

Lab 4: Scan beyond IDS and Firewall

Lab Scenario

As a professional ethical hacker or a pen tester, the next step after discovering the OS of the target IP address(es) is to perform network scanning without being detected by the network security perimeters such as the firewall and IDS. IDSs and firewalls are efficient security mechanisms; however, they still have some security limitations. You may be required to launch attacks to exploit these limitations using various IDS/firewall evasion techniques such as packet fragmentation, source routing, IP address spoofing, etc. Scanning beyond the IDS and firewall allows you to evaluate the target network's IDS and firewall security.

Lab Objectives

- Scan beyond IDS/firewall using various evasion techniques

Overview of Scanning beyond IDS and Firewall

An Intrusion Detection System (IDS) and firewall are the security mechanisms intended to prevent an unauthorized person from accessing a network. However, even IDSs and firewalls have some security limitations. Firewalls and IDSs intend to avoid malicious traffic (packets) from entering into a network, but certain techniques can be used to send intended packets to the target and evade IDSs/firewalls.

Techniques to evade IDS/firewall:

- **Packet Fragmentation:** Send fragmented probe packets to the intended target, which re-assembles it after receiving all the fragments
- **Source Routing:** Specifies the routing path for the malformed packet to reach the intended target
- **Source Port Manipulation:** Manipulate the actual source port with the common source port to evade IDS/firewall
- **IP Address Decoy:** Generate or manually specify IP addresses of the decoys so that the IDS/firewall cannot determine the actual IP address
- **IP Address Spoofing:** Change source IP addresses so that the attack appears to be coming in as someone else
- **Creating Custom Packets:** Send custom packets to scan the intended target beyond the firewalls
- **Randomizing Host Order:** Scan the number of hosts in the target network in a random order to scan the intended target that is lying beyond the firewall
- **Sending Bad Checksums:** Send the packets with bad or bogus TCP/UDP checksums to the intended target
- **Proxy Servers:** Use a chain of proxy servers to hide the actual source of a scan and evade certain IDS/firewall restrictions

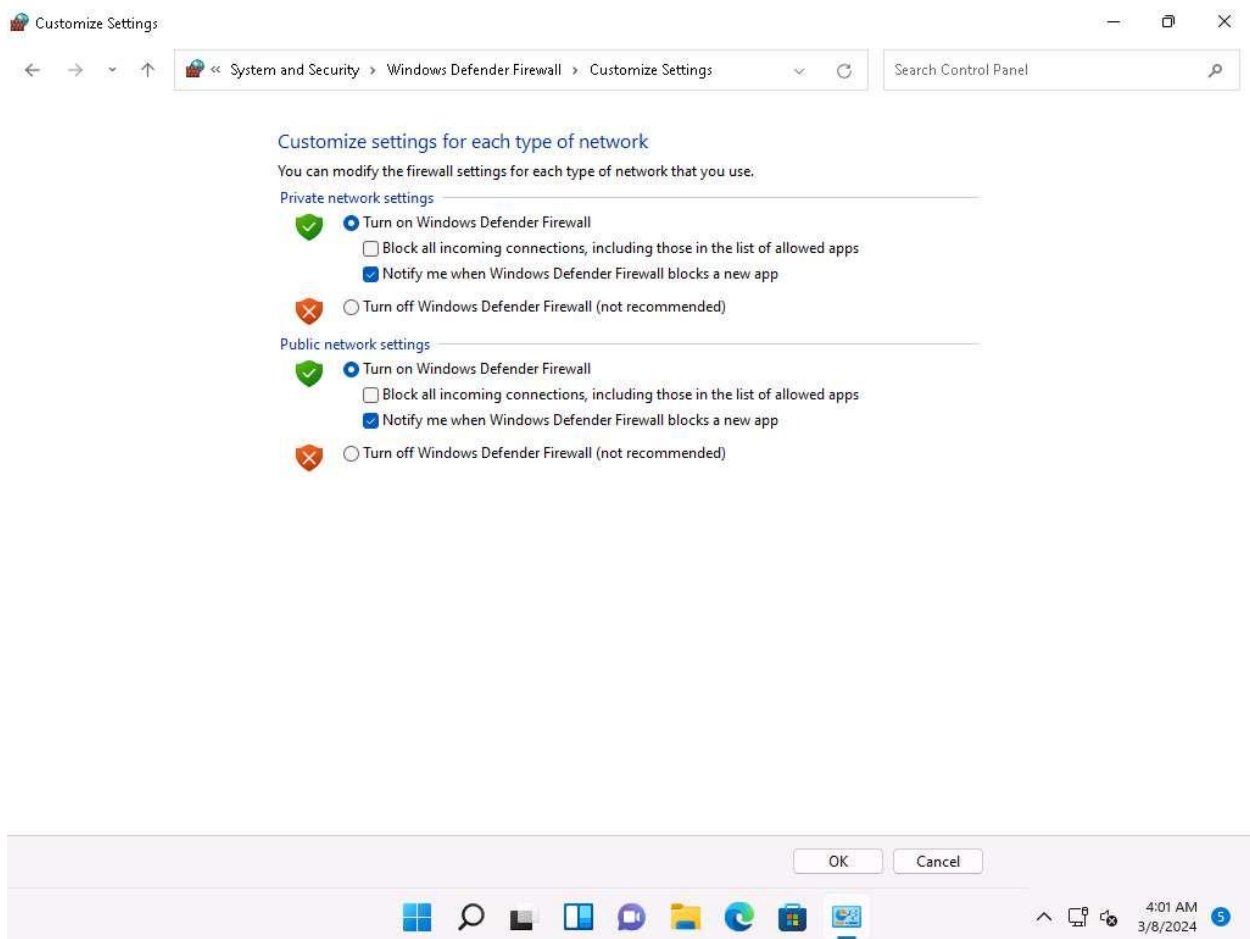
- **Anonymizers:** Use anonymizers that allow them to bypass Internet censors and evade certain IDS and firewall rules


Task 1: Scan beyond IDS/Firewall using various Evasion Techniques

Nmap offers many features to help understand complex networks with enabled security mechanisms and supports mechanisms for bypassing poorly implemented defenses. Using Nmap, various techniques can be implemented, which can bypass the IDS/firewall security mechanisms.

Here, we will use Nmap to evade IDS/firewall using various techniques such as packet fragmentation, source port manipulation, MTU, and IP address decoy.

1. Click [Windows 11](#) to switch to the **Windows 11** machine.
2. Navigate to **Control Panel --> System and Security --> Windows Defender Firewall --> Turn Windows Defender Firewall on or off**, enable Windows Defender Firewall and click **OK**, as shown in the screenshot.



3. Minimize the **Control Panel** window, click windows **Search** icon () on the **Desktop**. Search for **wireshark** in the search field and click **Open** to launch it.

4. The **Wireshark Network Analyzer** window appears, start capturing packets by double-clicking the available ethernet or interface (here, **Ethernet**).

If **Software Update** window appears, click **Remind me later**.

5. Click [Parrot Security](#) to switch to the **Parrot Security** machine. Open a **Terminal** window and execute **sudo su** to run the programs as a root user (When prompted, enter the password **toor**).

The password that you type will not be visible.

6. Now, run **cd** command to jump to the root directory.
7. In the terminal window, run **nmap -f [Target IP Address]** command, (here, the target machine is **Windows 11 [10.10.1.11]**).

-f switch is used to split the IP packet into tiny fragment packets.

Packet fragmentation refers to the splitting of a probe packet into several smaller packets (fragments) while sending it to a network. When these packets reach a host, IDSs and firewalls behind the host generally queue all of them and process them one by one. However, since this method of processing involves greater CPU consumption as well as network resources, the configuration of most of IDSs makes it skip fragmented packets during port scans.

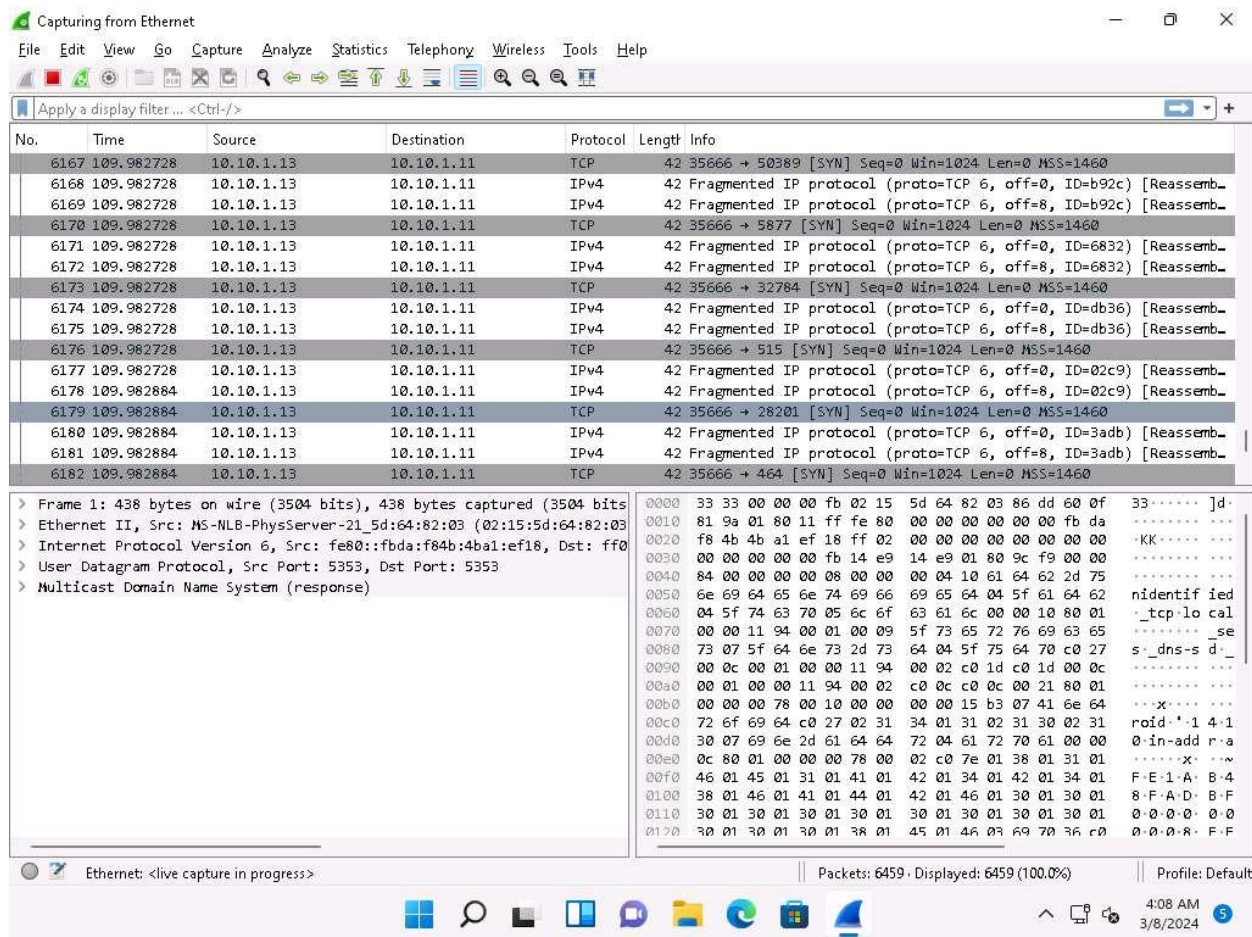
[more...](#)

8. Although **Windows Defender Firewall** is turned on in the target system (here, **Windows 11**), you can still obtain the results displaying all open TCP ports along with the name of services running on the ports, as shown in the screenshot.

```
Applications Places System [Icons] [Terminal] [Help]
nmap -f 10.10.1.11 - Parrot Terminal
File Edit View Search Terminal Help
[attacker@parrot]~$ sudo su
[sudo] password for attacker:
[root@parrot]~/home/attacker# cd
[root@parrot]~$ #nmap -f 10.10.1.11
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-08 07:07 EST
Nmap scan report for 10.10.1.11
Host is up (0.00099s latency).
Not shown: 995 filtered tcp ports (no-response)
PORT      STATE SERVICE
80/tcp    open  http
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
3389/tcp   open  ms-wbt-server
MAC Address: 00:15:5D:01:80:00 (Microsoft)

Nmap done: 1 IP address (1 host up) scanned in 5.12 seconds
[root@parrot]~$ #
```

9. Click [Windows 11](#) to switch to the **Windows 11** machine (target machine). You can observe the fragmented packets captured by the Wireshark, as shown in the screenshot.



10. Click [Parrot Security](#) to switch to the Parrot Security machine.

11. In the **Parrot Terminal** window, run **nmap -g 80 [Target IP Address]** command, (here, target IP address is **10.10.1.11**).

In this command, you can use the **-g** or **--source-port** option to perform source port manipulation.

Source port manipulation refers to manipulating actual port numbers with common port numbers to evade IDS/firewall: this is useful when the firewall is configured to allow packets from well-known ports like HTTP, DNS, FTP, etc.

12. The results appear, displaying all open TCP ports along with the name of services running on the ports, as shown in the screenshot.

Capturing from Ethernet

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
7357	201.745643	10.10.1.13	10.10.1.11	TCP	58	80 → 1086 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
7358	201.745655	10.10.1.13	10.10.1.11	TCP	58	80 → 691 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
7359	201.745659	10.10.1.13	10.10.1.11	TCP	58	80 → 6346 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
7360	201.745659	10.10.1.13	10.10.1.11	TCP	58	80 → 7100 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
7361	201.745660	10.10.1.13	10.10.1.11	TCP	58	80 → 19801 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
7362	201.745673	10.10.1.13	10.10.1.11	TCP	58	80 → 407 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
7363	201.745675	10.10.1.13	10.10.1.11	TCP	58	80 → 48080 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
7364	201.745675	10.10.1.13	10.10.1.11	TCP	58	80 → 1277 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
7365	201.745688	10.10.1.13	10.10.1.11	TCP	58	80 → 445 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
7366	201.745688	10.10.1.13	10.10.1.11	TCP	58	80 → 6005 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
7367	201.745688	10.10.1.13	10.10.1.11	TCP	58	80 → 1248 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
7368	201.745699	10.10.1.13	10.10.1.11	TCP	58	80 → 12174 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
7369	201.745699	10.10.1.13	10.10.1.11	TCP	58	80 → 5961 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
7370	201.745746	10.10.1.11	10.10.1.13	TCP	58	445 → 80 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
7371	201.745770	10.10.1.13	10.10.1.11	TCP	58	80 → 1352 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
7372	201.745785	10.10.1.13	10.10.1.11	TCP	58	80 → 1720 [SYN] Seq=0 Win=1024 Len=0 MSS=1460

Frame 1: 438 bytes on wire (3504 bits), 438 bytes captured (3504 bits) on interface 0
 Ethernet II, Src: MS-MLB-PhysServer-21_5d:64:82:03 (02:15:5d:64:82:03), Dst: ff:00::fbda:f84b:4ba1:ef18
 Internet Protocol Version 6, Src: fe80::fbda:f84b:4ba1:ef18, Dst: ff00::1
 User Datagram Protocol, Src Port: 5353, Dst Port: 5353
 Multicast Domain Name System (response)

0000 33 33 00 00 00 fb 02 15 5d 64 82 03 86 dd 60 0f 33]d
 0010 81 9a 01 80 11 ff fe 80 00 00 00 00 00 00 fb da KK
 0020 f8 4b 4b a1 ef 18 ff 02 00 00 00 00 00 00 00 00
 0030 00 00 00 00 00 fb 14 e9 14 e9 01 80 9c f9 00 00
 0040 84 00 00 00 00 08 00 00 00 04 10 61 64 62 2d 75
 0050 6e 69 64 65 6e 74 69 66 69 65 64 04 5f 61 64 62 nidentif ied
 0060 04 5f 74 63 70 05 6c 6f 63 61 6c 00 00 10 80 01 _tcp:lo cal
 0070 00 00 11 94 00 01 00 09 5f 73 65 72 76 69 63 65 _se
 0080 73 07 5f 64 6e 73 2d 73 64 04 5f 75 64 70 c0 27 s:_dns-s d_
 0090 00 0c 00 01 00 00 11 94 00 02 c0 1d c0 1d 00 0c
 00a0 00 01 00 00 11 94 00 02 c0 0c c0 0c 00 21 80 01
 00b0 00 00 00 78 00 10 00 00 00 00 15 b3 07 41 6e 64 x
 00c0 72 6f 69 64 c0 27 02 31 34 01 31 02 31 30 02 31 roid:'1 4:1
 00d0 30 07 69 6e 2d 61 64 64 72 04 61 72 70 61 00 00 0-in-add r:a
 00e0 0c 80 01 00 00 00 78 00 02 c0 7e 01 38 01 31 01 x
 00f0 46 01 45 01 31 01 41 01 42 01 34 01 42 01 34 01 F·E·1·A· B·4
 0100 38 01 46 01 41 01 44 01 42 01 46 01 30 01 30 01 8·F·A·D· B·F
 0110 30 01 30 01 30 01 30 01 30 01 30 01 30 01 30 01 0·0·0·0· 0·0
 0120 30 01 30 01 30 01 38 01 45 01 46 01 69 70 36 c0 0e 0·0·0·8·5·F·F

Ethernet: <live capture in progress> | Packets: 9630 · Displayed: 9630 (100.0%) | Profile: Default

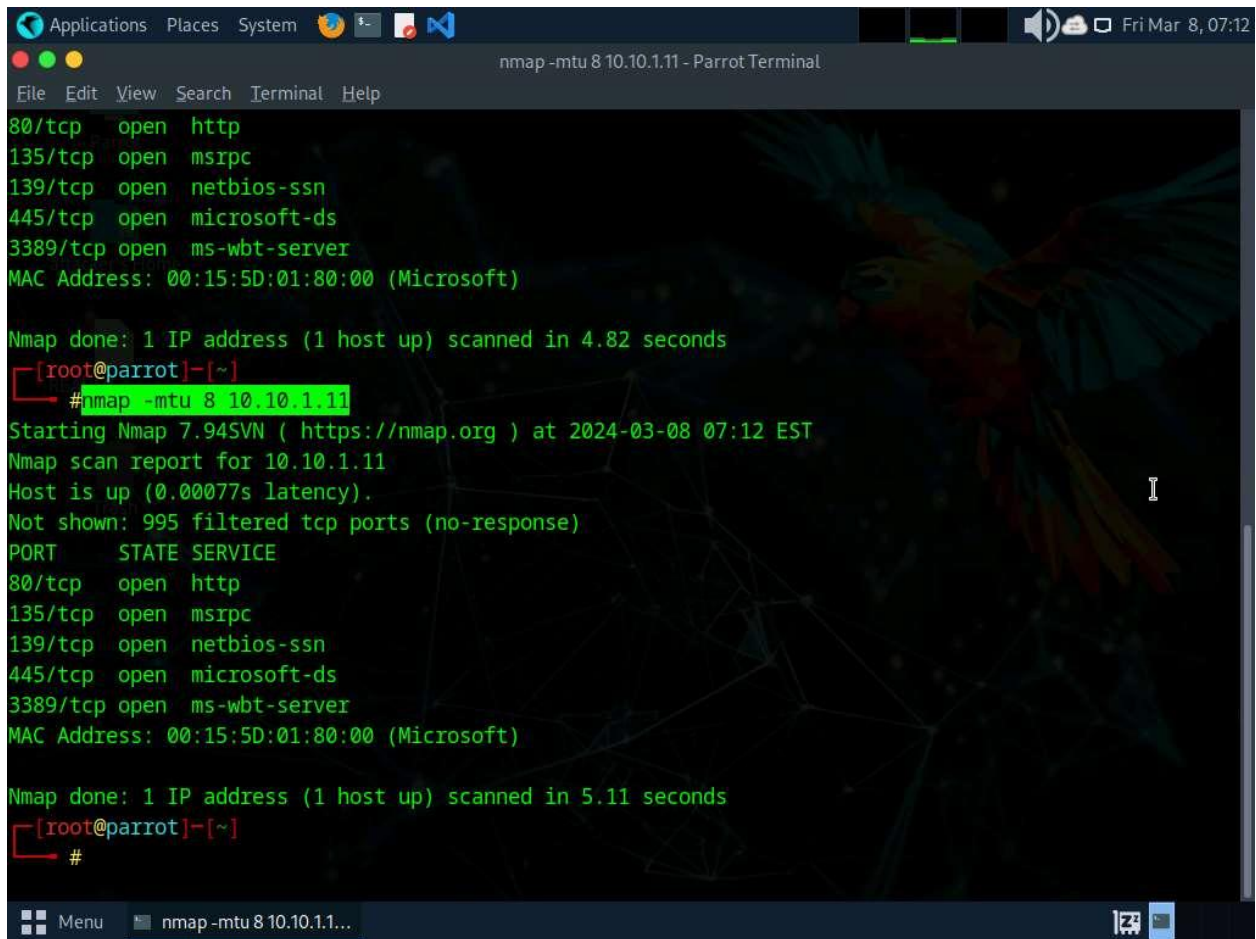
4:11 AM 3/8/2024

14. Click [Parrot Security](#) to switch to the Parrot Security machine.

15. Now, run **nmap -mtu 8 [Target IP Address]** command (here, target IP address is **10.10.1.11**).

In this command, **-mtu**: specifies the number of Maximum Transmission Unit (MTU) (here, **8** bytes of packets).

Using MTU, smaller packets are transmitted instead of sending one complete packet at a time. This technique evades the filtering and detection mechanism enabled in the target machine.

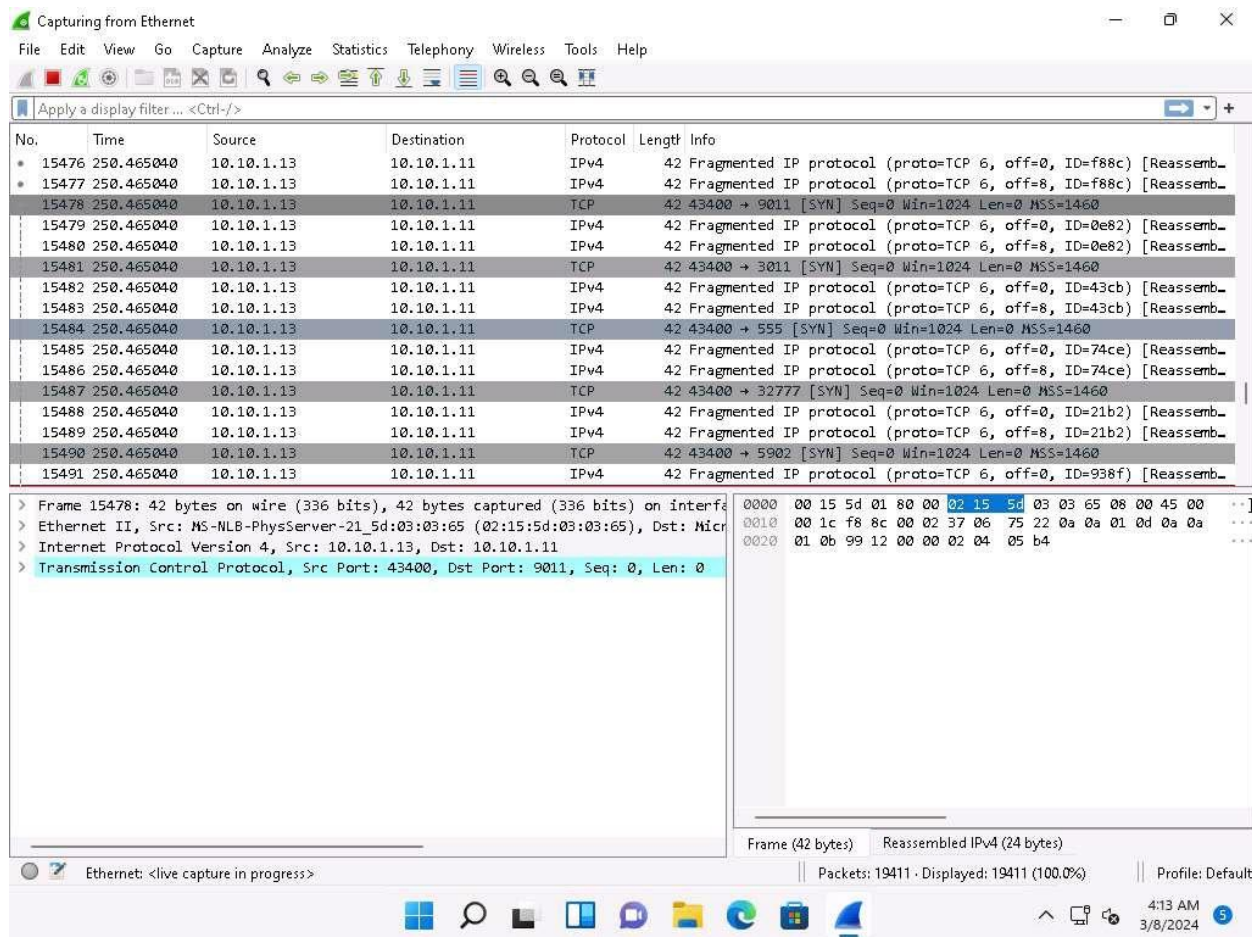


```
Applications Places System [Icons] [Terminal] [Help]
nmap -mtu 8 10.10.1.11 - Parrot Terminal
File Edit View Search Terminal Help
80/tcp open http
135/tcp open msrpc
139/tcp open netbios-ssn
445/tcp open microsoft-ds
3389/tcp open ms-wbt-server
MAC Address: 00:15:5D:01:80:00 (Microsoft)

Nmap done: 1 IP address (1 host up) scanned in 4.82 seconds
[root@parrot]~# nmap -mtu 8 10.10.1.11
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-08 07:12 EST
Nmap scan report for 10.10.1.11
Host is up (0.00077s latency).
Not shown: 995 filtered tcp ports (no-response)
PORT      STATE SERVICE
80/tcp    open  http
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
3389/tcp    open  ms-wbt-server
MAC Address: 00:15:5D:01:80:00 (Microsoft)

Nmap done: 1 IP address (1 host up) scanned in 5.11 seconds
[root@parrot]~#
```

16. Click [Windows 11](#) to switch to the **Windows 11** machine (target machine). In the **Wireshark** window, scroll-down and you can observe the fragmented packets having maximum length as 8 bytes, as shown in the screenshot.



17. Click [Parrot Security](#) to switch to the **Parrot Security** machine.

18. Now, run **nmap -D RND:10 [Target IP Address]** command (here, target IP address is **10.10.1.11**).

In this command, **-D**: performs a decoy scan and **RND**: generates a random and non-reserved IP addresses (here, **10**).

The IP address decoy technique refers to generating or manually specifying IP addresses of the decoys to evade IDS/firewall. This technique makes it difficult for the IDS/firewall to determine which IP address was actually scanning the network and which IP addresses were decoys. By using this command, Nmap automatically generates a random number of decoys for the scan and randomly positions the real IP address between the decoy IP addresses.

[more...](#)

```
Applications  Places  System  nmap -D RND:10 10.10.1.11 - Parrot Terminal
File Edit View Search Terminal Help

[root@parrot]~#
#nmap -D RND:10 10.10.1.11
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-08 07:13 EST
Nmap scan report for 10.10.1.11
Host is up (0.00067s latency).
Not shown: 995 filtered tcp ports (no-response)
PORT      STATE SERVICE
80/tcp    open  http
135/tcp   open  msrcp
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3389/tcp  open  ms-wbt-server
MAC Address: 00:15:5D:01:80:00 (Microsoft)

Nmap done: 1 IP address (1 host up) scanned in 4.93 seconds
[root@parrot]~#
#
```

19. Now, click [Windows 11](#) to switch to the **Windows 11** machine (target machine). In the **Wireshark** window, scroll-down and you can observe the packets displaying the multiple IP addresses in the source section, as shown in the screenshot.

Capturing from Ethernet

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
29763	471.392823	79.151.148.5	10.10.1.11	TCP	58	47813 → 1154 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
29764	471.392834	141.148.68.58	10.10.1.11	TCP	58	47813 → 8701 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
29765	471.392834	106.164.1.88	10.10.1.11	TCP	58	47813 → 5225 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
29766	471.392840	98.250.85.21	10.10.1.11	TCP	58	47813 → 1154 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
29767	471.392840	10.10.1.13	10.10.1.11	TCP	58	47813 → 1154 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
29768	471.392854	32.100.83.111	10.10.1.11	TCP	58	47813 → 5225 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
29769	471.392854	79.151.148.5	10.10.1.11	TCP	58	47813 → 5225 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
29770	471.392859	206.223.77.12	10.10.1.11	TCP	58	47813 → 8701 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
29771	471.392868	55.148.118.186	10.10.1.11	TCP	58	47813 → 8701 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
29772	471.392868	110.166.90.67	10.10.1.11	TCP	58	47813 → 5225 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
29773	471.392870	10.10.1.13	10.10.1.11	TCP	58	47813 → 8701 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
29774	471.392870	98.250.85.21	10.10.1.11	TCP	58	47813 → 5225 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
29775	471.392872	115.126.193.182	10.10.1.11	TCP	58	47813 → 8701 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
29776	471.392888	55.148.118.186	10.10.1.11	TCP	58	47813 → 5225 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
29777	471.392889	10.10.1.13	10.10.1.11	TCP	58	47813 → 5225 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
29778	471.392888	141.148.68.58	10.10.1.11	TCP	58	47813 → 5225 [SYN] Seq=0 Win=1024 Len=0 MSS=1460

Frame 1: 438 bytes on wire (3504 bits), 438 bytes captured (3504 bits) on interface 0
 Ethernet II, Src: MS-NLB-PhysServer-21_5d:64:82:03 (02:15:5d:64:82:03), Dst: ff:ff:ff:ff:ff:ff
 Internet Protocol Version 6, Src: fe80::fbda:f84b:4ba1:ef18, Dst: ff02::1:3::3
 User Datagram Protocol, Src Port: 5353, Dst Port: 5353
 Multicast Domain Name System (response)

0000 33 33 00 00 00 fb 02 15 5d 64 82 03 86 dd 60 0f 33.....]d
 0010 81 9a 01 80 11 ff fe 80 00 00 00 00 00 00 fb da
 0020 f8 4b 4b a1 ef 18 ff 02 00 00 00 00 00 00 00 00KK.....
 0030 00 00 00 00 00 fb 14 e9 14 e9 01 80 9c f9 00 00
 0040 84 00 00 00 00 08 00 00 00 04 10 61 64 62 2d 75
 0050 6e 69 64 65 6e 74 69 66 69 65 64 04 5f 61 64 62 nidentif ied
 0060 04 5f 74 63 70 05 6c 6f 63 61 6c 00 00 10 80 01 _tcp:lo cal
 0070 00 00 11 94 00 01 00 09 5f 73 65 72 76 69 63 65_se
 0080 73 07 5f 64 6e 73 2d 73 64 04 5f 75 64 70 c0 27 s:_dns-s d_
 0090 00 0c 00 01 00 00 11 94 00 02 c0 1d c0 1d 00 0c
 00a0 00 01 00 00 11 94 00 02 c0 0c c0 0c 00 21 80 01
 00b0 00 00 00 78 00 10 00 00 00 00 15 b3 07 41 6e 64 ..x.....
 00c0 72 6f 69 64 c0 27 02 31 34 01 31 02 31 30 02 31 roid:'1 4:1
 00d0 30 07 69 6e 2d 61 64 64 72 04 61 72 70 61 00 00 0-in-add r-a
 00e0 0c 80 01 00 00 00 78 00 02 c0 7e 01 38 01 31 01x.....
 00f0 46 01 45 01 31 01 41 01 42 01 34 01 42 01 34 01 F-E-1-A-B-4
 0100 38 01 46 01 41 01 44 01 42 01 46 01 30 01 30 01 8-F-A-D-B-F
 0110 30 01 30 01 30 01 30 01 30 01 30 01 30 01 30 01 0-0-0-0-0-0
 0120 30 01 30 01 30 01 38 01 45 01 46 01 69 70 36 c0 0-0-0-8-F-F

Ethernet: <live capture in progress> | Packets: 38253 · Displayed: 38253 (100.0%) | Profile: Default

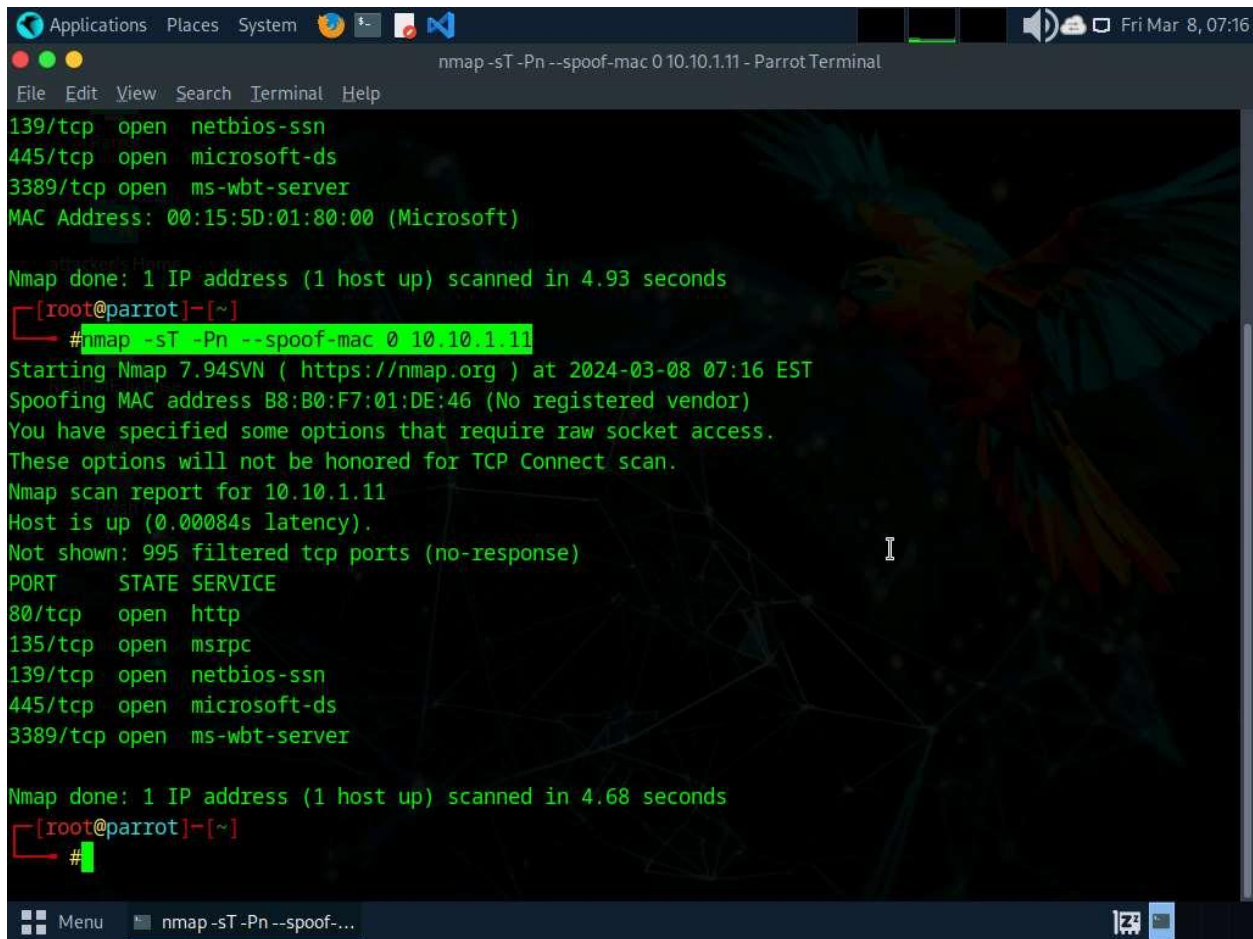
4:15 AM 3/8/2024

20. Click [Parrot Security](#) to switch to the Parrot Security machine.

21. In the terminal window, run **nmap -sT -Pn --spoof-mac 0 [Target IP Address]** command (here, target IP address is **10.10.1.11**).

In this command **--spoof-mac 0** represents randomizing the MAC address, **-sT**: performs the TCP connect/full open scan, **-Pn** is used to skip the host discovery.

MAC address spoofing technique involves spoofing a MAC address with the MAC address of a legitimate user on the network. This technique allows you to send request packets to the targeted machine/network pretending to be a legitimate host.



```
Applications Places System nmap -sT -Pn --spoof-mac 0 10.10.1.11 - Parrot Terminal
File Edit View Search Terminal Help
139/tcp open netbios-ssn
445/tcp open microsoft-ds
3389/tcp open ms-wbt-server
MAC Address: 00:15:5D:01:80:00 (Microsoft)

Nmap done: 1 IP address (1 host up) scanned in 4.93 seconds
[root@parrot]~# nmap -sT -Pn --spoof-mac 0 10.10.1.11
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-08 07:16 EST
Spoofing MAC address B8:B0:F7:01:DE:46 (No registered vendor)
You have specified some options that require raw socket access.
These options will not be honored for TCP Connect scan.
Nmap scan report for 10.10.1.11
Host is up (0.00084s latency).
Not shown: 995 filtered tcp ports (no-response)
PORT      STATE SERVICE
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3389/tcp  open  ms-wbt-server

Nmap done: 1 IP address (1 host up) scanned in 4.68 seconds
[root@parrot]~#
```

22. Click [Windows 11](#) to switch to the **Windows 11** machine (target machine). In the **Wireshark** window, scroll-down and you can observe the captured TCP, as shown in the screenshot.

Capturing from Ethernet

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1079	107.747115	10.10.1.13	10.10.1.11	IPv4	42	Fragmented IP protocol (proto=TCP 6, off=0, ID=ce22) [Reassemb...
1080	107.747115	10.10.1.13	10.10.1.11	IPv4	42	Fragmented IP protocol (proto=TCP 6, off=8, ID=ce22) [Reassemb...
1081	107.747115	10.10.1.13	10.10.1.11	TCP	42	35666 → 1433 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
1082	107.747115	10.10.1.13	10.10.1.11	IPv4	42	Fragmented IP protocol (proto=TCP 6, off=0, ID=c12a) [Reassemb...
1083	107.747115	10.10.1.13	10.10.1.11	IPv4	42	Fragmented IP protocol (proto=TCP 6, off=8, ID=c12a) [Reassemb...
1084	107.747115	10.10.1.13	10.10.1.11	TCP	42	35666 → 6566 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
1085	107.747275	10.10.1.11	10.10.1.13	TCP	58	3389 → 35669 [SYN, ACK] Seq=0 Ack=1 Win=64000 Len=0 MSS=1460
1086	107.747536	10.10.1.13	10.10.1.11	TCP	54	35669 → 3389 [RST] Seq=1 Win=0 Len=0
1087	107.749672	10.10.1.13	10.10.1.11	IPv4	42	Fragmented IP protocol (proto=TCP 6, off=0, ID=cf54) [Reassemb...
1088	107.749672	10.10.1.13	10.10.1.11	IPv4	42	Fragmented IP protocol (proto=TCP 6, off=8, ID=cf54) [Reassemb...
1089	107.749672	10.10.1.13	10.10.1.11	TCP	42	35666 → 4550 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
1090	107.749672	10.10.1.13	10.10.1.11	IPv4	42	Fragmented IP protocol (proto=TCP 6, off=0, ID=b878) [Reassemb...
1091	107.749672	10.10.1.13	10.10.1.11	IPv4	42	Fragmented IP protocol (proto=TCP 6, off=8, ID=b878) [Reassemb...
1092	107.749672	10.10.1.13	10.10.1.11	TCP	42	35666 → 5000 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
1093	107.749672	10.10.1.13	10.10.1.11	IPv4	42	Fragmented IP protocol (proto=TCP 6, off=0, ID=ae32) [Reassemb...
1094	107.749672	10.10.1.13	10.10.1.11	IPv4	42	Fragmented IP protocol (proto=TCP 6, off=8, ID=ae32) [Reassemb...

> Frame 1: 438 bytes on wire (3504 bits), 438 bytes captured (3504 bits)
 > Ethernet II, Src: MS-NLB-PhysServer-21_5d:64:82:03 (02:15:5d:64:82:03)
 > Internet Protocol Version 6, Src: fe80::fbda:f84b:4ba1:ef18, Dst: ff02::c
 > User Datagram Protocol, Src Port: 5353, Dst Port: 5353
 > Multicast Domain Name System (response)

0000 33 33 00 00 00 fb 02 15 5d 64 82 03 86 dd 60 0f 33.....]d
 0010 81 9a 01 80 11 ff fe 80 00 00 00 00 00 00 fb da
 0020 f8 4b 4b a1 ef 18 ff 02 00 00 00 00 00 00 00 00 ...KK.....
 0030 00 00 00 00 00 fb 14 e9 14 e9 01 80 9c f9 00 00
 0040 84 00 00 00 00 08 00 00 00 04 10 61 64 62 2d 75
 0050 6e 69 64 65 6e 74 69 66 69 65 64 04 5f 61 64 62 nidentif ied
 0060 04 5f 74 63 70 05 6c 6f 63 61 6c 00 00 10 80 01 _tcp_lo cal
 0070 00 00 11 94 00 01 00 09 5f 73 65 72 76 69 63 65_se
 0080 73 07 5f 64 6e 73 2d 73 64 04 5f 75 64 70 c0 27 s_dns-s d_...
 0090 00 0c 00 01 00 00 11 94 00 02 c0 1d c0 1d 00 0c
 00a0 00 01 00 00 11 94 00 02 c0 0c c0 0c 00 21 80 01
 00b0 00 00 00 78 00 10 00 00 00 00 15 b3 07 41 6e 64 ...x.....
 00c0 72 6f 69 64 c0 27 02 31 34 01 31 02 31 30 02 31 roid-1 4-1
 00d0 30 07 69 6e 2d 61 64 64 72 04 61 72 70 61 00 00 0-in-add r-a
 00e0 0c 80 01 00 00 00 78 00 02 c0 7e 01 38 01 31 01x...
 00f0 46 01 45 01 31 01 41 01 42 01 34 01 42 01 34 01 F-E-1-A- B-4
 0100 38 01 46 01 41 01 44 01 42 01 46 01 30 01 30 01 8-F-A-D- B-F
 0110 30 01 30 01 30 01 30 01 30 01 30 01 30 01 30 01 0-0-0-0-0-0
 0120 30 01 30 01 30 01 38 01 45 01 46 01 69 70 36 c0 0-0-0-0-0-0

Ethernet: <live capture in progress> | Packets: 40620 · Displayed: 40620 (100.0%) | Profile: Default

4:17 AM 3/8/2024

23. This concludes the demonstration of evading IDS and firewall using various evasion techniques in Nmap.

24. Close all open windows and document all the acquired information.

Question 3.4.1.1

Use the Nmap tool to scan beyond the IDS/firewall of the target machine (Windows 11). Enter the Nmap option that is used to split the IP packet into tiny fragment packets. Note: Turn on Windows Firewall to perform this task.