CCNA Cyber Ops (Version 1.1) – Chapter 7: Network Attacks: A Deeper Look

itexamanswers.net/ccna-cyber-ops-version-1-1-chapter-7-network-attacks-a-deeper-look.html

June 13, 2019

Objectives

Upon completion of this chapter, you will be able to answer the following questions:

- What is the importance of network monitoring?
- How is network monitoring conducted?
- How do IP vulnerabilities enable network attacks?
- How do TCP and UDP vulnerabilities enable network attacks?
- What are the IP vulnerabilities?
- How do network application vulnerabilities enable network attacks?

Key Terms

This chapter uses the following key terms. You can find the definitions in the Glossary.

network TAP
port mirroring
Switch Port Analyzer (SPAN)
session hijacking
OS fingerprinting
amplification and reflection techniques
non-blind spoofing
blind spoofing
ARP cache poisoning
fast flux
double IP flux
domain generation algorithms
DNS tunneling
iFrame
HTTP 302 cushioning
domain shadowing
homoglyphs
SQL injection
cross-site scripting (XSS)

Introduction (7.0)

Cybersecurity analysts use a variety of tools to identify attacks. A solid understanding of protocol vulnerabilities is essential to using these tools.

This chapter first covers the importance of traffic monitoring and how it is conducted. This is followed by an in-depth discussion of the vulnerabilities to network protocols and services including IP, TCP, UDP, ARP, DNS, DHCP, HTTP, and email.

Class Activity 7.0.1.2: What's Going On?

In this activity, you will identify the processes running on a computer, theprotocol they are using, and their local and remote port addresses.

Network Monitoring and Tools (7.1)

In this section, you will learn about network traffic monitoring.

Introduction to Network Monitoring (7.1.1)

In this topic, you will learn the importance of network monitoring.

Network Security Topology (7.1.1.1)

"All networks are targets" is a common adage used to describe the current landscape of network security. Therefore, to mitigate threats, all networks must be secured and protected as best as possible.

This requires a defense-in-depth approach. It requires using proven methods and secure infrastructure consisting of firewalls, intrusion detection systems (IDSs)/intrusion prevention systems (IPSs), and endpoint security software. These methods and technologies are used to introduce automated monitoring to the network, creating alerts or even automatically blocking offensive devices when something goes wrong.

However, for large networks, an extra layer of protection must be added. Devices such as firewalls and IPSs operate based on preconfigured rules. They monitor traffic and compare it against the configured rules. If there is a match, the traffic is handled according to the rule. This works relatively seamlessly, but sometimes, legitimate traffic is mistaken for unauthorized traffic. Called false positives, these situations require human eyes to see and evaluate them before they can be validated. An important part of the job of the security analyst is to review all alerts generated by network devices and validate their nature. Was that file downloaded by user X really malware? Is that website visited by user Y really malicious? Is the printer on the third floor really compromised because it is trying to connect to a server that is out on the Internet? All these questions are commonly asked by security analysts daily. It is their job to determine the correct answers.

Monitoring the Network (7.1.1.2)

The day-to-day operation of a network consists of common patterns of traffic flow, bandwidth usage, and resource access. Together, these patterns identify the normal network behavior. Security analysts must be intimately familiar with the normal network behavior because abnormal network behavior typically indicates a problem.

To discover the normal network behavior, network monitoring must be implemented. Various tools are used to help discover normal network behavior including IDS, packet analyzers, SNMP, NetFlow, and others.

Some of these tools require captured network data. There are two common methods used to capture traffic and send it to network monitoring devices:

- Network Terminal Access Points (TAPs)
- Traffic mirroring using Switched Port Analyzer (SPAN)

Both of these methods are discussed in this chapter.

Network TAPs (7.1.1.3)

A network TAP is typically a passive splitting device implemented inline between a device of interest and the network. A TAP forwards all traffic including physical layer errors to an analysis device.

Figure 7-1 displays a sample topology displaying a TAP installed between a network firewall and the internal router.

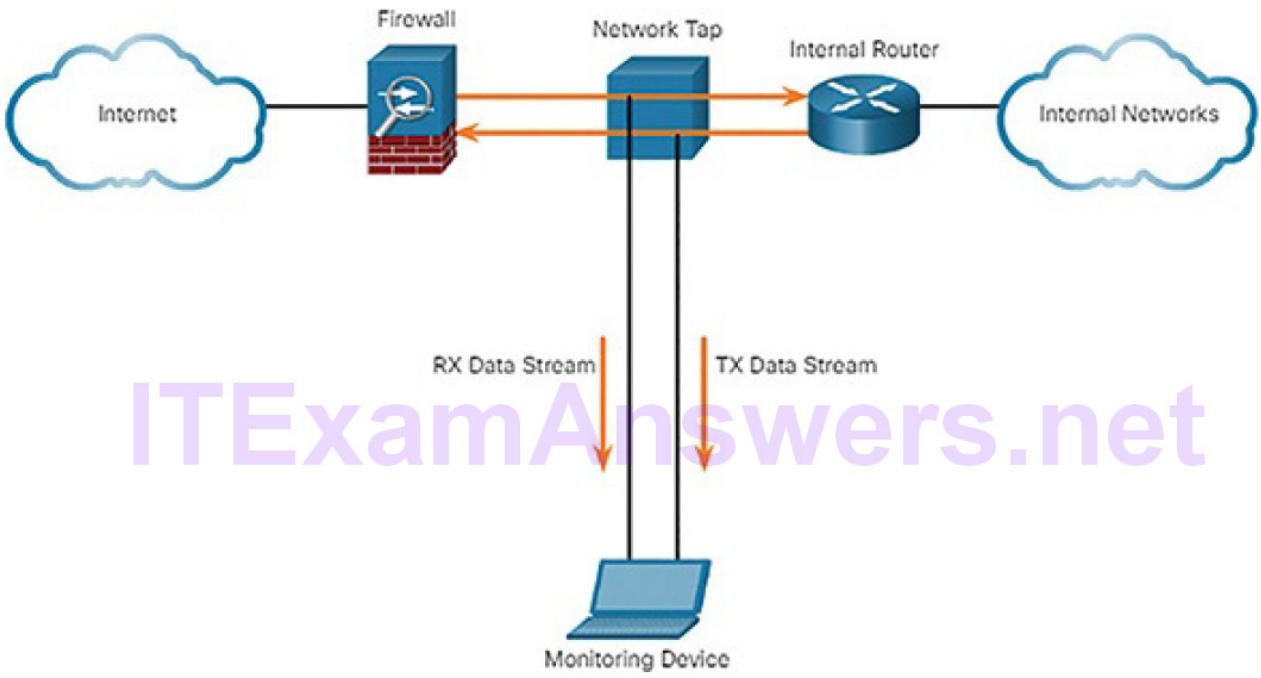

Figure 7-1 Implementing a TAP in a Sample Network

Notice how the TAP simultaneously sends both the transmit (TX) data stream from the internal router and the receive (RX) data stream to the internal router on separate, dedicated channels. This ensures that all data arrives at the monitoring device in real time. Therefore, network performance is not affected or degraded by monitoring the connection.

TAPs are also typically fail-safe, which means if it fails or loses power, traffic between the firewall and internal router is not affected.

Traffic Mirroring and SPAN (7.1.1.4)

Network switches segment the network by design, limiting the amount of traffic visible by the network monitoring device. Because data capturing for network monitoring requires all traffic to be captured, special techniques must be employed to bypass the network segmentation imposed by network switches.
Port mirroring is one of these techniques. Supported by many enterprise switches, port mirroring enables the switch to copy frames of one or more ports to a Switch Port Analyzer (SPAN) port connected to an analysis device.

SPAN terminology includes

Ingress traffic: Traffic that enters the switch.

Egress traffic: Traffic that leaves the switch.

Source (SPAN) port: A port that is monitored as traffic enters it before being replicated (mirrored) to the destination ports.

Destination (SPAN) port: A port that mirrors source ports. Destination SPAN ports often connect to analysis devices such as a packet analyzer or an IDS.

Figure 7-2 displays a sample topology that shows a switch interconnecting two hosts. The switch will forward ingress traffic on F0/1 and egress traffic on F0/2 to the destination SPAN port G0/1 connecting to an IDS.

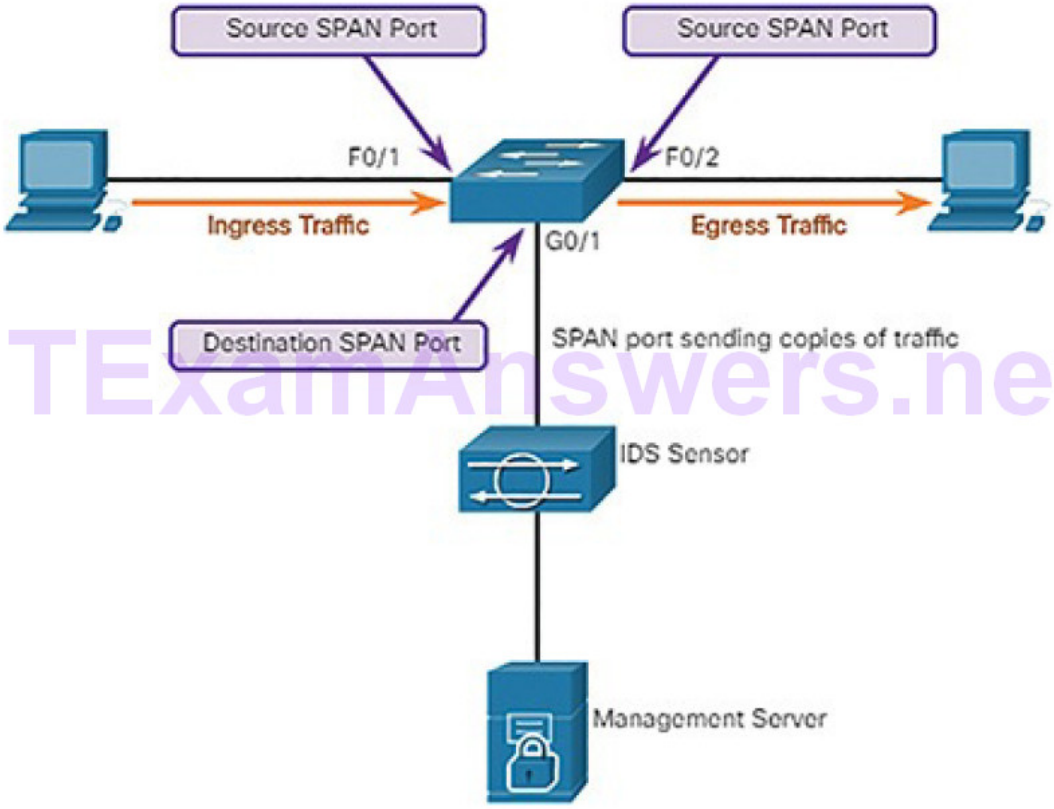

Figure 7-2 SPAN Operation

The association between source ports and a destination port is called a SPAN session. In a single session, one or multiple ports can be monitored. On some Cisco switches, session traffic can be copied to more than one destination port. Alternatively, a source VLAN can be specified in which all ports in the source VLAN become sources of SPAN traffic. Each SPAN session can have ports or VLANs as sources, but not both.

Note
A variation of SPAN called Remote SPAN (RSPAN) enables a network administrator to use the flexibility of VLANs to monitor traffic on remote switches.

Introduction to Network Monitoring Tools (7.1.2)

In this topic, you will learn how network monitoring is conducted.

Network Security Monitoring Tools (7.1.2.1)

Common tools used for network security monitoring include:

- Network protocol analyzers (Wireshark and tcpdump)
- NetFlow
- Security information and event management (SIEM) systems

It is also common for security analysts to rely on log files and Simple Network Management Protocol (SNMP) to discover normal network behavior.

Practically all systems generate log files to communicate and record their operations. By closely monitoring log files, a security analyst can gather extremely valuable information.

SNMP allows analysts to ask for and receive information about the operation of network devices, and is another good tool for monitoring the behavior of a network.

Security analysts must be familiar with all of these tools.

Network Protocol Analyzers (7.1.2.2)

Network protocol analyzers (or “packet sniffer” applications) are programs used to capture traffic. Often including a graphical interface, protocol analyzers show what is happening on the network. Analysts can use these applications to see network exchanges down to the packet level. If a computer has been infected with malware and is currently attacking other computers in the network, the analyst can see that clearly by capturing real-time network traffic and analyzing the packets.

Not only used for security analysis, network protocol analyzers are also veryuseful for network troubleshooting, software and protocol development, and education. For instance, in security forensics, a security analyst may attempt to reconstruct an incident from relevant packet captures.

Wireshark, shown in Figure 7-3, has become a very popular network protocol analyzer tool that is used in Windows, Linux, and Mac OS environments. Captured frames are saved in a PCAP file. PCAP files contain the frame information, interface information, packet length, and timestamps.

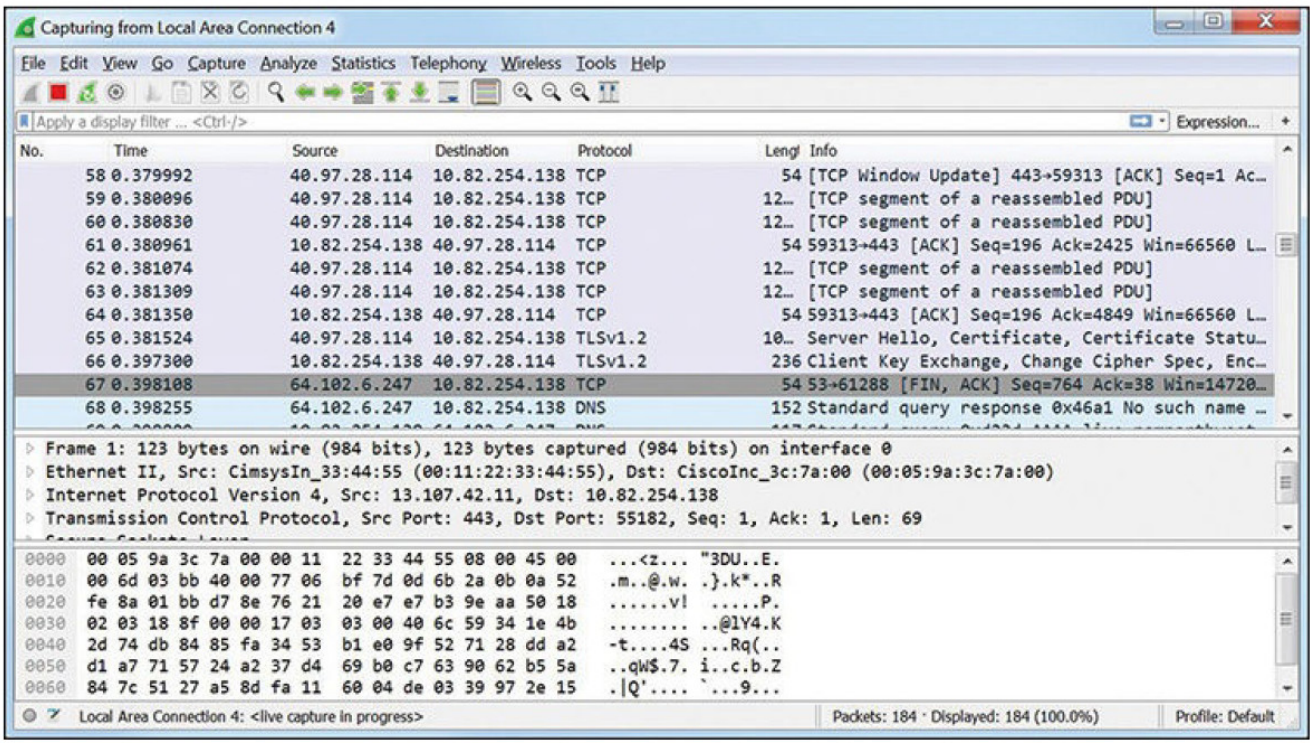

Figure 7-3 Sample Wireshark Output

Performing a long-term packet capture produces large PCAP files.

Wireshark can also open files that contain captured traffic from other software such as the **tcpdump** utility. Popular among UNIX-like systems such as Linux, **tcpdump** is a powerful utility with numerous command-line options. Example 7-1 displays a sample **tcpdump** capture of ping packets.

Example 7-1 Sample **tcpdump** Output

```
[root@secOps analyst]# tcpdump -i h1-eth0 -n

tcpdump: verbose output suppressed, use -v or -vv for full protocol
decode
listening on h1-eth0, link-type EN10MB (Ethernet), capture size
262144 bytes
10:42:19.841549 IP 10.0.0.12 > 10.0.0.11: ICMP echo request, id 2279,
seq 5,
length 64
10:42:19.841570 IP 10.0.0.11 > 10.0.0.12: ICMP echo reply, id 2279,
seq 5,
length 64
10:42:19.854287 IP 10.0.0.12 > 10.0.0.11: ICMP echo request, id 2279,
seq 6,
length 64
10:42:19.854304 IP 10.0.0.11 > 10.0.0.12: ICMP echo reply, id 2279,
seq 6,
length 64
10:42:19.867446 IP 10.0.0.12 > 10.0.0.11: ICMP echo request, id 2279,
seq 7,
length 64
10:42:19.867468 IP 10.0.0.11 > 10.0.0.12: ICMP echo reply, id 2279,
seq 7,
length 64
^C
6 packets captured
6 packets received by filter
0 packets dropped by kernel

[root@secOps analyst]#
```

Note
windump is a Microsoft Windows variant of **tcpdump,** and **tshark** is Wireshark command-line tool similar to **tcpdump.**

NetFlow (7.1.2.3)

NetFlow is a Cisco IOS technology that provides 24×7 statistics on packetsflowing through a Cisco router or multilayer switch. NetFlow is the standard for collecting IP operational data in IP networks. NetFlow is now supported on non-Cisco platforms.

NetFlow can be used for network and security monitoring, network planning, and traffic analysis. It provides a complete audit trail of basic information about every IP flow forwarded on a device. This information includes the source and destination device IP information, the time of the communication, and the amount of data transferred. NetFlow does not capture the actual content on the flow. NetFlow functionality is often compared to a telephone bill. The bill identifies the destination number and the time and duration of the call. However, it does not display the content of the telephone conversation.

Although NetFlow stores flow information in a local cache on the device, it should always be configured to forward data to a NetFlow collector such as Cisco Stealthwatch.

For example, in Figure 7-4, PC1 connects to PC2 using an application such as HTTPS. NetFlow can monitor that application connection, tracking byte and packet counts for that individual application flow. It then pushes the statistics over to an external server called a NetFlow collector.

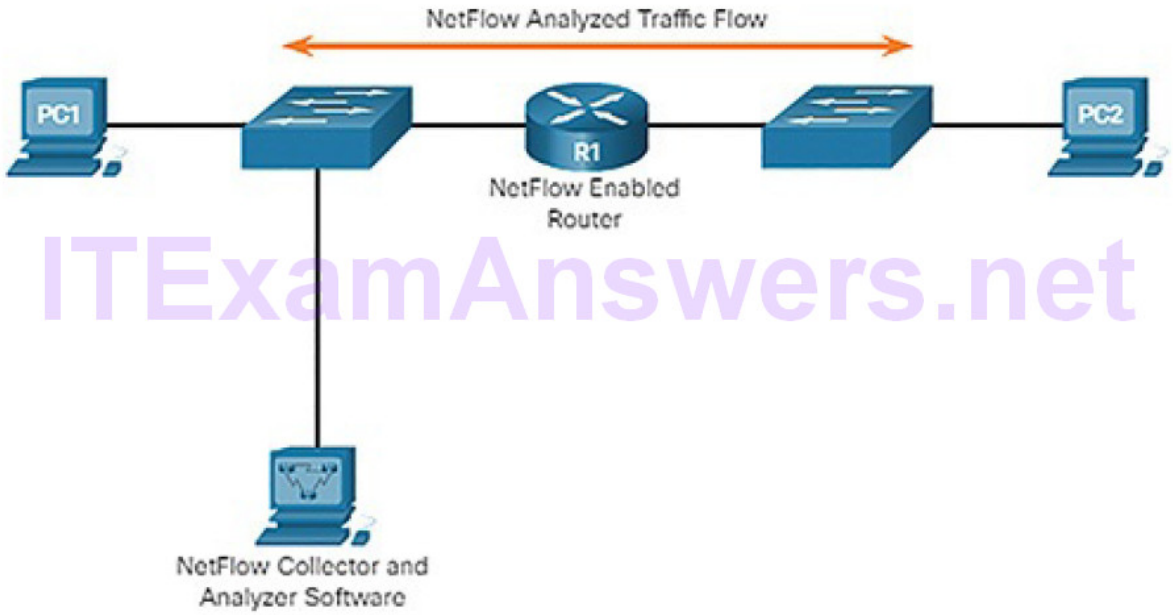

Figure 7-4 NetFlow in the Network

NetFlow collectors like Cisco Stealthwatch can also perform advanced functions including:

Flow stitching: It groups individual entries into flows.

Flow deduplication: It filters duplicate incoming entries from multiple NetFlow clients.

NAT stitching: It simplifies flows with NAT entries.

Cisco Stealthwatch has many more features than just NetFlow.

SIEM (7.1.2.4)

Security information and event management (SIEM) is a technology used in enterprise organizations to provide real-time reporting and long-term analysis of security events.

SIEM includes the following essential functions:

Forensic analysis: Provides the ability to search logs and event records from sources throughout the organization. It provides more complete information for forensic analysis.

Correlation: Examines logs and events from different systems or applications, speeding detection of and reaction to security threats.

Aggregation: Reduces the volume of event data by consolidating duplicate event records.

Reporting: Presents the correlated and aggregated event data in real-time monitoring and long-term summaries.

SIEM provides details on the source of suspicious activity:

- User information such as username, authentication status, location
- Device information such as manufacturer, model, OS version, MAC address, network connection method, and location
- Posture information such as whether the device is compliant with the security policy, has up-to-date antivirus files, and is updated with latest OS patches

Using this information, network security analysts can quickly and accurately assess the significance of any security event and answer the critical questions, such as:

- Who is associated with this event?
- Does the user have access to other sensitive resources?Does this event represent a potential compliance issue?
- Is it an important user with access to intellectual property or sensitive information?
- Is the user authorized to access that resource?
- What kind of device is being used?

SIEM Systems (7.1.2.5)

Several SIEM systems exist. Splunk is one of the more popular proprietary SIEM systems used by Security Operation Centers.

As an open source option, this course uses the ELK suite for SIEM functionality. ELK is an acronym for three open source products from Elastic:

Elasticsearch: Document-oriented, full-text search engine

Logstash: Pipeline processing system that connects “inputs” to “outputs” with optional “filters” in between

Kibana: Browser-based analytics and search dashboard for Elasticsearch

Activity 7.1.2.6: Identify the Network Monitoring Tool

Refer to the online course to complete this Activity.

Packet Tracer 7.1.2.7: Logging Network Activity

In this activity, you will intercept credentials using a sniffer device, while observing an FTP session. An exchange of syslog messages will also be intercepted by a sniffer device.

Attacking the Foundation (7.2)

In this section, you will learn how TCP/IP vulnerabilities enable network attacks.

IP Vulnerabilities and Threats (7.2.1)

In this topic, you will learn how IP vulnerabilities enable network attacks.

IPv4 and IPv6 (7.2.1.1)

IP was designed as a connectionless protocol. It provides the necessary functions to deliver a packet from a source host to a destination host over an interconnected system of networks. The protocol was not designed to track and manage the flow of packets. These functions, if required, are performed primarily by TCP at Layer 4.

IP makes no effort to validate whether the source IP address contained in a packet actually came from that source. For this reason, threat actors can send packets using a spoofed source IP address. In addition, threat actors can tamper with the other fields in the IP header to carry out their attacks. Therefore, it is important for security analysts to understand the different fields in both the IPv4 and IPv6 headers.

The IPv4 Packet Header (7.2.1.2)

There are ten fields in the IPv4 packet header, as shown in Figure 7-5:

Version: Contains a 4-bit binary value set to 0100 that identifies this as an IPv4 packet.

Internet Header Length: A 4-bit field containing the length of the IP header. The minimum length of an IP header is 20 bytes.

Differentiated Services or DiffServ (DS): Formerly called the Type of Service (ToS) field, the DS field is an 8-bit field used to determine the priority of each packet. The 6 most significant bits of the DiffServ field are the Differentiated Services Code Point (DSCP). The last 2 bits are the Explicit Congestion Notification (ECN) bits.

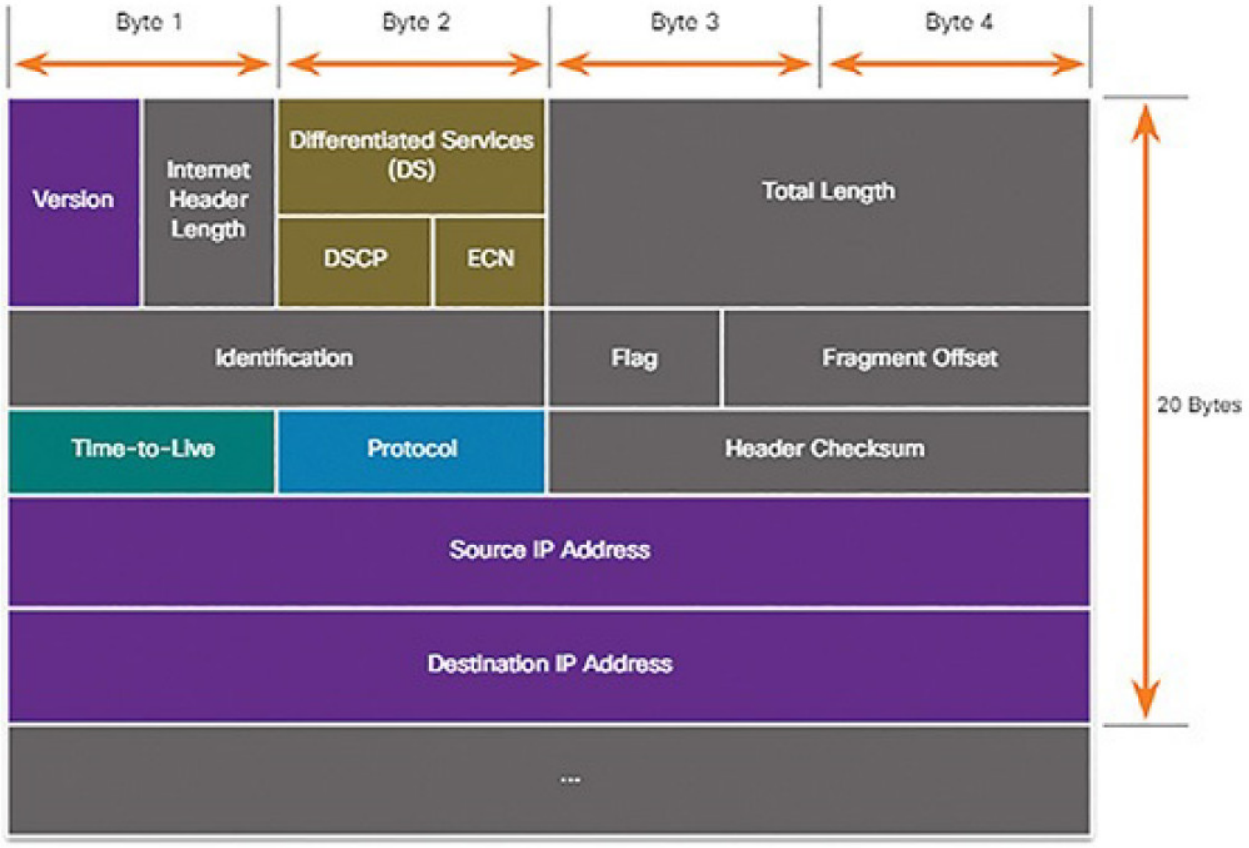

Figure 7-5 IPv4 Packet Header

Total Length: Specifies the length of the IP packet that includes the IP header and the user data. The Total Length field is 2 bytes, so the maximum size of an IP packet is 65,535 bytes.

Identification, Flag, and Fragment Offset: As an IP packet moves through the Internet, it might need to cross a route that cannot handle the size of the packet. The packet will be divided, or fragmented, into smaller packets and reassembled later. These fields are used to fragment and reassemble packets.

Time-to-Live (TTL): Contains an 8-bit binary value that is used to limit the lifetime of a packet. The packet sender sets the initial TTL value, and it is decreased by a value of 1 each time the packet is processed by a router. If the TTL field decrements to 0, the router discards the packet and sends an Internet Control Message Protocol (ICMP) Time Exceeded message to the source IP address.

Protocol: Used to identify the next-level protocol. This 8-bit binary value indicates the data payload type that the packet is carrying, which enables the network layer to pass the data to the appropriate upper-layer protocol. Common values include ICMP (1), TCP (6), and UDP (17).

Header Checksum: A value that is calculated based on the contents of the IP header. Used to determine if any errors have been introduced during transmission.

Source IPv4 Address: Contains a 32-bit binary value that represents the source IPv4 address of the packet. The source IPv4 address is always a unicast address.

Destination IPv4 Address: Contains a 32-bit binary value that represents the destination IPv4 address of the packet.

Options and Padding: This is a field that varies in length from 0 to a multiple of 32 bits. If the option values are not a multiple of 32 bits, 0s are added or padded to ensure that this field contains a multiple of 32 bits.

The IPv6 Packet Header (7.2.1.3)

There are eight fields in the IPv6 packet header, as shown Figure 7-6:

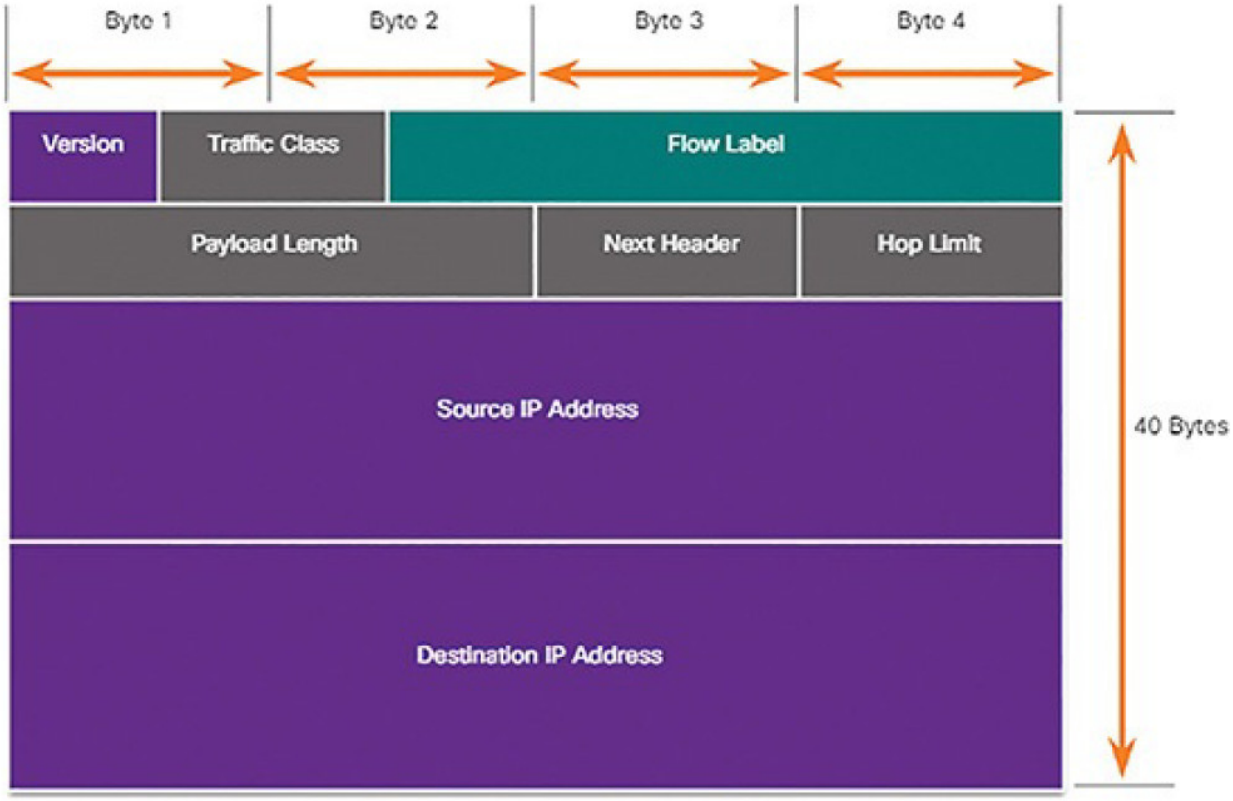

Figure 7-6 IPv6 Packet Header

Version: This field contains a 4-bit binary value set to 0110 that identifiesthis as an IPv6 packet.

Traffic Class: This 8-bit field is equivalent to the IPv4 Differentiated Services (DS) field.

Flow Label: This 20-bit field suggests that all packets with the same flow label receive the same type of handling by routers.

Payload Length: This 16-bit field indicates the length of the data portion or payload of the IPv6 packet.

Next Header: This 8-bit field is equivalent to the IPv4 Protocol field. It indicates the data payload type that the packet is carrying, enabling the network layer to pass the data to the appropriate upper-layer protocol.

Hop Limit: This 8-bit field replaces the IPv4 TTL field. This value is decremented by a value of 1 by each router that forwards the packet. When the counter reaches 0, the packet is discarded, and an ICMPv6 Time Exceeded message is forwarded to the sending host, indicating that the packet did not reach its destination because the hop limit was exceeded.

Source IPv6 Address: This 128-bit field identifies the IPv6 address of the sending host.

Destination IPv6 Address: This 128-bit field identifies the IPv6 address of the receiving host.

An IPv6 packet may also contain extension headers (EHs), which provide optional network layer information. Extension headers are optional and are placed between the IPv6 header and the payload. EHs are used for fragmentation, security, to support mobility, and more.

Unlike IPv4, routers do not fragment routed IPv6 packets.

IP Vulnerabilities (7.2.1.4)

There are different types of attacks targeting IP. These are some of the more common IP-related attacks:

ICMP attacks: Threat actors use ICMP echo packets (pings) to discover subnets and hosts on a protected network, to generate DoS flood attacks, and to alter host routing tables.

Denial-of-service (DoS) attacks: Threat actors attempt to preventlegitimate users from accessing information or services.

Distributed DoS (DDoS) attacks: Similar to a DoS attack, but features a simultaneous, coordinated attack from multiple source machines.

Address spoofing attacks: Threat actors spoof the source IP address in an IP packet to perform blind spoofing or non-blind spoofing.

Man-in-the-middle (MITM) attacks: Threat actors position themselves between a source and destination to transparently monitor, capture, and control the communication. They could simply eavesdrop by inspecting captured packets or alter packets and forward them to their original destination.
Session hijacking: Threat actors gain access to the physical network, and then use an MITM attack to hijack a session.

ICMP Attacks (7.2.1.5)

ICMP was developed to carry diagnostic messages and to report error conditions when routes, hosts, and ports are unavailable. ICMP messages are generated by devices when a network error or outage occurs. The **ping** command is a user-generated ICMP message, called an Echo Request, used to verify connectivity to a destination.

Threat actors use ICMP for reconnaissance and scanning attacks. This enables them to launch information-gathering attacks to map out a network topology, discover which hosts are active (reachable), identify the host operating system (OS fingerprinting), and determine the state of a firewall.

Threat actors also use ICMP for DoS attacks, as shown in the ICMP flood attack in Figure 7-7.

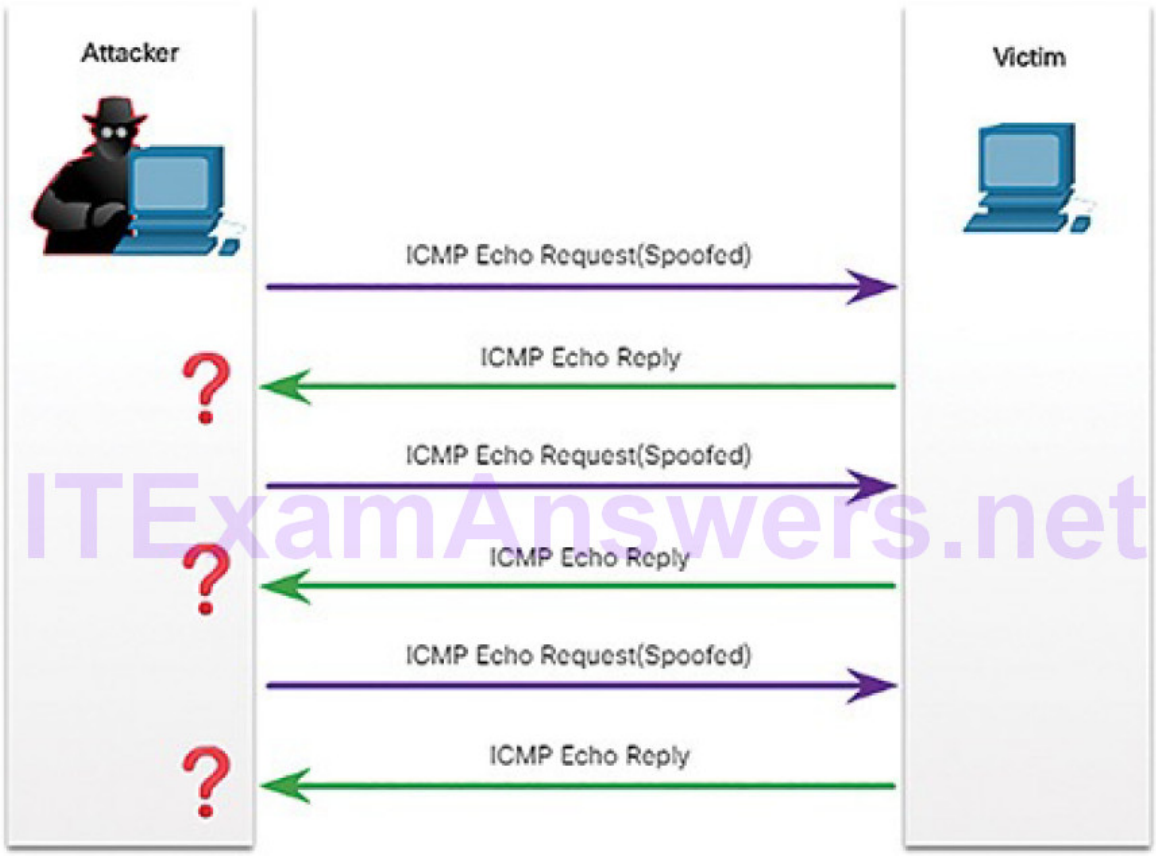

Figure 7-7 ICMP Flood

Note
ICMP for IPv4 (ICMPv4) and ICMP for IPv6 (ICMPv6) are susceptible to similar types of attacks.

Common ICMP messages of interest to threat actors include:

ICMP Echo Request and Echo Reply: These are used to perform host verification and DoS attacks.

ICMP Unreachable: This is used to perform network reconnaissance and scanning attacks.

ICMP Mask Reply: This is used to map an internal IP network.

ICMP Redirect: This is used to lure a target host into sending all traffic through a compromised device and create an MITM attack.

ICMP Router discovery: This is used to inject bogus route entries into the routing table of a target host.

Networks should have strict ICMP access control list (ACL) filtering on the network edge to avoid ICMP probing from the Internet. Security analysts should be able to detect ICMP-related attacks by looking at captured traffic and logfiles. In the case of large networks, security devices such as firewalls and IDSs should detect such attacks and generate alerts to the security analysts.

DoS Attacks (7.2.1.6)

DoS is one of the most common type of attacks. The goal of a DoS attack is to prevent legitimate users from gaining access to websites, email, online accounts, and other services.

There are two major sources of DoS attacks:

Maliciously formatted packets: Threat actors craft a maliciously formatted packet and forward it to a susceptible host, causing it to crash or become extremely slow.

Overwhelming quantity of traffic: Threat actors overwhelm a target network, host, or application, causing it to crash or become extremely slow.

A DDoS attack combines multiple DoS attacks. In Figure 7-8, the attacker first infects computers to create zombies.

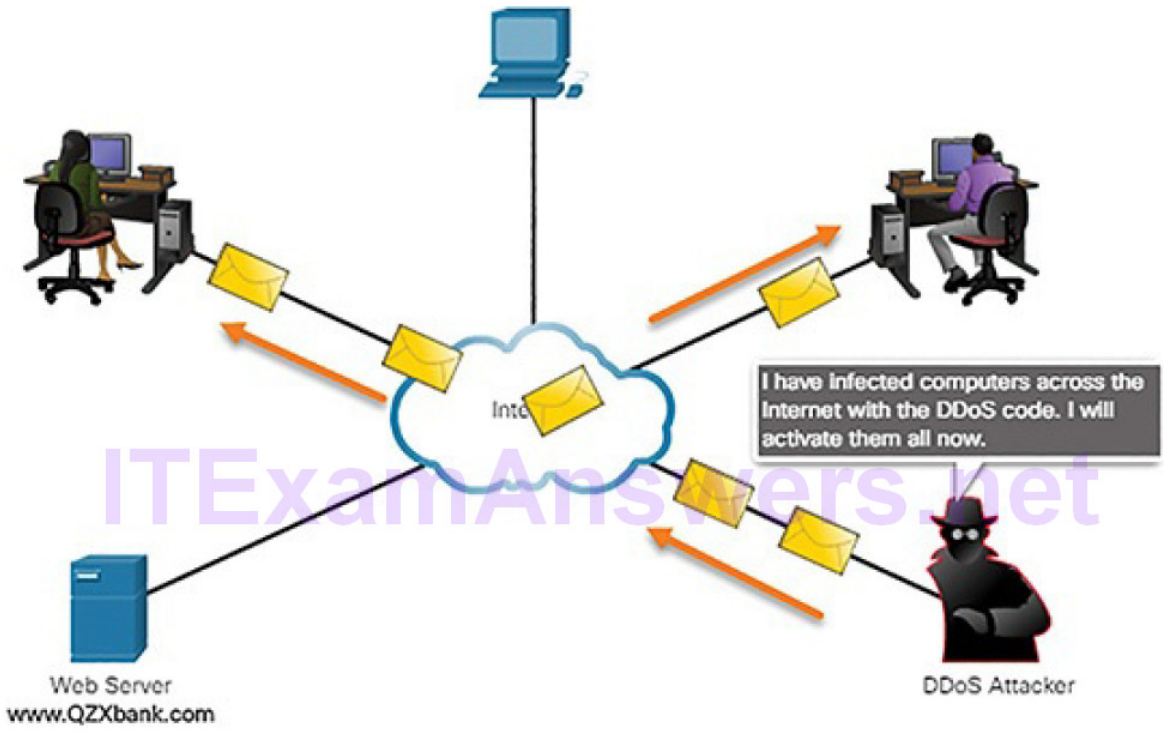

Figure 7-8 DDoS Attack: Activating Zombies

Then, in Figure 7-9, the attacker activates the zombies to attack the victimserver.

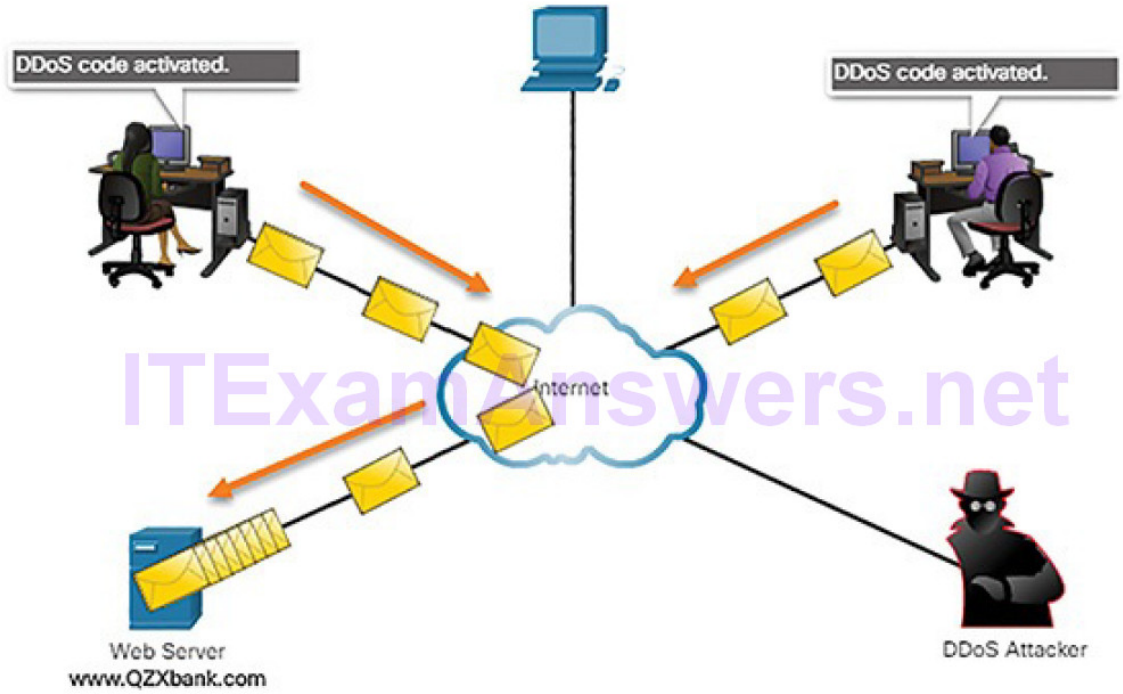

Figure 7-9 DDoS Attack: Zombies Attack Target

ICMP is often used to create DoS attacks. For example, threat actors use ICMP messages to significantly saturate and slow down the target device.

Amplification and Reflection Attacks (7.2.1.7)

Threat actors often use amplification and reflection techniques to create DoS attacks. The example in Figure 7-10 illustrates how an amplification and reflection technique called a Smurf attack is used to overwhelm a target host:

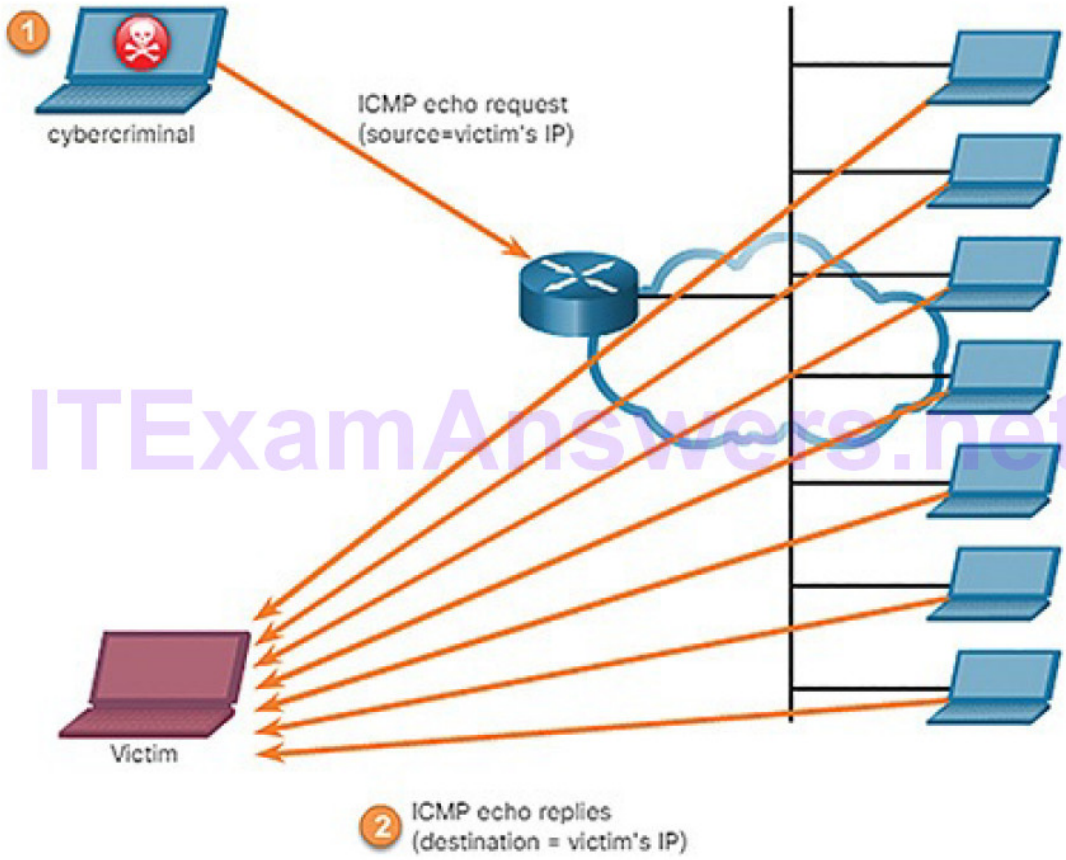

Figure 7-10 Amplification and Reflection Attack

1. Amplification: The threat actor forwards ICMP echo request messages that contain the source IP address of the victim to a large number of hosts.

2. Reflection: These hosts all reply to the spoofed IP address of the victim to overwhelm it.

Note
Newer forms of amplification and reflection attacks are now being used, such as DNS-based reflection and amplification attacks and Network Time Protocol (NTP) amplification attacks.

Threat actors also use resource exhaustion attacks to consume the resources of a target host to crash it or to consume the resources of a network to adversely affect its operation.

DDoS Attacks (7.2.1.8)

A DDoS attack is similar in intent to a DoS attack, except that a DDoS attack increases in magnitude because it originates from multiple, coordinated sources.

DDoS attacks also introduce new terms such as botnet, handler systems, and zombie computers.

A DDoS attack could proceed as follows:

1. The threat actor (botmaster) builds or purchases the use of a botnet of zombie hosts. The command-and-control (CnC) server communicates with zombies over a covert channel using IRC, P2P, DNS, HTTP, or HTTPS.

2. Zombie computers continue to scan and infect more targets to create more zombies.

3. When ready, the botmaster uses the handler systems to make the botnet of zombies carry out the DDoS attack on the chosen target.

In Figure 7-11, the threat actor communicates with the zombies using the CnC server to launch a DDoS attack against the victim's infrastructure.

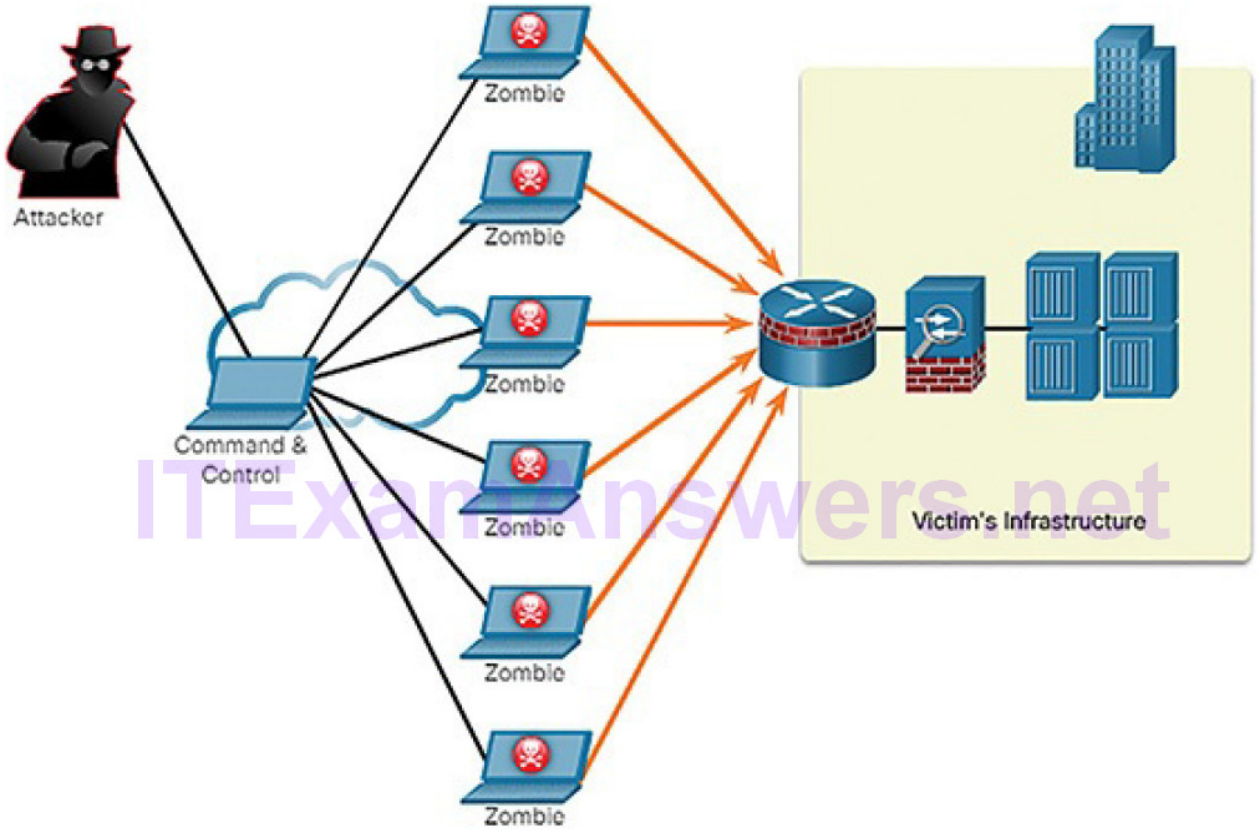

Figure 7-11 DDoS Attack

Bots have a worm-like ability to self-propagate but they can also be used to log keystrokes, gather passwords, capture and analyze packets, gather financial information, launch DoS attacks, relay spam, and open back doors on the infected host.

There are many potential sources of DoS and DDoS attacks. While DDoS attacks are very easy to detect, they are hard to combat. Insecure IoT devices have been exploited to exponentially increase the size of botnets. There are a few countermeasures that can be used to fight these attacks:

- Implement firewalls and IPS monitoring
- Rate-limit incoming and outgoing traffic to normal baseline settings
- Maximize the memory and harden all devices

Address Spoofing Attacks (7.2.1.9)

IP address spoofing attacks occur when a threat actor creates packets with false source IP address information to either hide the identity of the sender or to pose as another legitimate user. The attacker can then gain access to otherwise inaccessible data or circumvent security configurations. Spoofing is usually incorporated into another attack such as a Smurf attack.

Spoofing attacks can be conducted as follows:

Non-blind spoofing: The threat actor can see the traffic that is being sent between the host and the target. Non-blind spoofing is used by the threat actor to inspect the reply packet from the target victim. Reasons for non-blind spoofing include determining the state of a firewall, sequence-number prediction, or hijacking an authorized session.

Blind spoofing: The threat actor cannot see the traffic that is being sent between the host and the target. Blind spoofing is used in DoS attacks.

MAC address spoofing attacks are used when threat actors have access to the internal network. Threat actors alter the MAC address of their host to match another known MAC address of a target host, as shown in Figure 7-12.

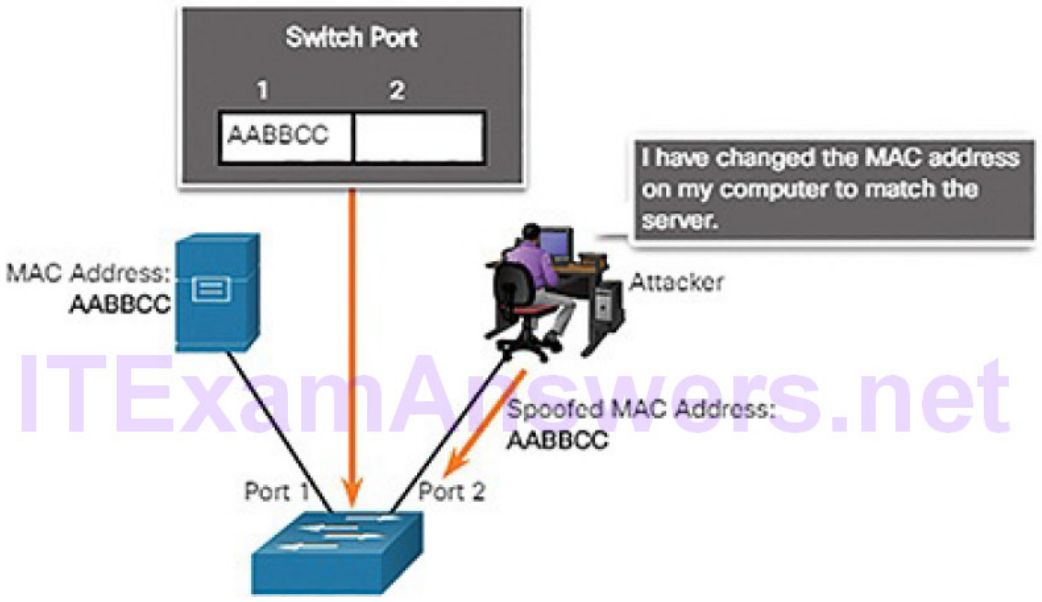

Figure 7-12 Attacker Spoofs a Server's MAC Address

The attacking host then sends a frame throughout the network with the newly configured MAC address. When the switch receives the frame, it examines the source MAC address. The switch overwrites the current CAM table entry and assigns the MAC address to the new port, as shown in Figure 7-13. It then forwards frames destined for the target host to the attacking host.

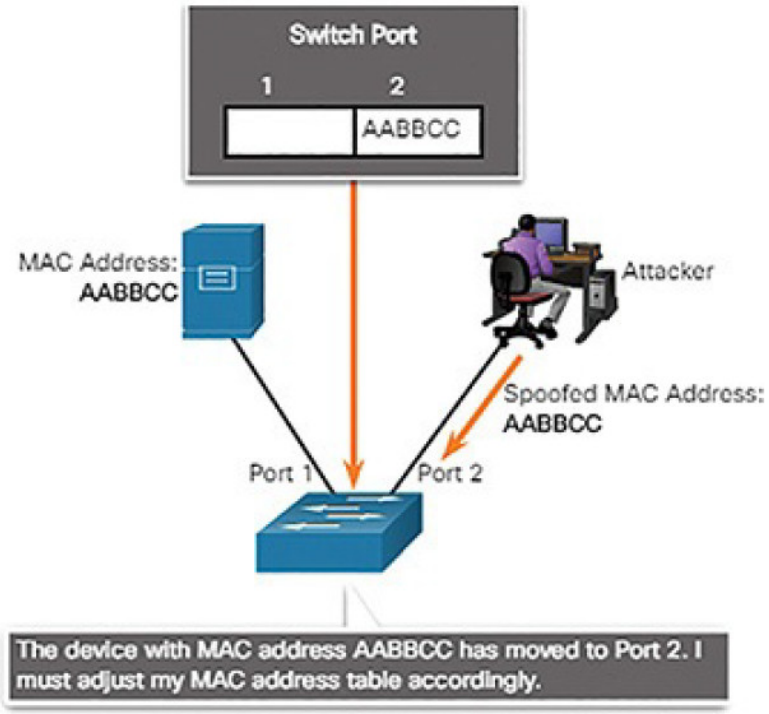

Figure 7-13 Switch Updates CAM Table with Spoofed Address

Application or service spoofing is another spoofing example. A threat actor can connect a rogue DHCP server to create an MITM condition.

Activity 7.2.1.10: Identify the IP Vulnerability
Refer to the online course to complete this Activity.

Lab 7.2.1.11: Observing a DDoS Attack
In this lab, you will simulate and observe a DoS and DDoS attack.

TCP and UDP Vulnerabilities (7.2.2)

In this topic, you will learn how TCP and UDP vulnerabilities enable network attacks.

TCP (7.2.2.1)

Like IP, TCP is also vulnerable. TCP segment information appears immediately after the IP header. The fields of the TCP segment and the flags for the Control Bits field are displayed in Figure 7-14.

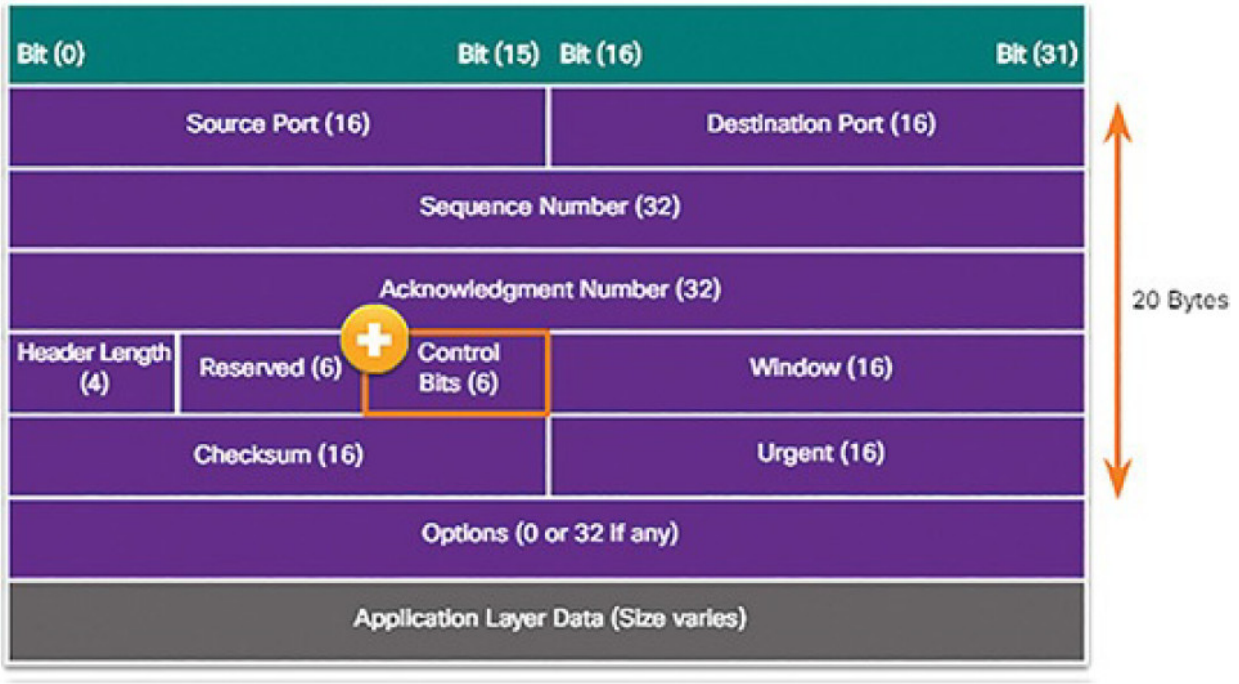

Figure 7-14 TCP Segment

The control bits are as follows:

URG: Urgent pointer field significant
ACK: Acknowledgment field significant
PSH: Push function
RST: Reset the connection
SYN: Synchronize sequence numbers
FIN: No more data from sender

TCP provides the following services:

Reliable delivery: Reliable communication is the largest benefit of TCP. TCP incorporates acknowledgments to guarantee delivery, instead of relying on upper-layer protocols to detect and resolve errors. If a timely acknowledgment is not received, the sender retransmits the data. But requiring acknowledgments of received data can cause substantial delays.

Flow control: TCP implements flow control to address the delay issue. Rather than acknowledge one segment at a time, multiple segments can be acknowledged with a single acknowledgment segment.

Stateful communication: TCP stateful communication between two parties happens by way of a TCP three-way handshake. Before data can be transferred using TCP, a three-way handshake opens the TCP connection as shown in Figure 7-15. If both sides agree to the TCP connection, data can be sent and received by both parties using TCP.

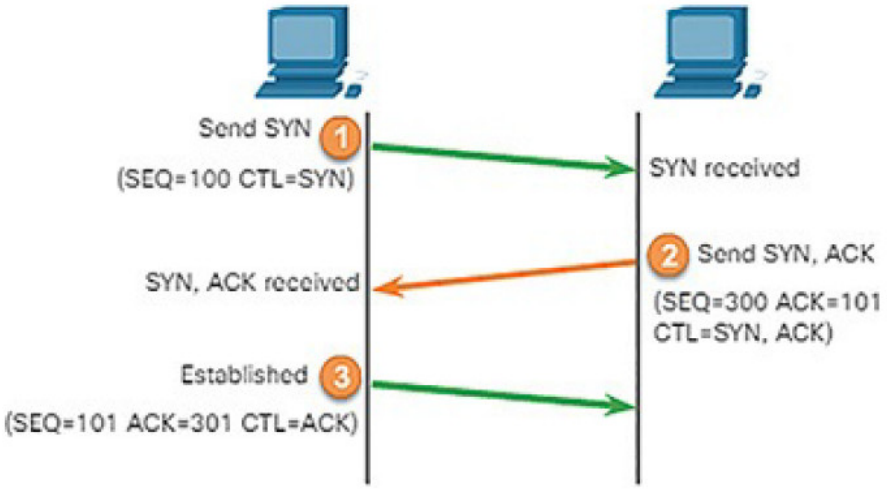

Figure 7-15 Establishing a TCP Connection

Examples of application layer protocols that make use of TCP reliability include HTTP, SSL/TLS, FTP, DNS zone transfers, and others.

TCP Attacks (7.2.2.2)

Though TCP is a connection-oriented and reliable protocol, it still has vulnerabilities that can be exploited.

The TCP protocol is vulnerable to port scanning. Network applications use TCP or UDP ports. Threat actors conduct port scans of target devices to discover which services they offer.

The TCP SYN flood attack exploits the TCP three-way handshake. As illustrated in Figure 7-16, the threat actor continually sends TCP SYN session request packets with a randomly spoofed source IP address to an intended target.

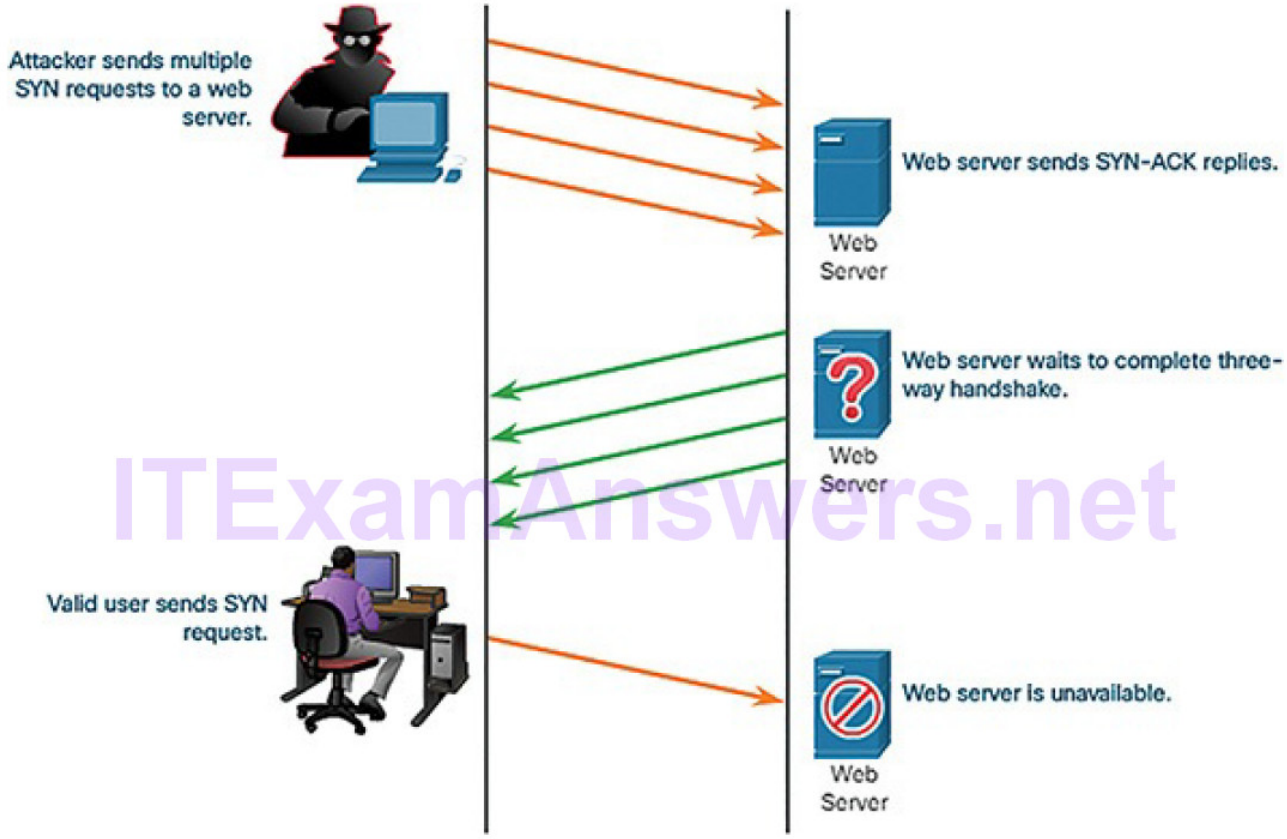

Figure 7-16 Sample TCP SYN Flood Attack

The target device replies with a TCP SYN-ACK packet to the spoofed IP address and waits for a TCP ACK packet. Those responses never arrive. Eventually thetarget host is overwhelmed with half-open TCP connections and denies TCP services (e.g., email, file transfer, or http) to legitimate users.

A TCP reset attack can be used to terminate a TCP communication between two hosts. Figure 7-17 displays how TCP uses a four-way exchange to close the TCP connection using a pair of FIN and ACK segments from each TCP endpoint.

A TCP connection can also be torn down when it receives an RST bit. This is an abrupt way to tear down the TCP connection and inform the receiving host to immediately stop using the TCP connection. A threat actor could do a TCP reset attack and send a spoofed packet containing a TCP RST to one or both endpoints.

TCP session hijacking is another TCP vulnerability. Although difficult to conduct, it enables a threat actor to overtake an already-authenticated host as it communicates with the target. The threat actor would have to spoof the IP address of one host, predict the next sequence number, and send an ACK to the other host. If successful, the threat actor could send data to, but not receive data from, the target device.

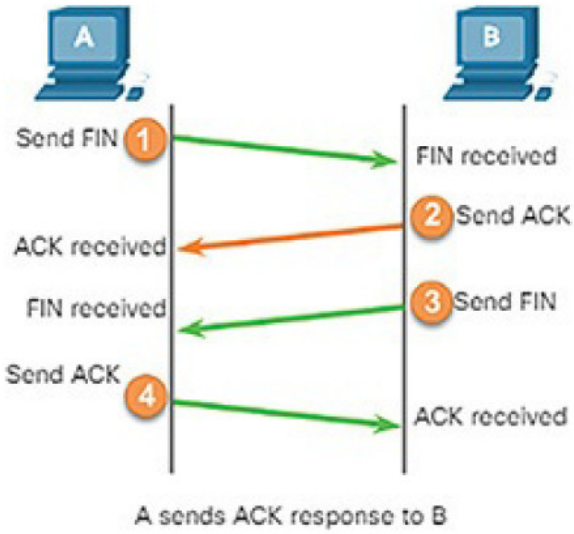

Figure 7-17 Terminating a TCP Connection

UDP and UDP Attacks (7.2.2.3)

UDP is a simple protocol that provides the basic transport layer functions. UDP is commonly used by DNS, TFTP, NFS, and SNMP. It is also used with real-time applications such as media streaming or VoIP. UDP is a connectionless transportlayer protocol. It has much lower overhead than TCP because it is not connection-oriented and does not offer the sophisticated retransmission, sequencing, and flow control mechanisms that provide reliability. The UDP segment structure is much smaller than TCP's segment structure, and is shown in Figure 7-18.

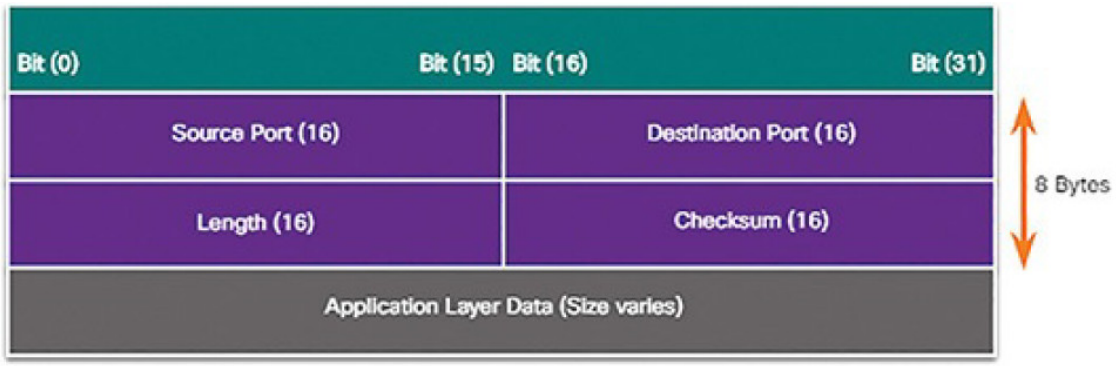

Figure 7-18 UDP Data Structure

This does not mean that applications that use UDP are always unreliable, nor does it mean that UDP is an inferior protocol. It means that these functions are not provided by the transport layer protocol and must be implemented elsewhere if required.

The low overhead of UDP makes it very desirable for protocols that make simple request and reply transactions. For example, using TCP for DHCP would introduce unnecessary network traffic. If there is a problem with a request or a reply, the device simply sends the request again if no response is received.

UDP is not protected by any encryption. It is possible to add encryption to UDP, but it is not available by default. The lack of encryption allows anyone to look at the traffic, change it, and send it on to its destination. Changing the data in the traffic will alter the 16-bit checksum, but the checksum is optional and not always used. When the checksum is used, the attacker can create a new checksum based on the new data payload, and record it in the header as a new checksum. The destination device will find that the checksum matches the data without knowing the data has been altered.

This type of attack is not the most widely used. It is more common to see a UDP attack where all of the resources on a network are consumed. This is called a UDP flood attack. To do this, the attacker must use a tool like UDP Unicorn or Low Orbit Ion Cannon (LOIC) that sends a flood of UDP packets, often from aspoofed host, to a server on the subnet. The program will sweep through all of the known ports trying to find closed ports. This will cause the server to reply with an ICMP port unreachable message. Because there are so many closed ports on the server, this causes so much traffic on the segment that almost all of the bandwidth gets used. The result is very similar to a DoS attack.

Lab 7.2.2.4: Observing TCP Anomalies
In this lab, you will complete the following objectives:

- Load Mininet and Start Services
- Enable Wireshark to Capture and Observe Normal Traffic Packets
- Use Wireshark to Capture SYN Flood Attack Packets
- Stopping Mininet

Attacking What We Do (7.3)

In this section, you will learn how common network applications and services are vulnerable to attack.

IP Services (7.3.1)

In this topic, you will learn about IP vulnerabilities.

ARP Vulnerabilities (7.3.1.1)

Hosts broadcast an ARP Request to other hosts on the segment to determine the MAC address of a host with a particular IP address. All hosts on the subnet receive and process the ARP Request. The host with the matching IP address in the ARP Request (Figure 7-19) sends an ARP Reply (Figure 7-20).

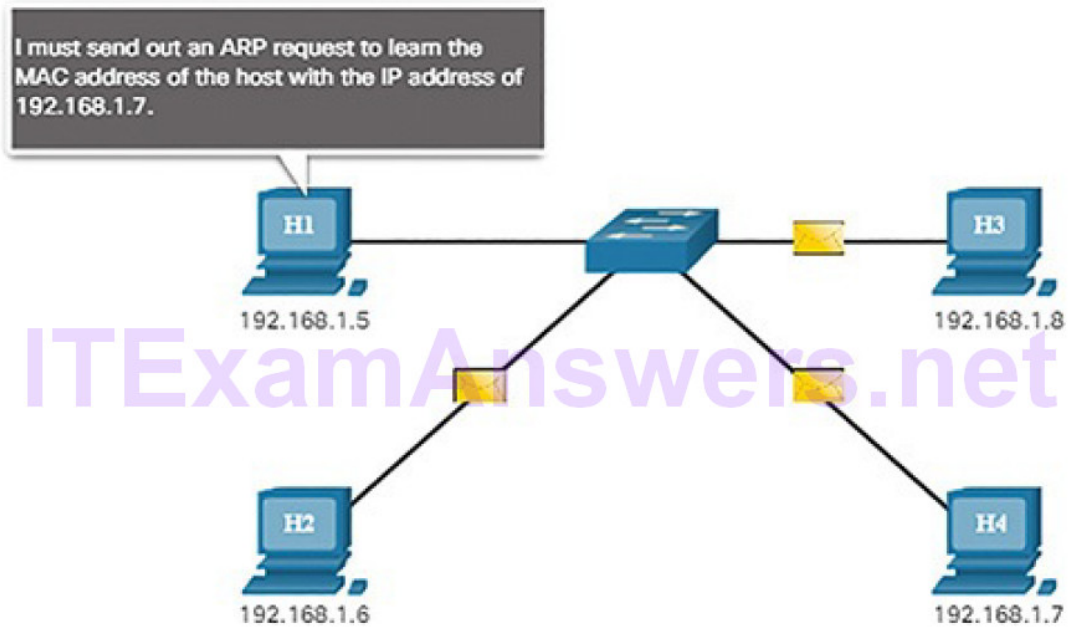

Figure 7-19 The ARP Process: ARP Request

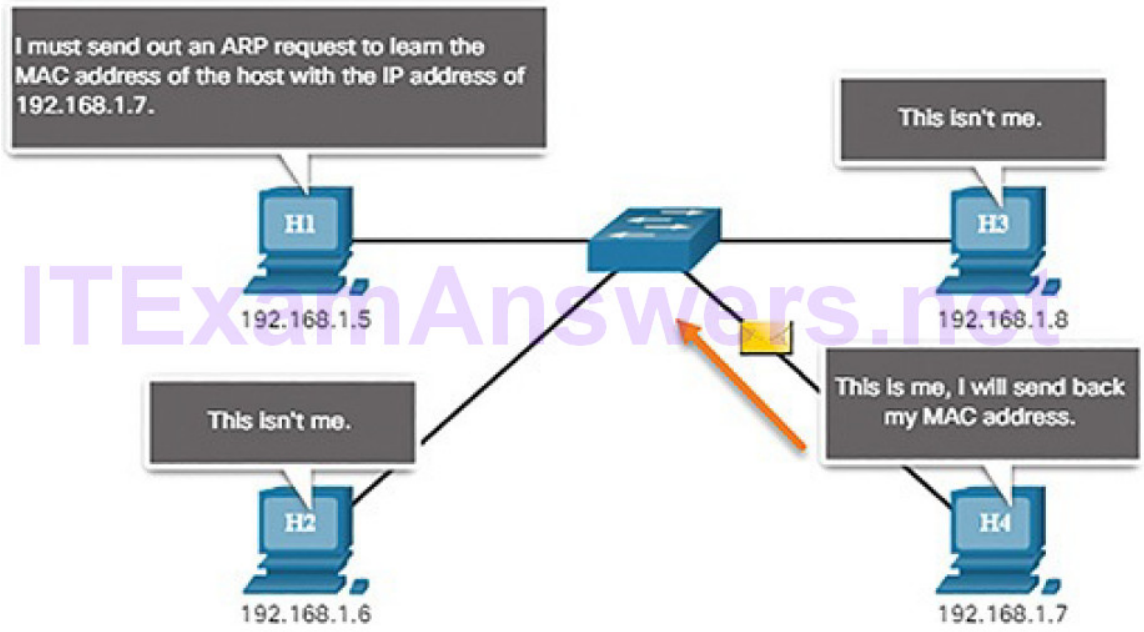

Figure 7-20 The ARP Process: ARP Reply

Any client can send an unsolicited ARP Reply called a “gratuitous ARP.” This is often done when a device first boots up to inform all other devices on the local network of the new device’s MAC address. When a host sends a gratuitous ARP, other hosts on the subnet store the MAC address and IP address contained in the gratuitous ARP in their ARP tables.

However, this feature of ARP also means that any host can claim to be the owner of any IP/MAC they choose. A threat actor can poison the ARP cache of deviceson the local network, creating an MITM attack to redirect traffic. The goal is to target a victim host and have it

change its default gateway to the threat actor's device. This positions the threat actor in between the victim and all other systems outside of the local subnet.

ARP Cache Poisoning (7.3.1.2)

To see how ARP cache poisoning works, consider the following example. In Figure 7-21, PC-A requires the MAC address of its default gateway (R1) and therefore sends an ARP Request for the MAC address of 192.168.10.1.

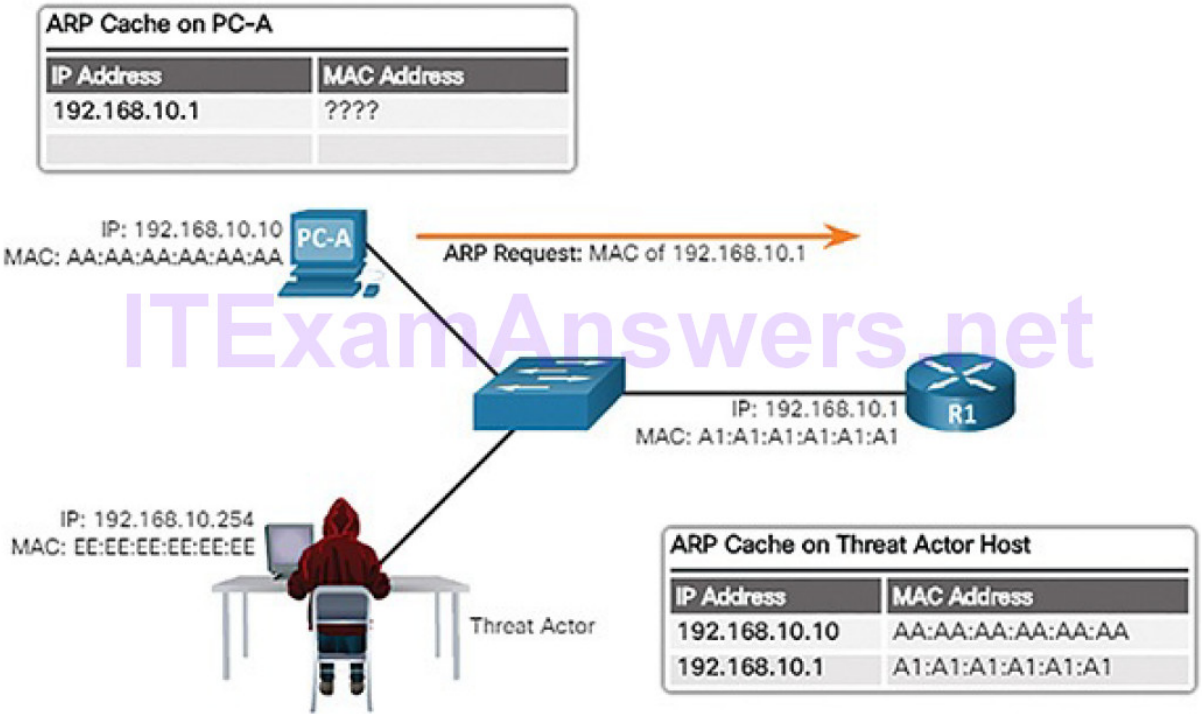

Figure 7-21 PC-A Sends an ARP Request to the Default Gateway

In Figure 7-22, R1 updates its ARP cache with the IP and MAC addresses of PC-A and sends an ARP Reply to PC-A, which then updates its ARP cache with the IP and MAC addresses of R1.

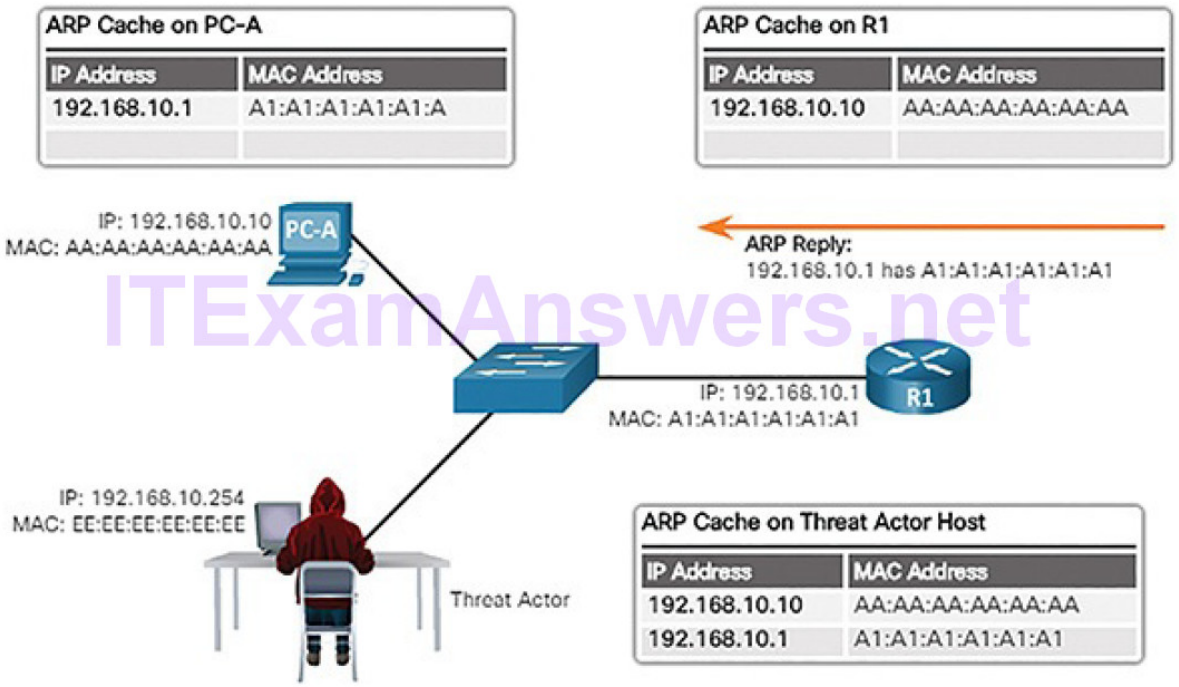

Figure 7-22 R1 Sends ARP Reply

In Figure 7-23, the threat actor sends two spoofed gratuitous ARP Replies using its own MAC address for the indicated destination IP addresses. PC-A updates its ARP cache with its default gateway now pointing to the threat actor's host MAC. R1 also updates its ARP cache with the IP address of PC-A pointing to the threat actor's MAC address.

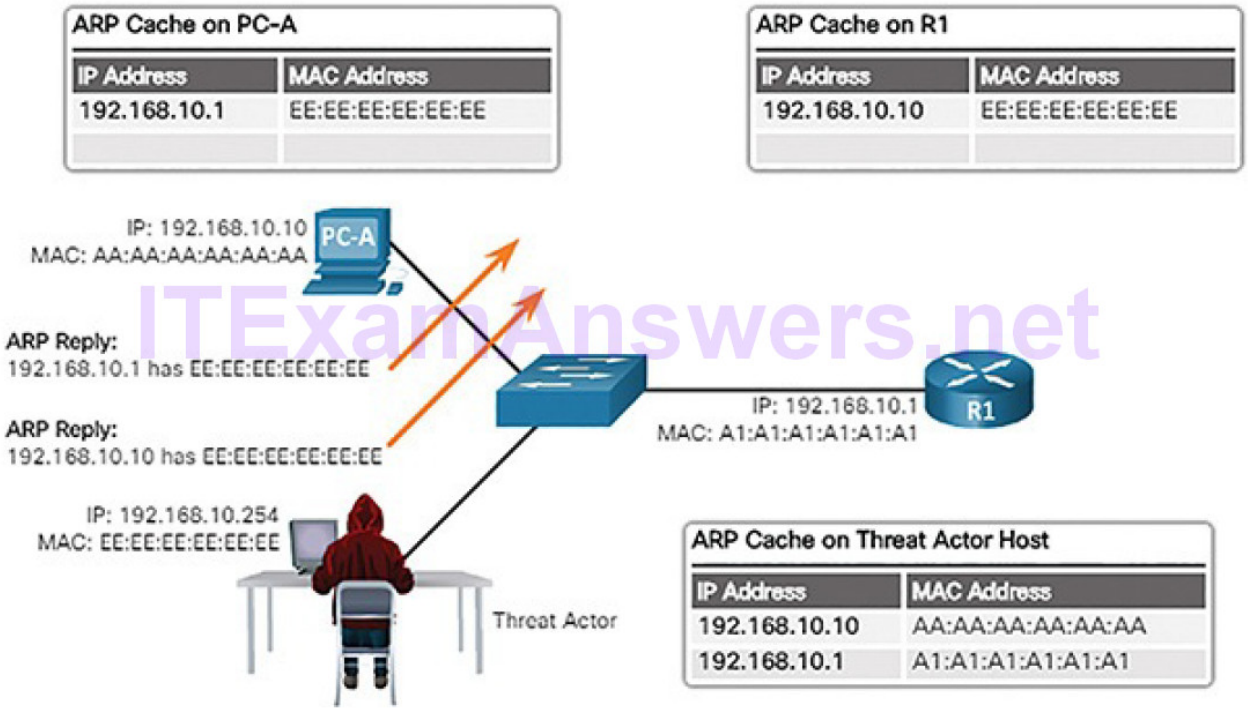

Figure 7-23 Threat Actor Sends Spoofed Gratuitous Replies

The threat actor's host is now doing an ARP poisoning attack.

Note
There are many tools available on the Internet to create ARP MITM attacks, including dsniff, Cain & Abel, Ettercap, Yersinia, and others.

The ARP poisoning attack can be

- **Passive:** Threat actors steal confidential information.
- **Active:** Threat actors modify data in transit or inject malicious data.

DNS Attacks (7.3.1.3)

The Domain Name Service (DNS) protocol defines an automated service that matches resource names with the required numeric network address. It includes the format for queries, responses, and data and uses resource records (RRs) to identify the type of DNS response.

Securing DNS is often overlooked. However, it is crucial to the operation of a network and should be secured accordingly.

Many organizations use the services of publicly open DNS servers such as Google DNS (8.8.8.8) to provide responses to queries. This type of DNS server is called an open resolver. A DNS open resolver answers queries from clients outside of its administrative domain.

DNS open resolvers are vulnerable to multiple malicious activities, including

DNS cache poisoning attacks: Threat actors send spoofed, falsified RR information to a DNS resolver to redirect users from legitimate sites to malicious sites. DNS cache poisoning attacks can all be used to inform the DNS resolver to use a malicious name server that is providing RR information for malicious activities.

DNS amplification and reflection attacks: Threat actors use DNS open resolvers to increase the volume of attacks and to hide the true source of an attack. This technique is used in DoS or DDoS attacks. These attacks are possible because the open resolver will respond to queries from anyone asking a question. Threat actors send DNS messages to the open resolvers using the IP address of a target host (victim).

DNS resource utilization attacks: Threat actors launch a DoS attack that consumes the resources of the DNS open resolvers. Examples of such resources include CPU, memory, and socket buffers. This DoS attack consumes all the available resources to negatively affect the operations of the DNS open resolver. The impact of this DoS attack may require the DNS open resolver to be rebooted or services to be stopped and restarted.

To hide their identity, threat actors also use the following DNS stealth techniques to carry out their attacks:

Fast flux: Threat actors use this technique to hide their phishing and malware delivery sites behind a quickly changing network of compromised DNS hosts. The DNS IP addresses are continuously changed within minutes. Botnets often employ fast flux techniques to effectively hide (i.e., cloak) malicious servers from being detected.

Double IP flux: Threat actors use this technique to rapidly change the hostname to IP address mappings and to also change the authoritative name server. This increases the difficulty of identifying the source of the attack.

Domain generation algorithms: Threat actors use this technique in malware to randomly generate domain names that can then be used as rendezvous points to their command and control (CnC) servers.

Other threats to DNS include DNS shadowing attacks and DNS tunneling. DNS tunneling is discussed next. Domain shadowing involves the threat actor compromising a parent domain and creating multiple subdomains to be used during the attacks.

DNS Tunneling (7.3.1.4)

Botnets have become a popular attack method of threat actors. Most often, botnets are used to spread malware or launch DDoS and phishing attacks.

DNS in the enterprise is sometimes overlooked as a protocol which can be used by botnets. Because of this, when DNS traffic is determined to be part of an incident, the attack is already over. It is necessary for the security analyst to be able to detect when an attacker is using DNS tunneling to steal data, and prevent and contain the attack. To accomplish this, the security analyst must implement a solution that can block the outbound communications from the infected hosts.

Threat actors who use DNS tunneling place non-DNS traffic within DNS traffic. This method often circumvents security solutions. For the threat actor to use DNS tunneling, the different types of DNS records such as TXT, MX, SRV, NULL, A, or CNAME are altered. For example, the TXT record can store the most commands for sending to the infected hosts over DNS replies. A DNS tunneling attack using TXT works like this:

1. The data is split into multiple encoded chunks.
2. Each chunk is placed into a lower level domain name label of the DNS query.
3. Because there is no response from the local or networked DNS for the query, the request is sent to the ISP's recursive DNS servers.
4. The recursive DNS service forwards the query to the attacker's authoritative name server.
5. The process is repeated until all of the queries containing the chunks are sent.
6. When the attacker's authoritative name server receives the DNS queries from the infected devices, it sends responses for each DNS query, which contain the encapsulated, encoded

commands.
7. The malware on the compromised host recombines the chunks and executes the commands hidden within.

To be able to stop DNS tunneling, a filter that inspects DNS traffic must be used.Pay particular attention to DNS queries that are longer than average, or those that have a suspicious domain name. Also, DNS solutions, like Cisco OpenDNS, block much of the DNS tunneling traffic by identifying suspicious domains.

DHCP (7.3.1.5)

DHCP servers dynamically provide IP configuration information including IP address, subnet mask, default gateway, DNS servers, and more to clients. The typical sequence of DHCP message exchange between client and server is displayed in Figure 7-24.

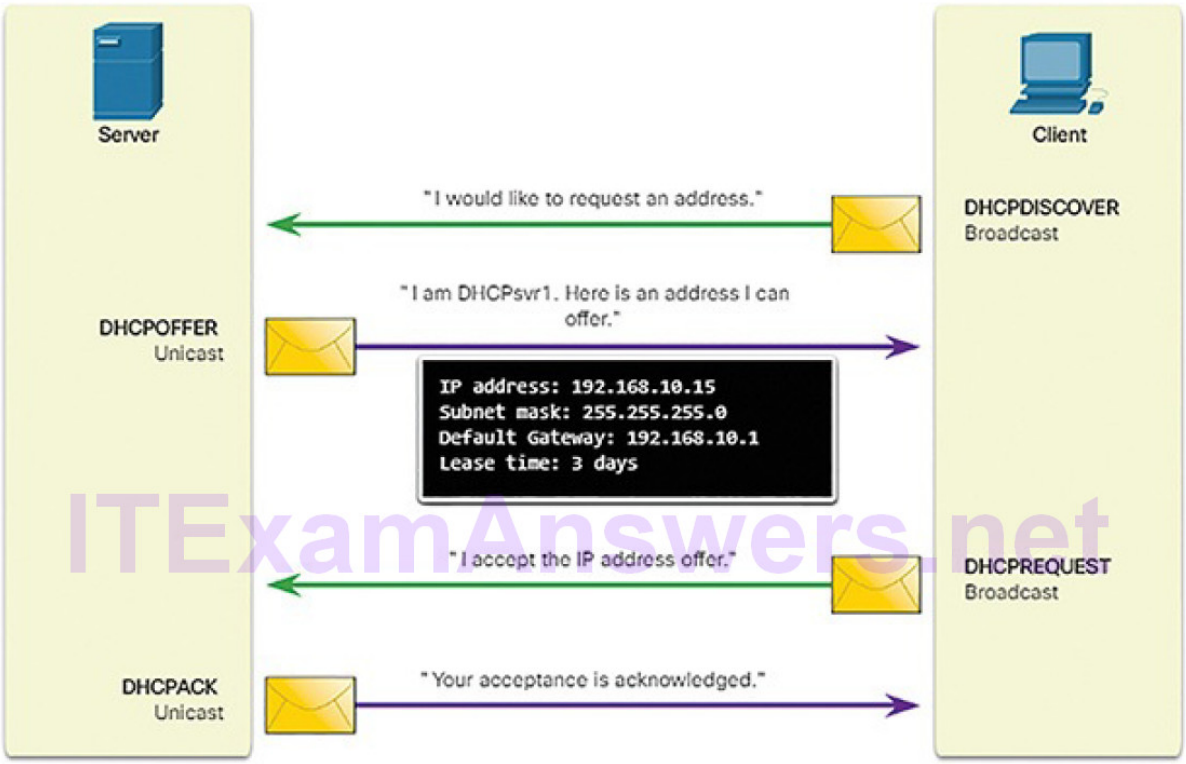

Figure 7-24 Normal DHCP Operation

DHCP is vulnerable to DHCP spoofing attacks. A DHCP spoofing attack occurs when a rogue DHCP server is connected to the network and provides false IP configuration parameters to legitimate clients. A rogue server can provide a variety of misleading information:

Wrong default gateway: The threat actor provides an invalid gateway orthe IP address of its host to create an MITM attack. This may go entirely undetected as the intruder intercepts the data flow through the network.

Wrong DNS server: The threat actor provides an incorrect DNS server address pointing the user to a malicious website.

Wrong IP address: The threat actor provides an invalid IP address, invalid default gateway IP address, or both invalid IP address and default gateway. The threat actor then creates a DoS attack on the DHCP client.

Figures 7-25 through 7-28 illustrate a DHCP spoofing attack. Assume a threat actor has successfully connected a rogue DHCP server to a switch port on the same subnet as the target clients. The goal of the rogue server is to provide clients with false IP configuration information.

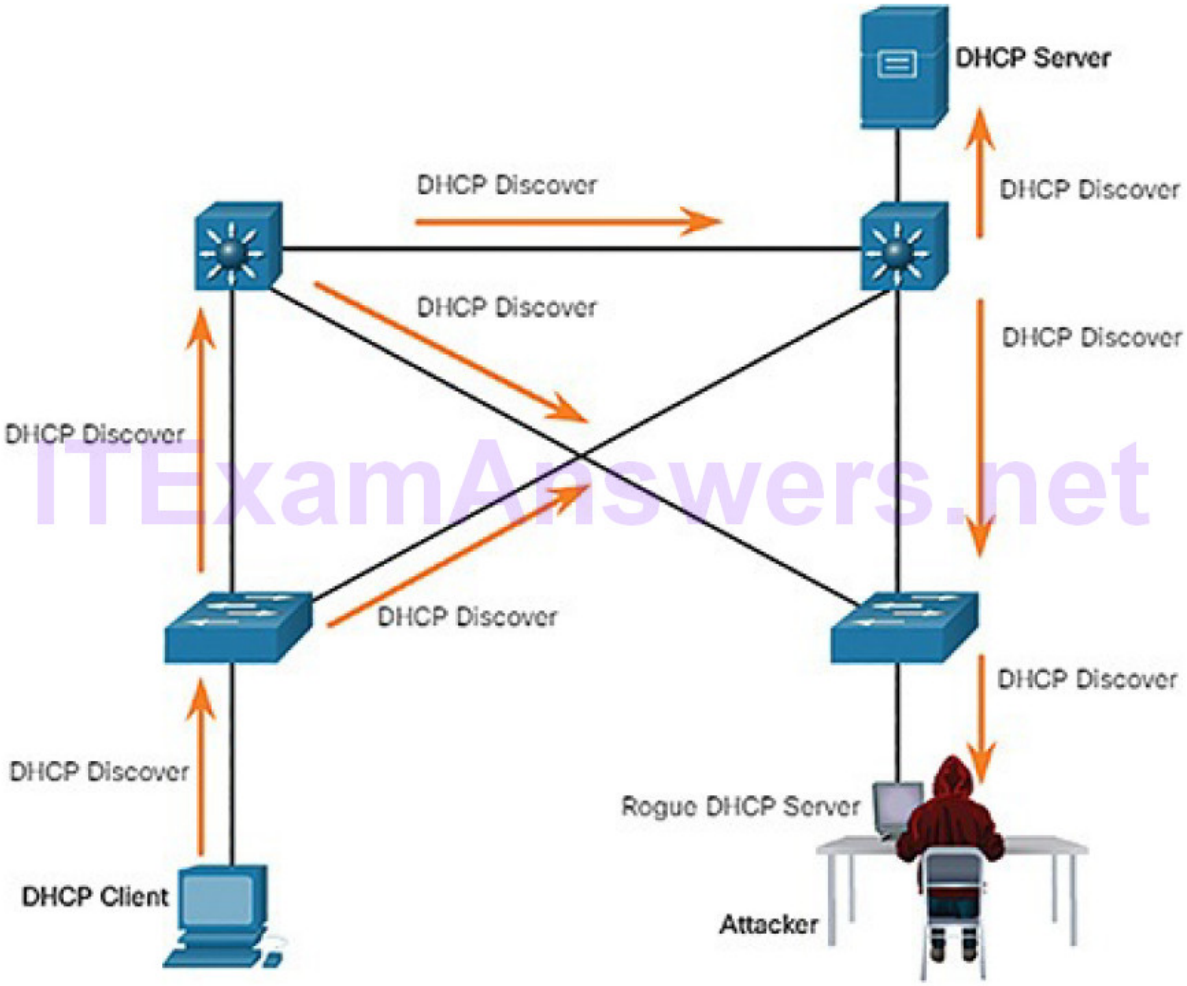

Figure 7-25 Client Broadcasts DHCP Discovery Messages

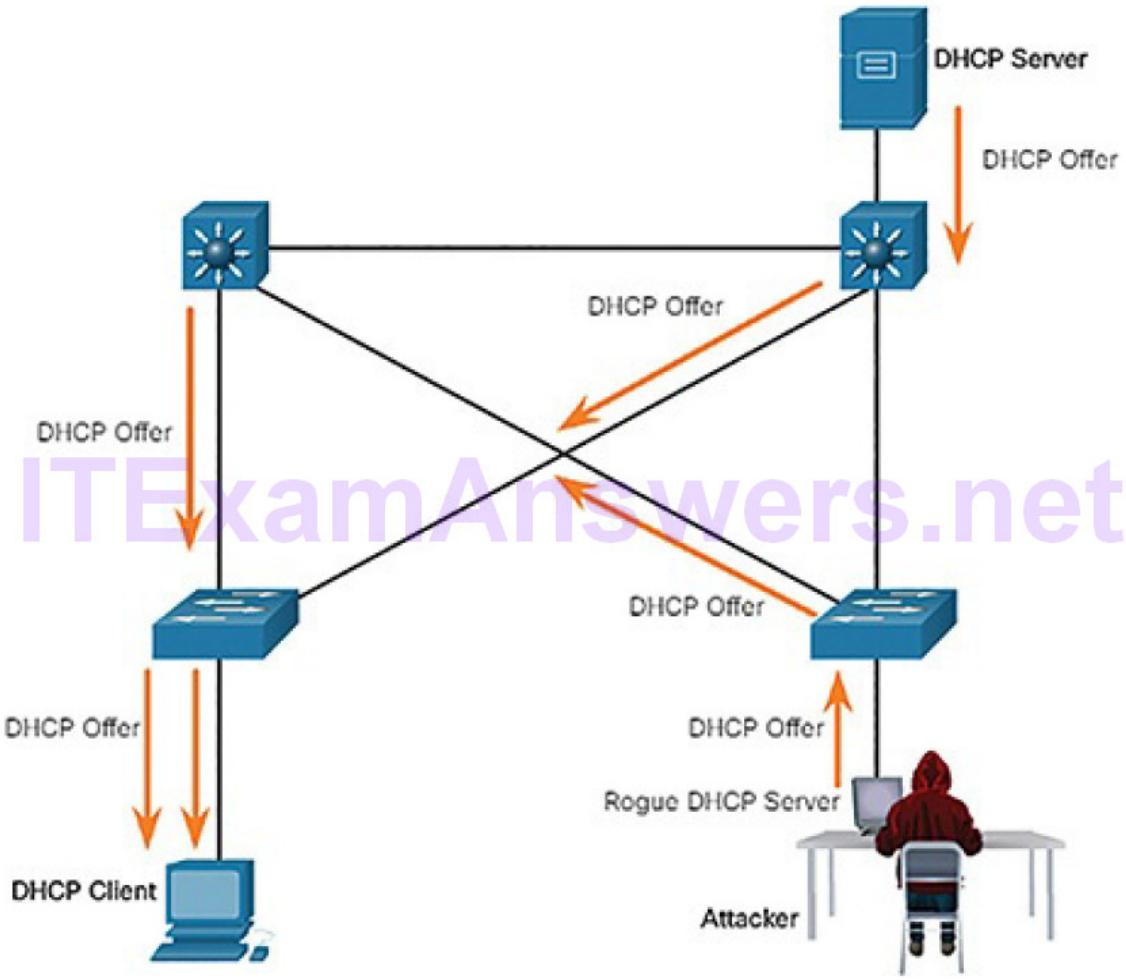

Figure 7-26 Legitimate and Rogue DHCP Reply

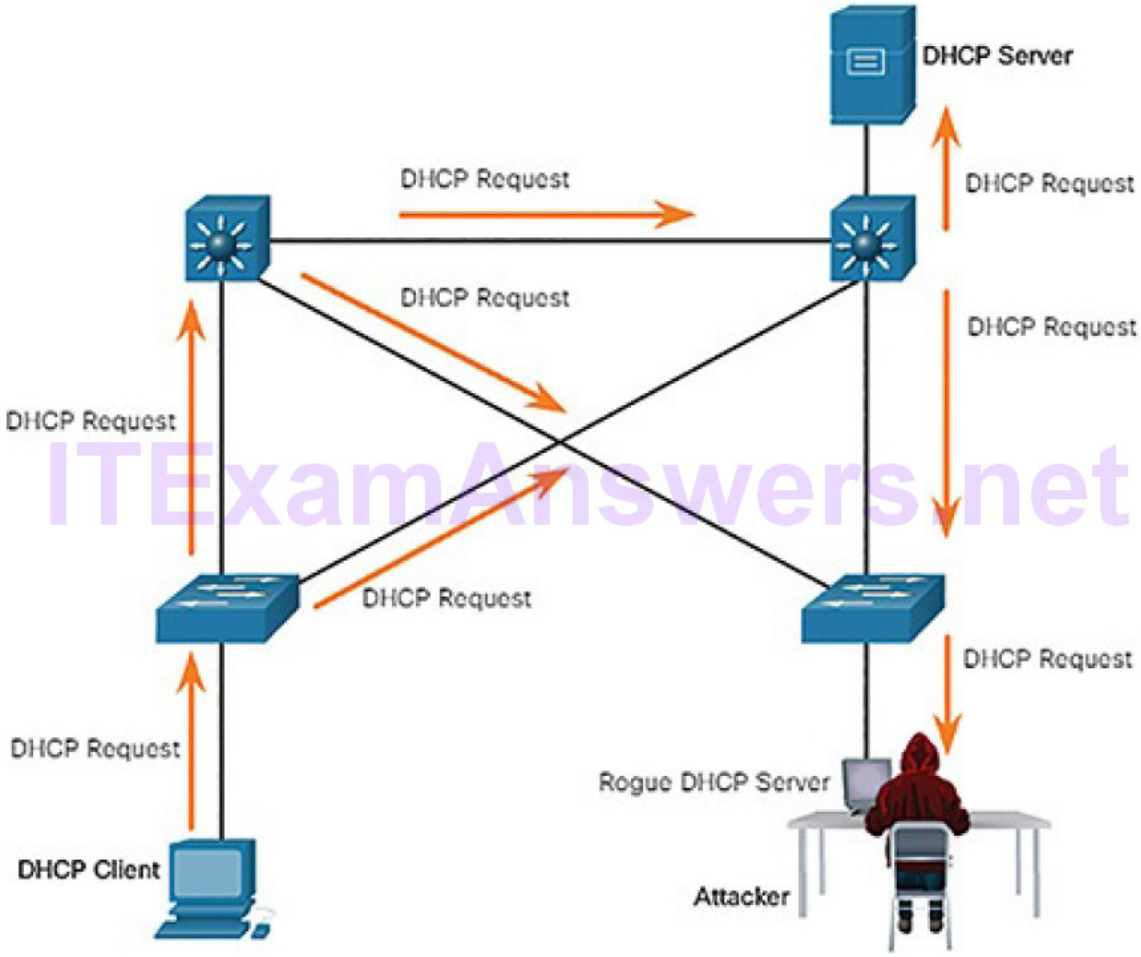

Figure 7-27 Client Accepts Rogue DHCP Offer

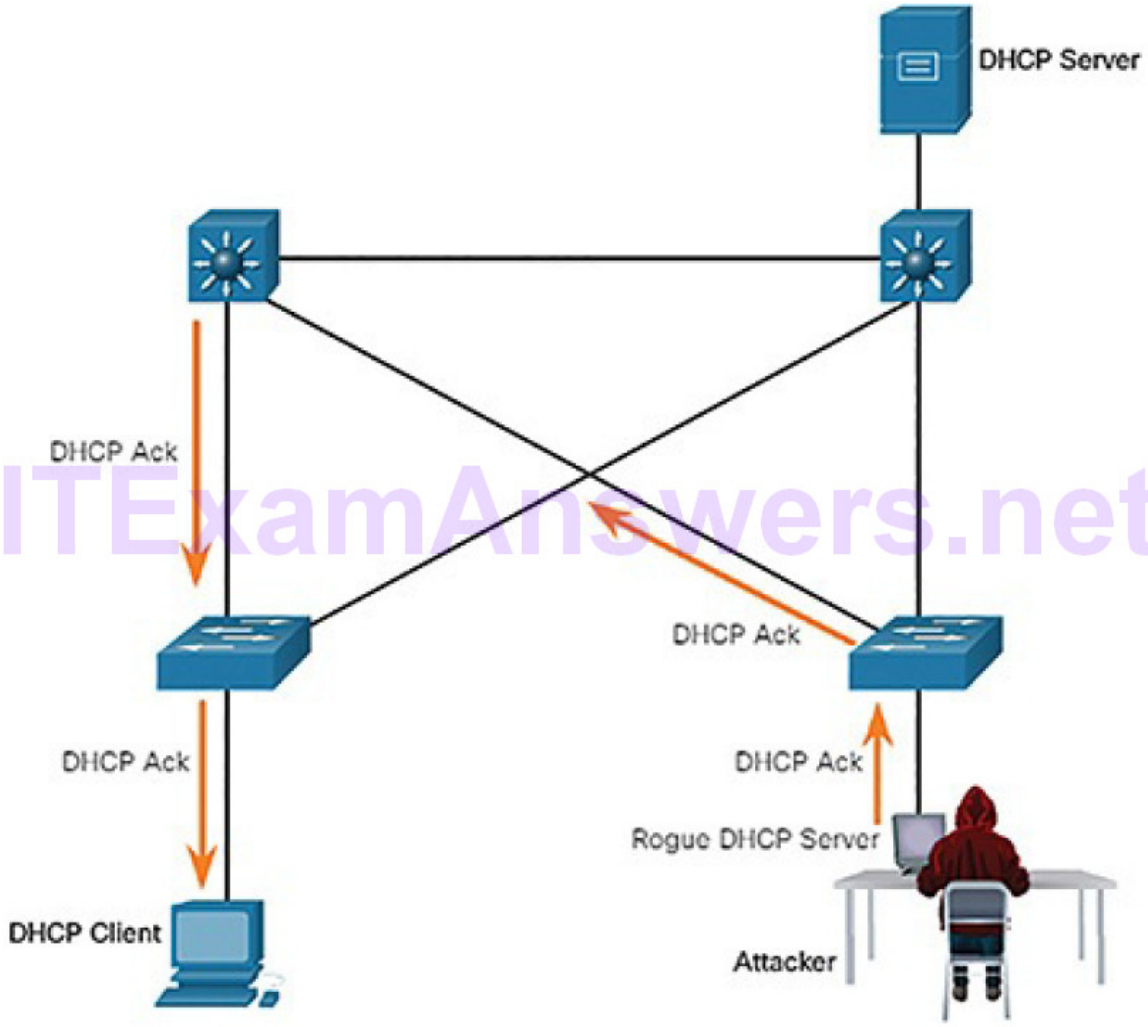

Figure 7-28 Rogue Acknowledges

In Figure 7-25, a legitimate client connects to the network and requires IP configuration parameters. Therefore, the client broadcasts a DHCP Discover request looking for a response from a DHCP server. Both servers will receive the message. Figure 7-26 illustrates how the legitimate and rogue DHCP servers each respond with valid IP configuration parameters. The client will reply to the first offer received.

In this scenario, the client received the rogue offer first. It broadcasts a DHCP request accepting the parameters from the rogue server as shown in Figure 7-27. The legitimate and rogue server will receive the request. However, as shown in Figure 7-28, only the rogue server unicasts a reply to the client to acknowledge its request. The legitimate server will cease communicating with the client.

DHCP is also vulnerable to a DHCP starvation attack. The goal of this attack is to create a DoS for connecting clients. DHCP starvation attacks require an attack tool such as Gobbler. Gobbler forwards DHCP discovery messages with bogusMAC addresses in an attempt to lease the entire pool of addresses.

Lab 7.3.1.6: Exploring DNS Traffic

In this lab, you will complete the following objectives:

- Capture DNS Traffic
- Explore DNS Query Traffic
- Explore DNS Response Traffic

Enterprise Services (7.3.2)

In this topic, you will learn how network application vulnerabilities enable network attacks.

HTTP and HTTPS (7.3.2.1)

Internet browsers are used by almost everyone. Blocking web browsing completely is not an option because businesses need access to the web, without undermining web security.

To investigate web-based attacks, security analysts must have a good understanding of how a standard web-based attack works. These are the common stages of a typical web attack:

1. The victim unknowingly visits a web page that has been compromised by malware.

2. The compromised web page redirects the user, often through many compromised servers, to a site containing malicious code.

3. The user visits this site with malicious code and their computer becomes infected. This is known as a drive-by download. When the user visits the site, an exploit kit scans the software running on the victim's computer, including the OS, Java, and Flash player, looking for an exploit in the software. The exploit kit is often a PHP script and provides the attacker with a management console to manage the attack.

4. After identifying a vulnerable software package running on the victim's computer, the exploit kit contacts the exploit kit server to download code that can use the vulnerability to run malicious code on the victim'scomputer.

5. After the victim's computer has been compromised, it connects to the malware server and downloads a payload. This could be malware, or a file download service that downloads other malware.

6. The final malware package is run on the victim's computer.

Independent of the type of attack being used, the main goal of the threat actor is to ensure the victim's web browser ends up on the threat actor's web page, which then serves out the malicious exploit to the victim.

Some malicious sites take advantage of vulnerable plugins or browser vulnerabilities to compromise the client's system. Larger networks rely on IDSs to scan downloaded files for malware. If detected, the IDSs issue alerts and records the event to log files for later analysis.

Server connection logs can often reveal information about the type of scan or attack. The different types of connection status codes are listed here:

Informational 1xx: A provisional response, consisting only of the Status-Line and optional headers. It is terminated by an empty line. There are no required headers for this class of status code. Servers MUST NOT send a 1xx response to an HTTP/1.0 client except under experimental conditions.

Successful 2xx: The client's request was successfully received, understood, and accepted.

Redirection 3xx: Further action must be taken by the user agent to fulfill the request. A client SHOULD detect infinite redirection loops, because these loops generate network traffic for each redirection.

Client Error 4xx: For cases in which the client seems to have erred. Except when responding to a HEAD request, the server SHOULD include an entity containing an explanation of the situation, and if it is temporary. User agents SHOULD display any included entity to the user.

Server Error 5xx: For cases where the server is aware that it has erred, or it cannot perform the request. Except when responding to a HEAD request, the server SHOULD include an entity containing an explanation of the error situation, and if it is temporary. User agents SHOULD display any included entity to the user.

To defend against web-based attacks, the following countermeasures should be used:

- Always update the OS and browsers with current patches and updates.
- Use a web proxy like Cisco Cloud Web Security or Cisco Web Security Appliance to block malicious sites.
- Use the best security practices from the Open Web Application Security Project (OWASP) when developing web applications.
- Educate end users by showing them how to avoid web-based attacks.

Malicious iFrames

Threat actors often make use of malicious inline frames (iFrames). An iFrame is an HTML element that allows the browser to load another web page from another source. iFrame attacks have become very common, as they are often used to insert advertisements from other sources into the page. In some instances, the iFrame page that is loaded consists of only a few pixels. This makes it very hard for the user to see. Because the iFrame is run in the page, it can be used to deliver a malicious exploit.

These are some of the ways to prevent or reduce malicious iFrames:

- Use a web proxy like Cisco Cloud Web Security or Cisco Web Security Appliance to block malicious sites.
- Because attackers often change the source of the iFrame in a compromised website, make sure web developers do not use iFrames to isolate any content from third parties from a website.
- Use a service such as Cisco OpenDNS to prevent users from navigating to websites that are known to be malicious.
- Make sure the end user understands what an IFrame is and that threat actors have been using this method often in web-based attacks.

HTTP 302 Cushioning

Another type of HTTP attack is the HTTP 302 cushioning attack. Threat actors use the 302 Found HTTP response status code to direct the user's web browser to the new location. Threat actors often use legitimate HTTP functions such as HTTP redirects to carry out their attacks. HTTP allows servers to redirect a client's HTTP request to a different server. HTTP redirection is used, for example, when web content has moved to a different URL or domain name. Thisallows old URLs and bookmarks to continue to function. Therefore, security analysts should understand how a function such as HTTP redirection works and how it can be used during attacks.

When the response from the server is a 302 Found status, it also provides the URL in the location field. The browser believes that the new location is the URL provided in the header. The browser is invited to request this new URL. This redirect function can be used multiple times until the browser finally lands on the page that contains the exploit. The redirects may be difficult to detect due to the fact that legitimate redirects frequently occur on the network.

These are some ways to prevent or reduce HTTP 302 cushioning attacks:

- Use a web proxy like Cisco Cloud Web Security or Cisco Web Security Appliance to block malicious sites.
- Use a service such as Cisco OpenDNS to prevent users from navigating to websites that are known to be malicious.
- Make sure the end user understands how the browser is redirected through a series of HTTP 302 redirections.

Domain Shadowing

When a threat actor wishes to create a domain shadowing attack, they must first compromise a domain. Then they must create multiple subdomains of that domain to be used for the attacks. Hijacked domain registration logins are then used to create the many subdomains

needed. After these subdomains have been created, attackers can use them as they wish even if they are found out to be malicious domains. They can simply make more from the parent domain. The following sequence is typically used by threat actors:

1. The website becomes compromised.
2. HTTP 302 cushioning is used.
3. Domain shadowing is used.
4. An exploit kit landing page is created.
5. Malware is spread through its payload.

These are some ways to prevent or reduce domain shadowing attacks:

- Secure all domain owner accounts. Use strong passwords and use two-factor authentication to secure these powerful accounts.
- Use a web proxy like Cisco Cloud Web Security or Cisco Web Security Appliance to block malicious sites.
- Use a service such as Cisco OpenDNS to prevent users from navigating to websites that are known to be malicious.
- Make sure that domain owners validate their registration accounts and look for any subdomains that they have not authorized.

Email (7.3.2.2)

Over the past 25 years, email has evolved from a tool used primarily by technical and research professionals to become the backbone of corporate communications. Each day, more than 100 billion corporate email messages are exchanged. As the level of use rises, security becomes a greater priority. The way that users access email today also increases the opportunity for the threat of malware to be introduced. It used to be that corporate users accessed text-based email from a corporate server. The corporate server was on a workstation that was protected by the company's firewall. Today, HTML messages are accessed from many different devices that are often not protected by the company's firewall. HTML allows more attacks because of the amount of access that can sometimes bypass different security layers.

The following are examples of email threats:

Attachment-based attacks: Threat actors embed malicious content in business files such as an email from the IT department. Legitimate users open malicious content. Malware is used in broad attacks often targeting a specific business vertical to seem legitimate, enticing users working in that vertical to open attachments, or click embedded links.

Email spoofing: Threat actors create email messages with a forged sender address that is meant to fool the recipient into providing money or sensitive information. For example, a bank sends you an email asking you to update your credentials. When this email displays the

identical bank logo as mail you have previously opened that was legitimate, it has a higher chance of being opened, having attachments opened and links clicked. The spoofed email may even ask you to verify your credentials so that the bank is assured that you are you, exposing your login information.

Spam email: Threat actors send unsolicited email containing advertisements or malicious files. This type of email is sent most often tosolicit a response, telling the threat actor that the email is valid and a user has opened the spam.

Open mail relay server: Threat actors take advantage of enterprise servers that are misconfigured as open mail relays to send large volumes of spam or malware to unsuspecting users. The open mail relay is an SMTP server that allows anybody on the Internet to send mail. Because anyone can use the server, they are vulnerable to spammers and worms. Very large volumes of spam can be sent by using an open mail relay. It is important that corporate email servers are never set up as an open relay. This will considerably reduce the amount of unsolicited emails.

Homoglyphs: Threat actors can use text characters that are very similar or even identical to legitimate text characters. These can be used in phishing emails to make them look very convincing. In DNS, these characters are very different from the real thing. When the DNS record is searched, a completely different URL is found when the link with the homoglyph is used in the search.

Just like any other service that is listening to a port for incoming connections, SMTP servers also may have vulnerabilities. Always keep SMTP software up to date with security and software patches and updates. To further prevent threat actors from completing their task of fooling the end user, implement countermeasures. Use a security appliance specific to email such as the Cisco Email Security Appliance. This will help to detect and block many known types of threats such as phishing, spam, and malware. Also, educate the end user. When attacks make it by the security measures in place, and they will sometimes, the end user is the last line of defense. Teach them how to recognize spam, phishing attempts, suspicious links and URLs, and homoglyphs, and teach them to never open suspicious attachments.

Web-Exposed Databases (7.3.2.3)

Web applications commonly connect to a relational database to access data. Because relational databases often contain sensitive data, databases are a frequent target for attacks.

Command Injection

Attackers are able to execute commands on a web server's OS through a webapplication that is vulnerable. This might occur if the web application provides input fields to the attacker for entering malicious data. The attacker's commands that are executed through the web

application have the same permissions as the web application. This type of attack is used because often there is insufficient validation of input. SQL injection and XSS are two different types of command injection.

SQL Injection

SQL is the language used to query a relational database. Threat actors use SQL injections to breach the relational database, create malicious SQL queries, and obtain sensitive data from the relational database.

One of the most common database attacks is the SQL injection attack. The SQL injection attack consists of inserting a SQL query via the input data from the client to the application. A successful SQL injection exploit can read sensitive data from the database, modify database data, execute administration operations on the database, and, sometimes, issue commands to the operating system.

Unless an application uses strict input data validation, it will be vulnerable to the SQL injection attack. If an application accepts and processes user-supplied data without any input data validation, a threat actor could submit a maliciously crafted input string to trigger the SQL injection attack.

Security analysts should be able to recognize suspicious SQL queries in order to detect if the relational database has been subjected to SQL injection attacks. They need to be able to determine which user ID was used by the threat actor to log in, then identify any information or further access the threat actor could have leveraged after a successful login.

Cross-Site Scripting

Not all attacks are initiated from the server side. Cross-site scripting (XSS) is where web pages that are executed on the client side, within their own web browser, are injected with malicious scripts. These scripts can be used by Visual Basic, JavaScript, and others to access a computer, collect sensitive information, or deploy more attacks and spread malware. As with SQL injection, this is often due to the attacker posting content to a trusted website with a lack of input validation. Future visitors to the trusted website will be exposed to the content provided by the attacker.

These are the two main types of XSS:

Stored (persistent): This is permanently stored on the infected server and is received by all visitors to the infected page.

Reflected (non-persistent): This only requires that the malicious script is located in a link and visitors must click the infected link to become infected.

These are some ways to prevent or reduce command injection attacks:

- Use the items listed in the OWASP XSS prevention cheat sheet for web application developers.
- Use an IPS implementation to detect and prevent malicious scripts.
- Use a web proxy like Cisco Cloud Web Security or Cisco Web Security Appliance to block malicious sites.
- Use a service such as Cisco OpenDNS to prevent users from navigating to websites that are known to be malicious.
- As with all other security measures, be sure to educate end users. Teach them to identify phishing attacks and notify infosec personnel when they are suspicious of anything security-related.

Lab 7.3.2.4: Attacking a MySQL Database

In this lab, you will view a PCAP file from a previous attack against a SQL database.

Lab 7.3.2.5: Reading Server Logs

In this lab, you will complete the following objectives:

- Reading Log Files with Cat, More, and Less
- Log Files and Syslog
- Log Files and Journalctl

Summary (7.4)

In this chapter, you learned the importance of network monitoring and the tools used by cybersecurity analysts. These tools include port mirroring, protocol analyzers, and SIEMs.

You also learned about the inherent vulnerabilities in network protocols and services.

IP is vulnerable to a variety of attacks, including:

- ICMP attacks
- DoS attacks
- DDoS attacks
- Address spoofing attacks
- Man-in-the-middle attack (MITM)
- Session hijacking

TCP is also vulnerable to TCP SYN flood attacks, TCP reset attacks, and TCP session hijacking attacks. UDP is vulnerable to checksum modification attacks and UDP flood attacks.

IP services have a several of vulnerabilities, including:

- ARP cache poisoning
- DNS attacks, including poisoning, amplification and reflection, resource utilization, and stealth attacks
- DNS tunneling for botnets and other malicious activity
- DHCP spoofing and starvation attacks
- Web attacks through unsecure HTTP, iFrames, and HTTP 302 cushioning
- SQL injection attacks
- Cross-site scripting attacks

Practice

The following activities provide practice with the topics introduced in this chapter. The Labs and Class Activities are available in the companion CCNA Cybersecurity Operations Lab Manual (ISBN: 9781587134388). The PacketTracer Activity instructions are also in the Labs & Study Guide. The PKA files are found in the online course.

Class Activities

Class Activity 7.0.1.2: What's Going On?

Labs

Lab 7.2.1.11: Observing a DDoS Attack
Lab 7.2.2.4: Observing TCP Anomalies
Lab 7.3.1.6: Exploring DNS Traffic
Lab 7.3.2.4: Attacking a MySQL Database
Lab 7.3.2.5: Reading Server Logs

Packet Tracer Activities

Packet Tracer 7.1.2.7: Logging Network Activity