

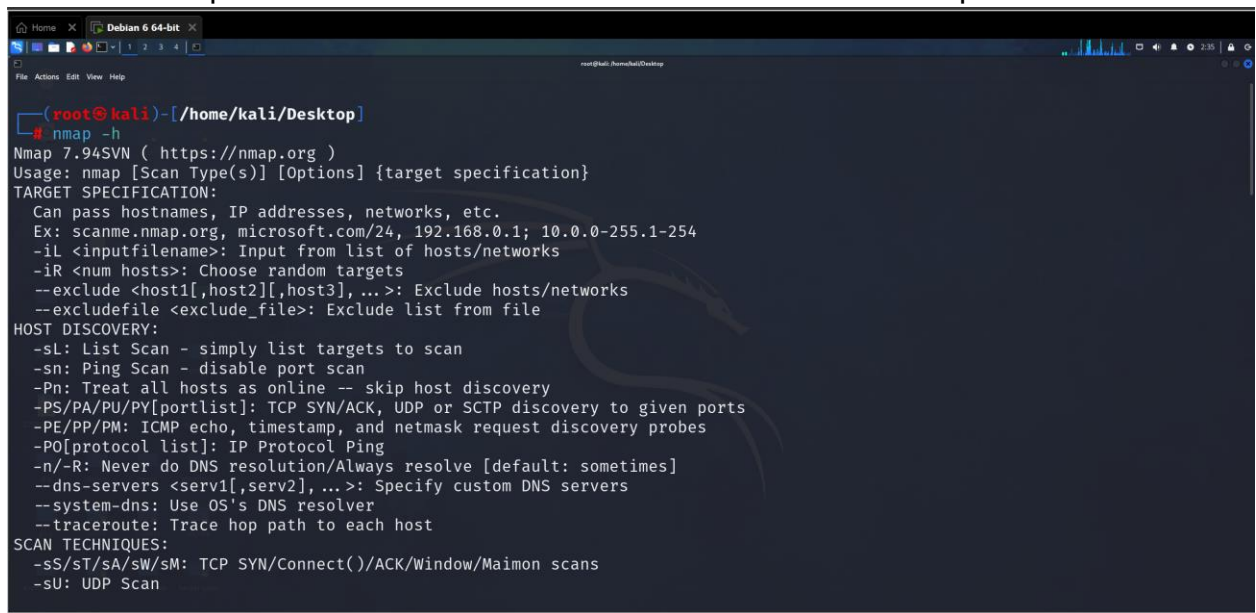
Activity 5-1: Getting to Know Nmap

Time Required: 30 minutes

Objective: Learn the basic commands and syntax of Nmap.

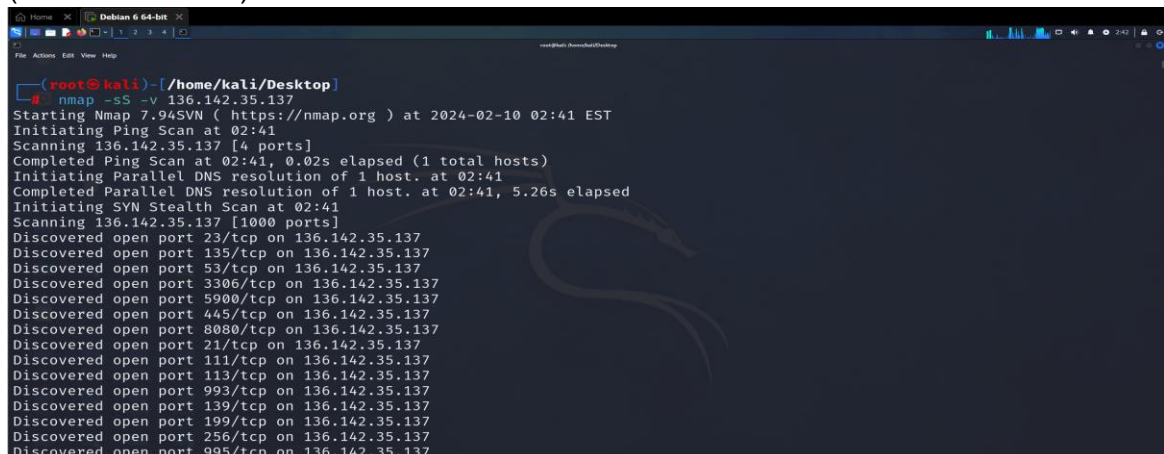
Description: In this activity, you use Nmap to perform quick scans of a network. You send a SYN packet to a host on the attack network your instructor has supplied. In this example, the attack network IP addresses are 136.142.35.137 to 136.142.35.140. Make sure to follow the rules of engagement, and don't perform port scanning on any systems not included in the IP range your instructor gives you.

1. Boot your computer into Linux. Open a command shell by clicking the Terminal icon on the panel taskbar. Type `nmap -h` | `less` and press Enter to see all available Nmap commands. You can scroll to review the command parameters.



```
(root@kali)-[/home/kali/Desktop]
# nmap -h
Nmap 7.94SVN ( https://nmap.org )
Usage: nmap [Scan Type(s)] [Options] {target specification}
TARGET SPECIFICATION:
  Can pass hostnames, IP addresses, networks, etc.
  Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254
  -iL <inputfilename>: Input from list of hosts/networks
  -iR <num hosts>: Choose random targets
  --exclude <host1[,host2][,host3], ...>: Exclude hosts/networks
  --excludefile <exclude_file>: Exclude list from file
HOST DISCOVERY:
  -sl: List Scan - simply list targets to scan
  -sn: Ping Scan - disable port scan
  -Pn: Treat all hosts as online -- skip host discovery
  -PS/PA/PY[portlist]: TCP SYN/ACK, UDP or SCTP discovery to given ports
  -PE/PP/PM: ICMP echo, timestamp, and netmask request discovery probes
  -PO[protocol list]: IP Protocol Ping
  -n/-R: Never do DNS resolution/Always resolve [default: sometimes]
  --dns-servers <serv1[,serv2], ...>: Specify custom DNS servers
  --system-dns: Use OS's DNS resolver
  --traceroute: Trace hop path to each host
SCAN TECHNIQUES:
  -sS/sT/sA/sW/sM: TCP SYN/Connect()/ACK/Window/Maimon scans
  -sU: UDP Scan
```

2. To send a SYN packet to an IP address in your attack range, type `nmap -sS -v 136.142.35.137` and press Enter. What are the results of your SYN scan? (SCREENSHOT)



```
(root@kali)-[/home/kali/Desktop]
# nmap -sS -v 136.142.35.137
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-10 02:41 EST
Initiating Ping Scan at 02:41
Scanning 136.142.35.137 [4 ports]
Completed Ping Scan at 02:41, 0.02s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 02:41
Completed Parallel DNS resolution of 1 host. at 02:41, 5.26s elapsed
Initiating SYN Stealth Scan at 02:41
Scanning 136.142.35.137 [1000 ports]
Discovered open port 23/tcp on 136.142.35.137
Discovered open port 135/tcp on 136.142.35.137
Discovered open port 53/tcp on 136.142.35.137
Discovered open port 3306/tcp on 136.142.35.137
Discovered open port 5900/tcp on 136.142.35.137
Discovered open port 445/tcp on 136.142.35.137
Discovered open port 8080/tcp on 136.142.35.137
Discovered open port 21/tcp on 136.142.35.137
Discovered open port 111/tcp on 136.142.35.137
Discovered open port 113/tcp on 136.142.35.137
Discovered open port 993/tcp on 136.142.35.137
Discovered open port 139/tcp on 136.142.35.137
Discovered open port 199/tcp on 136.142.35.137
Discovered open port 256/tcp on 136.142.35.137
Discovered open port 995/tcp on 136.142.35.137
```

3 . Next, try sending a new SYN packet to a different IP address in your attack range. What are the results of this new scan? Do you see any differences? If so, list them. (SCREENSHOT)

```
root@kali: ~/home/kali/Desktop
# nmap -sS -v 136.142.35.135
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-10 02:44 EST
Initiating Ping Scan at 02:44
Scanning 136.142.35.135 [4 ports]
Completed Ping Scan at 02:44, 0.02s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 02:44
Completed Parallel DNS resolution of 1 host. at 02:44, 0.39s elapsed
Initiating SYN Stealth Scan at 02:44
Scanning www.dmap.pitt.edu (136.142.35.135) [1000 ports]
Discovered open port 80/tcp on 136.142.35.135
Discovered open port 199/tcp on 136.142.35.135
Discovered open port 1720/tcp on 136.142.35.135
Discovered open port 1723/tcp on 136.142.35.135
Discovered open port 445/tcp on 136.142.35.135
Discovered open port 143/tcp on 136.142.35.135
Discovered open port 139/tcp on 136.142.35.135
Discovered open port 993/tcp on 136.142.35.135
Discovered open port 554/tcp on 136.142.35.135
Discovered open port 8080/tcp on 136.142.35.135
Discovered open port 113/tcp on 136.142.35.135
Discovered open port 995/tcp on 136.142.35.135
Discovered open port 5900/tcp on 136.142.35.135
Discovered open port 8888/tcp on 136.142.35.135
Discovered open port 3306/tcp on 136.142.35.135
```

4 . Nmap can scan through a range of IP addresses, so entering one IP address at a time isn't necessary. To send a SYN packet to every IP address in your attack range, type `nmap -sS -v 136.142.35.137-140` and press Enter. (SCREENSHOT)

```
root@kali: ~/home/kali/Desktop
# nmap -sS -v 136.142.35.137-140
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-10 02:45 EST
Initiating Ping Scan at 02:45
Scanning 4 hosts [4 ports/host]
Completed Ping Scan at 02:45, 0.04s elapsed (4 total hosts)
Initiating Parallel DNS resolution of 4 hosts. at 02:45
Completed Parallel DNS resolution of 4 hosts. at 02:45, 0.41s elapsed
Initiating SYN Stealth Scan at 02:45
Scanning 4 hosts [1000 ports/host]
Discovered open port 22/tcp on 136.142.35.138
Discovered open port 22/tcp on 136.142.35.139
Discovered open port 22/tcp on 136.142.35.140
Discovered open port 22/tcp on 136.142.35.137
Discovered open port 554/tcp on 136.142.35.138
Discovered open port 554/tcp on 136.142.35.139
Discovered open port 554/tcp on 136.142.35.140
Discovered open port 554/tcp on 136.142.35.137
Discovered open port 1720/tcp on 136.142.35.138
Discovered open port 1720/tcp on 136.142.35.139
Discovered open port 1720/tcp on 136.142.35.140
Discovered open port 1720/tcp on 136.142.35.137
Discovered open port 111/tcp on 136.142.35.138
Discovered open port 111/tcp on 136.142.35.139
Discovered open port 111/tcp on 136.142.35.140
```

6. To see the output in a format you can scroll, press the Up Arrow key, add the | less option to the end of the Nmap command, and press Enter. The command should look like this: `nmap -sS -v 136.142.35.137-140 | less`.

```
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-10 02:46 EST
Initiating Ping Scan at 02:46
Scanning 4 hosts [4 ports/host]
Completed Ping Scan at 02:46, 0.03s elapsed (4 total hosts)
Initiating Parallel DNS resolution of 4 hosts. at 02:46
Completed Parallel DNS resolution of 4 hosts. at 02:46, 0.18s elapsed
Initiating SYN Stealth Scan at 02:46
Scanning 4 hosts [1000 ports/host]
Discovered open port 110/tcp on 136.142.35.138
Discovered open port 110/tcp on 136.142.35.139
Discovered open port 110/tcp on 136.142.35.140
Discovered open port 110/tcp on 136.142.35.137
Discovered open port 995/tcp on 136.142.35.138
Discovered open port 995/tcp on 136.142.35.139
Discovered open port 995/tcp on 136.142.35.140
Discovered open port 995/tcp on 136.142.35.137
Discovered open port 554/tcp on 136.142.35.138
Discovered open port 554/tcp on 136.142.35.139
Discovered open port 554/tcp on 136.142.35.140
Discovered open port 1720/tcp on 136.142.35.140
Discovered open port 256/tcp on 136.142.35.140
Discovered open port 199/tcp on 136.142.35.137
Discovered open port 8080/tcp on 136.142.35.137
Discovered open port 1025/tcp on 136.142.35.138
Discovered open port 8080/tcp on 136.142.35.139
:
```

7. Next, add one more parameter to the Nmap command to determine which computers in your attack range have the SMTP service or HTTP service running. Using what you've learned so far in this activity, enter the command and note the output. (Hint: What ports do SMTP and HTTP use?) The command's output might vary, but what's important is learning how to build on the Nmap command. You can select specific ports in the Nmap command, so not all 65,000 ports must be scanned.

```
(root@kali)-[/home/kali/Desktop]
# nmap -p 25,80 136.142.35.137-140

Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-10 02:47 EST
Nmap scan report for 136.142.35.137
Host is up (0.024s latency).

PORT      STATE      SERVICE
25/tcp    filtered  smtp
80/tcp    open       http

Nmap scan report for 136.142.35.138
Host is up (0.023s latency).

PORT      STATE      SERVICE
25/tcp    filtered  smtp
80/tcp    open       http

Nmap scan report for 136.142.35.139
Host is up (0.023s latency).

PORT      STATE      SERVICE
25/tcp    filtered  smtp
80/tcp    open       http
```

8. Leave the Terminal shell open for the next activity.

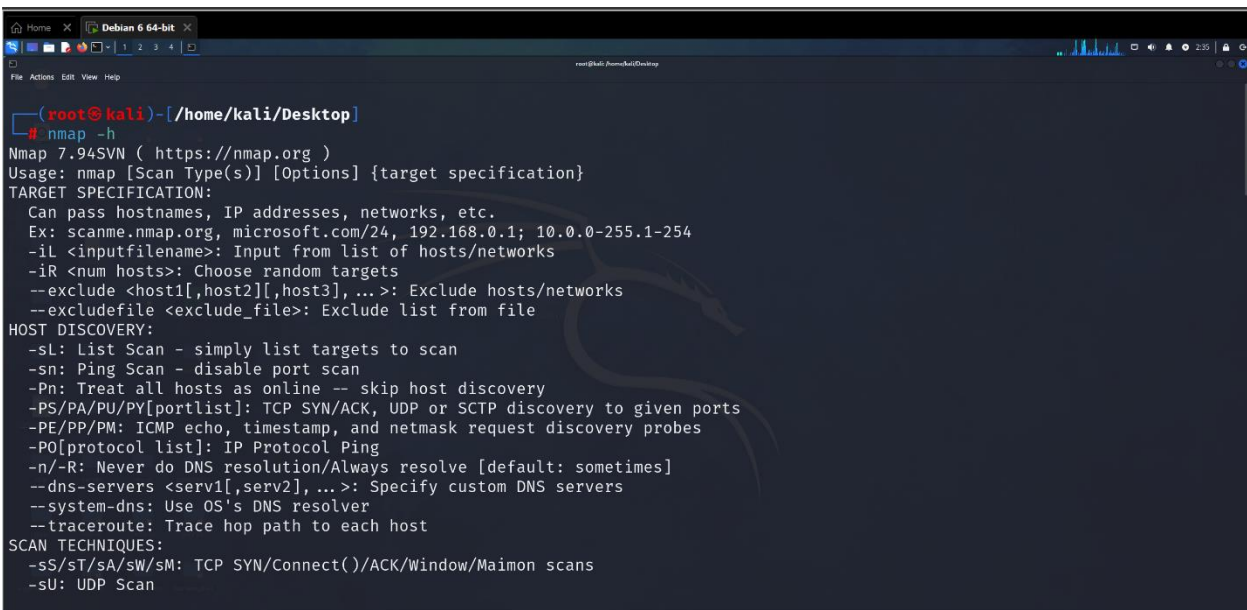
Activity 5-2: Using Additional Nmap Commands

Time Required: 30 minutes

Objective: Perform more complex port-scanning attacks with Nmap.

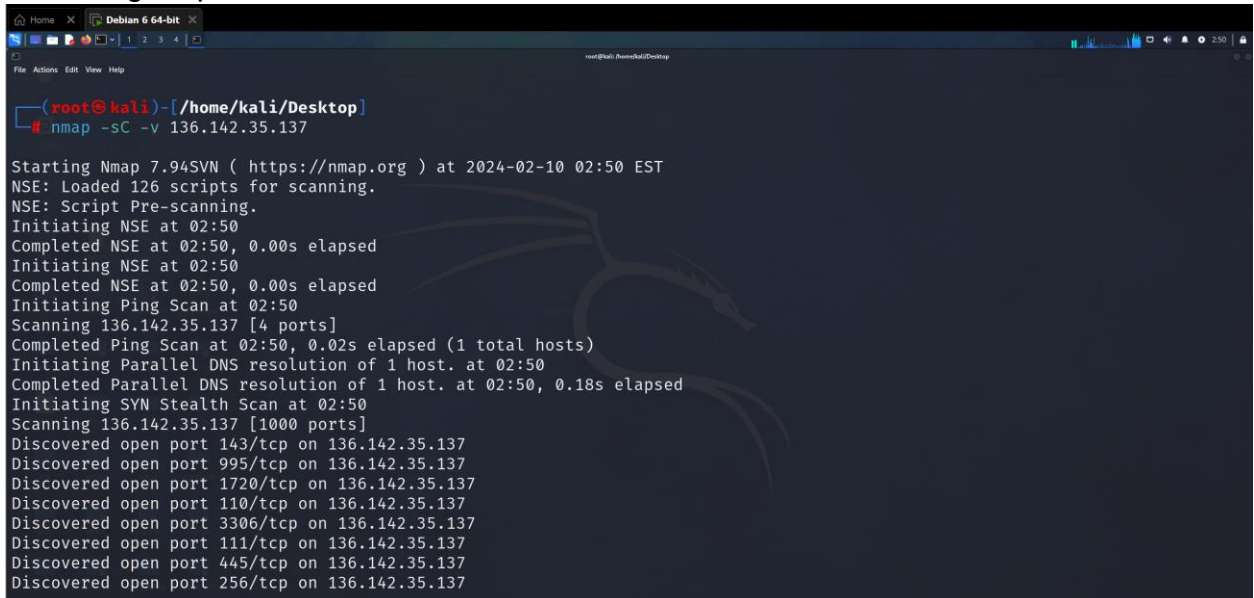
Description: In this activity, you continue to use Nmap for port scanning on your attack network. You add to the parameters used in Activity 5-1 using Nmap scripts to discover more information about the remote host. You should practice these commands until they are second nature, but Fyodor developed a well-written help page (called a “man page” in UNIX/Linux circles) that you can use as a resource. You begin this activity by looking at this help page.

1. If a Terminal window isn't open, boot your computer into Kali Linux, open a Terminal shell, and at the command prompt, type `man nmap` and press Enter. You can see that this command produces more information than the `nmap -h` command. Don't be concerned about memorizing the manual; just know it's there when you need it.



```
(root@kali)~/Desktop
# nmap -h
Nmap 7.94SVN ( https://nmap.org )
Usage: nmap [Scan Type(s)] [Options] {target specification}
TARGET SPECIFICATION:
  Can pass hostnames, IP addresses, networks, etc.
  Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254
  -iL <inputfilename>: Input from list of hosts/networks
  -iR <num hosts>: Choose random targets
  --exclude <host1[,host2][,host3],...>: Exclude hosts/networks
  --excludefile <exclude_file>: Exclude list from file
HOST DISCOVERY:
  -sL: List Scan - simply list targets to scan
  -sn: Ping Scan - disable port scan
  -Pn: Treat all hosts as online -- skip host discovery
  -PS/PA/PU/PY[portlist]: TCP SYN/ACK, UDP or SCTP discovery to given ports
  -PE/PP/PM: ICMP echo, timestamp, and netmask request discovery probes
  -PO[protocol list]: IP Protocol Ping
  -n/-R: Never do DNS resolution/Always resolve [default: sometimes]
  --dns-servers <serv1[,serv2],...>: Specify custom DNS servers
  --system-dns: Use OS's DNS resolver
  --traceroute: Trace hop path to each host
SCAN TECHNIQUES:
  -sS/sT/sA/sW/sM: TCP SYN/Connect()/ACK/Window/Maimon scans
  -sU: UDP Scan
```

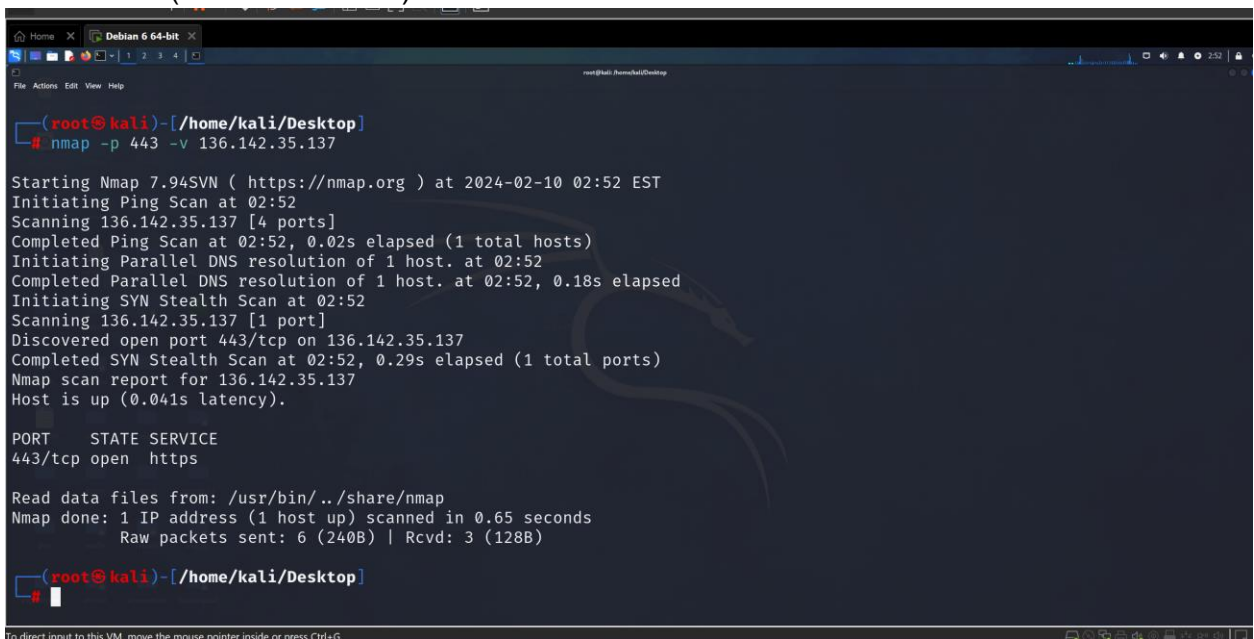

- Next, enter the command to send a default script scan to 136.142.35.137 (`nmap -sC -v 136.142.35.137`). You can read more about the default scripts included with the default scan setting at <https://nmap.org/nsedoc/categories/default.html>. What are the results of the script scan? What brand and version of HTTP server is running on ports 80 and 443?



```
(root@kali)~/home/kali/Desktop
# nmap -sC -v 136.142.35.137

Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-10 02:50 EST
NSE: Loaded 126 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 02:50
Completed NSE at 02:50, 0.00s elapsed
Initiating NSE at 02:50
Completed NSE at 02:50, 0.00s elapsed
Initiating Ping Scan at 02:50
Scanning 136.142.35.137 [4 ports]
Completed Ping Scan at 02:50, 0.02s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 02:50
Completed Parallel DNS resolution of 1 host. at 02:50, 0.18s elapsed
Initiating SYN Stealth Scan at 02:50
Scanning 136.142.35.137 [1000 ports]
Discovered open port 143/tcp on 136.142.35.137
Discovered open port 995/tcp on 136.142.35.137
Discovered open port 1720/tcp on 136.142.35.137
Discovered open port 110/tcp on 136.142.35.137
Discovered open port 3306/tcp on 136.142.35.137
Discovered open port 111/tcp on 136.142.35.137
Discovered open port 445/tcp on 136.142.35.137
Discovered open port 256/tcp on 136.142.35.137
```

- Now, limit the scope so you scan only port 443 by using the `-p` flag (`nmap -p443 -v 136.142.35.137`). This makes the Nmap scan more targeted and less noticeable. (SCREENSHOT)



```
(root@kali)~/home/kali/Desktop
# nmap -p 443 -v 136.142.35.137

Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-10 02:52 EST
Initiating Ping Scan at 02:52
Scanning 136.142.35.137 [4 ports]
Completed Ping Scan at 02:52, 0.02s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 02:52
Completed Parallel DNS resolution of 1 host. at 02:52, 0.18s elapsed
Initiating SYN Stealth Scan at 02:52
Scanning 136.142.35.137 [1 port]
Discovered open port 443/tcp on 136.142.35.137
Completed SYN Stealth Scan at 02:52, 0.29s elapsed (1 total ports)
Nmap scan report for 136.142.35.137
Host is up (0.041s latency).

PORT      STATE SERVICE
443/tcp   open  https

Read data files from: /usr/bin/./share/nmap
Nmap done: 1 IP address (1 host up) scanned in 0.65 seconds
Raw packets sent: 6 (240B) | Rcvd: 3 (128B)

(root@kali)~/home/kali/Desktop
```

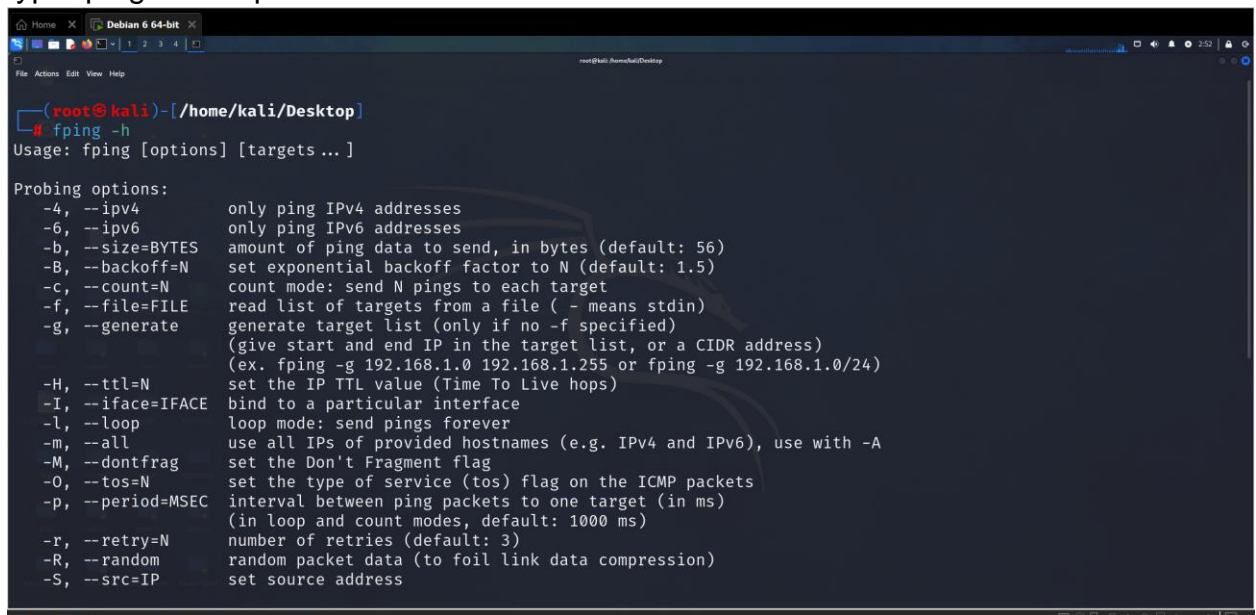
Activity 5-3: Crafting IP Packets with Fping and Hping3

Time Required: 30 minutes

Objective: Learn to craft IP packets with Fping and Hping3.

Description: In this activity, you see how security testers can craft IP packets to find out what services are running on a network. The more ways you know how to send a packet to an unsuspecting port on a computer and get a response, the better. If a computer doesn't respond to an ICMP packet sent to a particular port, it doesn't mean any packet sent to the same port will get the same response. You might need to send different packets to get the results you need for a thorough security test.

1. If necessary, boot your computer into Linux. Open a Terminal shell, and then type `fping -h` and press Enter.



```
(root@kali)-[/home/kali/Desktop]
# fping -h
Usage: fping [options] [targets ...]

Probing options:
-4, --ipv4          only ping IPv4 addresses
-6, --ipv6          only ping IPv6 addresses
-b, --size=BYTES    amount of ping data to send, in bytes (default: 56)
-B, --backoff=N     set exponential backoff factor to N (default: 1.5)
-c, --count=N       count mode: send N pings to each target
-f, --file=FILE     read list of targets from a file (- means stdin)
-g, --generate      generate target list (only if no -f specified)
                   (give start and end IP in the target list, or a CIDR address)
                   (ex. fping -g 192.168.1.0 192.168.1.255 or fping -g 192.168.1.0/24)
-H, --ttl=N         set the IP TTL value (Time To Live hops)
-I, --iface=IFACE   bind to a particular interface
-l, --loop          loop mode: send pings forever
-m, --all           use all IPs of provided hostnames (e.g. IPv4 and IPv6), use with -A
-M, --dontfrag     set the Don't Fragment flag
-O, --tos=N         set the type of service (tos) flag on the ICMP packets
-p, --period=MSEC   interval between ping packets to one target (in ms)
                   (in loop and count modes, default: 1000 ms)
-r, --retry=N       number of retries (default: 3)
-R, --random        random packet data (to foil link data compression)
-S, --src=IP        set source address
```

2. To see the live computers in the attack range, type `fping -g BeginningIPAddress EndingIPAddress` and press Enter. Note the results. (Be sure to use the beginning and ending IP addresses in your attack range.) (SCREENSHOT)

```
Debian 6 64-bit
root@kali:~/Desktop# fping -g 192.168.1.1 192.168.1.255

192.168.1.1 is alive
192.168.1.4 is alive
192.168.1.2 is alive
ICMP Host Unreachable from 192.168.1.4 for ICMP Echo sent to 192.168.1.9
ICMP Host Unreachable from 192.168.1.4 for ICMP Echo sent to 192.168.1.8
ICMP Host Unreachable from 192.168.1.4 for ICMP Echo sent to 192.168.1.7
ICMP Host Unreachable from 192.168.1.4 for ICMP Echo sent to 192.168.1.6
ICMP Host Unreachable from 192.168.1.4 for ICMP Echo sent to 192.168.1.3
ICMP Host Unreachable from 192.168.1.4 for ICMP Echo sent to 192.168.1.46
ICMP Host Unreachable from 192.168.1.4 for ICMP Echo sent to 192.168.1.45
ICMP Host Unreachable from 192.168.1.4 for ICMP Echo sent to 192.168.1.44
ICMP Host Unreachable from 192.168.1.4 for ICMP Echo sent to 192.168.1.43
ICMP Host Unreachable from 192.168.1.4 for ICMP Echo sent to 192.168.1.42
ICMP Host Unreachable from 192.168.1.4 for ICMP Echo sent to 192.168.1.41
ICMP Host Unreachable from 192.168.1.4 for ICMP Echo sent to 192.168.1.40
ICMP Host Unreachable from 192.168.1.4 for ICMP Echo sent to 192.168.1.39
ICMP Host Unreachable from 192.168.1.4 for ICMP Echo sent to 192.168.1.38
ICMP Host Unreachable from 192.168.1.4 for ICMP Echo sent to 192.168.1.37
ICMP Host Unreachable from 192.168.1.4 for ICMP Echo sent to 192.168.1.36
ICMP Host Unreachable from 192.168.1.4 for ICMP Echo sent to 192.168.1.35
ICMP Host Unreachable from 192.168.1.4 for ICMP Echo sent to 192.168.1.34
ICMP Host Unreachable from 192.168.1.4 for ICMP Echo sent to 192.168.1.33
```

- Next, type `hping3 -S IPAddressAttackedComputer` (substituting an IP address from your attack range) and press Enter. By using the `-S` parameter, you craft a TCP SYN packet.

```
Debian 6 64-bit
root@kali:~/Desktop# sudo hping3 -c 3 -S -p 80 192.168.10.1

HPING 192.168.10.1 (eth0 192.168.10.1): S set, 40 headers + 0 data bytes

--- 192.168.10.1 hping statistic ---
3 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms

root@kali:~/Desktop#
```

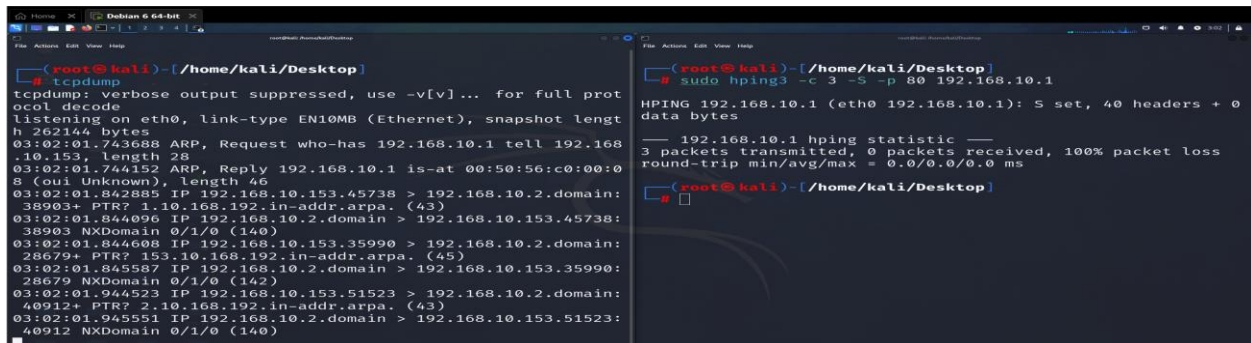
- Open another Terminal shell, and then type `tcpdump` and press Enter. (SCREENSHOT)

```
Debian 6 64-bit
root@kali:~/Desktop# tcpdump

tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
03:00:35.786362 ARP, Request who-has 192.168.10.2 tell 192.168.10.1, length 46
03:00:35.869686 IP 192.168.10.153.55118 > 192.168.10.2.domain: 44073+ PTR? 2.10.168.192.in-addr.arpa. (43)
03:00:36.012460 IP 192.168.10.2.domain > 192.168.10.153.55118: 44073 NXDomain 0/1/0 (140)
03:00:36.013057 IP 192.168.10.153.42095 > 192.168.10.2.domain: 57289+ PTR? 1.10.168.192.in-addr.arpa. (43)
03:00:36.166701 IP 192.168.10.2.domain > 192.168.10.153.42095: 57289 NXDomain 0/1/0 (140)
03:00:36.167781 IP 192.168.10.153.48378 > 192.168.10.2.domain: 26189+ PTR? 153.10.168.192.in-addr.arpa. (45)
03:00:36.358990 IP 192.168.10.2.domain > 192.168.10.153.48378: 26189 NXDomain 0/1/0 (142)
03:00:36.695666 ARP, Request who-has 192.168.10.2 tell 192.168.10.1, length 46
03:00:37.696141 ARP, Request who-has 192.168.10.2 tell 192.168.10.1, length 46
03:00:38.788313 ARP, Request who-has 192.168.10.2 tell 192.168.10.1, length 46
03:00:39.697356 ARP, Request who-has 192.168.10.2 tell 192.168.10.1, length 46
03:00:40.697204 ARP, Request who-has 192.168.10.2 tell 192.168.10.1, length 46
03:00:41.102640 ARP, Request who-has 192.168.10.2 tell 192.168.10.153, length 28
03:00:41.103121 ARP, Reply 192.168.10.2 is-at 00:50:56:f1:21:fe (oui Unknown), length 46
03:00:41.791222 ARP, Request who-has 192.168.10.2 tell 192.168.10.1, length 46
```

- Arrange both shell windows next to each other so that you can observe what happens after entering the `Hping3` command type `hping3 -S`

IPAddressAttackedComputer, and press Enter. Watch the Tcpcmdump window fill with the traffic that's generated. To stop Tcpcmdump from capturing packets, press Ctrl+C in that shell window. (SCREENSHOT)



```
(root@kali)-[/home/kali/Desktop]
# tcpdump
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
03:02:01.743688 ARP, Request who-has 192.168.10.1 tell 192.168.10.153, length 28
03:02:01.744152 ARP, Reply 192.168.10.1 is-at 00:50:56:c0:00:08 (oui Unknown), length 46
03:02:01.842885 IP 192.168.10.153.45738 > 192.168.10.2.domain: 38903+ PTR? 1.10.168.192.in-addr.arpa. (43)
03:02:01.844096 IP 192.168.10.2.domain > 192.168.10.153.45738: 38903 NXDomain 0/1/0 (140)
03:02:01.844608 IP 192.168.10.153.35990 > 192.168.10.2.domain: 28679+ PTR? 153.10.168.192.in-addr.arpa. (45)
03:02:01.845587 IP 192.168.10.2.domain > 192.168.10.153.35990: 28679 NXDomain 0/1/0 (142)
03:02:01.944523 IP 192.168.10.153.51523 > 192.168.10.2.domain: 40912+ PTR? 2.10.168.192.in-addr.arpa. (43)
03:02:01.945551 IP 192.168.10.2.domain > 192.168.10.153.51523: 40912 NXDomain 0/1/0 (140)

(root@kali)-[/home/kali/Desktop]
# sudo hping3 -c 3 -S -p 80 192.168.10.1
HPING 192.168.10.1 (eth0 192.168.10.1): S set, 40 headers + 0 data bytes
--- 192.168.10.1 hping statistic ---
3 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
(root@kali)-[/home/kali/Desktop]
```

6. Consult the Hping3 help pages and experiment with creating different types of packets. Note the differences in network traffic generated with the Tcpcmdump command. Security testers need to understand how slight variations in packets sent to an attacked computer can produce different results. For example, if a computer doesn't respond to a SYN packet, try sending an ACK packet. What happens when a FIN packet is sent? If you aren't having any success, try sending the same packets to different ports. Does this method change the response from the attacked computer?


```
root@kali: /home/kali/Desktop
File Actions Edit View Help

(root@kali)-[/home/kali/Desktop]
# sudo hping3 -F -p <80> 192.168.10.2

zsh: no such file or directory: 80

(root@kali)-[/home/kali/Desktop]
# sudo hping3 -F -p 80 192.168.10.2

HPING 192.168.10.2 (eth0 192.168.10.2): F set, 40 headers + 0
data bytes
len=46 ip=192.168.10.2 ttl=128 id=22888 sport=80 flags=RA seq=
0 win=32767 rtt=7.5 ms
len=46 ip=192.168.10.2 ttl=128 id=22889 sport=80 flags=RA seq=
1 win=32767 rtt=4.1 ms
len=46 ip=192.168.10.2 ttl=128 id=22890 sport=80 flags=RA seq=
2 win=32767 rtt=2.0 ms
len=46 ip=192.168.10.2 ttl=128 id=22891 sport=80 flags=RA seq=
3 win=32767 rtt=6.0 ms
len=46 ip=192.168.10.2 ttl=128 id=22892 sport=80 flags=RA seq=
4 win=32767 rtt=2.8 ms
len=46 ip=192.168.10.2 ttl=128 id=22893 sport=80 flags=RA seq=
5 win=32767 rtt=9.7 ms
len=46 ip=192.168.10.2 ttl=128 id=22894 sport=80 flags=RA seq=
6 win=32767 rtt=7.1 ms
```

By experimenting with different types of packets and observing the network traffic using tcpdump, you can gain insights into how slight variations in packets can produce different results and how target systems respond to different types of packets. This understanding is crucial for security testers to assess network security and identify potential vulnerabilities.

When you're done, close both shells.

Conclusions:

Through Activities 5-1 to 5-3, I gained practical insights into network scanning and packet crafting, enhancing my security testing proficiency. Experimenting with Nmap, Fping, and Hping3 underscored the importance of nuanced approaches for comprehensive vulnerability assessment.

References:

- 1- Bhuyan, M. H., Bhattacharyya, D. K., & Kalita, J. K. (2011). Surveying port scans and their detection methodologies. *The Computer Journal*, 54(10), 1565-1581.
- 2- Asrak, Z. (2020). Penetration Testing Tools.
- 3- Ganzy, E. G. (2014). Scalable Asset Discovery, Vulnerability Scanning, and Penetration Testing for Remote Sites and Wireless Spectrums Utilizing an Embedded Linux Plug-PwniPlug and the Raspberry Pi B+ as a Sample Pen Test (No. KSC-E-DAA-TN19263).