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

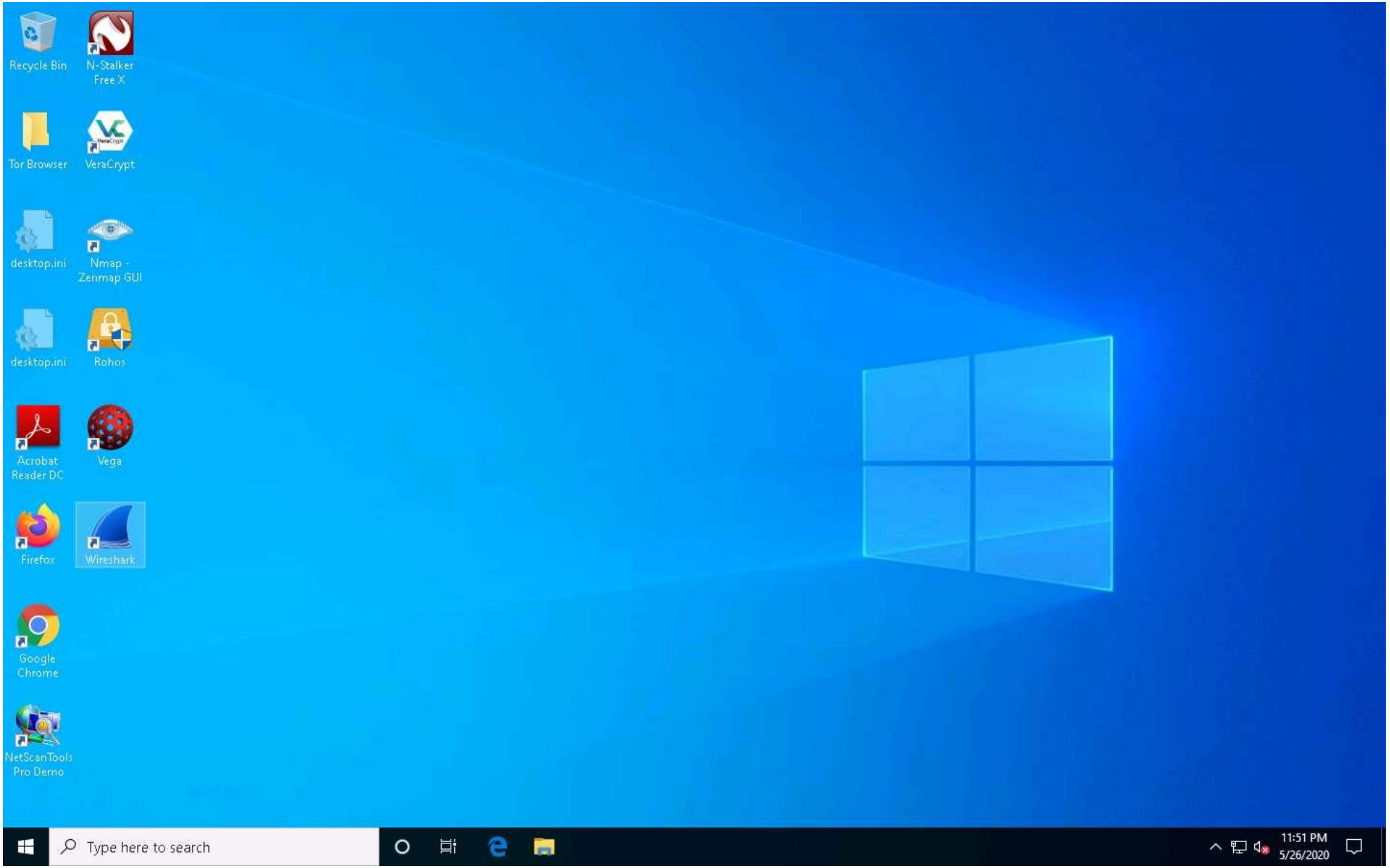
Operating System (OS)	Time To Live	TCP Window Size
FreeBSD	64	65535
OpenBSD	64	16384
Windows 95	32	8192
Windows 2000	128	16384
Windows XP	128	65535
Windows 98, Vista and 7 (Server 2008)	128	8192
iOS 12.4 (Cisco Routers)	255	4128
Solaris 7	255	8760
AIX 4.3	64	16384

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.

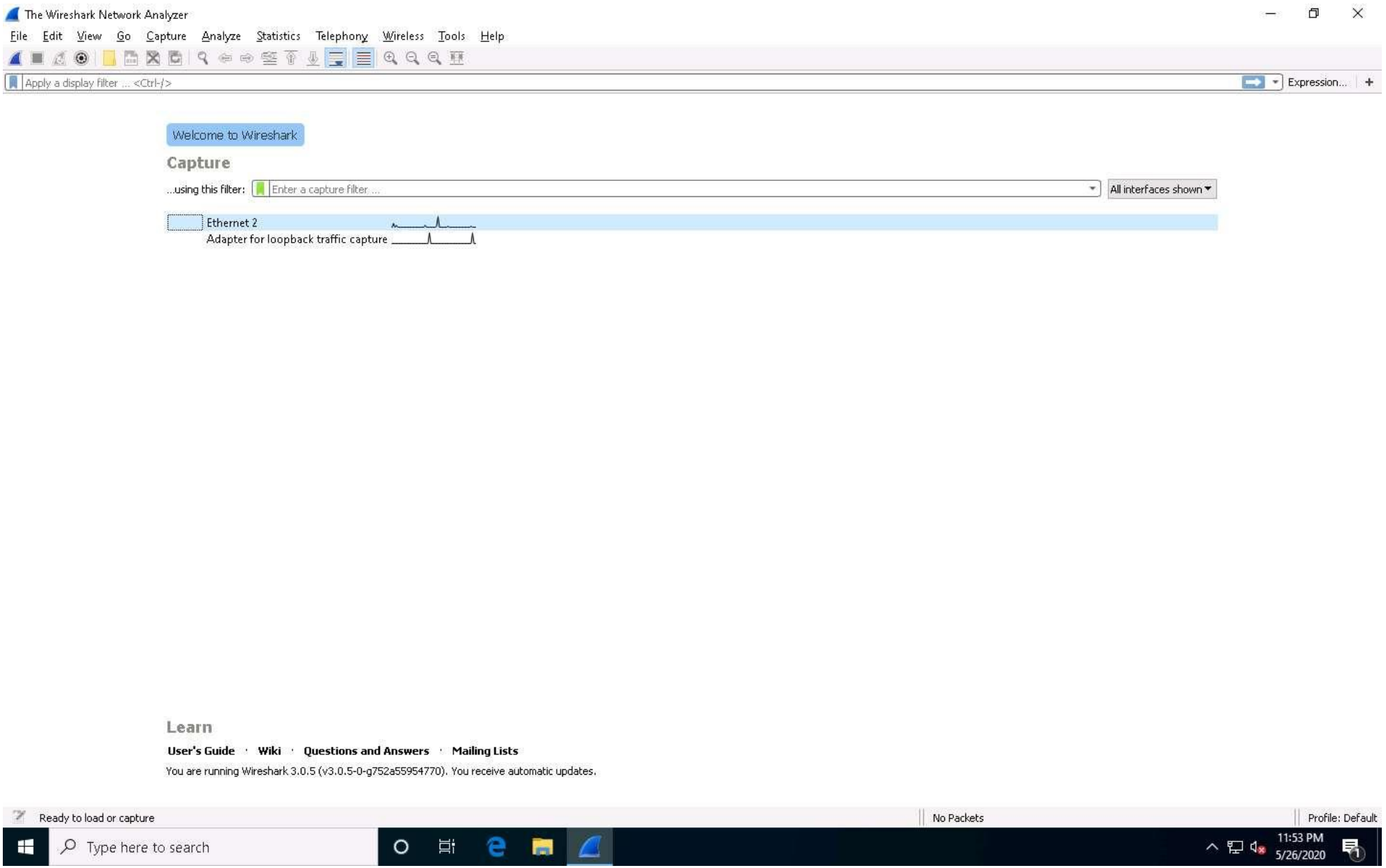
Here, we will use the Wireshark tool to perform OS discovery on the target host(s).

1. Click Windows 10 to switch to the **Windows 10** machine.
2. In the **Desktop**, double-click **Wireshark** shortcut.



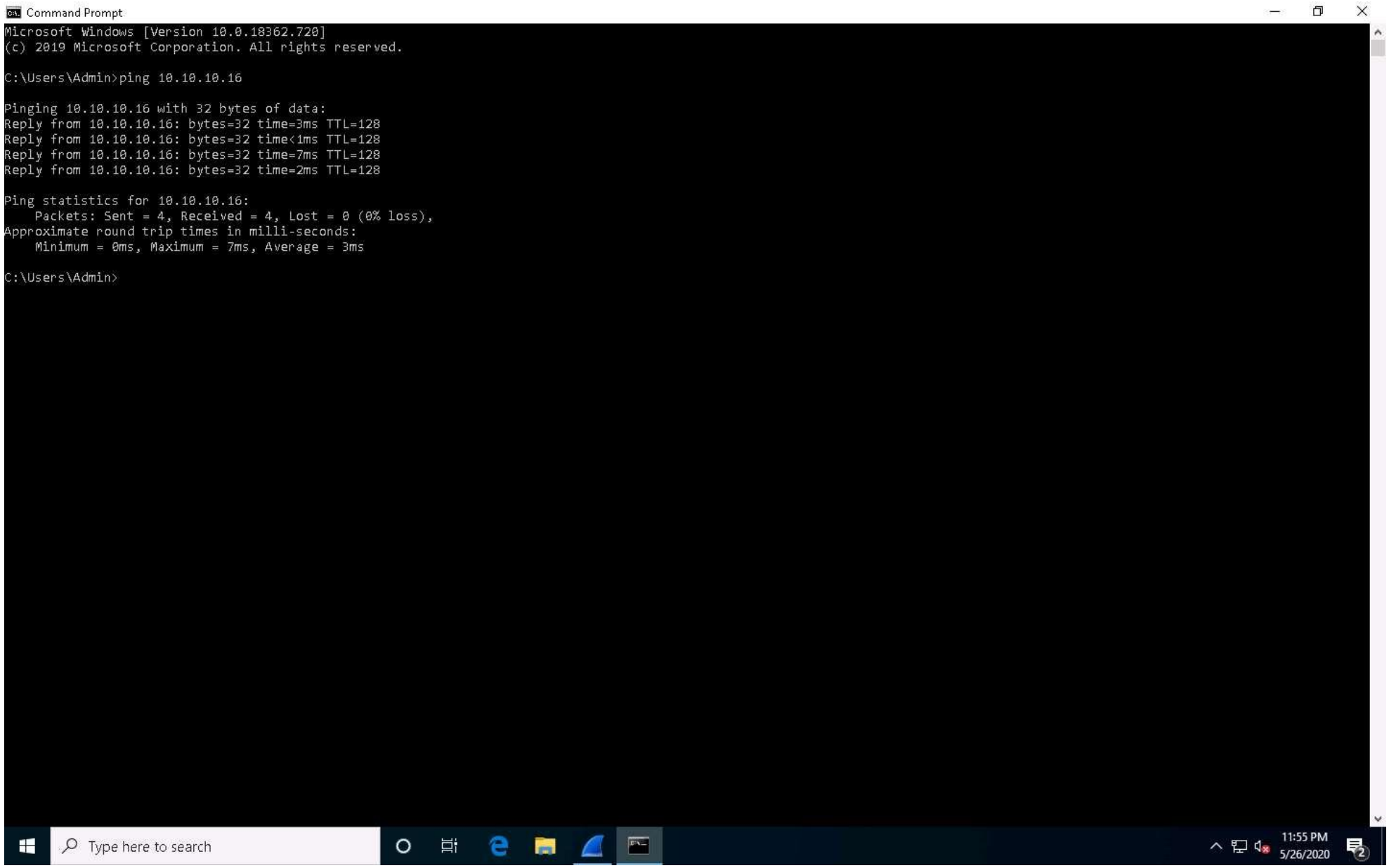
3. ☐ The **Wireshark Network Analyzer** main window appears; double-click the available ethernet or interface (here, **Ethernet2**) to start the packet capture, as shown in the screenshot.

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

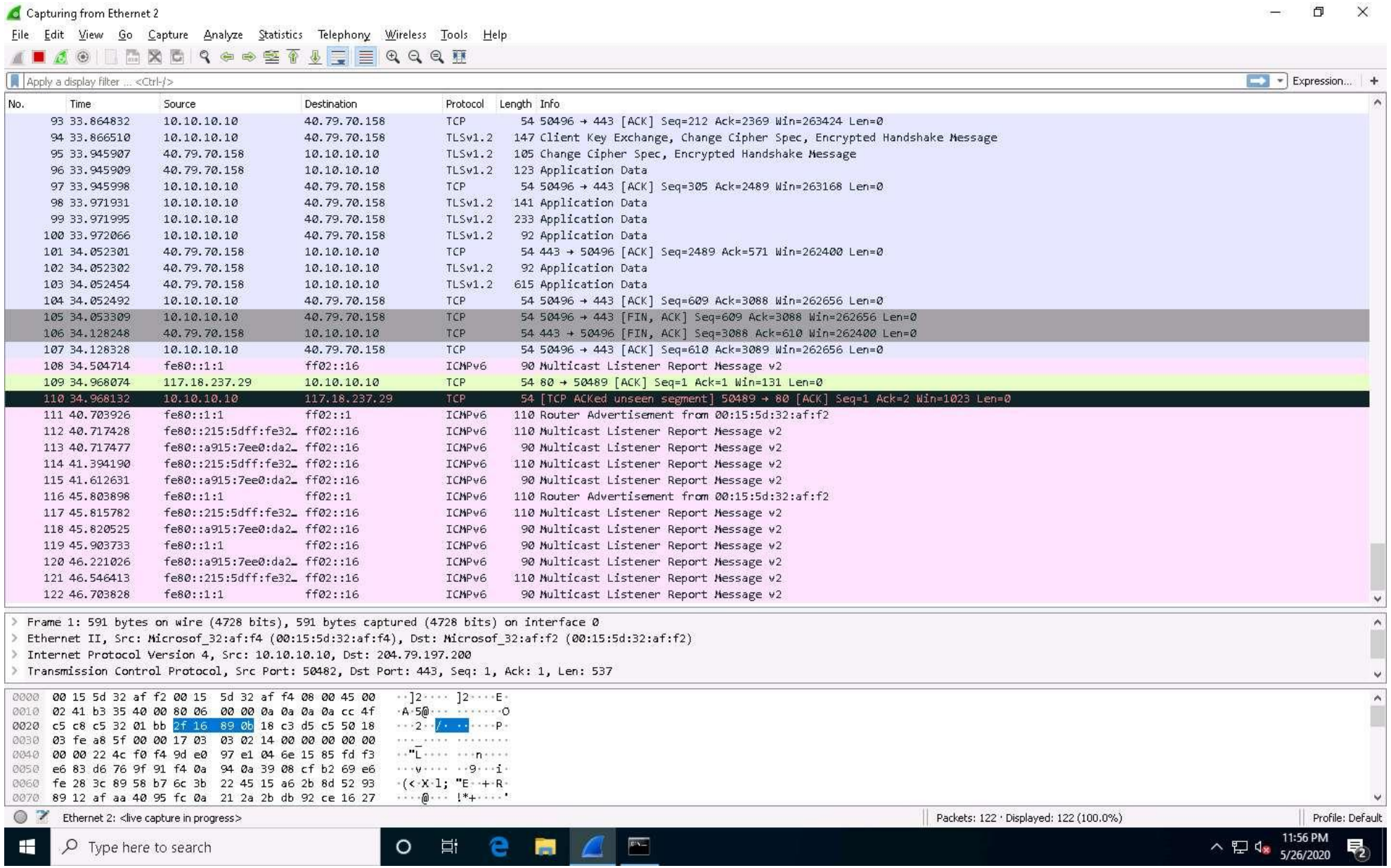


4. ☐ Open the **Command Prompt**, type **ping 10.10.10.16** and press **Enter**.

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

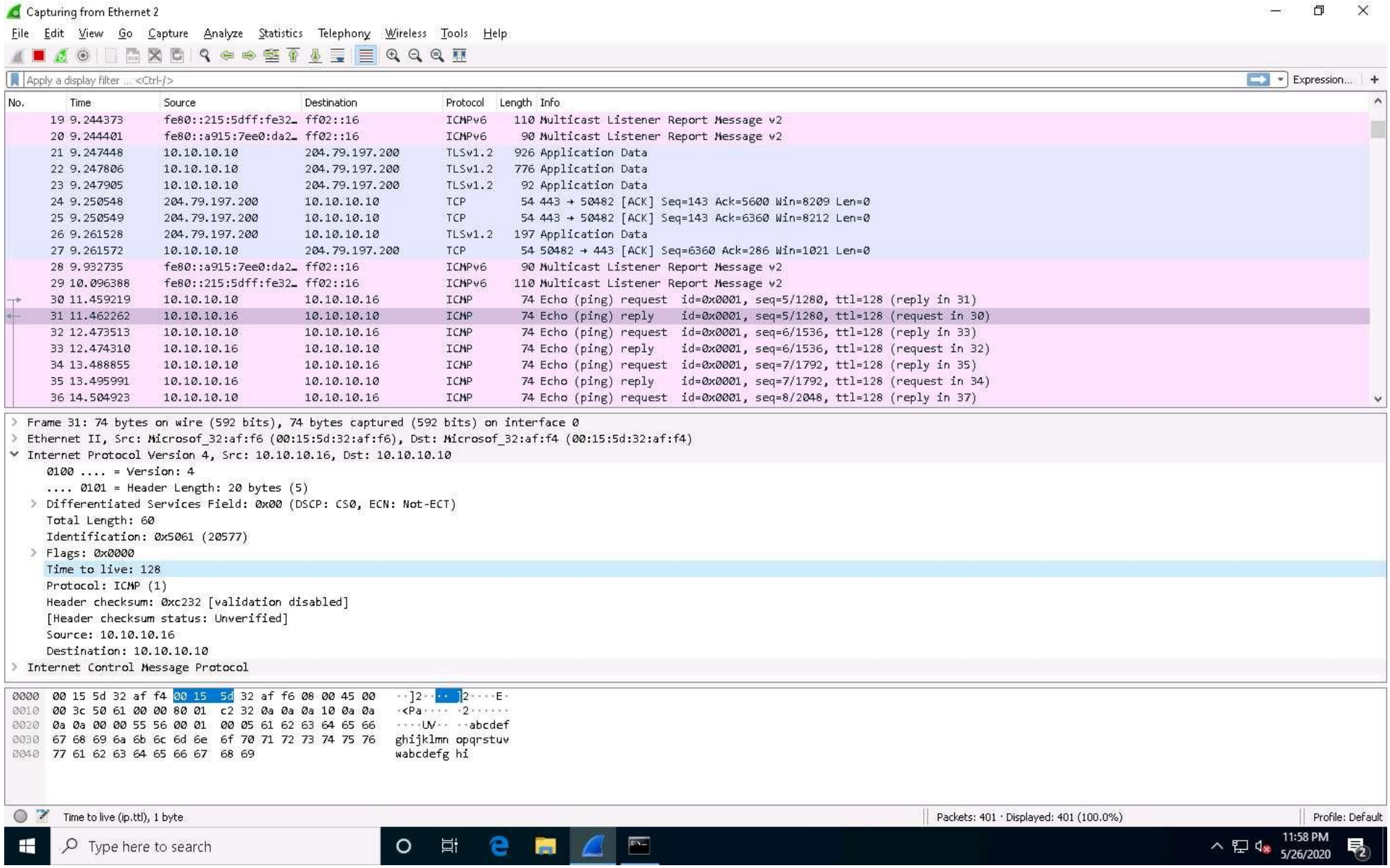


5. ☐ Observe the packets captured by **Wireshark**.

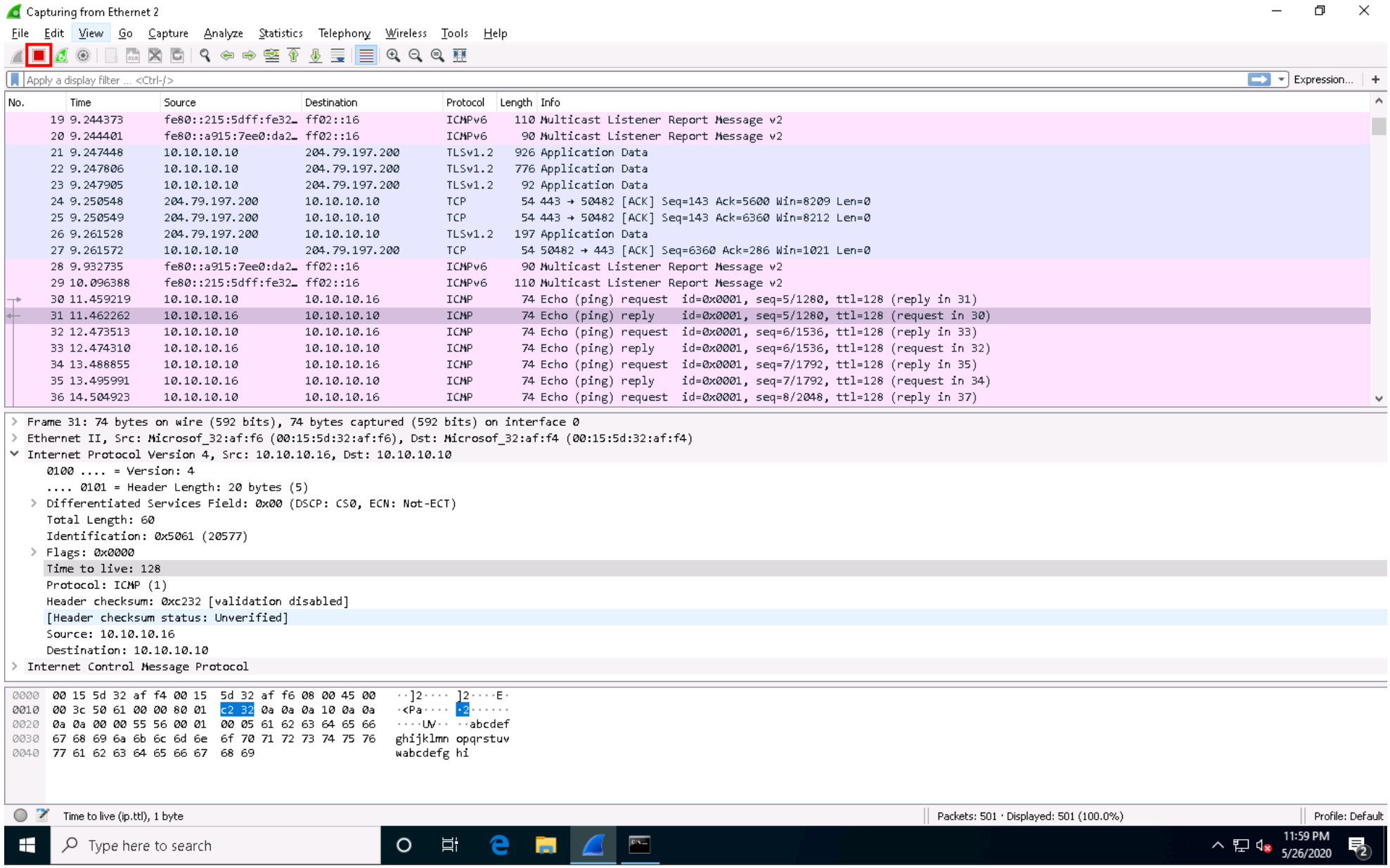


6. Choose any packet of the ICMP reply from the **Windows Server 2016** (**10.10.10.16**) to **Windows 10** (**10.10.10.10**) machines and expand the **Internet Protocol Version 4** node in the **Packet Details** pane.

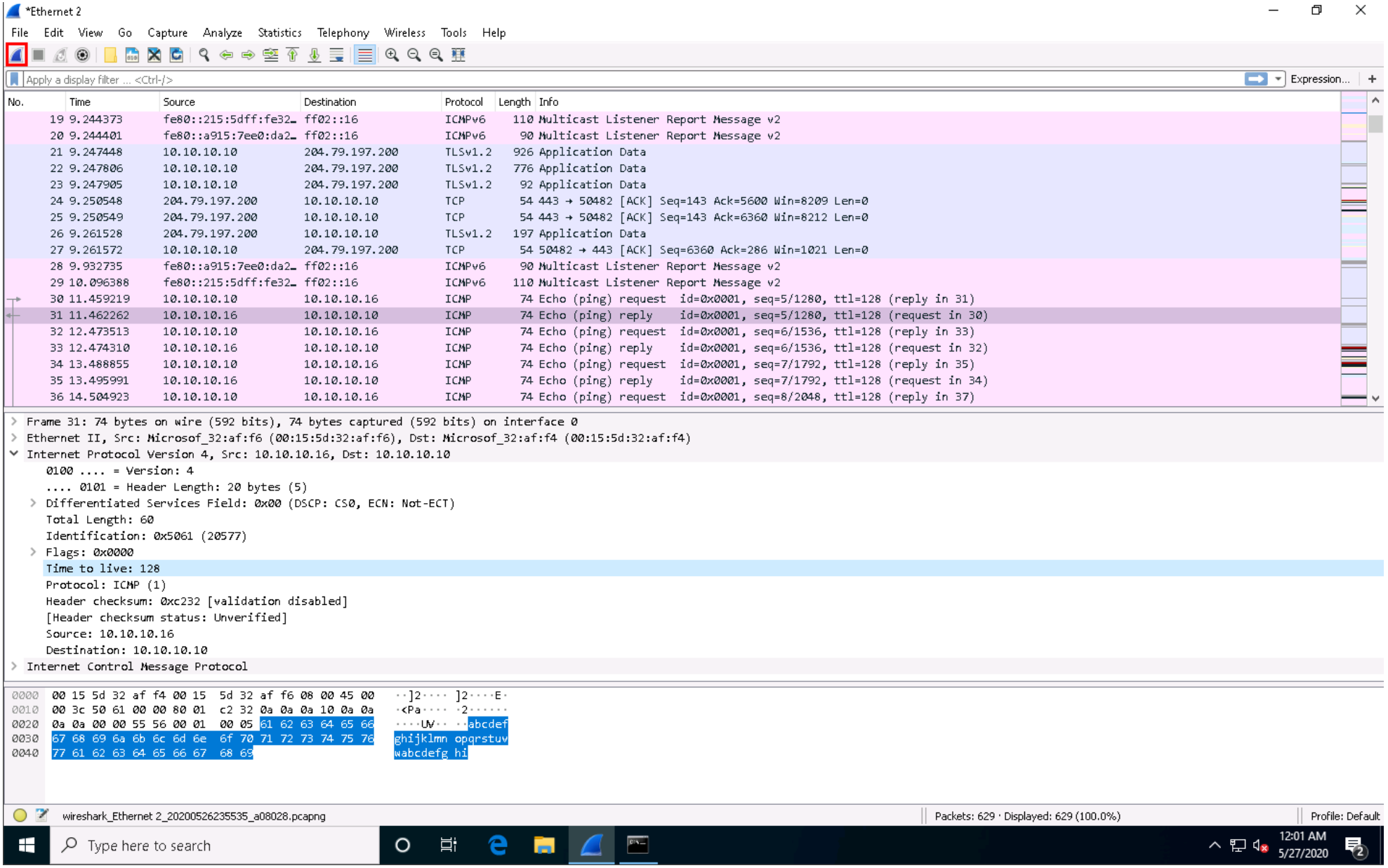
7. ☐ The TTL value is recorded as **128**, which means that the ICMP reply possibly came from a Windows-based machine.



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

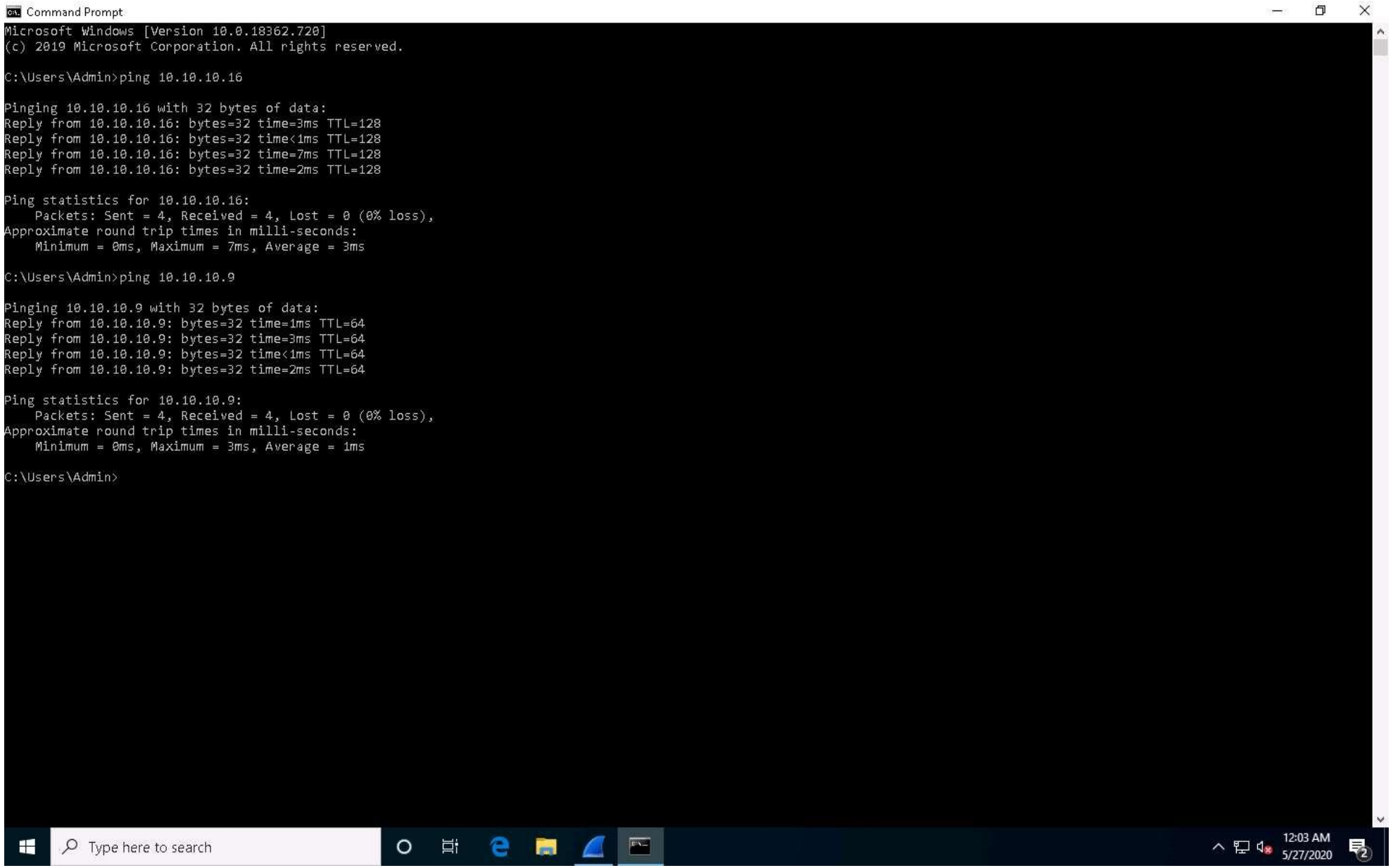


9. Now, click the **Start capturing packets** button from the toolbar. If an **Unsaved packets...** pop-up appears, click **Continue without Saving**.



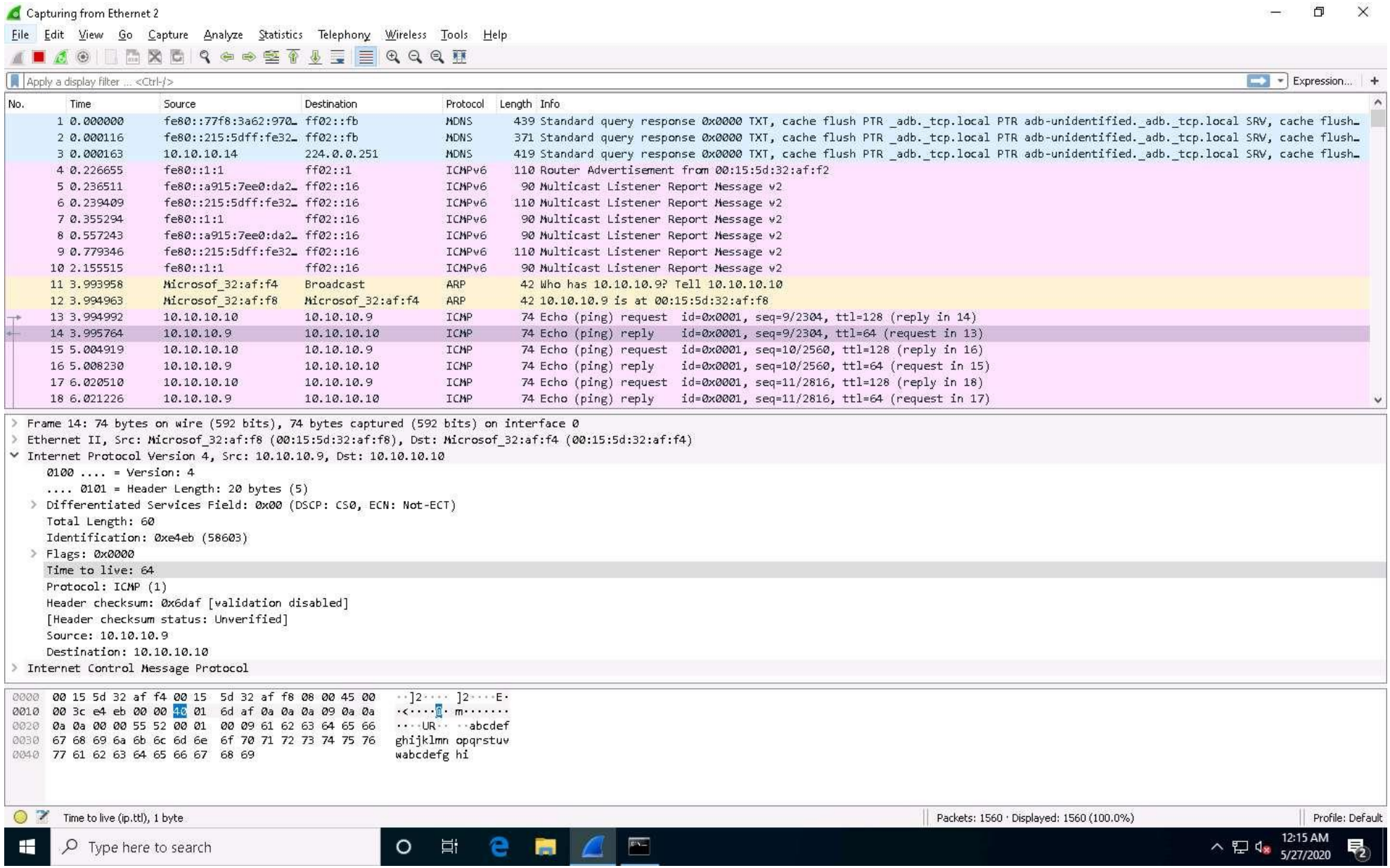
10. Wireshark will start capturing the new packets.
11. In the **Command Prompt** window, type **ping 10.10.10.9** and press **Enter**.

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



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.



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.

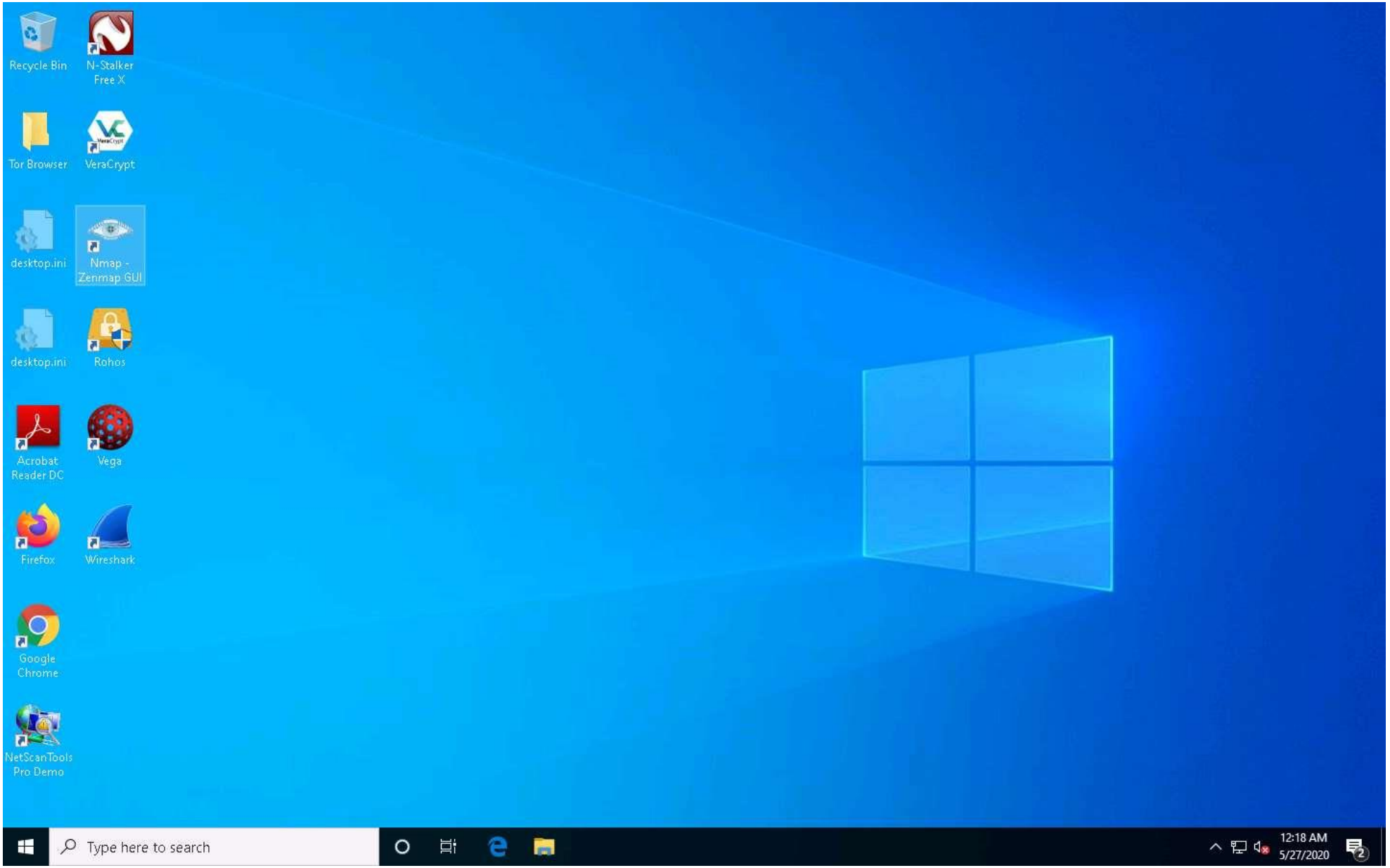
17. ☐ Close all open windows and document all the acquired information.

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.

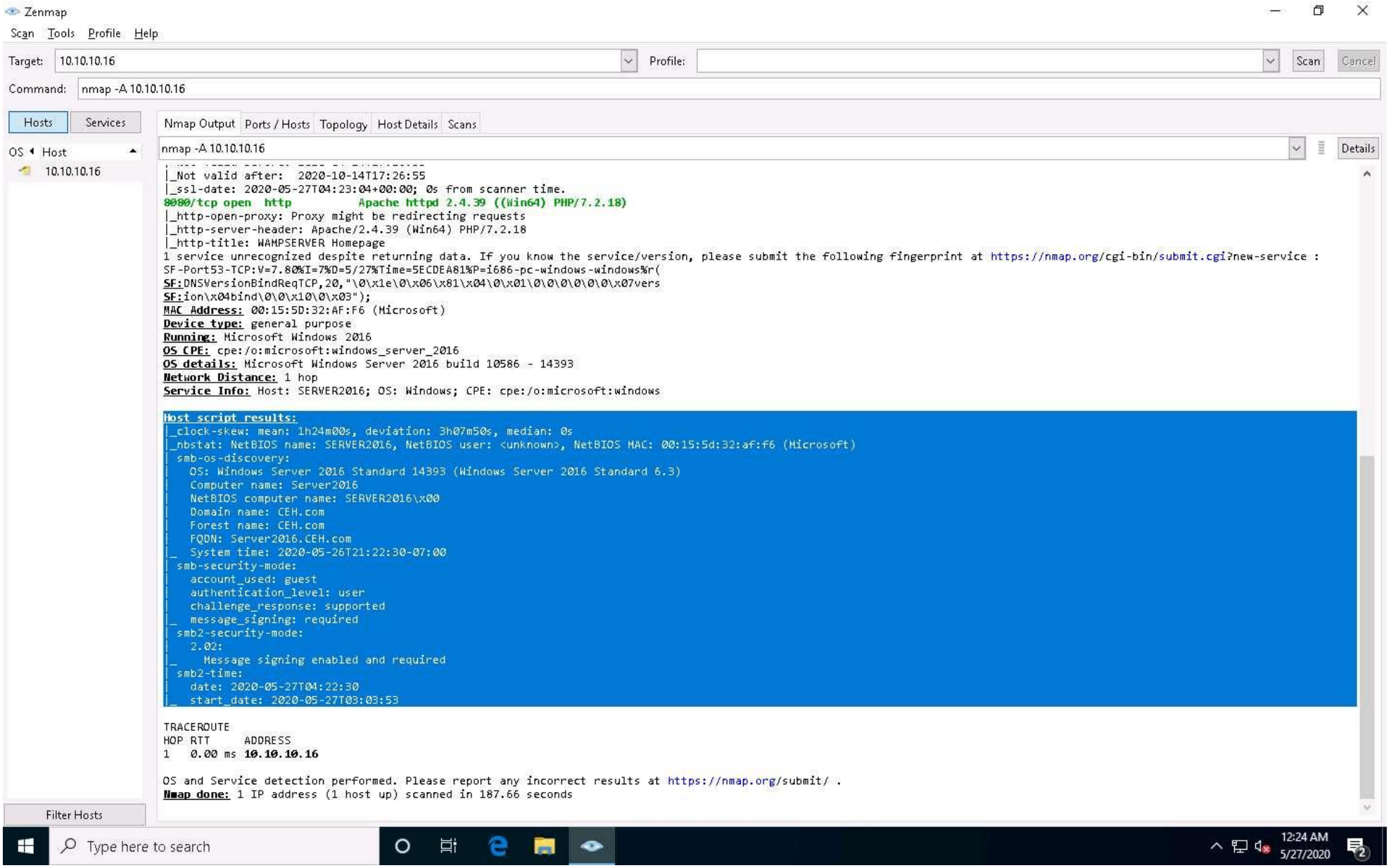


2. ☐ The **Zenmap GUI** appears. In the **Command** field, type the command **nmap -A [Target IP Address]** (here, the target machine is **Windows Server 2016** [**10.10.10.16**]) and click **Scan**.

-A: to perform an aggressive scan.

The scan takes aprroximately 10 minutes to complete.

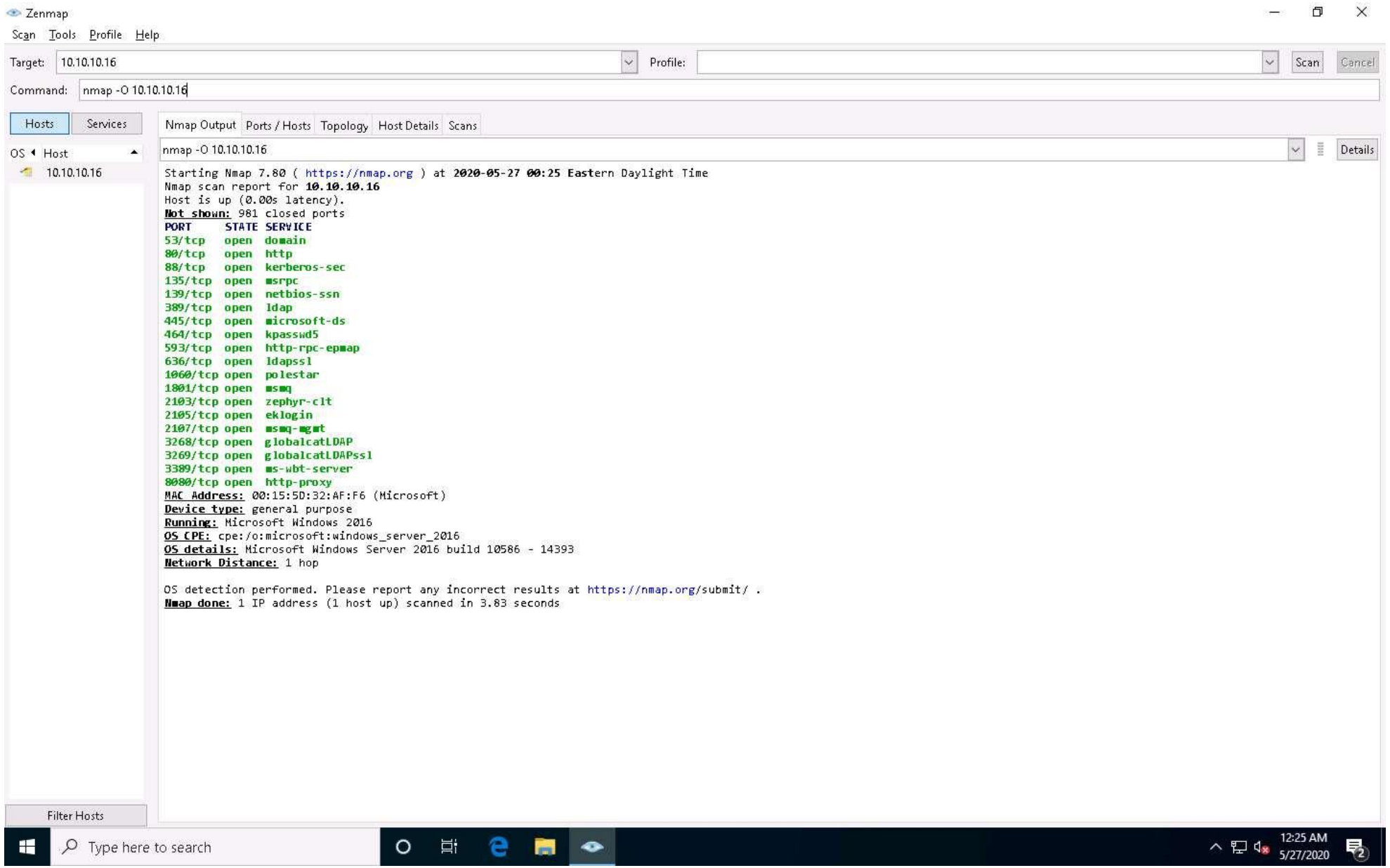
3. The scan results appear, displaying the open ports and running services along with their versions and target details such as OS, computer name, NetBIOS computer name, etc. under the **Host script results** section.



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.

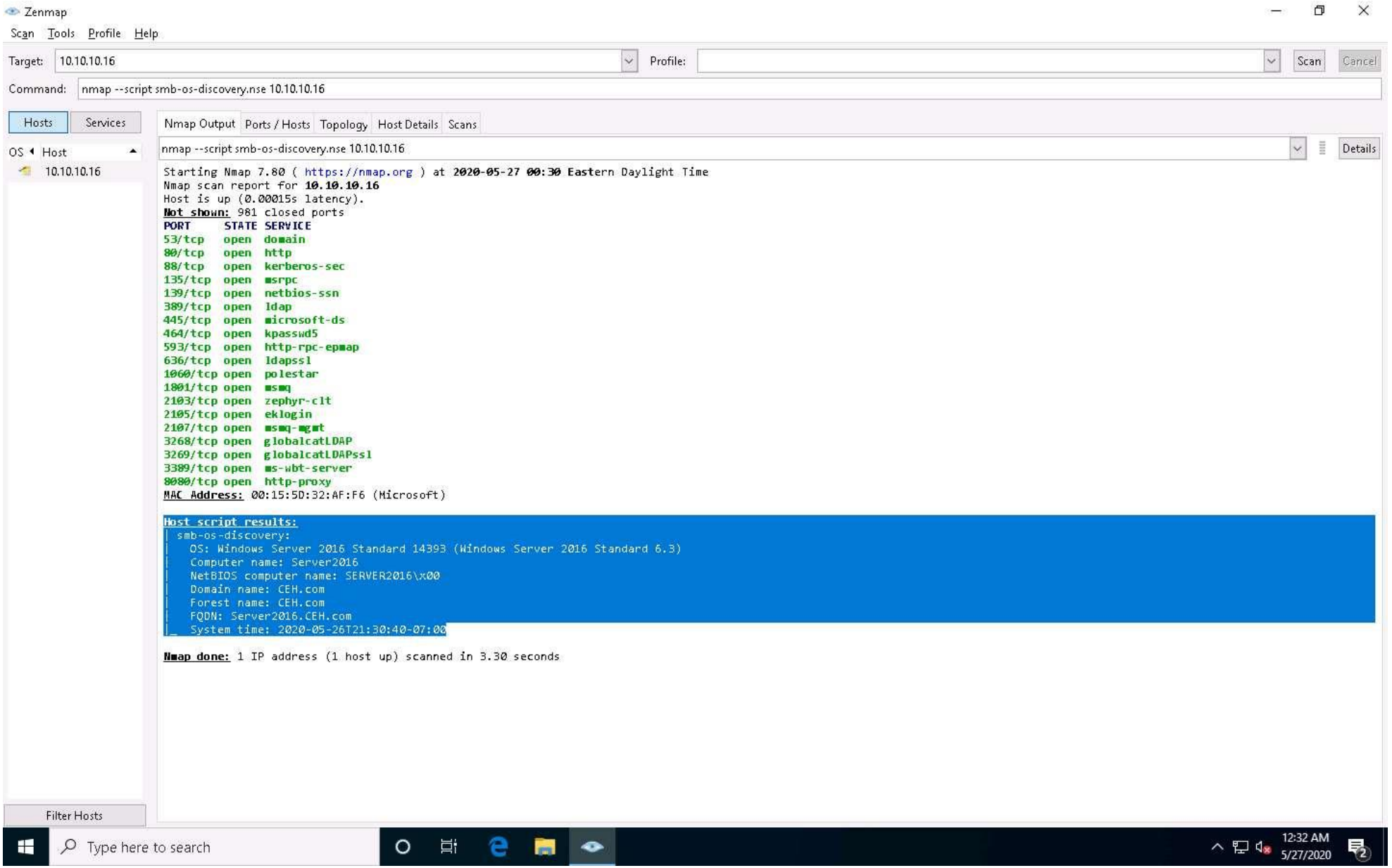
5. The scan results appear, displaying information about open ports, respective services running on the open ports, and the name of the OS running on the target system.



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.



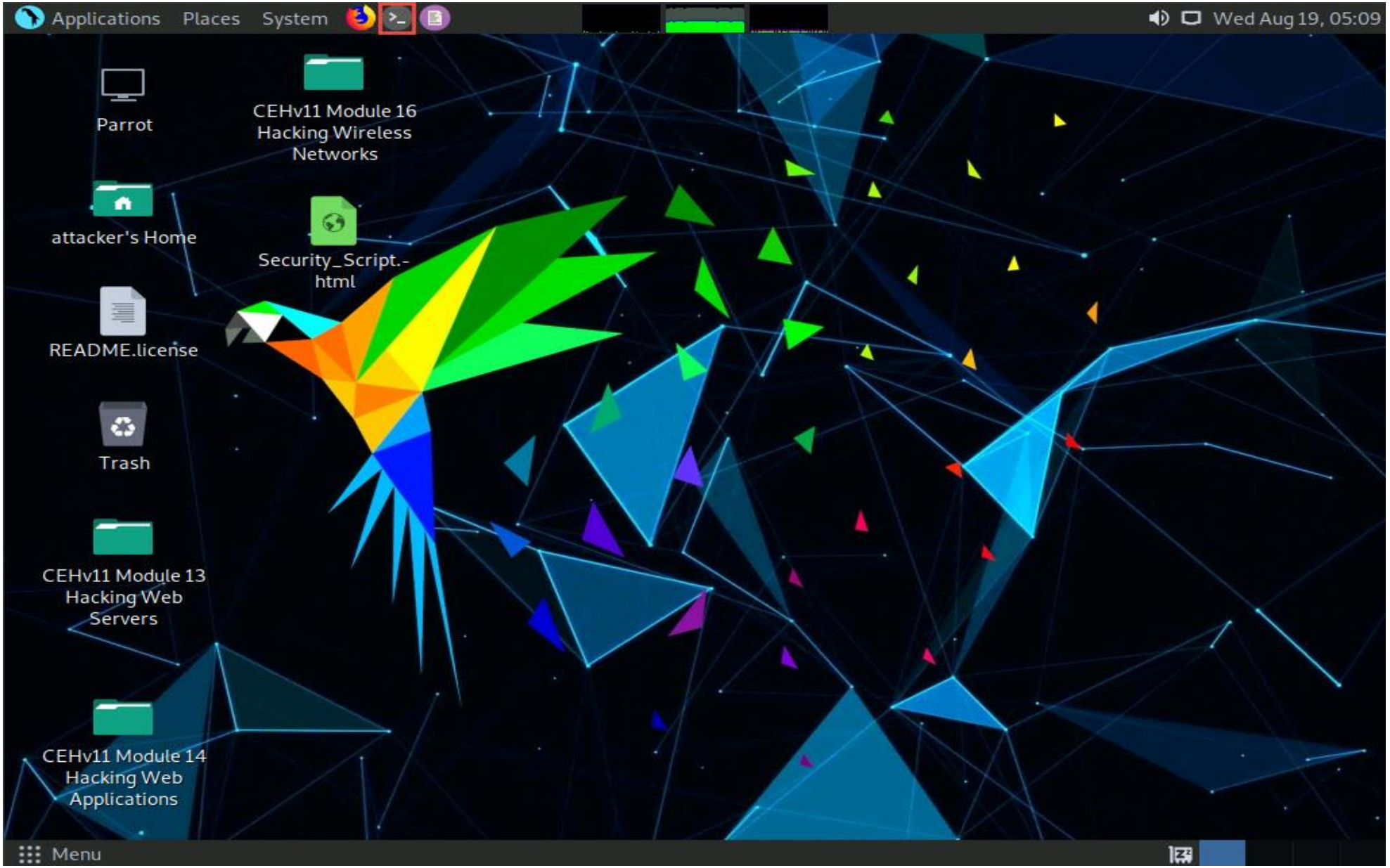
8. ☐ This concludes the demonstration of discovering the OS running on the target system using Nmap.
9. ☐ Close all open windows and document all the acquired information.

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.

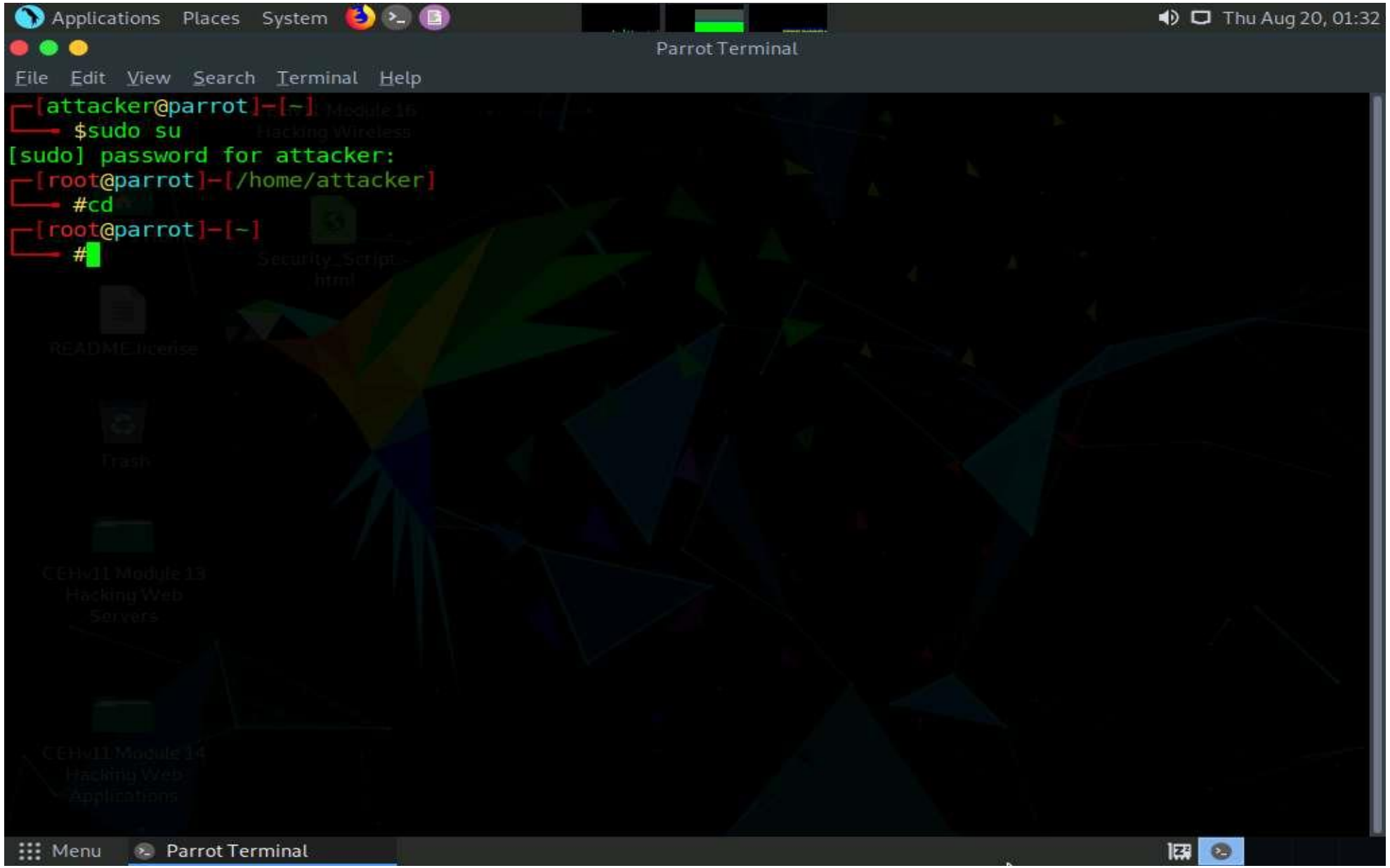
1. Click Parrot Security to switch to the **Parrot Security** machine.
2. Click the **MATE Terminal** icon at the top of the **Desktop** window to open a Terminal window.



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.

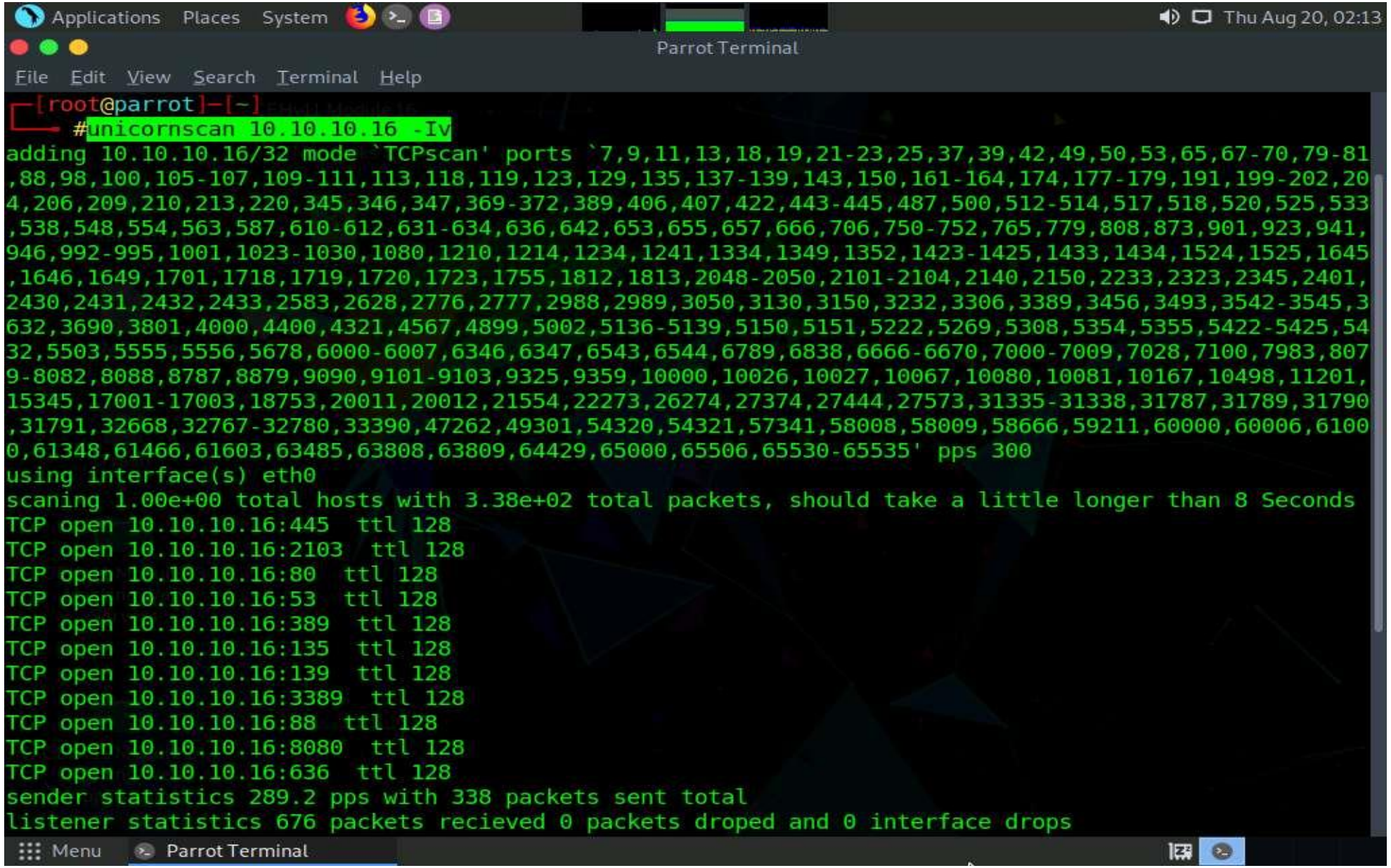


6. A **Parrot Terminal** window appears. In the terminal window, type **unicornscan [Target IP Address] -Iv** (here, the target machine is **Windows Server 2016** [**10.10.10.16**]) and press **Enter**.

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

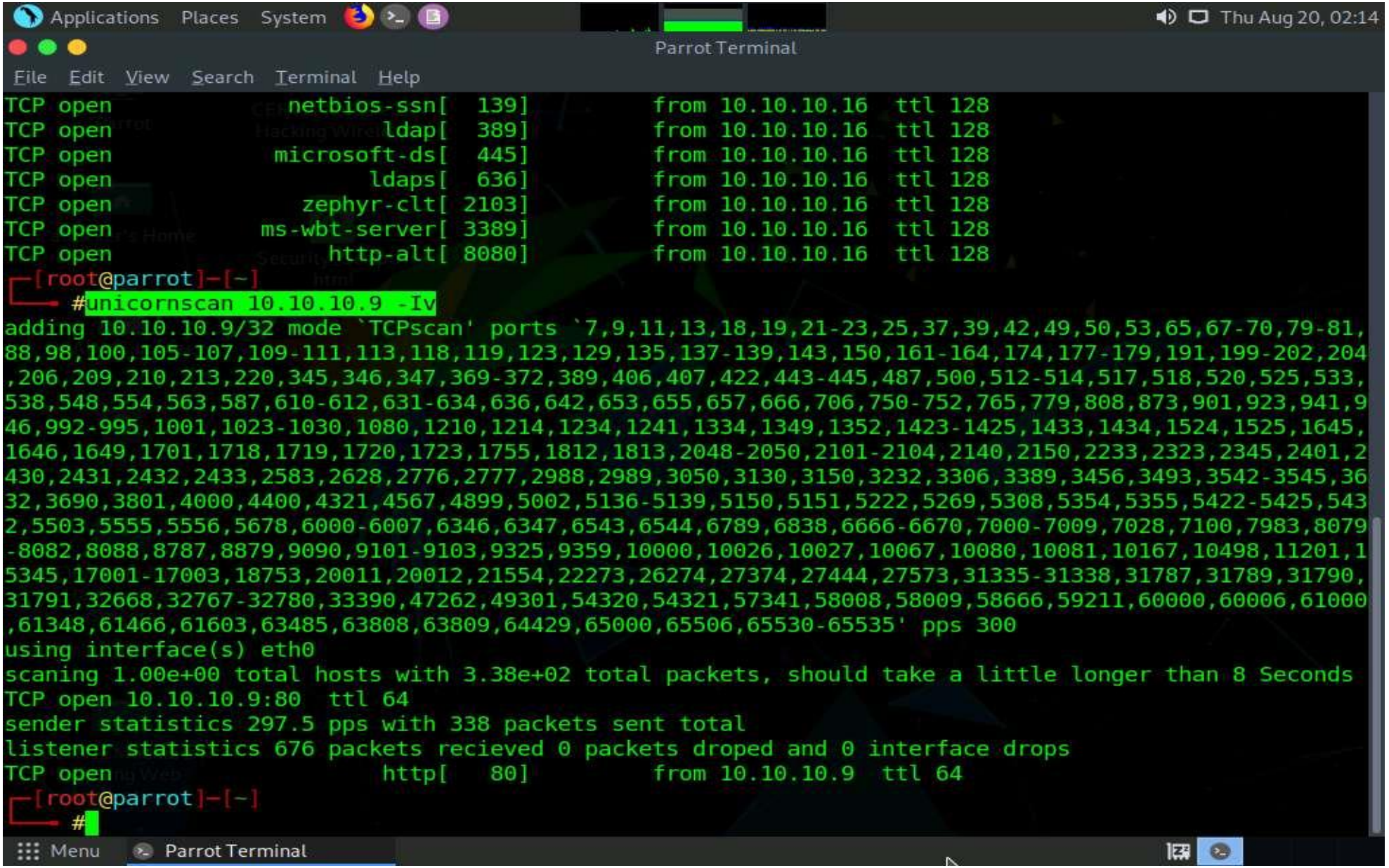
7. The scan results appear, displaying the open TCP ports along with the obtained TTL value of **128**. As shown in the screenshot, the **ttl** values acquired after the scan are **128**; hence, the OS is possibly Microsoft Windows (Windows 7/8/8.1/10 or Windows Server 2008/12/16).

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



8. ☐ In the **Parrot Terminal** window, type **unicornscan [Target IP Address] -Iv** (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).



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.