ASSIGNMENT – 5

APARNAA MAHALAXMI ARULLJOTHI (A20560995)

Part I: Scanning - probing the target

2,3- To scan a single IP address

Here a single IP address is scanned using the nmap command.

```
┌──(kali㉿kali)-[~]
└─$ nmap 192.168.1.30
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-03-30 15:14 EDT
Nmap scan report for 192.168.1.30
Host is up (0.00067s latency).
Not shown: 988 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
631/tcp   open  ipp
3306/tcp  open  mysql
6667/tcp  open  irc
8080/tcp  open  http-proxy
8181/tcp  open  intermapper
10010/tcp open  rxapi
MAC Address: 08:00:27:63:D2:AC (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.39 seconds

┌──(kali㉿kali)-[~]
```

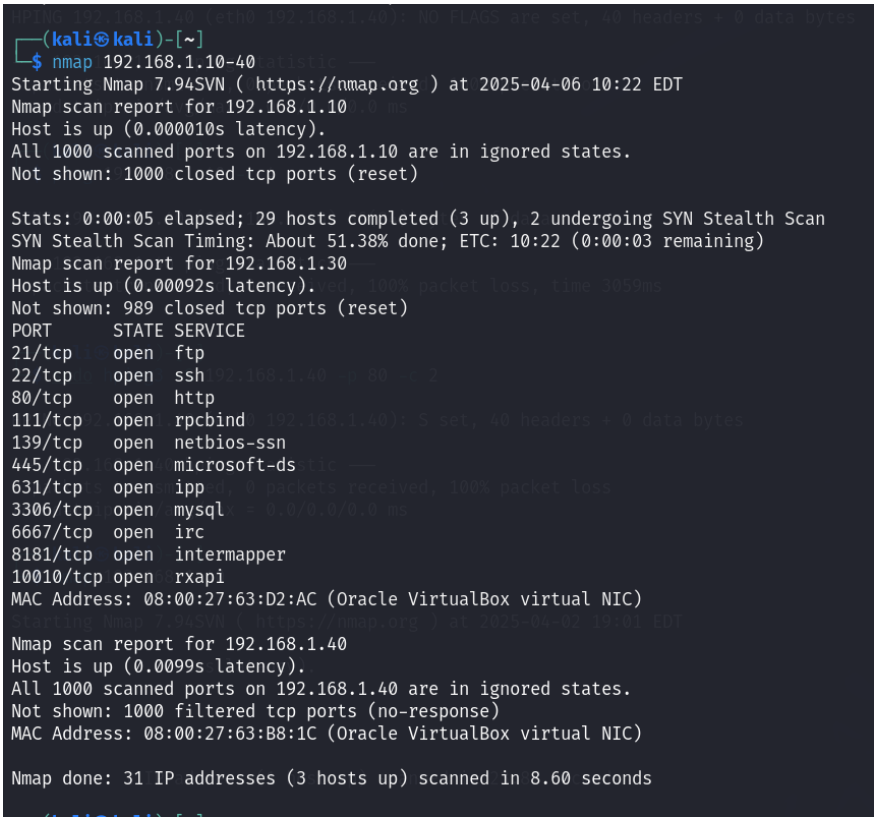
4,5- To scan a host

Here a host is scanned using nmap scanme

```
┌──(kali㉿kali)-[~]
└─$ nmap scanme.nmap.org
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-03-30 14:44 EDT
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.014s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
Not shown: 998 filtered tcp ports (no-response)
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 5.68 seconds

┌──(kali㉿kali)-[~]
└─$
```

6,7 - Scan a range of IPs

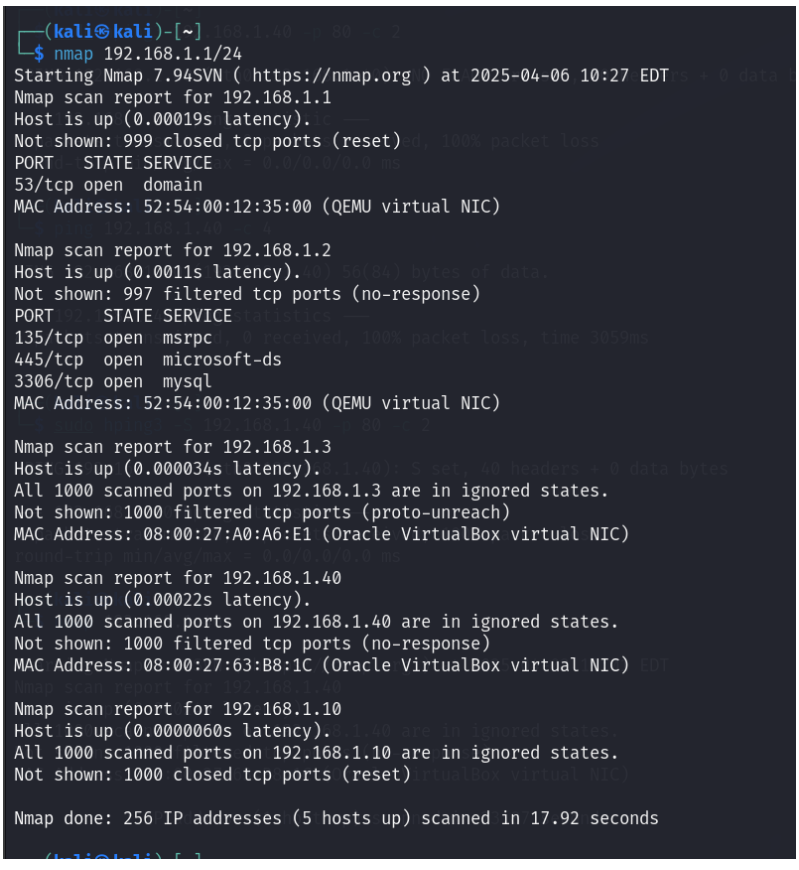
Here a range of IPs are scanned. Here I have scanned from 192.168.1.10 to 192.168.1.40. Here machine 192.168.1.10,30,40 are up and 192.168.1.20 machine is down.



```
┌──(kali㉿kali)-[~]
└─$ nmap 192.168.1.10-40
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-04-06 10:22 EDT
Nmap scan report for 192.168.1.10
Host is up (0.000010s latency).
All 1000 scanned ports on 192.168.1.10 are in ignored states.
Not shown: 1000 closed tcp ports (reset)

Stats: 0:00:05 elapsed; 29 hosts completed (3 up), 2 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 51.38% done; ETC: 10:22 (0:00:03 remaining)
Nmap scan report for 192.168.1.30
Host is up (0.00092s latency).
Not shown: 989 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
631/tcp   open  ipp
3306/tcp  open  mysql
6667/tcp  open  irc
8181/tcp  open  intermapper
10010/tcp open  rxapi
MAC Address: 08:00:27:63:D2:AC (Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.1.40
Host is up (0.0099s latency).
All 1000 scanned ports on 192.168.1.40 are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)
MAC Address: 08:00:27:63:B8:1C (Oracle VirtualBox virtual NIC)

Nmap done: 31 IP addresses (3 hosts up) scanned in 8.60 seconds
```

8,9- To scan a subnet

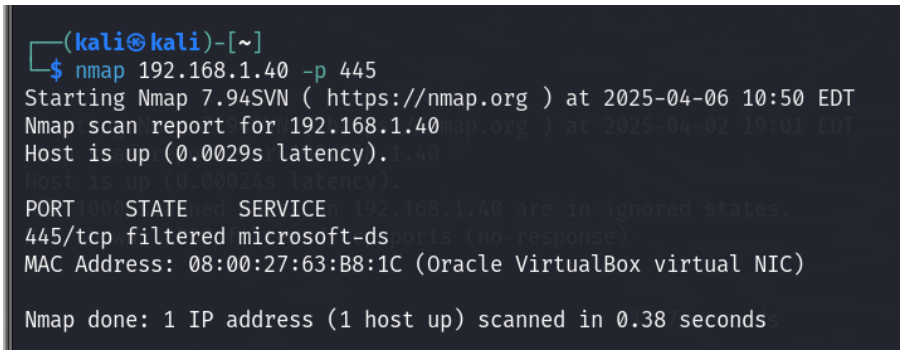
Here we scan the entire subnet of the ip 192.168.1.1. The result shows that 5 devices are alive on the subnet



```
┌──(kali㉿kali)-[~]
└─$ nmap 192.168.1.1/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-04-06 10:27 EDT
Nmap scan report for 192.168.1.1
Host is up (0.00019s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE
53/tcp open  domain
MAC Address: 52:54:00:12:35:00 (QEMU virtual NIC)

Nmap scan report for 192.168.1.2
Host is up (0.0011s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT    STATE SERVICE
135/tcp open  msrpc
445/tcp open  microsoft-ds
3306/tcp open  mysql
MAC Address: 52:54:00:12:35:00 (QEMU virtual NIC)

Nmap scan report for 192.168.1.3
Host is up (0.000034s latency).
All 1000 scanned ports on 192.168.1.3 are in ignored states.
Not shown: 1000 filtered tcp ports (proto-unreach)
MAC Address: 08:00:27:A0:A6:E1 (Oracle VirtualBox virtual NIC)

Nmap scan report for 192.168.1.40
Host is up (0.00022s latency).
All 1000 scanned ports on 192.168.1.40 are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)
MAC Address: 08:00:27:63:B8:1C (Oracle VirtualBox virtual NIC)

Nmap scan report for 192.168.1.10
Host is up (0.0000060s latency).
All 1000 scanned ports on 192.168.1.10 are in ignored states.
Not shown: 1000 closed tcp ports (reset)

Nmap done: 256 IP addresses (5 hosts up) scanned in 17.92 seconds
```

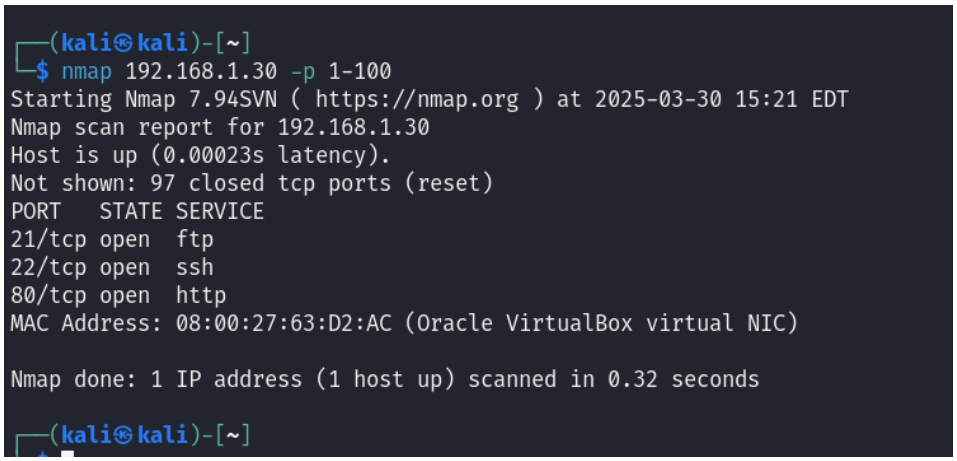
10,11- To scan a single port such as port 445

This is used to scan a single port like port 445. Here when this scan is done we get the result as the port 445 is open. Scanning a single port generates less traffic and less likely to trigger alarms. To do this we use **-p** and then mention the port number.



12,13- To scan a range of ports, e.g., 1-100,

Here we scan a range of ports using the command **-p** and then mention the range that we want to scan



14,15 - To scan 100 most common ports (fast)

This scans the most commonly used 100 ports. To do this we use **-f** option. This is known as the fast scan.

```
┌──(kali㉿kali)-[~]
└─$ nmap 192.168.1.30 -f
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-03-30 15:21 EDT
Nmap scan report for 192.168.1.30
Host is up (0.00026s latency).
Not shown: 988 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
631/tcp   open  ipp
3306/tcp  open  mysql
6667/tcp  open  irc
8080/tcp  open  http-proxy
8181/tcp  open  intermapper
10010/tcp open  rxapi
MAC Address: 08:00:27:63:D2:AC (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.33 seconds

┌──(kali㉿kali)-[~]
```

192

16,17- To Test all the ports: -p-/ -p 1-65535

These commands are used to test all the port available. This can be done using,

-p 1-65535:

```
┌──(kali㉿kali)-[~]
└─$ nmap 192.168.1.30 -p 1-65535
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-04-06 10:57 EDT
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 1.52 seconds

┌──(kali㉿kali)-[~]
└─$ nmap 192.168.1.30 -p 1-65535
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-04-06 10:59 EDT
Nmap scan report for 192.168.1.30
Host is up (0.000064s latency).
Not shown: 65519 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
631/tcp   open  ipp
3306/tcp  open  mysql
3500/tcp  open  rtmp-port
6667/tcp  open  irc
6697/tcp  open  ircs-u
8067/tcp  open  infi-async
8080/tcp  open  http-proxy
8181/tcp  open  intermapper
10010/tcp open  rxapi
48547/tcp open  unknown
MAC Address: 08:00:27:63:D2:AC (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 1.70 seconds
```

We can also do this by using,

-p- :

```
┌──(kali㉿kali)-[~]
└─$ nmap 192.168.1.30 -p-
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-04-06 11:06 EDT
Nmap scan report for 192.168.1.30
Host is up (0.000067s latency).
Not shown: 65519 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
631/tcp   open  ipp
3306/tcp  open  mysql
3500/tcp  open  rtmp-port
6667/tcp  open  irc
6697/tcp  open  ircs-u
8067/tcp  open  infi-async
8080/tcp  open  http-proxy
8181/tcp  open  intermapper
10010/tcp open  rxapi
48547/tcp open  unknown
MAC Address: 08:00:27:63:D2:AC (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 1.43 seconds
```

18,19,20- To scan the top 5 ports and reports if closed or open,

The option **–top-ports** is used to mention the top most commonly used ports that are needed to be scanned. Here I have scanned top 5,1 and 2 ports

Top 5:

```
┌──(kali㉿kali)-[~]
└─$ nmap 192.168.1.30 --top-ports 5
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-03-30 15:24 EDT
Nmap scan report for 192.168.1.30
Host is up (0.00059s latency).

PORT    STATE  SERVICE
21/tcp  open   ftp
22/tcp  open   ssh
23/tcp  closed telnet
80/tcp  open   http
443/tcp closed https
MAC Address: 08:00:27:63:D2:AC (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.28 seconds

┌──(kali㉿kali)-[~]
```

Top 1:

```
┌──(kali㉿kali)-[~]
└─$ nmap 192.168.1.30 --top-ports 1
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-03-30 15:25 EDT
Nmap scan report for 192.168.1.30
Host is up (0.00060s latency).

PORT   STATE SERVICE
80/tcp open  http
MAC Address: 08:00:27:63:D2:AC (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.26 seconds
```

Top 2:

```
┌──(kali㉿kali)-[~]
└─$ nmap 192.168.1.30 --top-ports 2
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-03-30 15:25 EDT
Nmap scan report for 192.168.1.30
Host is up (0.00064s latency).

PORT   STATE  SERVICE
23/tcp closed telnet
80/tcp open   http
MAC Address: 08:00:27:63:D2:AC (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.27 seconds

┌──(kali㉿kali)-[~]
```

21,22 - performing a port scan

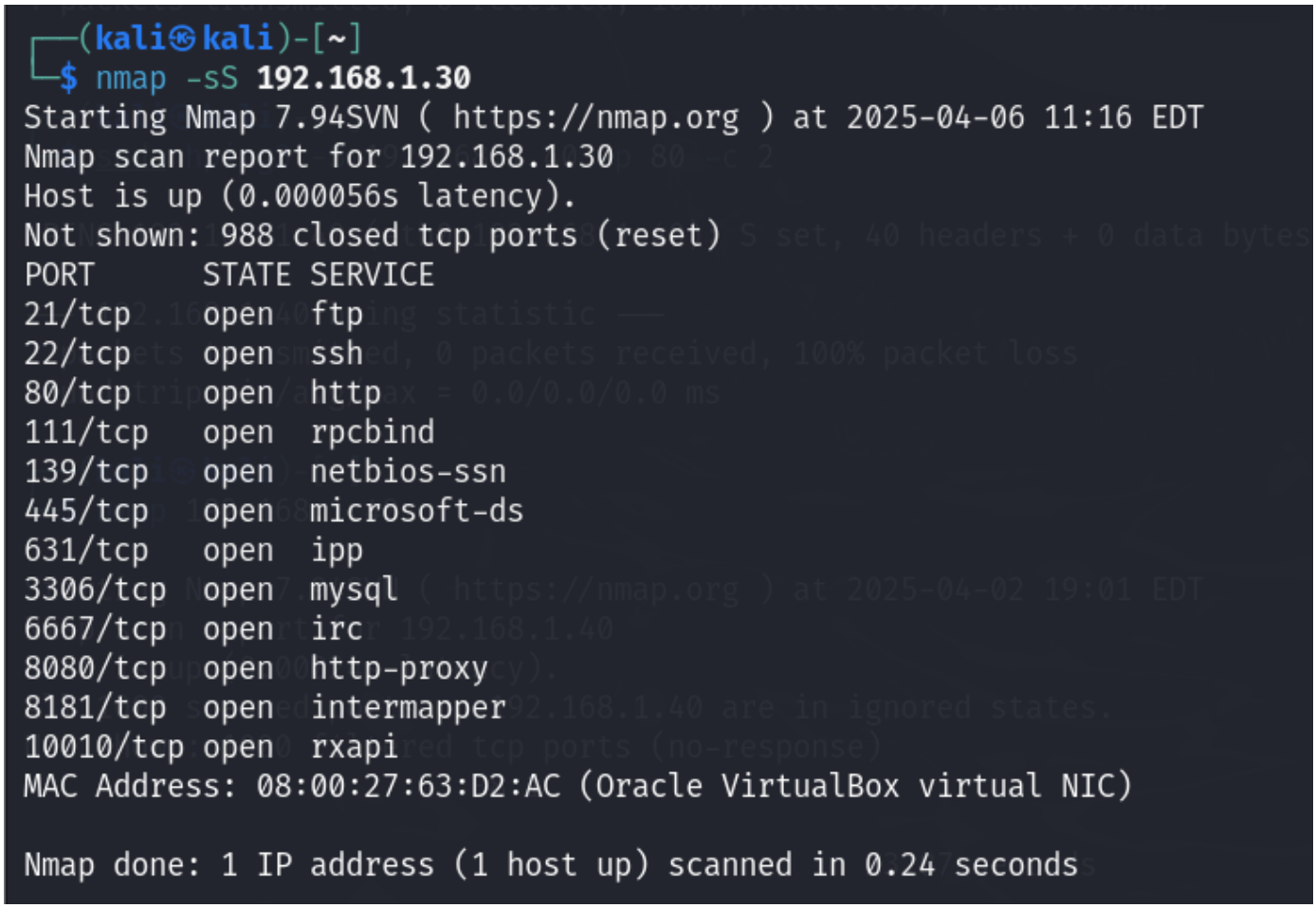
This is used to know the exact packets that are sent by namp while performing port scan. To this we add the option of packet tracing by using **–packet-trace**

```
┌──(kali㉿kali)-[~]
└─$ nmap 192.168.1.30 -p445 --packet-trace
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-03-30 15:27 EDT
SENT (0.0908s) ARP who-has 192.168.1.30 tell 192.168.1.10
RCVD (0.0914s) ARP reply 192.168.1.30 is-at 08:00:27:63:D2:AC
NSOCK INFO [0.2140s] nsock_iod_new2(): nsock_iod_new (IOD #1)
NSOCK INFO [0.2140s] nsock_connect_udp(): UDP connection requested to 192.168.0.1:53 (IOD #1) EID 8
NSOCK INFO [0.2140s] nsock_read(): Read request from IOD #1 [192.168.0.1:53] (timeout: -1ms) EID 18
NSOCK INFO [0.2140s] nsock_write(): Write request for 43 bytes to IOD #1 EID 27 [192.168.0.1:53]
NSOCK INFO [0.2140s] nsock_trace_handler_callback(): Callback: CONNECT SUCCESS for EID 8 [192.168.0.1:53]
NSOCK INFO [0.2140s] nsock_trace_handler_callback(): Callback: WRITE SUCCESS for EID 27 [192.168.0.1:53]
NSOCK INFO [0.2290s] nsock_trace_handler_callback(): Callback: READ SUCCESS for EID 18 [192.168.0.1:53] (43 bytes): '~...........30.1.168.192.in-addr.arpa.....
NSOCK INFO [0.2290s] nsock_read(): Read request from IOD #1 [192.168.0.1:53] (timeout: -1ms) EID 34
NSOCK INFO [0.2290s] nsock_iod_delete(): nsock_iod_delete (IOD #1)
NSOCK INFO [0.2290s] nevent_delete(): nevent_delete on event #34 (type READ)
SENT (0.2434s) TCP 192.168.1.10:42140 > 192.168.1.30:445 S ttl=51 id=3590 iplen=44  seq=1541069574 win=1024 <mss 1460>
RCVD (0.2440s) TCP 192.168.1.30:445 > 192.168.1.10:42140 SA ttl=64 id=0 iplen=44  seq=2931298473 win=29200 <mss 1460>
Nmap scan report for 192.168.1.30
Host is up (0.00068s latency).

PORT    STATE SERVICE
445/tcp open  microsoft-ds
MAC Address: 08:00:27:63:D2:AC (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.27 seconds
```

1.4. Nmap port scan types

1,2- Scan using TCP SYN scan

The option **-sS** specifies TCP Syn scan. It is also known as half open scan. This is a stealthy and fast method that sends a SYN packet to see if a port is open, without completing the full handshake.



3,4 - Scan using TCP connect

Here a TCP Connect scan is performed usign the option **-sT**. It is also known as a full TCP scan. This scan uses the OS's networking stack to fully complete the TCP 3-way handshake (SYN → SYN-ACK → ACK).

```
┌──(kali㉿kali)-[~]
└─$ nmap -sT 192.168.1.30
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-03-30 15:31 EDT
Nmap scan report for 192.168.1.30
Host is up (0.0010s latency).
Not shown: 988 closed tcp ports (conn-refused)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
631/tcp   open  ipp
3306/tcp  open  mysql
6667/tcp  open  irc
8080/tcp  open  http-proxy
8181/tcp  open  intermapper
10010/tcp open  rxapi
MAC Address: 08:00:27:63:D2:AC (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.35 seconds

┌──(kali㉿kali)-[~]
```

5- Fin scan

Here a FIN scan is performed usign the option **-sF.** This is a stealthy scan technique. Instead of sending a SYN packet like normal scans, it sends a FIN packet.

```
┌──(kali㉿kali)-[~]
└─$ nmap -sF 192.168.1.30
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-03-30 15:32 EDT
Nmap scan report for 192.168.1.30
Host is up (0.00028s latency).
Not shown: 988 closed tcp ports (reset)
PORT      STATE         SERVICE
21/tcp    open|filtered ftp
22/tcp    open|filtered ssh
80/tcp    open|filtered http
111/tcp   open|filtered rpcbind
139/tcp   open|filtered netbios-ssn
445/tcp   open|filtered microsoft-ds
631/tcp   open|filtered ipp
3306/tcp  open|filtered mysql
6667/tcp  open|filtered irc
8080/tcp  open|filtered http-proxy
8181/tcp  open|filtered intermapper
10010/tcp open|filtered rxapi
MAC Address: 08:00:27:63:D2:AC (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 1.48 seconds

┌──(kali㉿kali)-[~]
```

6,7 - IP Protocol scan

Here a IP portocol scan is performed using the option **-sO.** This is a type of scan that checks which IP protocols a host supports. It doesn't test services like web servers or SSH. Instead, it checks for protocols at the network layer

```
┌──(kali㉿kali)-[~]
└─$ nmap -sO 192.168.1.30
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-03-30 15:33 EDT
Stats: 0:00:36 elapsed; 0 hosts completed (1 up), 1 undergoing IPProto Scan
IPProto Scan Timing: About 28.84% done; ETC: 15:36 (0:01:29 remaining)
Stats: 0:00:43 elapsed; 0 hosts completed (1 up), 1 undergoing IPProto Scan
IPProto Scan Timing: About 31.32% done; ETC: 15:36 (0:01:34 remaining)
Stats: 0:01:21 elapsed; 0 hosts completed (1 up), 1 undergoing IPProto Scan
IPProto Scan Timing: About 45.38% done; ETC: 15:36 (0:01:37 remaining)
Stats: 0:01:50 elapsed; 0 hosts completed (1 up), 1 undergoing IPProto Scan
IPProto Scan Timing: About 57.10% done; ETC: 15:37 (0:01:23 remaining)
Stats: 0:02:22 elapsed; 0 hosts completed (1 up), 1 undergoing IPProto Scan
IPProto Scan Timing: About 68.04% done; ETC: 15:37 (0:01:06 remaining)
Stats: 0:02:51 elapsed; 0 hosts completed (1 up), 1 undergoing IPProto Scan
IPProto Scan Timing: About 76.81% done; ETC: 15:37 (0:00:52 remaining)
Warning: 192.168.1.30 giving up on port because retransmission cap hit (10).
Stats: 0:03:14 elapsed; 0 hosts completed (1 up), 1 undergoing IPProto Scan
IPProto Scan Timing: About 78.98% done; ETC: 15:38 (0:00:52 remaining)
Stats: 0:04:38 elapsed; 0 hosts completed (1 up), 1 undergoing IPProto Scan
IPProto Scan Timing: About 99.99% done; ETC: 15:38 (0:00:00 remaining)
Stats: 0:04:43 elapsed; 0 hosts completed (1 up), 1 undergoing IPProto Scan
IPProto Scan Timing: About 99.99% done; ETC: 15:38 (0:00:00 remaining)
Nmap scan report for 192.168.1.30
Host is up (0.00056s latency).
Not shown: 250 closed n/a protocols (proto-unreach)
PROTOCOL STATE         SERVICE
1        open          icmp
2        open|filtered igmp
6        open          tcp
17       open          udp
103      open|filtered pim
136      open|filtered udplite
MAC Address: 08:00:27:63:D2:AC (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 296.68 seconds

┌──(kali㉿kali)-[~]
└─$
```

8,10- ICMP echo request ping

Here we have enabled **-PE** which uses a ICMP echo request ping to the destination. In the normal type of ICMP echo request, a combination of TCP and ACK pings is sent. Using option -PE, the ICMP echo request can be specified as the nmap ping method without coupling TCP ACK ping .

```
┌──(kali㉿kali)-[~]
└─$ nmap -PE 192.168.1.30

Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-03-30 16:04 EDT
Nmap scan report for 192.168.1.30
Host is up (0.00023s latency).
Not shown: 988 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
631/tcp   open  ipp
3306/tcp  open  mysql
6667/tcp  open  irc
8080/tcp  open  http-proxy
8181/tcp  open  intermapper
10010/tcp open  rxapi
MAC Address: 08:00:27:63:D2:AC (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.35 seconds

┌──(kali㉿kali)-[~]
└─$
```

9,10 – verbose mode

This mode is enebled using the option **-v**. The verbose mode of nmap allows us to get more information from the scan output

```
┌──(kali㉿kali)-[~]
└─$ nmap -v 192.168.1.30
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-03-30 16:05 EDT
Initiating ARP Ping Scan at 16:05
Scanning 192.168.1.30 [1 port]
Completed ARP Ping Scan at 16:05, 0.09s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 16:05
Completed Parallel DNS resolution of 1 host. at 16:05, 0.22s elapsed
Initiating SYN Stealth Scan at 16:05
Scanning 192.168.1.30 [1000 ports]
Discovered open port 21/tcp on 192.168.1.30
Discovered open port 139/tcp on 192.168.1.30
Discovered open port 3306/tcp on 192.168.1.30
Discovered open port 22/tcp on 192.168.1.30
Discovered open port 111/tcp on 192.168.1.30
Discovered open port 445/tcp on 192.168.1.30
Discovered open port 80/tcp on 192.168.1.30
Discovered open port 8080/tcp on 192.168.1.30
Discovered open port 8181/tcp on 192.168.1.30
Discovered open port 10010/tcp on 192.168.1.30
Discovered open port 6667/tcp on 192.168.1.30
Discovered open port 631/tcp on 192.168.1.30
Completed SYN Stealth Scan at 16:05, 0.07s elapsed (1000 total ports)
Nmap scan report for 192.168.1.30
Host is up (0.00051s latency).
Not shown: 988 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
631/tcp   open  ipp
3306/tcp  open  mysql
6667/tcp  open  irc
8080/tcp  open  http-proxy
8181/tcp  open  intermapper
10010/tcp open  rxapi
MAC Address: 08:00:27:63:D2:AC (Oracle VirtualBox virtual NIC)

Read data files from: /usr/share/nmap
Nmap done: 1 IP address (1 host up) scanned in 0.53 seconds
           Raw packets sent: 1001 (44.028KB) | Rcvd: 1001 (40.076KB)

┌──(kali㉿kali)-[~]
└─
```

2.Service and OS detection

2.1. Detect OS and services

1,2- The command **-A** is used to scan and search for the OS and the OS version on a host. From running this scan, I found that the machine runs on linux 26.32

```
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6.32
OS details: Linux 2.6.32
Network Distance: 0 hops
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

3,4- to detect only the OS type, we can use the option **-O**

```
10010/tcp open  rxapi
MAC Address: 08:00:27:63:D2:AC (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop
```

2.2. Standard service detection

The option **-sV** is used to detect the services running. Here Nmap probes open ports to determine the application/service name, version number, OS or software stack involved

```
┌──(kali㉿kali)-[~]
└─$ nmap -sV 192.168.1.30
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-03-30 22:41 EDT
Stats: 0:00:56 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 91.67% done; ETC: 22:42 (0:00:05 remaining)
Nmap scan report for 192.168.1.30
Host is up (0.00015s latency).
Not shown: 988 closed tcp ports (reset)
PORT      STATE SERVICE     VERSION
21/tcp    open  ftp         ProFTPD 1.3.5
22/tcp    open  ssh         OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.10 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http        Apache httpd 2.4.7
111/tcp   open  rpcbind     2-4 (RPC #100000)
139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
631/tcp   open  ipp         CUPS 1.7
3306/tcp  open  mysql       MySQL (unauthorized)
6667/tcp  open  irc         UnrealIRCd
8080/tcp  open  http        Jetty 8.1.7.v20120910
8181/tcp  open  http        WEBrick httpd 1.3.1 (Ruby 2.3.7 (2018-03-28))
10010/tcp open  rxapi?
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-Port10010-TCP:V=7.94SVN%I=7%D=3/30%Time=67EA00E3%P=x86_64-pc-linux-gnu%
SF:r(GenericLines,67,"HTTP/1\.1\x20400\x20Bad\x20Request\r\nContent-Type:\
SF:x20text/plain;\x20charset=utf-8\r\nConnection:\x20close\r\n\r\n400\x20B
SF:ad\x20Request")%r(GetRequest,8F,"HTTP/1\.0\x20404\x20Not\x20Found\r\nDa
SF:te:\x20Sun,\x2030\x20Mar\x202025\x2022:04:16\x20GMT\r\nContent-Length:\
SF:x2019\r\nContent-Type:\x20text/plain;\x20charset=utf-8\r\n\r\n404:\x20P
SF:age\x20Not\x20Found")%r(HTTPOptions,8F,"HTTP/1\.0\x20404\x20Not\x20Foun
SF:d\r\nDate:\x20Sun,\x2030\x20Mar\x202025\x2022:04:16\x20GMT\r\nContent-L
SF:ength:\x2019\r\nContent-Type:\x20text/plain;\x20charset=utf-8\r\n\r\n40
SF:4:\x20Page\x20Not\x20Found")%r(RTSPRequest,67,"HTTP/1\.1\x20400\x20Bad\
SF:x20Request\r\nContent-Type:\x20text/plain;\x20charset=utf-8\r\nConnecti
SF:on:\x20close\r\n\r\n400\x20Bad\x20Request")%r(Help,67,"HTTP/1\.1\x20400
SF:\x20Bad\x20Request\r\nContent-Type:\x20text/plain;\x20charset=utf-8\r\n
SF:Connection:\x20close\r\n\r\n400\x20Bad\x20Request")%r(SSLSessionReq,67,
SF:"HTTP/1\.1\x20400\x20Bad\x20Request\r\nContent-Type:\x20text/plain;\x20
SF:charset=utf-8\r\nConnection:\x20close\r\n\r\n400\x20Bad\x20Request")%r(
SF:TerminalServerCookie,67,"HTTP/1\.1\x20400\x20Bad\x20Request\r\nContent-
SF:Type:\x20text/plain;\x20charset=utf-8\r\nConnection:\x20close\r\n\r\n40
SF:0\x20Bad\x20Request")%r(TLSSessionReq,67,"HTTP/1\.1\x20400\x20Bad\x20Re
SF:quest\r\nContent-Type:\x20text/plain;\x20charset=utf-8\r\nConnection:\x
SF:20close\r\n\r\n400\x20Bad\x20Request")%r(Kerberos,67,"HTTP/1\.1\x20400\
SF:x20Bad\x20Request\r\nContent-Type:\x20text/plain;\x20charset=utf-8\r\nC
SF:onnection:\x20close\r\n\r\n400\x20Bad\x20Request")%r(FourOhFourRequest,
SF:8F,"HTTP/1\.0\x20404\x20Not\x20Found\r\nDate:\x20Sun,\x2030\x20Mar\x202
SF:025\x2022:04:41\x20GMT\r\nContent-Length:\x2019\r\nContent-Type:\x20tex
SF:t/plain;\x20charset=utf-8\r\n\r\n404:\x20Page\x20Not\x20Found")%r(LPDSt
SF:ring,67,"HTTP/1\.1\x20400\x20Bad\x20Request\r\nContent-Type:\x20text/pl
SF:ain;\x20charset=utf-8\r\nConnection:\x20close\r\n\r\n400\x20Bad\x20Requ
SF:est")%r(LDAPSearchReq,67,"HTTP/1\.1\x20400\x20Bad\x20Request\r\nContent
SF:-Type:\x20text/plain;\x20charset=utf-8\r\nConnection:\x20close\r\n\r\n4
SF:00\x20Bad\x20Request");
MAC Address: 08:00:27:63:D2:AC (Oracle VirtualBox virtual NIC)
Service Info: Hosts: 127.0.0.1, UBUNTU, irc.TestIRC.net; OSs: Unix, 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 87.70 seconds
```

2.3. More aggressive service detection

Here an aggressive scan is done using **-version-intensity.** Here -**sV** enables version detection which tries to determine the exact software and version running on each open port and the **--version-intensity** controls how aggressive Nmap is when probing services. The Scale is from 0 to 9 where 0 is the lightest and quickest. 9 is most aggressive. 5 is a balanced level which tries a decent detection without being too intrusive or slow.

```
┌──(kali㉿kali)-[~]
└─$ nmap -sV --version-intensity 5 192.168.1.30
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-03-30 22:44 EDT
Stats: 0:00:21 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 91.67% done; ETC: 22:45 (0:00:02 remaining)
Nmap scan report for 192.168.1.30
Host is up (0.00018s latency).
Not shown: 988 closed tcp ports (reset)
PORT      STATE SERVICE     VERSION
21/tcp    open  ftp         ProFTPD 1.3.5
22/tcp    open  ssh         OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.10 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http        Apache httpd 2.4.7
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
631/tcp   open  ipp         CUPS 1.7
3306/tcp  open  mysql       MySQL (unauthorized)
6667/tcp  open  irc         UnrealIRCd
8080/tcp  open  http        Jetty 8.1.7.v20120910
8181/tcp  open  http        WEBrick httpd 1.3.1 (Ruby 2.3.7 (2018-03-28))
10010/tcp open  rxapi?
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-Port10010-TCP:V=7.94SVN%I=5%D=3/30%Time=67EA01AB%P=x86_64-pc-linux-gnu%
SF:r(GenericLines,67,"HTTP/1\.1\x20400\x20Bad\x20Request\r\nContent-Type:\
SF:x20text/plain;\x20charset=utf-8\r\nConnection:\x20close\r\n\r\n400\x20B
SF:ad\x20Request")%r(GetRequest,8F,"HTTP/1\.0\x20404\x20Not\x20Found\r\nDa
SF:te:\x20Sun,\x2030\x20Mar\x202025\x2022:07:36\x20GMT\r\nContent-Length:\
SF:x2019\r\nContent-Type:\x20text/plain;\x20charset=utf-8\r\n\r\n404:\x20P
SF:age\x20Not\x20Found")%r(HTTPOptions,8F,"HTTP/1\.0\x20404\x20Not\x20Foun
SF:d\r\nDate:\x20Sun,\x2030\x20Mar\x202025\x2022:07:36\x20GMT\r\nContent-L
SF:ength:\x2019\r\nContent-Type:\x20text/plain;\x20charset=utf-8\r\n\r\n40
SF:4:\x20Page\x20Not\x20Found")%r(RTSPRequest,67,"HTTP/1\.1\x20400\x20Bad\
SF:x20Request\r\nContent-Type:\x20text/plain;\x20charset=utf-8\r\nConnecti
SF:on:\x20close\r\n\r\n400\x20Bad\x20Request")%r(Help,67,"HTTP/1\.1\x20400
SF:\x20Bad\x20Request\r\nContent-Type:\x20text/plain;\x20charset=utf-8\r\n
SF:Connection:\x20close\r\n\r\n400\x20Bad\x20Request")%r(SSLSessionReq,67,
SF:"HTTP/1\.1\x20400\x20Bad\x20Request\r\nContent-Type:\x20text/plain;\x20
SF:charset=utf-8\r\nConnection:\x20close\r\n\r\n400\x20Bad\x20Request")%r(
SF:TLSSessionReq,67,"HTTP/1\.1\x20400\x20Bad\x20Request\r\nContent-Type:\x
SF:20text/plain;\x20charset=utf-8\r\nConnection:\x20close\r\n\r\n400\x20Ba
SF:d\x20Request")%r(Kerberos,67,"HTTP/1\.1\x20400\x20Bad\x20Request\r\nCon
SF:tent-Type:\x20text/plain;\x20charset=utf-8\r\nConnection:\x20close\r\n\
SF:r\n400\x20Bad\x20Request")%r(SIPOptions,67,"HTTP/1\.1\x20400\x20Bad\x20
SF:Request\r\nContent-Type:\x20text/plain;\x20charset=utf-8\r\nConnection:
SF:\x20close\r\n\r\n400\x20Bad\x20Request");
MAC Address: 08:00:27:63:D2:AC (Oracle VirtualBox virtual NIC)
Service Info: Hosts: 127.0.0.1, UBUNTU, irc.TestIRC.net; OSs: Unix, 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 26.58 seconds
```

2.4. Lighter banner-grabbing detection

Here light banner grabbing is done.here the -**sV** performs version detection on detected services and the **--version-intensity** is set to 0 which is the lightest, least aggressive level of service probing. This sends very few probes, minimizing scan time and detection risk. This may also miss some service versions due to fewer probes.

```
┌──(kali㉿kali)-[~]
└─$ nmap -sV --version-intensity 0 192.168.1.30
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-03-30 22:57 EDT
Nmap scan report for 192.168.1.30
Host is up (0.00014s latency).
Not shown: 988 closed tcp ports (reset)
PORT      STATE SERVICE     VERSION
21/tcp    open  ftp         ProFTPD 1.3.5
22/tcp    open  ssh         OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.10 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http        Apache httpd 2.4.7
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
631/tcp   open  ipp         CUPS 1.7
3306/tcp  open  mysql       MySQL (unauthorized)
6667/tcp  open  irc         UnrealIRCd
8080/tcp  open  http        Jetty 8.1.7.v20120910
8181/tcp  open  http        WEBrick httpd 1.3.1 (Ruby 2.3.7 (2018-03-28))
10010/tcp open  rxapi?
MAC Address: 08:00:27:63:D2:AC (Oracle VirtualBox virtual NIC)
Service Info: Hosts: 127.0.0.1, UBUNTU, irc.TestIRC.net; OSs: Unix, 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 11.54 seconds

┌──(kali㉿kali)-[~]
└─$
```

2.5. Nmap output formats

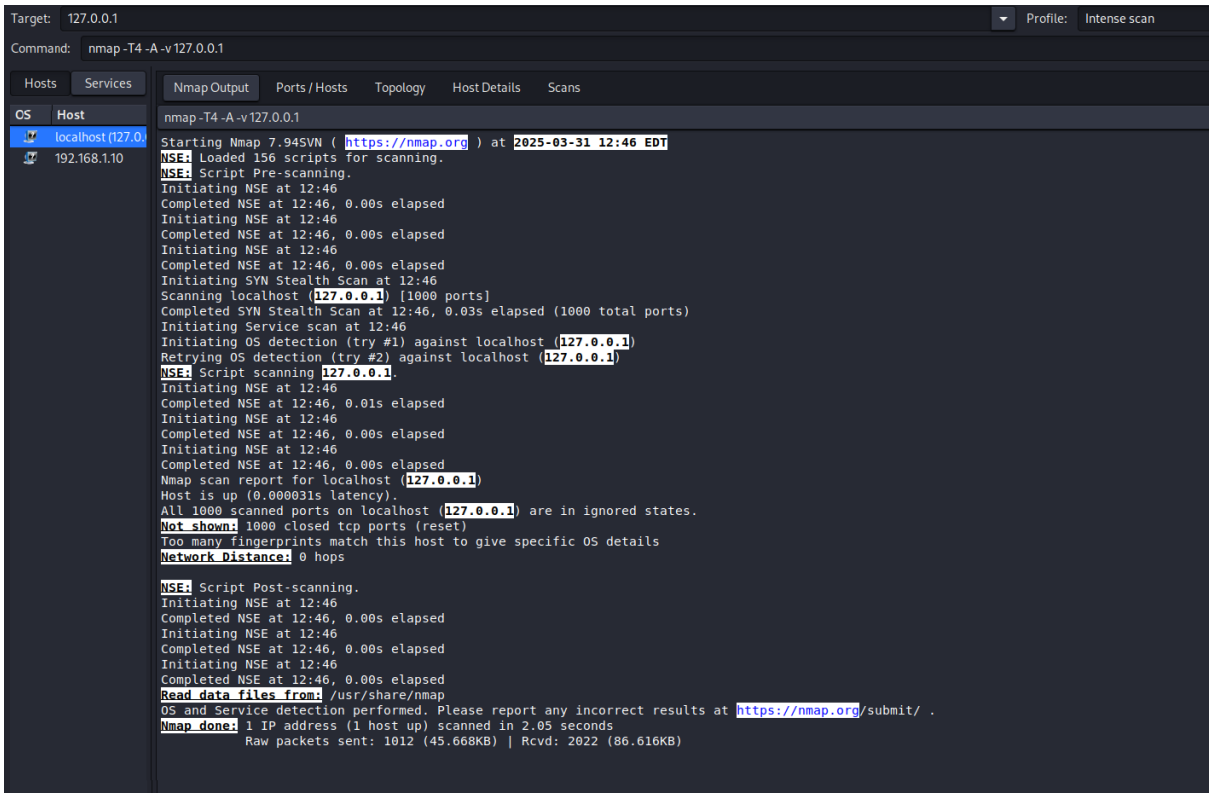
The **-oN** option tells Nmap to save the scan results in a readable text format to a file named output.txt.

```
┌──(kali㉿kali)-[~]
└─$ nmap -oN output.txt 192.168.1.30
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-03-30 22:58 EDT
Nmap scan report for 192.168.1.30
Host is up (0.00023s latency).
Not shown: 988 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
631/tcp   open  ipp
3306/tcp  open  mysql
6667/tcp  open  irc
8080/tcp  open  http-proxy
8181/tcp  open  intermapper
10010/tcp open  rxapi
MAC Address: 08:00:27:63:D2:AC (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.22 seconds

┌──(kali㉿kali)-[~]
└─$ ls
192.168.1.30   Documents   Music               'New Graph (1) (recovered at 2025-03-15 00-32-24).mtgl'   output.txt   pingScan_2.sh   Templates
Desktop        Downloads   'New Graph (1).mtgl'  'New Graph (2).mtgl'                                    Pictures     Public          Videos
```

Part II: Use Zenmap to Scan a Target Network

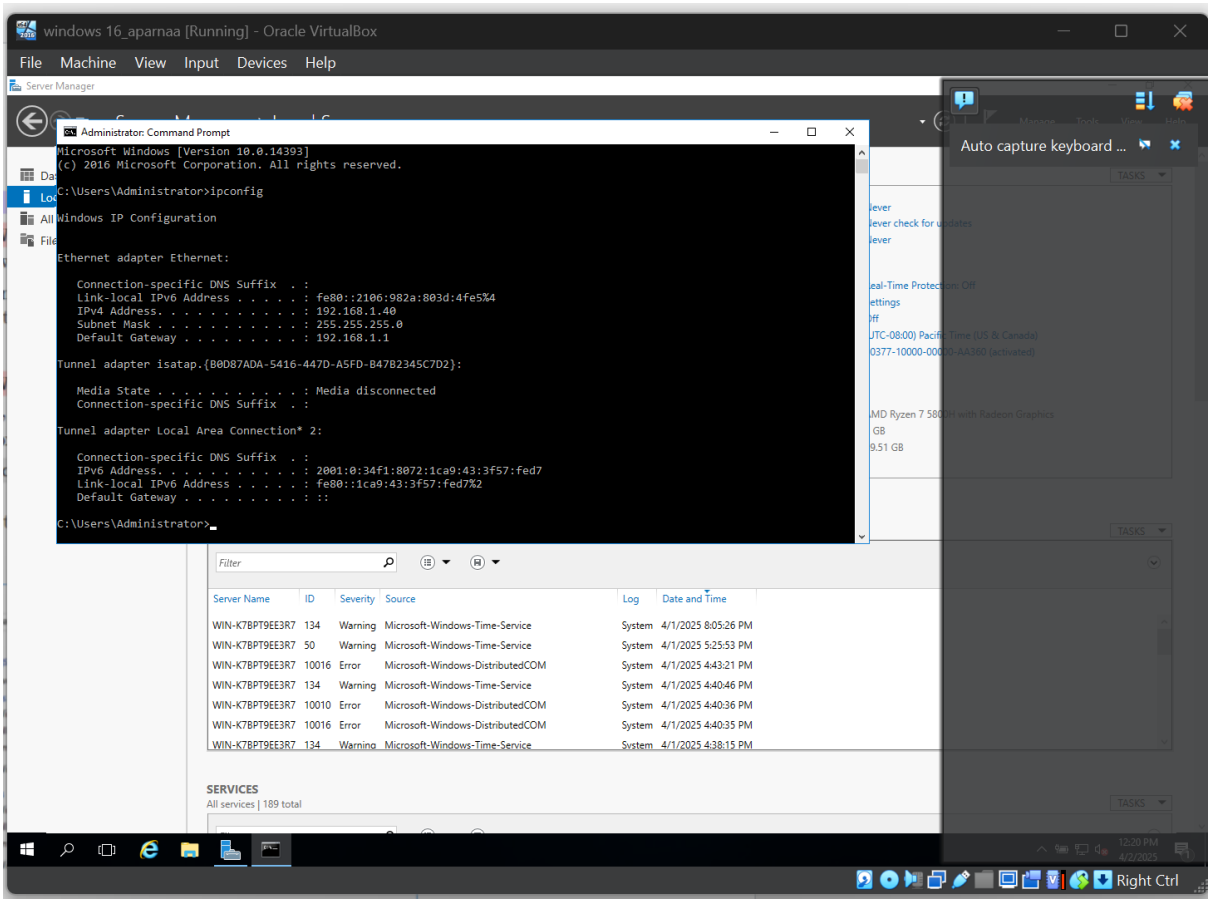
3.2. Port Scanning Your Own Linux Machine With zenmap.

Here we used Zenmap to perform an Intense Scan on the local Kali Linux machine (127.0.0.1). The scan results showed that all 1000 ports are closed, which is a usual behavior since Kali does not run network services by default. This means there are no active listeners on those ports, and the system responds with RST (reset) packets to all SYN requests.



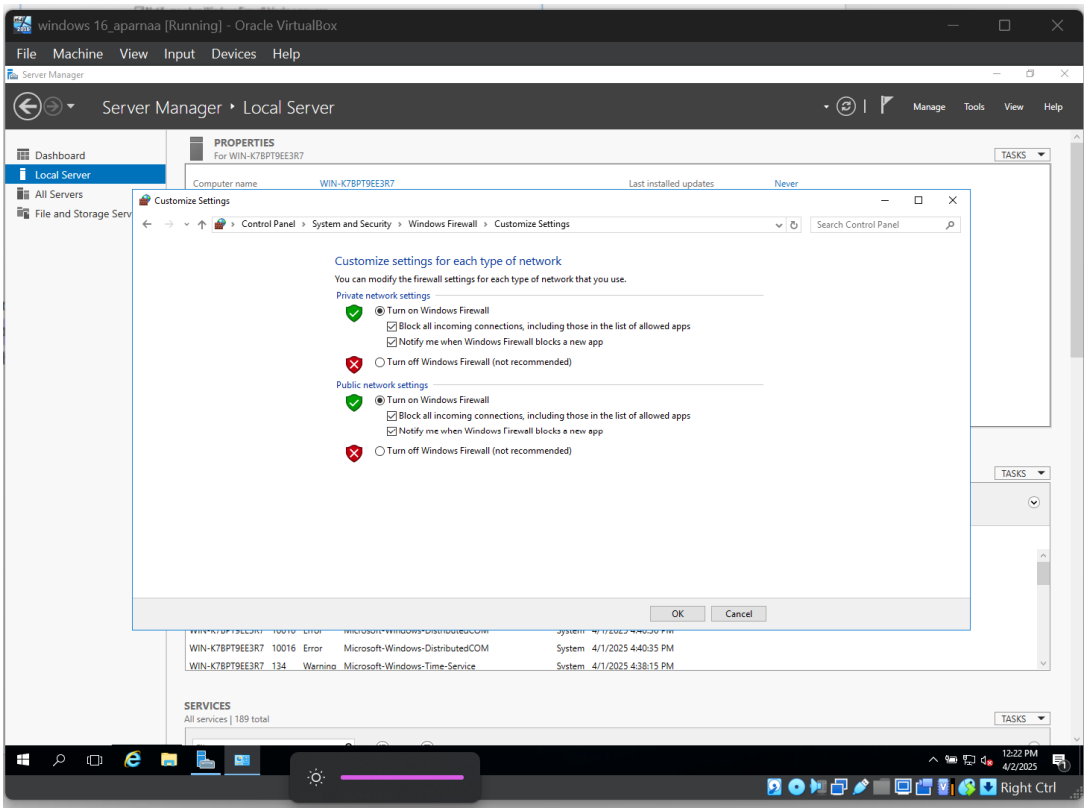
3.3. Finding the IP Address of our Windows VM

In this task, we opened the Command Prompt on the Windows Server 2016 VM and ran the ipconfig command to find the machine's IP address. The IP address identified was 192.168.1.40



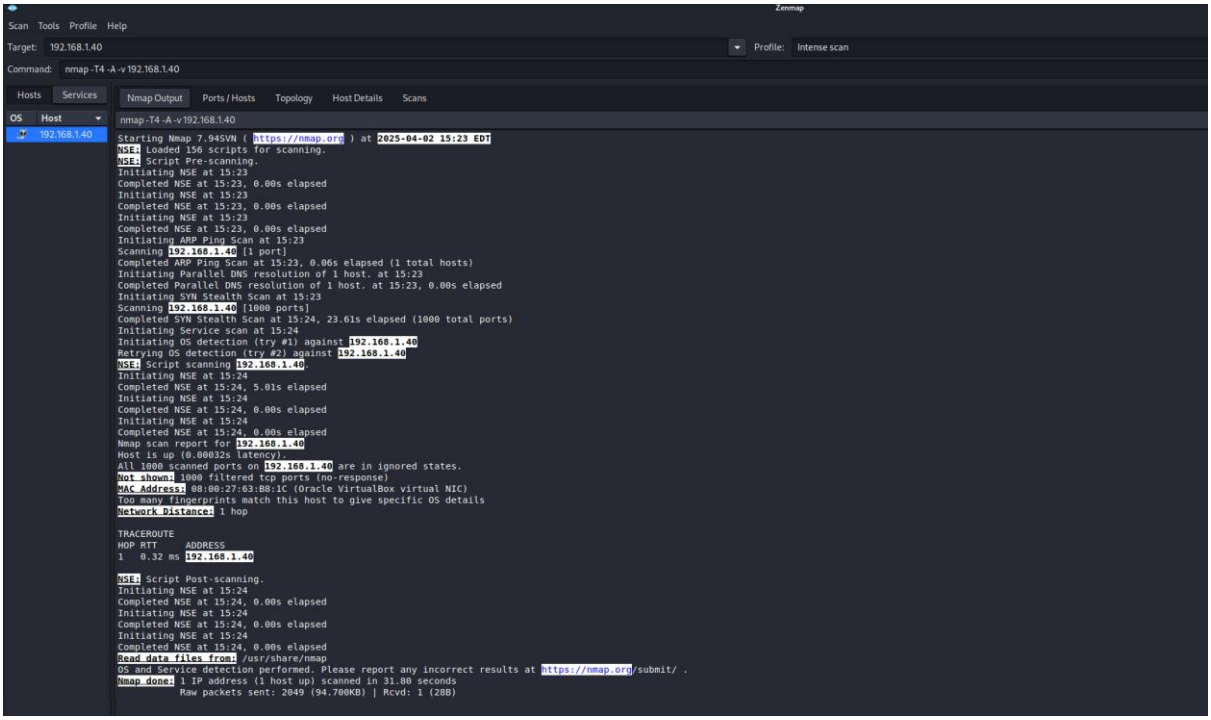
3.4. Setting Windows VM Firewall to Block All Incoming Connections

we configured the Windows Firewall on the Windows Server 2016 VM to block all incoming connections for both private and public networks.



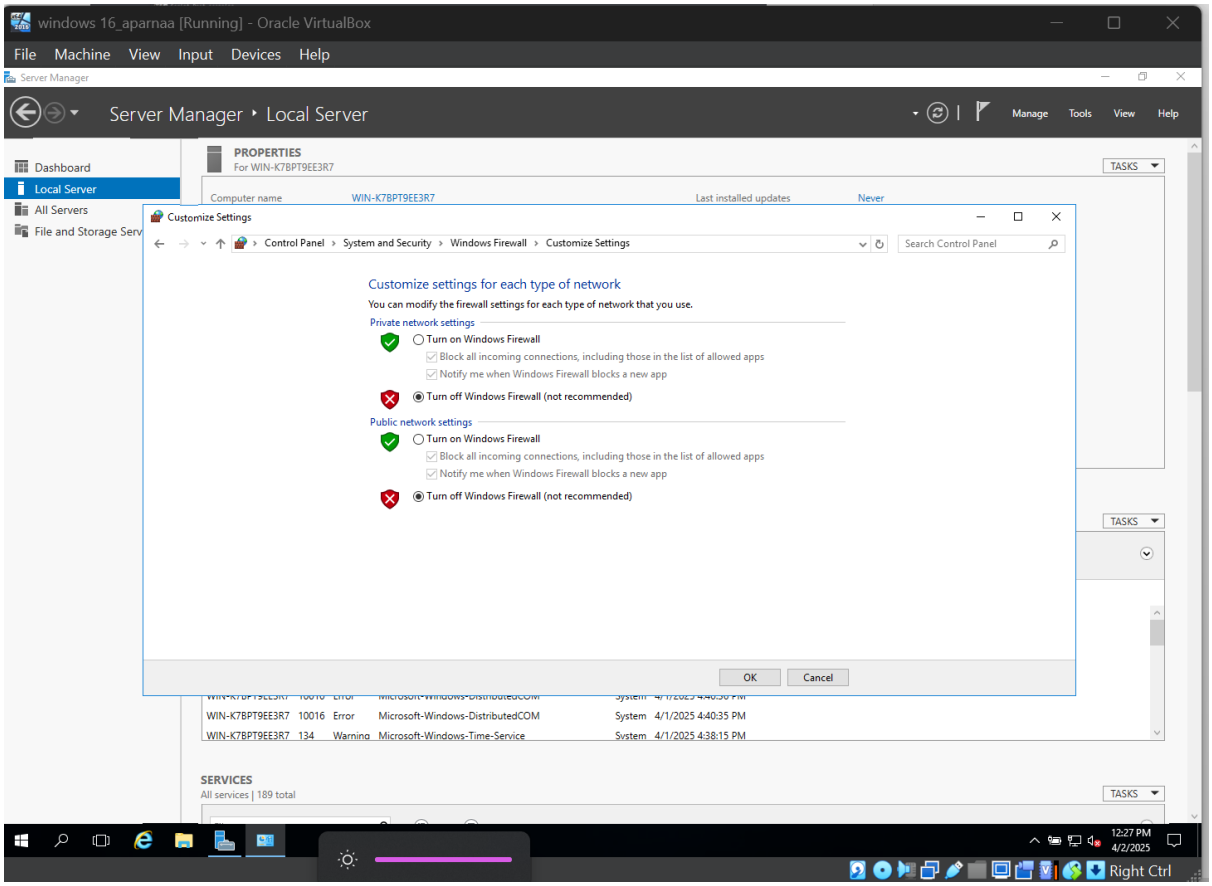
3.5. Scanning our Windows VM with Incoming Connections Blocked

we scanned the Windows VM (192.168.1.40) using Zenmap after enabling the firewall to block all incoming connections. The scan results show that all 1000 scanned ports are in ignored states. This behavior is expected when a firewall is configured to drop incoming traffic silently, causing the scanner to time out instead of receiving reset (RST) packets that would indicate closed ports.



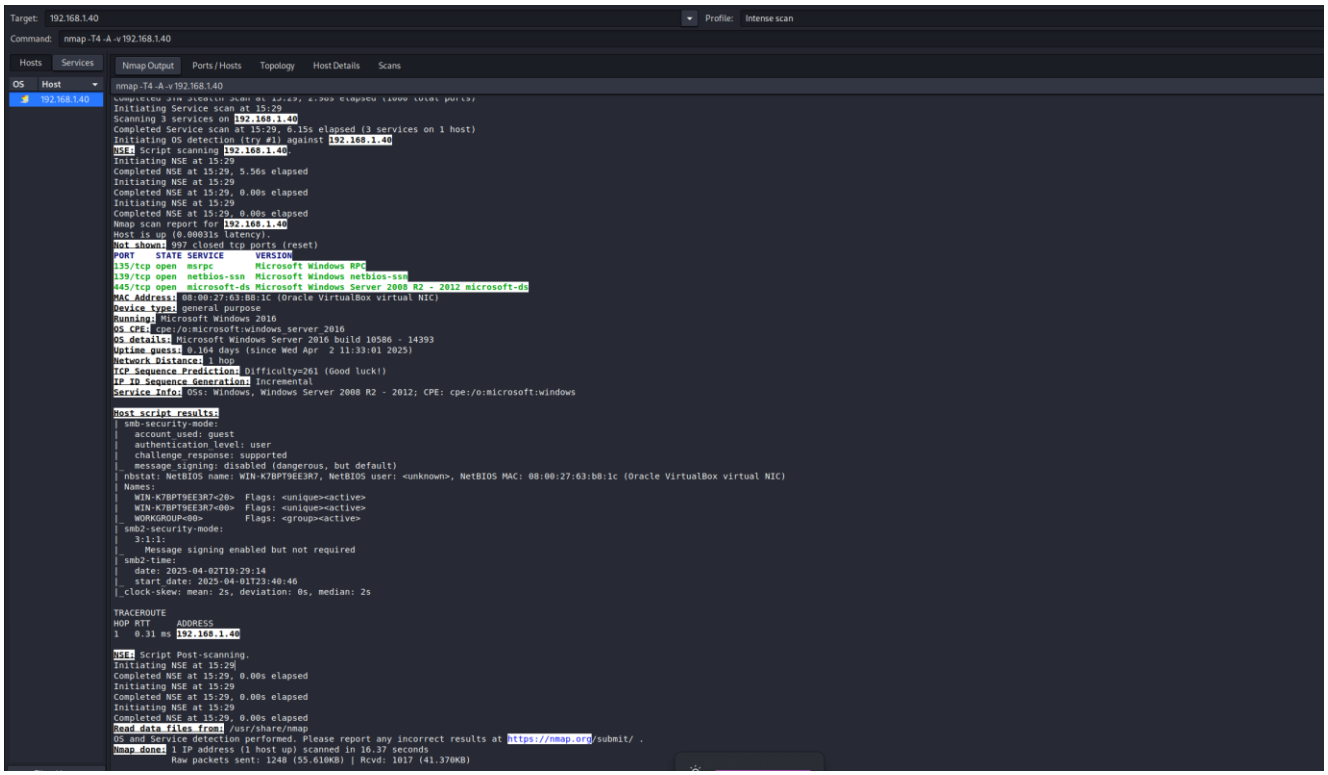
3.6. Setting Windows VM Firewall to OFF

The Windows Firewall on the Windows Server 2016 VM was turned off for both private and public network settings



3.7. Scanning our Windows VM With the Firewall Off

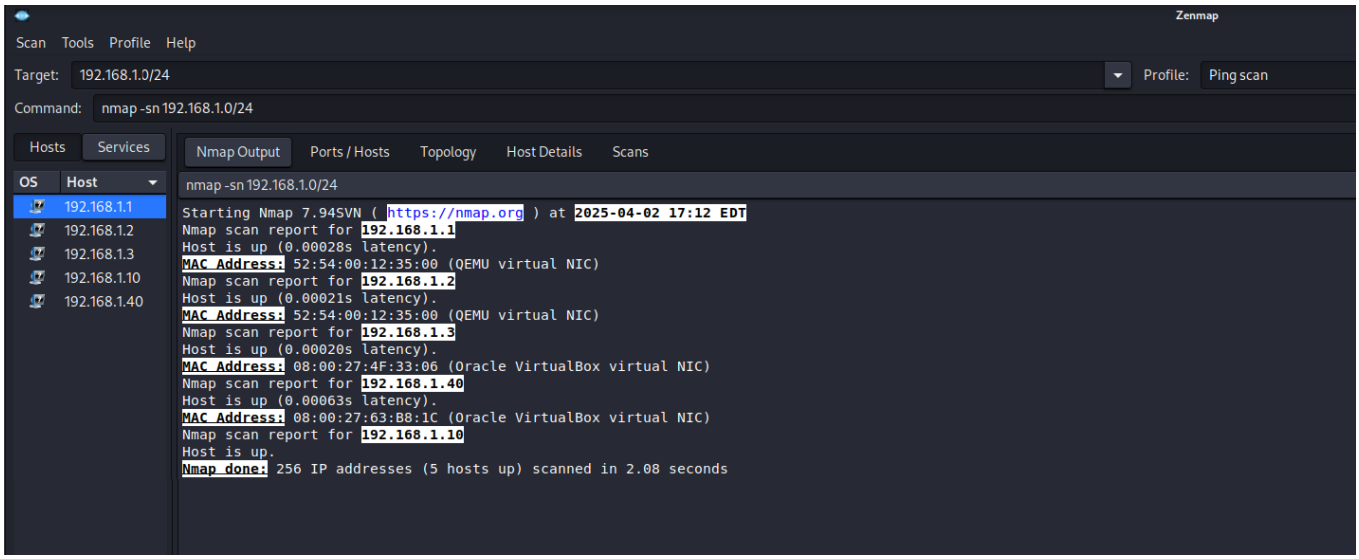
we scanned the Windows VM (192.168.1.40) using Zenmap after disabling the firewall, allowing incoming connections. The scan successfully detected multiple open ports, including 135, 139, and 445.



4. Part III: Analyzing a Port Scan

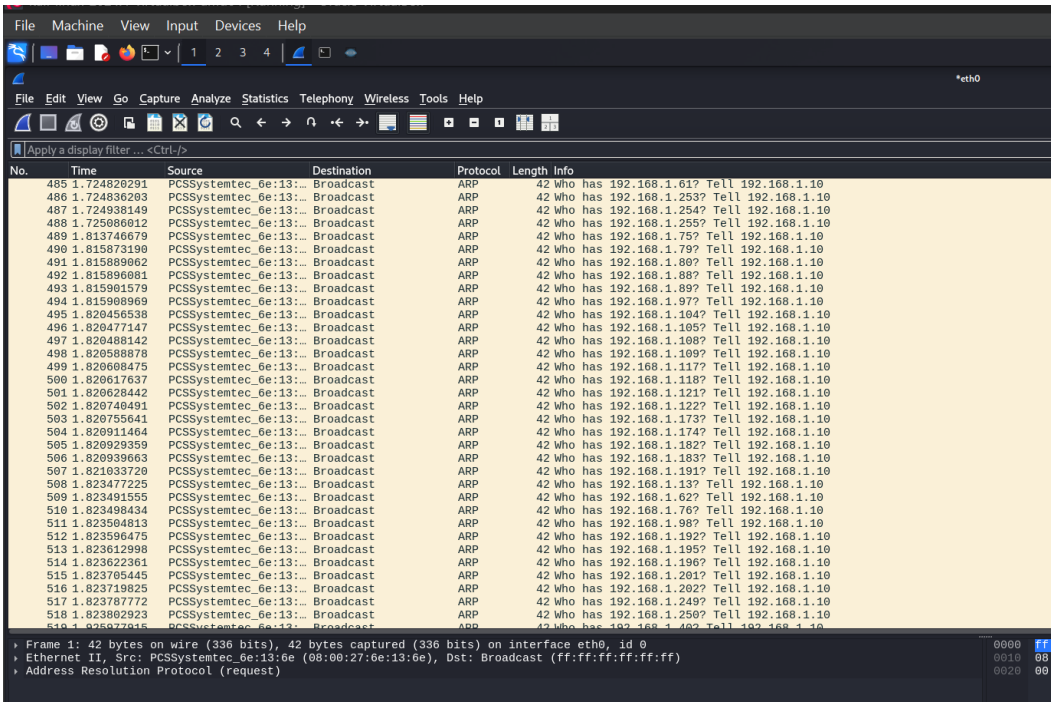
4.1. Starting The Wireshark Network Analyzer.

I used Zenmap with the Ping Scan profile to detect all active devices on the local network (192.168.1.0/24). After starting Wireshark to monitor traffic and running the scan, multiple hosts were discovered, including the Kali Linux VM, Windows VM, and other available devices



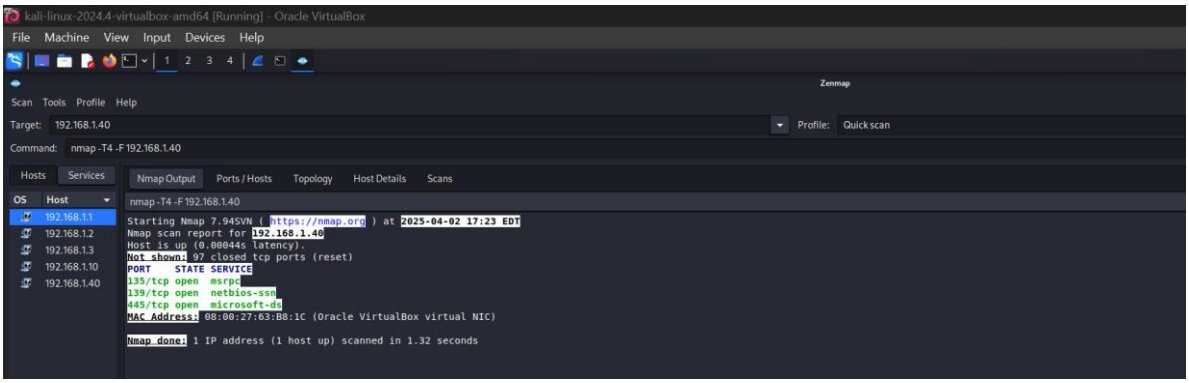
4.3. Using Wireshark to Analyze the Ping Sweep

Here nmap performed a ping sweep using ARP requests to detect active hosts on a local network. In Wireshark, this is seen as multiple ARP "Who has" broadcasts, showing Nmap querying each IP address. This method is used instead of ICMP because ARP works reliably within local networks.

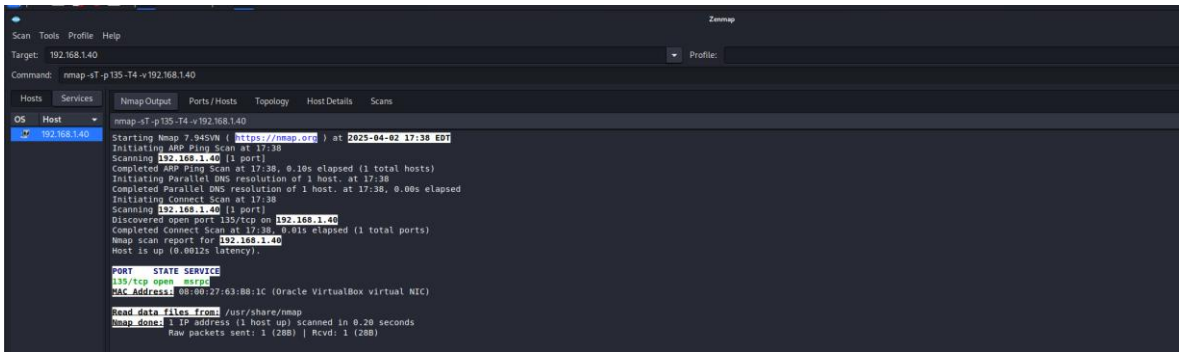


4.4. Performing a Quick Scan of the Windows Machine

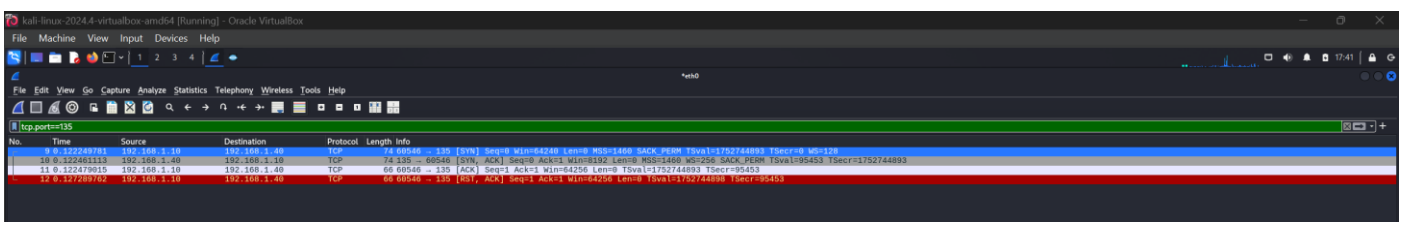
I have scanned windows machine using zenmap using quick scan. Here the port 135 is open along with other ports like 139,445



The next command used is **nmap -p 135 -T4 -v 192.168.1.40** . This is used in zenmap after starting wireshark. This command scans 192.168.1.40 for the ports 135. This does connect scan.



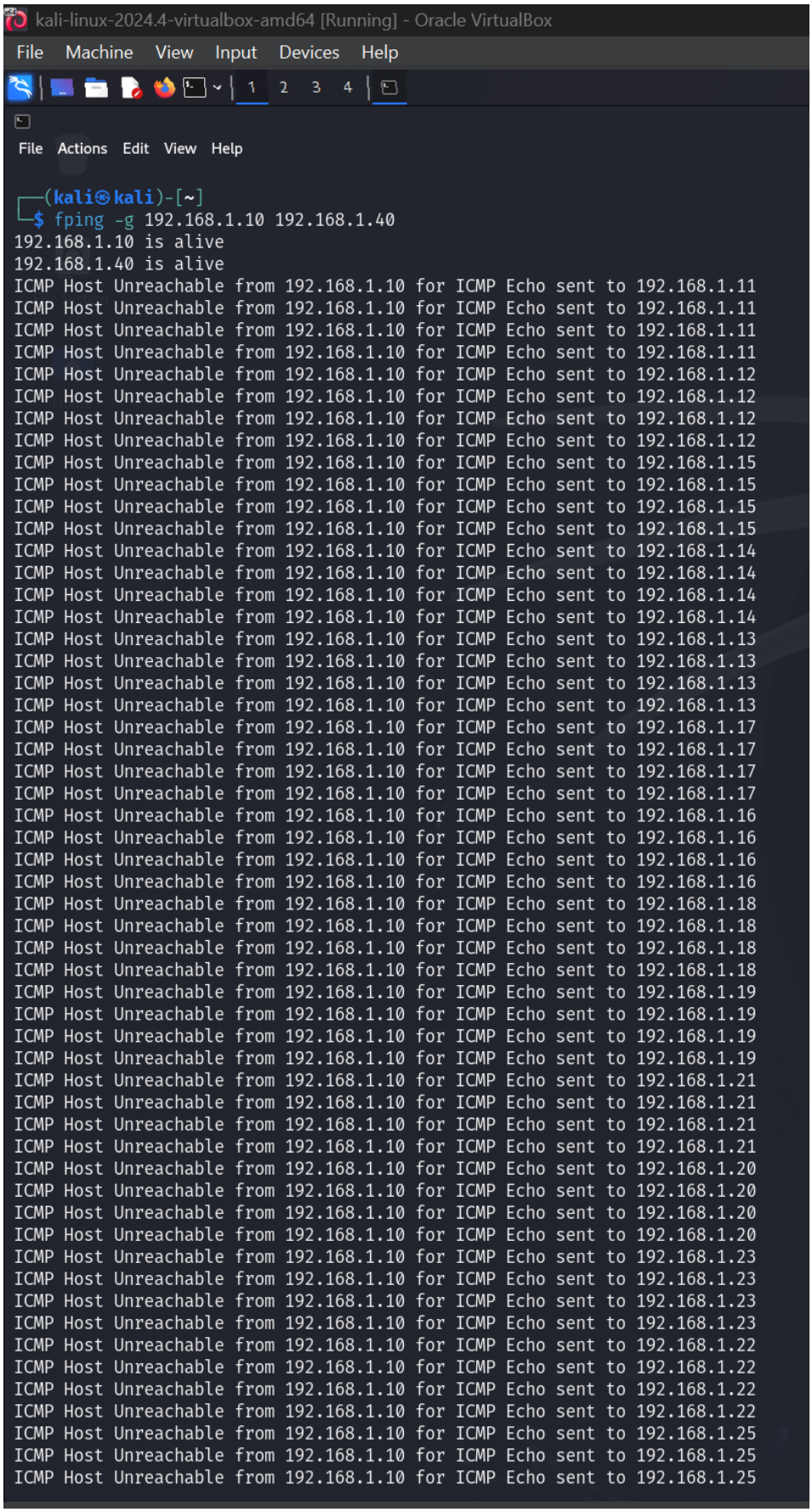
In wireshark window, in the filter box **tcp.port==135** is used filter out only the port 135. This shows a complete three way handshake followed by a rst to end the session.



5. Part V: Crafting IP Packets with Fping and Hping

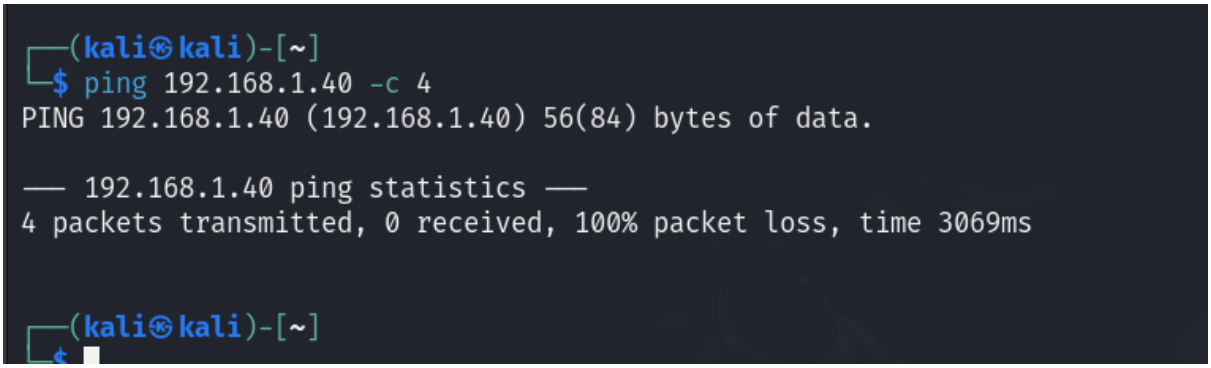
5.1. fping.

The fping -g command is used to send ICMP echo requests to a range of IP addresses (192.168.1.10 to 192.168.1.40) to identify live hosts. The output shows that only 192.168.1.10 and 192.168.1.40 responded as alive, while the rest were marked unreachable due to no ICMP reply.

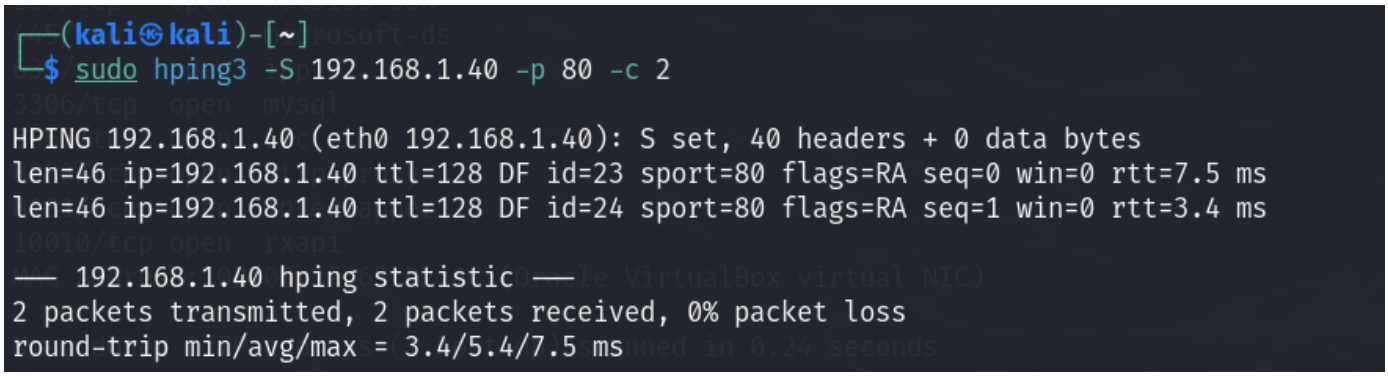


5.2. Hping3.

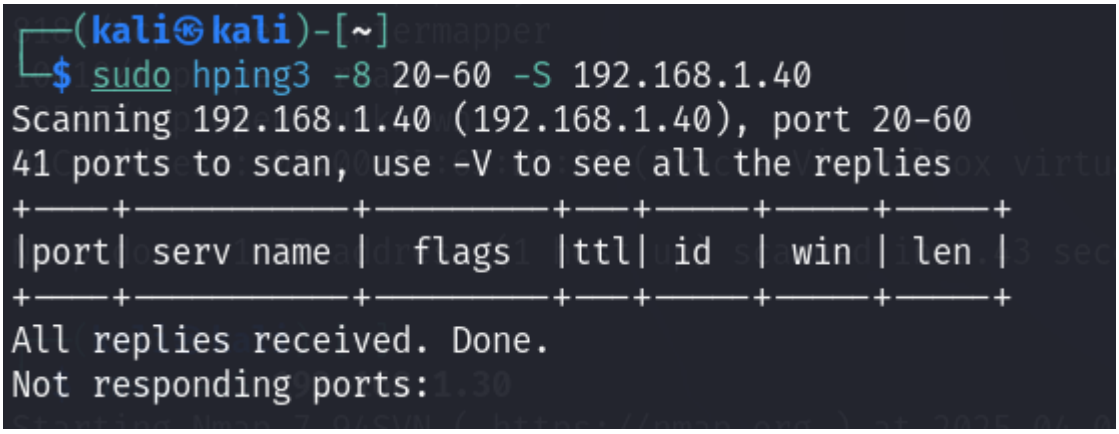
1 –The ping command sent four ICMP echo requests to 192.168.1.40, but all packets were lost, indicating no response from the target. This is likely because the target has ICMP responses disabled. As a result, traditional ping cannot confirm if the host is up.



2,3,4,5 – we can see we have received successful responses from our target. This means that the 192.168.1.40 device is online and that port 80 is open



6,7- Here we perform port scan on a range of network ports on a target device. Using the **hping3 -8 20-60 -S 192.168.1.40** command



8,9,10,11,12- here by analysing tcpdumpp and hping3 side by side we can see that tcpdump showed packet headers and responses confirming packet delivery. Additional ACK/FIN scans showed different traffic patterns, useful for stealth scanning.

```
zsh: corrupt history file /home/kali/.zsh_history
┌──(kali㉿kali)-[~]
└─$ sudo hping3 -8 20-80 -S 192.168.1.40
[sudo] password for kali:
Scanning 192.168.1.40 (192.168.1.40), port 20-80
61 ports to scan, use -V to see all the replies
+----+-----------+---------+---+-----+-----+-----+
|port| serv name |  flags  |ttl| id  | win | len |
+----+-----------+---------+---+-----+-----+-----+
All replies received. Done.
Not responding ports:
```

```
┌──(kali㉿kali)-[~]
└─$ sudo tcpdump
[sudo] password for kali:
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
16:54:35.879803 IP 192.168.1.10.1395 > 192.168.1.40.ftp-data: Flags [S], seq 845492249, win 512, length 0
16:54:35.879836 IP 192.168.1.10.1395 > 192.168.1.40.ftp: Flags [S], seq 1345402690, win 512, length 0
16:54:35.879843 IP 192.168.1.10.1395 > 192.168.1.40.ssh: Flags [S], seq 499934757, win 512, length 0
16:54:35.879859 IP 192.168.1.10.1395 > 192.168.1.40.telnet: Flags [S], seq 2008805173, win 512, length 0
16:54:35.879866 IP 192.168.1.10.1395 > 192.168.1.40.24: Flags [S], seq 1635766179, win 512, length 0
16:54:35.879871 IP 192.168.1.10.1395 > 192.168.1.40.smtp: Flags [S], seq 504618077, win 512, length 0
16:54:35.879876 IP 192.168.1.10.1395 > 192.168.1.40.26: Flags [S], seq 261991468, win 512, length 0
16:54:35.879881 IP 192.168.1.10.1395 > 192.168.1.40.27: Flags [S], seq 1478187756, win 512, length 0
16:54:35.879886 IP 192.168.1.10.1395 > 192.168.1.40.28: Flags [S], seq 507056283, win 512, length 0
16:54:35.879891 IP 192.168.1.10.1395 > 192.168.1.40.29: Flags [S], seq 1690664391, win 512, length 0
16:54:35.879896 IP 192.168.1.10.1395 > 192.168.1.40.30: Flags [S], seq 382259115, win 512, length 0
16:54:35.879901 IP 192.168.1.10.1395 > 192.168.1.40.31: Flags [S], seq 864855079, win 512, length 0
16:54:35.879907 IP 192.168.1.10.1395 > 192.168.1.40.32: Flags [S], seq 1896834217, win 512, length 0
16:54:35.879986 IP 192.168.1.10.1395 > 192.168.1.40.33: Flags [S], seq 286259085, win 512, length 0
16:54:35.879992 IP 192.168.1.10.1395 > 192.168.1.40.34: Flags [S], seq 1107140518, win 512, length 0
16:54:35.879993 IP 192.168.1.10.1395 > 192.168.1.40.35: Flags [S], seq 383802083, win 512, length 0
16:54:35.879999 IP 192.168.1.10.1395 > 192.168.1.40.36: Flags [S], seq 1858280102, win 512, length 0
16:54:35.880021 IP 192.168.1.10.1395 > 192.168.1.40.time: Flags [S], seq 1809628549, win 512, length 0
16:54:35.880028 IP 192.168.1.10.1395 > 192.168.1.40.38: Flags [S], seq 1317436551, win 512, length 0
16:54:35.880059 IP 192.168.1.10.1395 > 192.168.1.40.39: Flags [S], seq 1741963377, win 512, length 0
16:54:35.880067 IP 192.168.1.10.1395 > 192.168.1.40.40: Flags [S], seq 351428778, win 512, length 0
16:54:35.880094 IP 192.168.1.10.1395 > 192.168.1.40.41: Flags [S], seq 1165341490, win 512, length 0
16:54:35.880102 IP 192.168.1.10.1395 > 192.168.1.40.42: Flags [S], seq 34270849, win 512, length 0
16:54:35.880123 IP 192.168.1.10.1395 > 192.168.1.40.whois: Flags [S], seq 1650510039, win 512, length 0
16:54:35.880151 IP 192.168.1.10.1395 > 192.168.1.40.44: Flags [S], seq 1457183580, win 512, length 0
16:54:35.880178 IP 192.168.1.10.1395 > 192.168.1.40.45: Flags [S], seq 1530848366, win 512, length 0
16:54:35.880206 IP 192.168.1.10.1395 > 192.168.1.40.46: Flags [S], seq 429804464, win 512, length 0
16:54:35.880245 IP 192.168.1.10.1395 > 192.168.1.40.47: Flags [S], seq 258297398, win 512, length 0
16:54:35.880262 IP 192.168.1.40.ftp-data > 192.168.1.10.1395: Flags [R.], seq 0, ack 845492250, win 0, length
16:54:35.880263 IP 192.168.1.40.ftp > 192.168.1.10.1395: Flags [R.], seq 0, ack 1345402691, win 0, length 0
16:54:35.880263 IP 192.168.1.40.ssh > 192.168.1.10.1395: Flags [R.], seq 0, ack 499934758, win 0, length 0
16:54:35.880263 IP 192.168.1.40.telnet > 192.168.1.10.1395: Flags [R.], seq 0, ack 2008805174, win 0, length 0
16:54:35.880264 IP 192.168.1.40.24 > 192.168.1.10.1395: Flags [R.], seq 0, ack 1635766180, win 0, length 0
16:54:35.880264 IP 192.168.1.40.smtp > 192.168.1.10.1395: Flags [R.], seq 0, ack 504618078, win 0, length 0
16:54:35.880265 IP 192.168.1.40.26 > 192.168.1.10.1395: Flags [R.], seq 0, ack 261991469, win 0, length 0
16:54:35.880265 IP 192.168.1.40.27 > 192.168.1.10.1395: Flags [R.], seq 0, ack 1478187757, win 0, length 0
16:54:35.880281 IP 192.168.1.40.28 > 192.168.1.10.1395: Flags [R.], seq 0, ack 507056284, win 0, length 0
16:54:35.880282 IP 192.168.1.40.29 > 192.168.1.10.1395: Flags [R.], seq 0, ack 1690664392, win 0, length 0
16:54:35.880282 IP 192.168.1.40.30 > 192.168.1.10.1395: Flags [R.], seq 0, ack 382259116, win 0, length 0
16:54:35.880282 IP 192.168.1.40.31 > 192.168.1.10.1395: Flags [R.], seq 0, ack 864855080, win 0, length 0
16:54:35.880283 IP 192.168.1.40.32 > 192.168.1.10.1395: Flags [R.], seq 0, ack 1896834218, win 0, length 0
16:54:35.880283 IP 192.168.1.40.33 > 192.168.1.10.1395: Flags [R.], seq 0, ack 286259086, win 0, length 0
16:54:35.880283 IP 192.168.1.40.34 > 192.168.1.10.1395: Flags [R.], seq 0, ack 1107140519, win 0, length 0
16:54:35.880284 IP 192.168.1.40.35 > 192.168.1.10.1395: Flags [R.], seq 0, ack 383802084, win 0, length 0
16:54:35.880296 IP 192.168.1.40.36 > 192.168.1.10.1395: Flags [R.], seq 0, ack 1858280103, win 0, length 0
16:54:35.880296 IP 192.168.1.40.time > 192.168.1.10.1395: Flags [R.], seq 0, ack 1809628550, win 0, length 0
16:54:35.880297 IP 192.168.1.40.38 > 192.168.1.10.1395: Flags [R.], seq 0, ack 1317436552, win 0, length 0
16:54:35.880307 IP 192.168.1.10.1395 > 192.168.1.40.48: Flags [S], seq 623146432, win 512, length 0
16:54:35.880314 IP 192.168.1.10.1395 > 192.168.1.40.tacacs: Flags [S], seq 654230817, win 512, length 0
16:54:35.880341 IP 192.168.1.10.1395 > 192.168.1.40.50: Flags [S], seq 1794292688, win 512, length 0
16:54:35.880366 IP 192.168.1.40.39 > 192.168.1.10.1395: Flags [R.], seq 0, ack 1741963378, win 0, length 0
16:54:35.880367 IP 192.168.1.40.40 > 192.168.1.10.1395: Flags [R.], seq 0, ack 351428779, win 0, length 0
16:54:35.880367 IP 192.168.1.40.41 > 192.168.1.10.1395: Flags [R.], seq 0, ack 1165341491, win 0, length 0
```

13,14-

SYN Packet (Standard scan)

Received RA (RST + ACK) flags in response.This means the port is closed or actively rejecting connections, but the host is alive and responding.

```
┌──(kali㉿kali)-[~]
└─$ sudo hping3 -S 192.168.1.40 -p 80 -c 3

HPING 192.168.1.40 (eth0 192.168.1.40): S set, 40 headers + 0 data bytes
len=46 ip=192.168.1.40 ttl=128 DF id=336 sport=80 flags=RA seq=0 win=0 rtt=3.3 ms
len=46 ip=192.168.1.40 ttl=128 DF id=337 sport=80 flags=RA seq=1 win=0 rtt=7.1 ms
len=46 ip=192.168.1.40 ttl=128 DF id=338 sport=80 flags=RA seq=2 win=0 rtt=3.1 ms

— 192.168.1.40 hping statistic —
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 3.1/4.5/7.1 ms
```

ACK Packet (Firewall Mapping / Stealth)

Got R (RST) responses. This shows the firewall is stateless or unfiltered for ACK packets. Useful to probe firewall rule behavior or map network topology.

```
┌──(kali㉿kali)-[~]
└─$ sudo hping3 -A 192.168.1.40 -p 80 -c 3

HPING 192.168.1.40 (eth0 192.168.1.40): A set, 40 headers + 0 data bytes
len=46 ip=192.168.1.40 ttl=128 DF id=339 sport=80 flags=R seq=0 win=0 rtt=3.4 ms
len=46 ip=192.168.1.40 ttl=128 DF id=340 sport=80 flags=R seq=1 win=0 rtt=6.1 ms
len=46 ip=192.168.1.40 ttl=128 DF id=341 sport=80 flags=R seq=2 win=0 rtt=1.5 ms

— 192.168.1.40 hping statistic —
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 1.5/3.7/6.1 ms
```

3. FIN Packet (Used in stealth scans)

Also received RA (RST + ACK) flags. Indicates that port 80 is closed, and the system is rejecting FIN packets with a reset

```
┌──(kali㉿kali)-[~]
└─$ sudo hping3 -F 192.168.1.40 -p 80 -c 3
HPING 192.168.1.40 (eth0 192.168.1.40): F set, 40 headers + 0 data bytes
len=46 ip=192.168.1.40 ttl=128 DF id=342 sport=80 flags=RA seq=0 win=0 rtt=6.9 ms
len=46 ip=192.168.1.40 ttl=128 DF id=343 sport=80 flags=RA seq=1 win=0 rtt=7.1 ms
len=46 ip=192.168.1.40 ttl=128 DF id=344 sport=80 flags=RA seq=2 win=0 rtt=6.6 ms

— 192.168.1.40 hping statistic —
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 6.6/6.9/7.1 ms
```

- When the target computer (192.168.1.40) received a SYN packet, it responded with RA (RST + ACK), indicating the port (80) is closed but reachable. This shows the host is alive, but the service is not accepting connections.
- Sending an ACK packet resulted in an R (RST) response, confirming that the system is not stateful and is treating ACKs as invalid. This behavior helps identify the presence of a firewall
- A FIN packet, often used in stealth scans, also triggered an RA (RST + ACK) response from the Windows target. This is expected, as Windows systems usually respond to FIN probes on closed ports with RSTs.

To further explore, I sent the same packets (SYN/ACK/FIN) to other ports like 135.

- The SYN packet received a SYN-ACK response, indicating the port is open and the service is listening.
- In contrast, ACK packets received RST replies (no session exists)
- FIN packets triggered RST-ACK responses

These results validate that port 135 is open. This confirms that sending the same packets to different ports did change the response from the attacked computer.

When I sent SYN, ACK, and FIN packets to **port 80** (which were closed), the target responded with either RST or RST+ACK, indicating that the ports were not open and not accepting connections.

However, when I sent the same types of packets to **port 135**, the target responded differently - a SYN packet to port 135 triggered a SYN-ACK response and the ACK and FIN packets still resulted in RST or RST+ACK, but the response time confirms that the system was handling open port traffic differently

These changes in response help reveal which ports are **open, closed, or filtered**.