



Lab4: Network Penetration Testing Methodology-Internal (1/3)

INFO40587: ETHICAL HACKING

Kevin Harianto | 991602128 | June 1, 2024

Contents

Contents

Contents	1
Executive Summary	2
Exercise 1: Scanning with the Tool Netdiscover	3
Exercise 2: Scanning and Scripting with hping3	8
Exercise 3: Scanning and Building a Target Database	16
Exercise 4: Using Workspaces and db_nmap	20
Exercise 5: Performing Passive OS Fingerprinting to Obtain Remote Operating System Information	26
Exercise 6: OS Fingerprinting with Nmap	30
Exercise 7: Scanning with Dmitry	32
Conclusion.....	35

Executive Summary

{state the objectives, approaches, methods/tools used, learning outcome, comments/overall observations}.

The objective of this lab is to provide knowledge on the network, system and user enumeration and other penetration testing methodologies that include:

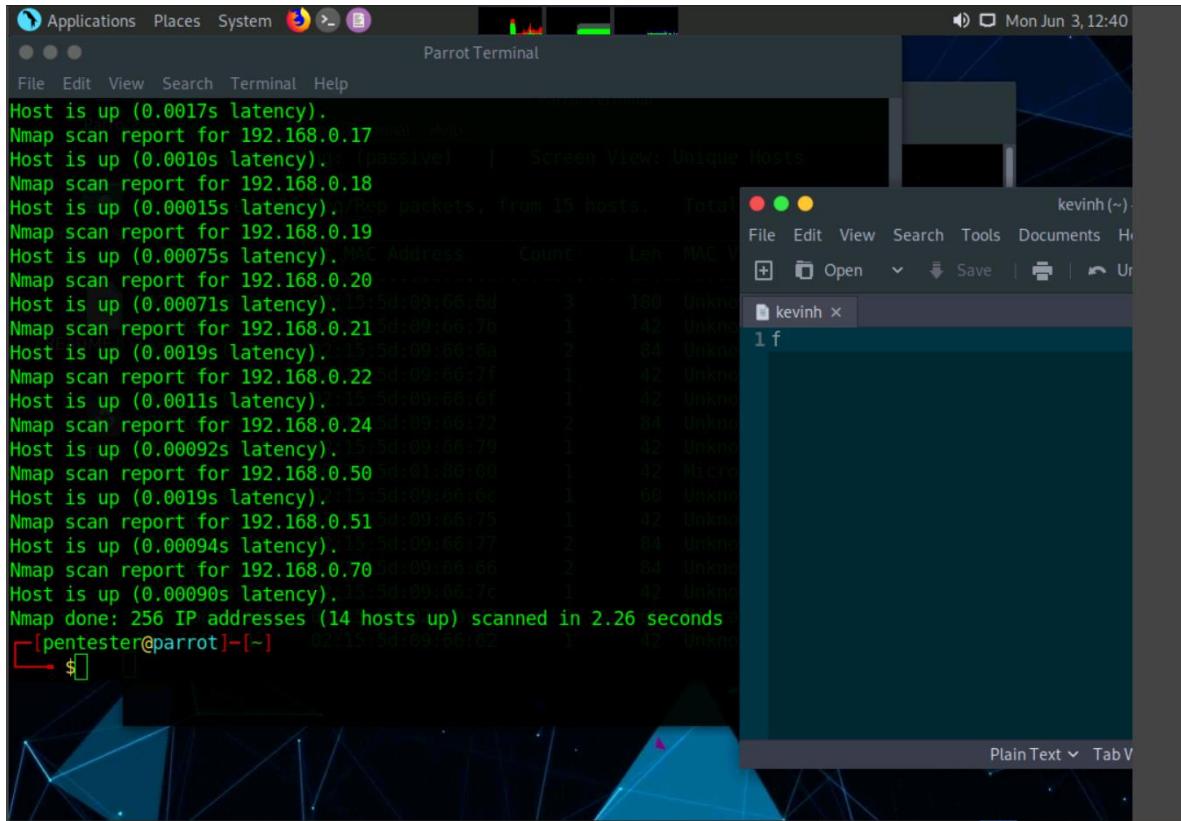
- Service enumeration
- Password audits
- Vulnerability Assessment
- OS pentesting
- Privilege Escalation

Some of the approaches that I have learned about are in relation to reconnaissance where I have leveraged tools such as nmap, ftp, Dmitry and Metasploit to gain insight into how the target database functions. This leads to me learning about the tools involved in the reconnaissance stage, where I was able to observe how tools such as nmap and Dmitry work in relation towards gaining insight into how the database make connections as well as determine whether there are any open ports for a possible attack to be launched against. I also observed how db_nmap can execute lengthy scanning to gain a higher in-depth level of understanding in the target IP Address for future attacks as well as ways in which Metasploit is able to provide a more centralized approach in relation to reconnaissance. From being able to map out the database all the way to obtaining the OS and its version.

Exercise 1: Scanning with the Tool Netdiscover

1.1 OUTPUT SCREENSHOTS

Exercise 1, Step 6: In a new terminal window, enter a Nmap ping sweep to generate traffic. To do a ping sweep, type nmap -sn 192.168.0.0/24 and press Enter.



The screenshot shows a Parrot OS desktop environment. In the foreground, a terminal window titled "Parrot Terminal" displays the output of a Nmap ping sweep. The command entered was "nmap -sn 192.168.0.0/24". The output shows 14 hosts up on the network. In the background, there is a file manager window titled "kevinh (-)" showing a single file named "1 f". The desktop has a dark theme with a blue and green geometric pattern as the wallpaper.

```
Host is up (0.0017s latency).
Nmap scan report for 192.168.0.17
Host is up (0.0010s latency). (passive) | Screen View: Unique Hosts
Nmap scan report for 192.168.0.18
Host is up (0.00015s latency). /Rep packets, from 15 hosts. Total
Nmap scan report for 192.168.0.19
Host is up (0.00075s latency). MAC Address Count Len MAC V
Nmap scan report for 192.168.0.20
Host is up (0.00071s latency). 15:5d:09:66:6d 3 180 Unknown
Nmap scan report for 192.168.0.21 5d:09:66:7b 1 42 Unknown
Host is up (0.0019s latency). 15:5d:09:66:6a 2 84 Unknown
Nmap scan report for 192.168.0.22 5d:09:66:7f 1 42 Unknown
Host is up (0.0011s latency). 15:5d:09:66:6f 1 42 Unknown
Nmap scan report for 192.168.0.24 5d:09:66:72 2 84 Unknown
Host is up (0.00092s latency). 15:5d:09:66:79 1 42 Unknown
Nmap scan report for 192.168.0.50 5d:01:00:00 1 42 Microsoft
Host is up (0.0019s latency). 15:5d:09:66:6c 1 69 Unknown
Nmap scan report for 192.168.0.51 5d:09:66:75 1 42 Unknown
Host is up (0.00094s latency). 15:5d:09:66:77 2 84 Unknown
Nmap scan report for 192.168.0.70 5d:09:66:66 2 84 Unknown
Host is up (0.00090s latency). 15:5d:09:66:7c 1 42 Unknown
Nmap done: 256 IP addresses (14 hosts up) scanned in 2.26 seconds
[pentester@parrot] [-] 02:15:5d:09:66:62 1 42 Unknown
```

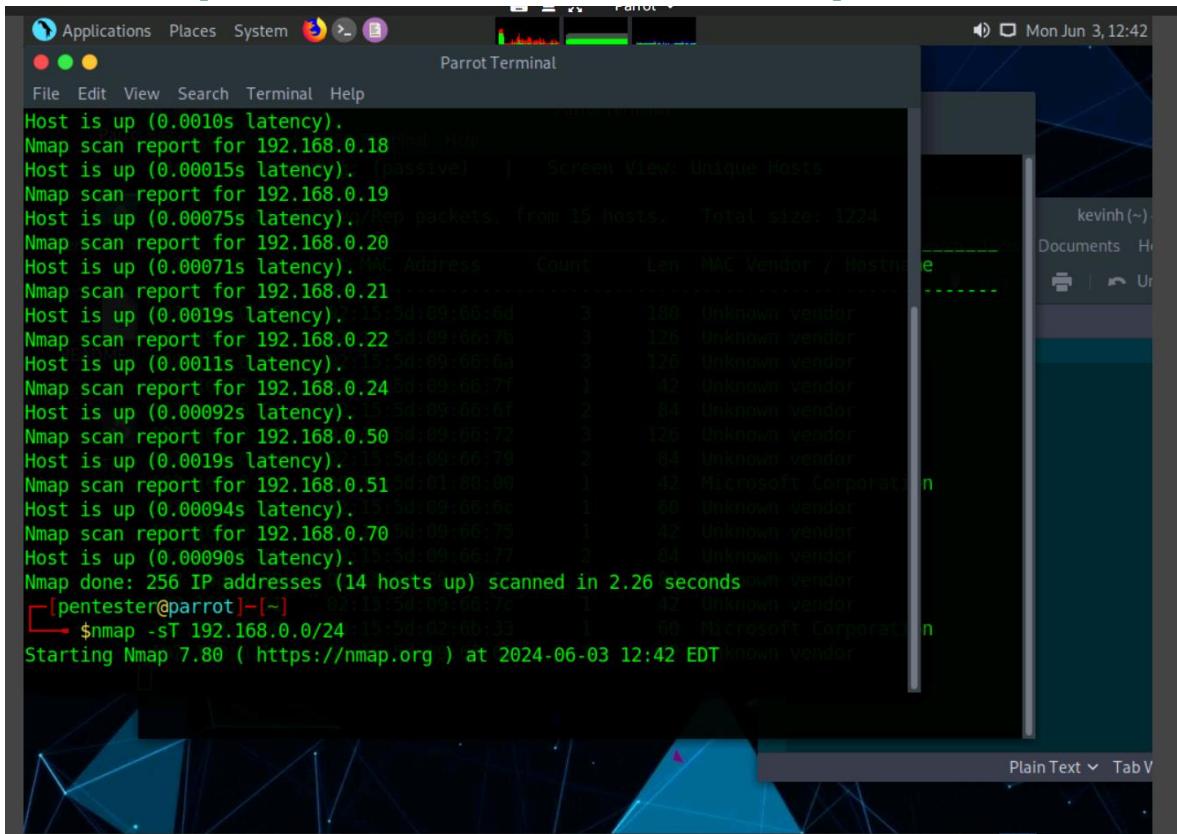
Exercise 1, Step 7: Switch back to netdiscover window to view the output

The screenshot shows a Parrot OS desktop environment with a terminal window titled "Parrot Terminal". The terminal displays the results of an nmap scan. The output includes:

```
Host is up 192.168.0.15 (IP: 075s latency)
Nmap scan report for 192.168.0.18
Host is up Currently scanning: (passive) | Screen View: Unique Hosts
Nmap scan report for 192.168.0.19
Host is up 23 Captured ARP Req/Rep packets, from 15 hosts. Total size: 1056
Nmap scan report for 192.168.0.20
Host is up IP: 075s latency At MAC Address Count Len MAC Vendor / Hostname
Nmap scan report for 192.168.0.21
Host is up 192.168.0.15 ency 02:15:5d:09:66:6d 3 180 Unknown vendor
Nmap scan report for 192.19.19.23
Host is up 192.168.0.7 ency 02:15:5d:09:66:6a 2 84 Unknown vendor
Nmap scan report for 192.168.0.10
Host is up 192.168.0.9 ency 02:15:5d:09:66:7f 1 42 Unknown vendor
Nmap scan report for 192.168.0.17
Host is up 192.168.0.22 ency 02:15:5d:09:66:72 2 84 Unknown vendor
Nmap scan report for 192.168.0.24
Host is up 192.168.0.21 ency 02:15:5d:09:66:6c 1 60 Unknown vendor
Nmap scan report for 192.168.0.70
Host is up 192.168.0.19 ency 02:15:5d:09:66:77 2 84 Unknown vendor
Nmap scan report for 192.168.0.20
Host is up 192.168.0.51 ency 02:15:5d:09:66:7c 1 42 Unknown vendor
Nmap done: 192.168.0.50 less: 00:15:5d:02:6b:33 scanned 1 2 266 Microsoft Corporation
[pentest] 192.168.0.1 02:15:5d:09:66:62 1 42 Unknown vendor
```

The terminal window has a red border around the command line area. The desktop background features a blue geometric pattern.

Exercise 1, Step 8: At times, some machines may not be discovered due to reasons such as the existence of a firewall or some other filter. At such situation, you may use a Transmission Control Protocol (TCP) scan to confirm the existence of the new machine. To perform a TCP scan, enter the command nmap -sT 192.168.0.0/24.



The screenshot shows a terminal window titled "Parrot Terminal" running on a Parrot OS desktop environment. The terminal displays the results of an Nmap scan for hosts on the network segment 192.168.0.0/24. The output includes:

- Host discovery: "Host is up (0.0010s latency).", "Nmap scan report for 192.168.0.18".
- Host discovery: "Host is up (0.00015s latency). (passive)" for 192.168.0.19.
- Host discovery: "Host is up (0.00075s latency)." for 192.168.0.20.
- Host discovery: "Host is up (0.00071s latency)." for 192.168.0.21.
- Host discovery: "Host is up (0.0019s latency)." for 192.168.0.22.
- Host discovery: "Host is up (0.0011s latency)." for 192.168.0.24.
- Host discovery: "Host is up (0.00092s latency)." for 192.168.0.50.
- Host discovery: "Host is up (0.0019s latency)." for 192.168.0.51.
- Host discovery: "Host is up (0.00094s latency)." for 192.168.0.70.
- Host discovery: "Host is up (0.00090s latency)." for 192.168.0.77.
- Summary: "Nmap done: 256 IP addresses (14 hosts up) scanned in 2.26 seconds".
- Final command: \$nmap -sT 192.168.0.0/24.
- Information: "Starting Nmap 7.80 (https://nmap.org) at 2024-06-03 12:42 EDT".

Exercise 1, Step 11: The targets will be displayed on the screen after some time, as shown in the screenshot.

The screenshot shows a terminal window titled "Parrot Terminal" running on a Parrot OS desktop environment. The window displays the output of a network scanning command. The title bar includes the application menu, system status icons, and the date/time "Mon Jun 3, 12:43". The terminal window has tabs at the top labeled "Currently scanning: Finished!" and "Screen View: Unique Hosts". The main text area shows the following message: "0 Captured ARP Req/Rep packets, from 0 hosts. Total size: 0 bytes". Below this, a table lists network interface statistics:

IP	At	MAC Address	Count	Len	MAC Vendor / Hostname
192.168.0.15	02:15:5d:09:66:6d		4	240	Unknown vendor
172.19.19.23	02:15:5d:09:66:70		4	168	Unknown vendor
192.168.0.7	02:15:5d:09:66:6a		4	168	Unknown vendor
192.168.0.10	02:15:5d:09:66:7f		2	84	Unknown vendor
192.168.0.9	02:15:5d:09:66:6f		3	126	Unknown vendor
192.168.0.17	02:15:5d:09:66:72		3	126	Unknown vendor
192.168.0.22	02:15:5d:09:66:79		3	120	Unknown vendor
192.168.0.24	00:15:5d:01:80:00		2	84	Microsoft Corporation
192.168.0.21	02:15:5d:09:66:6c		2	120	Unknown vendor
192.168.0.70	02:15:5d:09:66:75		2	84	Unknown vendor
192.168.0.19	02:15:5d:09:66:77		3	126	Unknown vendor
192.168.0.20	02:15:5d:09:66:66		2	84	Unknown vendor
192.168.0.51	02:15:5d:09:66:7c		2	84	Unknown vendor
192.168.0.50	00:15:5d:02:66:33		1	60	Microsoft Corporation
192.168.0.1	02:15:5d:09:66:62		2	84	Unknown vendor

The terminal window also shows a file browser sidebar on the right with icons for "kevin (~)", "Documents", "Home", and "Trash". The bottom of the terminal window has buttons for "Plain Text" and "Tab View", and a message "Click to switch to 'Workspace 4'".

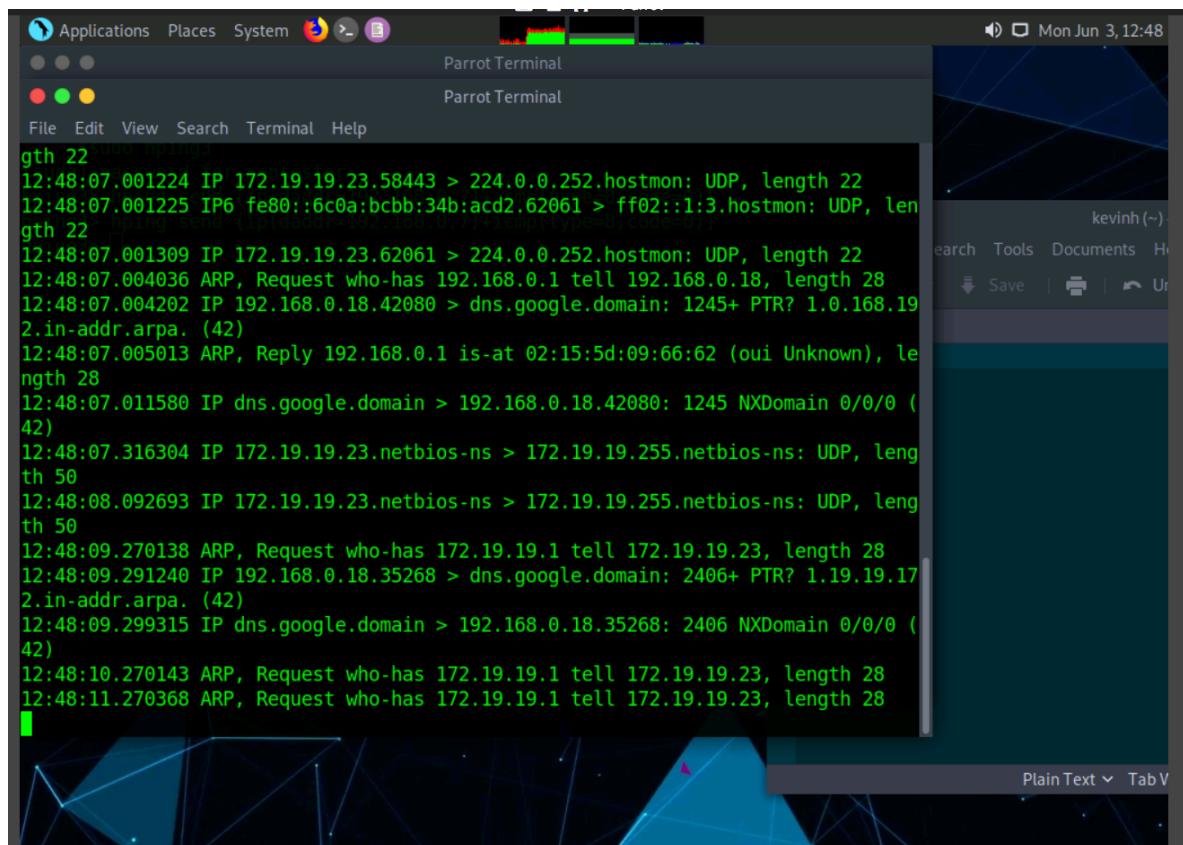
1.2 Questions

NOTE: there are no Questions in Exercise 1

Exercise 2: Scanning and Scripting with hping3

2.1 OUTPUT SCREENSHOTS

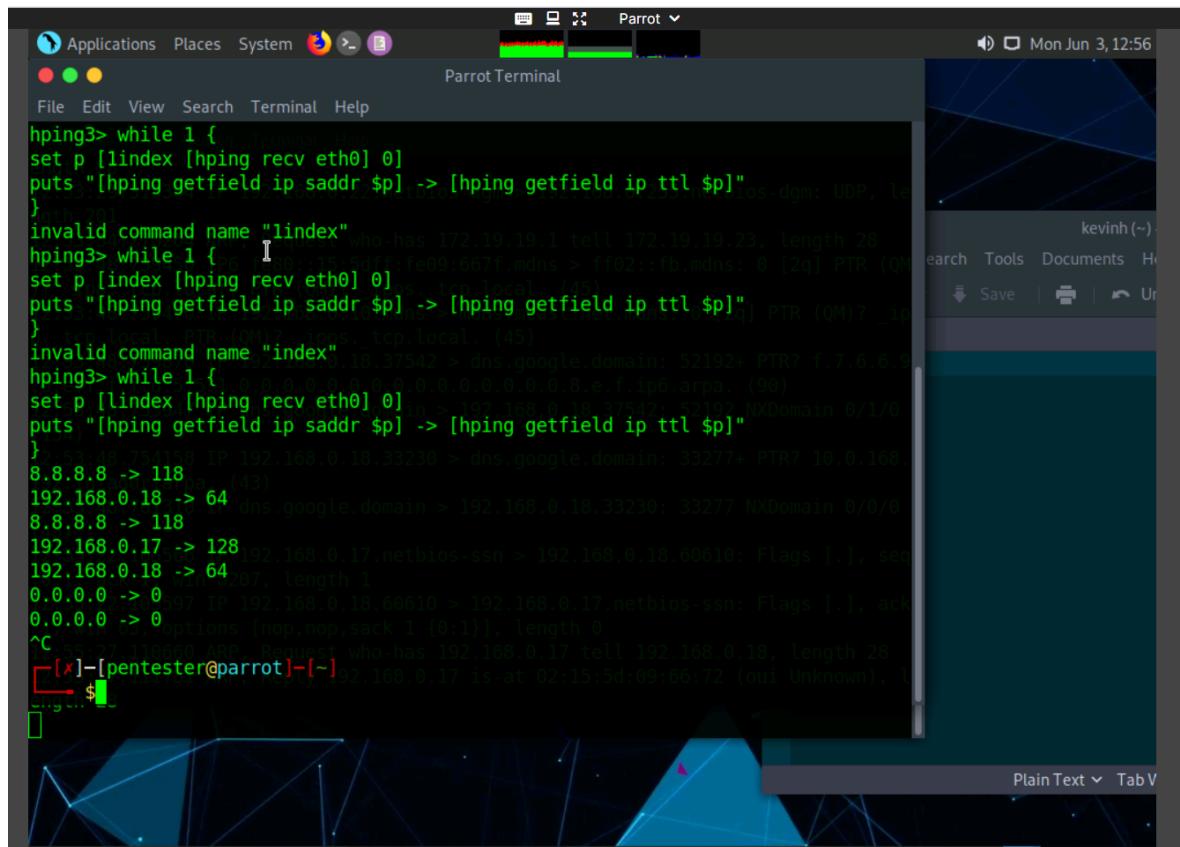
Exercise 2, Step 7: If the ICMP echo request is not visible, verify it by opening a new terminal window; type sudo tcpdump -i eth0 and press Enter. If you are asked you enter a password, type toor and press Enter. This will capture the network traffic. Run the command again and watch the output of the tcpdump command.



```
gth 22
12:48:07.001224 IP 172.19.19.23.58443 > 224.0.0.252.hostmon: UDP, length 22
12:48:07.001225 IP6 fe80::6c0a:bcbb:34b:acd2.62061 > ff02::1:3.hostmon: UDP, len
gth 22
12:48:07.001309 IP 172.19.19.23.62061 > 224.0.0.252.hostmon: UDP, length 22
12:48:07.004036 ARP, Request who-has 192.168.0.1 tell 192.168.0.18, length 28
12:48:07.004202 IP 192.168.0.18.42080 > dns.google.domain: 1245+ PTR? 1.0.168.19
2.in-addr.arpa. (42)
12:48:07.005013 ARP, Reply 192.168.0.1 is-at 02:15:5d:09:66:62 (oui Unknown), le
ngth 28
12:48:07.011580 IP dns.google.domain > 192.168.0.18.42080: 1245 NXDomain 0/0/0 (42)
12:48:07.316304 IP 172.19.19.23.netbios-ns > 172.19.19.255.netbios-ns: UDP, leng
th 50
12:48:08.092693 IP 172.19.19.23.netbios-ns > 172.19.19.255.netbios-ns: UDP, leng
th 50
12:48:09.270138 ARP, Request who-has 172.19.19.1 tell 172.19.19.23, length 28
12:48:09.291240 IP 192.168.0.18.35268 > dns.google.domain: 2406+ PTR? 1.19.19.17
2.in-addr.arpa. (42)
12:48:09.299315 IP dns.google.domain > 192.168.0.18.35268: 2406 NXDomain 0/0/0 (42)
12:48:10.270143 ARP, Request who-has 172.19.19.1 tell 172.19.19.23, length 28
12:48:11.270368 ARP, Request who-has 172.19.19.1 tell 172.19.19.23, length 28
```

Exercise 2, Step 10: Enter a simple loop to receive packets. In the hping3 terminal window, enter the following command:

```
while 1 {  
  
    set p [lindex [hping recv eth0] 0]  
  
    puts "[hping getfield ip saddr $p] -> [hping getfield ip ttl $p]"  
  
}
```

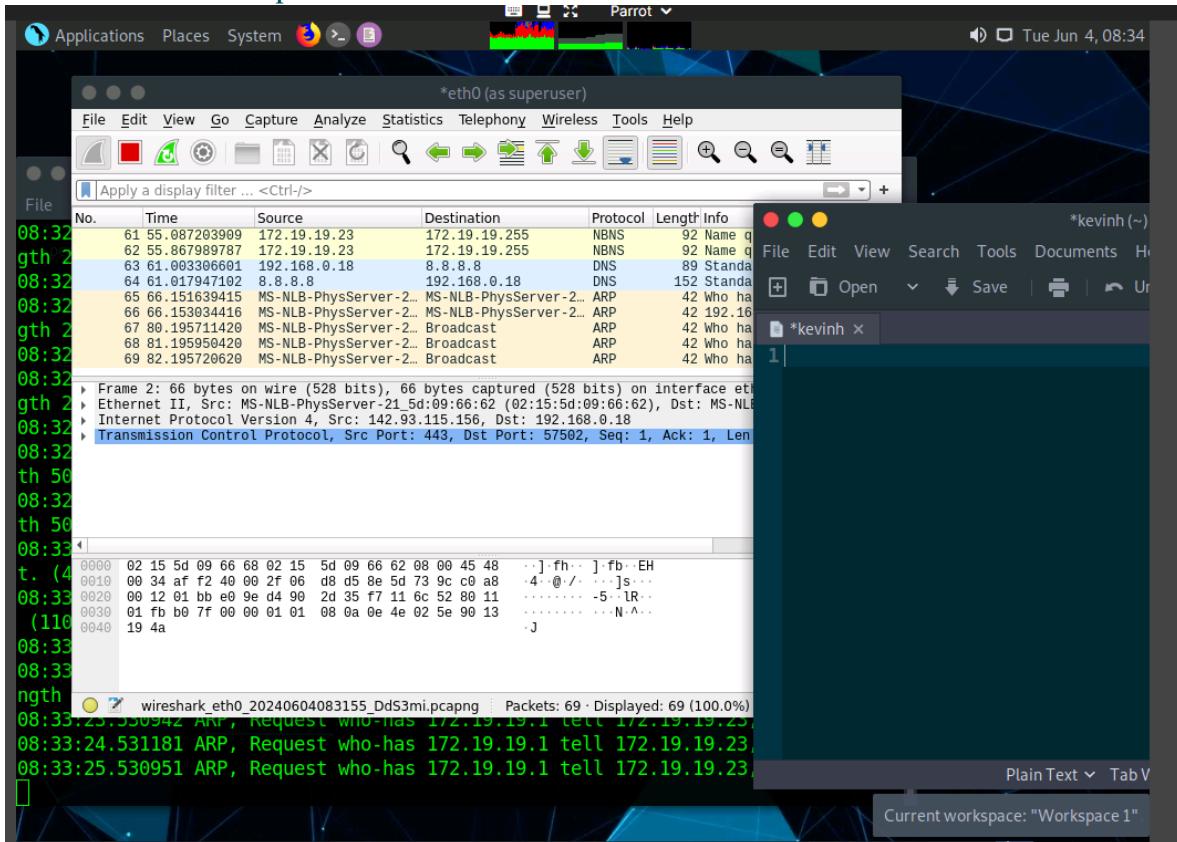


The screenshot shows a terminal window titled "Parrot Terminal" running on a Parrot OS desktop environment. The terminal window has a dark blue background with a light blue header bar. The title bar says "Parrot Terminal". The menu bar includes "File", "Edit", "View", "Search", "Terminal", and "Help". The status bar at the bottom right shows "Mon Jun 3, 12:56". The terminal itself is displaying the following text:

```
hping3> while 1 {  
set p [lindex [hping recv eth0] 0]  
puts "[hping getfield ip saddr $p] -> [hping getfield ip ttl $p]"  
}  
invalid command name "lindex"  
hping3> while 1 {  
set p [index [hping recv eth0] 0]  
puts "[hping getfield ip saddr $p] -> [hping getfield ip ttl $p]"  
}  
invalid command name "index"  
hping3> while 1 {  
set p [lindex [hping recv eth0] 0]  
puts "[hping getfield ip saddr $p] -> [hping getfield ip ttl $p]"  
}  
192.168.0.18->118 IP 192.168.0.18.37542 > dns.google.domain: 52192+ PTR? f.7.6.6.9  
192.168.0.18->64 IP 192.168.0.18.37542 > 192.168.0.18.33230: 33277 NXDomain 0/0/0  
192.168.0.17->128 IP 192.168.0.17.netbios-ssn > 192.168.0.18.60610: Flags [.], seq  
192.168.0.18->64 IP 192.168.0.18.60610 > 192.168.0.17.netbios-ssn: Flags [.], ack  
0.0.0.0->0 0.0.0.0->0 0.0.0.0->0 0.0.0.0->0 0.0.0.0->0 0.0.0.0->0 0.0.0.0->0 0.0.0.0->0  
^C  
[x]-[pentester@parrot]-[~]$
```

Exercise 2, Step 12: The hping3 tool allows users to send messages. Accordingly, send the message as a string. Open a new terminal window, type sudo hping3 -2 -p 500 192.168.0.7 -d 139 -E attack.sig, and press Enter. Type toor in the password field and press Enter. This will send the packet to port 139 from port 500.

Exercise 2, Step 15: The window shows the Internet Security Association and Key Management Protocol (ISAKMP) traffic, as you are using User Datagram Protocol (UDP) port 500. The lower window also shows that the message you specified is carried within the packet.



Exercise 2, Step 16: The message in the packet can also be displayed using tcpdump:
Type sudo tcpdump -i eth0 -nX in the terminal window. If you are asked to enter the password, enter toor.

```
*eth0 (as superuser)
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
File Edit View Terminal Help
1171 packets captured 192.168.0.7 192.168.0.18 TCP 54 10080
1171 packets received by filter 192.168.0.18 TCP 54 11371
0 packets dropped by kernel MS-NLB-PhysServer-2... MS-NLB-PhysServer-2... ARP 157 Solicit
[pentester@parrot]:~[~] $ sudo hping3 --scan known 192.168.0.7 -S -v
[sudo] password for pentester: 
Scanning 192.168.0.7 (192.168.0.7), port known
279 ports to scan, use -V to see all the replies
+-----+
|port| serv name | flags |ttl| id | win | len |
+-----+
 445 microsoft-d: .S..A... 128 39172 8192 44
  80 http     : .S..A... 128 40708 8192 44
 135 epmap    : .S..A... 128 41988 8192 44
 139 netbios-ssn: .S..A... 128 42244 8192 44
  21 ftp      : e.S..A... 128 53252 8192 44
 3389 ms-wbt-serv: .S..A... 128 52996 8192 44
All replies received. Done.
Not responding ports:
[pentester@parrot]:~[~] $ sudo tcpdump -i eth0 -nX
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes

```

Exercise 2, Step 18: Hping3 is a powerful scanning tool that, in the previous example, only showed the known option for the ports listed in /etc/services. Next, specify a range to scan. In the terminal window, enter sudo hping3 --scan '0-3000' 192.168.0.7 -S, as shown in the screenshot

The screenshot shows a Kali Linux desktop environment with several windows open:

- NetworkMiner**: A packet analysis tool window titled "Parrot (as superuser)". It displays network traffic on interface "eth0". The list of captured packets includes:
 - 0x0080: 0000 0000 0000 0000 0000 0000 0000 1100
 - 0x0090: 0021 0000 0000 0000 0000 00e8 0300 0000 .!.....
 - 0x00a0: 0000 0000 0021 0056 0003 0001 0000 0002!..V.
 - 0x00b0: 0032 005c 4d41 494c 534c 4f54 5c42 524f ..2.\MAILSLOT\BRO
 - 0x00c0: 5753 4500 0100 80fc 0a00 5345 5256 4552 WSE.....SERVER
 - 0x00d0: 3230 3139 0000 0000 0000 0a00 0790 0000 2019.....
 - 0x00e0: 0f01 55aa 00 ..U.
- Parrot Terminal**: A terminal window showing the output of the command `sudo hping3 --scan '0-3000' 192.168.0.7 -S`. The output indicates a scan of port 0-3000 on 192.168.0.7, with 3001 ports scanned. The results table shows the following services:

port	serv name	flags	ttl	id	win	len
135	epmap	: S.A...	128	13830	8192	44
139	netbios-ssn	: S.A...	128	14086	8192	44
445	microsoft-d	: S.A...	128	27142	8192	44
21	ftp	: S.A...	128	25863	8192	44
80	http	: S.A...	128	26119	8192	44
- File Viewer**: A window titled "kevin (~)" showing a file with the following content:

```
*kevin (~)
RE
^C
2 packets captured
2 packets received by filter
0 packets dropped by kernel
[pentester@parrot]~[~]
└─ $sudo hping3 --scan '0-3000' 192.168.0.7 -S
Scanning 192.168.0.7 (192.168.0.7), port 0-3000
3001 ports to scan, use -V to see all the replies
|port| serv name | flags | ttl | id | win | len |
+---+-----+-----+-----+-----+-----+
| 135 | epmap | : S.A... | 128 | 13830 | 8192 | 44
| 139 | netbios-ssn | : S.A... | 128 | 14086 | 8192 | 44
| 445 | microsoft-d | : S.A... | 128 | 27142 | 8192 | 44
| 21 | ftp | : S.A... | 128 | 25863 | 8192 | 44
| 80 | http | : S.A... | 128 | 26119 | 8192 | 44

```

Exercise 2, Step 23: The file contents begin to appear in the first terminal as shown in the following screenshot

```
len=128 ip=127.0.0.1 ttl=64 id=44578 icmp_seq=19
rtt=3.2 ms
len=128 ip=127.0.0.1 ttl=64 id=44694 icmp_seq=20
rtt=6.3 ms
len=128 ip=127.0.0.1 ttl=64 id=44763 icmp_seq=21
rtt=2.9 ms
len=128 ip=127.0.0.1 ttl=64 id=44967 icmp_seq=22
rtt=2.9 ms
len=128 ip=127.0.0.1 ttl=64 id=45244 icmp_seq=23
rtt=8.9 ms
len=128 ip=127.0.0.1 rtt=6.2 ms
len=128 ip=127.0.0.1 uidd:x:111:117::/run/uidd:/usr/sbin/nologin
rtt=5.6 ms
len=128 ip=127.0.0.1 debian-tor:x:112:118::/var/lib/tor:/bin/false
rtt=4.8 ms
len=128 ip=127.0.0.1 redsocks:x:113:119::/var/run/redsocks:/usr/sbin/nologin
rtt=5.2 ms
len=128 ip=127.0.0.1 freerad:x:114:122::/etc/freeradius:/usr/sbin/nologin
rtt=5.3 ms
len=128 ip=127.0.0.1 iodine:x:115:65534::/var/run/iodine:/usr/sbin/nologin
rtt=1.2 ms
len=128 ip=127.0.0.1 miredo:x:116:65534::/var/run/miredo:/usr/sbin/nologin
rtt=1.1 ms
len=128 ip=127.0.0.1 dnsmasq:x:117:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
rtt=1.1 ms
len=128 ip=127.0.0.1 redis:x:118:125::/var/lib/redis:/usr/sbin/nologin
rtt=1.1 ms
len=128 ip=127.0.0.1 arpwatch:x:119:126:ARP Watcher,,,:/var/lib/arpwatch:/bin/sh
rtt=1.1 ms
len=128 ip=127.0.0.1 usbmux:x:120:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
rtt=1.1 ms
len=128 ip=127.0.0.1 tcpdump:x:121:127::/nonexistent:/usr/sbin/nologin
rtt=1.1 ms
len=128 ip=127.0.0.1 rtkit:x:122:129:RealtimeKit,,,:/proc:/usr/sbin/nologin
rtt=1.1 ms
len=128 ip=127.0.0.1 sshd:x:123:65534::/run/sshd:/usr/sbin/nologin
rtt=1.1 ms
len=128 ip=127.0.0.1 postgres:x:124:130:PostgreSQL administrator,,,:/var/lib/pgsql:/bin/sh
rtt=1.1 ms
len=128 ip=127.0.0.1 avahi:x:125:132:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/bin/sh
rtt=1.1 ms
len=128 ip=127.0.0.1 stunnel4:x:126:134::/var/run/stunnel4:/usr/sbin/nologin
rtt=1.1 ms
len=128 ip=127.0.0.1 sslh:x:127:136::/nonexistent:/usr/sbin/nologin
rtt=1.1 ms
len=128 ip=127.0.0.1 nm-openvpn:x:128:137:NetworkManager OpenVPN,,,:/var/lib/NetworkManager:/bin/sh
```

2.2 QUESTIONS

The screenshot shows a penetration testing environment with a network graph at the top left and a terminal window below it. The terminal window displays repeated 'login' entries. To the right is a question card for 'Network Penetration Testing Methodology-Internal'.

Question 1: Enter the hping3 command that can be used to find the list of known open ports/services running on the target IP, 192.168.0.7.

Answer: sudo hping3 --scan known 192.168.0.7

Score: Correct

Question 2: In the Parrot machine, use the hping3 tool to scan the target IP, 192.168.0.7. Enter the service running on port 135.

Answer: epmap

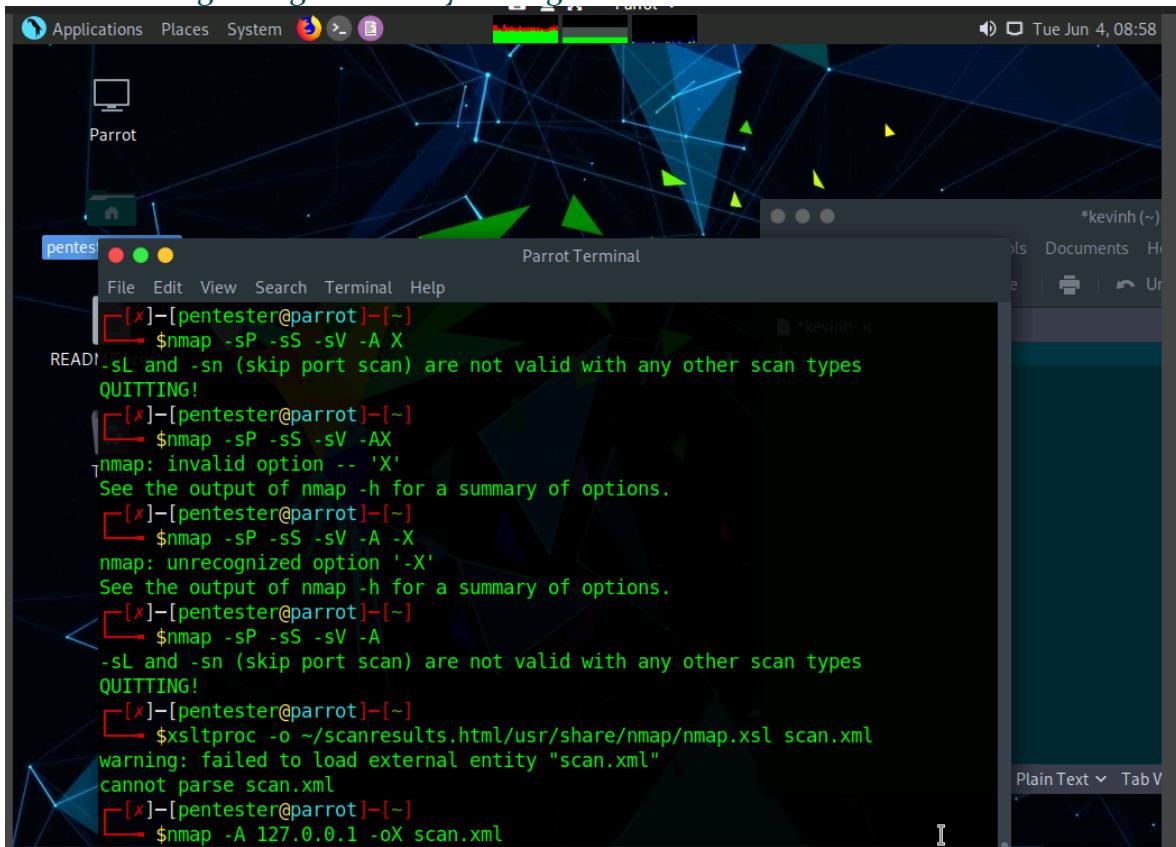
Score: Correct

Progress bar: 2 Hr 25 Min Remaining

Exercise 3: Scanning and Building a Target Database

3.1 OUTPUT SCREENSHOTS

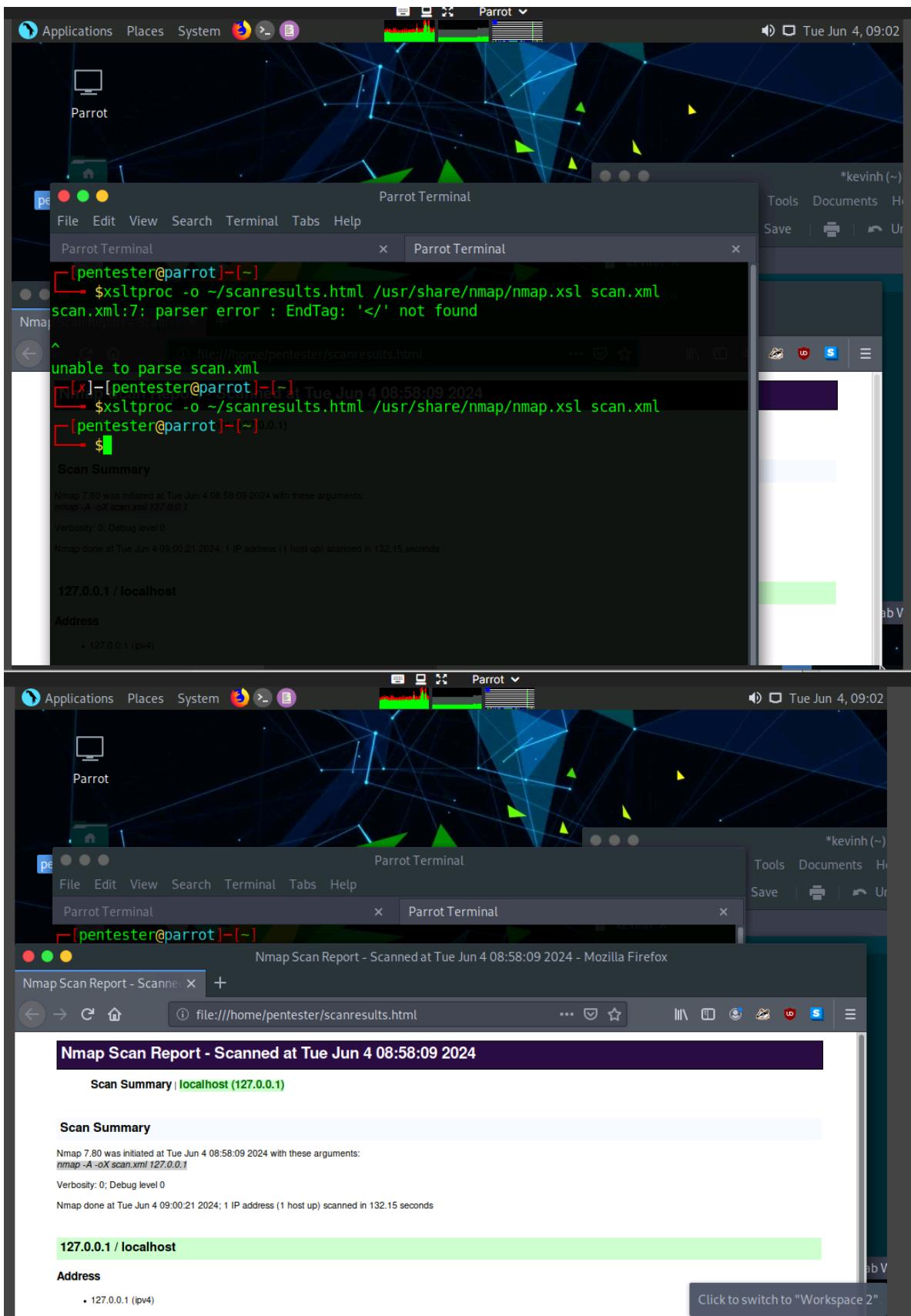
Exercise 3, Step 5: The output can be made into an XML format by adding “X” to the output option. This requires converting the output to HTML. Prior browsers could render the XML format, but this is not reliable, since most browsers no longer allow such rendering owing to security settings.



The screenshot shows a terminal window titled "Parrot Terminal" running on a Parrot OS desktop. The terminal displays the following command sequence:

```
[x]-[pentester@parrot]~$ nmap -sP -sS -sV -A X
nmap: invalid option -- 'X'
See the output of nmap -h for a summary of options.
[x]-[pentester@parrot]~$ nmap -sP -sS -sV -AX
nmap: invalid option -- 'X'
See the output of nmap -h for a summary of options.
[x]-[pentester@parrot]~$ nmap -sP -sS -sV -X
nmap: unrecognized option '-X'
See the output of nmap -h for a summary of options.
[x]-[pentester@parrot]~$ nmap -sP -sS -sV -A
nmap: invalid option -- 'A'
See the output of nmap -h for a summary of options.
[x]-[pentester@parrot]~$ nmap -sP -sS -sV -A X
nmap: invalid option -- 'X'
See the output of nmap -h for a summary of options.
[x]-[pentester@parrot]~$ xsltproc -o ~/scanresults.html /usr/share/nmap/nmap.xsl scan.xml
warning: failed to load external entity "scan.xml"
cannot parse scan.xml
[x]-[pentester@parrot]~$ nmap -A 127.0.0.1 -oX scan.xml
```

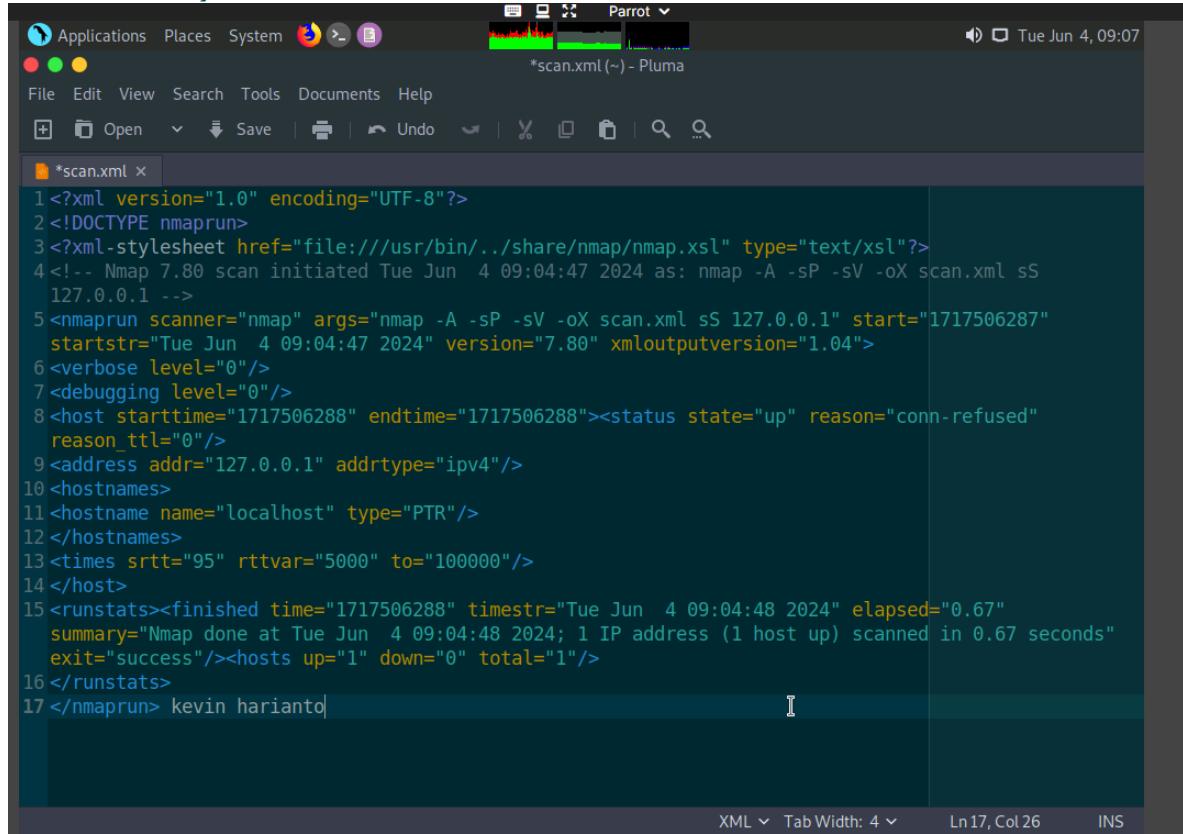
Exercise 3, Step 8: The XML format is a good choice for preparing and creating the database.



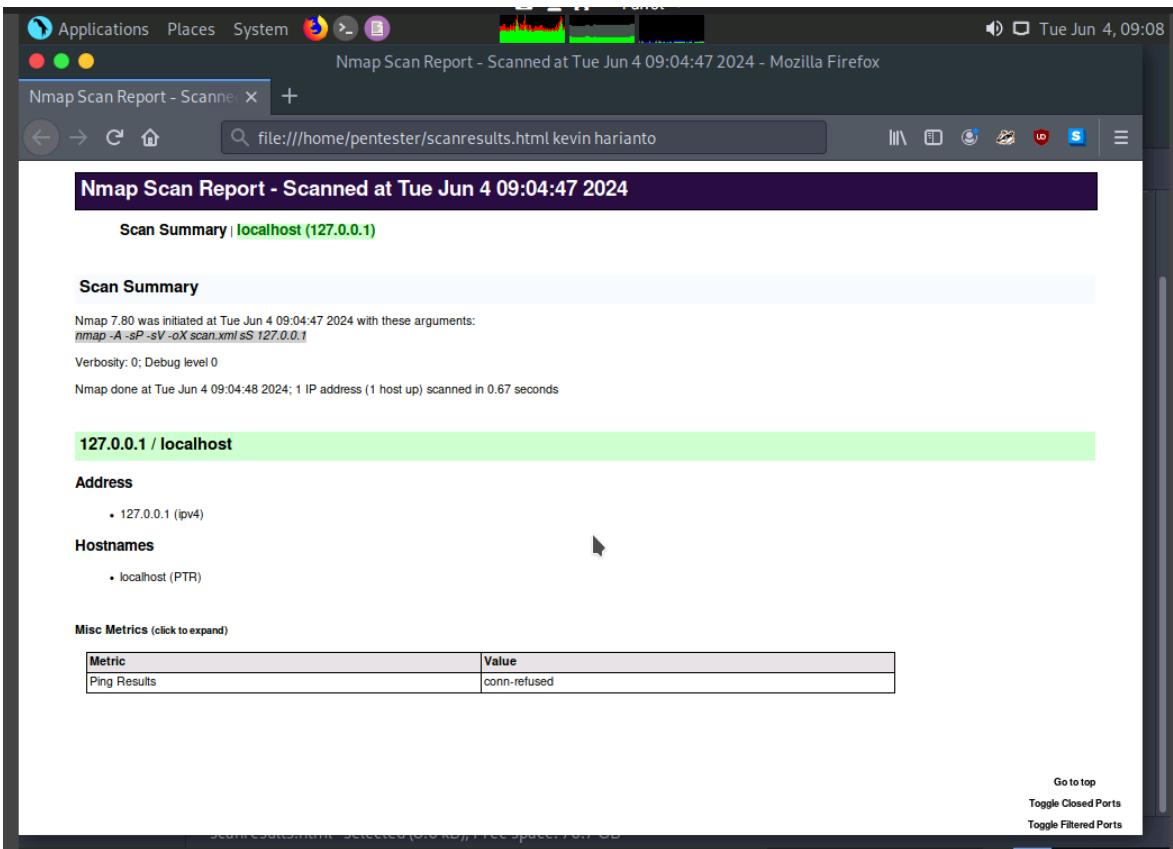
Exercise 3, Step 12: From this point forward, create a target database for every

opportunity, range, or environment.

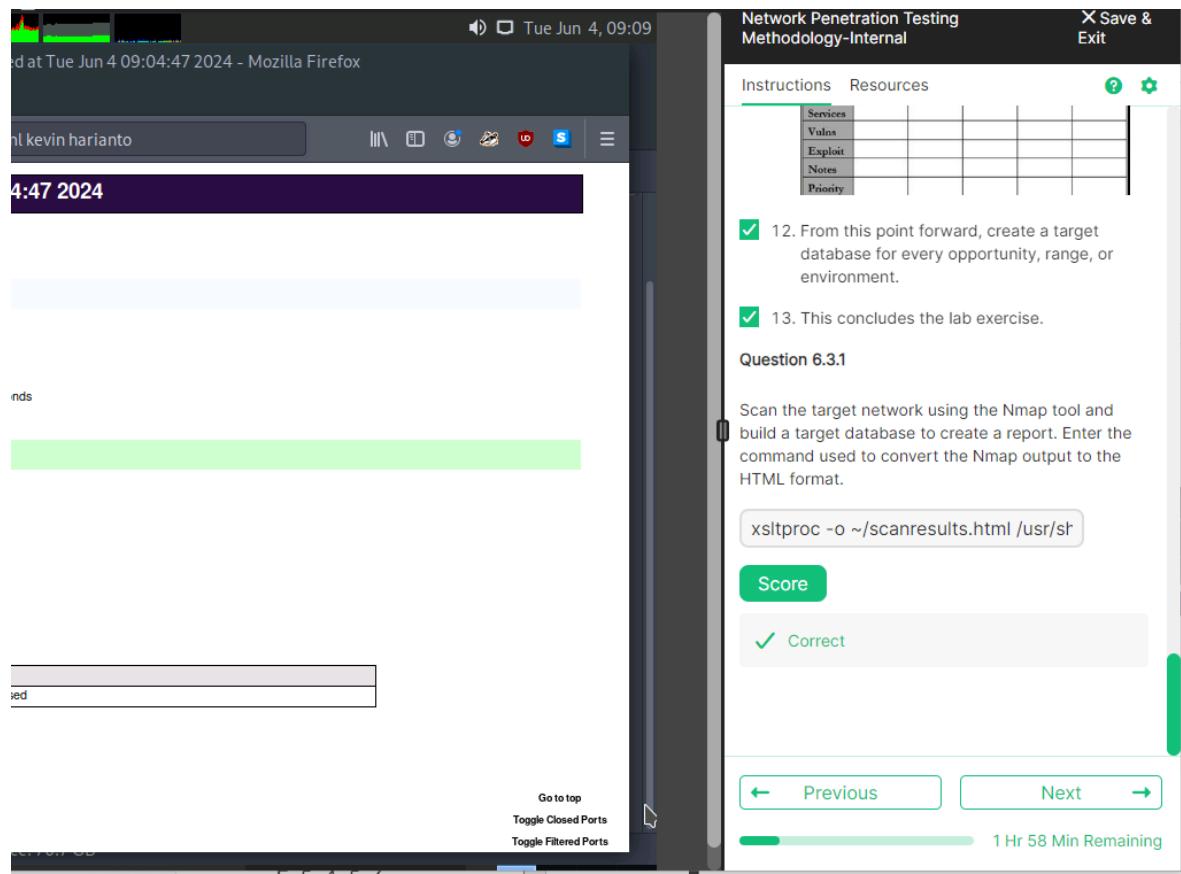
NOTE: Due to the lack of software to display the table in this type of format, despite leveraging Firefox and utilizing multiple Nmap commands, I was unable to structure the data neatly but was still able to obtain it nonetheless.



```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE nmaprun>
3 <?xml-stylesheet href="file:///usr/bin/../share/nmap/nmap.xsl" type="text/xsl"?>
4 <!-- Nmap 7.80 scan initiated Tue Jun 4 09:04:47 2024 as: nmap -A -sP -sV -oX scan.xml sS
127.0.0.1 -->
5 <nmaprun scanner="nmap" args="nmap -A -sP -sV -oX scan.xml sS 127.0.0.1" start="1717506287"
startstr="Tue Jun 4 09:04:47 2024" version="7.80" xmloutputversion="1.04">
6 <verbose level="0"/>
7 <debugging level="0"/>
8 <host starttime="1717506288" endtime="1717506288"><status state="up" reason="conn-refused"
reason_ttl="0"/>
9 <address addr="127.0.0.1" addrtype="ipv4"/>
10 <hostnames>
11 <hostname name="localhost" type="PTR"/>
12 </hostnames>
13 <times srtt="95" rttvar="5000" to="100000"/>
14 </host>
15 <runstats><finished time="1717506288" timestr="Tue Jun 4 09:04:48 2024" elapsed="0.67"
summary="Nmap done at Tue Jun 4 09:04:48 2024; 1 IP address (1 host up) scanned in 0.67 seconds"
exit="success"/><hosts up="1" down="0" total="1"/>
16 </runstats>
17 </nmaprun> kevin harianto|
```



3.2 QUESTIONS



Exercise 4: Using Workspaces and db_nmap

4.1 OUTPUT SCREENSHOTS

Exercise 4, Step 7: To find out the status of the database, type db_status in the terminal window, as shown in the screenshot.

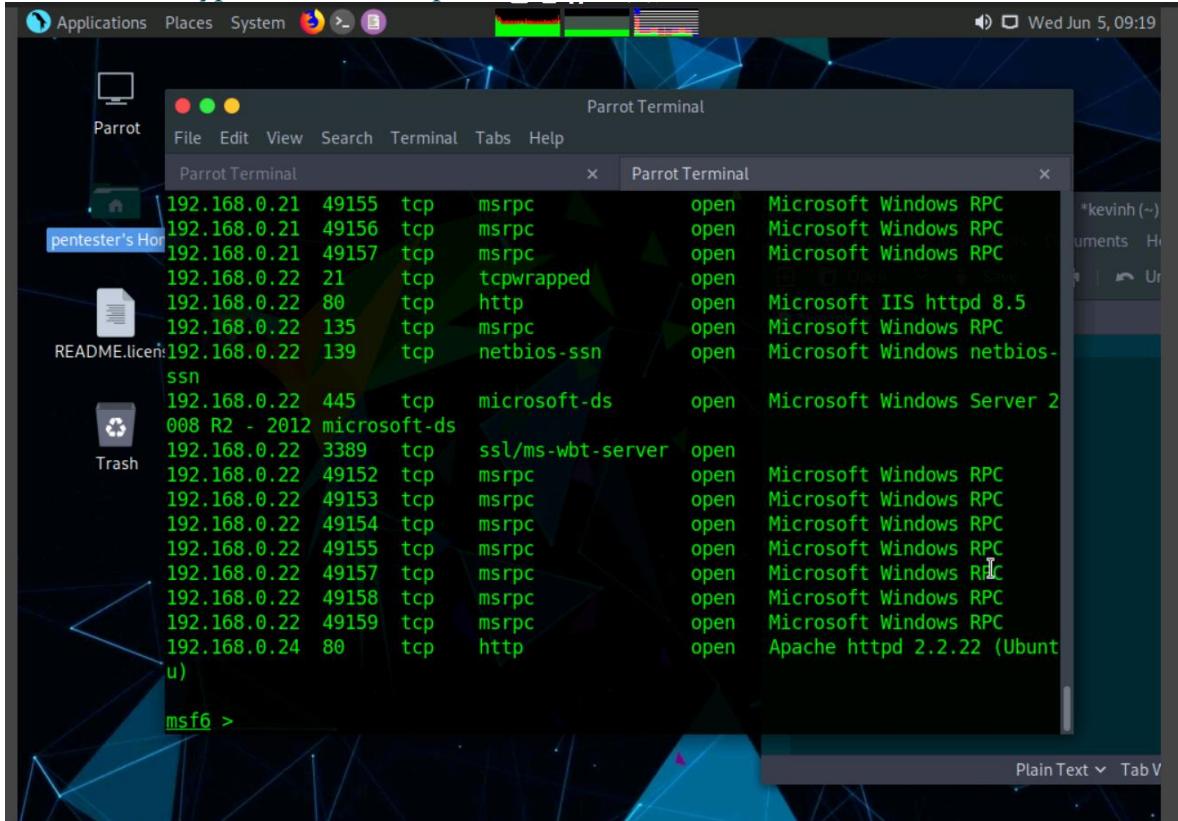
The screenshot shows a Parrot OS desktop environment. In the foreground, a terminal window titled "Parrot Terminal" is open, displaying Metasploit exploit code. The code includes a "README.license" section with a hex dump of the exploit payload, followed by a warning about killing the interrupt handler and a kernel panic message. Below this, it shows a list of exploit modules and a Metasploit tip about enabling HTTP request and response logging. The terminal prompt is "msf6 >". In the background, a file browser window titled "Parrot Terminal" is visible, showing a file named "README.license". The desktop interface includes a dock with icons for Applications, Places, System, and a terminal, and a taskbar at the bottom.

Exercise 4, Step 10: Next, use the tool to conduct the scanning methodology. Enter db_nmap -sP 192.168.0.0/24, as shown in the screenshot.

The screenshot shows a Parrot OS desktop environment. A terminal window titled "Parrot Terminal" is open, displaying the output of an Nmap scan. The output shows multiple hosts being scanned, with details like host status, MAC addresses, and operating systems. The terminal prompt is "msf6 >". The desktop interface is similar to the previous screenshot, with a dock and a taskbar.

Exercise 4, Step 14: You have now conducted the bulk of the scanning methodology.

Sufficient data have been stored in the workspace. To examine the database information, type services and press Enter, as shown in the screenshot.



The screenshot shows a Parrot OS desktop environment. In the foreground, a terminal window titled "Parrot Terminal" is open, displaying a list of network services. The output of the command "services" is as follows:

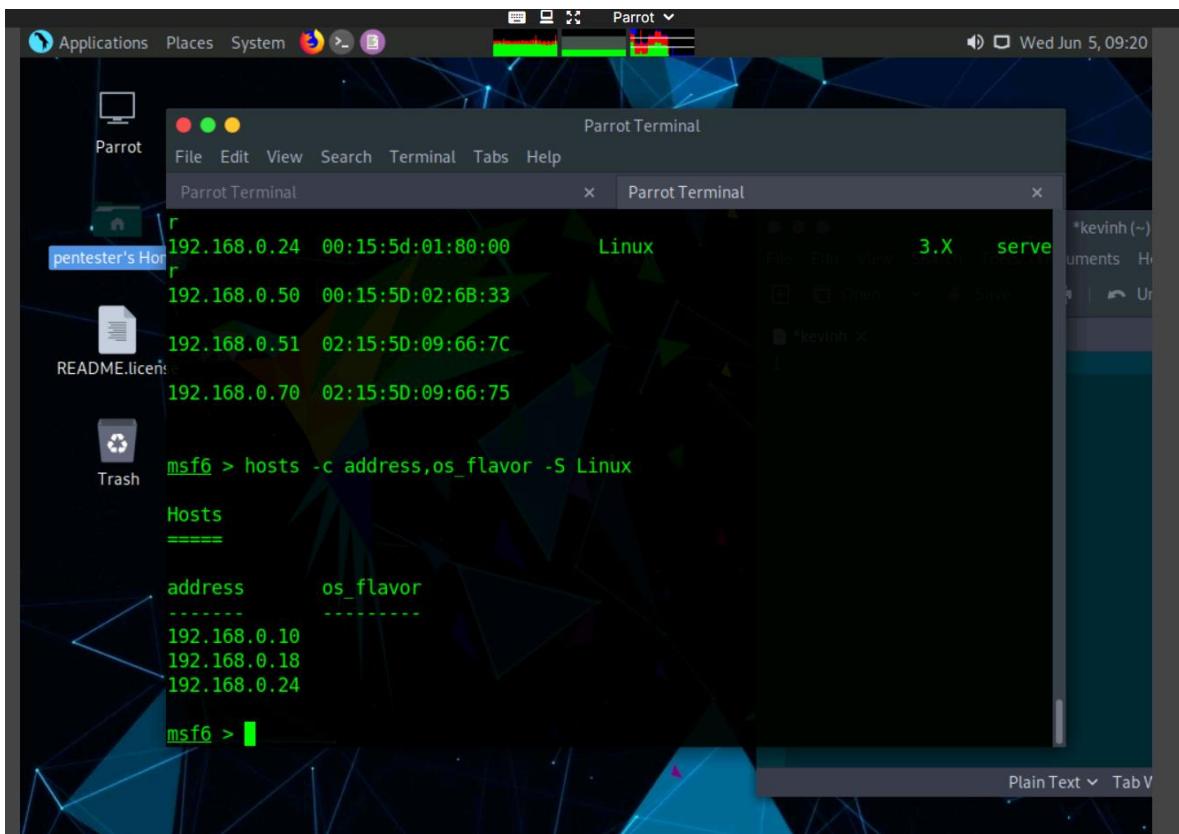
```
192.168.0.21 49155  tcp   msrpc      open  Microsoft Windows RPC
192.168.0.21 49156  tcp   msrpc      open  Microsoft Windows RPC
192.168.0.21 49157  tcp   msrpc      open  Microsoft Windows RPC
192.168.0.22  21    tcp   tcpwrapped  open
192.168.0.22  80    tcp   http       open  Microsoft IIS httpd 8.5
192.168.0.22  135   tcp   msrpc      open  Microsoft Windows RPC
192.168.0.22  139   tcp   netbios-ssn open  Microsoft Windows netbios-ssn
192.168.0.22  445   tcp   microsoft-ds open  Microsoft Windows Server 2008 R2 - 2012 microsoft-ds
192.168.0.22  3389  tcp   ssl/ms-wbt-server open
192.168.0.22  49152  tcp   msrpc      open  Microsoft Windows RPC
192.168.0.22  49153  tcp   msrpc      open  Microsoft Windows RPC
192.168.0.22  49154  tcp   msrpc      open  Microsoft Windows RPC
192.168.0.22  49155  tcp   msrpc      open  Microsoft Windows RPC
192.168.0.22  49157  tcp   msrpc      open  Microsoft Windows RPC
192.168.0.22  49158  tcp   msrpc      open  Microsoft Windows RPC
192.168.0.22  49159  tcp   msrpc      open  Microsoft Windows RPC
192.168.0.24  80    tcp   http       open  Apache httpd 2.2.22 (Ubuntu)
msf6 >
```

Exercise 4, Step 16: Next, examine the database list of hosts; type hosts, as shown in the screenshot

192.168.0.17 02:15:5d:09:66:72
192.168.0.18
192.168.0.19 02:15:5d:09:66:77
192.168.0.20 02:15:5d:09:66:66
192.168.0.21 02:15:5d:09:66:6c
192.168.0.22 02:15:5d:09:66:79
192.168.0.24 00:15:5d:01:80:00
192.168.0.50 00:15:5d:02:68:33
192.168.0.51 02:15:5d:09:66:7C
192.168.0.70 02:15:5d:09:66:75

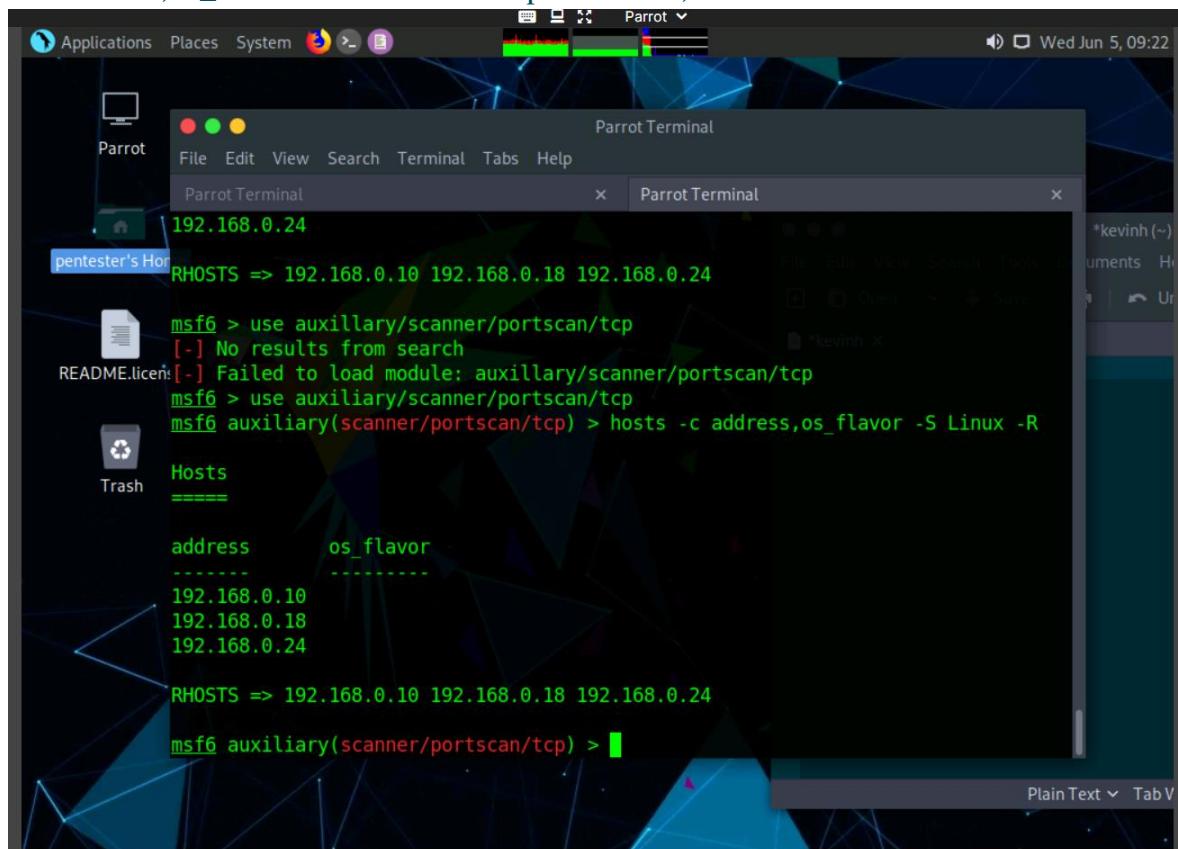
Windows Longhorn device 2.6.X serve client
Windows 7 serve client
Windows 2012 serve client
Windows 7 serve client
Windows 2012 serve client
Linux 3.X serve

Exercise 4, Step 20: 'Note that you can also search all entries for a specific target. If you wish to find only Linux-based machines from the scan, use the "-S" option. This option can be combined with our previous example to fine-tune the results. Type hosts -c address,os_flavor -S Linux.'



```
Parrot Terminal
File Edit View Search Terminal Tabs Help
Parrot Terminal x Parrot Terminal x
pentester's Hosts
192.168.0.24 00:15:5d:01:80:00
192.168.0.50 00:15:5D:02:6B:33
192.168.0.51 02:15:5D:09:66:7C
192.168.0.70 02:15:5D:09:66:75
READMe.license
msf6 > hosts -c address,os_flavor -S Linux
Hosts
=====
address      os_flavor
-----
192.168.0.10
192.168.0.18
192.168.0.24
msf6 >
```

Exercise 4, Step 22: Input the data into the scanner by using the R option; type hosts -c address,os_flavor -S Linux -R and press Enter, as shown in the screenshot.



```
Parrot Terminal
File Edit View Search Terminal Tabs Help
Parrot Terminal x Parrot Terminal x
192.168.0.24
RHOSTS => 192.168.0.10 192.168.0.18 192.168.0.24
READMe.license
msf6 > use auxillary/scanner/portscan/tcp
[-] No results from search
[-] Failed to load module: auxillary/scanner/portscan/tcp
msf6 > use auxiliary/scanner/portscan/tcp
msf6 auxiliary(scanner/portscan/tcp) > hosts -c address,os_flavor -S Linux -R
Hosts
=====
address      os_flavor
-----
192.168.0.10
192.168.0.18
192.168.0.24
RHOSTS => 192.168.0.10 192.168.0.18 192.168.0.24
msf6 auxiliary(scanner/portscan/tcp) >
```

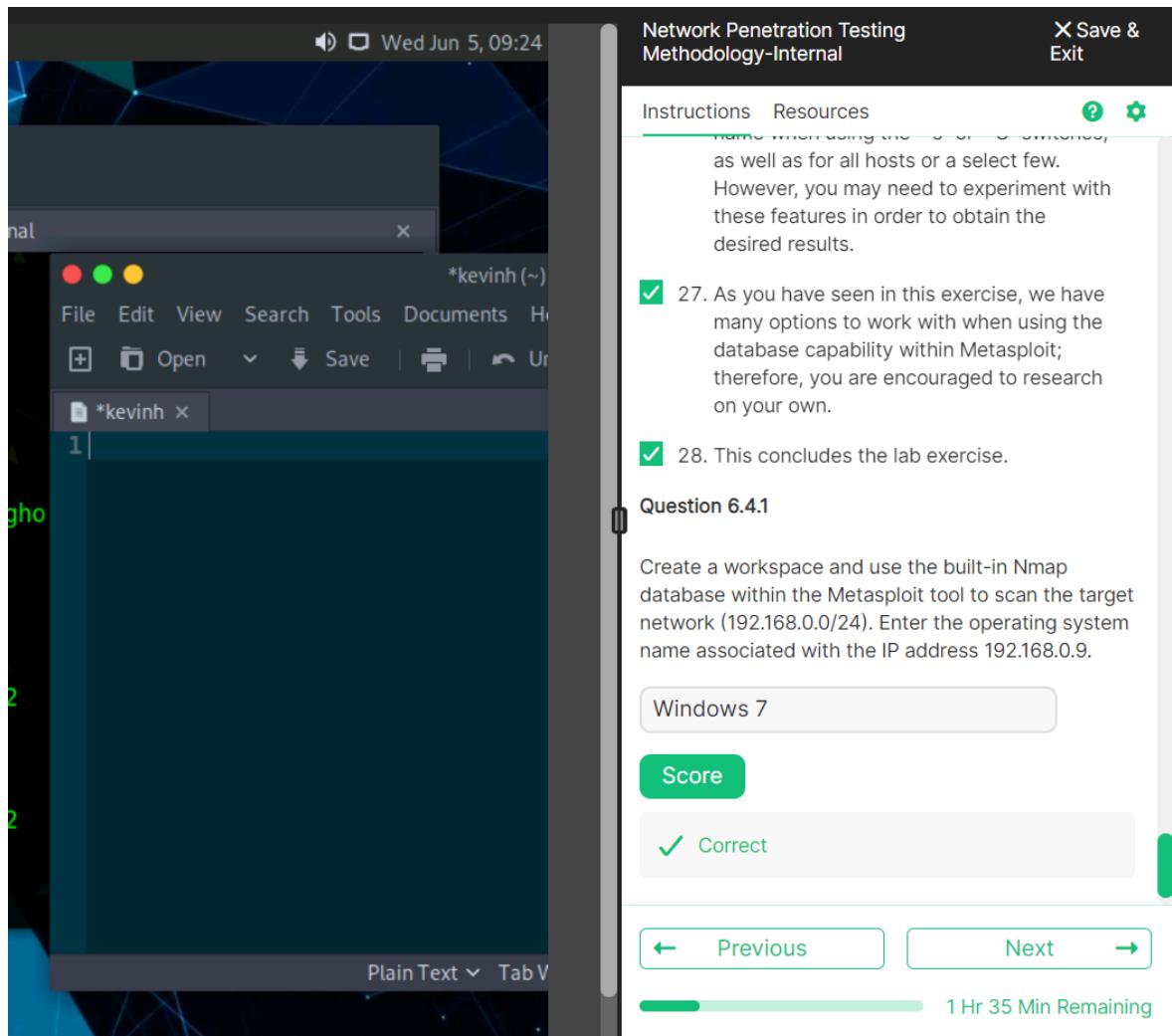
Exercise 4, Step 24: Once you are ready, type run and press Enter. The scan will be conducted against the target added to the database, as shown in the screenshot.

The screenshot shows a Kali Linux desktop environment with a terminal window open in the Parrot Terminal application. The terminal displays the following Metasploit command-line session:

```
[+] No results from search
[-] Failed to load module: auxillary/scanner/portscan/tcp
msf6 > use auxiliary/scanner/portscan/tcp
msf6 auxiliary(scanner/portscan/tcp) > hosts -c address,os_flavor -S Linux -R
[*] kevinh (~) u...ument...ur

[+] 192.168.0.10:          - 192.168.0.10:22 - TCP OPEN
[+] 192.168.0.10:          - 192.168.0.10:21 - TCP OPEN
[+] 192.168.0.10:          - 192.168.0.10:80 - TCP OPEN
```

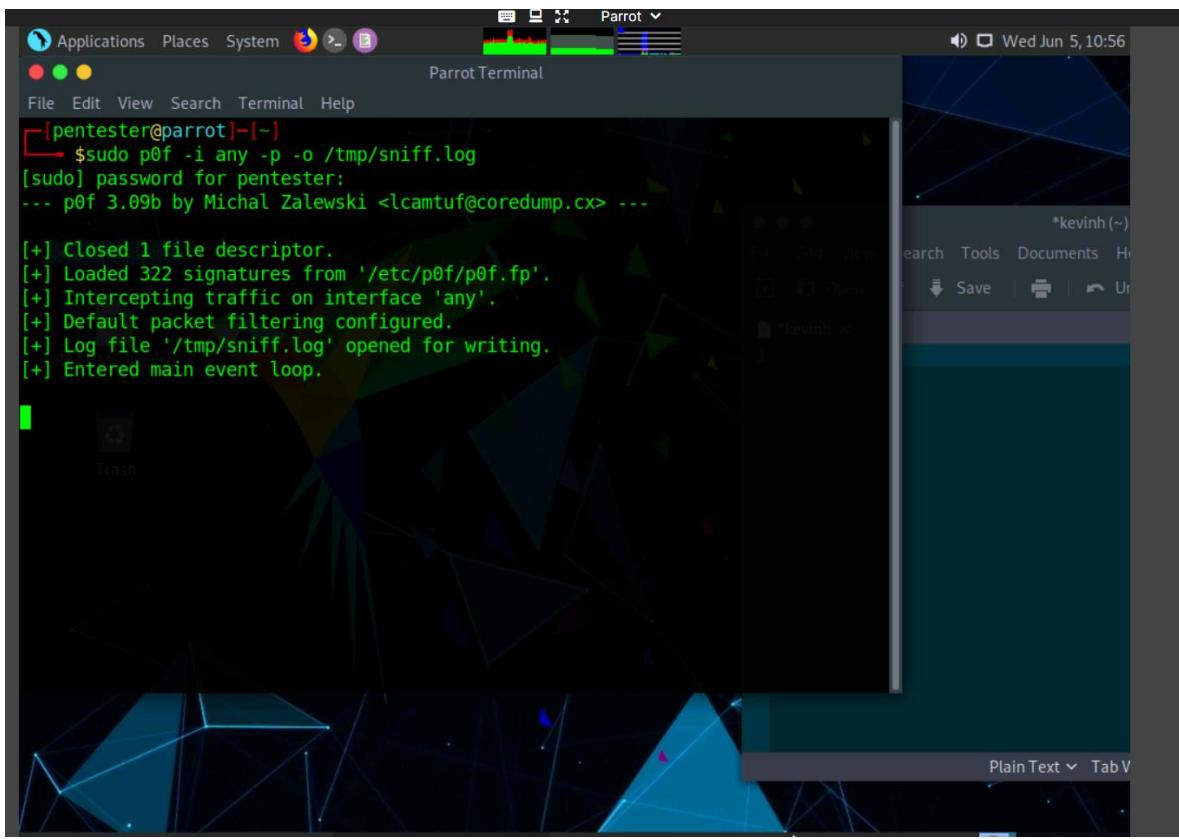
4.2 QUESTIONS



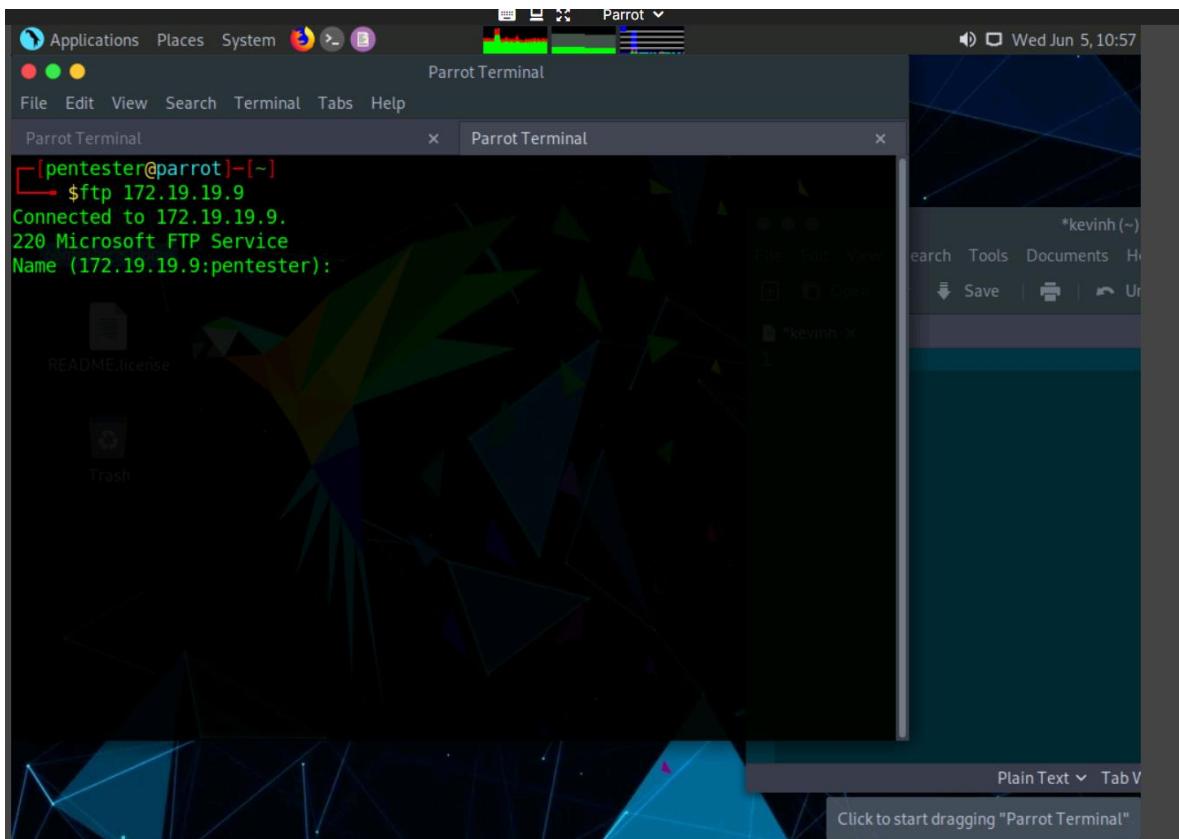
Exercise 5: Performing Passive OS Fingerprinting to Obtain Remote Operating System Information

5.1 OUTPUT SCREENSHOTS

Exercise 5, Step 4: Now, launch a command line terminal, type sudo pof -i any -p o /tmp/sniff.log and press Enter. Type toor and press Enter when prompted for password. pof begins to listen on all the interfaces of Parrot, and whenever it captures a packet, it decodes the header information and guesses the operating system.



Exercise 5, Step 5: Now, launch another command line terminal, type `ftp 172.19.19.9` and press Enter. This will ask you to enter login credentials. By doing so, the client i.e., Parrot machine will send the request and the machine hosting the FTP server will respond to the query.



Exercise 5, Step 6: Switch to the command line terminal where pof is running and scroll up the window. You will observe that pof has analyzed all the requests and responses and decoded them to display information such as OS, raw signature and raw mtu. In this lab, pof identified the operating system as Windows 7 or 8 (or its equivalent). Scroll down the window to view the header information of each packet decoded by the tool.

The screenshot shows a Parrot OS desktop environment. In the foreground, a terminal window titled "Parrot Terminal" displays the following configuration script:

```
| link      = Ethernet or modem
| raw_mtu   = 1500
|
----[ kevintester's Home ]-----
|[ 172.19.19.18/49692 -> 172.19.19.9/21 (syn+ack) ]-
| server    = 172.19.19.9/21
| os        = Windows 7 or 8
| dist      = 0
| params    = none
| raw_sig   = 4:128+0:0:1460:8192,8:mss,nop,ws,sok,ts:df,id+:0
|
----[ Trash ]-----
|[ 172.19.19.18/49692 -> 172.19.19.9/21 (mtu) ]-
| server    = 172.19.19.9/21
| link      = Ethernet or modem
| raw_mtu   = 1500
|
----
```

In the background, a file browser window titled "kevintester's Home" is open, showing a list of files and folders. The desktop background features a dark, geometric abstract pattern.

5.2 QUESTIONS

The screenshot displays a terminal window on the left and a Network Penetration Testing Methodology-Internal interface on the right.

Terminal Output:

```

root@kevinh:~# p0f -i eth0
[...]
processor: Intel(R) Dual Band Wireless-AC 7265
os: Windows 7 or 8
dist: 0
params: 0
raw_ipg: 4.125ms(0-1600,0-1937,0-0x0,0x0,0x0,0x0)
trust: 0
[...]
[!] 172.19.19.18/56976 -> 172.19.19.9/21 [syn+ack] [
server = 172.19.19.9/21
Link = Ethernet or wireless
raw_ipg = 1500
[...]

```

Network Penetration Testing Methodology-Internal Interface:

- Instructions** tab is selected.
- Resources** tab is available.
- Score:** Score is displayed.
- Feedback:** "Correct" is shown with a green checkmark.
- Time Remaining:** 1 Hr 31 Min Remaining.

Exercise 6: OS Fingerprinting with Nmap

6.1 OUTPUT SCREENSHOTS

Exercise 6, Step 6: In a terminal window, type sudo nmap -O 192.168.o.X, replacing the “X” with the required IP address number from your target database, as shown in the screenshot

A screenshot of a Parrot OS desktop environment. The terminal window, titled "Parrot Terminal", shows the following session:

```
[pentester@parrot] ~
└─ $ nmap -O 192.168.0.7
TCP/IP fingerprinting (for OS scan) requires root privileges.
QUITTING!
[x]~[pentester@parrot] ~
└─ $ sudo nmap -O 192.168.0.7
[sudo] password for pentester:
Starting Nmap 7.80 ( https://nmap.org ) at 2024-06-05 11:01 EDT
```

The desktop background features a dark, geometric abstract design. A file browser window is visible in the background, showing a file named "README.license" in the "kevin (~)" folder.

6.2 QUESTIONS

The screenshot shows a terminal window titled "Network Penetration Testing Methodology-Internal". The terminal background is dark blue with a network graph. The title bar includes a speaker icon, a window control button, the date "Wed Jun 5, 11:01", and buttons for "Save & Exit" and settings.

The terminal window contains the following content:

- Instructions** tab is selected.
- Resources** tab is available.
- A list of numbered steps:
 9. Note that Nmap is noisy and uses many packets to detect the OS; if stealth is a requirement, this may not be the best tool.
 10. It is imperative to use multiple tools—at least two to validate and verify the information that a tool discovers.
 11. Once you fully understand the process, you may continue to review and evaluate the tools that you need to be a professional security tester.
 12. This concludes the lab exercise.
- Question 6.6.1**
- Text: Perform passive OS fingerprinting using the Nmap tool and enter the operating system name associated with the IP address 192.168.0.7.
- An input field contains the text "Windows 7".
- A green "Score" button is present.
- A feedback message: "Correct" with a checkmark.
- Navigation buttons: "Previous" and "Next".
- A progress bar at the bottom indicates "1 Hr 28 Min Remaining".

Exercise 7: Scanning with Dmitry

7.1 OUTPUT SCREENSHOTS

Exercise 7, Step 4: Enter `sudo nmap -sn 192.168.0.0/24`. Type `toor` if you are asked to enter the password. An example of a partial output from the command is shown in the screenshot.

```
MAC Address: 02:15:5D:09:66:66 (Unknown)
Nmap scan report for 192.168.0.21
Host is up (0.0016s latency).
MAC Address: 02:15:5D:09:66:6C (Unknown)
Nmap scan report for 192.168.0.22
Host is up (0.00031s latency).
MAC Address: 02:15:5D:09:66:79 (Unknown)
Nmap scan report for 192.168.0.24
Host is up (0.00062s latency).
MAC Address: 00:15:5D:01:80:00 (Microsoft)
Nmap scan report for 192.168.0.50
Host is up (0.00094s latency).
MAC Address: 00:15:5D:02:6B:33 (Microsoft)
Nmap scan report for 192.168.0.51
Host is up (0.00037s latency).
MAC Address: 02:15:5D:09:66:7C (Unknown)
Nmap scan report for 192.168.0.70
Host is up (0.00065s latency).
MAC Address: 02:15:5D:09:66:75 (Unknown)
Nmap scan report for 192.168.0.18
Host is up.
Nmap done: 256 IP addresses (15 hosts up) scanned in 1.62 seconds
[pentester@parrot]~$
```

Exercise 7, Step 6: Open a terminal window and enter dmitry -pf 192.168.0.22, as shown in the screenshot

```
Deepmagic Information Gathering Tool
"There be some deep magic going on"

ERROR: Unable to locate Host Name for 192.168.0.22
Continuing with limited modules
HostIP:192.168.0.22
HostName:

Gathered TCP Port information for 192.168.0.22
-----
Port      State
21/tcp    open
80/tcp    open
135/tcp   open
139/tcp   open

Portscan Finished: Scanned 150 ports, 145 ports were in state closed

All scans completed, exiting
[pentester@parrot]~$
```

Exercise 7, Step 8: In the terminal window, enter dmitry -pb 192.168.0.22, as shown in the screenshot

```
$ dmitry -pb 192.168.0.22
Deepmagic Information Gathering Tool
"There be some deep magic going on"

ERROR: Unable to locate Host Name for 192.168.0.22
Continuing with limited modules
HostIP:192.168.0.22
HostName:
Gathered TCP Port information for 192.168.0.22

Port      State
21/tcp    open
>>
80/tcp    open

Portscan Finished: Scanned 150 ports, 147 ports were in state closed

All scans completed, exiting
[pentester@parrot]~$
```

7.2 QUESTIONS

The screenshot shows a terminal window displaying a port scan result from the `nmap` tool. The output includes:

```
Port      State    Service
22/tcp    open     ssh
23/tcp    open     telnet
80/tcp    open     http
135/tcp   open     msrpc
139/tcp   open     netbios-ssn
445/tcp   open     microsoft-ds
5935/tcp  open     http-alt
5936/tcp  open     http-alt
5937/tcp  open     http-alt
5938/tcp  open     http-alt
5939/tcp  open     http-alt
5940/tcp  open     http-alt
5941/tcp  open     http-alt
5942/tcp  open     http-alt
5943/tcp  open     http-alt
5944/tcp  open     http-alt
5945/tcp  open     http-alt
5946/tcp  open     http-alt
5947/tcp  open     http-alt
5948/tcp  open     http-alt
5949/tcp  open     http-alt
5950/tcp  open     http-alt
5951/tcp  open     http-alt
5952/tcp  open     http-alt
5953/tcp  open     http-alt
5954/tcp  open     http-alt
5955/tcp  open     http-alt
5956/tcp  open     http-alt
5957/tcp  open     http-alt
5958/tcp  open     http-alt
5959/tcp  open     http-alt
5960/tcp  open     http-alt
5961/tcp  open     http-alt
5962/tcp  open     http-alt
5963/tcp  open     http-alt
5964/tcp  open     http-alt
5965/tcp  open     http-alt
5966/tcp  open     http-alt
5967/tcp  open     http-alt
5968/tcp  open     http-alt
5969/tcp  open     http-alt
5970/tcp  open     http-alt
5971/tcp  open     http-alt
5972/tcp  open     http-alt
5973/tcp  open     http-alt
5974/tcp  open     http-alt
5975/tcp  open     http-alt
5976/tcp  open     http-alt
5977/tcp  open     http-alt
5978/tcp  open     http-alt
5979/tcp  open     http-alt
5980/tcp  open     http-alt
5981/tcp  open     http-alt
5982/tcp  open     http-alt
5983/tcp  open     http-alt
5984/tcp  open     http-alt
5985/tcp  open     http-alt
5986/tcp  open     http-alt
5987/tcp  open     http-alt
5988/tcp  open     http-alt
5989/tcp  open     http-alt
5990/tcp  open     http-alt
5991/tcp  open     http-alt
5992/tcp  open     http-alt
5993/tcp  open     http-alt
5994/tcp  open     http-alt
5995/tcp  open     http-alt
5996/tcp  open     http-alt
5997/tcp  open     http-alt
5998/tcp  open     http-alt
5999/tcp  open     http-alt
6000/tcp  open     http-alt
6001/tcp  open     http-alt
6002/tcp  open     http-alt
6003/tcp  open     http-alt
6004/tcp  open     http-alt
6005/tcp  open     http-alt
6006/tcp  open     http-alt
6007/tcp  open     http-alt
6008/tcp  open     http-alt
6009/tcp  open     http-alt
6010/tcp  open     http-alt
6011/tcp  open     http-alt
6012/tcp  open     http-alt
6013/tcp  open     http-alt
6014/tcp  open     http-alt
6015/tcp  open     http-alt
6016/tcp  open     http-alt
6017/tcp  open     http-alt
6018/tcp  open     http-alt
6019/tcp  open     http-alt
6020/tcp  open     http-alt
6021/tcp  open     http-alt
6022/tcp  open     http-alt
6023/tcp  open     http-alt
6024/tcp  open     http-alt
6025/tcp  open     http-alt
6026/tcp  open     http-alt
6027/tcp  open     http-alt
6028/tcp  open     http-alt
6029/tcp  open     http-alt
6030/tcp  open     http-alt
6031/tcp  open     http-alt
6032/tcp  open     http-alt
6033/tcp  open     http-alt
6034/tcp  open     http-alt
6035/tcp  open     http-alt
6036/tcp  open     http-alt
6037/tcp  open     http-alt
6038/tcp  open     http-alt
6039/tcp  open     http-alt
6040/tcp  open     http-alt
6041/tcp  open     http-alt
6042/tcp  open     http-alt
6043/tcp  open     http-alt
6044/tcp  open     http-alt
6045/tcp  open     http-alt
6046/tcp  open     http-alt
6047/tcp  open     http-alt
6048/tcp  open     http-alt
6049/tcp  open     http-alt
6050/tcp  open     http-alt
6051/tcp  open     http-alt
6052/tcp  open     http-alt
6053/tcp  open     http-alt
6054/tcp  open     http-alt
6055/tcp  open     http-alt
6056/tcp  open     http-alt
6057/tcp  open     http-alt
6058/tcp  open     http-alt
6059/tcp  open     http-alt
6060/tcp  open     http-alt
6061/tcp  open     http-alt
6062/tcp  open     http-alt
6063/tcp  open     http-alt
6064/tcp  open     http-alt
6065/tcp  open     http-alt
6066/tcp  open     http-alt
6067/tcp  open     http-alt
6068/tcp  open     http-alt
6069/tcp  open     http-alt
6070/tcp  open     http-alt
6071/tcp  open     http-alt
6072/tcp  open     http-alt
6073/tcp  open     http-alt
6074/tcp  open     http-alt
6075/tcp  open     http-alt
6076/tcp  open     http-alt
6077/tcp  open     http-alt
6078/tcp  open     http-alt
6079/tcp  open     http-alt
6080/tcp  open     http-alt
6081/tcp  open     http-alt
6082/tcp  open     http-alt
6083/tcp  open     http-alt
6084/tcp  open     http-alt
6085/tcp  open     http-alt
6086/tcp  open     http-alt
6087/tcp  open     http-alt
6088/tcp  open     http-alt
6089/tcp  open     http-alt
6090/tcp  open     http-alt
6091/tcp  open     http-alt
6092/tcp  open     http-alt
6093/tcp  open     http-alt
6094/tcp  open     http-alt
6095/tcp  open     http-alt
6096/tcp  open     http-alt
6097/tcp  open     http-alt
6098/tcp  open     http-alt
6099/tcp  open     http-alt
6000-6099/tcp open     http-alt
```

Instructions Resources

- 9. You are now viewing a banner grab as well as port scan, as shown in the above screenshot
- 10. If there is time, continue using the tool and explore more options.
- 11. This concludes the lab exercise.

Question 6.7.1

Perform port scanning on the target IP, 192.168.0.22, using the Dmitry tool. Enter the state of the port 135/TCP (Open/Closed/Filtered).

open

Score

Correct

← Previous Next →

1 Hr 25 Min Remaining

Conclusion

In Conclusion, I have learned about how to leverage netdiscover to execute fingerprinting, hping3 to learn about the network communications that occur internally as well as any open ports. I also have learned how to leverage nmap for scanning more in depth as well as gaining insight to the database. I also have learned about Dmitry in relation to gaining insight on the ports that the organization has left open as well as how to leverage ftp to execute remote connections.