

NAME: kelvin kimotho Waruingi

LinkedIn: [kelvin-kimotho](https://www.linkedin.com/in/kelvin-kimotho)

Network Enumeration with Nmap Module on HackTheBox



Here is my shareable link <https://academy.hackthebox.com/achievement/1476251/19>

Introduction to Nmap

Nmap (Network Mapper) is an open-source network scanning tool used for network discovery and security auditing.

We use it to identify devices and services on a network by sending packets and analyzing responses.

What Nmap Does

- **Host Discovery:** Identifies live systems or devices on a network.
- **Port Scanning:** Detects open ports on a system to assess which services are available.
- **Service Enumeration:** Identifies running services and their versions on discovered ports.
- **Operating System Detection:** Determines the operating system of a host based on network responses.
- **Network Mapping:** Creates a map of the network by discovering hosts, services, and vulnerabilities.
- **Vulnerability Scanning:** Identifies potential security weaknesses or misconfigurations in systems.
- **Firewall and IDS/IPS Evasion:** Bypasses firewalls and intrusion detection/prevention systems using specific scanning techniques.

- **Scripting:** Utilizes Nmap's scripting engine (NSE) for custom tasks like vulnerability scanning and network analysis.

Nmap Syntax. The basic command structure is “ **nmap <scan types> <options> <target>** “.

Scan Techniques. There are various scanning options in Nmap to identify open ports and services:

- **-sS:** TCP SYN Scan (default, fast scan)
- **-sT:** TCP Connect Scan
- **-sU:** UDP Scan
- **-sN/sF/sX:** TCP Null, FIN, Xmas scans
- **-sO:** IP protocol scan
- **-b:** FTP Bounce Scan (relay scan)

A SYN packet is sent without completing the handshake:

- If SYN-ACK is received → port is open.
- If RST is received → port is closed.
- No response → port is filtered (could be blocked by firewall).

Example Nmap Scan Command

- “ **sudo nmap -sS localhost** “. This command shows open ports (e.g., 22/tcp for SSH, 80/tcp for HTTP, 5432/tcp for PostgreSQL).

Enumeration

Enumeration is a critical phase in security assessments, focusing on identifying potential attack vectors rather than just gaining access.

Tools alone are not enough. effective enumeration requires understanding and interpreting the data they provide.

The goal of enumeration is to collect as much information as possible to identify vulnerabilities or attack vectors.

Types of Information we might be interested in include;

- Functions/resources that allow interaction or provide extra information.
- Information that leads to access or more critical details.

Many vulnerabilities stem from misconfigurations, ignorance, or improper security practices.

Manual vs. Automated Enumeration. While tools speed up the process, manual enumeration is crucial for bypassing security measures and gaining deeper insights.

Host Discovery in Nmap

Host discovery helps us determine which systems are online and accessible in a network, especially during penetration tests.

- The most effective method for host discovery is the **ICMP Echo Request** (ping).

Storing Results

- It's recommended to store all Nmap scan results for comparison, documentation, and reporting purposes. This helps track different tool outputs.

Scan Network Range

We can use the `-sn` option (disable port scanning) for host discovery over a network range (e.g., 10.129.2.0/24).

Example command is “**sudo nmap 10.129.2.0/24 -sn -oA tnet** “. This gives a list of active hosts in the network (IPs that respond to the ping).

Scan IP List

If provided with an IP list, we use the `-iL` option to scan those specific hosts. An example command is “**sudo nmap -sn -oA tnet -iL hosts.lst** “

Scan Multiple IPs

We can scan multiple IP addresses or specify a range using 10.129.2.18-20. An example command for range scan is “**sudo nmap -sn -oA tnet 10.129.2.18-20** “.

Scan Single IP

For scanning a single host to check if it's alive, we use **-sn** and disable port scanning. An example command for single IP is “**sudo nmap 10.129.2.18 -sn -oA host**”

Advanced Ping Techniques

Nmap uses **ARP pings** for local networks, which often detect a host before sending an ICMP request.

To ensure ICMP echo requests, we use **-PE** and **--packet-trace** to observe packet exchange. An example is “**sudo nmap 10.129.2.18 -sn -oA host -PE --packet-trace**”.

Disabling ARP Ping

To disable ARP pings and force ICMP echo requests, we use **--disable-arp-ping** option.

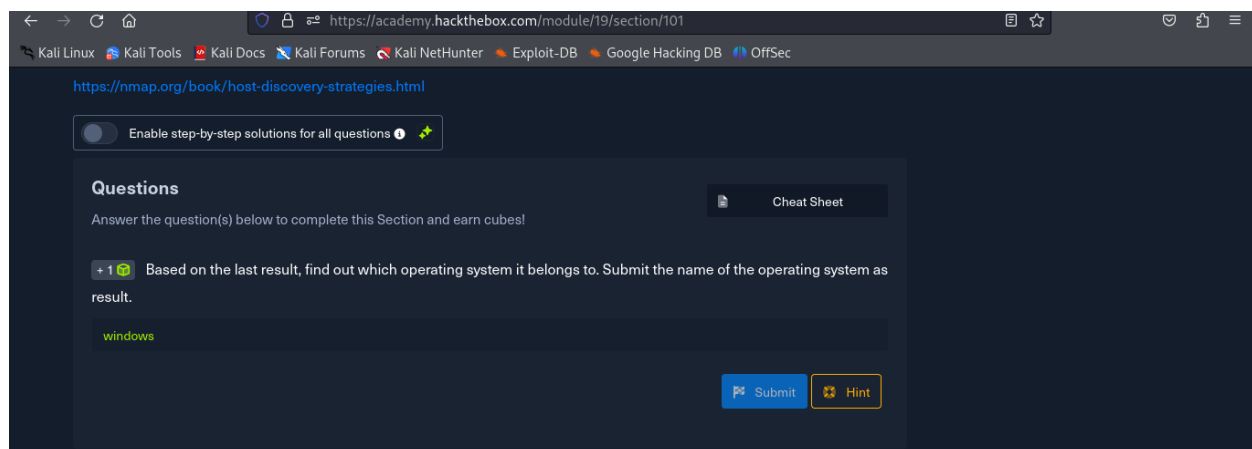
An example command is “**sudo nmap 10.129.2.18 -sn -oA host -PE --packet-trace --disable-arp-ping**”.

Reason for Results

We use the **--reason** option to explain why a host is marked as "alive" (e.g., received ARP or ICMP response). An example is “**sudo nmap 10.129.2.18 -sn -oA host -PE --reason**”.

Question: Based on the last result, find out which operating system it belongs to. Submit the name of the operating system as result.

Answer: windows



Host and Port Scanning

Nmap identifies several possible states for scanned ports

- **Open.** The port is accessible and ready to communicate (TCP connections, UDP datagrams, etc.).
- **Closed.** The target port exists but is not accepting connections (RST flag in TCP).
- **Filtered.** Nmap cannot determine if the port is open or closed, usually due to firewalls blocking or dropping packets.
- **Unfiltered.** Port is accessible but its state (open/closed) is unknown (TCP-ACK scan).
- **Open|Filtered.** Nmap cannot distinguish if the port is open or filtered, often due to firewall interference.
- **Closed|Filtered.** Typically seen with idle scans (IP ID idle scans), where the state can't be confirmed due to filtering.

Discovering Open TCP Ports

- **SYN Scan (-sS).** This is the default scan when run as root, sending a SYN packet and waiting for a response to infer port state. It's stealthy because it doesn't complete the TCP handshake.
- **TCP Connect Scan (-sT).** A less stealthy scan, as it completes the three-way handshake to determine if the port is open or closed. It can be detected more easily by intrusion detection/prevention systems (IDS/IPS).
- **Top Ports Scanning (--top-ports).** Scans the most frequently used ports based on Nmap's internal database (e.g., --top-ports=10 to scan the top 10).

Sample Nmap Commands

- “sudo nmap 10.129.2.28 --top-ports=10”. Scans the top 10 most frequent TCP ports.
- “sudo nmap 10.129.2.28 -p 21 --packet-trace -Pn -n --disable-arp-ping”. Tracks the packets during a scan of port 21 (FTP) with no ARP, DNS resolution, or ICMP Echo.
- **Connect Scan (-sT).** A more accurate but less stealthy method that completes the full TCP handshake (SYN → SYN-ACK → ACK).
- “sudo nmap 10.129.2.28 -p 443 --packet-trace --disable-arp-ping -Pn -n --reason -sT”. Performs a full TCP connection scan on port 443 (HTTPS) and traces the packets.

Filtered Ports

- These are ports that are either dropped or rejected by firewalls.
- **Dropping packets.** No response, indicating the port might be filtered.
- **Rejecting packets.** An ICMP "port unreachable" message is received.

UDP Port Scanning

- UDP scans (**-sU**) are slower than TCP scans because UDP is a connectionless protocol that doesn't provide feedback unless the service responds to the request.
- Ports that don't respond may be "**open|filtered**," indicating either that the port is open but not responding or that it is blocked by a firewall.
- A scan on port 137 (NetBIOS name service) for example might show as open, while port 100 (unknown service) could be closed if we receive an ICMP "port unreachable" response.

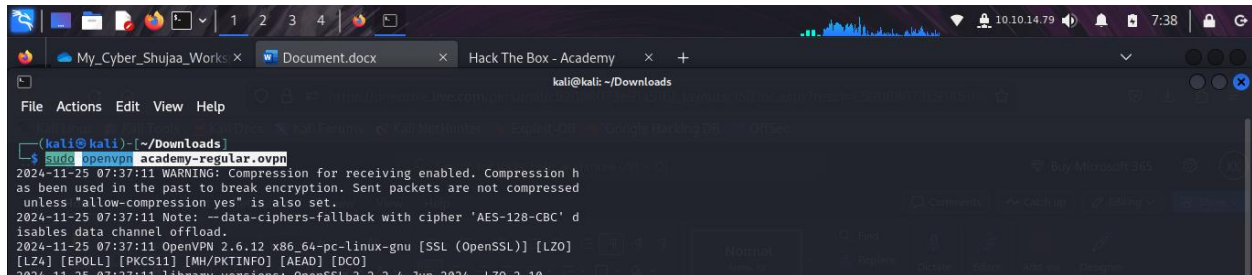
Scanning for Service Versions

- **Version Scan (-sV)** option. This can provide more details on the services running on open ports, including version information.

Further Analysis with --reason

The **--reason** option in Nmap provides an explanation of why a port is in a particular state (e.g., open, closed, filtered) based on the responses received.

At this point i had to perform A nmap scan on a target to answer the questions that followed. Since i prefer Using my machine instead of their provided attack boxes, I went ahead and downloaded a vpn configuration file in order to connect to their network and scan the target and answer the questions that followed. The comment to connect to their network via the vpn is “**sudo openvpn academy-regular.ovpn**”.



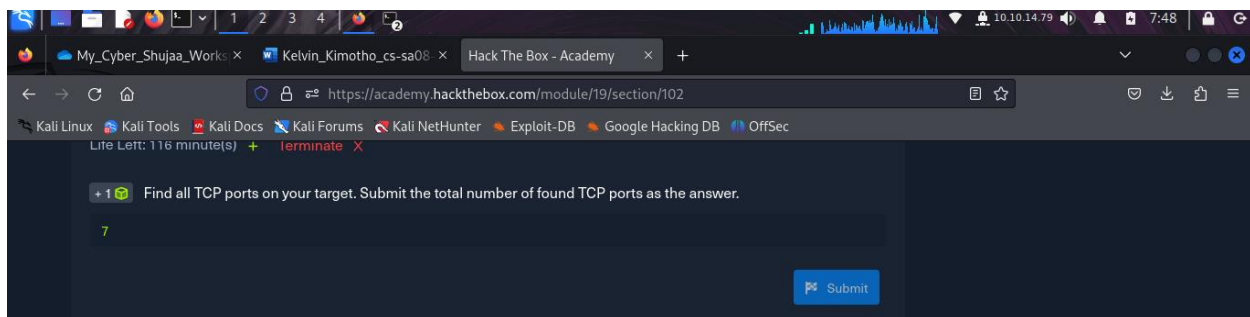
```
kali@kali: ~/Downloads
$ sudo openvpn academy-regular.ovpn
2024-11-25 07:37:11 WARNING: Compression for receiving enabled. Compression has been used in the past to break encryption. Sent packets are not compressed unless "allow-compression yes" is also set.
2024-11-25 07:37:11 Note: --data-ciphers-fallback with cipher 'AES-128-CBC' disables data channel offload.
2024-11-25 07:37:11 OpenVPN 2.6.12 x86_64-pc-linux-gnu [SSL (OpenSSL)] [LZO] [LZ4] [EPOLL] [PKCS11] [MH/PTINFO] [AEAD] [DCO]
2024-11-25 07:37:11 library versions: OpenSSL 3.2.2 4 Jun 2024, LZO 2.10
```

```
DDR=74:de:2b:10:8c:ac
2024-11-25 07:37:13 GD66: remote_host_ipv6=64:ff9b::9a39:a467
2024-11-25 07:37:13 net_route_v6_best_gw query: dst 64:ff9b::9a39:a467
2024-11-25 07:37:13 net_route_v6_best_gw result: via fe80::145e:74ff:fe1c:9201 dev wlan0
1 dev wlan0
2024-11-25 07:37:13 ROUTE6_GATEWAY fe80::145e:74ff:fe1c:9201 IFACE=wlan0
2024-11-25 07:37:13 TUN/TAP device tun0 opened
2024-11-25 07:37:13 net_iface_mtu_set: mtu 1500 for tun0
2024-11-25 07:37:13 net_iface_up: set tun0 up
2024-11-25 07:37:13 net_addr_v4_add: 10.10.14.79/23 dev tun0
2024-11-25 07:37:13 net_iface_mtu_set: mtu 1500 for tun0
2024-11-25 07:37:13 net_iface_up: set tun0 up
2024-11-25 07:37:13 net_addr_v6_add: dead:beef::104d/64 dev tun0
2024-11-25 07:37:13 net_route_v4_add: 10.10.10.0/23 via 10.10.14.1 dev [NULL]
table 0 metric -1
2024-11-25 07:37:13 net_route_v4_add: 10.129.0.0/16 via 10.10.14.1 dev [NULL]
table 0 metric -1
2024-11-25 07:37:13 add_route_ipv6(dead:beef::/64 → dead:beef::1 metric -1
) dev tun0
2024-11-25 07:37:13 net_route_v6_add: dead:beef::/64 via :: dev tun0 table 0
metric -1
2024-11-25 07:37:13 Initialization Sequence Completed
2024-11-25 07:37:13 Data Channel: cipher 'AES-256-CBC', auth 'SHA256', peer-i
d: 47, compression: 'lzo'
2024-11-25 07:37:13 Timers: ping 10, ping-restart 120
2024-11-25 07:37:13 Protocol options: explicit-exit-notify 1
```

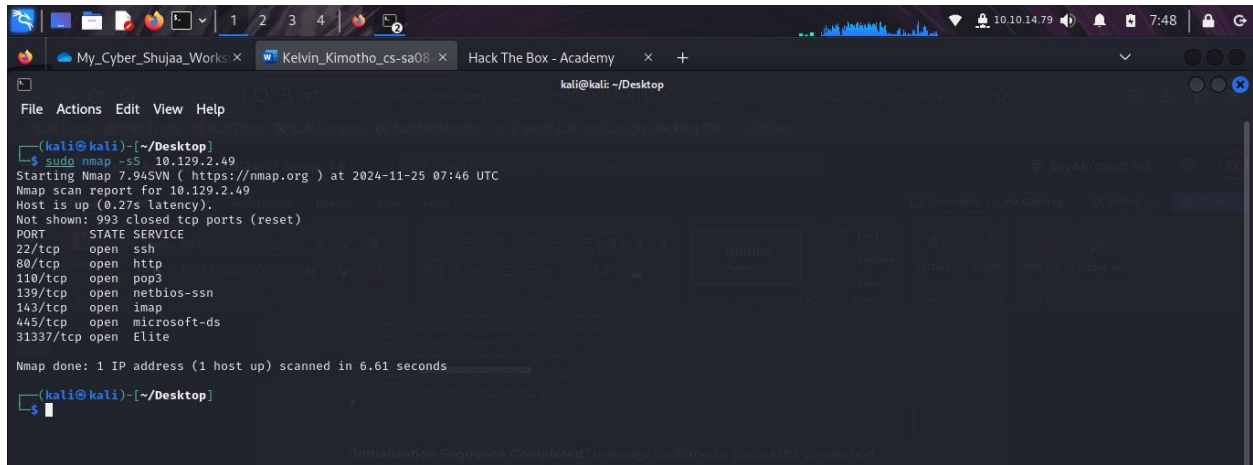
“Initialization Sequence Completed” message confirmed a successful connection.

Question: Find all TCP ports on your target. Submit the total number of found TCP ports as the answer.

Answer: 7



I performed a TCP scan by running the following command against the target “**sudo nmap -sS 10.129.2.49**”. The **-sS** flag helps us perform a **TCP SYN-Scan**. There we **7** open TCP ports.



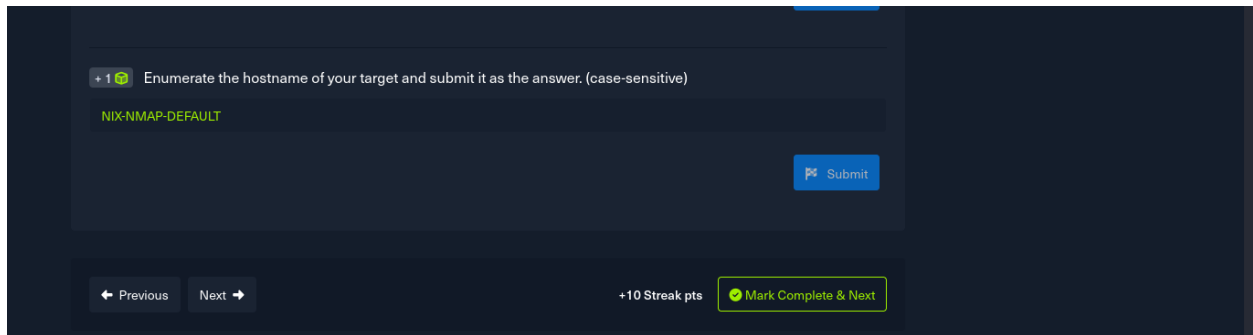
```
(kali@kali)-[~/Desktop]
$ sudo nmap -sV 10.129.2.49
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-25 07:46 UTC
Nmap scan report for 10.129.2.49
Host is up (0.27s latency).
Not shown: 993 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
110/tcp   open  pop3
139/tcp   open  netbios-ssn
143/tcp   open  imap
445/tcp   open  microsoft-ds
31337/tcp open  Elite

Nmap done: 1 IP address (1 host up) scanned in 6.61 seconds

(kali@kali)-[~/Desktop]
$
```

Question: Enumerate the hostname of your target and submit it as the answer. (case-sensitive)

Answer: NIX-NMAP-DEFAULT



I used the -sV option to discover services and their versions.


```
(kali@kali)-[~/Desktop]
$ sudo nmap -sV 10.129.2.49
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-25 07:54 UTC
Stats: 0:02:26 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 85.71% done; ETC: 07:57 (0:00:23 remaining)
Nmap scan report for 10.129.2.49
Host is up (0.29s latency).
Not shown: 993 closed tcp ports (reset)
PORT      STATE SERVICE        VERSION
22/tcp    open  ssh            OpenSSH 7.6p1 Ubuntu 4ubuntu0.7 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http           Apache httpd 2.4.29 ((Ubuntu))
110/tcp   open  pop3           Dovecot pop3d
139/tcp   open  netbios-ssn    Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
143/tcp   open  imap           Dovecot imapd (Ubuntu)
445/tcp   open  netbios-ssn    Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
31337/tcp open  Elite?
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port31337-TCP:V=7.94SVN%I=7%D-11/25%Time=67442D61%P=x86_64-pc-linux-gnu
SF:%r(GetRequest,1F,7220x20HTB{pr0F7pDv3r510nb4nn3r}\r\n");
Service Info: Host: NIX-NMAP-DEFAULT; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 169.50 seconds

(kali@kali)-[~/Desktop]
$
```

Saving Nmap Scan Results

Saving Results in Different Formats

- **Normal Output (-oN):** Saved with .nmap extension.
- **Grepable Output (-oG):** Saved with .gnmap extension.
- **XML Output (-oX):** Saved with .xml extension.
- **Save All Formats (-oA):** Saves results in all three formats with a common base filename.

An example Command to Save in All Formats

- “ **sudo nmap 10.129.2.28 -p- -oA target** “. This save results in three files: target.nmap, target.gnmap, target.xml.

```
kali@kali: ~/Desktop
File Actions Edit View Help
(kali@kali)~$ sudo nmap -A 10.129.2.49 -oA mytarget
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-25 08:06 UTC
Nmap scan report for 10.129.2.49
Host is up (0.24s latency).
Not shown: 993 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 7.6p1 Ubuntu 4ubuntu0.7 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|_ 2048 71:c1:89:90:7f:fd:4f:60:e0:54:f3:85:e6:35:6c:2b (RSA)
|_ 256 e1:8e:53:18:42:af:2a:de:c0:12:1e:2e:54:06:4f:70 (ECDSA)
|_ 256 1a:cc:ac:d4:94:5c:d6:1d:71:e7:39:de:14:27:3c:3c (ED25519)
80/tcp    open  http         Apache httpd 2.4.29 ((Ubuntu))
|_ http-title: Apache2 Ubuntu Default Page: It works
110/tcp   open  pop3         Dovecot pop3d
|_ pop3-capabilities: PIPELINING RESP-CODES TOP CAPA AUTH-RESP-CODE UIDL SASL
139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
143/tcp   open  imap         Dovecot imapd (Ubuntu)
|_ imap-capabilities: LOGINDISABLED A0001 ID OK listed Pre-login SASL-IR post-login more LITERAL+ have IMAP4rev1 capabilities ENABLE IDLE LOGIN-REFERRALS
445/tcp   open  netbios-ssn Samba smbd 4.7.6-Ubuntu (workgroup: WORKGROUP)
31337/tcp open  Elite?
|_ fingerprint-strings:
|_  GetRequest:
|_  220 HTB{pr0F7pDv3r510nb4nn3r}
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service:
SF-Port31337-TCP:V=7.94SVN:KI=7XD-11/25K:Time=67443019KP-x86_64-pc-linux-gnu
SF:Kr(GetRequest,1F,"220x20HTB{pr0F7pDv3r510nb4nn3r}\r\n");
Device type: general purpose
Running: Linux 5.X
OS CPE: cpe:/o:linux:linux_kernel:5.0
OS details: Linux 5.0
Network Distance: 2 hops
Service Info: Host: NIX-NMAP-DEFAULT; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Output File Formats
• Normal Output. Human-readable format with scan details (ports, state, services).
• Greppable Output. Text format for easy parsing, suitable for further automation.
• XML Output. Structured XML format, useful for creating HTML reports.

Host script results:
|_ smb2-time:
|_ date: 2024-11-25T08:09:31
|_ start_date: N/A

Authentication level: supported
|_ challenge_response: supported
|_ message_signing: disabled (dangerous, but default)
|_ nbstat: NetBIOS name: NIX-NMAP-DEFAULT, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)

TRACEROUTE (using port 1720/tcp)
HOP RTT ADDRESS
1 224.56 ms 10.10.14.1
2 225.46 ms 10.129.2.49

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 188.19 seconds

(kali@kali)~$ ls
mytarget.gnmap mytarget.nmap mytarget.xml

(kali@kali)~$
```

Output File Formats

- **Normal Output.** Human-readable format with scan details (ports, state, services).
- **Greppable Output.** Text format for easy parsing, suitable for further automation.
- **XML Output.** Structured XML format, useful for creating HTML reports.

Viewing the Output

- We cat target.nmap for normal output.
- cat target.gnmap for greppable output.
- cat target.xml for XML output.

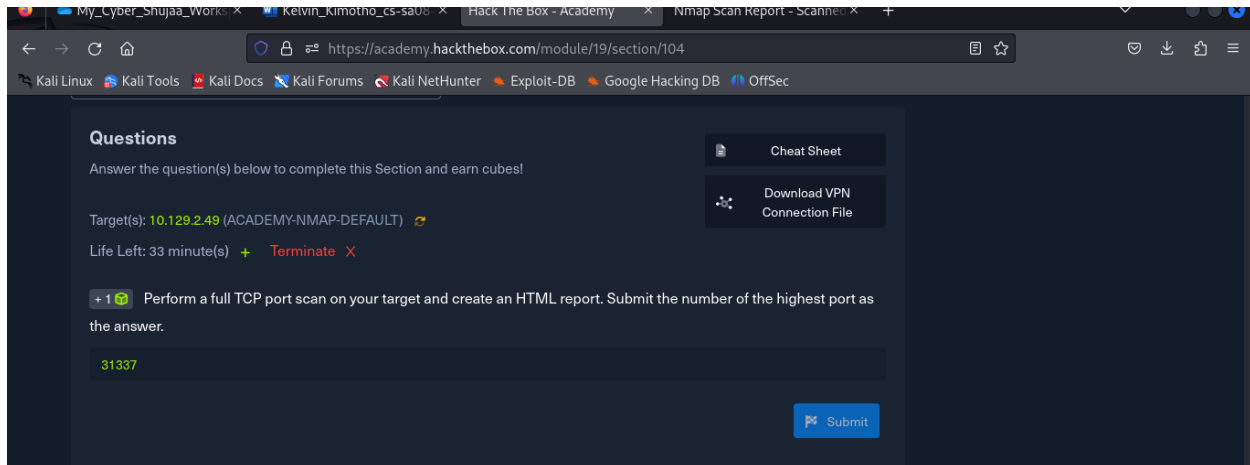
Creating HTML Reports from XML

- We use “**xsltproc target.xml -o target.html**” to convert XML to an HTML report.

- This report is easy to read and useful for documentation.

Question: Perform a full TCP port scan on your target and create an HTML report. Submit the number of the highest port as the answer.

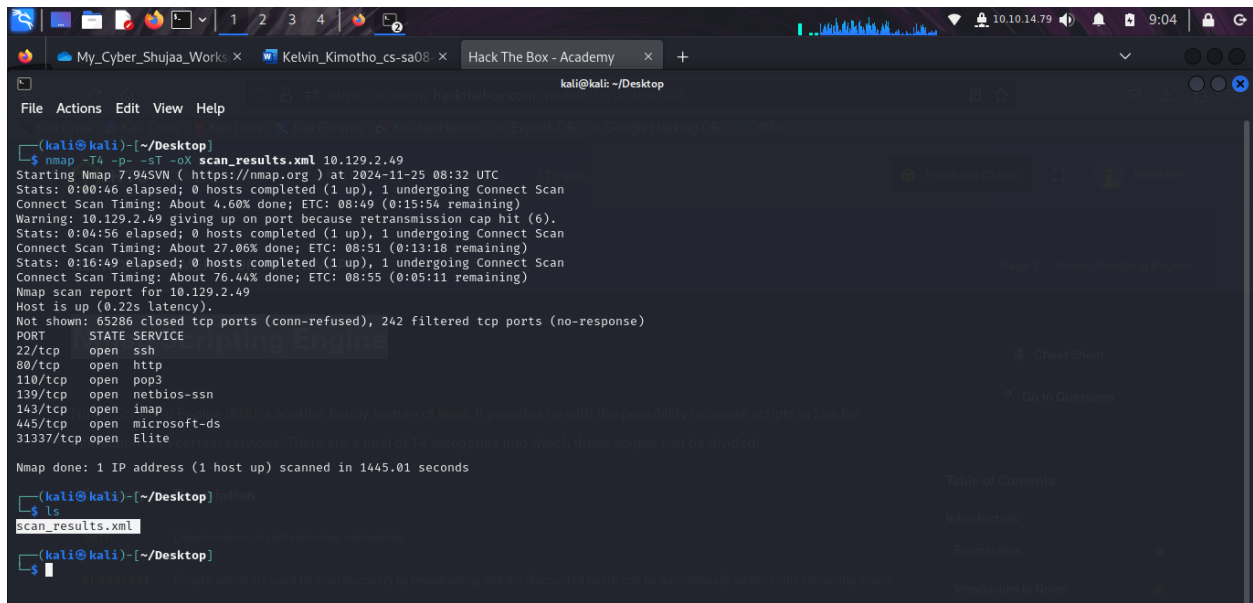
Answer: 31337



I performed a **TCP SYN-Scan** saving results saving the results in an xml file using **-oX** option.

“**nmap -T4 -p- -sT -oX scan_results.xml 10.129.2.49**”.

- **-p-**: This tells Nmap to scan all 65,535 possible TCP ports.
- **-sT**: This performs a TCP connect scan, which establishes a full TCP connection (handshake).
- **-oX scan_results.xml**: This option saves the output in XML format to a file called scan_results.xml.
- **-T4**: Increases the speed of the scan, which is more aggressive than the default.



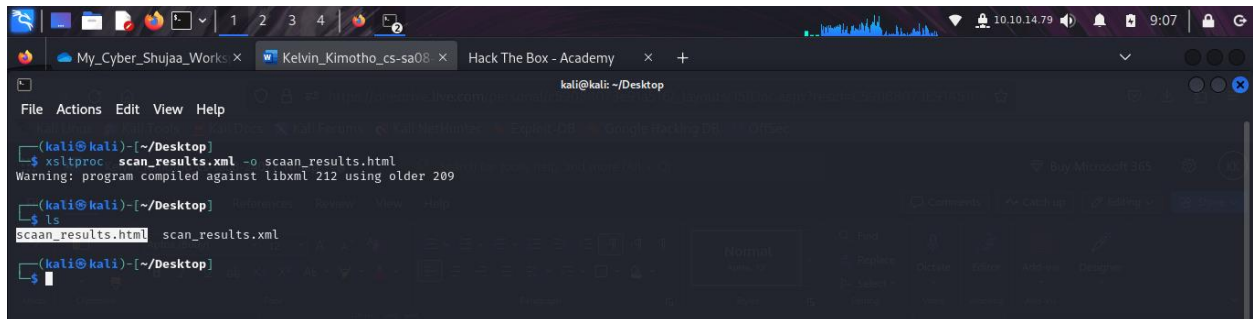
```
(kali@kali)-[~/Desktop]
$ nmap -T4 -p- -sT -oX scan_results.xml 10.129.2.49
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-25 08:32 UTC
Stats: 0:00:46 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 4.60% done; ETC: 08:49 (0:15:54 remaining)
Warning: 10.129.2.49 giving up on port because retransmission cap hit (6).
Stats: 0:04:56 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 27.06% done; ETC: 08:51 (0:13:18 remaining)
Stats: 0:16:49 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 76.44% done; ETC: 08:55 (0:05:11 remaining)
Nmap scan report for 10.129.2.49
Host is up (0.22s latency).
Not shown: 65286 closed tcp ports (conn-refused), 242 filtered tcp ports (no-response)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
110/tcp   open  pop3
139/tcp   open  netbios-ssn
143/tcp   open  imap
445/tcp   open  microsoft-ds
31337/tcp open  elite

Nmap done: 1 IP address (1 host up) scanned in 1445.01 seconds

(kali@kali)-[~/Desktop]
$ ls
scan_results.xml

(kali@kali)-[~/Desktop]
$
```

Then using **xsltproc** tool i converted the xml file to html. “**xsltproc scan_results.xml -o scan_report.html**”.



```
(kali@kali)-[~/Desktop]
$ xsltproc scan_results.xml -o scan_report.html
Warning: program compiled against libxml 212 using older 209

(kali@kali)-[~/Desktop]
$ ls
scan_report.html  scan_results.xml

(kali@kali)-[~/Desktop]
$
```

I then opened the html file on Firefox browser.



The Goal of Service Enumeration is to Identify the application and its version on the target system to search for vulnerabilities and possible exploits.

- We start with a quick port scan to minimize traffic and avoid detection by security mechanisms.
- Run a port scan in the background to detect all open ports using `-p-`.
- We use `-sV` to detect versions of services on open ports.

- We can use **sudo nmap 10.129.2.28 -p- -sV** to detect open ports and service versions.

- Scan progress can be viewed with the space bar during the scan.

Additional Scan Options

- **--stats-every=5s**: Displays scan progress every 5 seconds.
- **-v** or **-vv**: Increases verbosity, providing detailed information about open ports.

Service Information

- Nmap identifies open ports, service names, and version numbers, which helps to find vulnerabilities.
- Example ports are 22 (SSH), 25 (SMTP), 80 (HTTP) and their respective versions are listed.

Banner Grabbing

- Nmap looks at the banners of the scanned ports. If unable to identify version numbers from the banner, it uses signature-based matching.

Challenges with Nmap include

- Nmap may miss some details, such as when a banner is manipulated or missing.
- Some services may delay sending banners or require a manual connection to reveal more details.

Manual Banner Grabbing

- We can use **nc (Netcat)** to manually connect to the service and grab the banner.
- An example using netcat is, **nc -nv 10.129.2.28 port** manually connects to a service revealing the banner information.
- **Tcpdump** can be used to intercept network traffic and capture more detailed service information not shown by Nmap.

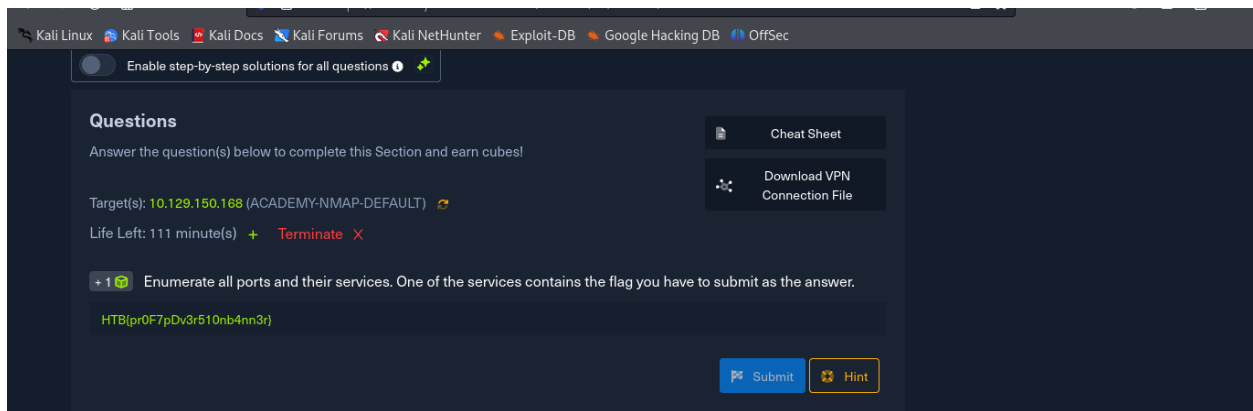
Tcpdump Process

- **Tcpdump** captures network traffic between the attacker and target, showing the three-way handshake and the banner information exchanged between client and server.

Service enumeration and version detection through Nmap, supplemented by manual methods like banner grabbing and Tcpdump, help us identify detailed service information and potential vulnerabilities.

Question: Enumerate all ports and their services. One of the services contains the flag you have to submit as the answer.

Answer: HTB{pr0F7pDv3r510nb4nn3r}



I performed the following scan on my target “ **nmap -sV -p- 10.129.2.49** “where,

- **-sV** option tells nmap to perform service version detection, which will reveal the services running on the open ports.
- **-p-** tells nmap to scan all 65535 possible ports (from 1 to 65535).

I discovered an unknown service “**Elite?**” which was running on port “**31337**”.

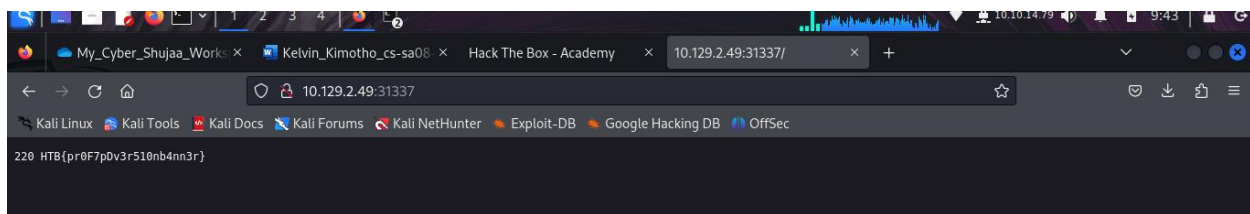
```
kali@kali: ~/Desktop
File Actions Edit View Help

(kali@kali)~$ sudo nmap -sV -T4 -p- 10.129.2.49
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-25 09:15 UTC
Stats: 0:00:02 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 0.29% done
Stats: 0:00:45 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 4.85% done; ETC: 09:31 (0:14:44 remaining)
Warning: 10.129.2.49 giving up on port because retransmission cap hit (6).
Stats: 0:08:00 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 48.14% done; ETC: 09:32 (0:08:36 remaining)
Stats: 0:15:37 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 90.69% done; ETC: 09:32 (0:01:36 remaining)
Stats: 0:17:24 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 99.99% done; ETC: 09:32 (0:00:00 remaining)
Stats: 0:19:41 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 85.71% done; ETC: 09:35 (0:00:22 remaining)
Nmap scan report for 10.129.2.49
Host is up (0.23s latency).
Not shown: 65499 closed tcp ports (reset), 29 filtered tcp ports (no-response)
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 7.6p1 Ubuntu 4ubuntu0.7 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http         Apache httpd 2.4.29 ((Ubuntu))
110/tcp   open  pop3         Dovecot pop3d
139/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
143/tcp   open  imap         Dovecot imapd (Ubuntu)
445/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
31337/tcp open  Elite?
1 Service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service:
SF-Port31337-TCP:V=7.94SVN%I=7%D=11/25%Time=67444459%P=x86_64-pc-linux-gnu
SF:R!(GetRequest,1F,"220x20HTB{pr0F7pDv3r510nb4nn3r}\r\n");
Service Info: Host: NIX-NMAP-DEFAULT; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1216.10 seconds

(kali@kali)~$
```

I then opened the target machine IP on Firefox web browser on port **31337** and an page with the flag was rendered. ”**http://10.129.2.49:31337**”.



I also tried curl tool to find the same by running the following command on my attacking machine terminal. ” **sudo curl http://10.129.150.168:31337**”

The above command failed and gave this error “**curl: (1) Received HTTP/0.9 when not allowed**”.

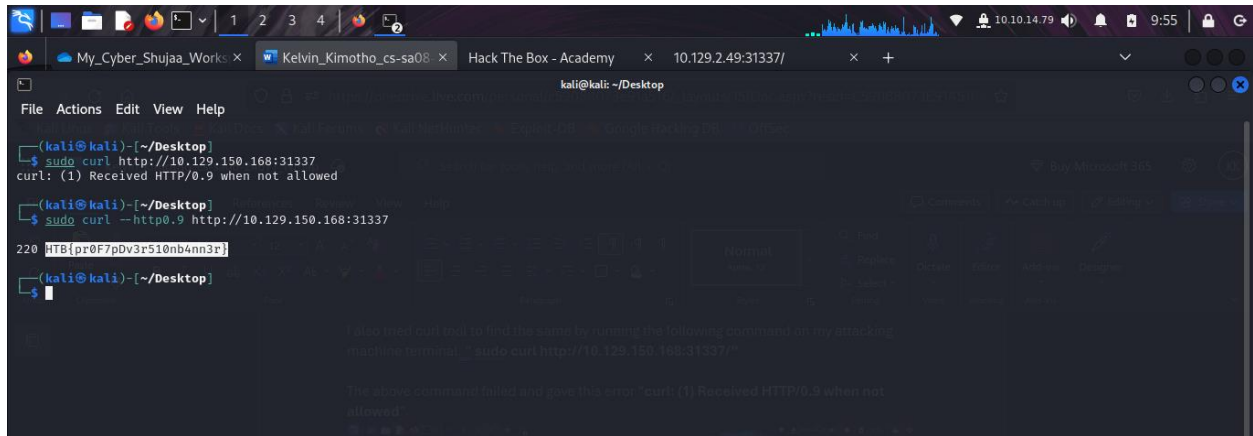
```
kali@kali: ~/Desktop
File Actions Edit View Help

(kali@kali)~$ sudo curl http://10.129.150.168:31337
curl: (1) Received HTTP/0.9 when not allowed

(kali@kali)~$
```

I did some research and came across a concept where we can **Force curl to accept HTTP/0.9** .

If we need to work with a server that is sending **HTTP/0.9** responses, we can tell curl to allow it by using the **--http0.9** option. My new command was “ **sudo curl --http0.9 http://10.129.150.168:31337**”. This enabled me to retrieve the flag using curl.



Nmap Scripting Engine

The **Nmap Scripting Engine (NSE)** is a powerful feature within Nmap, enabling advanced and automated scanning capabilities.

It allows the use of scripts, written in Lua, to interact with services on a target machine. These scripts extend Nmap's functionality, making it more versatile in detecting vulnerabilities, gathering information, and even exploiting weaknesses.

Nmap Scripting Engine Categories

There are 14 categories of scripts in NSE, each serving a distinct purpose.

Category	Description
auth	Checks for authentication credentials.
broadcast	Discovers hosts through broadcasting and adds them to the scan automatically.
brute	Brute-forces login credentials for services.
default	Executes a default set of scripts with the -sC option.
discovery	Identifies accessible services.
dos	Denial of Service scripts, which can harm systems, so these are used cautiously.
exploit	Attempts to exploit known vulnerabilities in a target system.

external	Utilizes external services for further processing.
fuzzer	Identifies vulnerabilities by sending varied or unexpected inputs.
intrusive	Scripts that might cause negative effects on the target system.
malware	Detects malware infections on the target.
safe	Non-intrusive scripts that do not perform destructive actions.
version	Detects the version of services running.
vuln	Identifies specific vulnerabilities in services or applications.

Using NSE Scripts in Nmap

We can invoke NSE scripts using various methods, such as running a default script set, specifying a category of scripts, or choosing individual scripts. Here are the key ways to use scripts:

Default Scripts. We can use the **-sC** option to run a set of default scripts.

- An example command format is “ **sudo nmap <target> -sC**”.

Specific Script Categories. To scan a target with all scripts in a certain category, use the **--script <category>** option.

- An example command format is “ **sudo nmap <target> --script <category>**”. For example, “ **--script vuln** “ would run all vulnerability-related scripts.

Define Specific Scripts. We can also specify individual scripts separated by commas.

- An example of the command format “ **sudo nmap <target> --script <script-name>,<script-name>,...**”.

An example with SMTP

Running specific scripts against an SMTP service using two scripts namely,

- **Banner** which retrieves a banner from the service.
- **smtp-commands**. Lists supported SMTP commands.

“ **sudo nmap 10.129.2.28 -p 25 --script banner,smtp-commands**” .

- The **banner** script reveals that the server is running, and **smtp-commands** shows supported commands such as VRFY, STARTTLS, and SIZE.

Aggressive Scanning with Nmap

The aggressive scan (-A) runs multiple tests in one go: **service version detection**, **OS detection**, **traceroute**, and **default scripts**.

It's useful for getting a deeper understanding of the target. An example given was “**sudo nmap 10.129.2.28 -p 80 -A**”. This would give us useful information about the target web application .

Vulnerability Scanning

We can also use the **vuln** category to search for vulnerabilities specific to the discovered services. For example, scanning HTTP (port 80) for vulnerabilities in WordPress and Apache

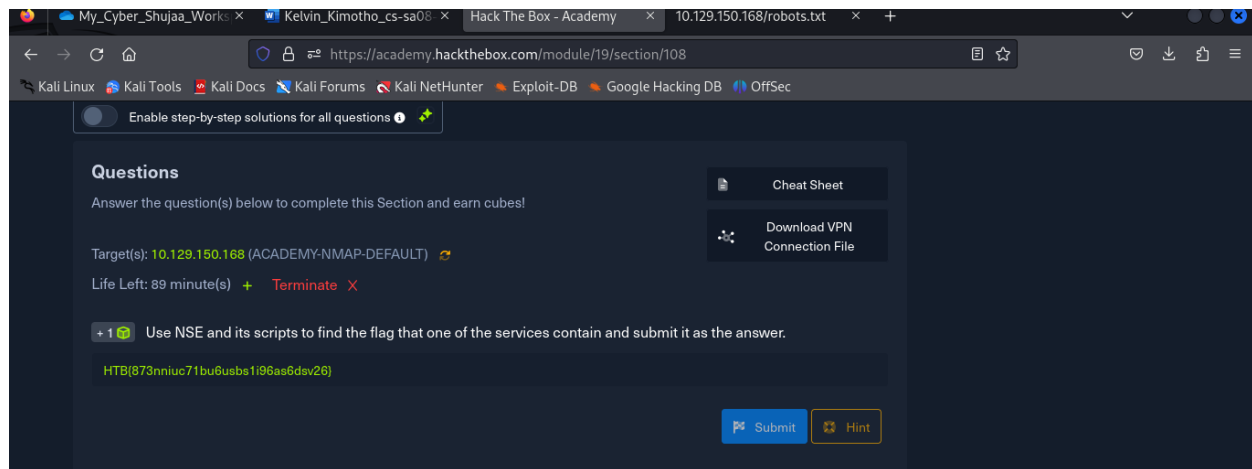
“**sudo nmap <Target> -p 80 -sV --script vuln**”. This would reveal information like:

This scan reveals

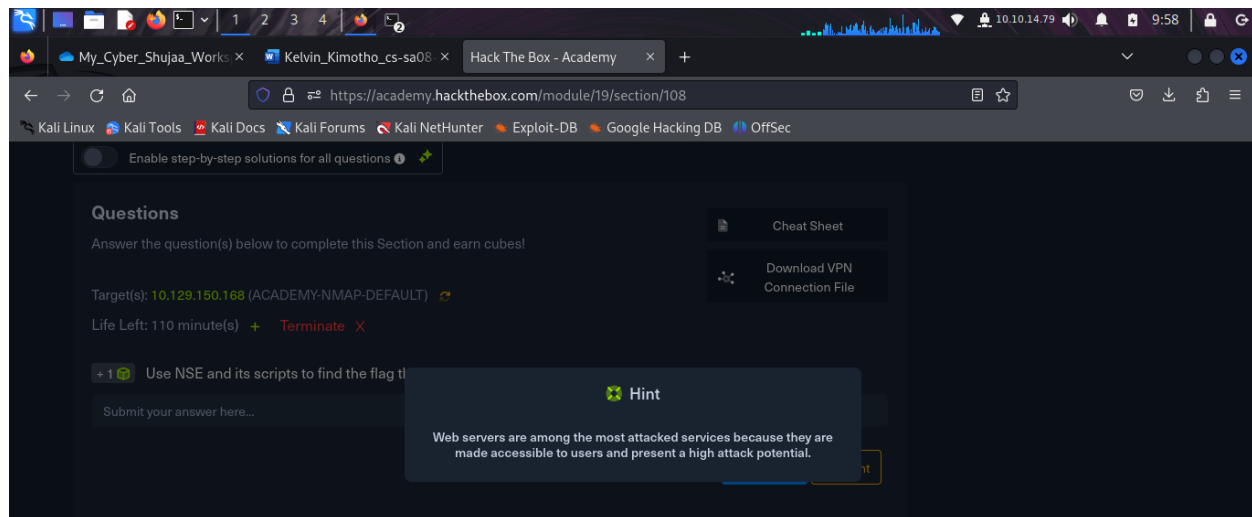
- WordPress version 5.3.4 is identified.
- The **admin** username is found on the WordPress installation.
- Known vulnerabilities for **Apache 2.4.29** (e.g., CVE-2019-0211) are highlighted.

Question: Use NSE and its scripts to find the flag that one of the services contain and submit it as the answer.

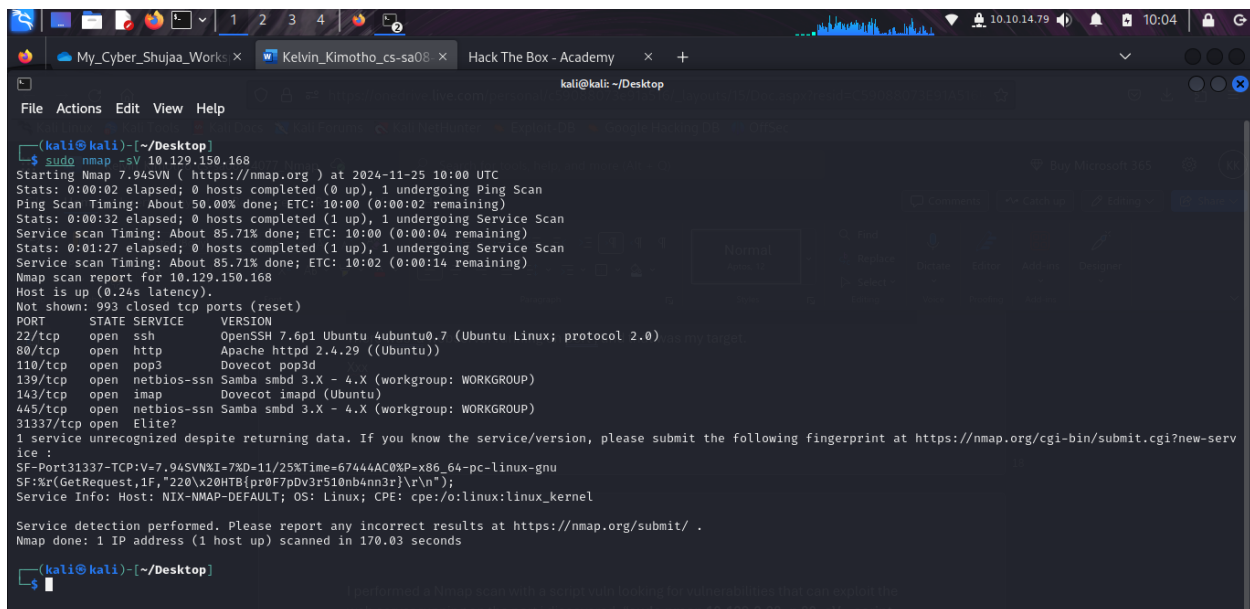
Answer: HTB{873nniuc71bu6usbs1i96as6dsv26}



Using the given hint, I was able to narrow down my scan targeting web services running on the target system.



There was a webserver running on port 80 (**Apache webserver**), and that was my target.

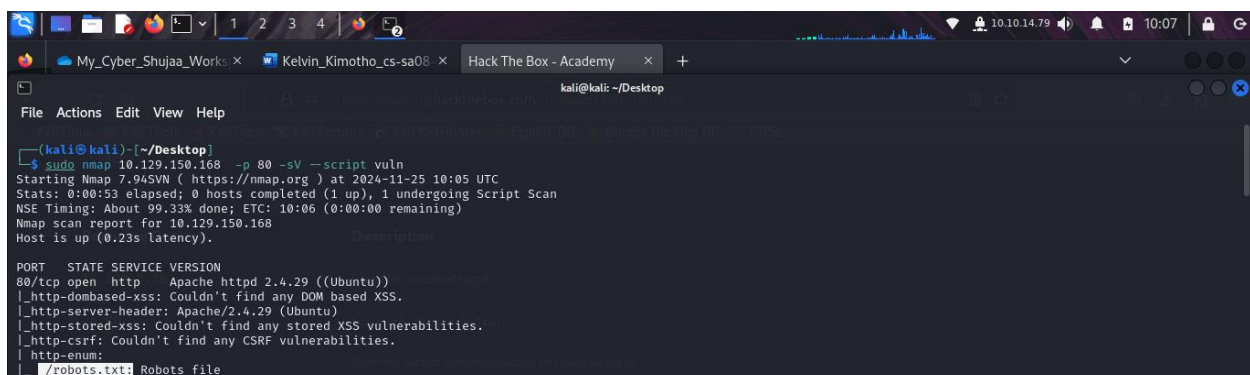


```
(kali@kali)-[~/Desktop]
$ sudo nmap -sV 10.129.150.168
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-25 10:00 UTC
Stats: 0:00:02 elapsed; 0 hosts completed (0 up), 1 undergoing Ping Scan
Ping Scan Timing: About 50.00% done; ETC: 10:00 (0:00:02 remaining)
Stats: 0:00:32 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 85.71% done; ETC: 10:00 (0:00:04 remaining)
Stats: 0:01:27 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 85.71% done; ETC: 10:02 (0:00:14 remaining)
Nmap scan report for 10.129.150.168
Host is up (0.24s latency).
Not shown: 993 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.6p1 Ubuntu 4ubuntu0.7 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http     Apache httpd 2.4.29 ((Ubuntu))
110/tcp   open  pop3     Dovecot pop3d
139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
143/tcp   open  imap     Dovecot imapd (Ubuntu)
445/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
31337/tcp open  Elite?
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port31337-TCP:V=7.94SVNXI=7%D=11/25%Time=67444AC0P=x86_64-pc-linux-gnu
SF:Rr(GetRequest,1F,"220"x20HTB[pr0F7pDv3r510nb4nn3r]\r\n");
Service Info: Host: NIX-NMAP-DEFAULT; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 170.03 seconds

(kali@kali)-[~/Desktop]
$
```

I performed a Nmap scan with a script vuln looking for vulnerabilities that can exploit the web server (**Apache**) running on the port i discovered. “**sudo nmap 10.129.150.168 -p 80 -sV --script vuln**”.

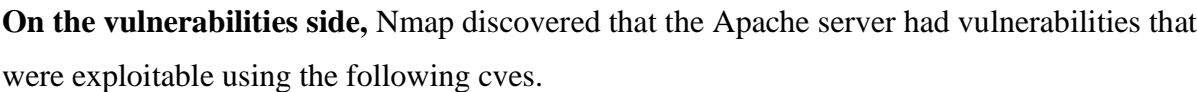


```
(kali@kali)-[~/Desktop]
$ sudo nmap 10.129.150.168 -p 80 -sV --script vuln
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-25 10:05 UTC
Stats: 0:00:53 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 99.33% done; ETC: 10:06 (0:00:00 remaining)
Nmap scan report for 10.129.150.168
Host is up (0.23s latency).
PORT      STATE SERVICE VERSION
80/tcp    open  http     Apache httpd 2.4.29 ((Ubuntu))
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
|_http-csrf: Couldn't find any CSRF vulnerabilities.
|_http-enum:
|_robots.txt: Robots file
|_vulnscan:

(kali@kali)-[~/Desktop]
$
```

I first discovered that there existed a **robots.txt** file in the server. Robots.txt files should be hidden and not accessible by the public. I had to check for a flag in there.

I opened the file on Firefox on this URL **http://10.129.150.168:80/robots.txt**. That’s how i found the flag.



Scan Performance is crucial for large networks or low bandwidth situations.

Here are the adjustable Options we can use to optimize scan performance

- **-T <0-5>**. Control scan speed/aggressiveness.
- **--min-parallelism <number>**. set the number of simultaneous parallel probes.
- **--max-rtt-timeout <time>**. Set maximum Round Trip Time for packets.

- **--min-rate <number>**. Set minimum number of packets to send per second.
- **--max-retries <number>**. Control retries on failed responses for a port.

Timeouts

- **RTT (Round-Trip-Time)**. Time to send a packet and get a response.
- **Default RTT Timeout** is 100ms.
- **For example**, scanning with **--initial-rtt-timeout 50ms --max-rtt-timeout 100ms** reduces scan time but finds fewer hosts.

Max Retries

- Setting **--max-retries** to 0 skips ports with no responses help speed up the scan.
- **For example**, Reducing retries from 10 to 0 led to fewer ports found but faster scan time.

Rates

- **Minimum Rate (--min-rate <number>)**. It sets the number of packets sent per second to speed up scans.
- **For example**, Setting **--min-rate 300** scans a network faster while finding the same number of open ports.

Timing

Timing Templates (-T <0-5>). Adjust scan aggressiveness, with values ranging from 0 (paranoid) to 5 (insane).

- **T0** is paranoid (very slow, avoids detection).
- **T1**. Sneaky (slow, avoids detection).
- **T2**. Polite (slows scan to avoid overwhelming target).
- **T3**. Normal (default).
- **T4**: Aggressive (faster, risks detection).
- **T5**. Insane (extremely fast, high risk of detection).

Firewall and IDS/IPS Evasion

Firewalls are software components that control network traffic based on predefined rules, filtering packets to prevent unauthorized access.

- Packets may be either **dropped** (no response) or **rejected** (returns an RST flag with ICMP error codes).

IDS (Intrusion Detection System) detect potential attacks by analyzing network traffic and alerts administrators.

IPS (Intrusion Prevention System) goes further by actively blocking detected attacks.

- Both systems analyze packets for attack signatures or anomalies.

Nmap Scanning Techniques

TCP ACK scan (-sA). This is harder for firewalls and IDS/IPS to detect than regular SYN scans, as it only sends packets with the ACK flag.

Firewalls may pass these packets if they can't determine if the connection was initiated internally or externally.

SYN scan (-sS) is detectable by firewalls since it sends a packet to initiate a connection using the SYN flag, which is often blocked by firewalls.

Port State Responses

- For SYN scan, RST (Reset) is returned if the port is open.
- For ACK scan. No response (port is filtered) or RST (if the port is open).

Firewall Evasion

- Firewalls may drop or reject packets based on rules.
- **For example**, Port 21 (FTP) and 25 (SMTP) may be filtered, while port 22 (SSH) might be open.

Detecting IDS/IPS Systems

- IDS/IPS detection is difficult because they passively monitor traffic.
- A penetration tester can use multiple VPS IP addresses to determine if the system is blocking suspicious traffic.
- Blocked VPS access indicates that the network has IDS/IPS in place.

Using Decoys

- **Decoy Scanning (-D).** Generates fake IP addresses to hide the origin of the scan and evade detection by IDS/IPS systems. This makes it harder for the system to identify the real attacker.
- **An example** using multiple decoys is (e.g., -D RND:5) disguises the real source IP address by placing it between other random addresses.

Advanced Evasion Techniques

- **Source IP Manipulation (-S).** Use a different source IP to avoid detection.
- **Using Specific Source Ports.** Scans conducted from trusted ports (e.g., port 53, typically used for DNS) can pass through firewalls if the firewall doesn't filter traffic on that port properly.
- **DNS Proxying.** By setting custom DNS servers or using port 53 for scans, an attacker may evade detection if the firewall trusts DNS traffic.

Testing Specific Firewall Rules

- Scanning ports such as 50000 with different source ports to determine if firewalls are blocking specific ranges.
- **For example,** A SYN scan from port 53 can bypass a firewall that accepts DNS traffic.

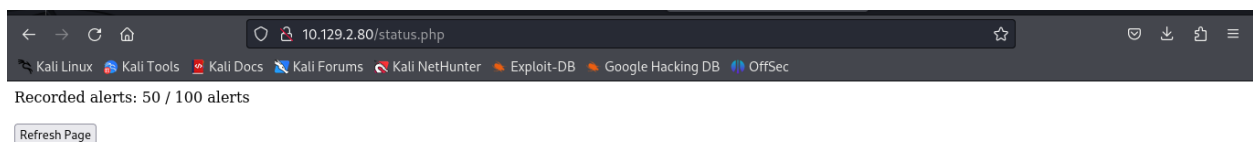
Combining Techniques for Evasion

- **Multiple Scans with Different Parameters.** Using combinations like decoys, SYN scan, or scanning from trusted ports (e.g., DNS port 53) can help avoid detection by firewalls or IDS/IPS systems.
- Testing with multiple VPS or different source IPs is a common approach to evade detection during penetration tests.

Firewall and IDS/IPS Evasion - Easy Lab

A company hired me to test their IT security defenses, including their IDS and IPS systems. My client wants to increase their IT security and will, therefore, make specific improvements to their IDS/IPS systems after each successful test. I do not know, however, according to which guidelines these changes will be made. My goal here is to find out specific information from the given situations.

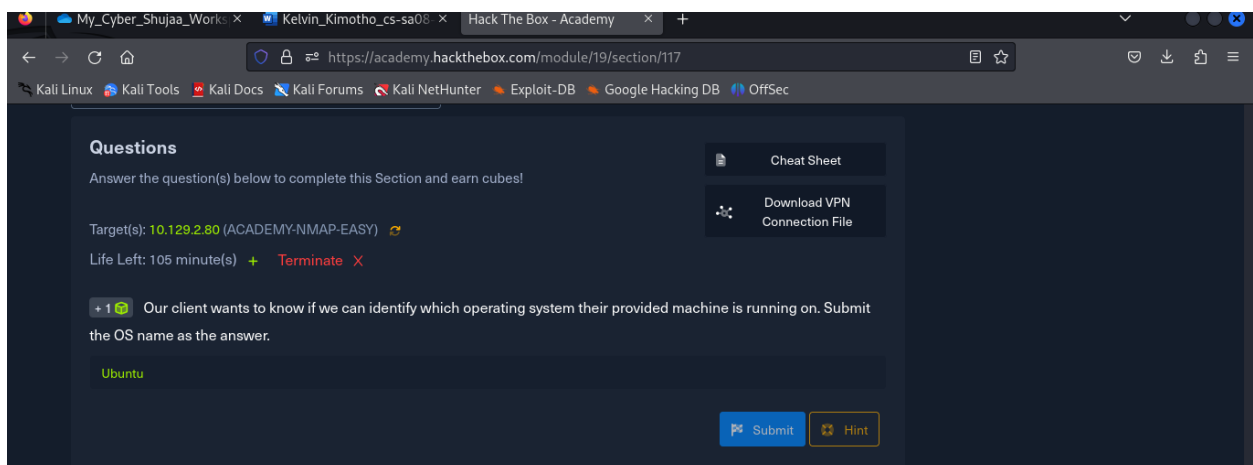
We are only ever provided with a machine protected by **IDS/IPS** systems and can be tested. we have access to a status web page at “<http://10.129.2.80/status.php>”.



This page shows us the number of alerts. We know that if we receive a specific number of alerts, we will be banned. Therefore, we have to test the target system as quietly as possible.

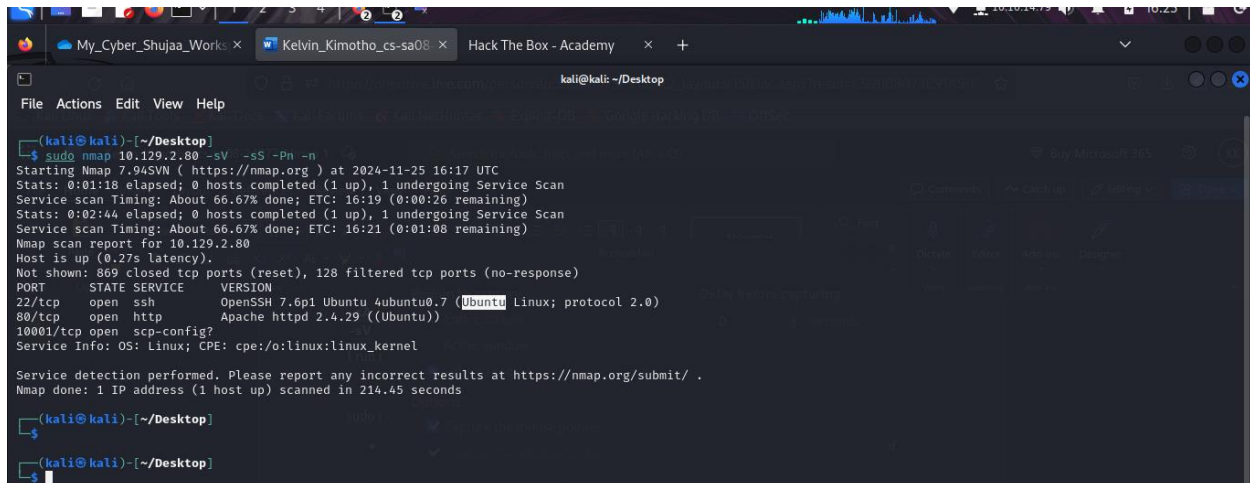
Question: Our client wants to know if we can identify which operating system their provided machine is running on. Submit the OS name as the answer.

Answer: Ubuntu



Packets with the **ACK** flag are often passed by the firewall because the firewall cannot determine whether the connection was first established from the external network or the internal network.

I ran the following command “**sudo nmap 10.129.2.80 -sV -sS -Pn -n**”

A screenshot of a Kali Linux terminal window. The terminal shows the execution of the command 'sudo nmap 10.129.2.80 -sV -sS -Pn -n'. The output indicates that the host is up and provides details about the scan, including the number of open ports (3) and the services running on them: ssh (OpenSSH 7.6p1 Ubuntu 4ubuntu0.7), http (Apache httpd 2.4.29), and scp-config? (10001/tcp). The terminal also shows the nmap version (7.94SVN) and the scan timing. The user's prompt is '(kali@kali)-[~/Desktop]' and the command prompt is '\$'.

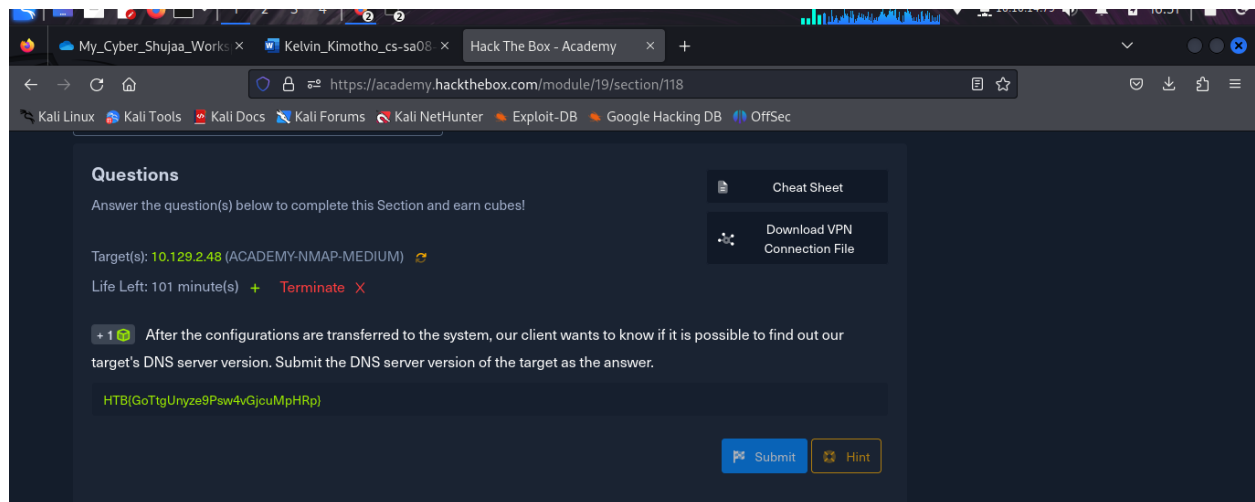
- **-sS** performs a **TCP SYN scan**, which is less intrusive and stealthy. It sends only the initial SYN packets to the target ports, making it harder for IDS/IPS systems to detect compared to a full TCP connection scan.
- **-Pn** skips host discovery (ping scan), assuming the target is alive even if no response is received to a ping. This is useful if the target machine blocks ICMP packets (ping) but still responds to other protocols.
- **-n** disables DNS resolution, which avoids the overhead of trying to resolve hostnames and prevents revealing any domain-related information, making the scan faster and more covert.
- **-sV** enables for services running version detection.

Firewall and IDS/IPS Evasion - Medium Lab

After we conducted the first test and submitted our results to our client, the administrators made some changes and improvements to the IDS/IPS and firewall. We could hear that the administrators were not satisfied with their previous configurations during the meeting, and they could see that the network traffic could be filtered more strictly.

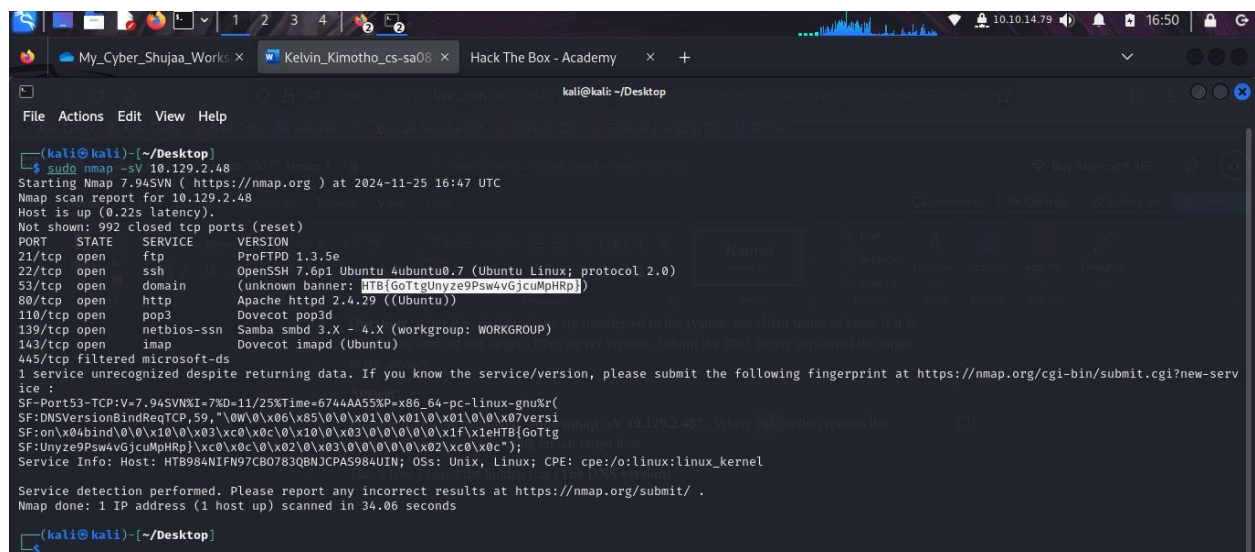
Question: After the configurations are transferred to the system, our client wants to know if it is possible to find out our target's DNS server version. Submit the DNS server version of the target as the answer.

Answer: HTB{GoTtgUnyze9Psw4vGjcuMpHRp}



I ran the following command “**sudo nmap -sV 10.129.2.48**”. Where **-sV** option reveals the versions of the services running on our target host.

That’s how i found the hidden flag (**The DNS version**).

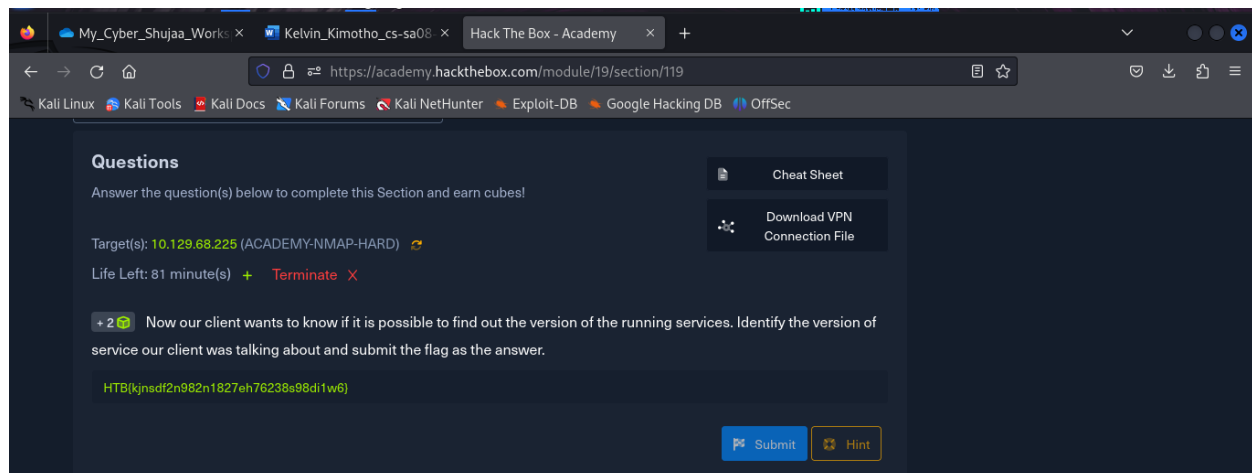


Firewall and IDS/IPS Evasion - Hard Lab

With our second test's help, our client was able to gain new insights and sent one of its administrators to a training course for IDS/IPS systems. As our client told us, the training would last one week. Now the administrator has taken all the necessary precautions and wants us to test this again because specific services must be changed, and the communication for the provided software had to be modified.

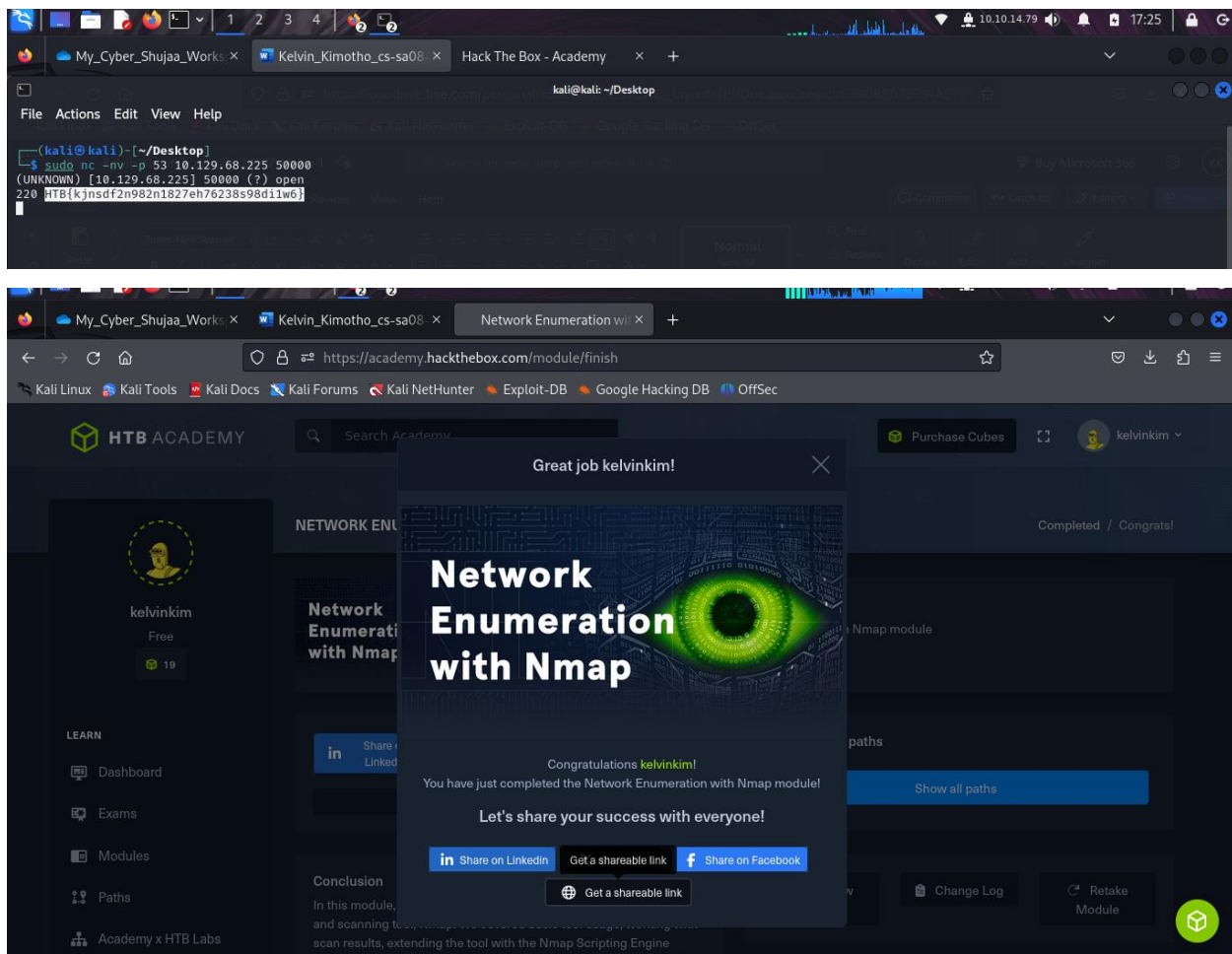
Question: Now our client wants to know if it is possible to find out the version of the running services. Identify the version of service our client was talking about and submit the flag as the answer.

Answer: HTB{kjnsdf2n982n1827eh76238s98di1w6}



The objective is to determine the versions of the currently active services. To achieve this, I used **netcat** tool. However, before proceeding, I needed to determine the specific port we are targeting. I accomplished this by conducting a full port scan (**-p-**) which will reveal the port **50000**.

Following this, I set up a netcat listener to operate between the DNS port 50000. I ran the following command “**sudo nc -nv -p 53 10.129.68.225 50000**”



Conclusion

In conclusion, completing the Network Enumeration with Nmap module on HackTheBox has significantly enhanced my understanding of network security assessments and the vital role Nmap plays in identifying vulnerabilities. By learning various scanning techniques and the different capabilities of Nmap, such as host discovery, port scanning, service enumeration, and OS detection, I now have a more comprehensive grasp of how to map out networks and uncover potential weaknesses. The practical application of these techniques in a controlled environment has also sharpened my skills in interpreting scan results and leveraging this data to identify attack vectors. This module has not only strengthened my technical abilities but also deepened my appreciation for the importance of thorough enumeration in the vulnerability assessment process.

