

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
- Create custom packets using Colasoft Packet Builder to scan beyond the IDS/firewall
- Create custom UDP and TCP packets using Hping3 to scan beyond the IDS/firewall
- Create custom packets using Nmap to scan beyond the IDS/firewall

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

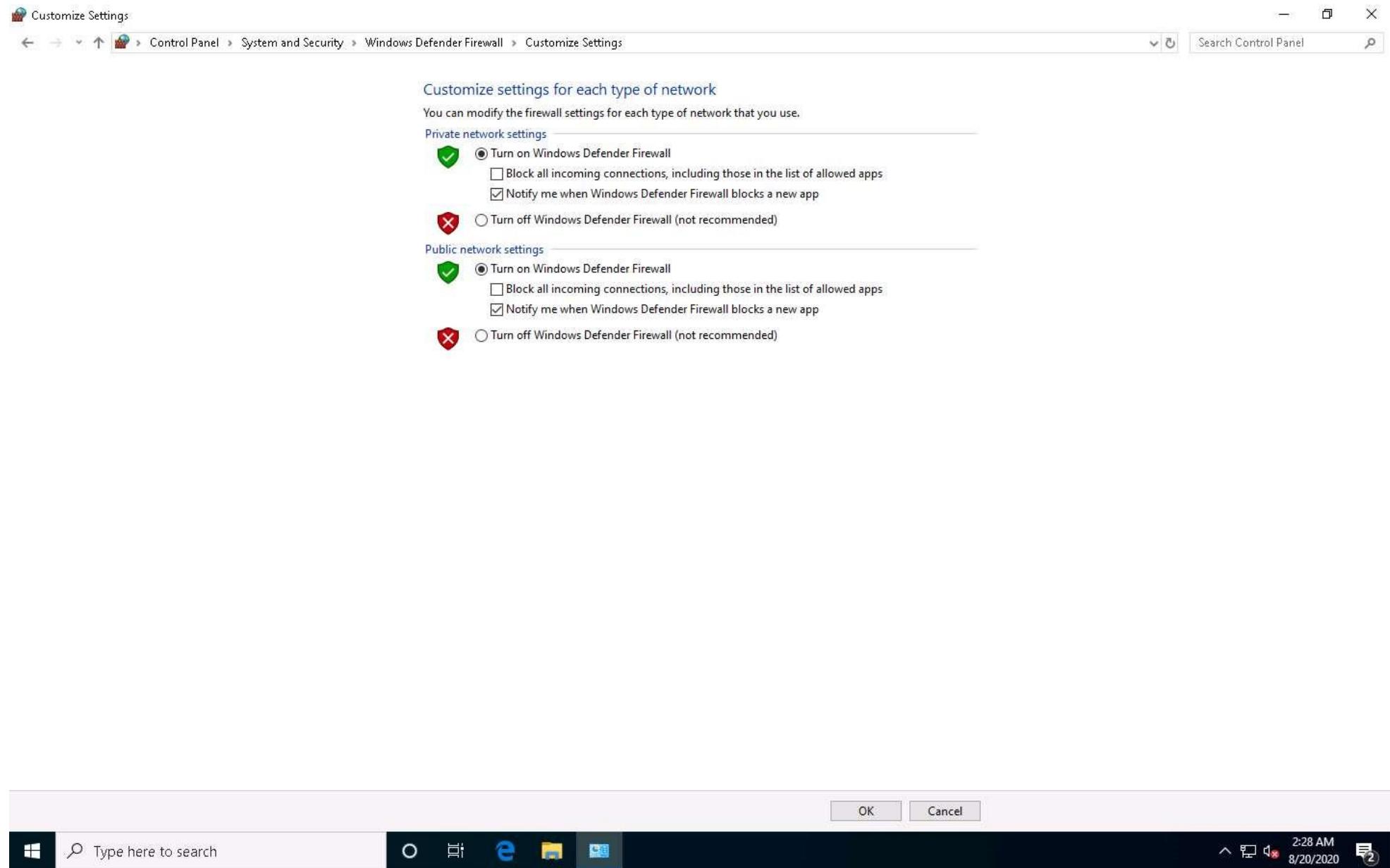
- **Anonymizers:** Use anonymizers that allow them to bypass Internet censors and evade certain IDS and firewall rules

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

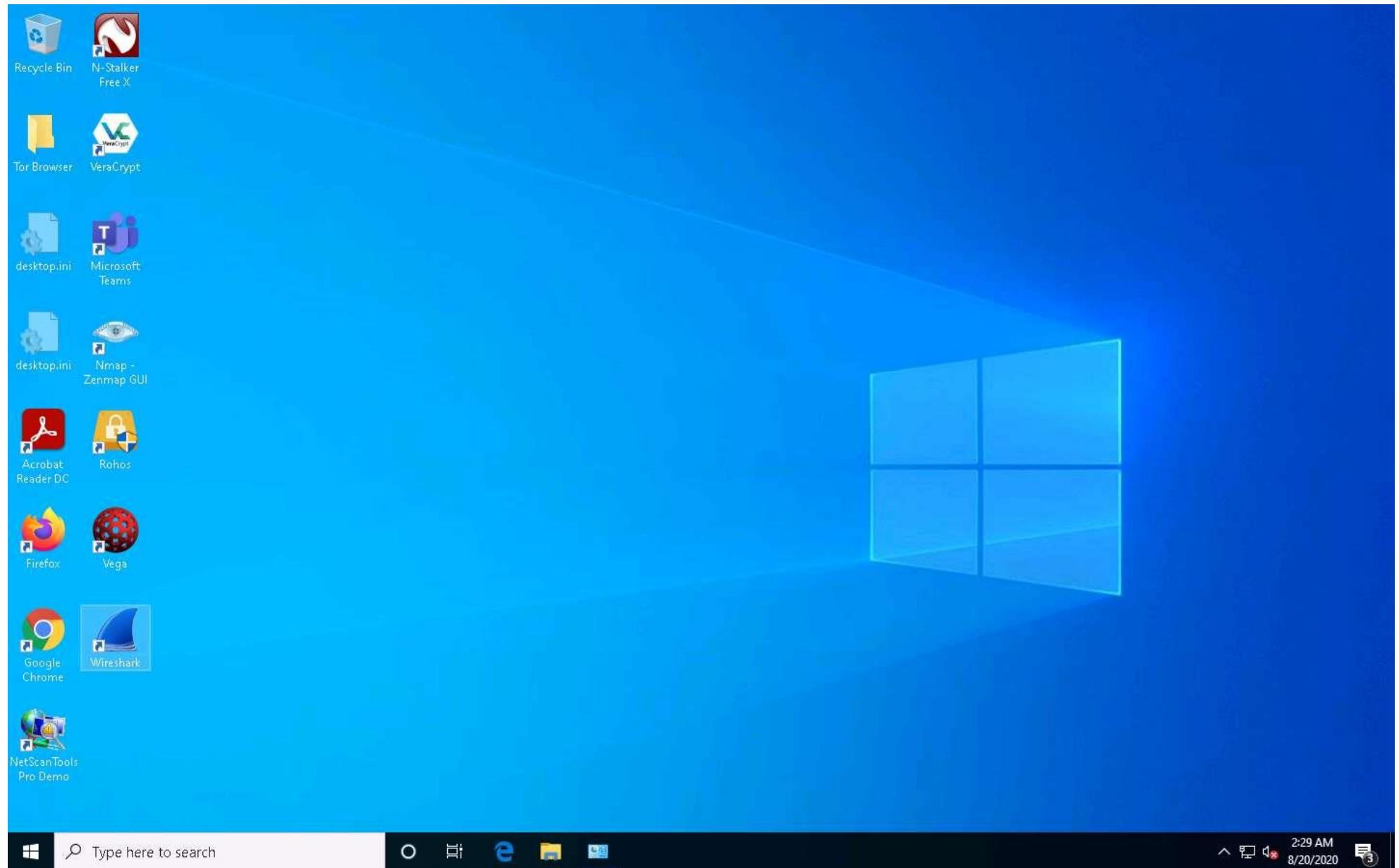
Nmap offers many features to help understand complex networks with enabled security mechanisms and supports mechanisms for bypassing poorly implemented defenses. Using Nmap, various techniques can be implemented, which can bypass the IDS/firewall security mechanisms.

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 10](#) to switch to the **Windows 10** 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.

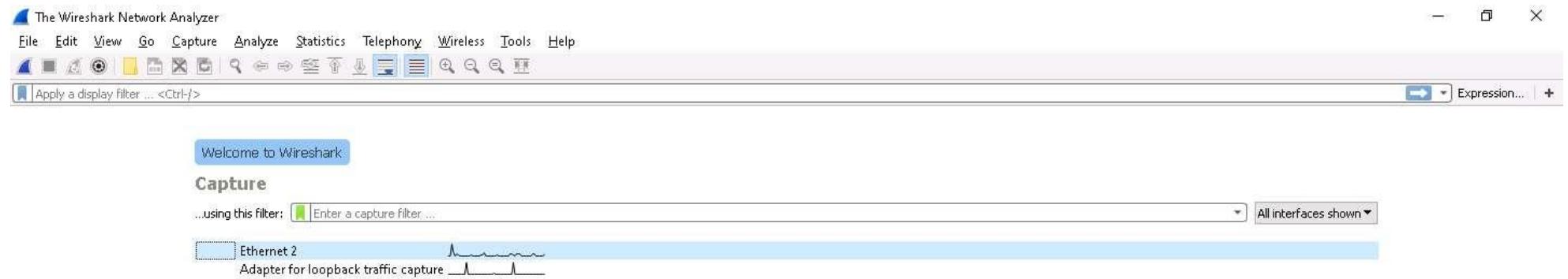


3. Minimize the **Control Panel** window, navigate to the **Desktop** and double-click **Wireshark** shortcut.



4. The **Wireshark Network Analyzer** window appears. Start capturing packets by double-clicking the available ethernet or interface (here, **Ethernet2**).

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



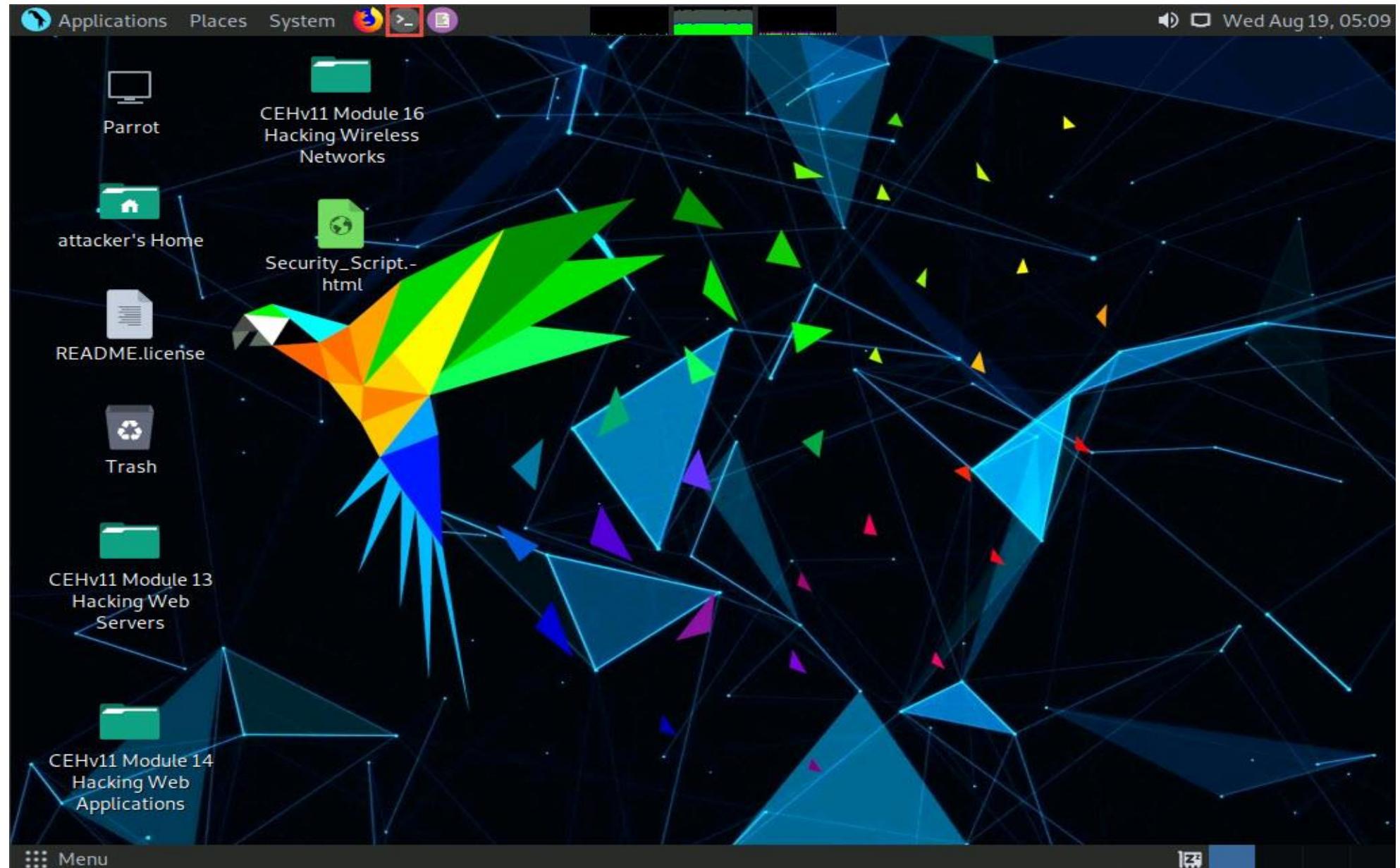
Learn

[User's Guide](#) · [Wiki](#) · [Questions and Answers](#) · [Mailing Lists](#)

You are running Wireshark 3.0.5 (v3.0.5-0-g752a55954770). You receive automatic updates.



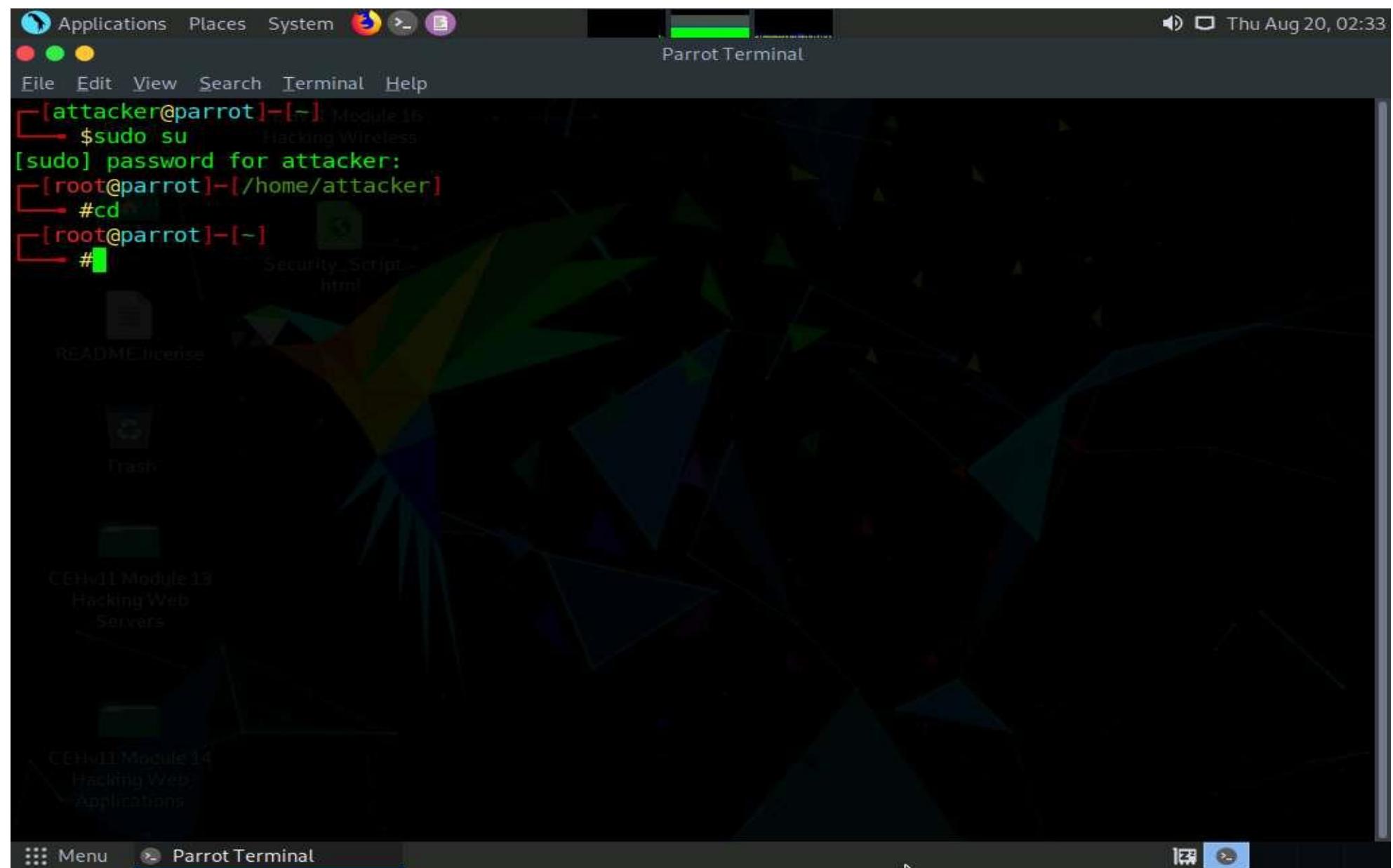
5. Click **Parrot Security** to switch to the **Parrot Security** machine.
6. Click the **MATE Terminal** icon in the top-left corner of the **Desktop** window to open a **Terminal** window.



7. A **Parrot Terminal** window appears. In the terminal window, type **sudo su** and press **Enter** to run the programs as a root user.
8. In the **[sudo] password for attacker** field, type **toor** as a password and press **Enter**.

The password that you type will not be visible.

9. Now, type **cd** and press **Enter** to jump to the root directory.



10. A **Parrot Terminal** window appears. In the terminal window, type **nmap -f [Target IP Address]**, (here, the target machine is **Windows 10 [10.10.10.10]**) and press **Enter**.

-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...](#)

11. Although **Windows Defender Firewall** is turned on in the target system (here, **Windows 10**), 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.

The screenshot shows a Parrot OS desktop environment. At the top, there's a dark header bar with icons for Applications, Places, System, and a volume slider. The date and time 'Thu Aug 20, 02:34' are also displayed. Below the header is a window titled 'Parrot Terminal'. The terminal window has a dark background with green text. It shows a command-line session:

```
[attacker@parrot] -[~] Module16
└─$ sudo su
[sudo] password for attacker:
[root@parrot] -[/home/attacker]
└─# cd
[root@parrot] -[~]
└─# nmap -f 10.10.10.10
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-20 02:33 EDT
Nmap scan report for 10.10.10.10
Host is up (0.0011s latency).
Not shown: 995 filtered ports
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:27:08:B2 (Microsoft)

Nmap done: 1 IP address (1 host up) scanned in 4.89 seconds
[root@parrot] -[~]
└─#
```

The terminal window is titled 'Parrot Terminal'. The desktop background features a dark, geometric pattern. In the bottom left corner, there's a small sidebar with the text 'CEHv11Module14' and 'Hacking Web Applications'.

12. In the **Parrot Terminal** window, type **nmap -g 80 [Target IP Address]**, (here, target IP address is **10.10.10.10**) and press **Enter**.

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.

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

The screenshot shows a terminal window titled "Parrot Terminal" running on a Parrot OS desktop environment. The terminal displays the output of an Nmap scan. The output shows the following details:

```
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-20 02:33 EDT
Nmap scan report for 10.10.10.10
Host is up (0.0011s latency).
Not shown: 995 filtered ports
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:27:08:B2 (Microsoft)

Nmap done: 1 IP address (1 host up) scanned in 4.89 seconds
```

Below the first scan, the terminal prompt shows the command used:

```
[root@parrot]# nmap -g 80 10.10.10.10
```

The terminal then performs a second scan with the same parameters:

```
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-20 02:34 EDT
Nmap scan report for 10.10.10.10
Host is up (0.00072s latency).
Not shown: 995 filtered ports
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:27:08:B2 (Microsoft)

Nmap done: 1 IP address (1 host up) scanned in 5.01 seconds
```

The terminal prompt shows the command used for the second scan:

```
[root@parrot]#
```

14. Now, type **nmap -mtu 8 [Target IP Address]** (here, target IP address is **10.10.10.10**) and press **Enter**.

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.

The screenshot shows a terminal window titled "Parrot Terminal" running on a Parrot OS desktop environment. The terminal displays two Nmap scans and a command to test MTU. The first scan (without MTU) shows standard ports 80, 135, 139, 445, and 3389 open. The second scan (with -mtu 8) also shows the same ports open. Finally, the command "# nmap -mtu 8 10.10.10.10" is run, which outputs the same results as the previous scan, demonstrating that MTU does not affect the scan results in this case.

```
Applications Places System Parrot Terminal
Thu Aug 20, 02:37
File Edit View Search Terminal Help
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-20 02:34 EDT
Nmap scan report for 10.10.10.10
Host is up (0.00072s latency).
Not shown: 995 filtered ports
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:27:08:B2 (Microsoft)

Nmap done: 1 IP address (1 host up) scanned in 5.01 seconds
[root@parrot]# ~
[root@parrot]# nmap -mtu 8 10.10.10.10
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-20 02:36 EDT
Nmap scan report for 10.10.10.10
Host is up (0.0010s latency).
Not shown: 995 filtered ports
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:27:08:B2 (Microsoft)

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

15. Now, type **nmap -D RND:10 [Target IP Address]** (here, target IP address is **10.10.10.10**) and press **Enter**.

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

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...](#)

The screenshot shows a terminal window titled "Parrot Terminal" running on Parrot OS. The terminal displays the output of an Nmap scan for the host 10.10.10.10. The scan results show several open ports: 80/tcp (http), 135/tcp (msrpc), 139/tcp (netbios-ssn), 445/tcp (microsoft-ds), and 3389/tcp (ms-wbt-server). The MAC address of the host is listed as 00:15:5D:27:08:B2 (Microsoft). The scan completed in 5.12 seconds.

```
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-20 02:36 EDT
Nmap scan report for 10.10.10.10
Host is up (0.0010s latency).
Not shown: 995 filtered ports
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:27:08:B2 (Microsoft)

Nmap done: 1 IP address (1 host up) scanned in 5.12 seconds
[root@parrot]# nmap -D RND:10 10.10.10.10
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-20 02:37 EDT
Nmap scan report for 10.10.10.10
Host is up (0.00099s latency).
Not shown: 995 filtered ports
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:27:08:B2 (Microsoft)

Nmap done: 1 IP address (1 host up) scanned in 5.15 seconds
[root@parrot]#
```

16. Now, click Windows 10 to switch to the **Windows 10** machine (target machine) and observe packets captured by Wireshark, which displays the multiple IP addresses in the source section, as shown in the screenshot.

Capturing from Ethernet 2

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

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

No.	Time	Source	Destination	Protocol	Length	Info
36622	398.335981	10.10.10.13	10.10.10.10	TCP	58	54689 → 981 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
36623	398.336009	17.71.80.192	10.10.10.10	TCP	58	54689 → 981 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
36624	398.336026	93.203.5.227	10.10.10.10	TCP	58	54689 → 981 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
36625	398.336026	2.33.254.7	10.10.10.10	TCP	58	54689 → 981 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
36626	398.336027	25.51.206.155	10.10.10.10	TCP	58	54689 → 981 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
36627	398.336028	128.107.167.252	10.10.10.10	TCP	58	54689 → 981 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
36628	398.336028	133.5.157.213	10.10.10.10	TCP	58	54689 → 981 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
36629	398.336029	189.153.182.185	10.10.10.10	TCP	58	54689 → 981 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
36630	398.336080	135.9.127.235	10.10.10.10	TCP	58	54689 → 1972 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
36631	398.336081	10.10.10.13	10.10.10.10	TCP	58	54689 → 1972 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
36632	398.336082	93.203.5.227	10.10.10.10	TCP	58	54689 → 1972 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
36633	398.336082	2.33.254.7	10.10.10.10	TCP	58	54689 → 1972 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
36634	398.336117	17.71.80.192	10.10.10.10	TCP	58	54689 → 1972 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
36635	398.336131	128.107.167.252	10.10.10.10	TCP	58	54689 → 1972 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
36636	398.336144	159.90.207.171	10.10.10.10	TCP	58	54689 → 1972 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
36637	398.336330	159.90.207.171	10.10.10.10	TCP	58	54689 → 981 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
36638	398.336331	155.226.162.165	10.10.10.10	TCP	58	54689 → 1972 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
36639	398.336332	25.51.206.155	10.10.10.10	TCP	58	54689 → 1972 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
36640	398.336332	133.5.157.213	10.10.10.10	TCP	58	54689 → 1972 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
36641	398.336333	189.153.182.185	10.10.10.10	TCP	58	54689 → 1972 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
36642	398.336333	10.10.10.13	10.10.10.10	TCP	58	54689 → 1040 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
36643	398.336334	93.203.5.227	10.10.10.10	TCP	58	54689 → 1040 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
36644	398.336335	17.71.80.192	10.10.10.10	TCP	58	54689 → 1040 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
36645	398.336335	128.107.167.252	10.10.10.10	TCP	58	54689 → 1040 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
36646	398.336336	159.90.207.171	10.10.10.10	TCP	58	54689 → 1040 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
36647	398.336337	189.153.182.185	10.10.10.10	TCP	58	54689 → 1040 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
36648	398.336427	155.226.162.165	10.10.10.10	TCP	58	54689 → 1040 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
36649	398.336427	135.9.127.235	10.10.10.10	TCP	58	54689 → 1040 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
36650	398.336428	2.33.254.7	10.10.10.10	TCP	58	54689 → 1040 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
36651	398.336429	25.51.206.155	10.10.10.10	TCP	58	54689 → 1040 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
36652	398.336429	133.5.157.213	10.10.10.10	TCP	58	54689 → 1040 [SYN] Seq=0 Win=1024 Len=0 MSS=1460

> Frame 1: 110 bytes on wire (880 bits), 110 bytes captured (880 bits) on interface 0
 > Ethernet II, Src: Microsoft_{27:08:b0} (00:15:5d:27:08:b0), Dst: IPv6mcast_01 (33:33:00:00:00:01)
 > Internet Protocol Version 6, Src: fe80::1:1, Dst: ff02::1
 > Internet Control Message Protocol v6

```
0000  33 33 00 00 00 01 00 15 5d 27 08 b0 86 dd 60 00 33.....].....  

0010  00 00 00 38 3a ff fe 80 00 00 00 00 00 00 00 00 ..8:.....  

0020  00 00 00 01 00 01 ff 02 00 00 00 00 00 00 00 .....  

0030  00 00 00 00 00 01 86 00 93 55 40 40 01 ee 00 .....U@0.....  

0040  00 00 00 00 00 00 01 f3 00 00 00 00 00 0a 0b 6c .....1.....  

0050  6f 63 61 6c 64 6f 6d 61 69 6e 00 00 00 00 05 01 ocaldoma in .....  

0060  00 00 00 00 05 dc 01 01 00 15 5d 27 08 b0 .....]....
```

Ethernet 2: <live capture in progress>

Type here to search

Packets: 37384 · Displayed: 37384 (100.0%)

Profile: Default

2:38 AM 8/20/2020

- This concludes the demonstration of evading IDS and firewall using various evasion techniques in Nmap.
- Close all open windows and document all the acquired information.

Task 2: Create Custom Packets using Colasoft Packet Builder to Scan beyond IDS/Firewall

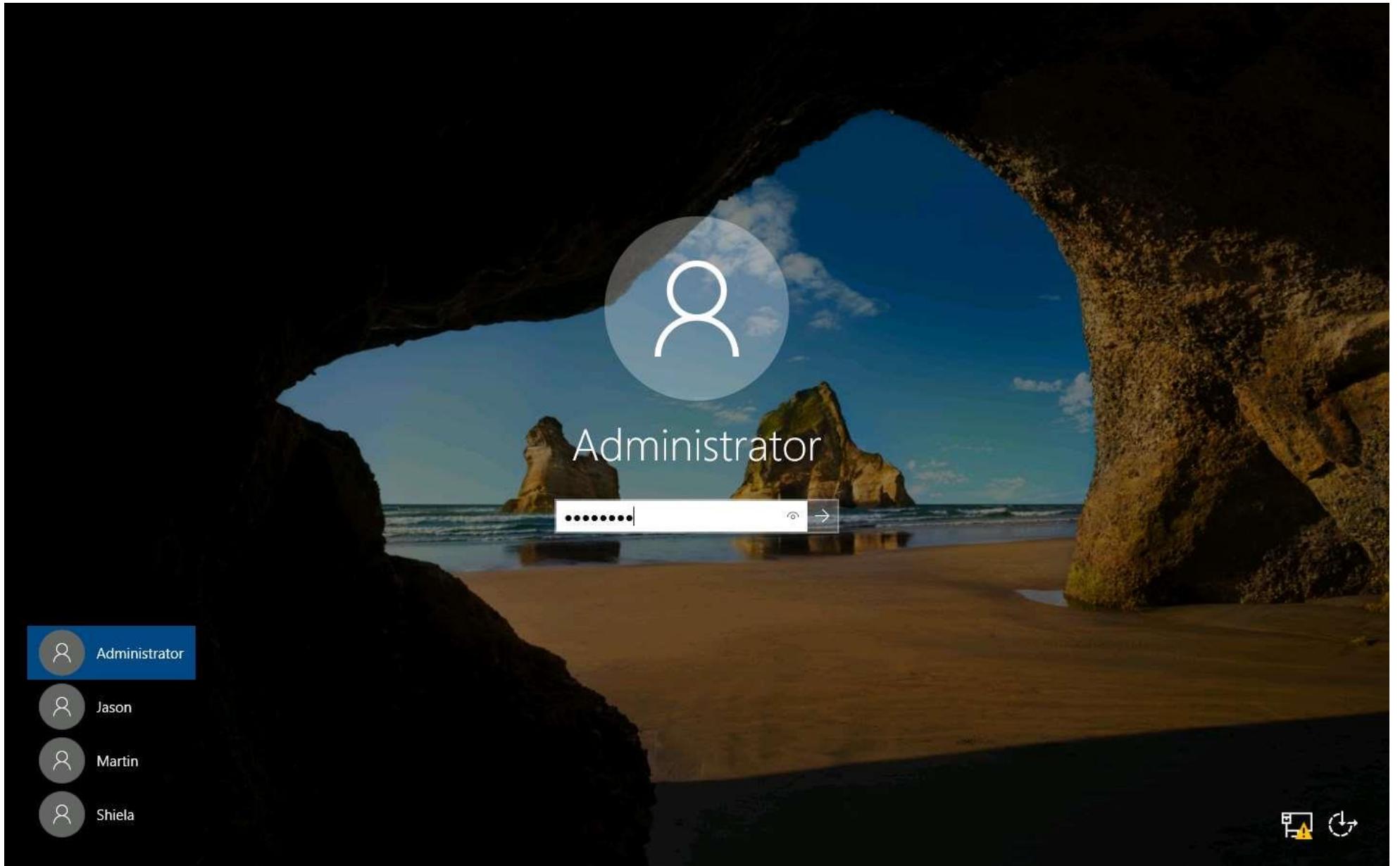
Colasoft Packet Builder is a tool that allows you to create custom network packets to assess network security. You can also select a TCP packet from the provided templates and change the parameters in the decoder editor, hexadecimal editor, or ASCII editor to create a packet. In addition to building packets, the Colasoft Packet Builder supports saving packets to packet files and sending packets to the network.

Here, we will use the Colasoft Packet Builder tool to create custom TCP packets to scan the target host by bypassing the IDS/firewall.

1. Click [Windows Server 2019](#) to switch to the **Windows Server 2019** machine.
2. Click [`Ctrl+Alt+Delete`](#) to activate the machine. By default, **Administration** user profile is selected, click [`Pa\$\$w0rd`](#) to paste the password in the Password field and press **Enter** to login.

Alternatively, you can also click **Pa\$\$w0rd** under **Windows Server 2019** machine thumbnail in the **Resources** pane or Click **Type Text | Type Password** button under Commands (**thunder** icon) menu.

Networks screen appears, click **Yes** to allow your PC to be discoverable by other PCs and devices on the network.

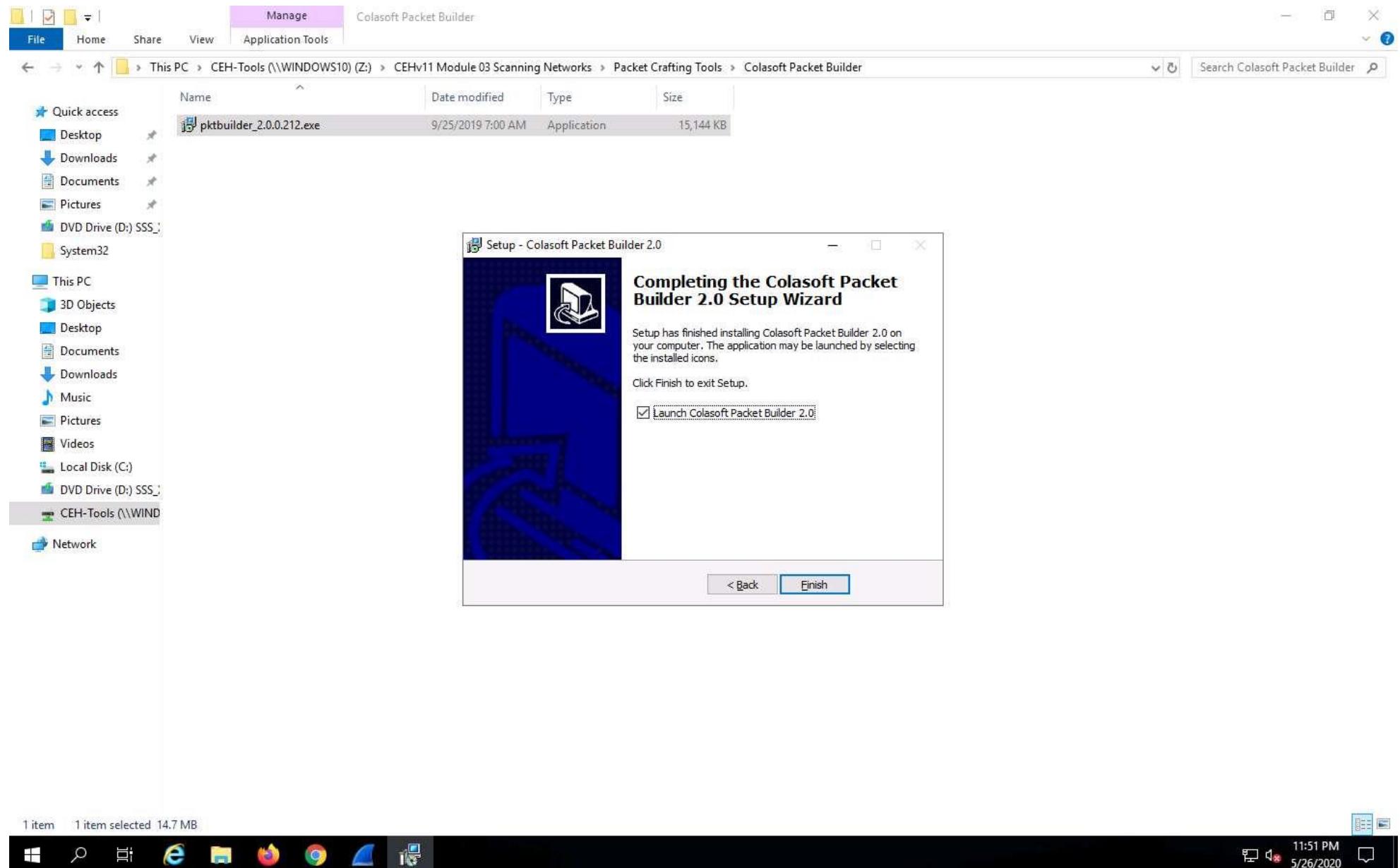


3. In the **Desktop**, double-click **Wireshark** shortcut.



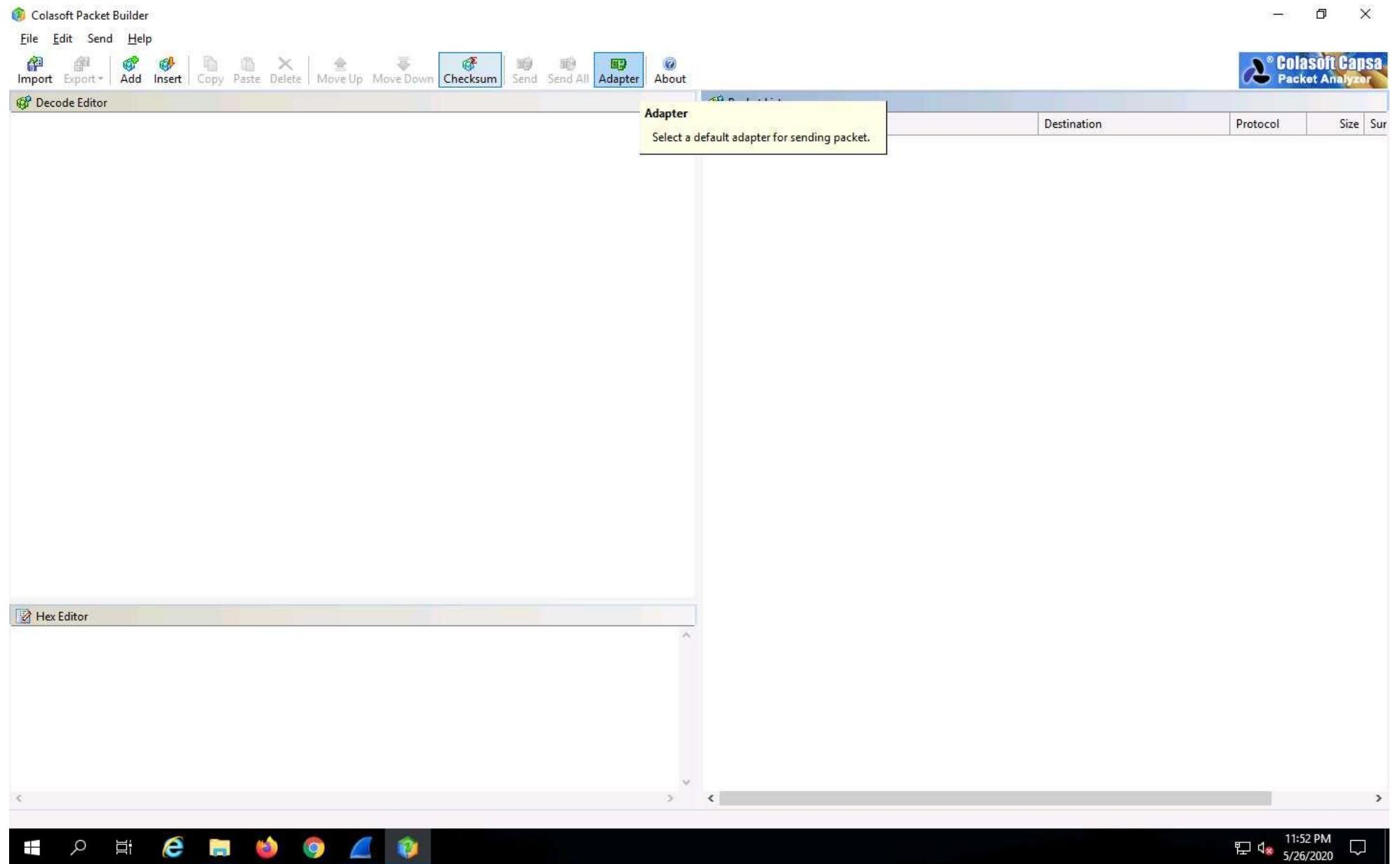
4. The **Wireshark Network Analyzer** main window appears; double-click the available ethernet or interface (here, **Ethernet**) to start the packet capture.
5. Navigate to **Z:\CEHv11 Module 03 Scanning Networks\Packet Crafting Tools\Colasoft Packet Builder** and double-click **pktbuilder_2.0.0.212.exe**.

6. Follow the wizard-driven installation steps to install **Colasoft Packet Builder**.
7. After the completion of the installation, click on the **Launch Colasoft Packet Builder 2.0** checkbox and click **Finish**.

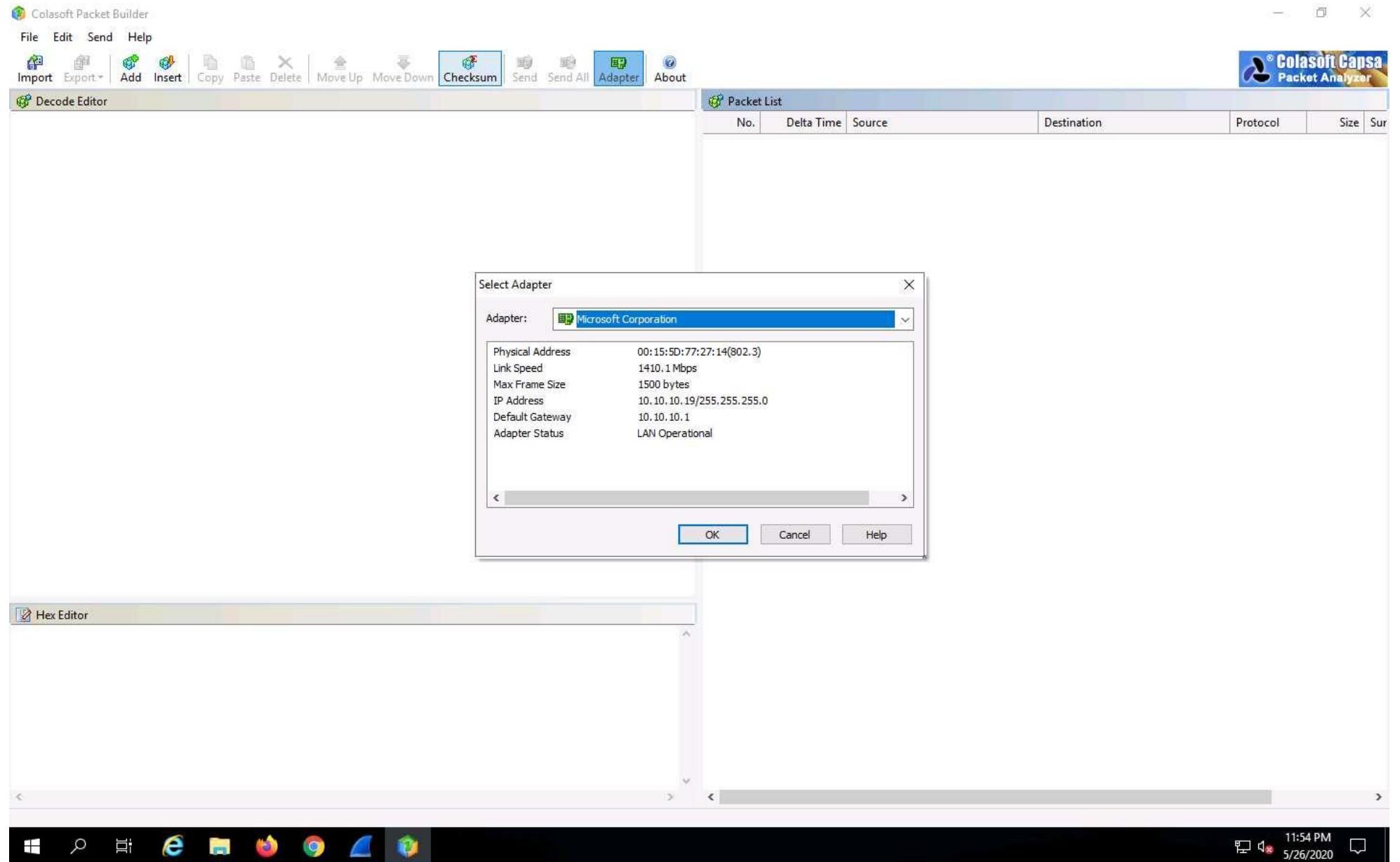


8. The **Colasoft Packet Builder** GUI appears; click on the **Adapter** icon, as shown in the screenshot.

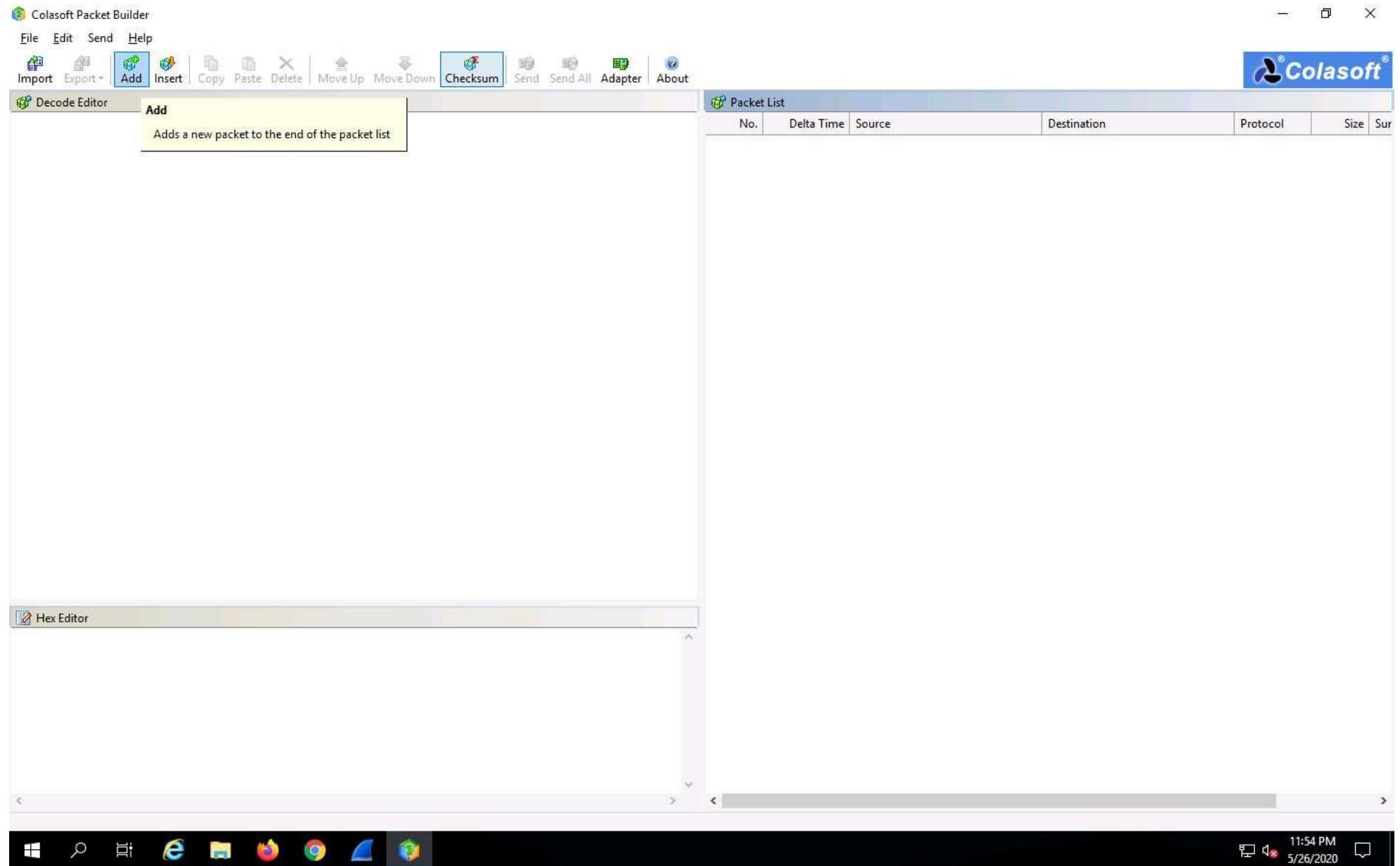
If a pop-up appears, close the window.



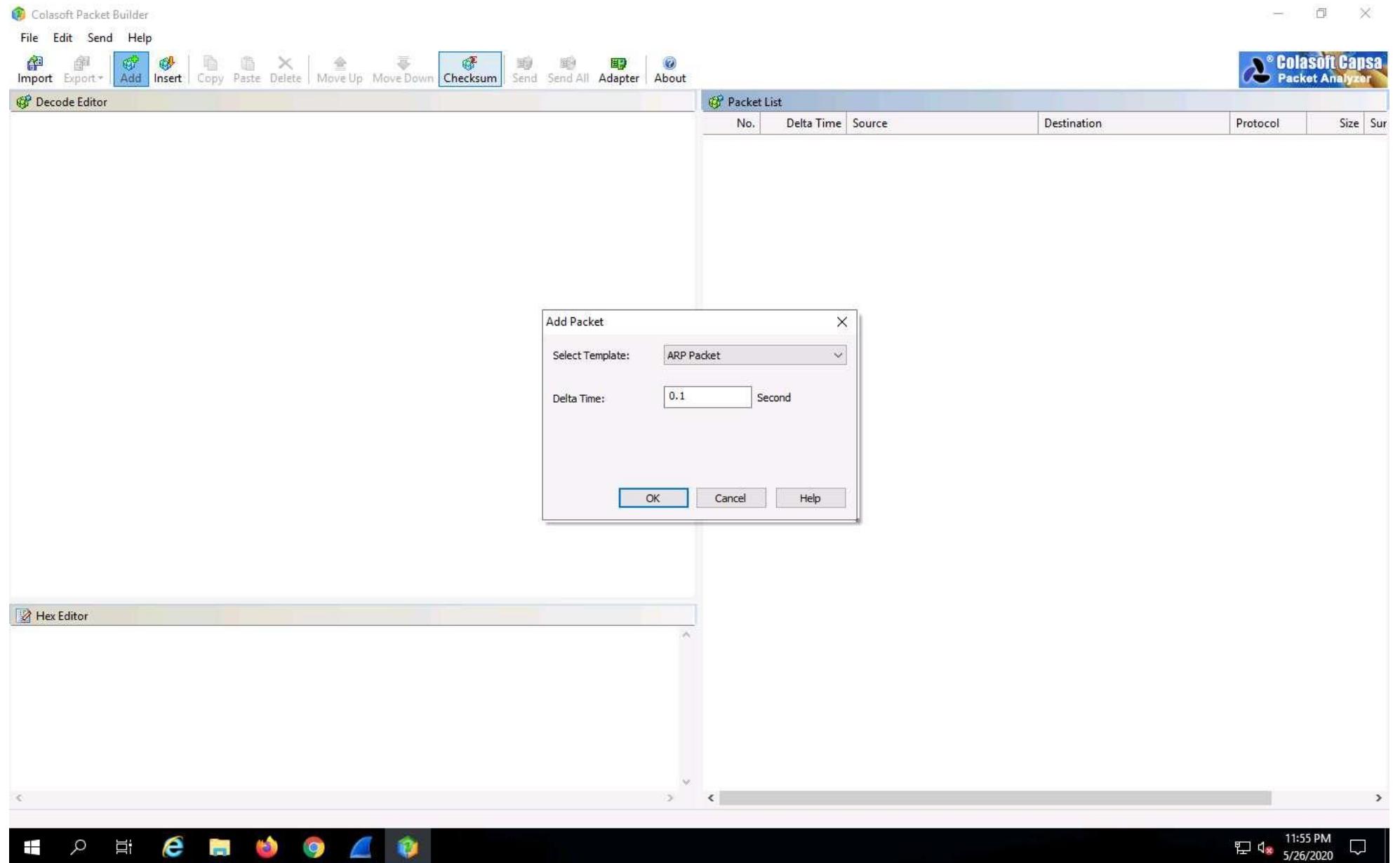
9. When the **Select Adapter** window appears, check the **Adapter** settings and click **OK**.



10. To add or create a packet, click the **Add** icon in the **Menu** bar.



11. In the **Add Packet** dialog box, select the **ARP Packet** template, set **Delta Time** as **0.1** seconds, and click **OK**.



12. You can view the added packets list on the right-hand side of the window, under **Packet List**.

Colasoft Packet Builder

File Edit Send Help

Import Export Add Insert Copy Paste Delete Move Up Move Down Checksum Send Send All Adapter About

Colasoft

Decode Editor

Packet Info:

- Packet Number: 000001
- Packet Length: 64
- Captured Length: 60
- Delta Time: 0.100000 Second

Ethernet Type II [0/14]

- Destination Address: FF:FF:FF:FF:FF:FF [0/6]
- Source Address: 00:00:00:00:00:00 [6/6]
- Protocol: 0x0806 (ARP) [12/2]

ARP - Address Resolution Protocol [14/28]

- Hardware type: 1 (Ethernet) [14/2]
- Protocol Type: 0x0800 [16/2]
- Hardware Address Length: 6 [18/1]
- Protocol Address Length: 4 [19/1]
- Type: 1 (ARP Request) [20/2]
- Source Physics: 00:00:00:00:00:00 [22/6]
- Source IP: 0.0.0.0 [28/4]
- Destination Physics: 00:00:00:00:00:00 [32/6]
- Destination IP: 0.0.0.0 [38/4]

Extra Data: [42/18]

- Number of Bytes: 18 bytes [42/18]

FCS:

- FCS: 0xF577BDD9 (Calculated)

Packet List

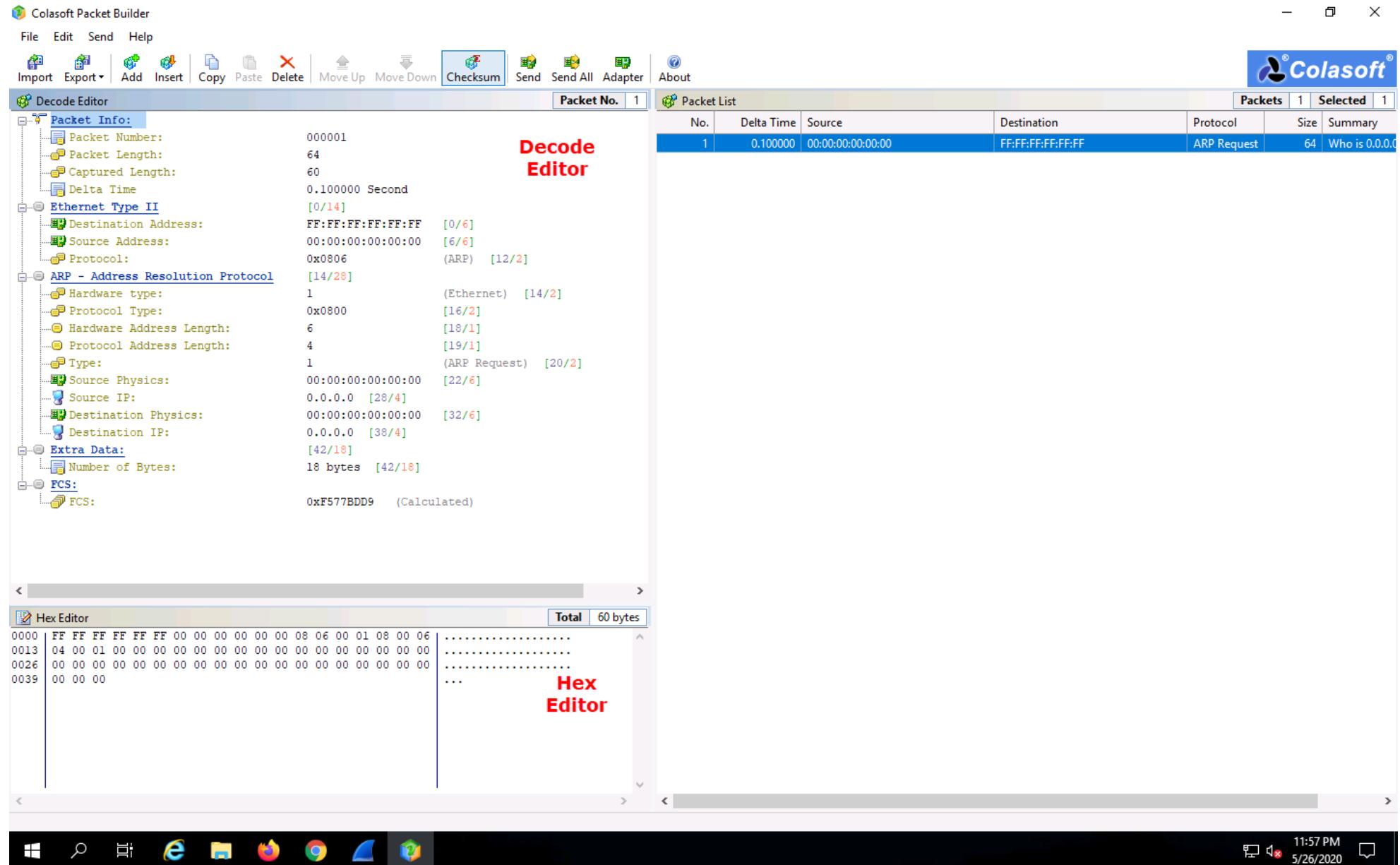
No.	Delta Time	Source	Destination	Protocol	Size	Summary
1	0.100000	00:00:00:00:00:00	FF:FF:FF:FF:FF:FF	ARP Request	64	Who is 0.0.0? Tell 0.0.0.

Hex Editor

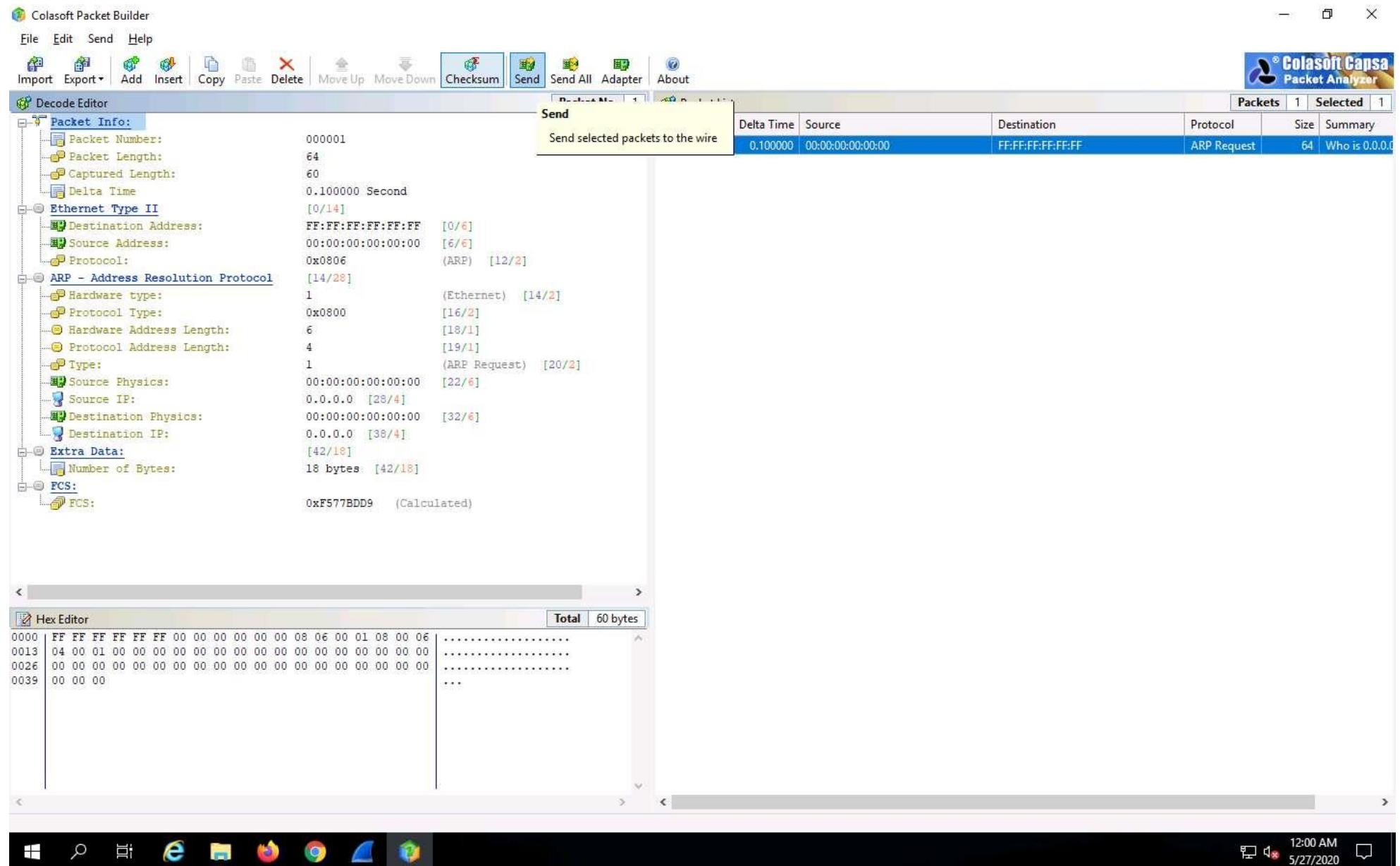
Total	60 bytes
0000	FF FF FF FF FF FF 00 00 00 00 00 00 08 06 00 01 08
0011	00 06 04 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00
0022	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0033	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Windows Taskbar: File Explorer, Edge, Firefox, Chrome, FileZilla, Colasoft Packet Builder, Task View, Start button, Search, Task View, 11:56 PM, 5/26/2020, Battery, Volume, Network, Chat

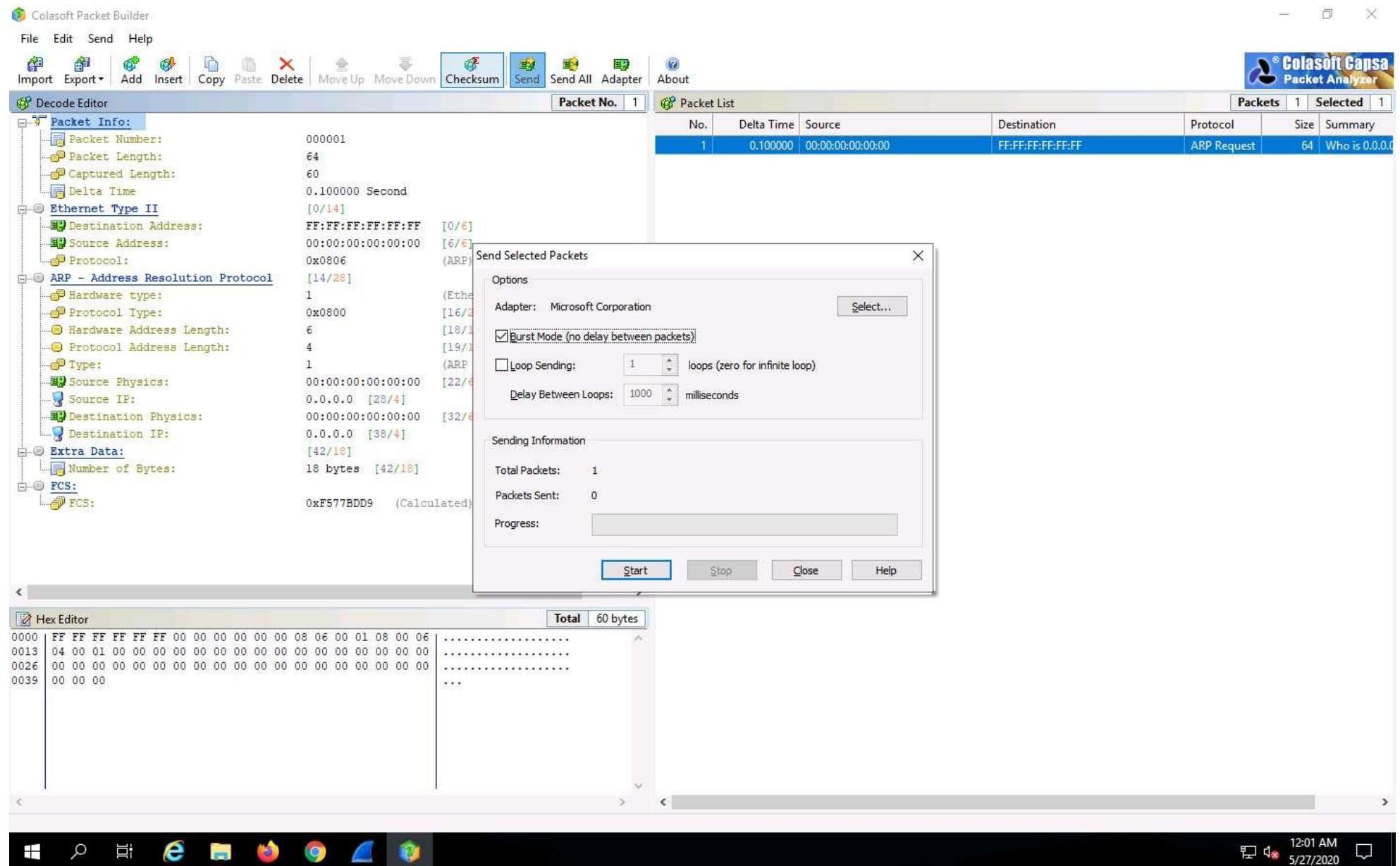
13. **Colasoft Packet Builder** allows you to edit the decoding information in the two editors, **Decode Editor** and **Hex Editor**, located in the left pane of the window.
- The **Decode Editor** section allows you to edit the packet decoding information by double-clicking the item that you wish to decode.
 - **Hex Editor** displays the actual packet contents in raw hexadecimal value on the left and its ASCII equivalent on the right.



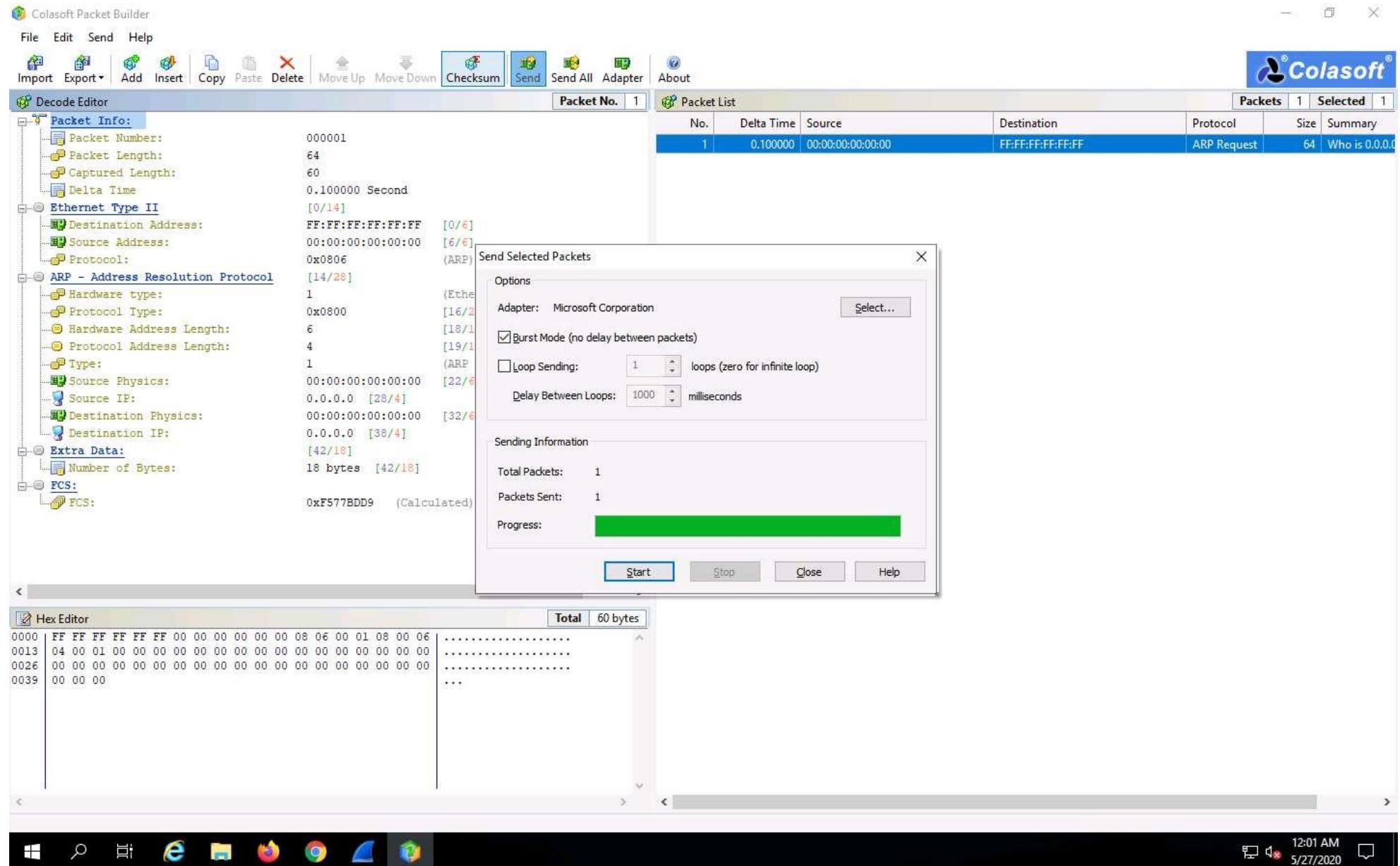
14. To send the packet, click **Send** from the **Menu** bar.



15. In the **Send Selected Packets** window, select the **Burst Mode (no delay between packets)** option, and then click **Start**.



16. After the **Progress** bar completes, click **Close**.



17. Now, when this ARP packet is broadcasted in the network, the active machines receive the packet, and a few start responding with an ARP reply. To evaluate which machine is responding to the ARP packet, you need to observe packets captured by the **Wireshark** tool.

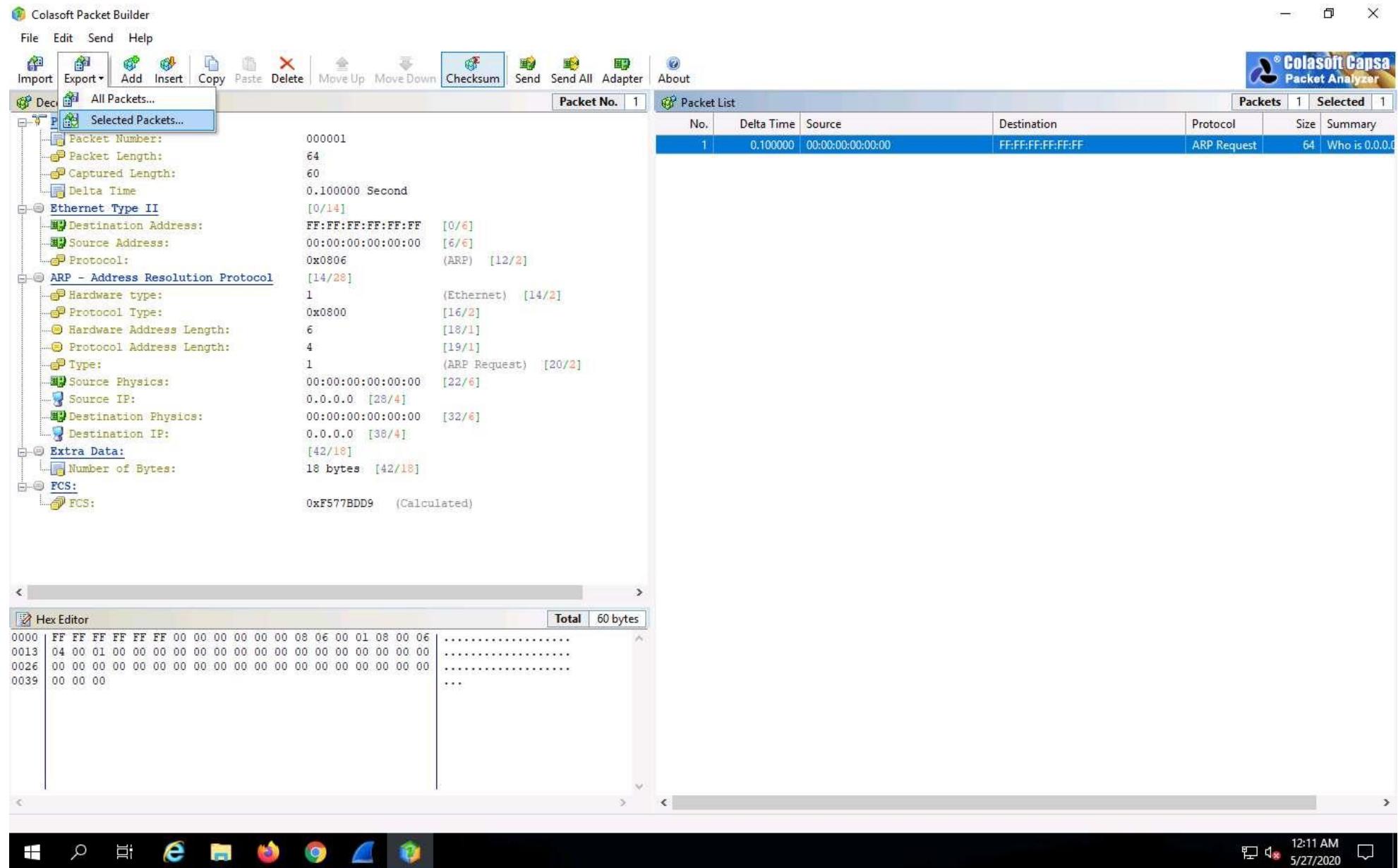
18. In the **Wireshark** window, click on the **Filter** field, type **arp** and press **Enter**. The ARP packets will be displayed, as shown in the screenshot.

Here, the host machine (**10.10.10.19**) is broadcasting ARP packets, prompting the target machines to reply to the message.

The screenshot shows the Wireshark interface with the following details:

- File menu:** File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, Help.
- Toolbar:** Standard icons for opening files, saving, zooming, and capturing.
- Filter bar:** Contains the filter text "arp".
- Panels:**
 - Packet List:** Shows a large number of ARP frames (e.g., 650, 651, 3494, 3495, 3638, 3639, 3647, 3648, 3718, 3719, 3836, 3837, 3848, 3849, 3938, 3939, 4032, 4033, 73858, 1097..., 1419..., 1420..., 1420..., 1420..., 1420..., 1420..., 1421..., 1421..., 1424..., 1424...) with details like source and destination MAC addresses and IP addresses, protocol (ARP), length, and info (e.g., "Who has 10.10.10.1? Tell 10.10.10.19").
 - Details pane:** Shows the structure of an ARP frame, including fields like Destination MAC, Source MAC, Operation (1), Destination IP, Source IP, and Hardware Type.
 - Bytes pane:** Shows the raw hex and ASCII representation of the ARP frame.
- Status bar:** Shows "Packets: 142891 · Displayed: 41 (0.0%) · Profile: Default".
- Taskbar:** Shows the Windows taskbar with icons for File Explorer, Edge, File History, Task View, and others.

19. Switch back to the **Colasoft Packet Builder** window, to export the packet, click **Export --> Selected Packets...**



20. In the **Save As** window, select a destination folder in the **Save in** field, specify **File name** and **Save as type**, and click **Save**.
 21. This saved file can be used for future reference.
 22. This concludes the demonstration of creating a custom TCP packets to scan the target host by bypassing the IDS/firewall.
 23. Close all open windows and document all the acquired information.
-

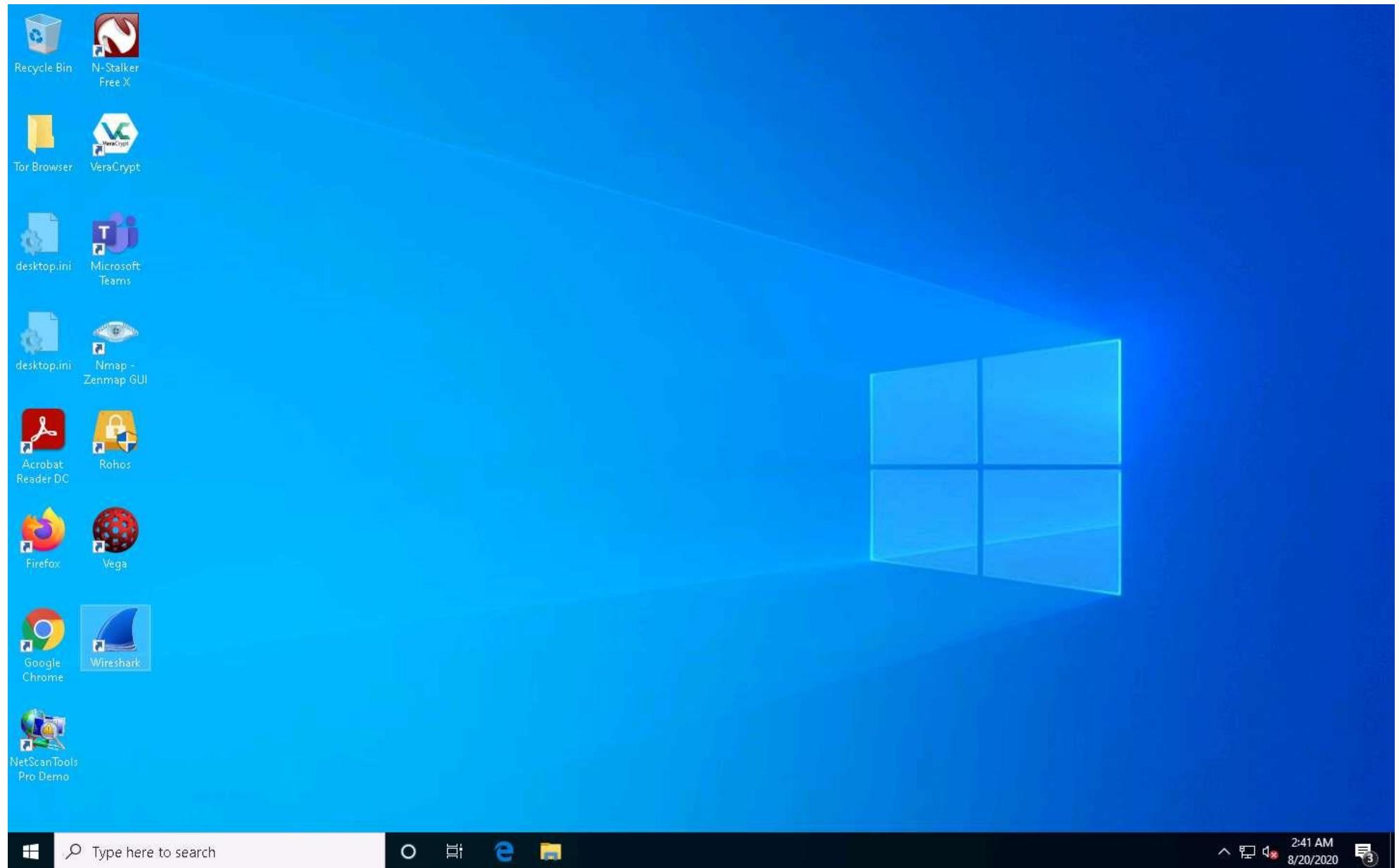
Task 3: Create Custom UDP and TCP Packets using Hping3 to Scan beyond IDS/Firewall

Hping3 is a scriptable program that uses the TCL language, whereby packets can be received and sent via a binary or string representation describing the packets.

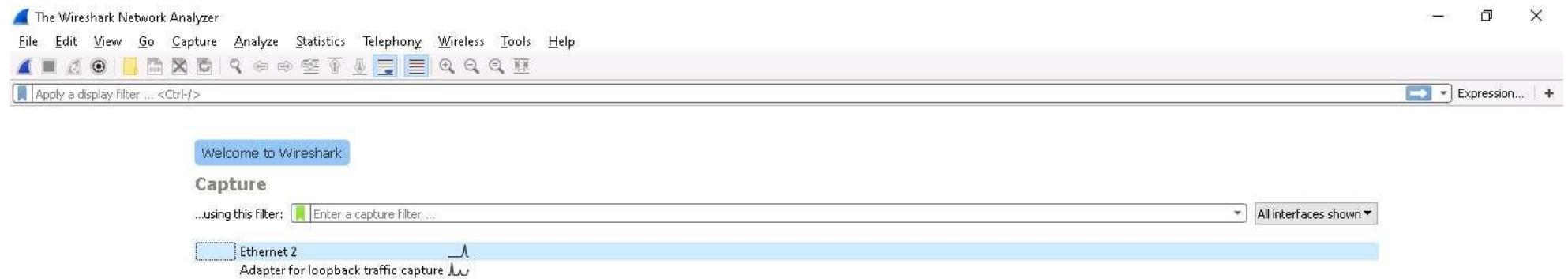
Here, we will use Hping3 to create custom UDP and TCP packets to evade the IDS/firewall in the target network.

Before beginning this task, ensure that the **Windows Defender Firewall** in the **Windows 10** machine is enabled.

1. Click [Windows 10](#) to switch to the **Windows 10** machine.
2. Navigate to the **Desktop**, double-click **Wireshark** shortcut.



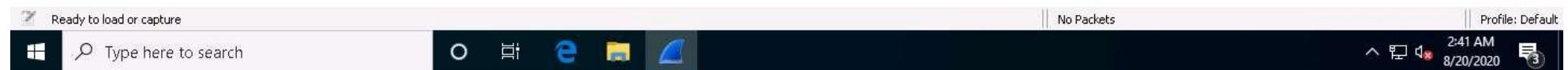
3. The **Wireshark Network Analyzer** window appears, double-click the available ethernet or interface (here, **Ethernet2**) to start the packet capture.



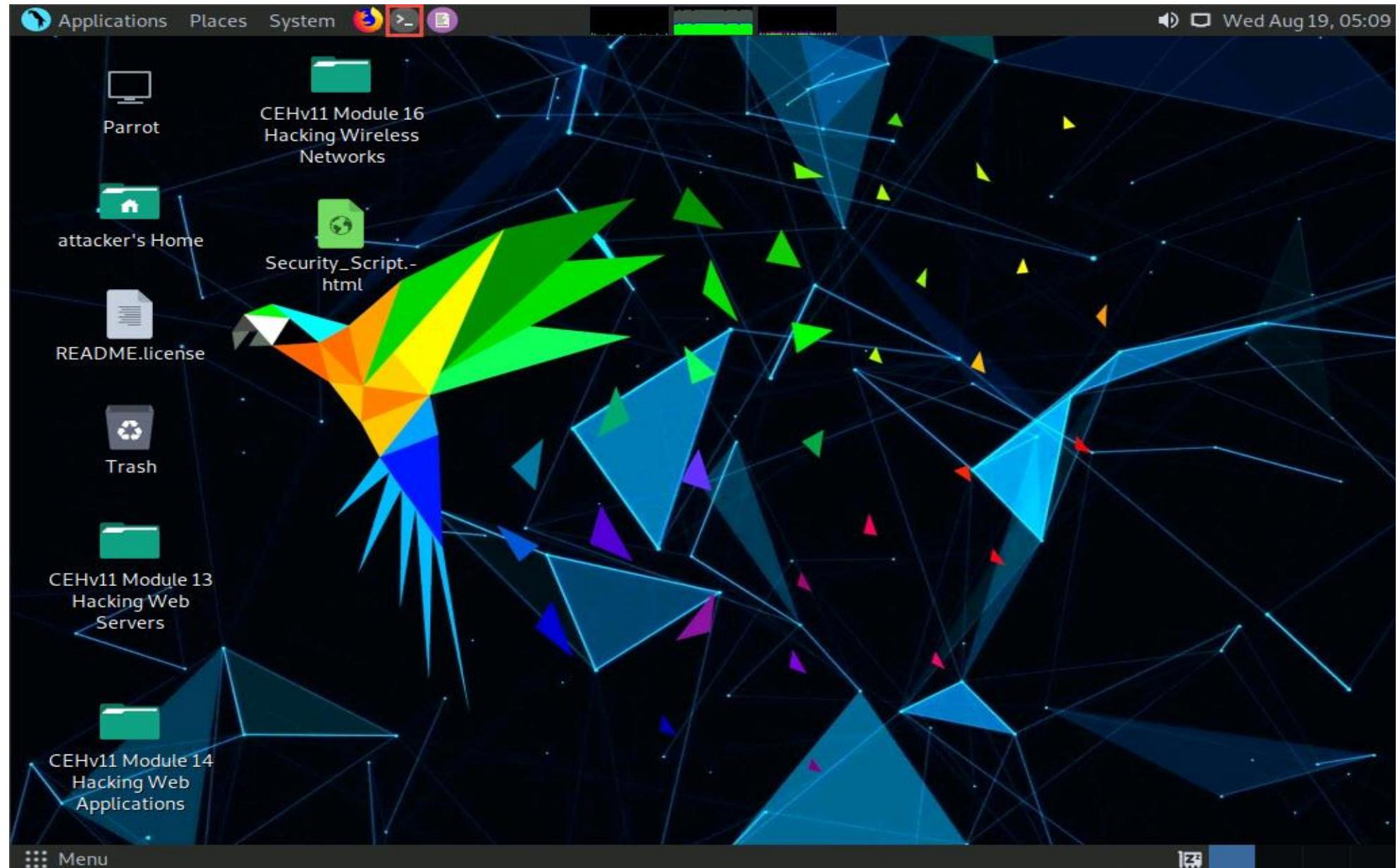
Learn

[User's Guide](#) · [Wiki](#) · [Questions and Answers](#) · [Mailing Lists](#)

You are running Wireshark 3.0.5 (v3.0.5-0-g752a55954770). You receive automatic updates.



4. Click **Parrot Security** to switch to the **Parrot Security** machine.
5. Click the **MATE Terminal** icon in the top-left corner of the **Desktop** window to open a **Terminal** window.



6. A **Parrot Terminal** window appears. In the terminal window, type **sudo su** and press **Enter** to run the programs as a root user.
7. In the **[sudo] password for attacker** field, type **toor** as a password and press **Enter**.

The password that you type will not be visible.

8. Now, type **cd** and press **Enter** to jump to the root directory.

A screenshot of the Parrot OS desktop environment. The desktop background is a dark, abstract geometric pattern. A terminal window titled "Parrot Terminal" is open in the top right corner, showing a command-line session:

```
[attacker@parrot] -[~] Module 16
└─$ sudo su      Hacking Wireless
[sudo] password for attacker:
[root@parrot] -[/home/attacker]
└─#cd
[root@parrot] -[~]
└─#
```

The desktop interface includes a menu bar with "Applications", "Places", "System", and "File", "Edit", "View", "Search", "Terminal", and "Help" options. A dock at the bottom features icons for "Parrot Terminal", "File Manager", "Terminal", and "Web Browser". On the left, a file browser window shows a folder structure with items like "README.license", "Trash", "CEHv11 Module 13 Hacking Web Servers", and "CEHv11 Module 14 Hacking Web Applications".

9. A Parrot Terminal window appears, type **hping3 [Target IP Address] --udp --rand-source --data 500** (here, the target machine is **Windows 10 [10.10.10.10]**) and press **Enter**.

Here, **--udp** specifies sending the UDP packets to the target host, **--rand-source** enables the random source mode and **--data** specifies the packet body size.

The screenshot shows a Parrot OS desktop environment. The terminal window is titled "Parrot Terminal" and contains the following command and output:

```
[attacker@parrot]~[-] Module16
└─$ sudo su
[sudo] password for attacker:
[root@parrot]~[/home/attacker]
└─# cd
[root@parrot]~[-]
└─# hping3 10.10.10.10 --udp --rand-source --data 500
HPING 10.10.10.10 (eth0 10.10.10.10): udp mode set, 28 headers + 500 data bytes
```

The desktop background features a dark, geometric pattern. On the left, there is a vertical dock with several icons and module names:

- CEHv11 Module 13: Hacking Wireless Servers
- CEHv11 Module 14: Hacking Web Applications
- CEHv11 Module 15: Hacking Network Protocols
- CEHv11 Module 16: Hacking Wireless Networks
- CEHv11 Module 17: Hacking Network Services
- CEHv11 Module 18: Hacking Cloud Services
- CEHv11 Module 19: Hacking IoT Devices
- CEHv11 Module 20: Hacking Mobile Devices
- CEHv11 Module 21: Hacking Cloud Services
- CEHv11 Module 22: Hacking Network Services
- CEHv11 Module 23: Hacking Network Protocols
- CEHv11 Module 24: Hacking Wireless Networks
- CEHv11 Module 25: Hacking Network Services

10. Now, click Windows 10 to switch to the **Windows 10** machine and observe the random UDP packets captured by **Wireshark**.

You can double-click any UDP packet and observe the detail.

12. Click **Parrot Security** to switch to the **Parrot Security** machine. In the **Parrot Terminal** window, first press **Control+C** and type **hping3 -S [Target IP Address] -p 80 -c 5** (here, target IP address is **10.10.10.10**), and then press **Enter**.

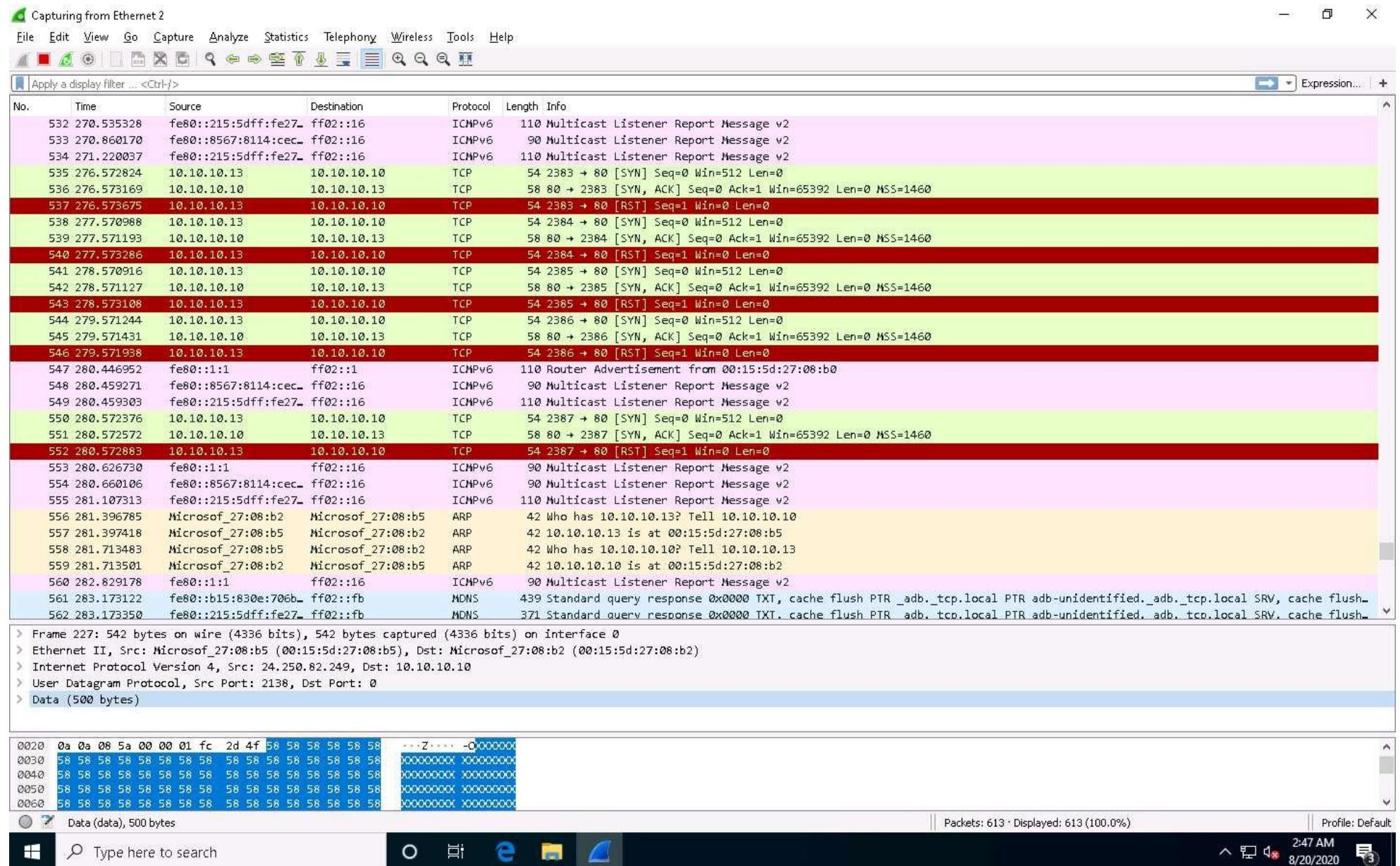
Here, **-S** specifies the TCP SYN request on the target machine, **-p** specifies assigning the port to send the traffic, and **-c** is the count of the packets sent to the target machine.

13. In the result, it is indicated that five packets were sent and received through port 80.

The screenshot shows a terminal window titled "Parrot Terminal" running on Parrot OS. The terminal displays a session where the user, "attacker", has gained root privileges ("root") on the "parrot" host. The user runs "hping3" to perform a SYN flood attack on a target IP of 10.10.10.10. The first part of the session shows an "HPING" attack with UDP mode, resulting in 100% packet loss. The second part shows a SYN flood attack with TCP mode, resulting in 0% packet loss. The terminal interface includes a menu bar with "Applications", "Places", "System", and "File Edit View Search Terminal Help". A system tray icon for "Parrot Terminal" is visible. The desktop background features a dark, abstract geometric pattern.

```
[attacker@parrot]~[-] Module16
└─$sudo su
[sudo] password for attacker:
[root@parrot]~[/home/attacker]
└─#cd
[root@parrot]~[-]
└─#hping3 10.10.10.10 --udp --rand-source --data 500
HPING 10.10.10.10 (eth0 10.10.10.10): udp mode set, 28 headers + 500 data bytes
^C
--- 10.10.10.10 hping statistic ---
88 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
[root@parrot]~[-]
└─#hping3 -S 10.10.10.10 -p 80 -c 5
HPING 10.10.10.10 (eth0 10.10.10.10): S set, 40 headers + 0 data bytes
len=44 ip=10.10.10.10 ttl=128 DF id=64870 sport=80 flags=SA seq=0 win=65392 rtt=6.2 ms
len=44 ip=10.10.10.10 ttl=128 DF id=64871 sport=80 flags=SA seq=1 win=65392 rtt=6.1 ms
len=44 ip=10.10.10.10 ttl=128 DF id=64872 sport=80 flags=SA seq=2 win=65392 rtt=5.9 ms
len=44 ip=10.10.10.10 ttl=128 DF id=64873 sport=80 flags=SA seq=3 win=65392 rtt=5.8 ms
len=44 ip=10.10.10.10 ttl=128 DF id=64874 sport=80 flags=SA seq=4 win=65392 rtt=5.6 ms
--- 10.10.10.10 hping statistic ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 5.6/5.9/6.2 ms
[root@parrot]~[-]
└─#
```

14. Now, click Windows 10 to switch to the target machine (i.e., **Windows 10**) and observe the TCP packets captured via **Wireshark**.



- Click **Parrot Security** to switch to the **Parrot Security** machine and try to flood the target machine (here, **Windows 10**) with TCP packets.
- In the **Parrot Terminal** window, type **hping3 [Target IP Address] --flood** (here, target IP address is **10.10.10.10**) and press **Enter**.

--flood: performs the TCP flooding.

17. Once you flood traffic to the target machine, it will respond in the hping3 terminal.

The screenshot shows a terminal window titled "Parrot Terminal" running on a Parrot OS desktop environment. The terminal displays a session where the user is performing a network attack. The session starts with the user becoming root via "sudo su". They then change directory to "/home/attacker" and run several instances of the hping3 tool to perform UDP and TCP flooding on the IP address 10.10.10.10. The terminal output shows the hping3 statistics for each type of traffic sent.

```
[attacker@parrot] -[~] Module16
└─$ sudo su          Hacking Wireless
[sudo] password for attacker:
[root@parrot] -[/home/attacker]
└─# cd
[root@parrot] -[~]
└─# hping3 10.10.10.10 --udp --rand-source --data 500
HPING 10.10.10.10 (eth0 10.10.10.10): udp mode set, 28 headers + 500 data bytes
^C
--- 10.10.10.10 hping statistic ---
88 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
[x]-[root@parrot] -[~]
└─# hping3 -S 10.10.10.10 -p 80 -c 5
HPING 10.10.10.10 (eth0 10.10.10.10): 5 set, 40 headers + 0 data bytes
len=44 ip=10.10.10.10 ttl=128 DF id=64870 sport=80 flags=SA seq=0 win=65392 rtt=6.2 ms
len=44 ip=10.10.10.10 ttl=128 DF id=64871 sport=80 flags=SA seq=1 win=65392 rtt=6.1 ms
len=44 ip=10.10.10.10 ttl=128 DF id=64872 sport=80 flags=SA seq=2 win=65392 rtt=5.9 ms
len=44 ip=10.10.10.10 ttl=128 DF id=64873 sport=80 flags=SA seq=3 win=65392 rtt=5.8 ms
len=44 ip=10.10.10.10 ttl=128 DF id=64874 sport=80 flags=SA seq=4 win=65392 rtt=5.6 ms
--- 10.10.10.10 hping statistic ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 5.6/5.9/6.2 ms
[root@parrot] -[~]
└─# hping3 10.10.10.10 --flood
HPING 10.10.10.10 (eth0 10.10.10.10): NO FLAGS are set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
```

18. Click **Windows 10** to switch to the **Windows 10** (target machine) and stop the packet capture in the **Wireshark** window after a while by click **Stop Capturing Packets** icon in the toolbar.
19. Observe the **Wireshark** window, which displays the TCP packet flooding from the host machine.

You can double-click the TCP packet stream to observe the TCP packet information.

*Ethernet 2

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

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

No.	Time	Source	Destination	Protocol	Length	Info
1995	413.420904	10.10.10.13	10.10.10.10	TCP	54	20610 → 0 [None] Seq=129917385 Win=512 Len=0
1995	413.420904	10.10.10.13	10.10.10.10	TCP	54	20611 → 0 [None] Seq=4147685616 Win=512 Len=0
1995	413.420988	10.10.10.13	10.10.10.10	TCP	54	20612 → 0 [None] Seq=66376711 Win=512 Len=0
1995	413.420989	10.10.10.13	10.10.10.10	TCP	54	20613 → 0 [None] Seq=4151639052 Win=512 Len=0
1995	413.420989	10.10.10.13	10.10.10.10	TCP	54	20614 → 0 [None] Seq=4231177897 Win=512 Len=0
1995	413.420990	10.10.10.13	10.10.10.10	TCP	54	20615 → 0 [None] Seq=3160010162 Win=512 Len=0
1995	413.420991	10.10.10.13	10.10.10.10	TCP	54	20616 → 0 [None] Seq=1423081129 Win=512 Len=0
1995	413.420991	10.10.10.13	10.10.10.10	TCP	54	[TCP Previous segment not captured] 20617 → 0 [None] Seq=1407048158 Win=512 Len=0
1995	413.420992	10.10.10.13	10.10.10.10	TCP	54	20618 → 0 [None] Seq=2608274530 Win=512 Len=0
1995	413.420992	10.10.10.13	10.10.10.10	TCP	54	20619 → 0 [None] Seq=1497143096 Win=512 Len=0
1995	413.420993	10.10.10.13	10.10.10.10	TCP	54	20620 → 0 [None] Seq=1206342186 Win=512 Len=0
1995	413.420993	10.10.10.13	10.10.10.10	TCP	54	20621 → 0 [None] Seq=3589077220 Win=512 Len=0
1995	413.420994	10.10.10.13	10.10.10.10	TCP	54	20622 → 0 [None] Seq=3747950841 Win=512 Len=0
1995	413.420994	10.10.10.13	10.10.10.10	TCP	54	20623 → 0 [None] Seq=495985964 Win=512 Len=0
1995	413.420995	10.10.10.13	10.10.10.10	TCP	54	20624 → 0 [None] Seq=4028089363 Win=512 Len=0
1995	413.420995	10.10.10.13	10.10.10.10	TCP	54	[TCP Previous segment not captured] 20625 → 0 [None] Seq=882054200 Win=512 Len=0
1995	413.420996	10.10.10.13	10.10.10.10	TCP	54	20626 → 0 [None] Seq=727451251 Win=512 Len=0
1995	413.420996	10.10.10.13	10.10.10.10	TCP	54	20627 → 0 [None] Seq=3340060773 Win=512 Len=0
1995	413.421089	10.10.10.13	10.10.10.10	TCP	54	20628 → 0 [None] Seq=515882615 Win=512 Len=0
1995	413.421090	10.10.10.13	10.10.10.10	TCP	54	20629 → 0 [None] Seq=421416592 Win=512 Len=0
1995	413.421090	10.10.10.13	10.10.10.10	TCP	54	20630 → 0 [None] Seq=493110813 Win=512 Len=0
1995	413.421091	10.10.10.13	10.10.10.10	TCP	54	20631 → 0 [None] Seq=649697607 Win=512 Len=0
1995	413.421091	10.10.10.13	10.10.10.10	TCP	54	20632 → 0 [None] Seq=123963910 Win=512 Len=0
1995	413.421092	10.10.10.13	10.10.10.10	TCP	54	20633 → 0 [None] Seq=3232483068 Win=512 Len=0
1995	413.421092	10.10.10.13	10.10.10.10	TCP	54	20634 → 0 [None] Seq=4038049953 Win=512 Len=0
1995	413.421093	10.10.10.13	10.10.10.10	TCP	54	20635 → 0 [None] Seq=3151615200 Win=512 Len=0
1995	413.421093	10.10.10.13	10.10.10.10	TCP	54	20636 → 0 [None] Seq=2942208221 Win=512 Len=0
1995	413.421094	10.10.10.13	10.10.10.10	TCP	54	20637 → 0 [None] Seq=64787610 Win=512 Len=0
1995	413.421153	10.10.10.13	10.10.10.10	TCP	54	20638 → 0 [None] Seq=519022899 Win=512 Len=0
1995	413.421154	10.10.10.13	10.10.10.10	TCP	54	20639 → 0 [None] Seq=3928052818 Win=512 Len=0
1995	413.421154	10.10.10.13	10.10.10.10	TCP	54	20640 → 0 [None] Seq=370118653 Win=512 Len=0
1995	413.421176	10.10.10.13	10.10.10.10	TCP	54	20641 → 0 [None] Seq=2860039674 Win=512 Len=0
1995	413.421177	10.10.10.13	10.10.10.10	TCP	54	20642 → 0 [None] Seq=513115964 Win=512 Len=0
1995	413.421177	10.10.10.13	10.10.10.10	TCP	54	[TCP Previous segment not captured] 20643 → 0 [None] Seq=1985658822 Win=512 Len=0

> Frame 1995554: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface 0

> Ethernet II, Src: Microsoft_22:7f:8f (00:15:5d:22:7f:8f), Dst: Microsoft_22:7f:8c (00:15:5d:22:7f:8c)

> Internet Protocol Version 4, Src: 10.10.10.13, Dst: 10.10.10.10

> Transmission Control Protocol, Src Port: 20661, Dst Port: 0, Seq: 3609109263, Len: 0

```
0000 00 15 5d 22 7f 8c 00 15 5d 22 7f 8f 08 00 45 00  ..]".... ]"....E
0010 00 28 2c e0 00 00 40 06 25 c6 0a 0a 0d 0a 0a  (,...@ %.....-
0020 0a 0a 50 b5 00 00 1f 58 9f 7f 4d 24 3c b4 50 00  ..P....X ..M$<..P-
```

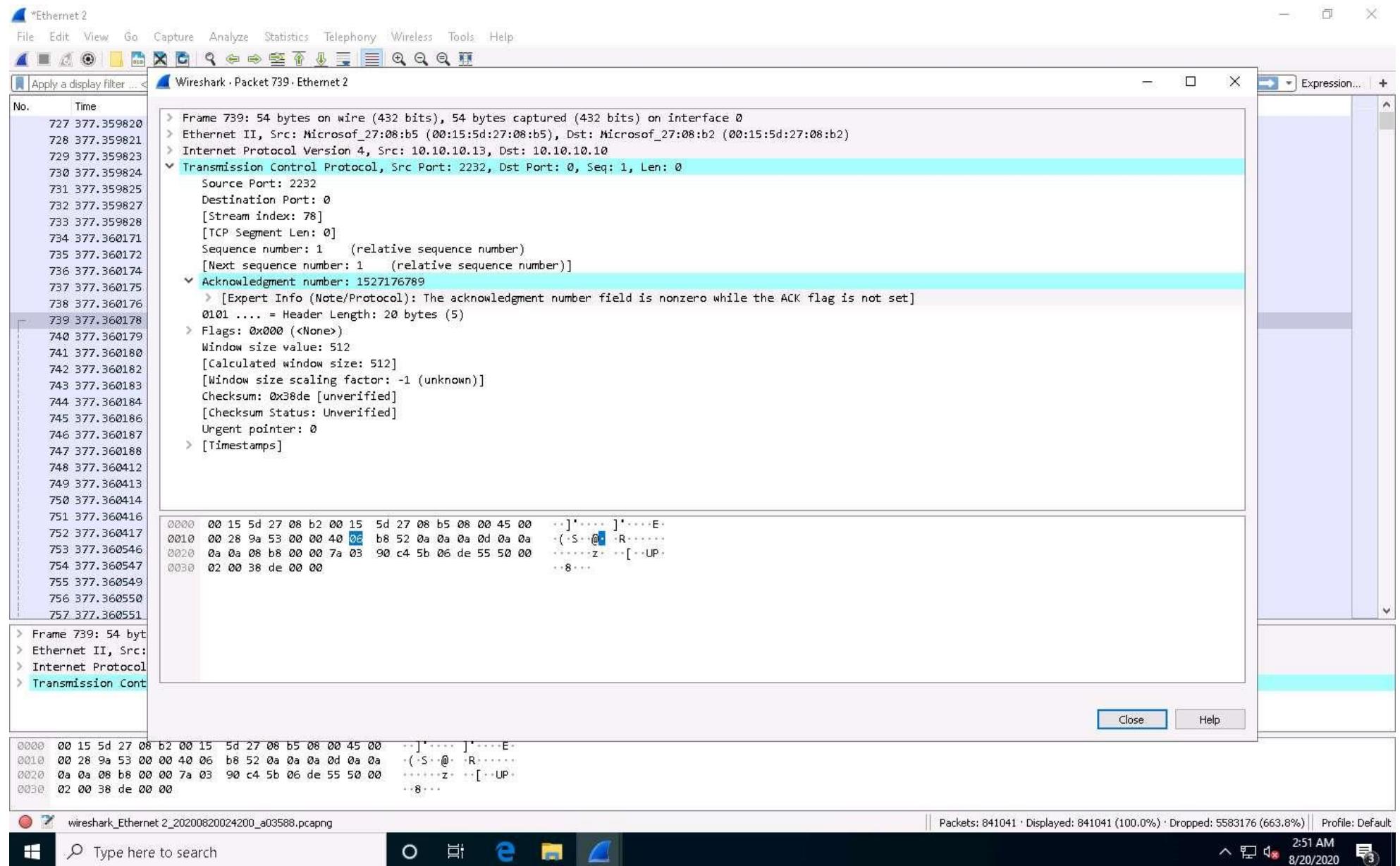
wireshark_Ethernet 2_20200528005930_a02604.pcapng

Packets: 2294503 · Displayed: 2294503 (100.0%) · Dropped: 3656997 (159.4%) · Profile: Default

Type here to search

1:07 AM 5/28/2020

20. The TCP packet stream displays the complete information of TCP packets such as the source and destination of the captured packet, source port, destination port, etc.



- This concludes the demonstration of evading the IDS and firewall using various evasion techniques in Hping3.
- Close all open windows and document all the acquired information.

Task 4: Create Custom Packets using Nmap to Scan beyond IDS/Firewall

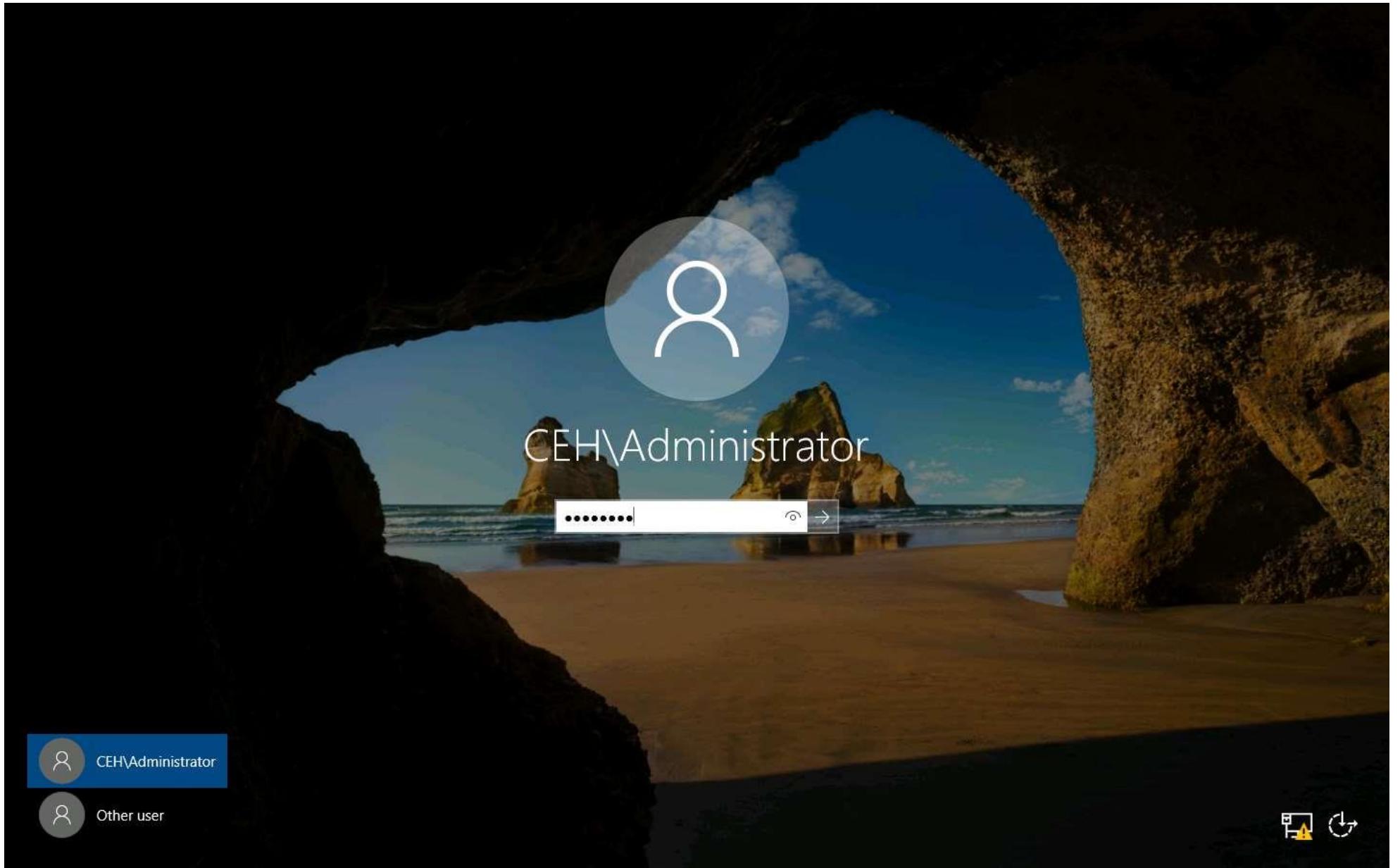
Nmap is a network scanning tool that can be used for sending customized data packets to scan the target host, thus bypassing various security mechanisms such as the IDS/firewall.

Here, we will use Nmap to perform various scanning techniques such as appending custom binary data, appending a custom string, appending random data, randomizing host order, and sending bad checksums to scan the target host beyond the IDS/firewall.

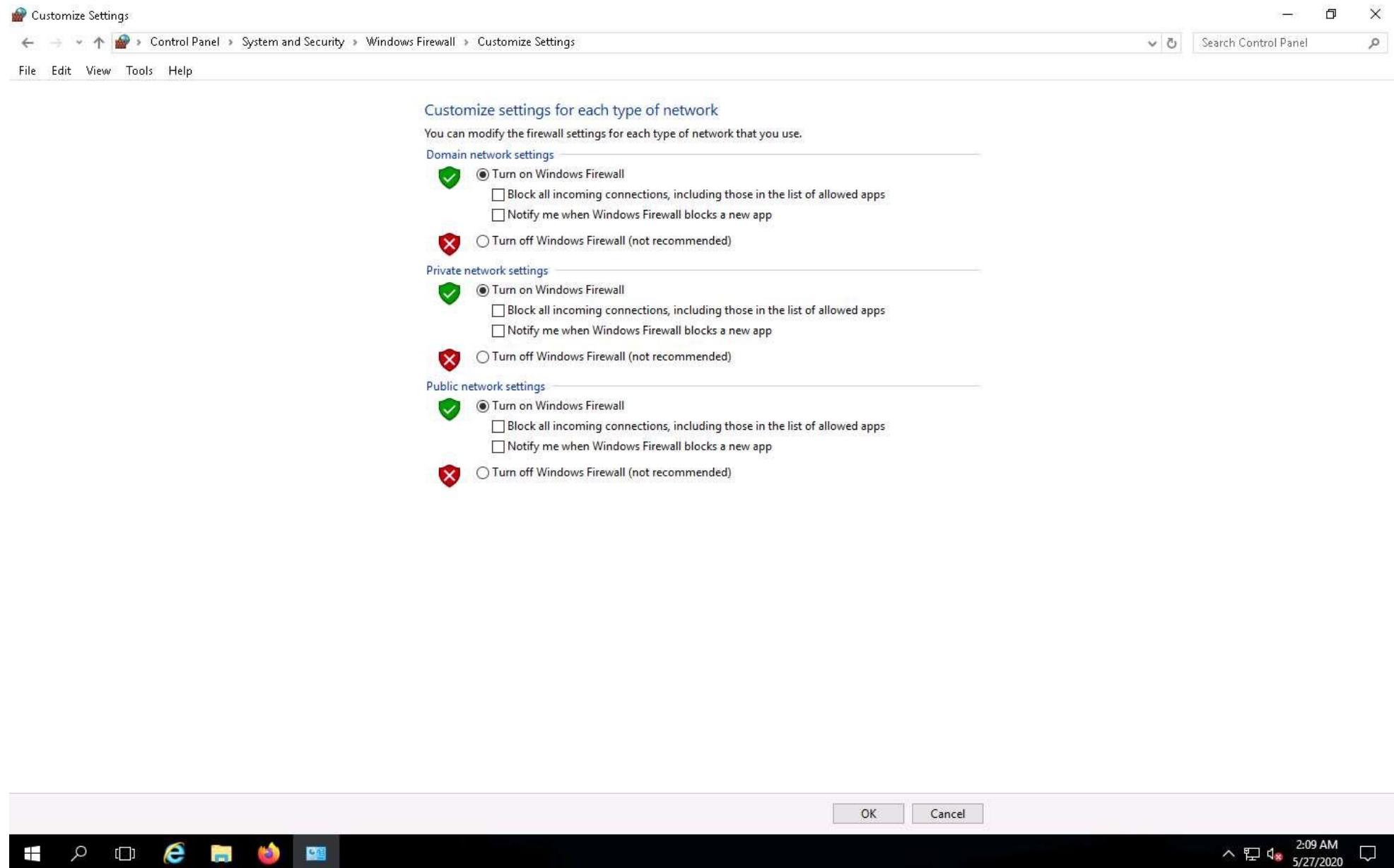
In this task, we are using the **Windows 10 (10.10.10.10)** machine as a host machine and the **Windows Server 2016 (10.10.10.16)** machine as a target machine.

1. Click **Windows Server 2016** to switch to the **Windows Server 2016** machine.
2. Click **Ctrl+Alt+Delete** to activate the machine. By default, **Administration** user profile is selected, click **Pa\$\$w0rd** to paste the password in the Password field and press **Enter** to login.

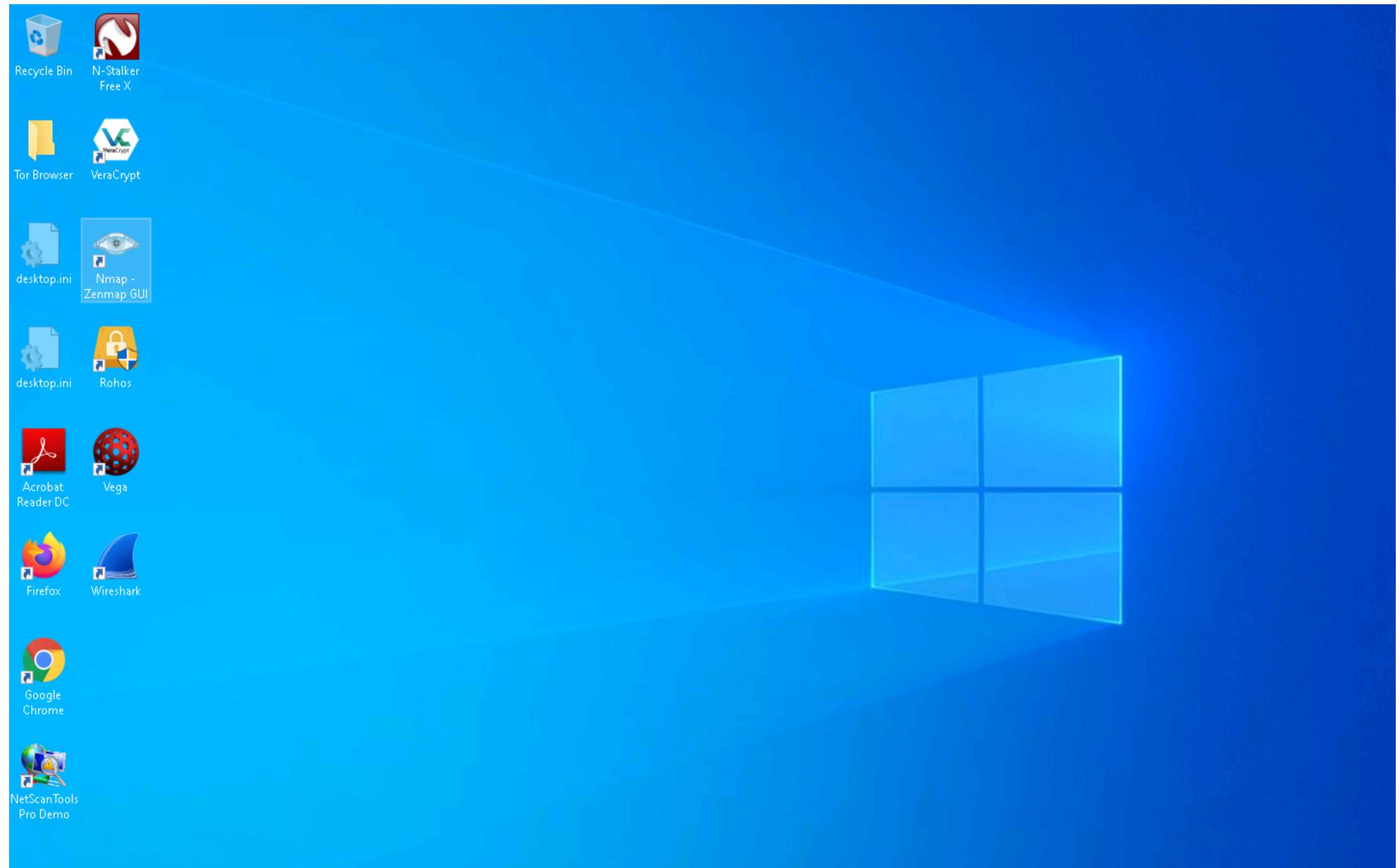
Alternatively, you can also click **Pa\$\$w0rd** under **Windows Server 2016** machine thumbnail in the **Resources** pane or Click **Type Text | Type Password** button under Commands (**thunder** icon) menu.



3. Navigate to **Control Panel** --> **System and Security** --> **Windows Firewall** --> **Turn Windows Firewall on or off**, enable Windows Firewall and click **OK**, as shown in the screenshot.



4. Click **Windows 10** to switch to the **Windows 10** machine and launch **Nmap** by double-clicking on the **Nmap - Zenmap GUI** shortcut available on the **Desktop**.



5. The **Nmap - Zenmap GUI** appears. In the **Command** field, type the command **nmap [Target IP Address] --data 0xdeadbeef** (here, target IP address is **10.10.10.16**) and click **Scan**.

Nmap uses **--data [hex string]** (here, **0xdeadbeef**) to send the binary data (0's and 1's) as payloads in the sent packets to scan beyond firewalls.

6. The scan results appear, displaying all open TCP ports and services running on the target machine, as shown in the screenshot.

The screenshot shows the Zenmap interface with the following details:

- Target:** 10.10.10.16
- Command:** nmap 10.10.10.16 --data 0xdeadbeef
- Hosts Tab:** Selected. Shows one host, 10.10.10.16.
- Services Tab:** Shows a table of open ports and services:

PORT	STATE	SERVICE
53/tcp	open	domain
80/tcp	open	http
88/tcp	open	kerberos-sec
135/tcp	open	msrpc
139/tcp	open	netbios-ssn
389/tcp	open	ldap
445/tcp	open	microsoft-ds
464/tcp	open	kpasswd5
593/tcp	open	http-rpc-epmap
636/tcp	open	ldaps
1079/tcp	open	asprovatalk
1801/tcp	open	msm
2103/tcp	open	zephyr-clt
2195/tcp	open	eklogin
2107/tcp	open	msm-mgt
3268/tcp	open	globalcatLDAP
3269/tcp	open	globalcatLDAPssl
3389/tcp	open	ms-wbt-server
- OS Tab:** Host 10.10.10.16 is listed.
- Details:** A button in the top right corner of the main pane.
- Bottom Bar:** Includes a "Filter Hosts" input field, a search bar, and a system tray with icons for battery, signal, volume, and date/time (5/27/2020, 5:16 AM).

7. In the **Command** field, type the command **nmap [Target IP Address] --data-string "Ph34r my l33t skills"** (here, target IP address is **10.10.10.16**) and click **Scan**.

Nmap uses **--data-string [string]** (here, "**Ph34r my l33t skills**") to send a regular string as payloads in the sent packets to the target machine for scanning beyond the firewall.

8. The scan results appear, displaying all open TCP ports and services running on the target machine, as shown in the screenshot.

Zenmap

Scan Tools Profile Help

Target: 10.10.10.16 Profile: Scan Cancel

Command: nmap 10.10.10.16 --data-string "Ph34r my l33t skills"

Hosts Services Nmap Output Ports / Hosts Topology Host Details Scans

OS Host ▾ 10.10.10.16

nmap 10.10.10.16 --data-string "Ph34r my l33t skills"

Starting Nmap 7.80 (https://nmap.org) at 2020-05-27 05:18 Eastern Daylight Time

Nmap scan report for 10.10.10.16

Host is up (0.00064s latency).

Not shown: 982 filtered ports

PORT	STATE	SERVICE
53/tcp	open	domain
80/tcp	open	http
88/tcp	open	kerberos-sec
135/tcp	open	msrpc
139/tcp	open	netbios-ssn
389/tcp	open	ldap
445/tcp	open	microsoft-ds
464/tcp	open	kpasswd5
593/tcp	open	http-rpc-epmap
636/tcp	open	ldaps
1079/tcp	open	asprovatalk
1801/tcp	open	msm
2103/tcp	open	zephyr-clt
2105/tcp	open	eklogin
2187/tcp	open	msm-gmt
3268/tcp	open	globalcatLDAP
3269/tcp	open	globalcatLDAPssl
3389/tcp	open	ms-wbt-server

MAC Address: 00:15:5D:27:9F:A3 (Microsoft)

Nmap done: 1 IP address (1 host up) scanned in 4.83 seconds

Filter Hosts

Type here to search

5:19 AM 5/27/2020

9. In the **Command** field, type the command **nmap --data-length 5 [Target IP Address]** (here, the target IP address is **10.10.10.16**) and click **Scan**.

Nmap uses **--data-length [len]** (here, **5**) to append the number of random data bytes to most of the packets sent without any protocol-specific payloads.

10. The scan results appear, displaying all open TCP ports and services running on the target machine, as shown in the screenshot.

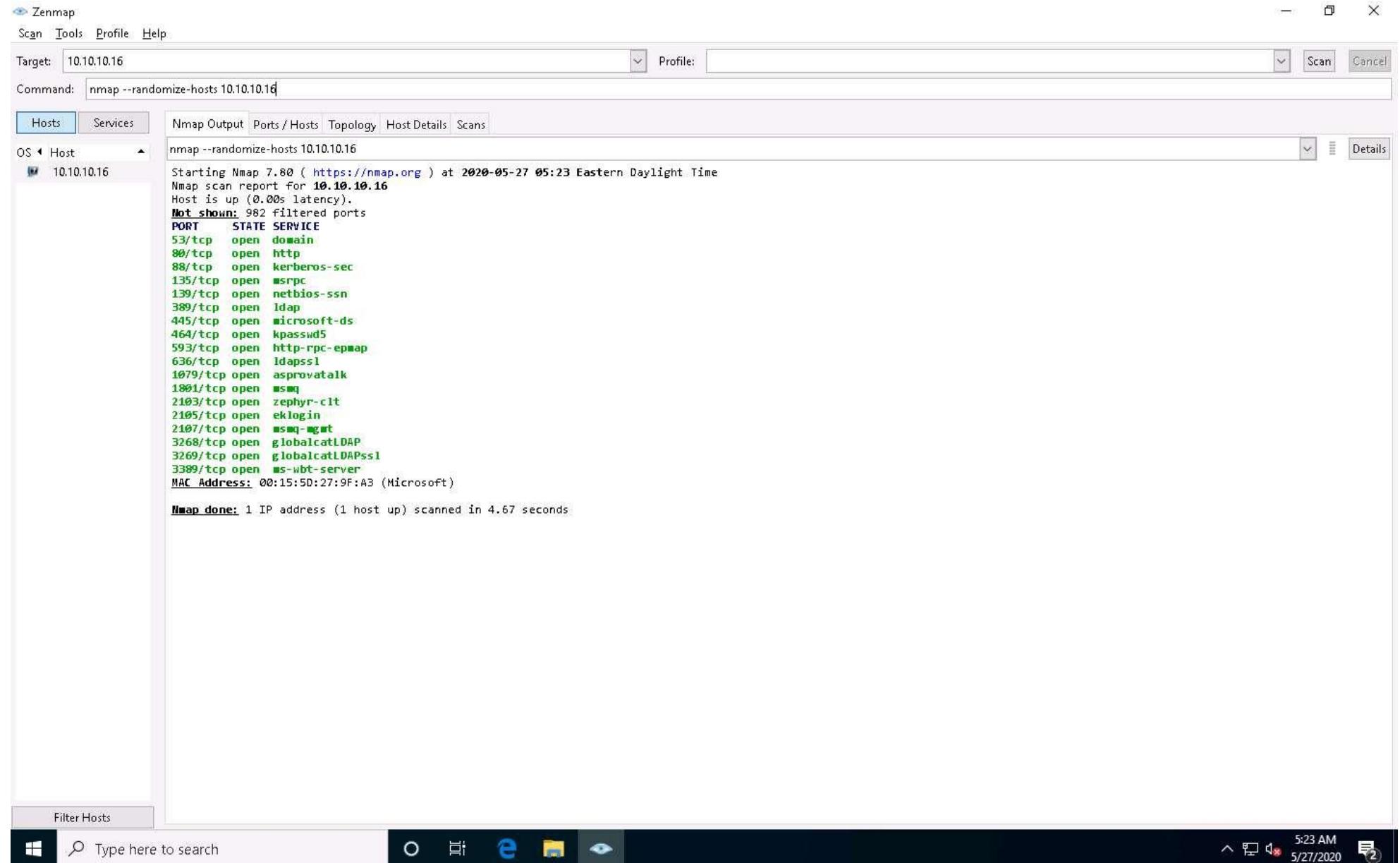
The screenshot shows the Zenmap interface with the following details:

- Target:** 10.10.10.16
- Command:** nmap --data-length 5 10.10.10.16
- Hosts:** 10.10.10.16
- OS:** Microsoft Windows 10 Pro
- Services:**
 - 53/tcp open domain
 - 80/tcp open http
 - 88/tcp open kerberos-sec
 - 135/tcp open msrpc
 - 139/tcp open netbios-ssn
 - 389/tcp open ldap
 - 445/tcp open microsoft-ds
 - 464/tcp open kpasswd5
 - 593/tcp open http-rpc-epmap
 - 636/tcp open ldaps
 - 1079/tcp open asprovatalk
 - 1801/tcp open msad
 - 2103/tcp open zephyr-clt
 - 2195/tcp open eklogin
 - 2197/tcp open msad-wgmt
 - 3268/tcp open globalcatLDAP
 - 3269/tcp open globalcatLDAPssl
 - 3389/tcp open ms-wbt-server
- MAC Address:** 00:15:5D:27:9F:A3 (Microsoft)
- Scan Output:** Starting Nmap 7.80 (https://nmap.org) at 2020-05-27 05:21 Eastern Daylight Time
Nmap scan report for 10.10.10.16
Host is up (0.00s latency).
Not shown: 982 filtered ports
PORT STATE SERVICE
53/tcp open domain
80/tcp open http
88/tcp open kerberos-sec
135/tcp open msrpc
139/tcp open netbios-ssn
389/tcp open ldap
445/tcp open microsoft-ds
464/tcp open kpasswd5
593/tcp open http-rpc-epmap
636/tcp open ldaps
1079/tcp open asprovatalk
1801/tcp open msad
2103/tcp open zephyr-clt
2195/tcp open eklogin
2197/tcp open msad-wgmt
3268/tcp open globalcatLDAP
3269/tcp open globalcatLDAPssl
3389/tcp open ms-wbt-server
MAC Address: 00:15:5D:27:9F:A3 (Microsoft)
Nmap done: 1 IP address (1 host up) scanned in 5.13 seconds

11. In the **Command** field, type the command **nmap --randomize-hosts [Target IP Address]** (here, the target IP address is **10.10.10.16**) and click **Scan**.

Nmap uses **--randomize-hosts** to scan the number of hosts in the target network in random order to scan the intended target that is beyond the firewall.

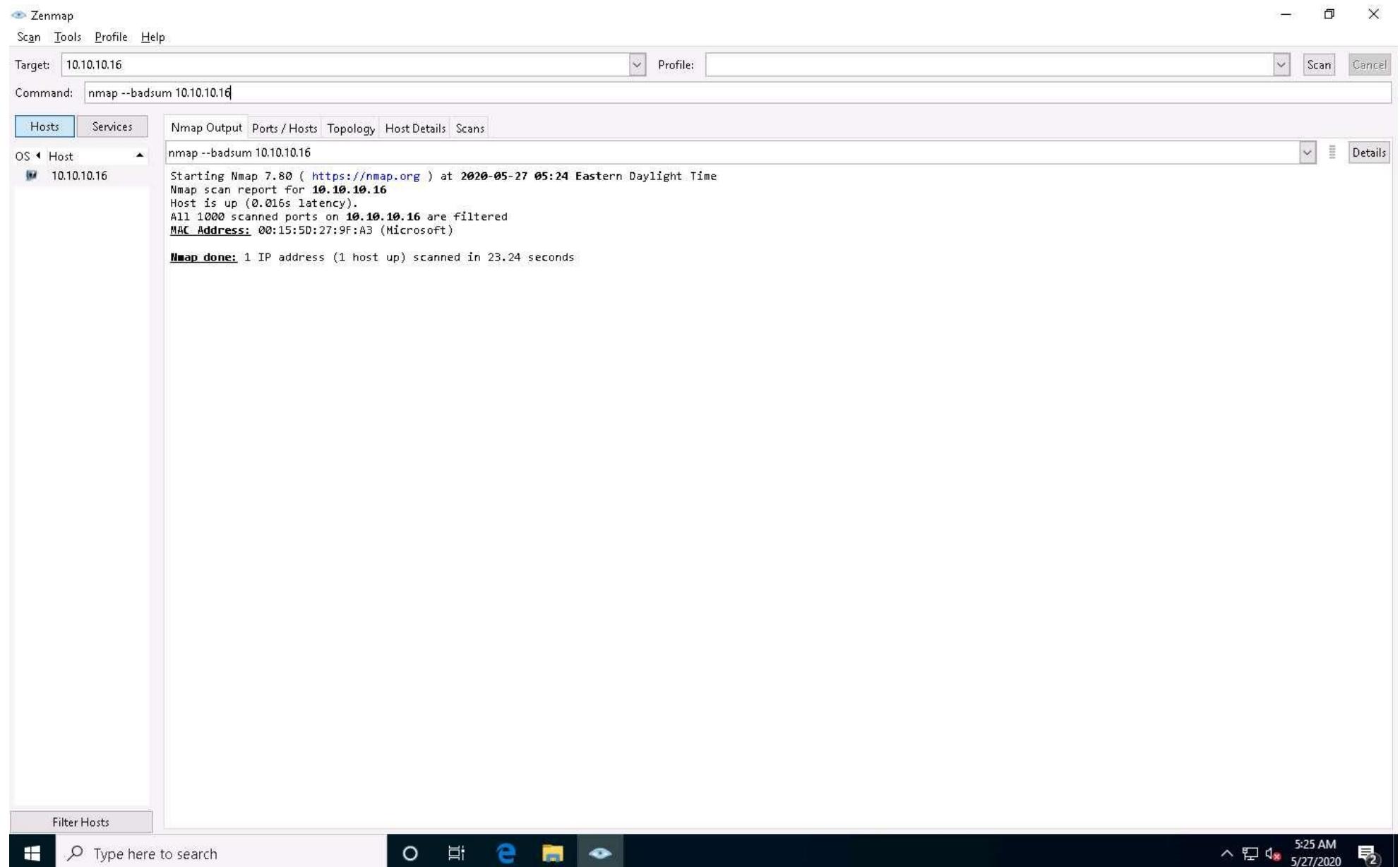
12. The scan results appear, displaying all open TCP ports and services running on the target machine, as shown in the screenshot.



13. In the **Command** field, type the command **nmap --badsum [Target IP Address]** (here, the target IP address is **10.10.10.16**) and click **Scan**.

Nmap uses **--badsum** to send the packets with bad or bogus TCP/UPD checksums to the intended target to avoid certain firewall rulesets.

14. The scan results appear, demonstrating that all ports are filtered, indicating that there is no response or the packets are dropped, and thus it can be inferred that the system is configured.



15. This concludes the demonstration of creating custom packets using Nmap to scan beyond the IDS and firewall.
16. You can also use other packet crafting tools such as **NetScanTools Pro** (<https://www.netscantools.com>), **Ostinato** (<https://www.ostinato.org>), and **WAN Killer** (<https://www.solarwinds.com>) to build custom packets to evade security mechanisms.
17. Close all open windows and document all the acquired information.
18. Turn off the **Windows Firewall** in the **Windows 10** and **Windows Server 2016** machines by navigating to **Control Panel --> System and Security --> Windows Defender Firewall --> Turn Windows Defender Firewall on or off.**

Customize settings for each type of network

You can modify the firewall settings for each type of network that you use.

Private network settings



Turn on Windows Defender Firewall

Block all incoming connections, including those in the list of allowed apps

Notify me when Windows Defender Firewall blocks a new app



Turn off Windows Defender Firewall (not recommended)

Public network settings



Turn on Windows Defender Firewall

Block all incoming connections, including those in the list of allowed apps

Notify me when Windows Defender Firewall blocks a new app



Turn off Windows Defender Firewall (not recommended)

OKCancel