. More information can be found at www.sans.org/reading-room/whitepapers/dns/detecting-dns-tunneling-34152.
- **YourFreedom:** Uses the NULL DNS record type. More information can be found at http://your-freedom.net/.

There are many other tools and DNS tunneling techniques. The following reference includes many additional types of tools and DNS exfiltration attacks: www.sans.org/reading-room/whitepapers/dns/detecting-dns-tunneling-34152.

DNS tunneling may be detected by analyzing the DNS packet payload or by using traffic analysis such as byte count and frequency of the DNS requests.

## ARP Cache Poisoning

Threat actors can attack hosts, switches, and routers connected to your Layer 2 network by poisoning the ARP caches of systems connected to the subnet and by intercepting traffic intended for other hosts on the subnet. Cisco switches support a feature called *dynamic ARP inspection* that validates ARP packets and intercepts, logs, and discards ARP packets with invalid IP-to-MAC address bindings. This feature also protects the network from certain man-in-the-middle attacks. The dynamic ARP inspection feature ensures that only valid ARP requests and responses are relayed by performing the following:

- Intercepting all ARP requests and responses on untrusted ports.
- Verifying that each of the intercepted packets has a valid IP-to-MAC address binding before updating the local ARP cache or before forwarding the packet to the respective destination host.
- Dropping invalid ARP packets.
- Determining if an ARP packet is valid based on IP-to-MAC address bindings stored in a trusted database. This database is called the *DHCP snooping binding database*.

On Cisco switches, you can enable dynamic ARP inspection on a per-VLAN basis with the **ip arp inspection vlan** *vlan-range* global configuration command. In environments without DHCP configured, dynamic ARP inspection can validate ARP packets against user-configured ARP access control lists for hosts with statically configured IP addresses. You can use the **arp access-list acl-name global configuration** command to define the ACL.

The following are some additional Layer 2 security best practices for securing your infrastructure:

- Select an unused VLAN (other than VLAN 1) and use that for the native VLAN for all your trunks. Do not use this native VLAN for any of your enabled access ports.
- Avoid using VLAN 1 anywhere because it is the default.
- Administratively configure switch ports as access ports so that users cannot negotiate a trunk and disable the negotiation of trunking (no Dynamic Trunking Protocol [DTP]).
- Limit the number of MAC addresses learned on a given port with the port security feature.

- Control spanning tree to stop users or unknown devices from manipulating it. You can do so by using the BPDU Guard and Root Guard features.
- Turn off Cisco Discovery Protocol (CDP) on ports facing untrusted or unknown networks that do not require CDP for anything positive. (CDP operates at Layer 2 and may provide attackers information you would rather not disclose.)
- On a new switch, shut down all ports and assign them to a VLAN that is not used for anything other than a parking lot. Then bring up the ports and assign correct VLANs as the ports are allocated and needed.

Several other Layer 2 security features can be used to protect your infrastructure:

- **Port Security:** This feature limits the number of MAC addresses to be learned on access switch posts.
- **BPDU Guard:** If BPDUs show up where they should not, the switch will protect itself.
- **Root Guard:** This feature controls which ports are not allowed to become root ports to remote switches.
- **Dynamic ARP inspection:** This feature was covered earlier in this section.
- **IP Source Guard:** This feature prevents spoofing of Layer 3 information by hosts.
- **802.1X:** This feature authenticates and authorizes users before allowing them to communicate to the rest of the network.
- **DHCP snooping:** This feature prevents rogue DHCP servers from impacting the network.
- **Storm control:** This feature limits the amount of broadcast or multicast traffic flowing through the switch.
- **Access control lists:** This feature provides Layer 3 and Layer 2 ACLs for traffic control and policy enforcement.

## Spoofing Attacks

In a spoofing attack an attacker impersonates another device to execute an attack. The following are a few examples of spoofing attacks:

- **IP address spoofing attack:** The attacker sends IP packets from a fake (or spoofed) source address to disguise itself. DDoS attacks typically use IP spoofing to make the packets appear to be from legitimate source IP addresses.
- **ARP spoofing attack:** The attacker sends spoofed ARP packets across the Layer 2 network to link the attacker's MAC address with the IP address of a legitimate host. The best practices covered in the previous section help mitigate ARP spoofing attacks.
- **DNS server spoofing attack:** The attacker modifies the DNS server to reroute a specific domain name to a different IP address. DNS server spoofing attacks are typically used to spread malware.

## Route Manipulation Attacks

Key Topic

There are different route manipulation attacks, but one of the most common is the BGP hijacking attack. Border Gateway Protocol (BGP) is a dynamic routing protocol used to route Internet traffic. The BGP hijacking attack can be launched by an attacker by configuring or compromising an edge router to announce prefixes that have not been assigned to his or her organization. If the malicious announcement contains a route that is more specific than the legitimate advertisement or presents a shorter path, the victim's traffic may be redirected to the attacker. In the past, threat actors have leveraged unused prefixes for BGP hijacking to avoid attention from the legitimate user or organization.

## Password Attacks

The following are a few examples of the most common password attacks:

Key Topic

- **Password-guessing attack:** This is the most common type of password attack, but some of these techniques may be very inefficient. Threat actors can guess passwords locally or remotely using either a manual or automated approach. Tools like Hydra (www.thc.org) can automate the process of password guessing. Automated password attack tools and crackers leverage different techniques. Some use a method called a *brute-force attack*, where the attacker tries every possible combination of characters for a password. Another technique is a password-guessing attack called a *dictionary attack*. Because most passwords consist of whole words, dates, and numbers, these tools use a dictionary of words, phrases, and even the most commonly used passwords (such as qwerty, password1, and so on). Other tools such as John the Ripper (www.openwall.com/john) and Cain & Abel (www.oxid.it) can take a hybrid approach from brute-force and dictionary attacks.

- **Password-resetting attack:** In many cases, it is easier to reset passwords than to use tools to guess them. Several cracking tools just attempt to reset passwords. In most cases, the attacker boots from a USB device to get around the typical Windows protections. Most password resetters contain a bootable version of Linux that can mount NTFS volumes and help the attacker locate and reset the administrator's password.

- **Password cracking:** These attacks work by taking a password hash and converting it to its plaintext original. In this case, the attacker needs tools such as extractors for hash guessing, rainbow tables for looking up plaintext passwords, and password sniffers to extract authentication information. The concept of rainbow tables is that the attacker computes possible passwords and their hashes in a given system and puts the results into a lookup table called a *rainbow table*. This allows an attacker to get a hash from the victim system and then just search for that hash in the rainbow table to get the plaintext password. To mitigate rainbow table attacks, you can disable LM hashes and use long and complex passwords.

- **Password sniffing:** The threat actor just sniffs authentication packets between a client and server and extracts password hashes or enough authentication information to begin the cracking process.

- **Password capturing:** This is typically done by using key loggers or Trojan horses.

### Wireless Attacks

The following are a few examples of wireless-specific attacks:

Key Topic

- **Installing a rogue access point:** The attacker basically installs an access point and can create a backdoor and obtain access to the network and its systems.
- **Jamming wireless signals and causing interference:** The purpose of this attack is to cause a full or partial denial-of-service condition in the wireless network.
- **War driving:** Attackers use this methodology to find wireless access points wherever they may be. The term *war driving* is used because the attacker can just drive around and get a huge amount of information over a very short period of time.
- **Bluejacking:** In this type of attack, the attacker sends unsolicited messages to another device via Bluetooth.
- **Evil twin attack:** This is done when the attacker is trying to create rogue access points to gain access to the network or steal information. Basically, the attacker purchases a wireless access point, plugs it into the network, and configures it exactly the same as the existing network.
- **IV attack:** The attacker can cause some modification on the initialization vector (IV) of a wireless packet that is encrypted during transmission. The goal of the attacker is to obtain a lot of information about the plaintext of a single packet and generate another encryption key that then can be used to decrypt other packets using the same IV.
- **WEP/ attack:** WEP and several versions of WPA are susceptible to different vulnerabilities and are considered weak. WEP should never be used. At the time of writing, WPA Version 3 is the latest version of WPA offering several fixes to known vulnerabilities in WPA Version 1 and Version 2 (such as the KRACK attacks [krackattacks.com]).
- **WPS attack:** This attack is carried out with WPS password-guessing tools to obtain the WPS passwords and use them to gain access to the network and its data.

## Types of Vulnerabilities

Understanding the weaknesses and vulnerabilities in a system or network is a huge step toward correcting these vulnerabilities or putting in appropriate countermeasures to mitigate threats against them. Potential network vulnerabilities abound, with many resulting from one or more of the following:

- Policy flaws
- Design errors
- Protocol weaknesses
- Misconfiguration
- Software vulnerabilities
- Human factors

- Malicious software
- Hardware vulnerabilities
- Physical access to network resources

Cisco and others have created databases that categorize threats in the public domain. The Common Vulnerabilities and Exposures (CVE) is a dictionary of publicly known security vulnerabilities and exposures. A quick search using your favorite search engine will lead you to the website. Also, the National Vulnerability Database (NVD) is a repository of standards-based vulnerability information; you can do a quick search for it too. (URLs change over time, so it is better to advise you to just do a quick search and click any links that interest you.)

The following are examples of the most common types of vulnerabilities:



Key Topic

- **API-based vulnerabilities:** These vulnerabilities are aimed to attack flaws in application programming interfaces (APIs).
- **Authentication and authorization bypass vulnerabilities:** These vulnerabilities are used to bypass authentication and authorization mechanisms of systems within a network.
- **Buffer overflow:** Earlier in this chapter you learned that a buffer overflow occurs when a program or software puts more data in a buffer than it can hold or when a program tries to put data in a memory location past a buffer. This is done so data outside the bounds of a block of allocated memory can corrupt other data or crash the program or operating system. In a worst-case scenario, this could lead to the execution of malicious code. Buffer overflows can occur in a variety of ways and, unfortunately, many error-prone techniques often are used to prevent them.
- **Cross-site scripting (XSS) vulnerability:** In this type of web application vulnerability, malicious scripts are injected into legitimate and trusted websites. An attacker can launch an attack against an XSS vulnerability using a web application to send malicious code (typically in the form of a browser-side script) to a different end user. XSS vulnerabilities are quite widespread and occur anywhere a web application uses input from a user within the output it generates without validating or encoding it. There are several types of XSS vulnerabilities (reflected, stored, and DOM-based XSS). Successful exploitation could result in installation or execution of malicious code, account compromise, session cookie hijacking, revelation or modification of local files, or site redirection. You typically find XSS vulnerabilities in search fields that echo a search string back to the user, HTTP headers, input fields that echo user data, hidden form fields, and error messages that return user-supplied text.
- **Cross-site request forgery (CSRF) vulnerability:** This type of vulnerability forces an end user to execute malicious steps on a web application. This is typically done after the user is authenticated to such an application. CSRF attacks generally target state-changing requests, and attackers cannot steal data because they have no way to see the response to the forged request. CSRF attacks are carried out by being combined with social engineering.

- **Cryptographic vulnerability:** This is a vulnerability or flaw in a cryptographic protocol or its implementation.
- **Deserialization of untrusted data vulnerability:** This type of vulnerability uses or causes malformed data or unexpected data to abuse application logic, cause a DoS attack, or execute arbitrary code.
- **Double free:** This vulnerability typically in C, C++, and similar languages occurs when **free()** is called more than once with the same memory address as an argument.
- **Insufficient entropy:** In this vulnerability a cryptographic application does not have proper entropy. For example, pseudo-random number generators (PRNGs) can be susceptible to insufficient entropy vulnerabilities and attacks when they are initialized.
- **SQL injection vulnerability:** In this type of vulnerability, attackers can insert or inject a SQL query via the input data from the client to the application or database. Attackers can exploit SQL injector vulnerabilities to read sensitive data from the database, modify or delete database data, execute administration operations on the database, and even issue commands to the operating system.

Key Topic

There are many more types of vulnerabilities. The Open Web Application Security Project (OWASP) provides good references to different types of vulnerabilities and how to mitigate them. OWASP is an international organization dedicated to educating industry professionals, creating tools, and evangelizing best practices for securing web applications and underlying systems. There are dozens of OWASP chapters around the world. It is recommended that you become familiar with OWASP's website (www.owasp.org) and guidance.

Key Topic

**Tip** The GitHub repository at my website (see https://h4cker.org/github) includes numerous other resources and links to other tools and intentionally vulnerable systems that you can deploy in your lab. I also created a learning environment called WebSploit for different cybersecurity and ethical hacking (penetration testing) training sessions, books, and video courses. WebSploit includes several vulnerable applications running in Docker containers and the tools that come in Kali Linux (as well as a few additional tools). Penetration testing skills are not required for the Cyber Ops Associates certification. However, practicing some of the attacks covered in this chapter may allow you to gain additional knowledge about the underlying vulnerabilities and methodologies to exploit such vulnerabilities. You can get more information and download WebSploit from https://websploit.org.

# Exam Preparation Tasks

## Review All Key Topics

Review the most important topics in the chapter, noted with the Key Topic icon in the outer margin of the page. Table 4-2 lists these key topics and the page numbers on which each is found.

Key Topic

**Table 4-2** Key Topics for Chapter 4

| Key Topic Element | Description | Page |
|---|---|---|
| Paragraph | Understanding passive vs. active reconnaissance | |
| Tip | Understanding Open-Source Intelligence (OSINT) | |
| List | Different types of port- and network-scanning techniques | |
| List | What are phishing, pharming, and malvertising? | |
| Section | Privilege Escalation Attacks | |
| Section | Backdoors | |
| Section | Buffer Overflows and Code Execution | |
| Section | Man-in-the-Middle Attacks | |
| List | Identifying the different types of DDoS attacks | |
| Paragraph | What are botnets? | |
| Paragraph | Reflected DDoS attacks | |
| Paragraph | What are amplification attacks? | |
| Section | Attack Methods for Data Exfiltration | |
| Paragraph | ARP cache poisoning | |
| Paragraph | Route manipulation attacks | |
| List | Different types of password attacks | |
| List | The most common attacks against wireless networks | |
| List | Defining and understanding different types of security vulnerabilities | |
| Paragraph | The Open Web Application Security Project (OWASP) | |
| Tip | Accessing Omar's GitHub repository and WebSploit labs | |



## Define Key Terms

Define the following key terms from this chapter, and check your answers in the glossary:

SQL injection, CSRF, XSS, buffer overflow, war driving, rainbow tables, DNS tunneling, botnet, backdoors

## Review Questions

The answers to these questions appear in Appendix A, "Answers to the 'Do I Know This Already?' Quizzes and Review Questions." For more practice with exam format questions, use the exam engine on the website.

1. What describes the use of a rainbow table?
2. What is the name given to a methodology used by attackers to find wireless access points wherever they may be?
3. What is a common web application vulnerability where malicious scripts are injected into legitimate and trusted websites?
4. What is a type of vulnerability that attackers can exploit to read sensitive data from the database, modify or delete database data, execute administration operations on the database, and even issue commands to the operating system?

5. What attack results when attackers place themselves in line between two devices that are communicating, with the intent of performing reconnaissance or manipulating the data as it moves between the devices?
6. What is a type of vulnerability where an attacker can use or cause malformed data or unexpected data to abuse an application's logic, cause a DoS attack, or execute arbitrary code?
7. What is a type of vulnerability that describes when a program or software puts more data in a buffer than it can hold or when a program tries to put data in a memory location past a buffer?
8. What type of attack is done when the attacker tries to create rogue access points to gain access to the network or steal information?
9. What is an attack where threat actors can attack hosts, switches, and routers connected to your Layer 2 network by poisoning the ARP caches of systems connected to the subnet and by intercepting traffic intended for other hosts on the subnet?
10. Cisco switches support a feature that validates ARP packets and intercepts, logs, and discards ARP packets with invalid IP-to-MAC address bindings. What is this feature called?