

Lab 3: Perform OS Discovery

Lab Scenario

As a professional ethical hacker or a pen tester, the next step after discovering the open ports and services running on the target range of IP addresses is to perform OS discovery. Identifying the OS used on the target system allows you to assess the system’s vulnerabilities and the exploits that might work on the system to perform additional attacks.

Lab Objectives

- Identify the target system’s OS with Time-to-Live (TTL) and TCP window sizes using Wireshark
- Perform OS discovery using Nmap Script Engine (NSE)
- Perform OS discovery using Unicornscan

Overview of OS Discovery/ Banner Grabbing

Banner grabbing, or OS fingerprinting, is a method used to determine the OS that is running on a remote target system.

There are two types of OS discovery or banner grabbing techniques:

- **Active Banner Grabbing** Specially crafted packets are sent to the remote OS, and the responses are noted, which are then compared with a database to determine the OS. Responses from different OSes vary, because of differences in the TCP/IP stack implementation.
- **Passive Banner Grabbing** This depends on the differential implementation of the stack and the various ways an OS responds to packets. Passive banner grabbing includes banner grabbing from error messages, sniffing the network traffic, and banner grabbing from page extensions.

Parameters such as TTL and TCP window size in the IP header of the first packet in a TCP session plays an important role in identifying the OS running on the target machine. The TTL field determines the maximum time a packet can remain in a network, and the TCP window size determines the length of the packet reported. These values differ for different OSes: you can refer to the following table to learn the TTL values and TCP window size associated with various OSes.

Operating System (OS)	Time To Live	TCP Window Size
Linux (Kernel 2.4 and 2.6)	64	5840
Google Linux	64	5720

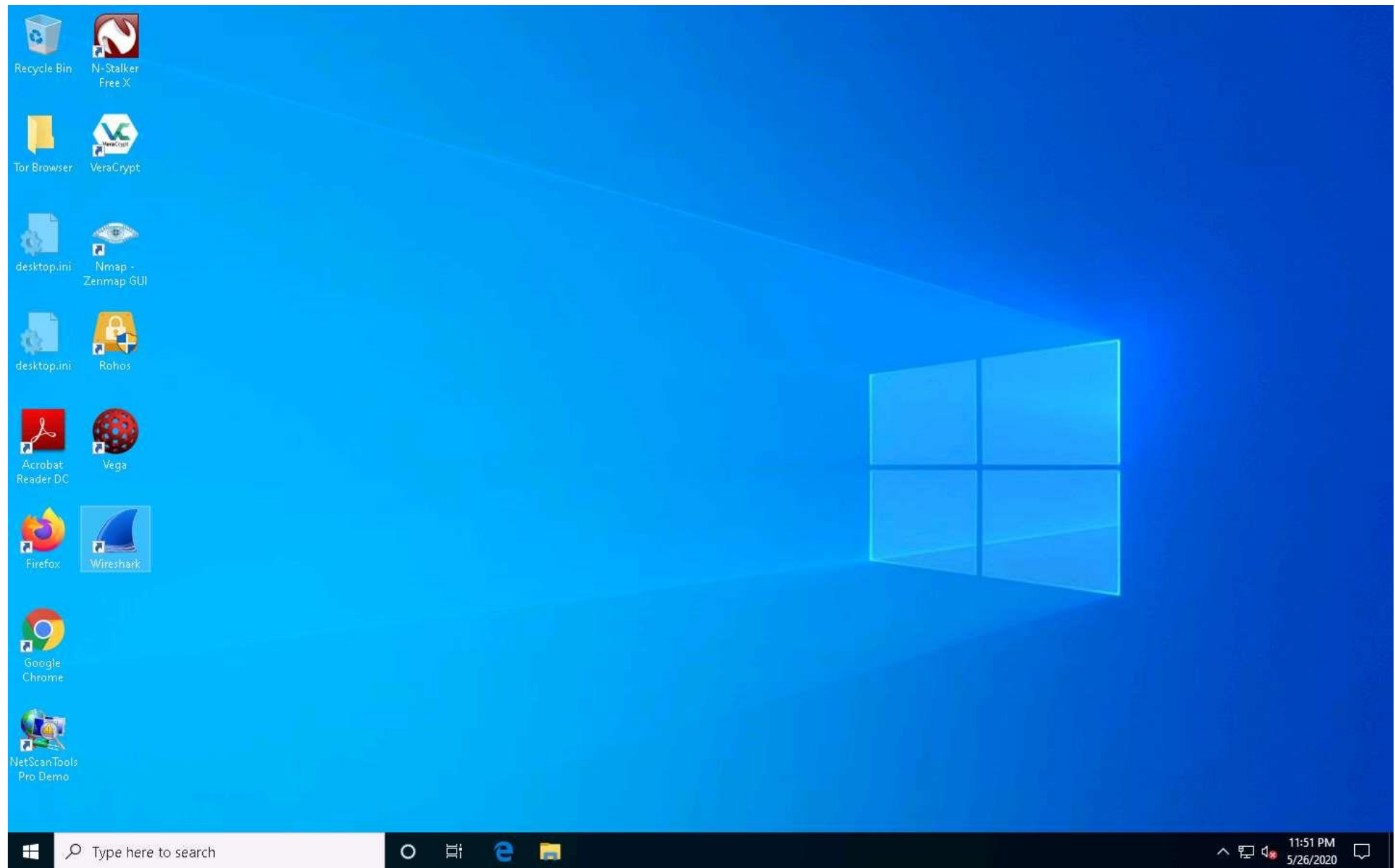
Operating System (OS)	Time To Live	TCP Window Size
FreeBSD	64	65535
OpenBSD	64	16384
Windows 95	32	8192
Windows 2000	128	16384
Windows XP	128	65535
Windows 98, Vista and 7 (Server 2008)	128	8192
iOS 12.4 (Cisco Routers)	255	4128
Solaris 7	255	8760
AIX 4.3	64	16384

Task 1: Identify the Target System's OS with Time-to-Live (TTL) and TCP Window Sizes using Wireshark

Wireshark is a network protocol analyzer that allows capturing and interactively browsing the traffic running on a computer network. It is used to identify the target OS through sniffing/capturing the response generated from the target machine to the request-originated machine. Further, you can observe the TTL and TCP window size fields in the captured TCP packet. Using these values, the target OS can be determined.

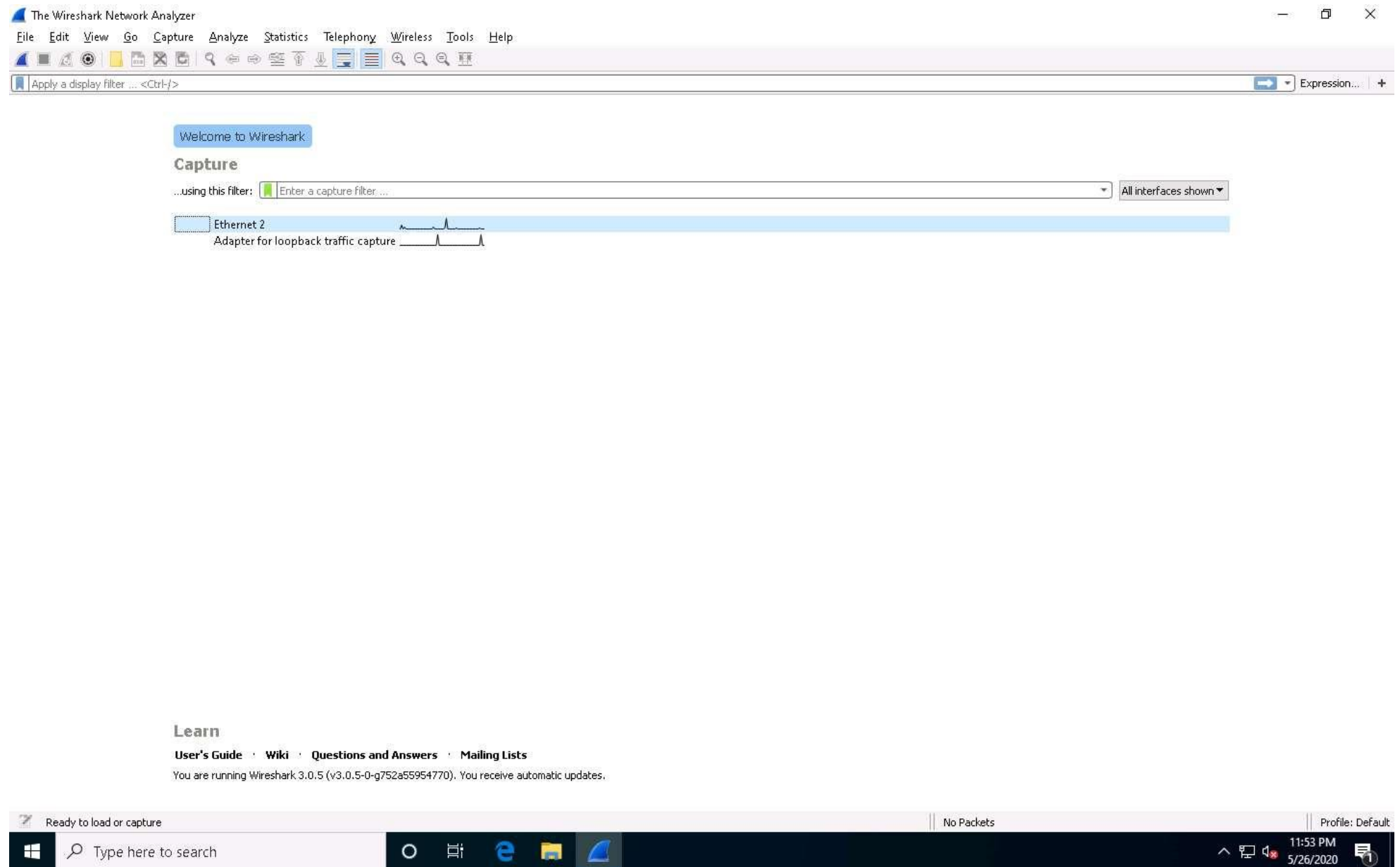
Here, we will use the Wireshark tool to perform OS discovery on the target host(s).

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



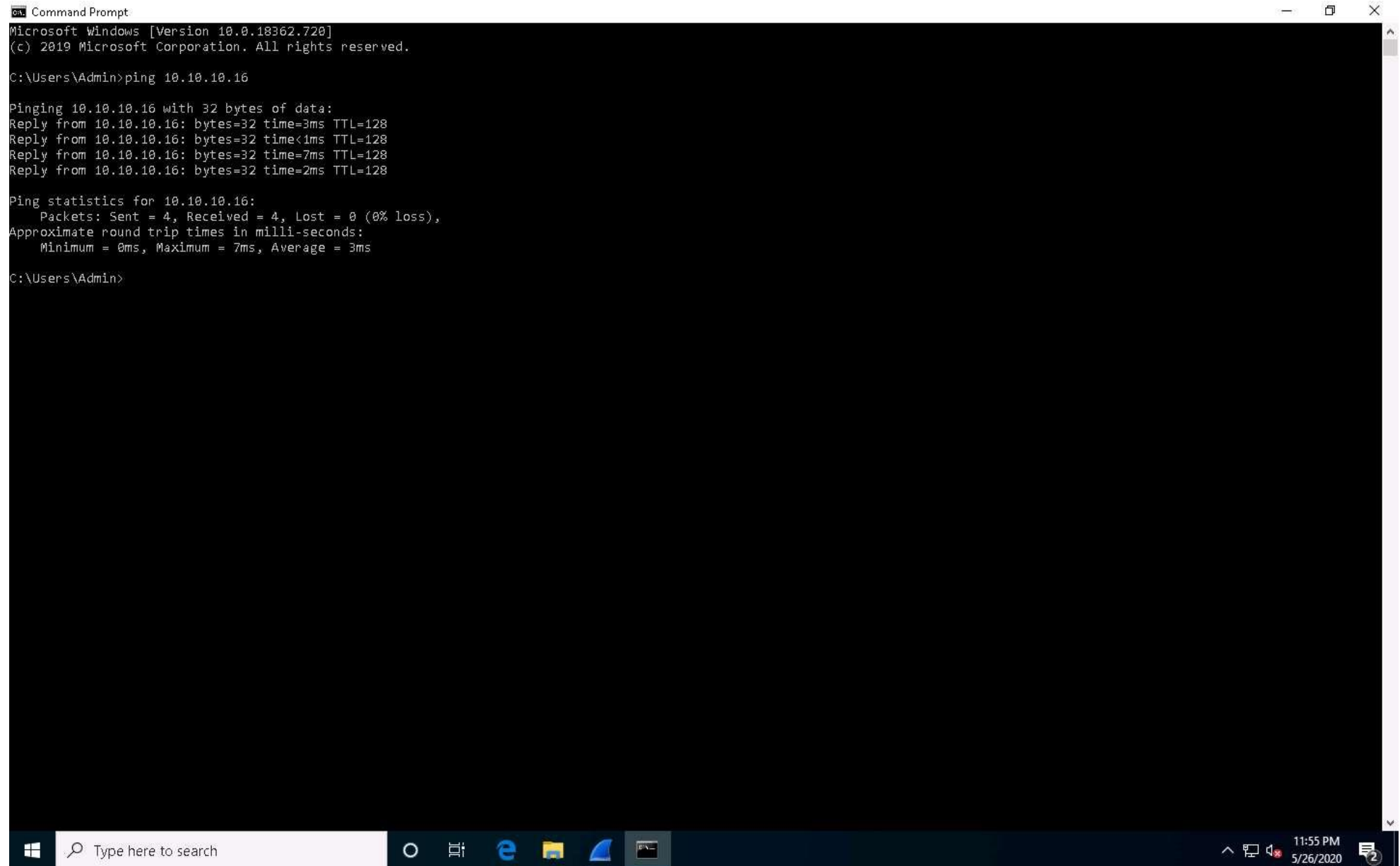
3. ☐ The **Wireshark Network Analyzer** main window appears; double-click the available ethernet or interface (here, **Ethernet2**) to start the packet capture, as shown in the screenshot.

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



4. ☐ Open the **Command Prompt**, type **ping 10.10.10.16** and press **Enter**.

10.10.10.16 is the IP address of the **Windows Server 2016** machine.



The screenshot shows a Windows Command Prompt window titled "Command Prompt". The text inside the window is as follows:

```
Microsoft Windows [Version 10.0.18362.720]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Admin>ping 10.10.10.16

Pinging 10.10.10.16 with 32 bytes of data:
Reply from 10.10.10.16: bytes=32 time=3ms TTL=128
Reply from 10.10.10.16: bytes=32 time<1ms TTL=128
Reply from 10.10.10.16: bytes=32 time=7ms TTL=128
Reply from 10.10.10.16: bytes=32 time=2ms TTL=128

Ping statistics for 10.10.10.16:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 7ms, Average = 3ms

C:\Users\Admin>
```

The window is part of a Windows 10 desktop environment. The taskbar at the bottom shows the Start button, a search bar with the text "Type here to search", and several pinned application icons including the Task View button, File Explorer, Microsoft Edge, and the Command Prompt. The system tray on the right shows the date and time as "11:55 PM 5/26/2020" and a notification icon with a number "2".

5. ☐ Observe the packets captured by **Wireshark**.

Capturing from Ethernet 2

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-F> Expression...

No.	Time	Source	Destination	Protocol	Length	Info
93	33.864832	10.10.10.10	40.79.70.158	TCP	54	50496 → 443 [ACK] Seq=212 Ack=2369 Win=263424 Len=0
94	33.866510	10.10.10.10	40.79.70.158	TLSv1.2	147	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
95	33.945907	40.79.70.158	10.10.10.10	TLSv1.2	105	Change Cipher Spec, Encrypted Handshake Message
96	33.945909	40.79.70.158	10.10.10.10	TLSv1.2	123	Application Data
97	33.945998	10.10.10.10	40.79.70.158	TCP	54	50496 → 443 [ACK] Seq=305 Ack=2489 Win=263168 Len=0
98	33.971931	10.10.10.10	40.79.70.158	TLSv1.2	141	Application Data
99	33.971995	10.10.10.10	40.79.70.158	TLSv1.2	233	Application Data
100	33.972066	10.10.10.10	40.79.70.158	TLSv1.2	92	Application Data
101	34.052301	40.79.70.158	10.10.10.10	TCP	54	443 → 50496 [ACK] Seq=2489 Ack=571 Win=262400 Len=0
102	34.052302	40.79.70.158	10.10.10.10	TLSv1.2	92	Application Data
103	34.052454	40.79.70.158	10.10.10.10	TLSv1.2	615	Application Data
104	34.052492	10.10.10.10	40.79.70.158	TCP	54	50496 → 443 [ACK] Seq=609 Ack=3088 Win=262656 Len=0
105	34.053309	10.10.10.10	40.79.70.158	TCP	54	50496 → 443 [FIN, ACK] Seq=609 Ack=3088 Win=262656 Len=0
106	34.128248	40.79.70.158	10.10.10.10	TCP	54	443 → 50496 [FIN, ACK] Seq=3088 Ack=610 Win=262400 Len=0
107	34.128328	10.10.10.10	40.79.70.158	TCP	54	50496 → 443 [ACK] Seq=610 Ack=3089 Win=262656 Len=0
108	34.504714	fe80::1:1	ff02::1:6	ICMPv6	90	Multicast Listener Report Message v2
109	34.968074	117.18.237.29	10.10.10.10	TCP	54	80 → 50489 [ACK] Seq=1 Ack=1 Win=131 Len=0
110	34.968132	10.10.10.10	117.18.237.29	TCP	54	[TCP ACKed unseen segment] 50489 → 80 [ACK] Seq=1 Ack=2 Win=1023 Len=0
111	40.703926	fe80::1:1	ff02::1	ICMPv6	110	Router Advertisement from 00:15:5d:32:af:f2
112	40.717428	fe80::215:5dff:fe32::	ff02::1:6	ICMPv6	110	Multicast Listener Report Message v2
113	40.717477	fe80::a915:7ee0:da2::	ff02::1:6	ICMPv6	90	Multicast Listener Report Message v2
114	41.394190	fe80::215:5dff:fe32::	ff02::1:6	ICMPv6	110	Multicast Listener Report Message v2
115	41.612631	fe80::a915:7ee0:da2::	ff02::1:6	ICMPv6	90	Multicast Listener Report Message v2
116	45.803898	fe80::1:1	ff02::1	ICMPv6	110	Router Advertisement from 00:15:5d:32:af:f2
117	45.815782	fe80::215:5dff:fe32::	ff02::1:6	ICMPv6	110	Multicast Listener Report Message v2
118	45.820525	fe80::a915:7ee0:da2::	ff02::1:6	ICMPv6	90	Multicast Listener Report Message v2
119	45.903733	fe80::1:1	ff02::1:6	ICMPv6	90	Multicast Listener Report Message v2
120	46.221026	fe80::a915:7ee0:da2::	ff02::1:6	ICMPv6	90	Multicast Listener Report Message v2
121	46.546413	fe80::215:5dff:fe32::	ff02::1:6	ICMPv6	110	Multicast Listener Report Message v2
122	46.703828	fe80::1:1	ff02::1:6	ICMPv6	90	Multicast Listener Report Message v2

> Frame 1: 591 bytes on wire (4728 bits), 591 bytes captured (4728 bits) on interface 0

> Ethernet II, Src: Microsof_32:af:f4 (00:15:5d:32:af:f4), Dst: Microsof_32:af:f2 (00:15:5d:32:af:f2)

> Internet Protocol Version 4, Src: 10.10.10.10, Dst: 204.79.197.200

> Transmission Control Protocol, Src Port: 50482, Dst Port: 443, Seq: 1, Ack: 1, Len: 537

```

0000  00 15 5d 32 af f2 00 15 5d 32 af f4 08 00 45 00  ..]2....]2....E.
0010  02 41 b3 35 40 00 80 06 00 00 0a 0a 0a cc 4f    .A.5@.....O
0020  c5 c8 c5 32 01 bb 2f 16 89 0b 18 c3 d5 c5 50 18  ..2../. ....P.
0030  03 fe a8 5f 00 00 17 03 03 02 14 00 00 00 00 00  ...L....n....
0040  00 00 22 4c f0 f4 9d e0 97 e1 04 6e 15 85 fd f3  ...x....9...i
0050  e6 83 d6 76 9f 91 f4 0a 94 0a 39 08 cf b2 69 e6  ...v....@...R.
0060  fe 28 3c 89 58 b7 6c 3b 22 45 15 a6 2b 8d 52 93  -(<X-1; "E--+R-
0070  89 12 af aa 40 95 fc 0a 21 2a 2b db 92 ce 16 27  ...@...!+....'

```

Ethernet 2: <live capture in progress> | Packets: 122 · Displayed: 122 (100.0%) | Profile: Default

6. ☐ Choose any packet of the ICMP reply from the **Windows Server 2016 (10.10.10.16)** to **Windows 10 (10.10.10.10)** machines and expand the **Internet Protocol Version 4** node in the **Packet Details** pane.

7. ☐ The TTL value is recorded as **128**, which means that the ICMP reply possibly came from a Windows-based machine.

Capturing from Ethernet 2

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

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

No.	Time	Source	Destination	Protocol	Length	Info
19	9.244373	fe80::215:5dff:fe32_	ff02::16	ICMPv6	110	Multicast Listener Report Message v2
20	9.244401	fe80::a915:7ee0:da2_	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
21	9.247448	10.10.10.10	204.79.197.200	TLSv1.2	926	Application Data
22	9.247806	10.10.10.10	204.79.197.200	TLSv1.2	776	Application Data
23	9.247905	10.10.10.10	204.79.197.200	TLSv1.2	92	Application Data
24	9.250548	204.79.197.200	10.10.10.10	TCP	54	443 → 50482 [ACK] Seq=143 Ack=5600 Win=8209 Len=0
25	9.250549	204.79.197.200	10.10.10.10	TCP	54	443 → 50482 [ACK] Seq=143 Ack=6360 Win=8212 Len=0
26	9.261528	204.79.197.200	10.10.10.10	TLSv1.2	197	Application Data
27	9.261572	10.10.10.10	204.79.197.200	TCP	54	50482 → 443 [ACK] Seq=6360 Ack=286 Win=1021 Len=0
28	9.932735	fe80::a915:7ee0:da2_	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
29	10.096388	fe80::215:5dff:fe32_	ff02::16	ICMPv6	110	Multicast Listener Report Message v2
30	11.459219	10.10.10.10	10.10.10.16	ICMP	74	Echo (ping) request id=0x0001, seq=5/1280, ttl=128 (reply in 31)
31	11.462262	10.10.10.16	10.10.10.10	ICMP	74	Echo (ping) reply id=0x0001, seq=5/1280, ttl=128 (request in 30)
32	12.473513	10.10.10.10	10.10.10.16	ICMP	74	Echo (ping) request id=0x0001, seq=6/1536, ttl=128 (reply in 33)
33	12.474310	10.10.10.16	10.10.10.10	ICMP	74	Echo (ping) reply id=0x0001, seq=6/1536, ttl=128 (request in 32)
34	13.488855	10.10.10.10	10.10.10.16	ICMP	74	Echo (ping) request id=0x0001, seq=7/1792, ttl=128 (reply in 35)
35	13.495991	10.10.10.16	10.10.10.10	ICMP	74	Echo (ping) reply id=0x0001, seq=7/1792, ttl=128 (request in 34)
36	14.504923	10.10.10.10	10.10.10.16	ICMP	74	Echo (ping) request id=0x0001, seq=8/2048, ttl=128 (reply in 37)

> Frame 31: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0

> Ethernet II, Src: Microsof_32:af:f6 (00:15:5d:32:af:f6), Dst: Microsof_32:af:f4 (00:15:5d:32:af:f4)

✓ Internet Protocol Version 4, Src: 10.10.10.16, Dst: 10.10.10.10

0100 = Version: 4

.... 0101 = Header Length: 20 bytes (5)

> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)

Total Length: 60

Identification: 0x5061 (20577)

> Flags: 0x0000

Time to live: 128

Protocol: ICMP (1)

Header checksum: 0xc232 [validation disabled]

[Header checksum status: Unverified]

Source: 10.10.10.16

Destination: 10.10.10.10

> Internet Control Message Protocol

0000 00 15 5d 32 af f4 00 15 5d 32 af f6 08 00 45 00 ..]2...[2...E-

0010 00 3c 50 61 00 00 80 01 c2 32 0a 0a 0a 10 0a 0a <Pa....2.....

0020 0a 0a 00 00 55 56 00 01 00 05 61 62 63 64 65 66UV...abcdef

0030 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 ghijklmn opqrstuv

0040 77 61 62 63 64 65 66 67 68 69 wabcdefg hi

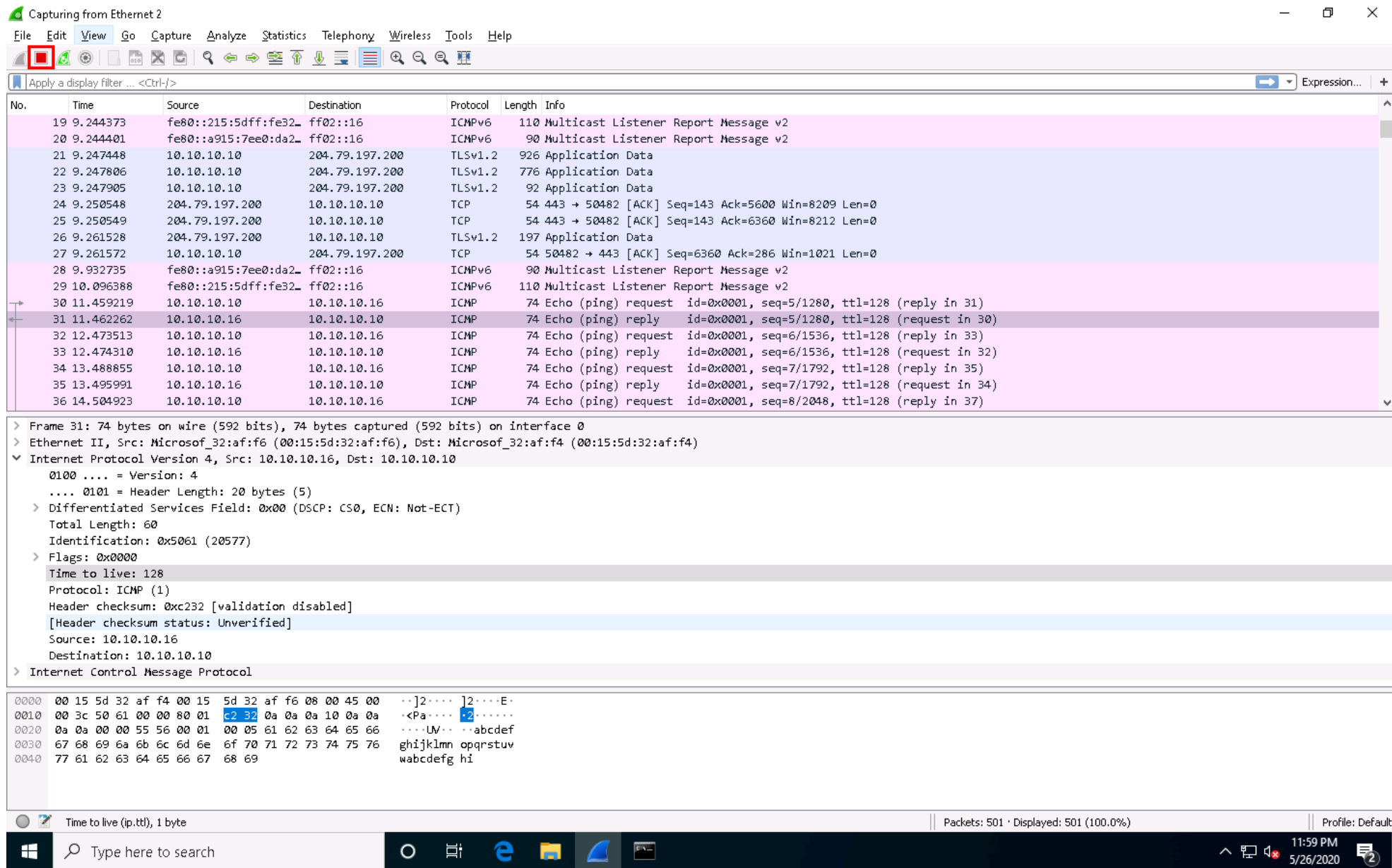
Time to live (ip.ttl), 1 byte

Packets: 401 · Displayed: 401 (100.0%)

Profile: Default

11:58 PM 5/26/2020

8.  Now, stop the capture in the **Wireshark** window by clicking on the **Stop** button from the toolbar.



Capturing from 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
19	9.244373	fe80::215:5dff:fe32_	ff02::16	ICMPv6	110	Multicast Listener Report Message v2
20	9.244401	fe80::a915:7ee0:da2_	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
21	9.247448	10.10.10.10	204.79.197.200	TLSv1.2	926	Application Data
22	9.247806	10.10.10.10	204.79.197.200	TLSv1.2	776	Application Data
23	9.247905	10.10.10.10	204.79.197.200	TLSv1.2	92	Application Data
24	9.250548	204.79.197.200	10.10.10.10	TCP	54	443 → 50482 [ACK] Seq=143 Ack=5600 Win=8209 Len=0
25	9.250549	204.79.197.200	10.10.10.10	TCP	54	443 → 50482 [ACK] Seq=143 Ack=6360 Win=8212 Len=0
26	9.261528	204.79.197.200	10.10.10.10	TLSv1.2	197	Application Data
27	9.261572	10.10.10.10	204.79.197.200	TCP	54	50482 → 443 [ACK] Seq=6360 Ack=286 Win=1021 Len=0
28	9.932735	fe80::a915:7ee0:da2_	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
29	10.096388	fe80::215:5dff:fe32_	ff02::16	ICMPv6	110	Multicast Listener Report Message v2
30	11.459219	10.10.10.10	10.10.10.16	ICMP	74	Echo (ping) request id=0x0001, seq=5/1280, ttl=128 (reply in 31)
31	11.462262	10.10.10.16	10.10.10.10	ICMP	74	Echo (ping) reply id=0x0001, seq=5/1280, ttl=128 (request in 30)
32	12.473513	10.10.10.10	10.10.10.16	ICMP	74	Echo (ping) request id=0x0001, seq=6/1536, ttl=128 (reply in 33)
33	12.474310	10.10.10.16	10.10.10.10	ICMP	74	Echo (ping) reply id=0x0001, seq=6/1536, ttl=128 (request in 32)
34	13.488855	10.10.10.10	10.10.10.16	ICMP	74	Echo (ping) request id=0x0001, seq=7/1792, ttl=128 (reply in 35)
35	13.495991	10.10.10.16	10.10.10.10	ICMP	74	Echo (ping) reply id=0x0001, seq=7/1792, ttl=128 (request in 34)
36	14.504923	10.10.10.10	10.10.10.16	ICMP	74	Echo (ping) request id=0x0001, seq=8/2048, ttl=128 (reply in 37)

> Frame 31: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0

> Ethernet II, Src: Microsof_32:af:f6 (00:15:5d:32:af:f6), Dst: Microsof_32:af:f4 (00:15:5d:32:af:f4)

✓ Internet Protocol Version 4, Src: 10.10.10.16, Dst: 10.10.10.10

0100 = Version: 4

.... 0101 = Header Length: 20 bytes (5)

> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)

Total Length: 60

Identification: 0x5061 (20577)

> Flags: 0x0000

Time to live: 128

Protocol: ICMP (1)

Header checksum: 0xc232 [validation disabled]

[Header checksum status: Unverified]

Source: 10.10.10.16

Destination: 10.10.10.10

> Internet Control Message Protocol

0000 00 15 5d 32 af f4 00 15 5d 32 af f6 08 00 45 00 ..j2.... }2....E

0010 00 3c 50 61 00 00 80 01 c2 32 0a 0a 0a 10 0a 0a <Pa.... 2.....

0020 0a 0a 00 00 55 56 00 01 00 05 61 62 63 64 65 66UV... ..abcdef

0030 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 ghijklmn opqrstuv

0040 77 61 62 63 64 65 66 67 68 69 wabcdefg hi

Time to live (ip.ttl), 1 byte

Packets: 501 · Displayed: 501 (100.0%)

Profile: Default

11:59 PM 5/26/2020

9. ☐ Now, click the **Start capturing packets** button from the toolbar. If an **Unsaved packets...** pop-up appears, click **Continue without Saving**.

The image shows the Wireshark network protocol analyzer interface. The top menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, and Help. Below the menu is a toolbar with various icons for file operations, capture control, and analysis. A display filter bar shows 'Apply a display filter ... <Ctrl-F>'. The main packet list pane displays a table of captured packets with columns for No., Time, Source, Destination, Protocol, Length, and Info. The selected packet (No. 31) is an ICMP Echo (ping) request from 10.10.10.16 to 10.10.10.10. The packet details pane below shows the structure of the selected packet, including Ethernet II, Internet Protocol Version 4, and Internet Control Message Protocol (ICMP). The packet bytes pane at the bottom shows the raw data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
19	9.244373	fe80::215:5dff:fe32...	ff02::16	ICMPv6	110	Multicast Listener Report Message v2
20	9.244401	fe80::a915:7ee0:da2...	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
21	9.247448	10.10.10.10	204.79.197.200	TLSv1.2	926	Application Data
22	9.247806	10.10.10.10	204.79.197.200	TLSv1.2	776	Application Data
23	9.247905	10.10.10.10	204.79.197.200	TLSv1.2	92	Application Data
24	9.250548	204.79.197.200	10.10.10.10	TCP	54	443 → 50482 [ACK] Seq=143 Ack=5600 Win=8209 Len=0
25	9.250549	204.79.197.200	10.10.10.10	TCP	54	443 → 50482 [ACK] Seq=143 Ack=6360 Win=8212 Len=0
26	9.261528	204.79.197.200	10.10.10.10	TLSv1.2	197	Application Data
27	9.261572	10.10.10.10	204.79.197.200	TCP	54	50482 → 443 [ACK] Seq=6360 Ack=286 Win=1021 Len=0
28	9.932735	fe80::a915:7ee0:da2...	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
29	10.096388	fe80::215:5dff:fe32...	ff02::16	ICMPv6	110	Multicast Listener Report Message v2
30	11.459219	10.10.10.10	10.10.10.16	ICMP	74	Echo (ping) request id=0x0001, seq=5/1280, ttl=128 (reply in 31)
31	11.462262	10.10.10.16	10.10.10.10	ICMP	74	Echo (ping) reply id=0x0001, seq=5/1280, ttl=128 (request in 30)
32	12.473513	10.10.10.10	10.10.10.16	ICMP	74	Echo (ping) request id=0x0001, seq=6/1536, ttl=128 (reply in 33)
33	12.474310	10.10.10.16	10.10.10.10	ICMP	74	Echo (ping) reply id=0x0001, seq=6/1536, ttl=128 (request in 32)
34	13.488855	10.10.10.10	10.10.10.16	ICMP	74	Echo (ping) request id=0x0001, seq=7/1792, ttl=128 (reply in 35)
35	13.495991	10.10.10.16	10.10.10.10	ICMP	74	Echo (ping) reply id=0x0001, seq=7/1792, ttl=128 (request in 34)
36	14.504923	10.10.10.10	10.10.10.16	ICMP	74	Echo (ping) request id=0x0001, seq=8/2048, ttl=128 (reply in 37)

> Frame 31: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
> Ethernet II, Src: Microsof_32:af:f6 (00:15:5d:32:af:f6), Dst: Microsof_32:af:f4 (00:15:5d:32:af:f4)
v Internet Protocol Version 4, Src: 10.10.10.16, Dst: 10.10.10.10
0100 = Version: 4
.... 0101 = Header Length: 20 bytes (5)
> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
Total Length: 60
Identification: 0x5061 (20577)
> Flags: 0x0000
Time to live: 128
Protocol: ICMP (1)
Header checksum: 0xc232 [validation disabled]
[Header checksum status: Unverified]
Source: 10.10.10.16
Destination: 10.10.10.10
> Internet Control Message Protocol

0000 00 15 5d 32 af f4 00 15 5d 32 af f6 08 00 45 00 ..]2....]2....E-
0010 00 3c 50 61 00 00 80 01 c2 32 0a 0a 0a 10 0a 0a <Pa.... -2.....
0020 0a 0a 00 00 55 56 00 01 00 05 61 62 63 64 65 66UV... ..abcdef
0030 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 ghijklmn opqrstuv
0040 77 61 62 63 64 65 66 67 68 69 wabcdefg hi

wireshark_Ethernet 2_20200526235535_a08028.pcapng | Packets: 629 · Displayed: 629 (100.0%) | Profile: Default

12:01 AM 5/27/2020

10. ☐ Wireshark will start capturing the new packets.
11. ☐ In the **Command Prompt** window, type **ping 10.10.10.9** and press **Enter**.

10.10.10.9 is the IP address of the **Ubuntu** machine.

```
Command Prompt
Microsoft Windows [Version 10.0.18362.720]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Admin>ping 10.10.10.16

Pinging 10.10.10.16 with 32 bytes of data:
Reply from 10.10.10.16: bytes=32 time=3ms TTL=128
Reply from 10.10.10.16: bytes=32 time<1ms TTL=128
Reply from 10.10.10.16: bytes=32 time=7ms TTL=128
Reply from 10.10.10.16: bytes=32 time=2ms TTL=128

Ping statistics for 10.10.10.16:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 7ms, Average = 3ms

C:\Users\Admin>ping 10.10.10.9

Pinging 10.10.10.9 with 32 bytes of data:
Reply from 10.10.10.9: bytes=32 time=1ms TTL=64
Reply from 10.10.10.9: bytes=32 time=3ms TTL=64
Reply from 10.10.10.9: bytes=32 time<1ms TTL=64
Reply from 10.10.10.9: bytes=32 time=2ms TTL=64

Ping statistics for 10.10.10.9:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 3ms, Average = 1ms

C:\Users\Admin>
```

12. ☐ Observe the packets captured by **Wireshark**.

13. ☐ Choose any packet of ICMP reply from the **Ubuntu (10.10.10.9)** to **Windows 10 (10.10.10.10)** machine and expand the **Internet Protocol Version 4** node in the **Packet Details** pane.
14. ☐ The TTL value is recorded as **64**, which means the ICMP reply possibly came from a Linux-based machine.

Capturing from Ethernet 2

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-F> Expression...

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	fe80::77f8:3a62:970...	ff02::fb	MDNS	439	Standard query response 0x0000 TXT, cache flush PTR _adb._tcp.local PTR adb-unidentified._adb._tcp.local SRV, cache flush...
2	0.000116	fe80::215:5dff:fe32...	ff02::fb	MDNS	371	Standard query response 0x0000 TXT, cache flush PTR _adb._tcp.local PTR adb-unidentified._adb._tcp.local SRV, cache flush...
3	0.000163	10.10.10.14	224.0.0.251	MDNS	419	Standard query response 0x0000 TXT, cache flush PTR _adb._tcp.local PTR adb-unidentified._adb._tcp.local SRV, cache flush...
4	0.226655	fe80::1:1	ff02::1	ICMPv6	110	Router Advertisement from 00:15:5d:32:af:f2
5	0.236511	fe80::a915:7ee0:da2...	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
6	0.239409	fe80::215:5dff:fe32...	ff02::16	ICMPv6	110	Multicast Listener Report Message v2
7	0.355294	fe80::1:1	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
8	0.557243	fe80::a915:7ee0:da2...	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
9	0.779346	fe80::215:5dff:fe32...	ff02::16	ICMPv6	110	Multicast Listener Report Message v2
10	2.155515	fe80::1:1	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
11	3.993958	Microsof_32:af:f4	Broadcast	ARP	42	Who has 10.10.10.9? Tell 10.10.10.10
12	3.994963	Microsof_32:af:f8	Microsof_32:af:f4	ARP	42	10.10.10.9 is at 00:15:5d:32:af:f8
13	3.994992	10.10.10.10	10.10.10.9	ICMP	74	Echo (ping) request id=0x0001, seq=9/2304, ttl=128 (reply in 14)
14	3.995764	10.10.10.9	10.10.10.10	ICMP	74	Echo (ping) reply id=0x0001, seq=9/2304, ttl=64 (request in 13)
15	5.004919	10.10.10.10	10.10.10.9	ICMP	74	Echo (ping) request id=0x0001, seq=10/2560, ttl=128 (reply in 16)
16	5.008230	10.10.10.9	10.10.10.10	ICMP	74	Echo (ping) reply id=0x0001, seq=10/2560, ttl=64 (request in 15)
17	6.020510	10.10.10.10	10.10.10.9	ICMP	74	Echo (ping) request id=0x0001, seq=11/2816, ttl=128 (reply in 18)
18	6.021226	10.10.10.9	10.10.10.10	ICMP	74	Echo (ping) reply id=0x0001, seq=11/2816, ttl=64 (request in 17)

> Frame 14: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0

> Ethernet II, Src: Microsof_32:af:f8 (00:15:5d:32:af:f8), Dst: Microsof_32:af:f4 (00:15:5d:32:af:f4)

> Internet Protocol Version 4, Src: 10.10.10.9, Dst: 10.10.10.10

0100 = Version: 4

.... 0101 = Header Length: 20 bytes (5)

> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)

Total Length: 60

Identification: 0xe4eb (58603)

> Flags: 0x0000

Time to live: 64

Protocol: ICMP (1)

Header checksum: 0x6daf [validation disabled]

[Header checksum status: Unverified]

Source: 10.10.10.9

Destination: 10.10.10.10

> Internet Control Message Protocol

```

0000  00 15 5d 32 af f4 00 15 5d 32 af f8 08 00 45 00  ..]2....]2....E.
0010  00 3c e4 eb 00 00 01 6d af 0a 0a 0a 09 0a 0a  ..<....m.....
0020  0a 0a 00 00 55 52 00 01 00 09 61 62 63 64 65 66  ....UR...abcdef
0030  67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76  ghijklmn opqrstuv
0040  77 61 62 63 64 65 66 67 68 69  wabcdefg hi
  
```

Time to live (ip.ttl), 1 byte

Packets: 1560 · Displayed: 1560 (100.0%)

Profile: Default

15. ☐ Stop the capture in the **Wireshark** window by clicking on the Stop button.
16. ☐ This concludes the demonstration of identifying the OS of the target system using Wireshark.

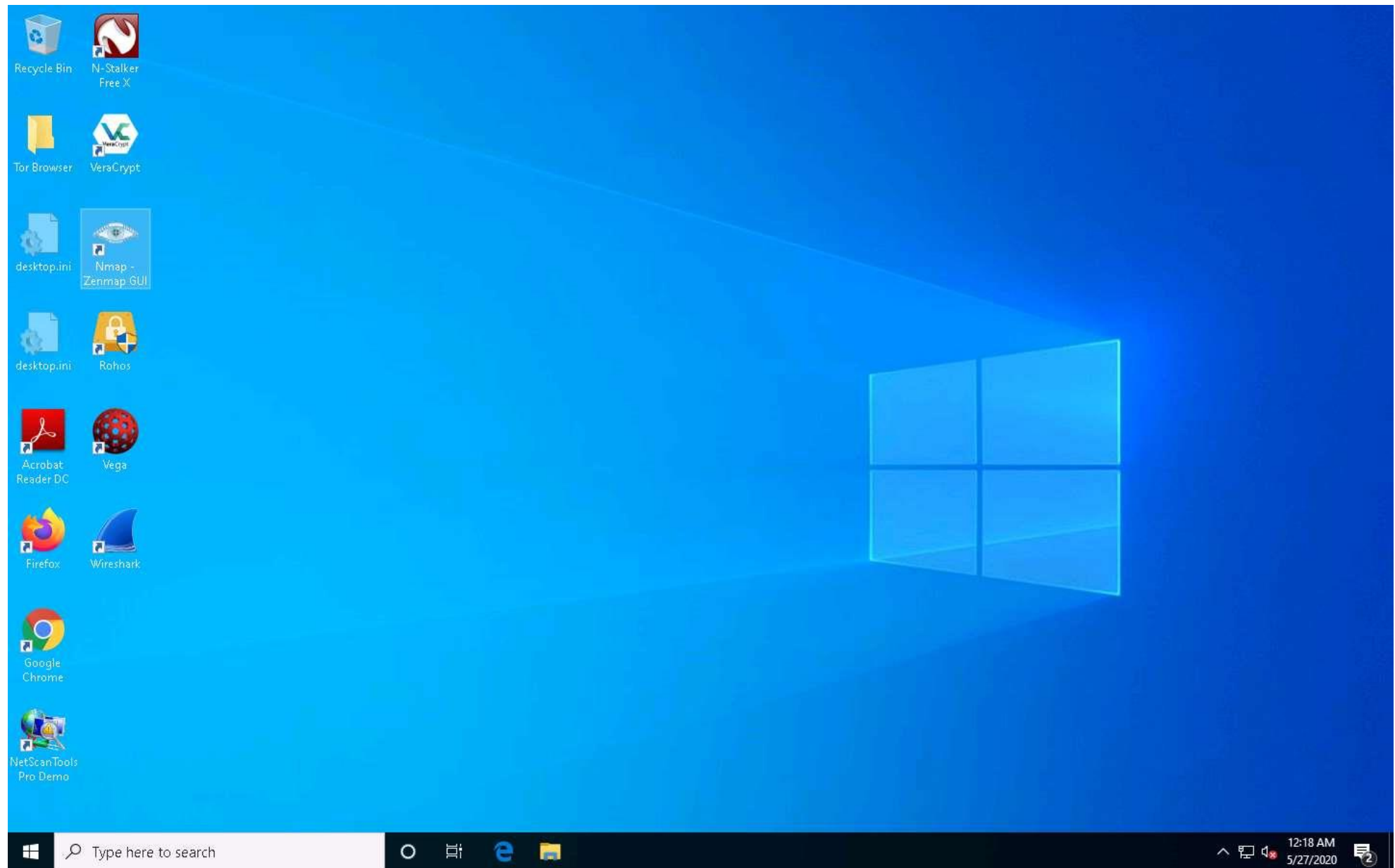
17. ☐ Close all open windows and document all the acquired information.
-

Task 2: Perform OS Discovery using Nmap Script Engine (NSE)

Nmap, along with Nmap Script Engine (NSE), can extract considerable valuable information from the target system. In addition to Nmap commands, NSE provides scripts that reveal all sorts of useful information from the target system. Using NSE, you may obtain information such as OS, computer name, domain name, forest name, NetBIOS computer name, NetBIOS domain name, workgroup, system time of a target system, etc.

Here, we will use Nmap to perform OS discovery using -A parameter, -O parameter, and NSE.

1. ☐ In the **Windows 10**, navigate to the Desktop and double-click **Nmap - Zenmap GUI** shortcut.

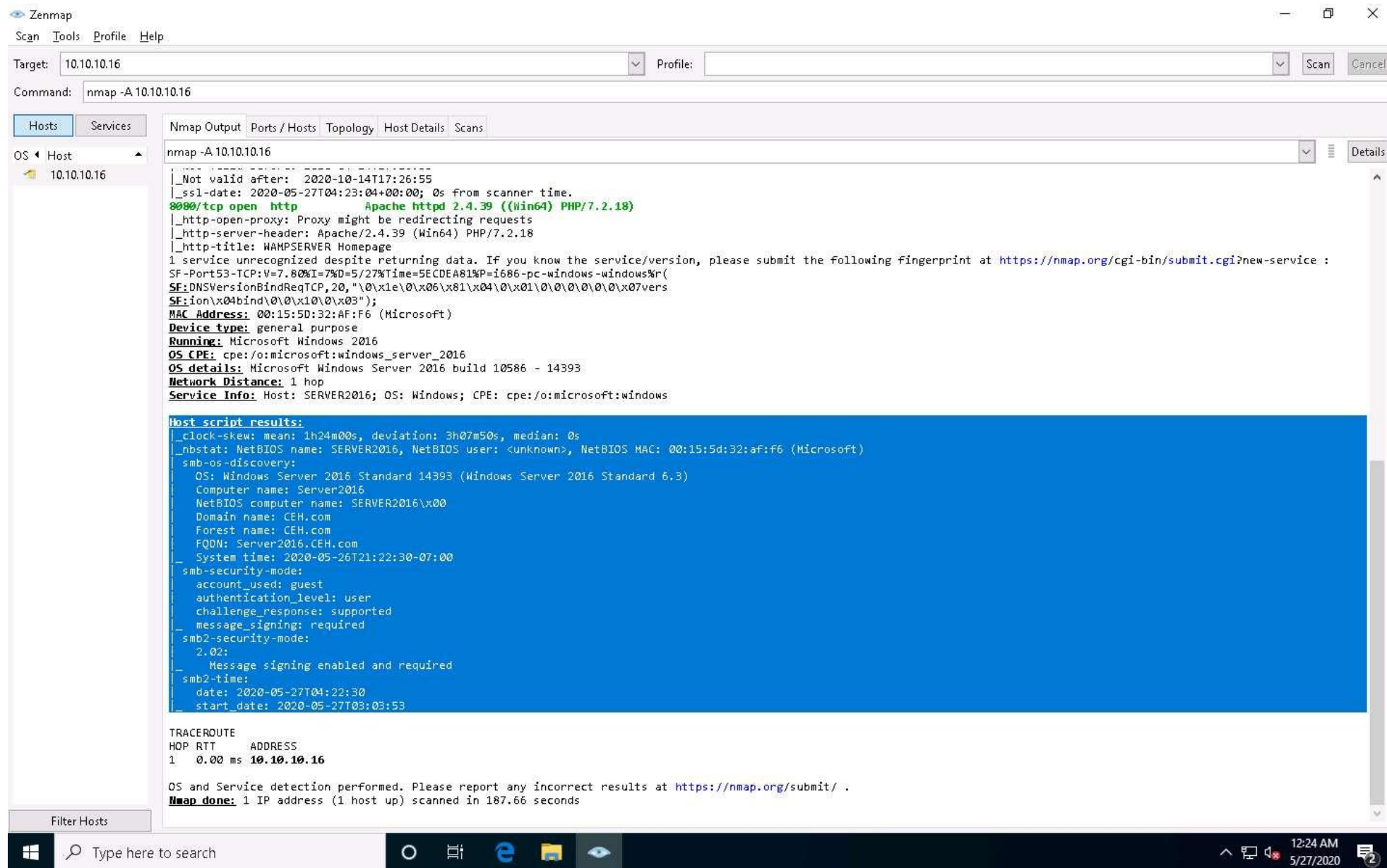


2. ☐ The **Zenmap GUI** appears. In the **Command** field, type the command **nmap -A [Target IP Address]** (here, the target machine is **Windows Server 2016 [10.10.10.16]**) and click **Scan**.

-A: to perform an aggressive scan.

The scan takes approximately 10 minutes to complete.

3. ☐ The scan results appear, displaying the open ports and running services along with their versions and target details such as OS, computer name, NetBIOS computer name, etc. under the **Host script results** section.



4. ☐ In the **Command** field, type the command **nmap -O [Target IP Address]** (here, the target machine is **Windows Server 2016 [10.10.10.16]**) and click **Scan**.

-O: performs the OS discovery.

5. ☐ The scan results appear, displaying information about open ports, respective services running on the open ports, and the name of the OS running on the target system.

Zenmap

Scan Tools Profile Help

Target: 10.10.10.16 Profile: Scan Cancel

Command: nmap -O 10.10.10.16

Hosts Services Nmap Output Ports / Hosts Topology Host Details Scans

OS Host

10.10.10.16

nmap -O 10.10.10.16

Starting Nmap 7.80 (<https://nmap.org>) at 2020-05-27 00:25 Eastern Daylight Time
Nmap scan report for 10.10.10.16
Host is up (0.00s latency).
Not shown: 981 closed 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	kpassud5
593/tcp	open	http-rpc-epmap
636/tcp	open	ldaps1
1060/tcp	open	polesar
1801/tcp	open	msmq
2103/tcp	open	zephyr-clt
2105/tcp	open	eklogin
2107/tcp	open	msmq-ngnt
3268/tcp	open	globalcatLDAP
3269/tcp	open	globalcatLDAPssl
3389/tcp	open	ms-wbt-server
8080/tcp	open	http-proxy

MAC Address: 00:15:5D:32:AF:F6 (Microsoft)
Device type: general purpose
Running: Microsoft Windows 2016
OS CPE: cpe:/o:microsoft:windows_server_2016
OS details: Microsoft Windows Server 2016 build 10586 - 14393
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at <https://nmap.org/submit/> .
Nmap done: 1 IP address (1 host up) scanned in 3.83 seconds

Filter Hosts

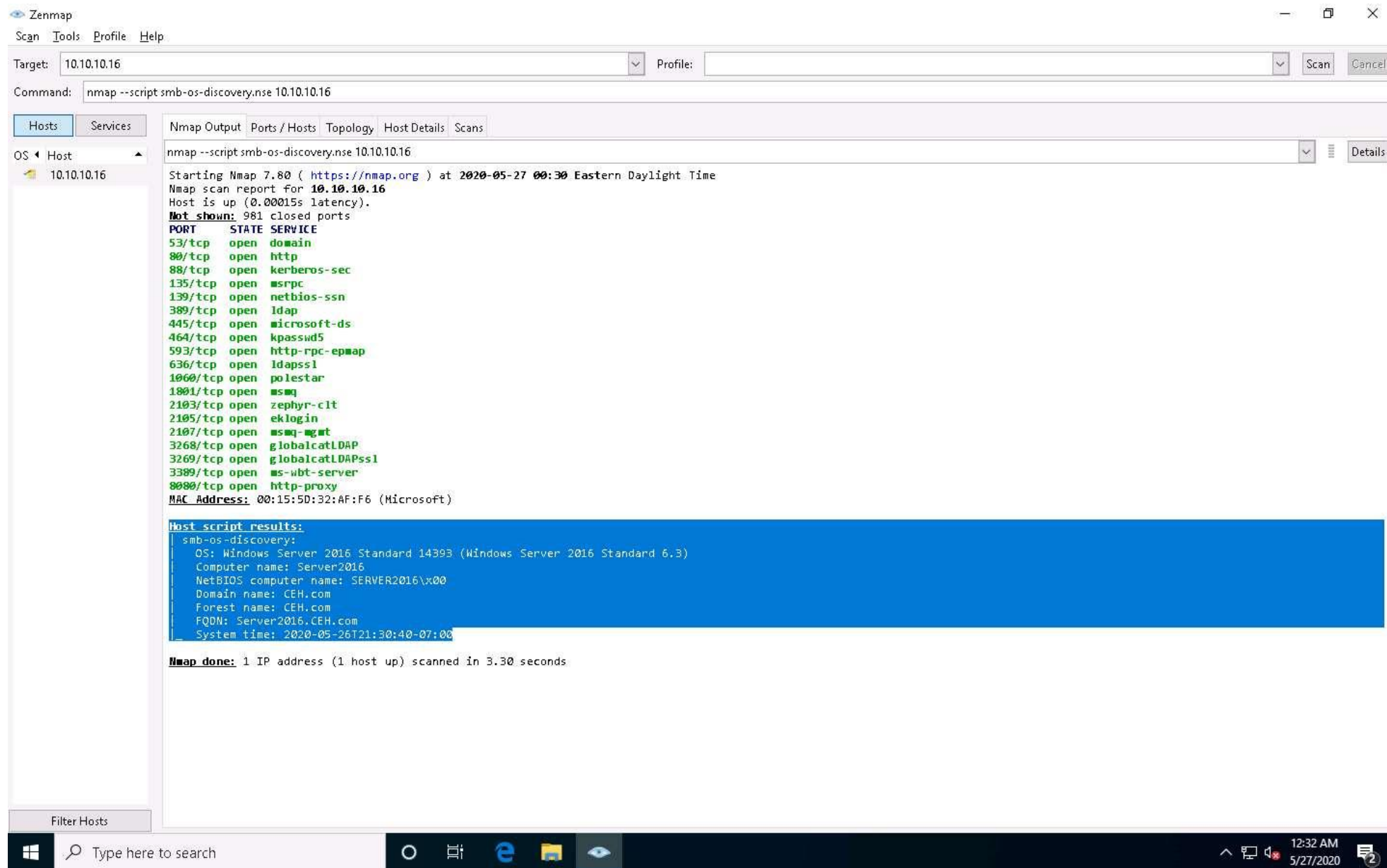
Type here to search

12:25 AM 5/27/2020

6. ☐ In the **Command** field, type the command **nmap --script smb-os-discovery.nse [Target IP Address]** (here, the target machine is **Windows Server 2016 [10.10.10.16]**) and click **Scan**.

--script: specifies the customized script and **smb-os-discovery.nse:** attempts to determine the OS, computer name, domain, workgroup, and current time over the SMB protocol (ports 445 or 139).

7. ☐ The scan results appear, displaying the target OS, computer name, NetBIOS computer name, etc. details under the **Host script results** section.



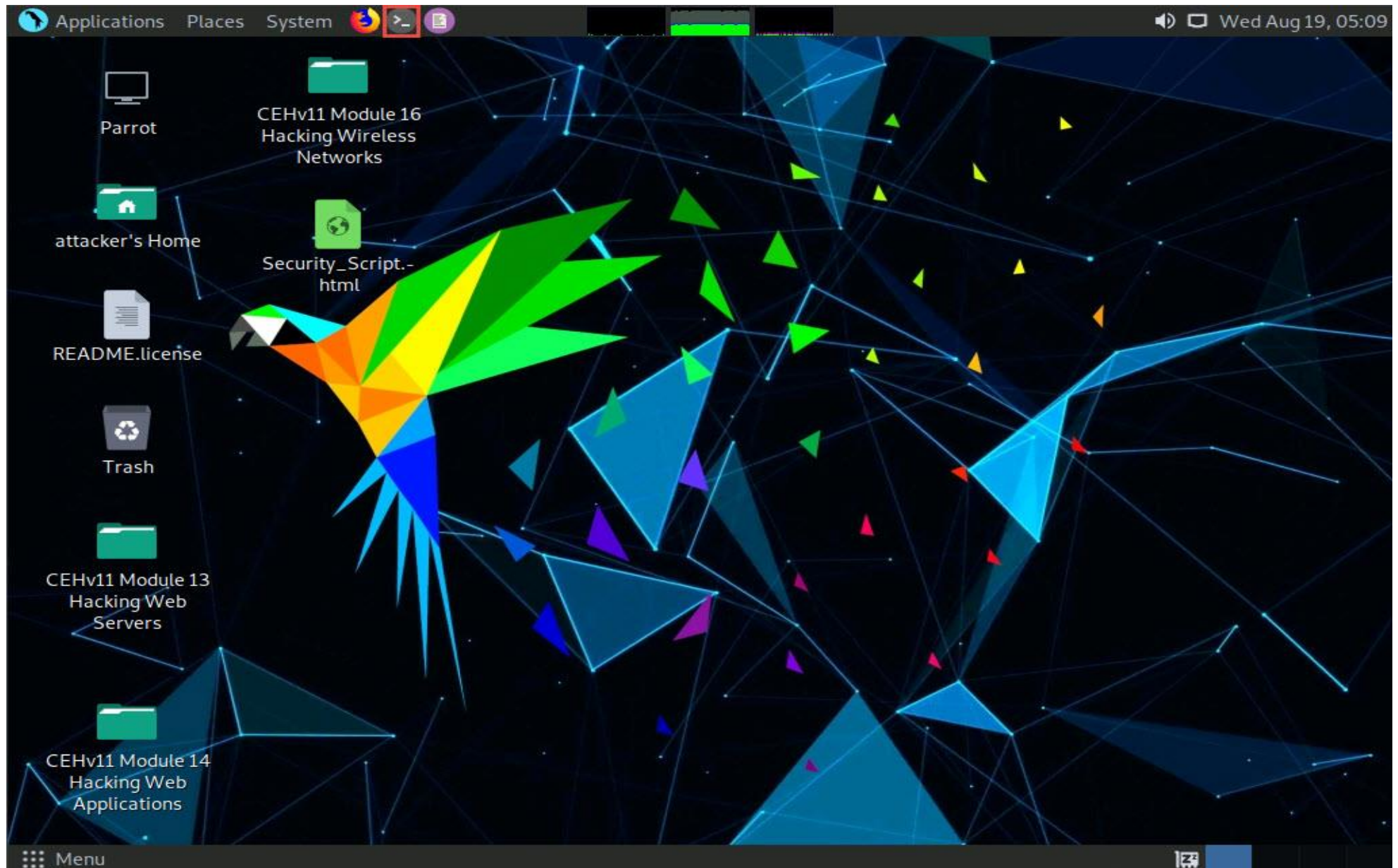
8. ☐ This concludes the demonstration of discovering the OS running on the target system using Nmap.
9. ☐ Close all open windows and document all the acquired information.

Task 3: Perform OS Discovery using Unicornscan

Unicornscan is a Linux-based command line-oriented network information-gathering and reconnaissance tool. It is an asynchronous TCP and UDP port scanner and banner grabber that enables you to discover open ports, services, TTL values, etc. running on the target machine. In Unicornscan, the OS of the target machine can be identified by observing the TTL values in the acquired scan result.

Here, we will use the Unicornscan tool to perform OS discovery on the target system.

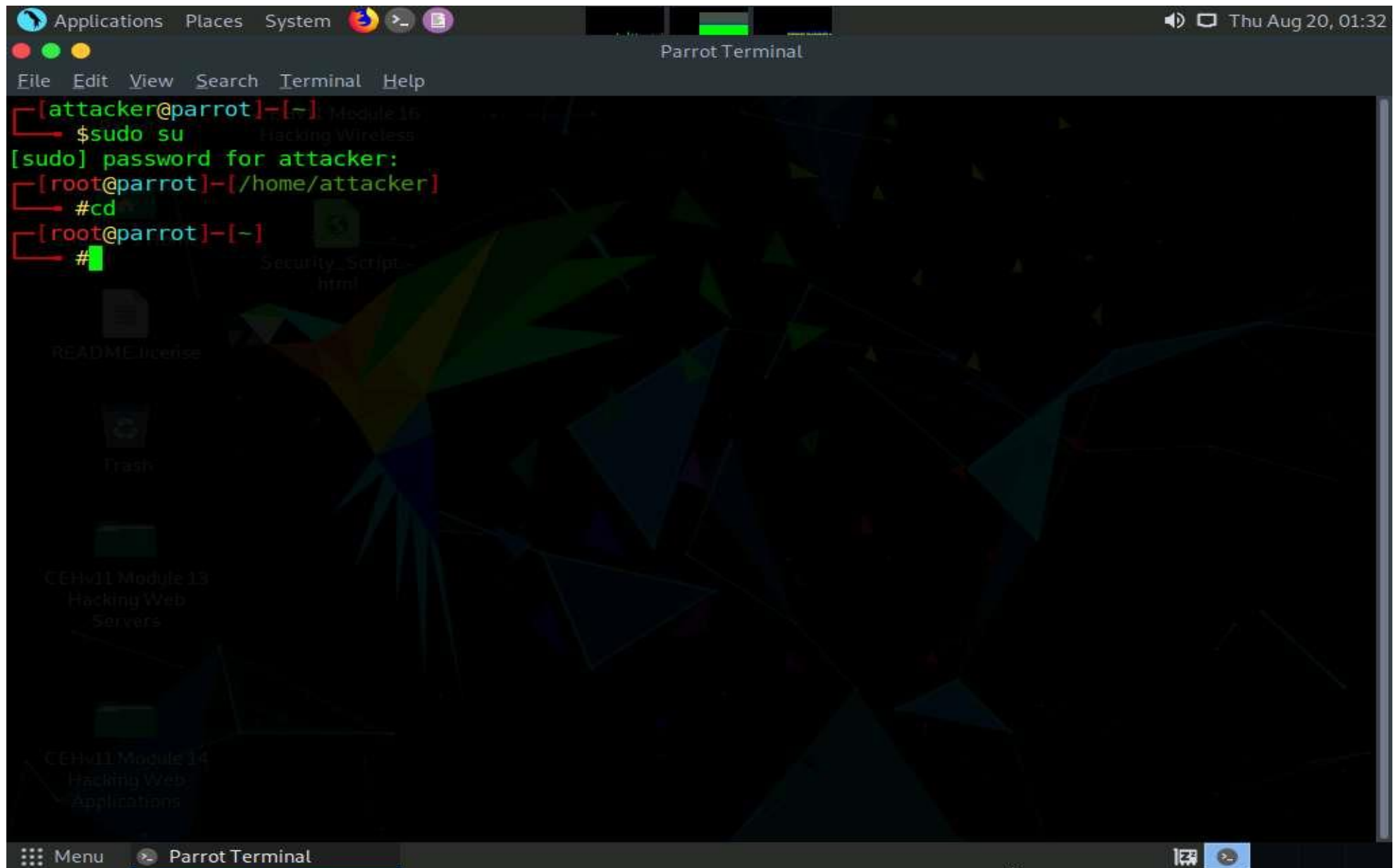
1. ☐ Click [Parrot Security](#) to switch to the **Parrot Security** machine.
2. ☐ Click the **MATE Terminal** icon at the top of the **Desktop** window to open a Terminal window.



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

The password that you type will not be visible.

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

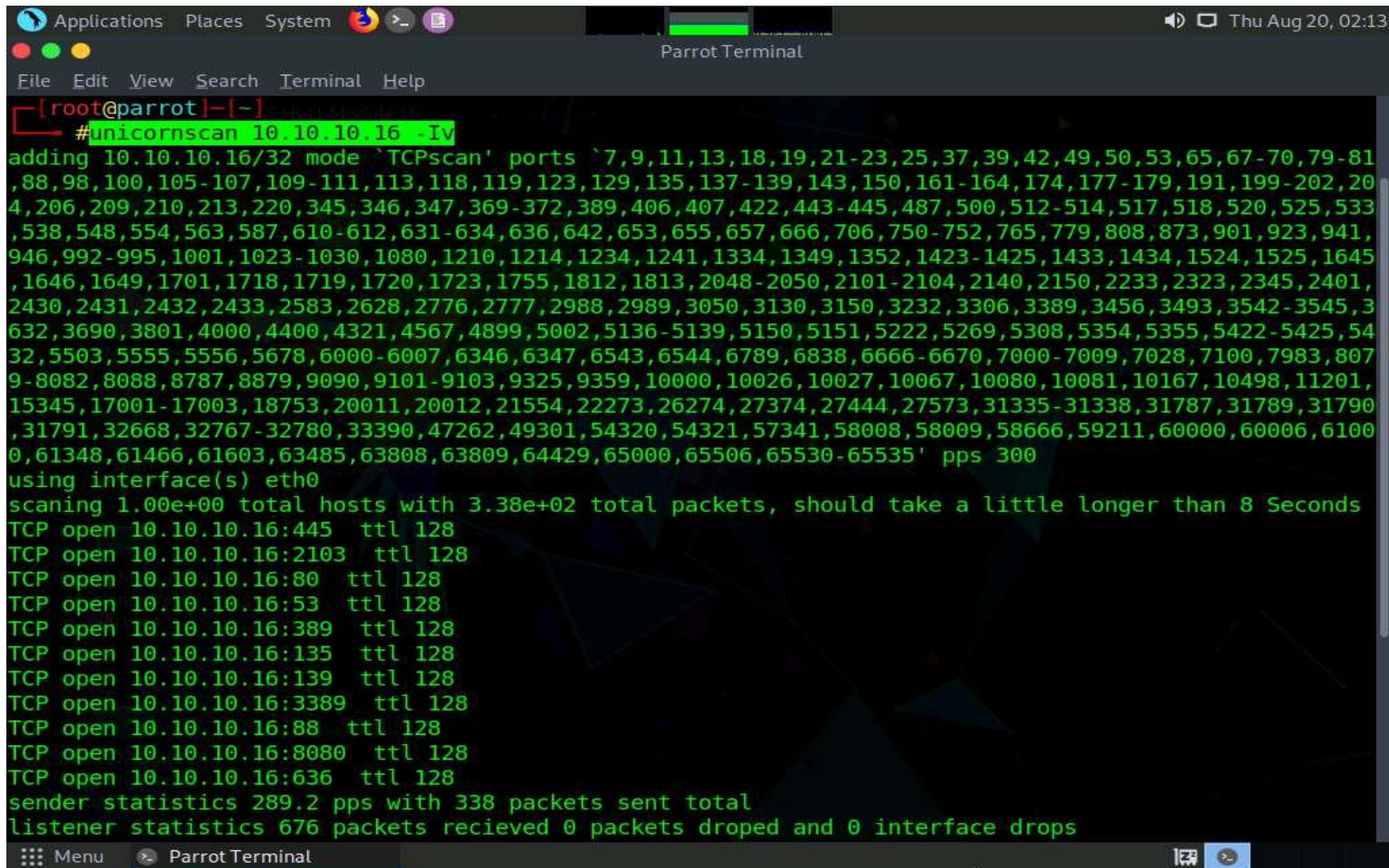


6. ☐ A **Parrot Terminal** window appears. In the terminal window, type **unicornscan [Target IP Address] -lv** (here, the target machine is **Windows Server 2016 [10.10.10.16]**) and press **Enter**.

In this command, **-I** specifies an immediate mode and **v** specifies a verbose mode.

7. ☐ The scan results appear, displaying the open TCP ports along with the obtained TTL value of **128**. As shown in the screenshot, the **ttl** values acquired after the scan are **128**; hence, the OS is possibly Microsoft Windows (Windows 7/8/8.1/10 or Windows Server 2008/12/16).

Here, the target machine is **Windows Server 2016 (10.10.10.16)**.

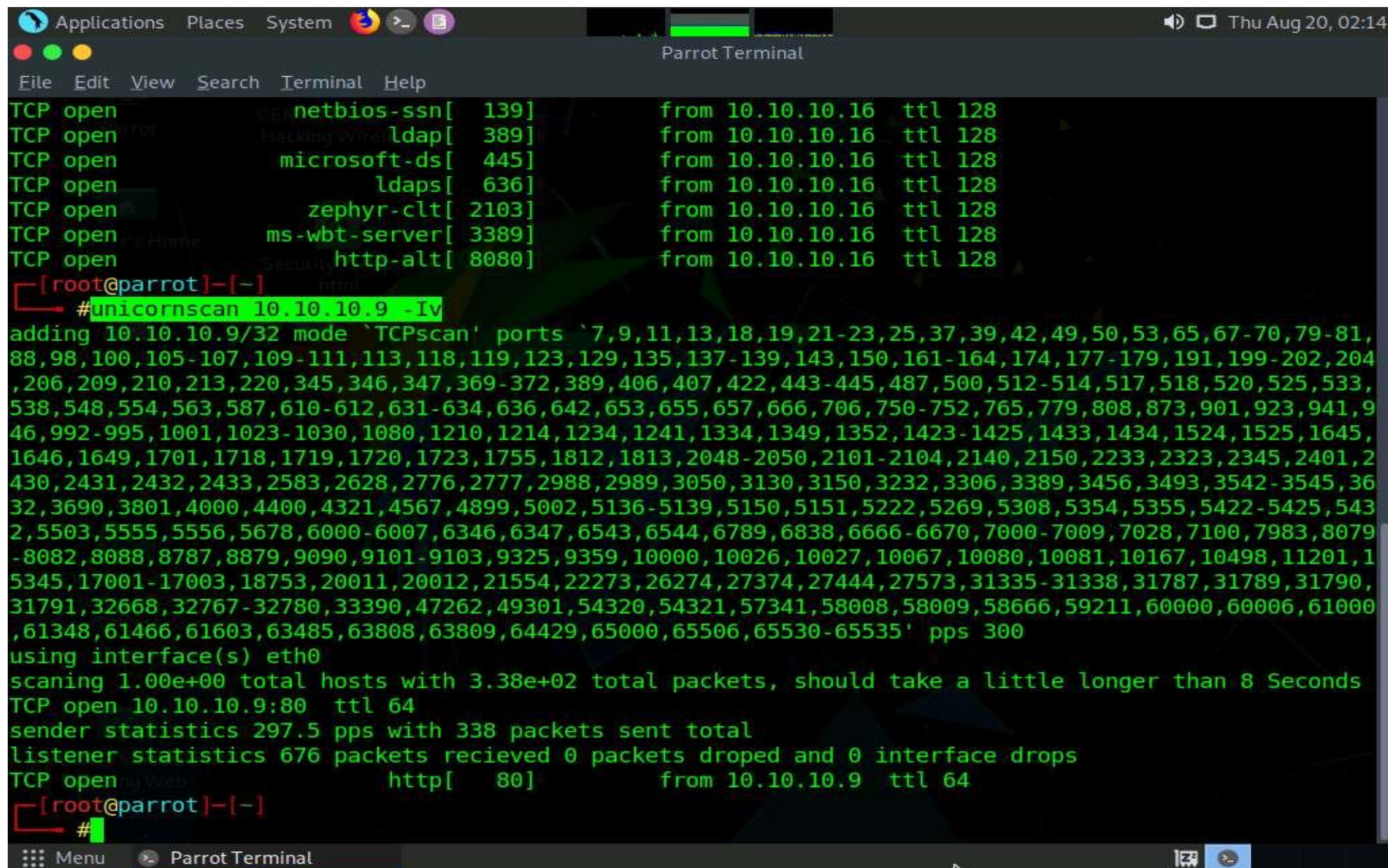


The screenshot shows a Parrot Terminal window with a dark theme. The title bar at the top includes 'Applications', 'Places', 'System', and 'Parrot Terminal'. The terminal prompt is '[root@parrot]~'. The command '#unicornscan 10.10.10.16 -lv' has been entered and executed. The output displays a list of ports being scanned, the interface used (eth0), the total number of hosts and packets, and the results of the scan, including open ports and statistics.

```
[root@parrot]~  
#unicornscan 10.10.10.16 -lv  
adding 10.10.10.16/32 mode 'TCPscan' ports `7,9,11,13,18,19,21-23,25,37,39,42,49,50,53,65,67-70,79-81  
,88,98,100,105-107,109-111,113,118,119,123,129,135,137-139,143,150,161-164,174,177-179,191,199-202,20  
4,206,209,210,213,220,345,346,347,369-372,389,406,407,422,443-445,487,500,512-514,517,518,520,525,533  
,538,548,554,563,587,610-612,631-634,636,642,653,655,657,666,706,750-752,765,779,808,873,901,923,941,  
946,992-995,1001,1023-1030,1080,1210,1214,1234,1241,1334,1349,1352,1423-1425,1433,1434,1524,1525,1645  
,1646,1649,1701,1718,1719,1720,1723,1755,1812,1813,2048-2050,2101-2104,2140,2150,2233,2323,2345,2401,  
2430,2431,2432,2433,2583,2628,2776,2777,2988,2989,3050,3130,3150,3232,3306,3389,3456,3493,3542-3545,3  
632,3690,3801,4000,4400,4321,4567,4899,5002,5136-5139,5150,5151,5222,5269,5308,5354,5355,5422-5425,54  
32,5503,5555,5556,5678,6000-6007,6346,6347,6543,6544,6789,6838,6666-6670,7000-7009,7028,7100,7983,807  
9-8082,8088,8787,8879,9090,9101-9103,9325,9359,10000,10026,10027,10067,10080,10081,10167,10498,11201,  
15345,17001-17003,18753,20011,20012,21554,22273,26274,27374,27444,27573,31335-31338,31787,31789,31790  
,31791,32668,32767-32780,33390,47262,49301,54320,54321,57341,58008,58009,58666,59211,60000,60006,6100  
0,61348,61466,61603,63485,63808,63809,64429,65000,65506,65530-65535' pps 300  
using interface(s) eth0  
scanning 1.00e+00 total hosts with 3.38e+02 total packets, should take a little longer than 8 Seconds  
TCP open 10.10.10.16:445  ttl 128  
TCP open 10.10.10.16:2103  ttl 128  
TCP open 10.10.10.16:80   ttl 128  
TCP open 10.10.10.16:53   ttl 128  
TCP open 10.10.10.16:389  ttl 128  
TCP open 10.10.10.16:135  ttl 128  
TCP open 10.10.10.16:139  ttl 128  
TCP open 10.10.10.16:3389 ttl 128  
TCP open 10.10.10.16:88   ttl 128  
TCP open 10.10.10.16:8080 ttl 128  
TCP open 10.10.10.16:636  ttl 128  
sender statistics 289.2 pps with 338 packets sent total  
listener statistics 676 packets recieved 0 packets dropped and 0 interface drops
```

8. ☐ In the **Parrot Terminal** window, type **unicornscan [Target IP Address] -lv** (here, the target machine is **Ubuntu [10.10.10.9]**) and press **Enter**.

9. ☐ The scan results appear, displaying the open TCP ports along with a TTL value of **64**. As shown in the screenshot, the **ttl** value acquired after the scan is **64**; hence, the OS is possibly a Linux-based machine (Google Linux, Ubuntu, Parrot, or Kali).



```
Applications Places System [Icons] [Terminal] [Help] Thu Aug 20, 02:14
Parrot Terminal
File Edit View Search Terminal Help
TCP open netbios-ssn[ 139] from 10.10.10.16 ttl 128
TCP open ldap[ 389] from 10.10.10.16 ttl 128
TCP open microsoft-ds[ 445] from 10.10.10.16 ttl 128
TCP open ldaps[ 636] from 10.10.10.16 ttl 128
TCP open zephyr-clt[ 2103] from 10.10.10.16 ttl 128
TCP open ms-wbt-server[ 3389] from 10.10.10.16 ttl 128
TCP open http-alt[ 8080] from 10.10.10.16 ttl 128
[root@parrot]-[~]
#unicornscan 10.10.10.9 -Iv
adding 10.10.10.9/32 mode `TCPscan' ports `7,9,11,13,18,19,21-23,25,37,39,42,49,50,53,65,67-70,79-81,
88,98,100,105-107,109-111,113,118,119,123,129,135,137-139,143,150,161-164,174,177-179,191,199-202,204
,206,209,210,213,220,345,346,347,369-372,389,406,407,422,443-445,487,500,512-514,517,518,520,525,533,
538,548,554,563,587,610-612,631-634,636,642,653,655,657,666,706,750-752,765,779,808,873,901,923,941,9
46,992-995,1001,1023-1030,1080,1210,1214,1234,1241,1334,1349,1352,1423-1425,1433,1434,1524,1525,1645,
1646,1649,1701,1718,1719,1720,1723,1755,1812,1813,2048-2050,2101-2104,2140,2150,2233,2323,2345,2401,2
430,2431,2432,2433,2583,2628,2776,2777,2988,2989,3050,3130,3150,3232,3306,3389,3456,3493,3542-3545,36
32,3690,3801,4000,4400,4321,4567,4899,5002,5136-5139,5150,5151,5222,5269,5308,5354,5355,5422-5425,543
2,5503,5555,5556,5678,6000-6007,6346,6347,6543,6544,6789,6838,6666-6670,7000-7009,7028,7100,7983,8079
-8082,8088,8787,8879,9090,9101-9103,9325,9359,10000,10026,10027,10067,10080,10081,10167,10498,11201,1
5345,17001-17003,18753,20011,20012,21554,22273,26274,27374,27444,27573,31335-31338,31787,31789,31790,
31791,32668,32767-32780,33390,47262,49301,54320,54321,57341,58008,58009,58666,59211,60000,60006,61000
,61348,61466,61603,63485,63808,63809,64429,65000,65506,65530-65535' pps 300
using interface(s) eth0
scanning 1.00e+00 total hosts with 3.38e+02 total packets, should take a little longer than 8 Seconds
TCP open 10.10.10.9:80 ttl 64
sender statistics 297.5 pps with 338 packets sent total
listener statistics 676 packets recieved 0 packets dropped and 0 interface drops
TCP open http[ 80] from 10.10.10.9 ttl 64
[root@parrot]-[~]
#
```


10. ☐ This concludes the demonstration of discovering the OS of the target machine using Unicornscan.
11. ☐ Close all open windows and document all the acquired information.