Free Questions for PT0-003

Shared by Moore on 09-08-2024

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: DragDrop

During a penetration test, you gain access to a system with a limited user interface. This machine appears to have access to an isolated network that you would like to port scan.

INSTRUCTIONS

Analyze the code segments to determine which sections are needed to complete a port scanning script.

Drag the appropriate elements into the correct locations to complete the script.

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

Drag and Drop Options

```
self.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()
}
```

```
exec_scan(sys.argv[1], $PORTS)
```

```
port_scan(sys.argv[1], ports)
```

```
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()
```

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

```
#!/usr/bin/python
```

```
ports - [21,22]
```

```
#!/usr/bin/ruby
```

```
run_scan(sys.argv[1],ports)
```

```
#!/usr/bin/bash
```

```
export $PORTS = 21,22
```

```
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()
```

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 adderss. Exiting...')
        exit(1)
    else:
[ ? ]
```


P2P exams

Answer:

See the Answer in the Premium Version!

Question 2

Question Type: MultipleChoice

A penetration tester is conducting reconnaissance on a target network. The tester runs the following Nmap command: nmap -sv -sT -p - 192.168.1.0/24. Which of the following describes the most likely purpose of this scan?

Options:

A- OS fingerprinting
B- Attack path mapping
C- Service discovery
D- User enumeration

Answer:

C

Explanation:

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

Command Breakdown:

nmap: The network scanning tool.

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

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

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

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

Purpose of the Scan:

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

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

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

Question 3

Question Type: MultipleChoice

Which of the following post-exploitation activities allows a penetration tester to maintain persistent access in a compromised system?

Options:

A- Creating registry keys
B- Installing a bind shell
C- Executing a process injection
D- Setting up a reverse SSH connection

Answer:

A

Explanation:

Maintaining persistent access in a compromised system is a crucial goal for a penetration tester after achieving initial access. Here's an explanation of each option and why creating registry keys is the preferred method:

Creating registry keys (Answer: A):

Advantages: This method is stealthy and can be effective in maintaining access over long periods, especially on Windows systems.

Example: Adding a new entry to the HKLM\Software\Microsoft\Windows\CurrentVersion\Run registry key to execute a malicious script upon system boot.

Drawbacks: This method is less stealthy and can be easily detected by network monitoring tools. It also requires an open port, which might be closed or filtered by firewalls.

Executing a process injection (Option C):

Drawbacks: While effective for evading detection, it doesn't inherently provide persistence. The injected code will typically be lost when the process terminates or the system reboots.

Setting up a reverse SSH connection (Option D):

Drawbacks: This method can be useful for maintaining a session but is less reliable for long-term persistence. It can be disrupted by network changes or monitoring tools.

Conclusion: Creating registry keys is the most effective method for maintaining persistent access in a compromised system, particularly in Windows environments, due to its stealthiness and reliability.

Installing a bind shell (Option B):

Question 4

Question Type: MultipleChoice

SIMULATION

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

-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

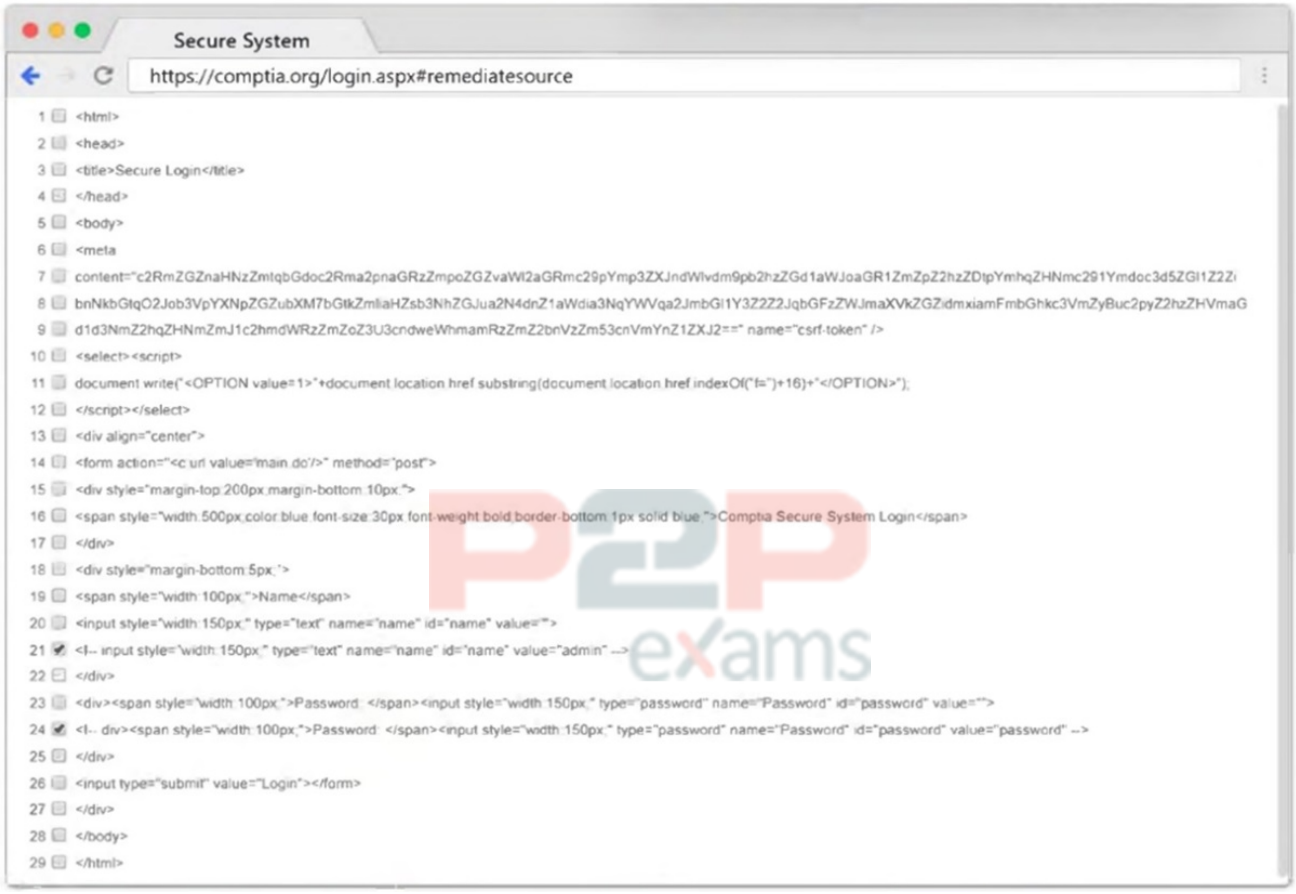
#!/usr/bin/ruby

#!/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 5

Question Type: MultipleChoice

A penetration tester needs to evaluate the order in which the next systems will be selected for testing. Given the following output:

Hostname	IP address	CVSS 2.0	EPSS
hrdatabase	192.168.20.55	9.9	0.50
financesite	192.168.15.99	8.0	0.01
legaldatabase	192.168.10.2	8.2	0.60
fileserver	192.168.125.7	7.6	0.90

Which of the following targets should the tester select next?

Options:

A- fileserver
B- hrdatabase
C- legaldatabase
D- financesite

Answer:

A

Explanation:

Evaluation Criteria:

CVSS (Common Vulnerability Scoring System): Indicates the severity of vulnerabilities, with higher scores representing more critical vulnerabilities.

EPSS (Exploit Prediction Scoring System): Estimates the likelihood of a vulnerability being exploited in the wild.

Analysis:

hrdatabase: CVSS = 9.9, EPSS = 0.50

financesite: CVSS = 8.0, EPSS = 0.01

legaldatabase: CVSS = 8.2, EPSS = 0.60

fileserver: CVSS = 7.6, EPSS = 0.90

Selection Justification:

fileserver has the highest EPSS score of 0.90, indicating a high likelihood of exploitation despite having a slightly lower CVSS score compared to other targets.

This makes it a critical target for immediate testing to mitigate potential exploitation risks.

Pentest Reference:

Risk Prioritization: Balancing between severity (CVSS) and exploitability (EPSS) is crucial for effective vulnerability management.

Risk Assessment: Evaluating both the impact and the likelihood of exploitation helps in making informed decisions about testing priorities.

By selecting the fileserver, the penetration tester focuses on a target that is highly likely to be exploited, addressing the most immediate risk based on the given scores.

Top of Form

Bottom of Form

Question 6

Question Type: MultipleChoice

A consultant starts a network penetration test. The consultant uses a laptop that is hardwired to the network to try to assess the network with the appropriate tools. Which of the following should the consultant engage first?

Options:

A- Service discovery
B- OS fingerprinting
C- Host discovery
D- DNS enumeration

Answer:

C

Explanation:

In network penetration testing, the initial steps involve gathering information to build an understanding of the network's structure, devices, and potential entry points. The process

generally follows a structured approach, starting from broad discovery methods to more specific identification techniques. Here's a comprehensive breakdown of the steps:

Host Discovery (Answer: C):

Objective: Identify live hosts on the network.

Tools & Techniques:

Ping Sweep: Using tools like nmap with the -sn option (ping scan) to check for live hosts by sending ICMP Echo requests.

ARP Scan: Useful in local networks, arp-scan can help identify all devices on the local subnet by broadcasting ARP requests.

nmap -sn 192.168.1.0/24

* Reference:

The GoBox HTB write-up emphasizes the importance of identifying hosts before moving to service enumeration.

The Forge HTB write-up also highlights using Nmap for initial host discovery in its enumeration phase.

* Service Discovery (Option A):

Objective: After identifying live hosts, determine the services running on them.

Tools & Techniques:

Nmap: Often used with options like -sV for version detection to identify services.

nmap -sV 192.168.1.100

* Reference:

As seen in multiple write-ups (e.g., Anubis HTB and Bolt HTB), service discovery follows host identification to understand the services available for potential exploitation.

* OS Fingerprinting (Option B):

Objective: Determine the operating system of the identified hosts.

Tools & Techniques:

Nmap: With the -O option for OS detection.

nmap -O 192.168.1.100

* Reference:

Accurate OS fingerprinting helps tailor subsequent attacks and is often performed after host and

service discovery, as highlighted in the write-ups.

* DNS Enumeration (Option D):

Objective: Identify DNS records and gather subdomains related to the target domain.

Tools & Techniques:

dnsenum, dnsrecon, and dig.

dnsenum example.com

DNS enumeration is crucial for identifying additional attack surfaces, such as subdomains and related services. This step is typically part of the reconnaissance phase but follows host discovery and sometimes service identification.

Conclusion: The initial engagement in a network penetration test is to identify the live hosts on the network (Host Discovery). This foundational step allows the penetration tester to map out active devices before delving into more specific enumeration tasks like service discovery, OS fingerprinting, and DNS enumeration. This structured approach ensures that the tester maximizes their understanding of the network environment efficiently and systematically.