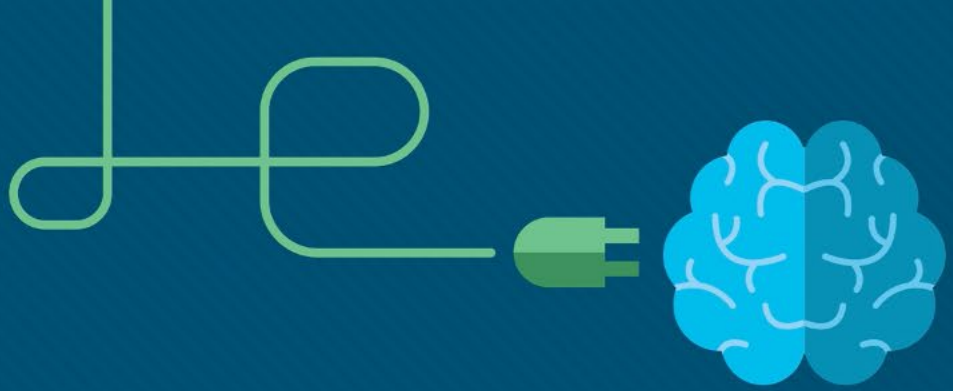


Module 4: Attacking What We Do

Cybersecurity Essentials 3.0

Module Objectives

Module Title: Attacking What We Do
Module Objective: Recommend measures to mitigate threats.

Topic Title	Topic Objective
IP Services	Explain IP service vulnerabilities.
Enterprise Services	Explain how network application vulnerabilities enable network attacks.
Mitigating Common Network Attacks	Recommend basic threat mitigation measures.

4.1 IP Services

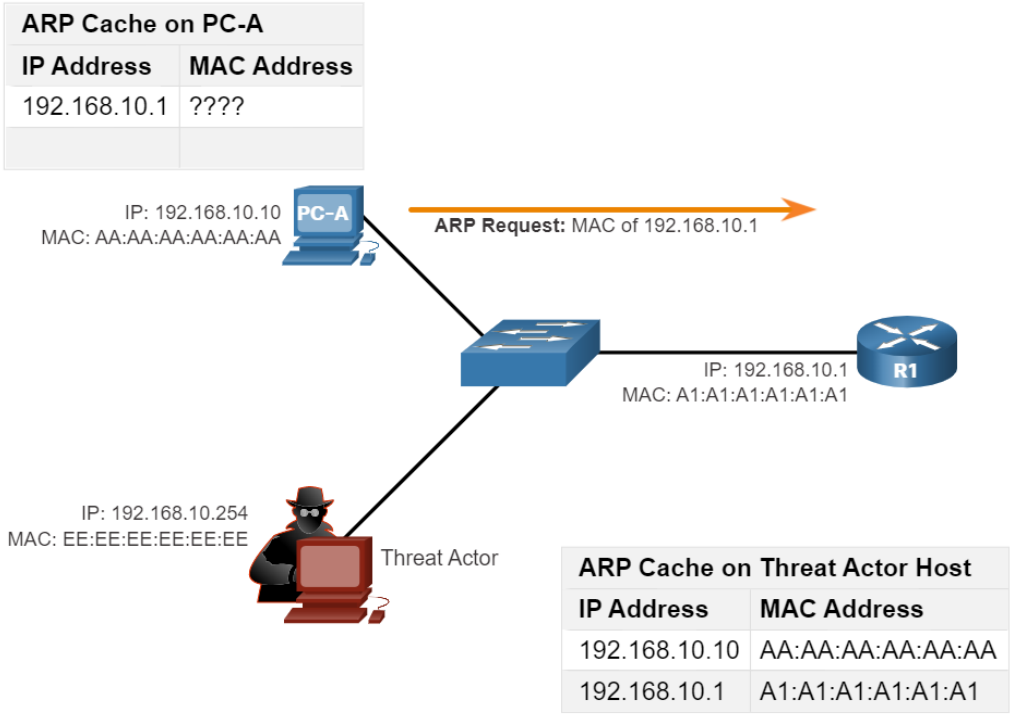
ARP Vulnerabilities

Hosts broadcast an ARP Request to other hosts on the network segment to determine the MAC address of a host with a particular IP address. The host with the matching IP address in the ARP Request sends an ARP Reply.

Any client can send an unsolicited ARP Reply called a “gratuitous ARP.”

- 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.
- Any host can claim to be the owner of any IP/MAC they choose.
- Threat actors can poison the ARP cache of devices on the local network, creating an MiTM attack to redirect traffic.
- The goal is to associate the threat actor’s MAC address with the IP address of the default gateway in the ARP caches of hosts on the LAN segment.
- It positions the threat actor between the victim and all other systems outside the local subnet.

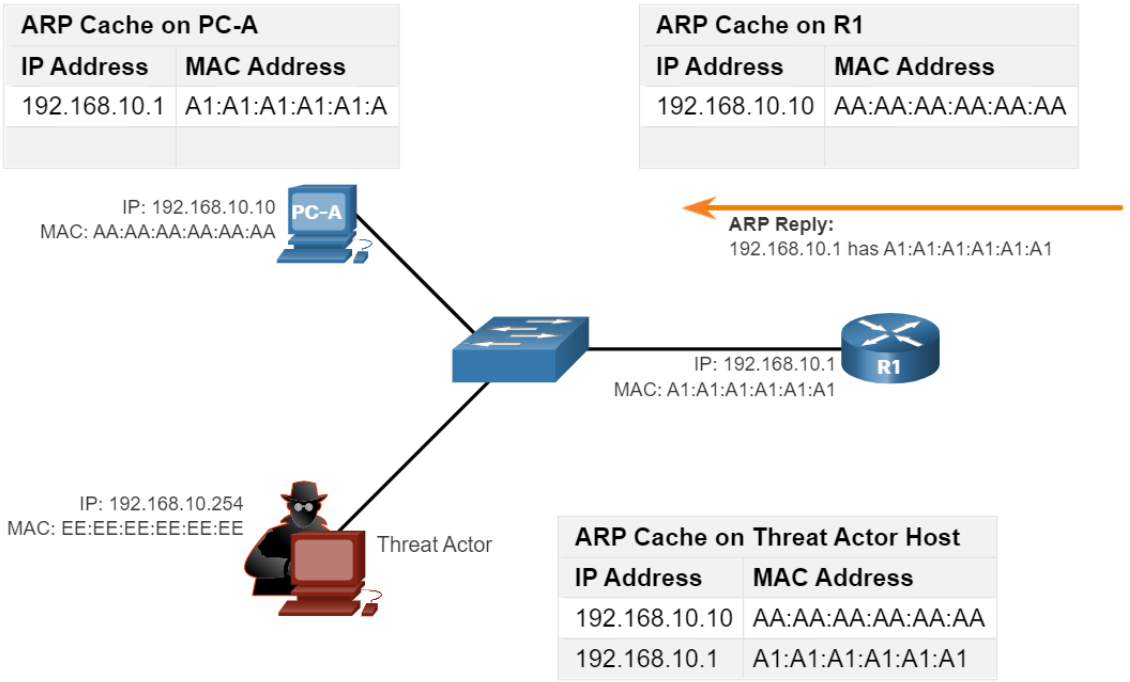
ARP Cache Poisoning



ARP Request

- The figure shows how ARP cache poisoning works.
- PC-A requires the MAC address of its default gateway (R1); therefore, it sends an ARP Request for the MAC address of 192.168.10.1.

ARP Cache Poisoning (Cont.)



ARP Reply

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

Spoofed Gratuitous ARP Replies

- Passive ARP poisoning occurs when threat actors steal confidential information.
- Active ARP poisoning occurs when threat actors modify data in transit or inject malicious data.

DNS Attacks

A **DNS open resolver** answers queries from clients outside of its administrative domain and is vulnerable to multiple malicious activities described in the table.

DNS Resolver Vulnerabilities	Description
DNS cache poisoning attacks	Threat actors send spoofed, falsified record resource (RR) information to a DNS resolver redirecting users from legitimate sites to malicious sites. DNS cache poisoning attacks can 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 DoS or DDoS attacks on DNS open resolvers to increase the volume of attacks and to hide the true source of an attack. Threat actors send DNS messages to the open resolvers using the IP address of a target host. These attacks are possible because the open resolver will respond to queries from anyone asking a question.
DNS resource utilization attacks	DoS attacks consume all the available resources to negatively affect the operations of the DNS open resolver. The impact of a DoS attack may require the DNS open resolver to be rebooted or services to be stopped and restarted.

DNS Attacks (Cont.)

Threat actors also use the **DNS stealth** techniques described in the table to carry out their attacks.

DNS Stealth Techniques	Description
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 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 (C&C) servers.

DNS Domain shadowing involves the threat actor gathering domain account credentials to silently create multiple sub-domains to be used during the attacks that typically point to malicious servers without alerting the actual owner of the parent domain.

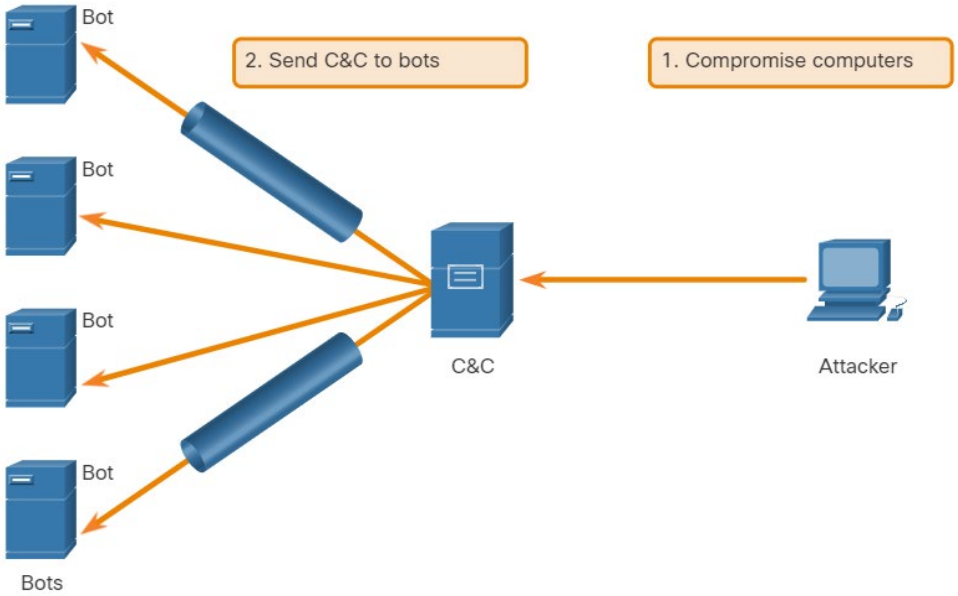
DNS Tunneling

Botnets use DNS protocol to spread malware or launch DDoS and phishing attacks. Security analysts must implement a solution to block outbound communications from the infected hosts to detect when an attacker is using DNS tunneling.

A DNS tunneling attack using TXT works like this:

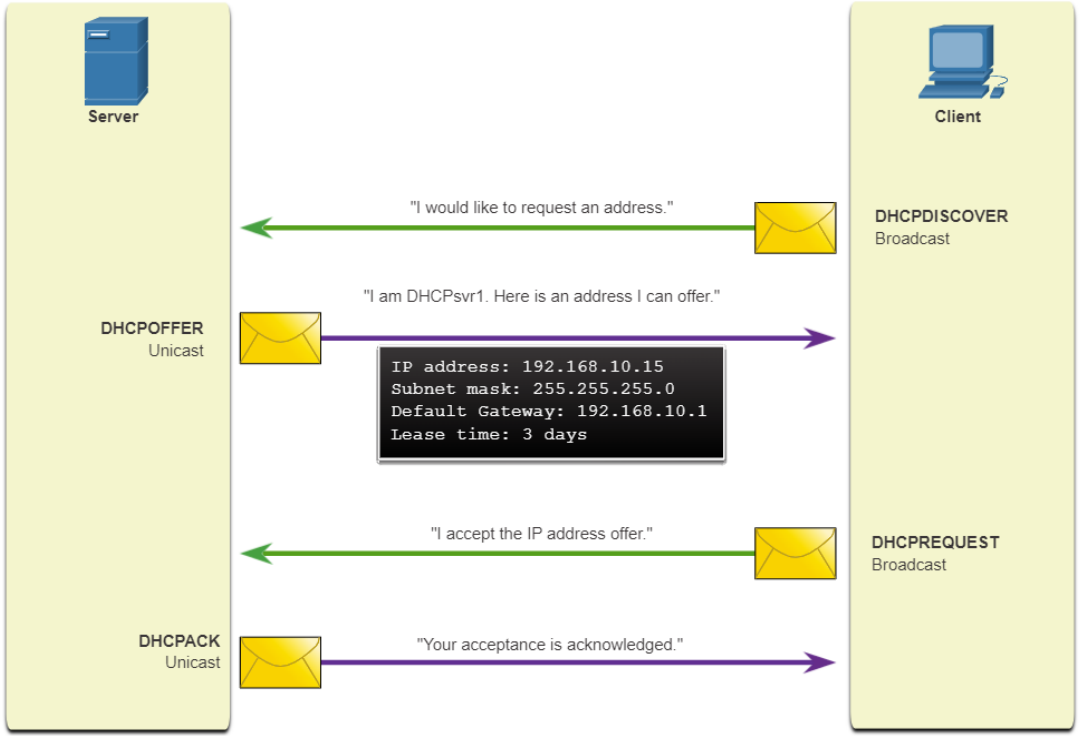
- The data is split into multiple encoded chunks.
- Each chunk is placed into a lower-level domain name label of the DNS query.
- Because there is no response from the local or networked DNS for the query, the request is sent to the ISP's recursive DNS servers.
- The recursive DNS service will forward the query to the attacker's authoritative name server.
- The process is repeated until all the queries containing the chunks are sent.
- When the attacker's authoritative name server receives the DNS queries from the infected devices, it sends responses for each DNS query, which contains the encapsulated, encoded commands.
- The malware on the compromised host recombines the chunks and executes the commands hidden within.

DNS Tunneling (Cont.)



- A filter that inspects DNS traffic must be used to stop DNS tunneling.
- Pay particular attention to DNS queries that are longer than average, or those that have a suspicious domain name.
- DNS security solutions, such as Cisco Umbrella, block much of the DNS tunneling traffic by identifying suspicious domains.
- Domains associated with Dynamic DNS services should be considered highly suspect.

DHCP



- DHCP servers dynamically provide IP configuration information to clients.
- The figure shows the typical sequence of a DHCP message exchange between client and server.

DHCP Attacks

Occur 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** - Threat actor provides an invalid gateway, or the 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** - Threat actor provides an incorrect DNS server address pointing the user to a malicious website.
- **Wrong IP address** - Threat actor provides an invalid IP address, invalid default gateway IP address, or both. The threat actor then creates a DoS attack on the DHCP client.

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.

DHCP Attacks (Cont.)

The steps in a DHCP spoofing attack are:

Steps	Description
1. Client Broadcasts DHCP Discovery Messages	A legitimate client connects to the network and requires IP configuration parameters. The client broadcasts a DHCP Discover request looking for a response from a DHCP server. A legitimate DHCP Server and a rogue server receive the message.
2. DHCP Servers Respond with Offers	The legitimate and rogue DHCP servers each respond with valid IP configuration parameters. The client replies to the first offer received.
3. Client Accepts Rogue DHCP Request	The client received the rogue offer first. It broadcasts a DHCP request accepting the parameters from the rogue server. The legitimate and rogue server each receive the request.
4. Rogue DHCP Acknowledges the Request	However, only the rogue server unicasts a reply to the client to acknowledge its request. The legitimate server stops communicating with the client because the request has already been acknowledged.

Lab - Exploring DNS Traffic

In this lab, you will complete the following objectives:

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

4.2 Enterprise Services

HTTP and HTTPS

The common stages of a typical web attack are:

- The victim unknowingly visits a web page that has been compromised by malware.
- The compromised web page redirects the user, often through many compromised servers, to a site containing malicious code.
- When the user visits this site with malicious code an exploit kit scans the software running on the victim's computer including the OS, Java, or Flash player looking for an exploit in the software.
- 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's computer.
- After the victim's computer has been compromised, it connects to the malware server and downloads a payload (malware, or a file download service that downloads other malware).
- The final malware package is run on the victim's computer.

HTTP and HTTPS (Cont.)

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** - This is 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** - This is 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** - This is for cases where the server is aware that it has erred or 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.

Common HTTP Exploits

Malicious iFrames

- Malicious iFrames are often used by threat actors.
- It is an HTML element that allows the browser to load another web page from another source.
- Threat actors compromise a webserver and modify web pages by adding HTML for the malicious iFrame. It can then be used to deliver a malicious exploit, such as spam advertising, an exploit kit, and other malware.

HTTP 302 Cushioning

- Threat actors attack by using the 302 Found HTTP response status code to redirect the user's web browser to a new location.
- The browser believes that the new location is the URL provided in the header and 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 since legitimate redirects frequently occur on the network.

Common HTTP Exploits (Cont.)

Domain Shadowing

- To attack, the threat actor must first compromise a domain, then create multiple subdomains of that domain.
- Hijacked domain registration logins are then used to create the many subdomains needed for attacks.
- Even if the subdomains are found out to be malicious domains, more can be made from the parent domain.
- The following sequence is typically used by threat actors:
    1. A website becomes compromised.
    2. HTTP 302 cushioning is used to send the browser to malicious websites.
    3. Domain shadowing is used to direct the browser to a compromised server.
    4. An exploit kit landing page is accessed.
    5. Malware downloads from the exploit kit landing page.

Email

As the level of email use rises, security becomes a greater priority. 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.
- **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.
- **Spam email** - Threat actors send unsolicited email containing advertisements or malicious files.
- **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.
- **Homoglyphs** - Threat actors can use text characters that are very similar or even identical to legitimate text characters.

Email (Cont.)

Just like any other service that is listening to a port for incoming connections, SMTP servers may also have vulnerabilities.

A few actions to help protect against threats include:

- Keep SMTP software up to date with security and software patches and updates.
- Implement countermeasures to further prevent threat actors from completing their task of fooling the end user.
- Use a security appliance specific to email such as the Cisco Email Security Appliance to help detect and block many known types of threats such as phishing, spam, and malware.
- Teach the end user how to recognize spam, phishing attempts, suspicious links and URLs, homoglyphs, and to never open suspicious attachments.

Web-Exposed Databases

Web applications commonly connect to a relational database to access sensitive data.

Code Injection

- Attackers execute commands on a web server's OS through a vulnerable web application.

SQL Injection

- Threat actors use SQL injections to breach the relational database, create malicious SQL queries, and obtain sensitive data from the relational database.
- SQL Injection is one of the most common database attacks.

Security analysts should be able to recognize suspicious SQL queries to detect if the relational database has been subjected to SQL injection attacks.

Security analysts 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.

Client-Side Scripting

Cross-Site Scripting (XSS)

XSS occurs when web pages that are executed on the client-side, within their own web browser, are injected with malicious scripts.

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 in a link and visitors must click the infected link to become infected.

To prevent or reduce XSS attacks:

- Be sure that web application developers are aware of XSS vulnerabilities and how to avoid them.
- Use an IPS implementation to detect and prevent malicious scripts.
- Use a web proxy to block malicious sites.
- Use a service such as Cisco Umbrella to prevent users from navigating to websites that are known to be malicious.
- Teach end users to identify phishing attacks and notify infosec personnel when they are suspicious of anything security-related.

Lab - Install a Virtual Machine on a Personal Computer

Objectives:

Part 1: Prepare a Computer for Virtualization
Part 2: Import a Virtual Machine into VirtualBox Inventory

Lab - Attacking a MySQL Database

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

- Part 1: Open Wireshark and load the PCAP file.
- Part 2: View the SQL Injection Attack.
- Part 3: The SQL Injection Attack continues…
- Part 4: The SQL Injection Attack provides system information.
- Part 5: The SQL Injection Attack and Table Information.
- Part 6: The SQL Injection Attack Concludes.

Lab - 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**

4.3 Mitigating Common Network Attacks

Defending the Network

Constant vigilance and ongoing education are required to defend your network against attack.

The best practices for securing a network are:

- Develop a written security policy for the company.
- Educate employees about the risks of social engineering, and develop strategies to validate identities over the phone, via email, or in person.
- Control physical access to systems.
- Use strong passwords and change them often.
- Encrypt and password-protect sensitive data.
- Implement security hardware and software such as firewalls, IPS devices, VPN devices, antivirus software, and content filtering.
- Perform backups and test the backed-up files on a regular basis.
- Shut down unnecessary services and ports.
- Keep patches up-to-date by installing them weekly or daily, if possible, to prevent buffer overflow and privilege escalation attacks.
- Perform security audits to test the network.

Mitigating Malware

Malware, including viruses, worms, and Trojan horses, can cause serious problems on networks and end devices.

Network administrators have several means (or countermeasures) of mitigating these attacks:

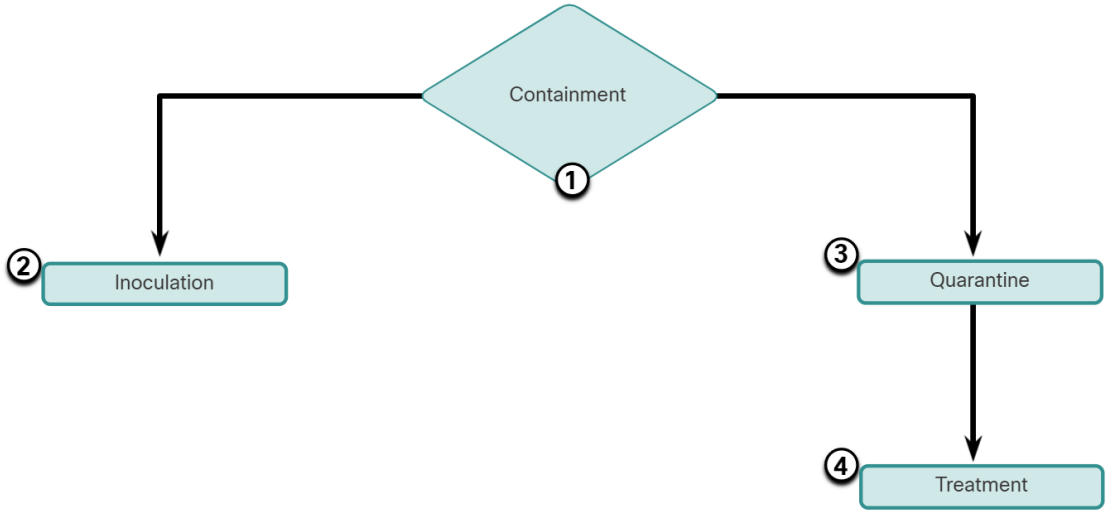
Antivirus software

- It helps prevent hosts from getting infected and spreading malicious code by detecting and eliminating viruses.
- They do not prevent viruses from entering the network, so a network security professional must be aware of the major viruses and keep track of security updates.

Prevent malware files from entering the network

- Security devices at the network perimeter can identify and remove known malware files based on their indicators of compromise.

Mitigating Worms



- Worms are more network-based than viruses.
- Worm mitigation requires diligence and coordination on the part of network security professionals.
- As shown in the figure, the response to a worm attack can be broken down into four phases: containment, inoculation, quarantine, and treatment.

Mitigating Worms (Cont.)

Phase	Response
1. Containment	The containment phase involves limiting the spread of a worm infection to areas of the network that are already affected. This requires compartmentalization and segmentation of the network to slow down or stop the worm and to prevent currently infected hosts from targeting and infecting other systems. Containment requires using both outgoing and incoming ACLs on routers and firewalls at control points within the network.
2. Inoculation	The inoculation phase runs parallel to, or after, the containment phase. During the inoculation phase, all uninfected systems are patched with the appropriate vendor patch. The inoculation process further deprives the worm of any available targets.
3. Quarantine	The quarantine phase involves tracking down and identifying infected machines within the contained areas and disconnecting, blocking, or removing them. This isolates these systems appropriately for the treatment phase.
4. Treatment	The treatment phase involves actively disinfecting infected systems. This can involve terminating the worm process, removing modified files or system settings that the worm introduced, and patching the vulnerability the worm used to exploit the system. Alternatively, in more severe cases, the system may need to be reinstalled to ensure that the worm and its by-products are removed.

Mitigating Reconnaissance Attacks

- Reconnaissance attacks are typically the precursor to other attacks that have the intent of gaining unauthorized access to a network or disrupting network functionality.
- A variety of technologies and devices can be used to monitor this type of activity and generate an alarm.
- Cisco's ASA provides intrusion prevention in a standalone device. Additionally, enterprise routers, such as Cisco Integrated Services Routers (ISR), support network-based intrusion prevention with additional software.
- Reconnaissance attacks can be mitigated in several ways, including the following:
  - Implementing authentication to ensure proper access.
  - Using encryption to render packet sniffer attacks useless.
  - Using anti-sniffer tools to detect packet sniffer attacks.
  - Implementing a switched infrastructure.
  - Using a firewall and IPS.

Mitigating Reconnaissance Attacks (Cont.)

- Anti-sniffer software and hardware tools detect changes in the response time of hosts to determine whether the hosts are processing more traffic than their own traffic loads would indicate.
- While this does not eliminate the threat, as part of an overall mitigation system, it can reduce the number of instances of threat.
- Encryption is also effective for mitigating packet sniffer attacks. If traffic is encrypted, using a packet sniffer is of little use because captured data is not readable.
- It is impossible to mitigate port scanning but using an IPS and firewall can limit the information that can be discovered with a port scanner.
- Ping sweeps can be stopped if ICMP echo and echo-reply are turned off on edge routers; however, when these services are turned off, network diagnostic data is lost.
- Additionally, port scans can be run without full ping sweeps. The scans simply take longer because inactive IP addresses are also scanned.

Mitigating Access Attacks

- Techniques for mitigating access attacks: strong password security, principle of minimum trust, cryptography, and applying operating system and application patches.

- A surprising number of access attacks are carried out through simple password guessing or brute-force dictionary attacks against passwords. To defend against this, create and enforce a strong authentication policy.

    - **Use strong passwords** - They are at least eight characters and contain uppercase letters, lowercase letters, numbers, and special characters.
    - **Disable accounts after a specified number of unsuccessful logins has occurred** - This helps to prevent continuous password attempts.

- The network should also be designed using the principle of minimum trust. This means that systems should not use one another unnecessarily.

Mitigating Access Attacks (Cont.)

- Cryptography is a critical component of any modern secure network. The more that traffic is encrypted, the fewer opportunities hackers have for intercepting data with man-in-the-middle attacks.
- The use of encrypted or hashed authentication protocols, along with a strong password policy, greatly reduces the probability of successful access attacks.
- Educate employees about the risks of social engineering, and develop strategies to validate identities over the phone, via email, or in person.
- Multifactor authentication (MFA) has become increasingly common and requires two or more independent means of verification.
- Access attacks can be detected by reviewing logs, bandwidth utilization, and process loads.

Mitigating DoS Attacks

- To minimize the number of DoS attacks, a network utilization software package should be always running and required by the organization's network security policy.
- DoS attacks can lead to problems in the network segments of the computers being attacked.
- Historically, DoS attacks were sourced from spoofed addresses.
- Cisco routers and switches support several anti-spoofing technologies, such as port security, DHCP snooping, IP Source Guard, Dynamic Address Resolution Protocol (DAI) Inspection, and access control lists (ACLs).

Lab - Recommend Threat Mitigation Measures

In this lab, you will complete the following objectives:

- Part 1: Review an Incident at a Video Production Company
- Part 2: Review an Incident at a Retail Company

4.4 Attacking What We Do Summary

What Did I Learn in this Module?

- The DNS protocol defines an automated service that matches resource names with the required numeric IP host address A Gratuitous ARP is an unsolicited ARP Reply that can be sent by any client.
- A DHCP spoofing attack occurs when a rogue DHCP server is connected to the network and provides false IP configuration parameters to legitimate clients.
- Code injection and SQL injection attacks exploit insufficiently validated input fields to send commands to databases or other applications to gain access to private information.
- Cross-Site Scripting (XSS) attacks occur when browsers execute malicious scripts on the client and provide threat actors with access to sensitive information on the local host.
- Antivirus software is the primary means of mitigating virus and Trojan horse attacks.
- Worm attack response can be broken down into four phases: containment, inoculation, quarantine, and treatment.
- Reconnaissance attack mitigation can include access authentication, encryption, and the use of anti-sniffer tools, firewalls, and IPS.
- Network security best practices can include employee education, the use of strong passwords and encryptions, backup file testing, unnecessary service and port shutdowns, updating patches, and performing security audits.