Scanning Networks

Module 03

Scanning a Target Network

Network scanning refers to a set of procedures performed to identify the hosts, ports, and services running in a network.

Lab Scenario

ICON KEY

Valuable information

Test your knowledge

Web exercise

Workbook review

Earlier, you gathered all possible information about the target such as organization information (employee details, partner details, web links, etc.), network information (domains, sub-domains, IP addresses, network topology, etc.), and system information (OS details, user accounts, passwords, etc.).

Now, as an ethical hacker, or as a penetration tester (hereafter, pen tester), your next step will be to perform port scanning and network scanning on the IP addresses that you obtained in the information-gathering phase. This will help you to identify an entry point into the target network.

Scanning itself is not the actual intrusion, but an extended form of reconnaissance in which the ethical hacker and pen tester learns more about the target, including information about open ports and services, OSes, and any configuration lapses. The information gleaned from this reconnaissance helps you to select strategies for the attack on the target system or network.

This is one of the most important phases of intelligence gathering, which enables you to create a profile of the target organization. In the process of scanning, you attempt to gather information, including the specific IP addresses of the target system that can be accessed over the network (live hosts), open ports, and respective services running on the open ports and vulnerabilities in the live hosts.

Tools demonstrated in this lab are available in E:\CEH-Tools\CEHv11 Module 03 Scanning Networks

Port scanning will help you identify open ports and services running on specific ports, which involves connecting to Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) system ports. Port scanning is also used to discover the vulnerabilities in the services running on a port.

The labs in this module will give you real-time experience in gathering information about the target organization using various network scanning and port scanning techniques.

Lab Objectives

The objective of this lab is to conduct network scanning, port scanning, analyzing the network vulnerabilities, etc.

Network scans are needed to:

- Check live systems and open ports
- Identify services running in live systems
- Perform banner grabbing/OS fingerprinting
- Identify network vulnerabilities
- Draw network diagrams of vulnerable hosts

Lab Environment

To carry out this lab, you need:

- Windows Server 2019 virtual machine
- Windows Server 2016 virtual machine
- Windows 10 virtual machine
- Ubuntu virtual machine
- Parrot Security virtual machine
- Web browsers with an Internet connection
- Administrator privileges to run the tools

Lab Duration

Time: 140 Minutes

Overview of Scanning Networks

Network scanning is the process of gathering additional detailed information about the target by using highly complex and aggressive reconnaissance techniques. The purpose of scanning is to discover exploitable communication channels, probe as many listeners as possible, and keep track of the responsive ones.

Types of scanning:

- **Port Scanning**: Lists open ports and services
- **Network Scanning**: Lists the active hosts and IP addresses
- **Vulnerability Scanning**: Shows the presence of known weaknesses

Lab Tasks

Ethical hackers and pen testers use numerous tools and techniques to scan the target network. Recommended labs that will assist you in learning various network scanning techniques include:

Lab No.	Lab Exercise Name	Core*	Self-study**	iLabs ***
1	Perform Host Discovery	√	√	√
	1.1 Perform Host Discovery using Nmap	√		√
	1.2 Perform Host Discovery using Angry IP Scanner		√	√
2	Perform Port and Service Discovery	√	√	√
	2.1 Perform Port and Service Discovery using MegaPing		√	√

	2.2 Perform Port and Service Discovery using NetScanTools Pro		√	√
	2.3 Explore Various Network Scanning Techniques using Nmap	√		√
	2.4 Explore Various Network Scanning Techniques using Hping3		√	√
3	Perform OS Discovery	√	√	√
	3.1 Identify the Target System's OS with Time-to-Live (TTL) and TCP Window Sizes using Wireshark		√	√
	3.2 Perform OS Discovery using Nmap Script Engine (NSE)	√		√
	3.3 Perform OS Discovery using Unicornscan		√	√
4	Scan beyond IDS and Firewall	√	√	√
	4.1 Scan beyond IDS/Firewall using various Evasion Techniques	√		√
	4.2 Create Custom Packets using Colasoft Packet Builder to Scan beyond IDS/Firewall		√	√
	4.3 Create Custom UDP and TCP Packets using Hping3 to Scan beyond IDS/Firewall	√		√
	4.4 Create Custom Packets using Nmap to Scan beyond IDS/Firewall		√	√
	4.5 Browse Anonymously using Proxy Switcher		√	
	4.6 Browse Anonymously using CyberGhost VPN		√	
5	Draw Network Diagrams		√	√
	5.1 Draw Network Diagrams using Network Topology Mapper		√	√
6	Perform Network Scanning using Various Scanning Tools	√		√
	6.1 Scan a Target Network using Metasploit	√		√

Remark

EC-Council has prepared a considered amount of lab exercises for student to practice during the 5-day class and at their free time to enhance their knowledge and skill.

***Core** - Lab exercise(s) marked under Core are recommended by EC-Council to be practised during the 5-day class.

****Self-study** - Lab exercise(s) marked under self-study is for students to practise at their free time. Steps to access the additional lab exercises can be found in the first page of CEHv11 volume 1 book.

***iLabs** - Lab exercise(s) marked under iLabs are available in our iLabs solution. iLabs is a cloud-based virtual lab environment preconfigured with vulnerabilities, exploits, tools and scripts, and can be accessed from anywhere with an Internet connection. If you are interested to learn more about our iLabs solution, please contact your training center or visit https://ilabs.eccouncil.org.

Lab Analysis

Analyze and document the results related to the lab exercise. Give your opinion on your target's security posture and exposure through public and free information.

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS RELATED TO THIS LAB.

Lab 1

Perform Host Discovery

Host discovery is the process of identifying active hosts in the target network.

ICON KEY

- Valuable Information
- Test Your Knowledge
- Web Exercise
- Workbook Review

Lab Scenario

As a professional ethical hacker or pen tester, you should be able to scan and detect the active network systems/devices in the target network. During the network scanning phase of security assessment, your first task is to scan the network systems/devices connected to the target network within a specified IP range and check for live systems in the target network.

Lab Objectives

- Perform host discovery using Nmap
- Perform host discovery using Angry IP Scanner

Lab Environment

Tools demonstrated in this lab are available in E:\CEH-Tools\CEHv11 Module 03 Scanning Networks

To carry out this lab, you need:

- Windows 10 virtual machine
- Windows Server 2016 virtual machine
- Ubuntu virtual machine
- Parrot Security virtual machine
- Web browsers with an Internet connection
- Administrator privileges to run the tools
- Nmap located at **E:\CEH-Tools\CEHv11 Module 03 Scanning Networks\Scanning Tools\Nmap**
- Angry IP Scanner located at **E:\CEH-Tools\CEHv11 Module 03 Scanning Networks\Ping Sweep Tools\Angry IP Scanner**
- You can also download the latest version of the above-mentioned tools from their official websites. If you decide to download the latest version, the screenshots shown in the lab might differ.

Lab Duration

Time: 10 Minutes

Overview of Host Discovery

Host discovery is considered the primary task in the network scanning process. It is used to discover the active/live hosts in a network. It provides an accurate status of the systems in the network, which, in turn, reduces the time spent on scanning every port on every system in a sea of IP addresses in order to identify whether the target host is up.

The following are examples of host discovery techniques:

- ARP ping scan
- UDP ping scan
  - ICMP ping scan (ICMP ECHO ping, ICMP timestamp, ping ICMP, and address mask ping)
  - TCP ping scan (TCP SYN ping and TCP ACK ping)
- IP protocol scan

Lab Tasks

TASK 1

Perform Host Discovery using Nmap

Nmap is a utility used for network discovery, network administration, and security auditing. It is also used to perform tasks such as network inventory, managing service upgrade schedules, and monitoring host or service uptime

Here, we will use Nmap to discover a list of live hosts in the target network. We can use Nmap to scan the active hosts in the target network using various host discovery techniques such as ARP ping scan, UDP ping scan, ICMP ECHO ping scan, ICMP ECHO ping sweep, etc.

1. Turn on one or more virtual machines. In this lab task, we have used the **Windows 10**, **Windows Server 2016**, and **Parrot Security** virtual machines.

2. In the **Windows 10** virtual machine, log in with the credentials Username: **Admin** and Password: **Pa$$w0rd**.

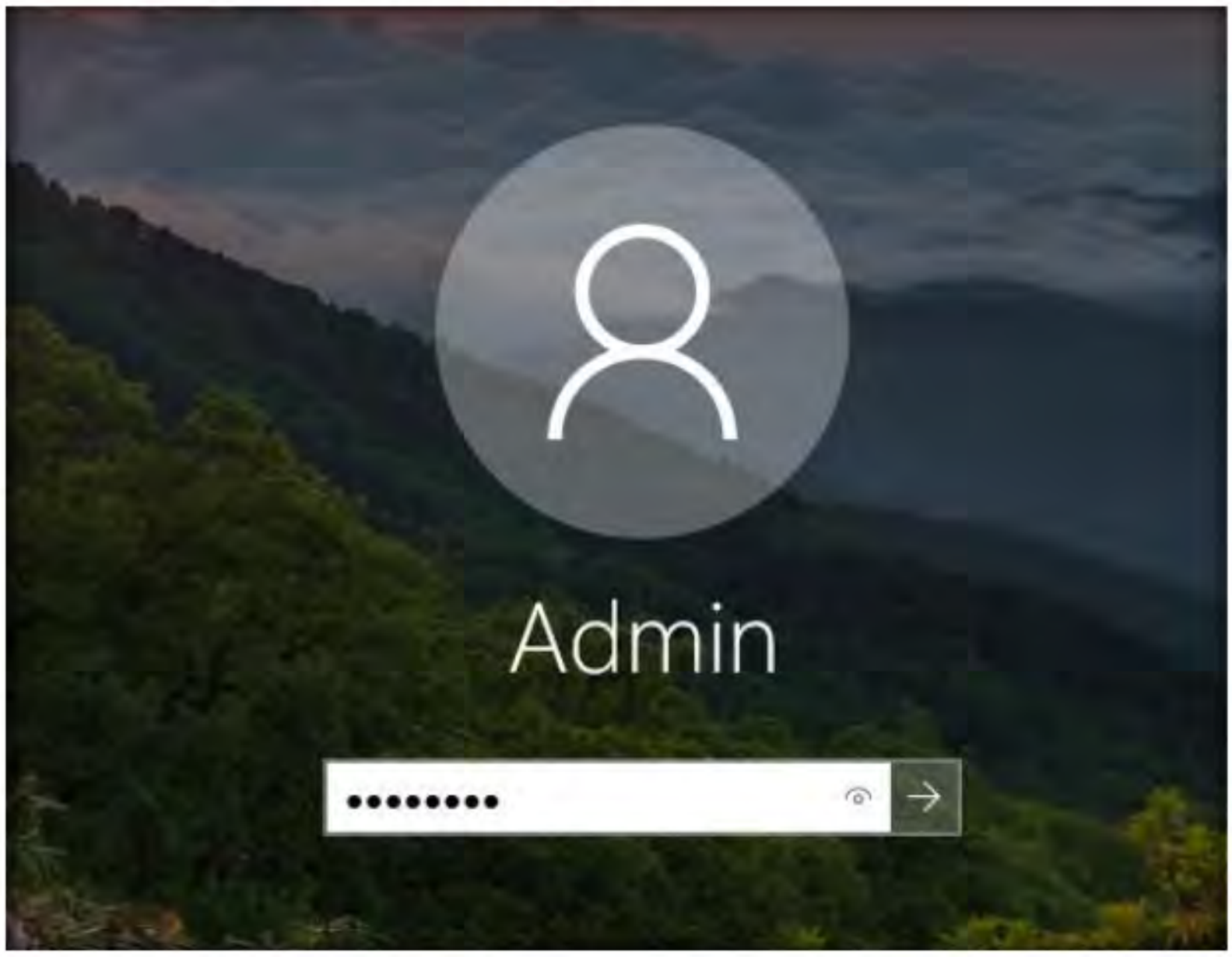


Figure 1.1.1: Login window

TASK 1.1

Install Nmap

3. Navigate to **E:\CEH-Tools\CEHv11 Module 03 Scanning Networks\Scanning Tools\Nmap**; then, double-click **nmap-7.80-setup.exe**.

Note: If a **User Account Control** window appears, click **Yes**.

Note: If an **Open File - Security Warning** pop-up appears, click **Run**.

4. In the **Nmap Setup** window, click **I Agree** and follow the installation steps to install Nmap using all default settings.

Note: If an **Npcap Setup** window appears, click **Install** and follow the installation steps. After the completion of the installation, click **Finish**.

5. In the **Nmap Setup** window, follow the wizard-driven steps to install Nmap and click **Finish**.

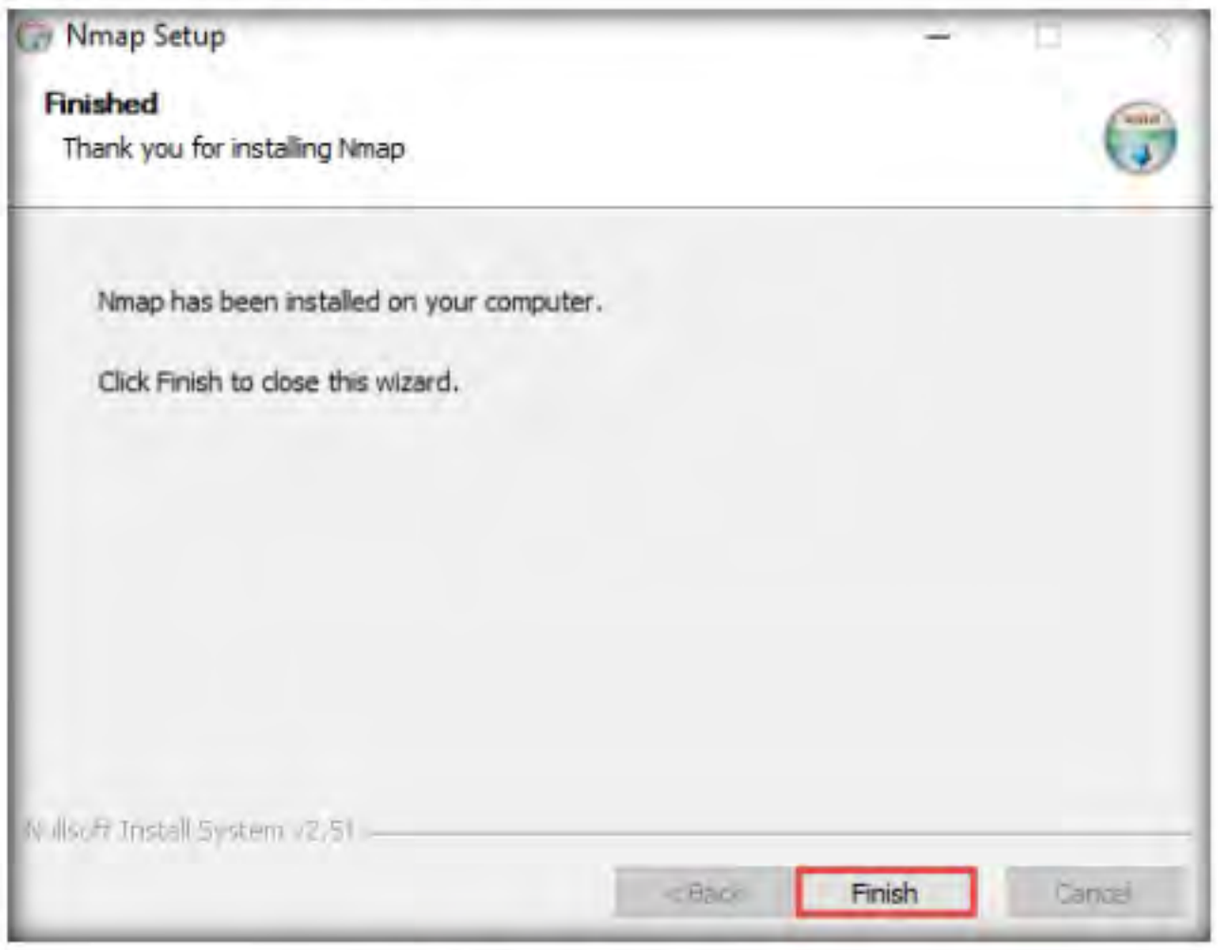


Figure 1.1.2: Nmap Setup pop-up wizard

6. On the completion of the installation, click on the **Start** menu on the bottom-left corner of **Desktop** and launch **Nmap - Zenmap GUI** from the applications, as shown in the screenshot.

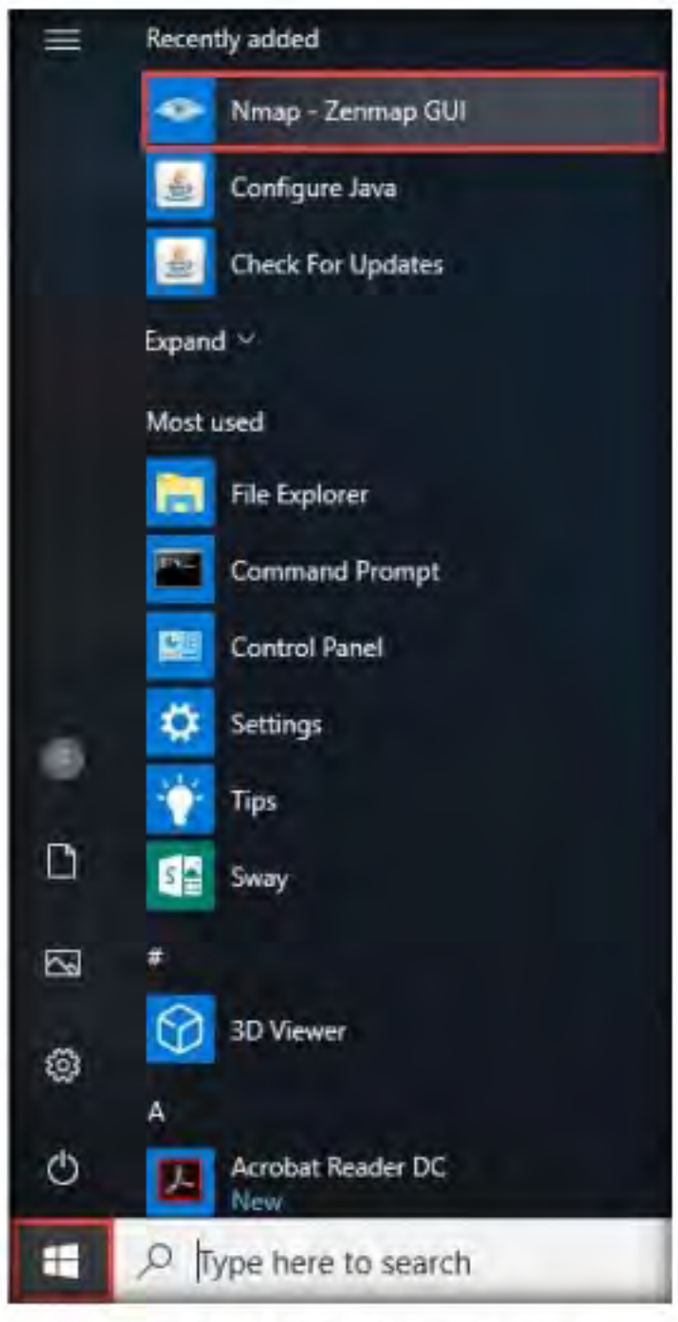


Figure 1.1.3: Launch Nmap - Zenmap

TASK 1.2

Perform ARP Ping Scan

7. The **Nmap - Zenmap** GUI appears; in the **Command** field, type the command **nmap -sn -PR <Target IP Address>** (here, the target IP address is **10.10.10.16**) and click **Scan**.

Note: **-sn**: disables port scan and **-PR**: performs ARP ping scan.

8. The scan results appear, indicating that the target **Host is up**, as shown in the screenshot.

Note: In this lab, we are targeting the **Windows Server 2016** (**10.10.10.16**) virtual machine.

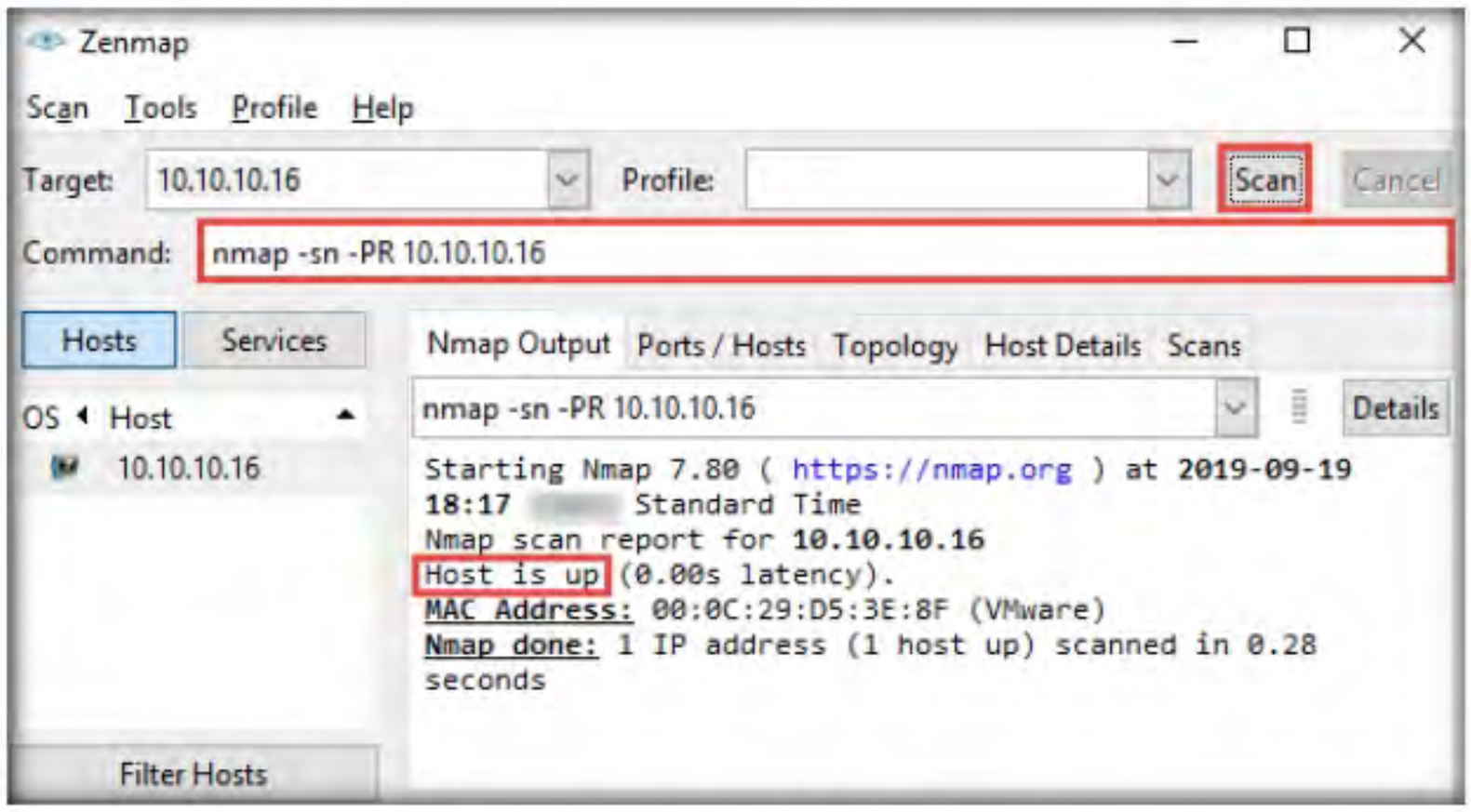


Figure 1.1.4: Zenmap scan results for ARP ping scan

Note: The ARP ping scan probes ARP request to target host; an ARP response means that the host is active.

TASK 1.3

Perform UDP Ping Scan

9. In the **Command** field, type **nmap -sn -PU <Target IP Address>**, (here, the target IP address is **10.10.10.16**) and click **Scan**. The scan results appear, indicating the target **Host is up**, as shown in the screenshot.

Note: **-PU**: performs the UDP ping scan.

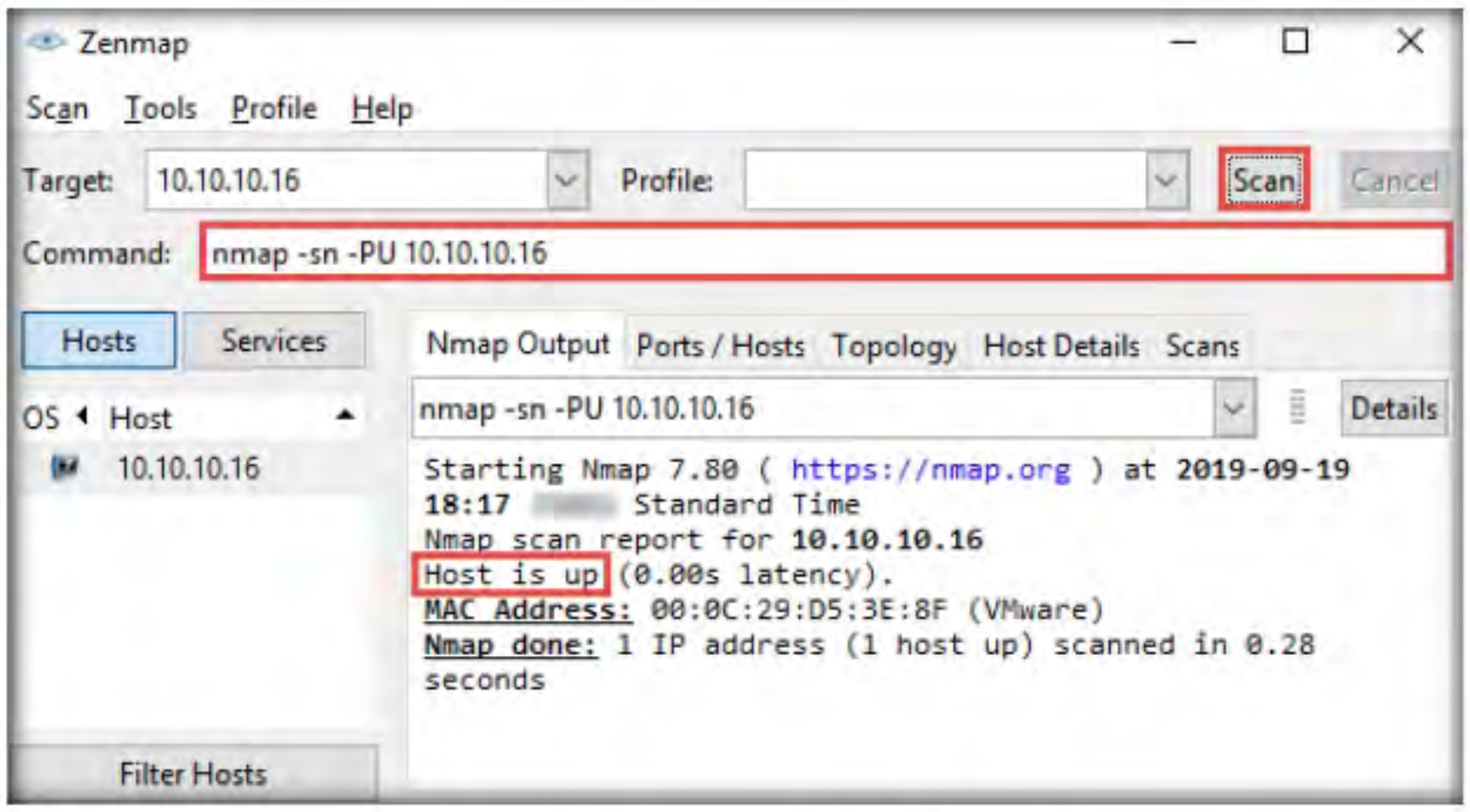


Figure 1.1.5: Zenmap scan results for UDP ping scan

Note: The UDP ping scan sends UDP packets to the target host; a UDP response means that the host is active. If the target host is offline or unreachable, various error messages such as "host/network unreachable" or "TTL exceeded" could be returned.

TASK 1.4

Perform ICMP ECHO Ping Scan

10. Now, we will perform the ICMP ECHO ping scan. In the **Command** field, type **nmap -sn -PE <Target IP Address>**, (here, the target IP address is **10.10.10.16**) and click **Scan**. The scan results appear, indicating that the target **Host is up**, as shown in the screenshot.

Note: **-PE**: performs the ICMP ECHO ping scan.

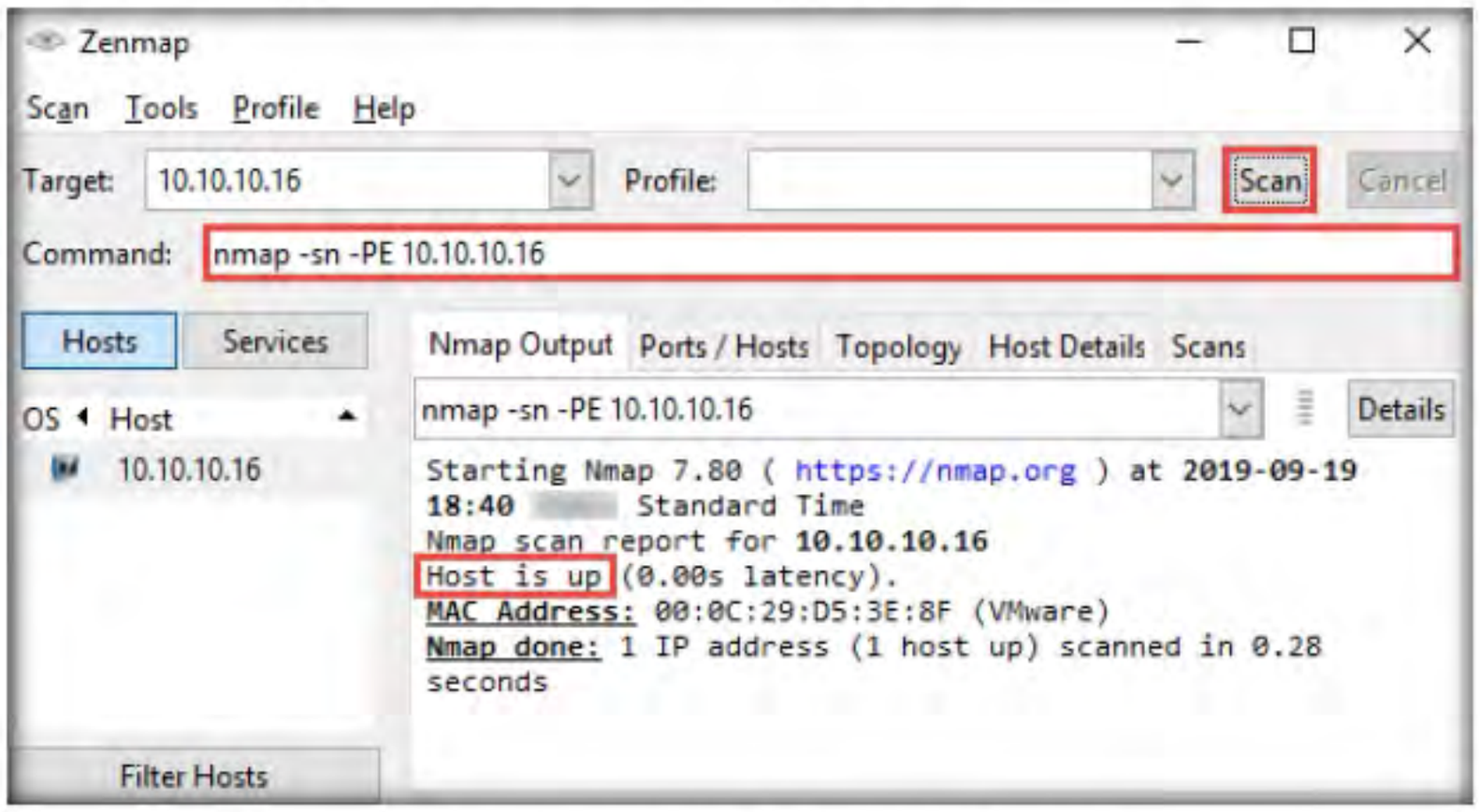


Figure 1.1.6: Zenmap scan results for ICMP ECHO ping scan

Note: The ICMP ECHO ping scan involves sending ICMP ECHO requests to a host. If the target host is alive, it will return an ICMP ECHO reply. This scan is useful for locating active devices or determining if the ICMP is passing through a firewall.

TASK 1.5

Perform ICMP ECHO Ping Sweep

11. Now, we will perform an ICMP ECHO ping sweep to discover live hosts from a range of target IP addresses. In the **Command** field, type **nmap -sn -PE <Target Range of IP Addresses>** (here, the target range of IP addresses is **10.10.10.11-20**) and click **Scan**. The scan results appear, indicating the target **Host is up**, as shown in the screenshot

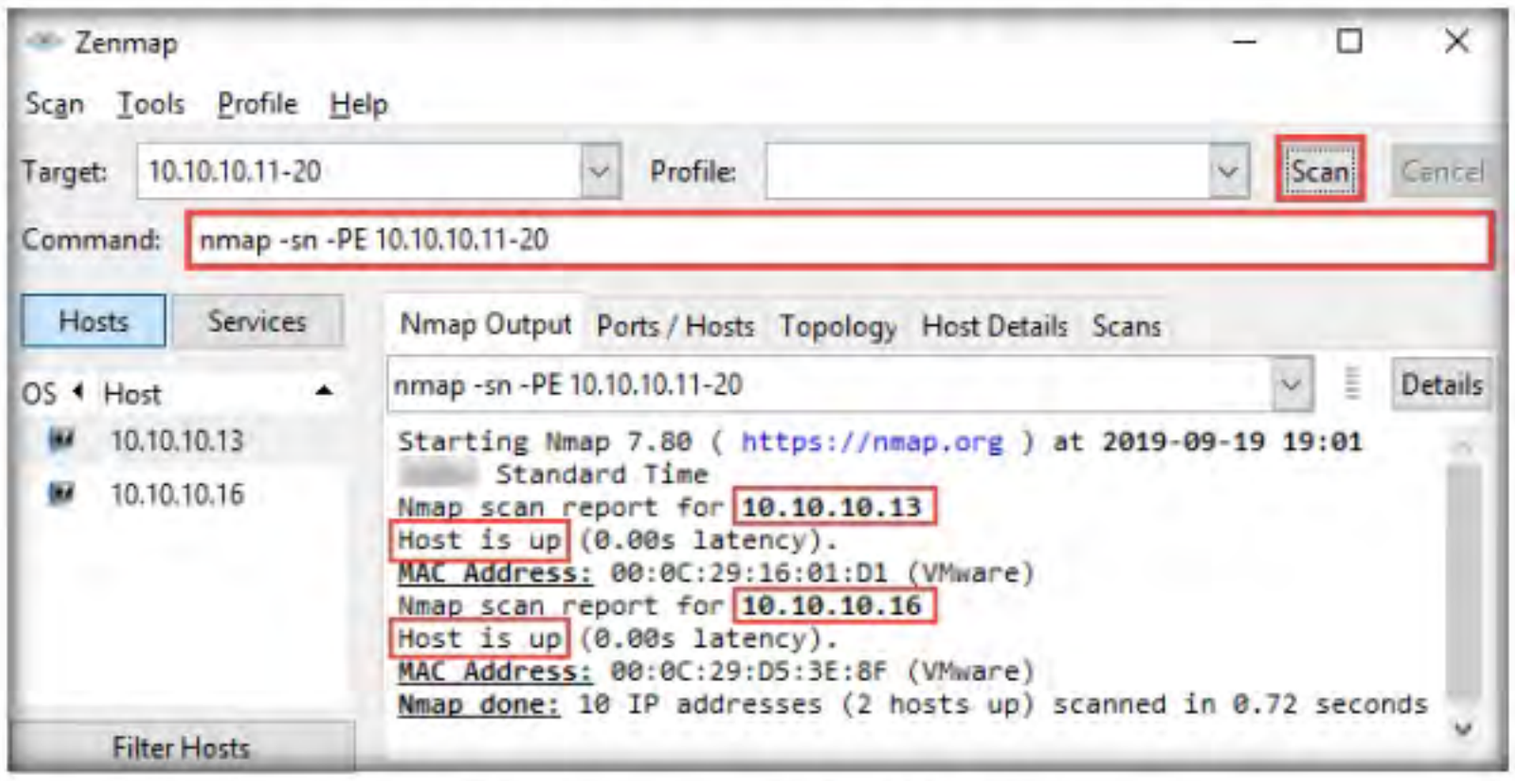


Figure 1.1.7: Zenmap scan results for ICMP ECHO ping sweep scan

Note: The ICMP ECHO ping sweep is used to determine the live hosts from a range of IP addresses by sending ICMP ECHO requests to multiple hosts. If a host is alive, it will return an ICMP ECHO reply.

12. Apart from the aforementioned network scanning techniques, you can also use the following scanning techniques to perform a host discovery on a target network.

- **ICMP Timestamp and Address Mask Ping Scan**: These techniques are alternatives for the traditional ICMP ECHO ping scan, which are used to determine whether the target host is live specifically when administrators block the ICMP ECHO pings.

  Example –

  ICMP timestamp ping scan

  # nmap -sn -PP <target IP address>

  ICMP address mask ping scan

  # nmap -sn -PM <target IP address>

- **TCP SYN Ping Scan**: This technique sends empty TCP SYN packets to the target host, ACK response means that the host is active.

  # nmap -sn -PS <target IP address>

- **TCP ACK Ping Scan**: This technique sends empty TCP ACK packets to the target host; an RST response means that the host is active.

  # nmap -sn -PA <target IP address>

- **IP Protocol Ping Scan**: This technique sends different probe packets of different IP protocols to the target host, any response from any probe indicates that a host is active.

  # nmap -sn -PO <target IP address>

13. This concludes the demonstration of discovering the target host(s) in the target network using various host discovery techniques.
14. Close all open windows and document all the acquired information.

TASK 2 Perform Host Discovery using Angry IP Scanner

Here, we will use the Angry IP Scanner tool to discover the active hosts in the target network.

1. Turn on the **Ubuntu** virtual machine.

Note: Ensure that the **Windows 10**, **Windows Server 2016**, and **Parrot Security** virtual machines are turned on.

2. In the **Windows 10** virtual machine, navigate to **E:\CEH-Tools\CEHv11 Module 03 Scanning Networks\Ping Sweep Tools\Angry IP Scanner** and double-click **ipscan-win64-3.6.1.exe**.

TASK 2.1

Install Angry IP Scanner

Angry IP Scanner is an open-source and cross-platform network scanner designed to scan IP addresses as well as ports. It simply pings each IP address to check if it is alive; then, optionally by resolving its hostname, determines the MAC address, scans ports, etc. The amount of gathered data about each host can be extended with plugins

3. Angry IP Scanner starts, and a **Getting Started** window pops up. Click **Next**, follow the wizard, and click **Close**.

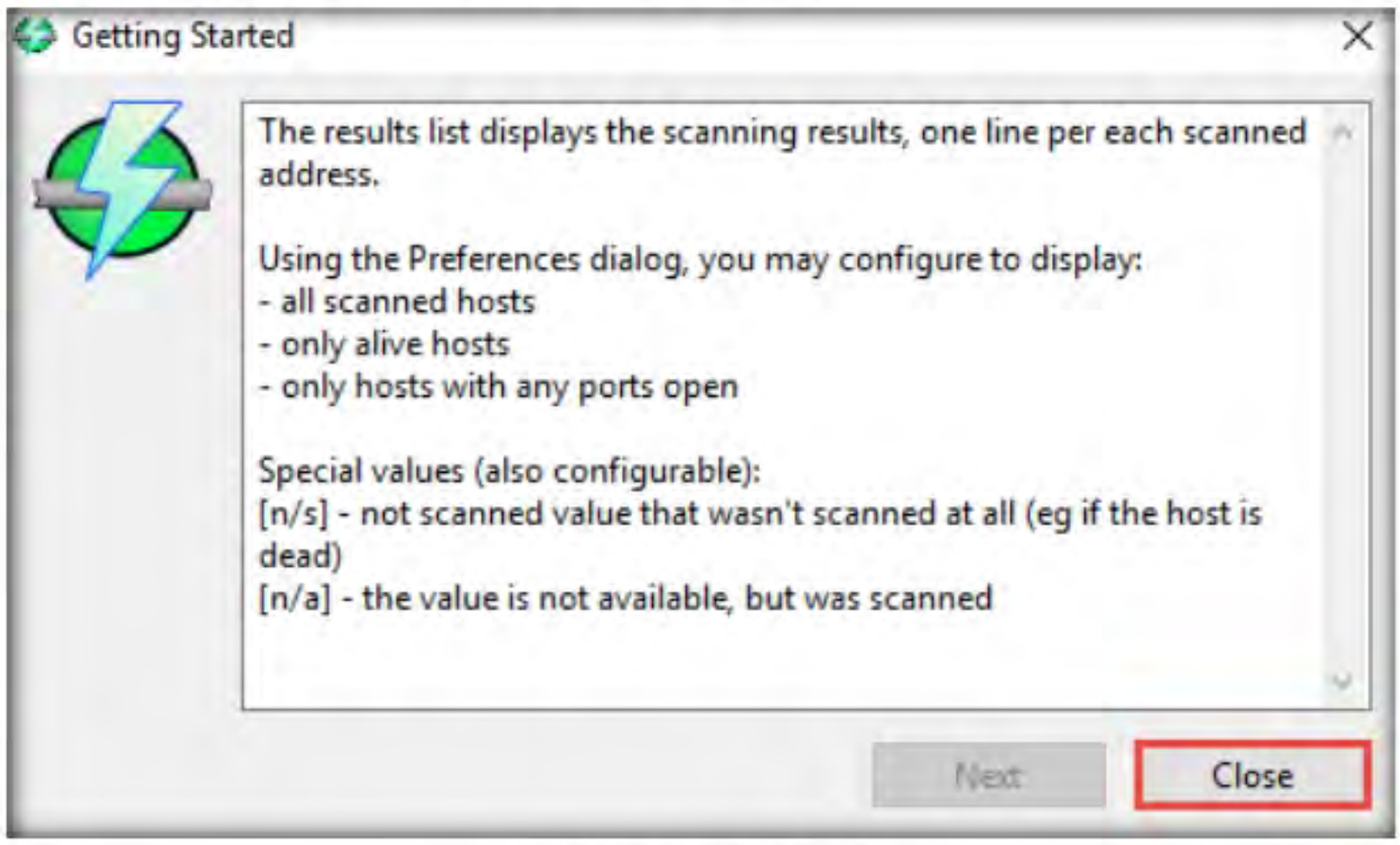


Figure 1.2.1: Getting Started window

4. The **IP Range - Angry IP Scanner** window appears, as shown in the screenshot.

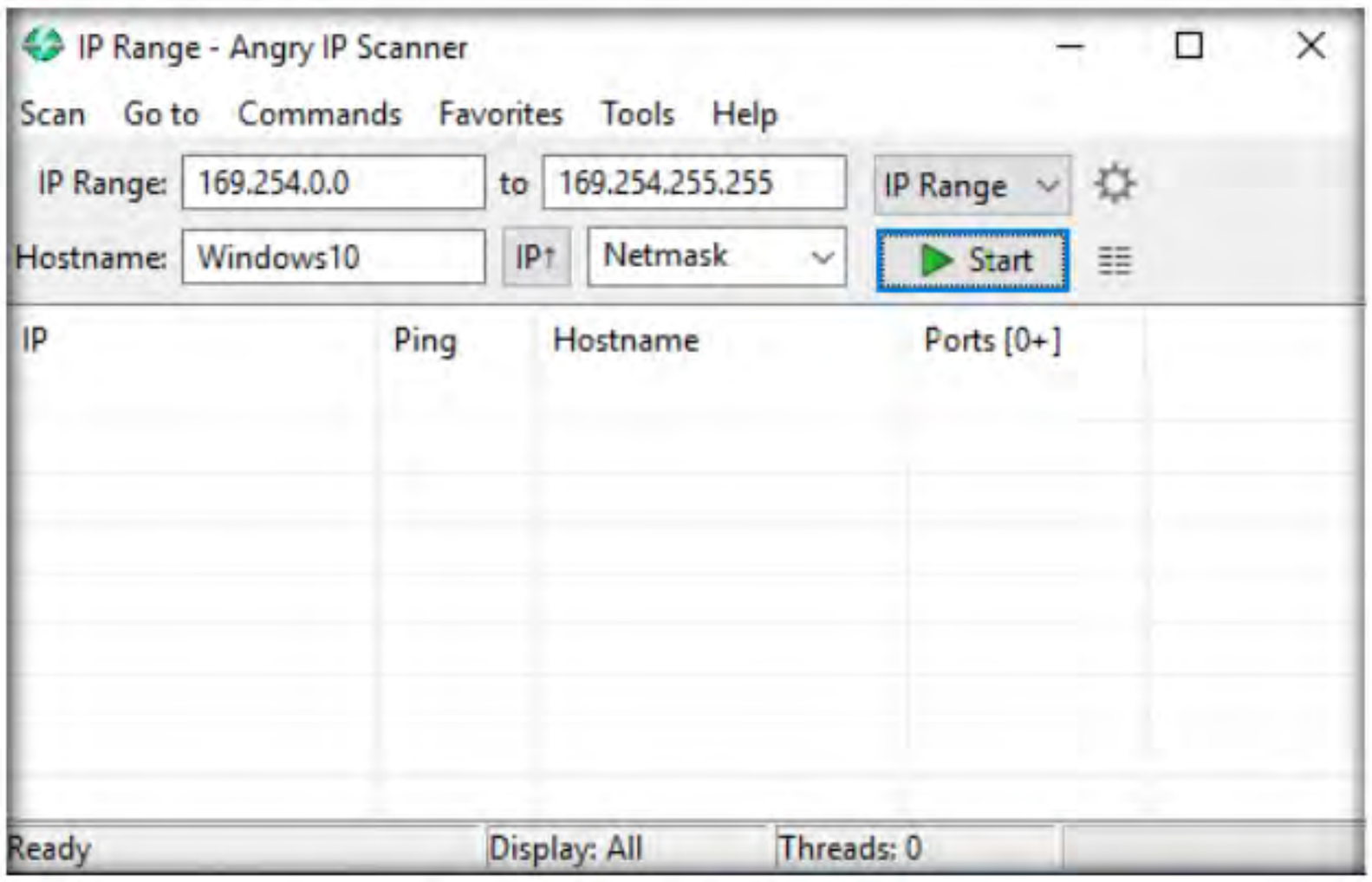


Figure 1.2.2: IP Range - Angry IP Scanner Window

TASK 2.2

Perform the Scan

5. In the **IP Range** fields, type the IP range as **10.10.10.0** to **10.10.10.255** and click the **Preferences** icon beside the **IP Range** menu, as shown in the screenshot.

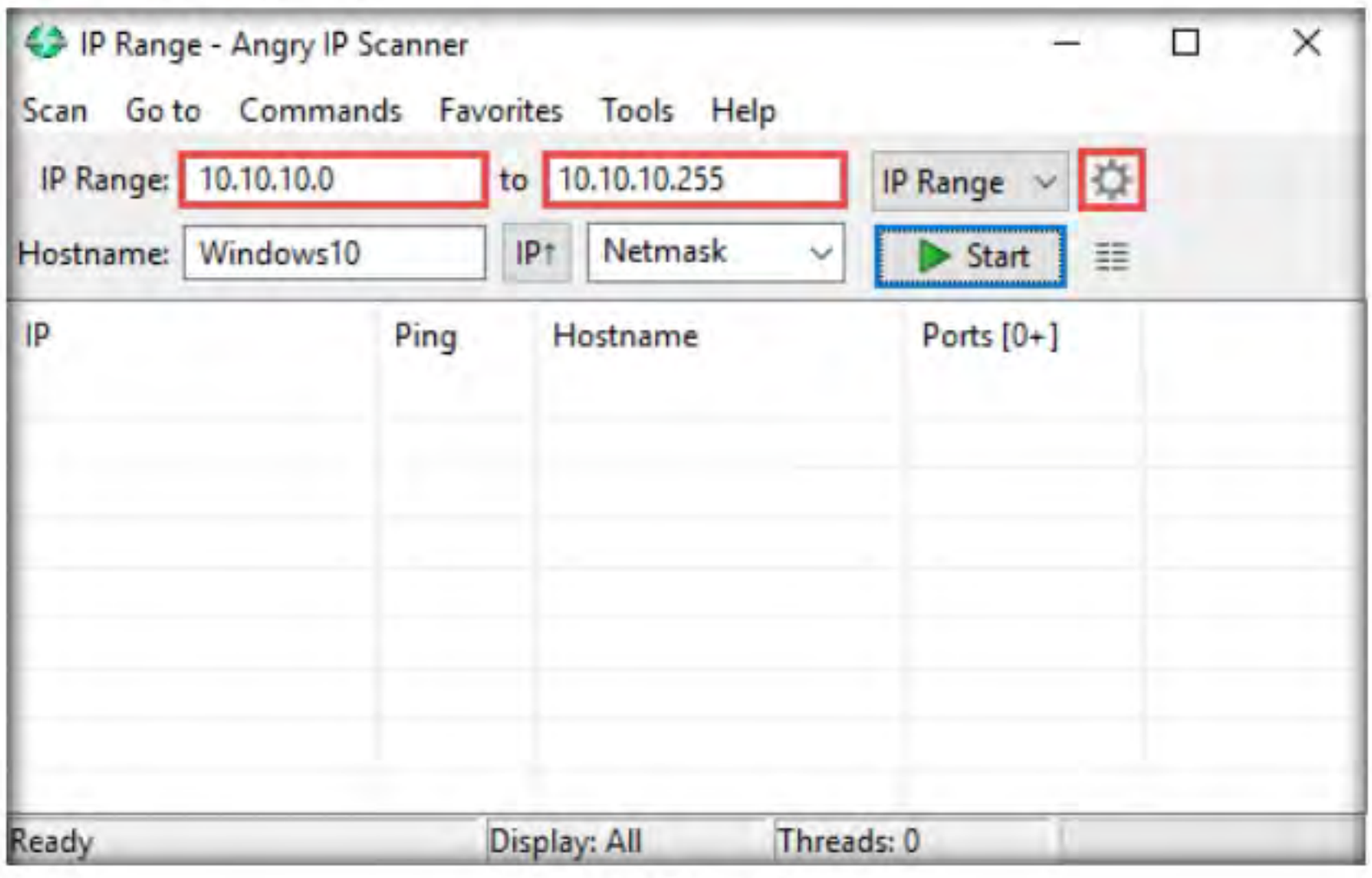


Figure 1.2.3: Filling the IP range

6. The **Preferences** window appears. In the **Scanning** tab, under the **Pinging** section, select the **Pinging method** as **Combined UDP+TCP** from the drop-down list.

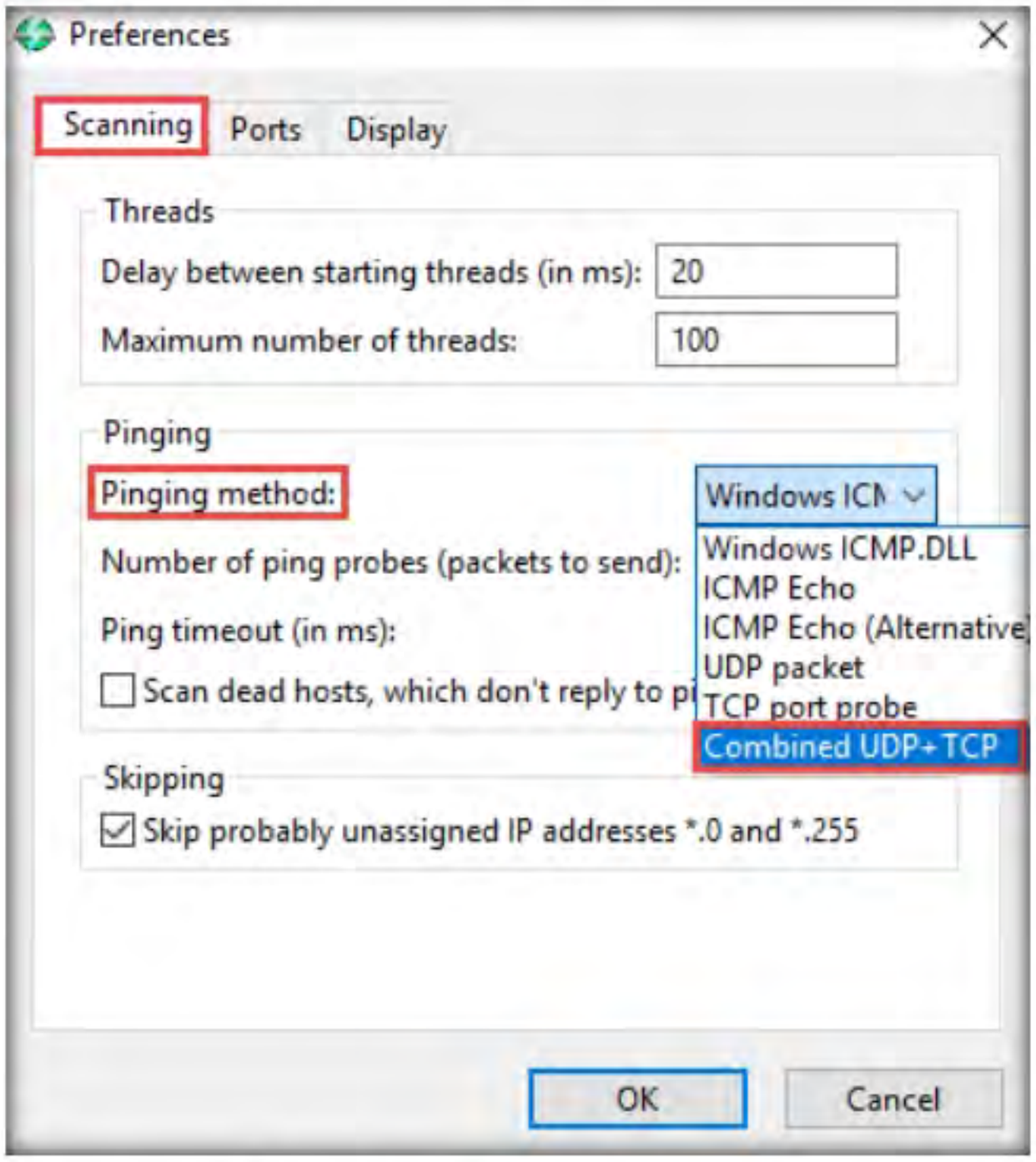


Figure 1.2.4: Angry IP Scanner Preferences window

7. Now, switch to the **Display** tab. Under the **Display in the results list** section, select the **Alive hosts (responding to pings) only** radio button and click **OK**.

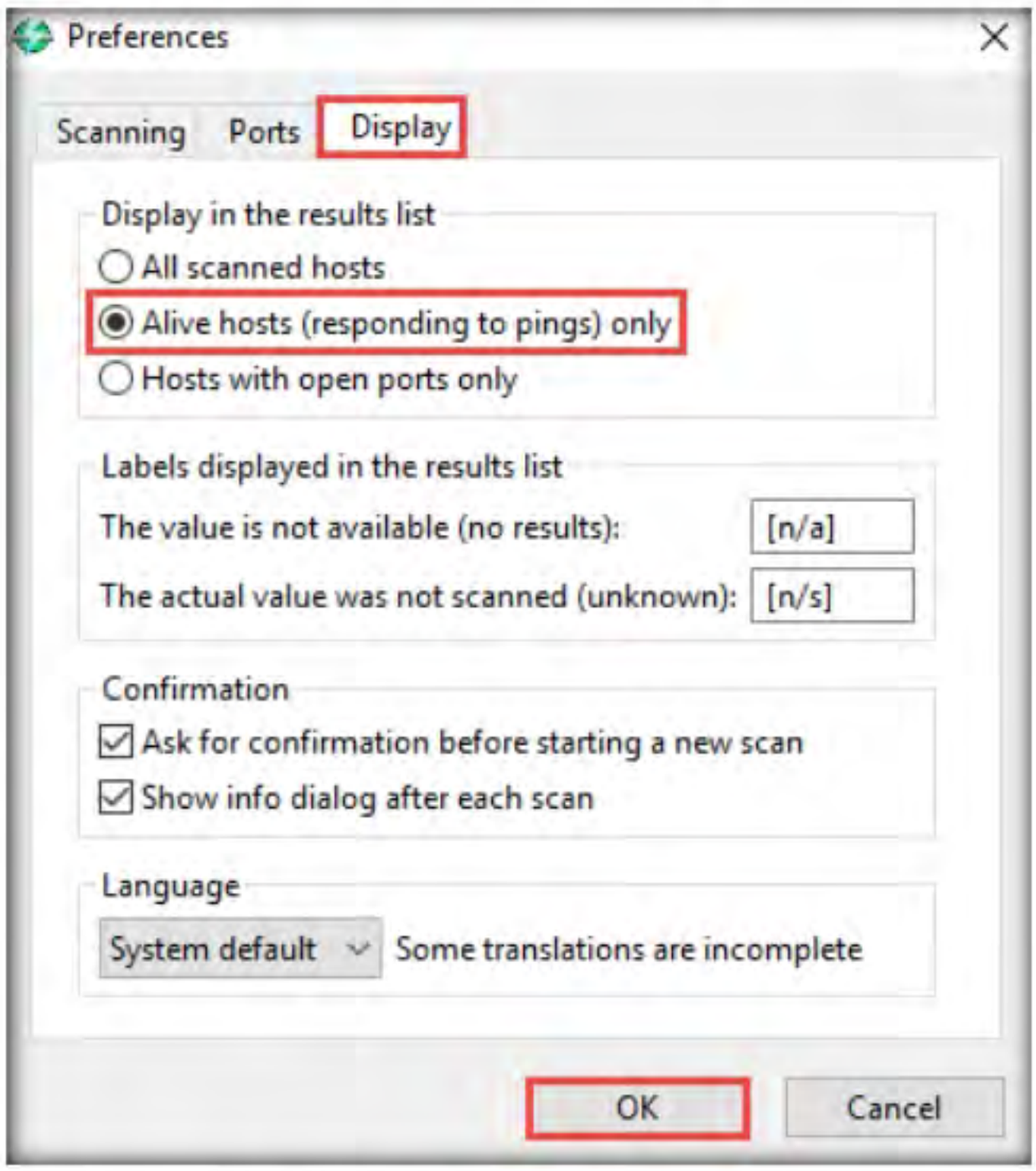


Figure 1.2.5: Display tab options in the preferences window

8. In the **IP Range - Angry IP Scanner** window, click the **Start** button to start scanning the IP range that you entered.

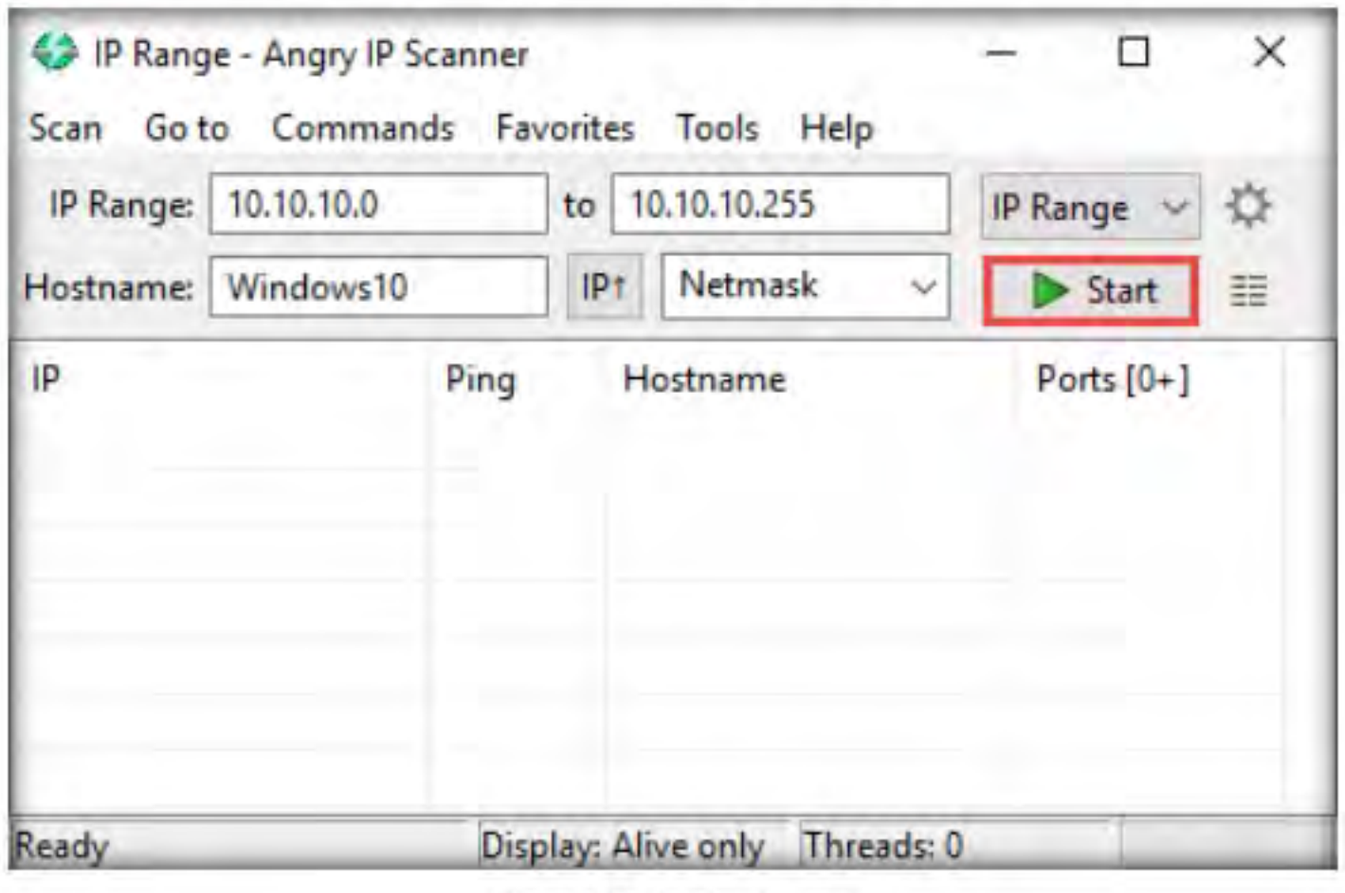


Figure 1.2.6: Starting the scan

9. **Angry IP Scanner** starts scanning the IP range and begins to list out the alive hosts found. Check the progress bar on the bottom-right corner to see the progress of the scanning.

Note: IP addresses may differ in your lab environment.

10. After the scanning is completed, a **Scan Statistics** pop-up appears. Note the total number of **Hosts alive** (here, 6) and click **Close**.

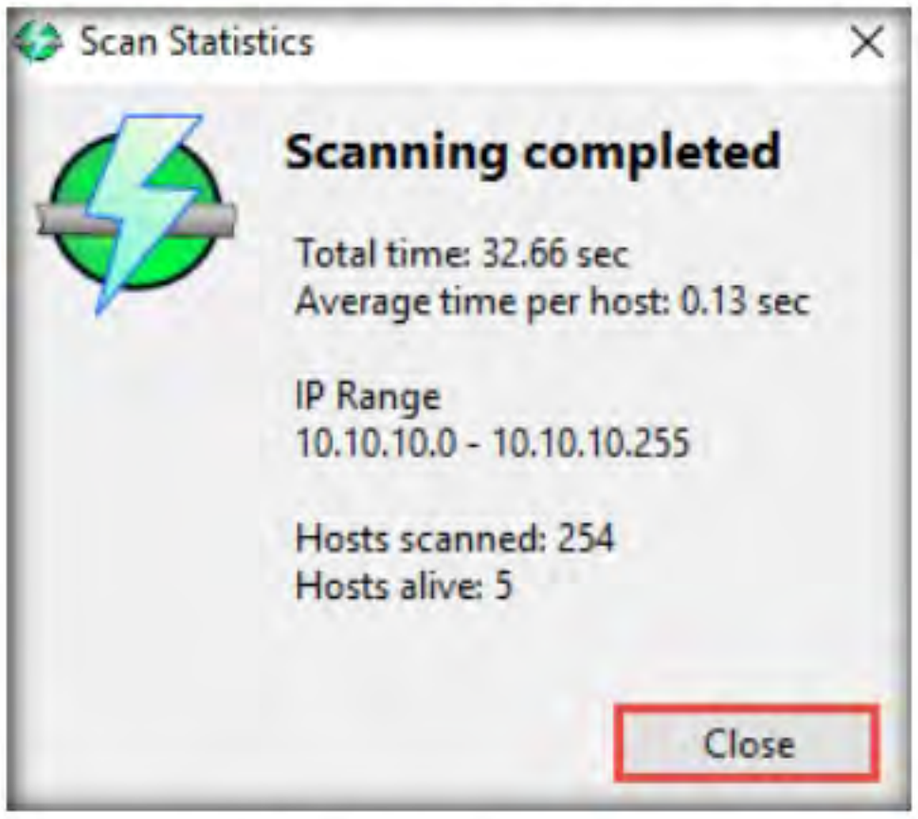


Figure 1.2.7: Scanning Completed prompt

You can also use other ping sweep tools such as **SolarWinds Engineer's Toolset** (https://www.solarwinds.com), **NetScanTools Pro** (https://www.netscantools.com), **Colasoft Ping Tool** (https://www.colasoft.com), **Visual Ping Tester** (http://www.pingtester.net), and **OpUtils** (https://www.manageengine.com) to discover active hosts in the target network

11. The results of the scan appear in the **IP Range - Angry IP Scanner** window. You can see all active IP addresses with their hostnames listed in the main window.

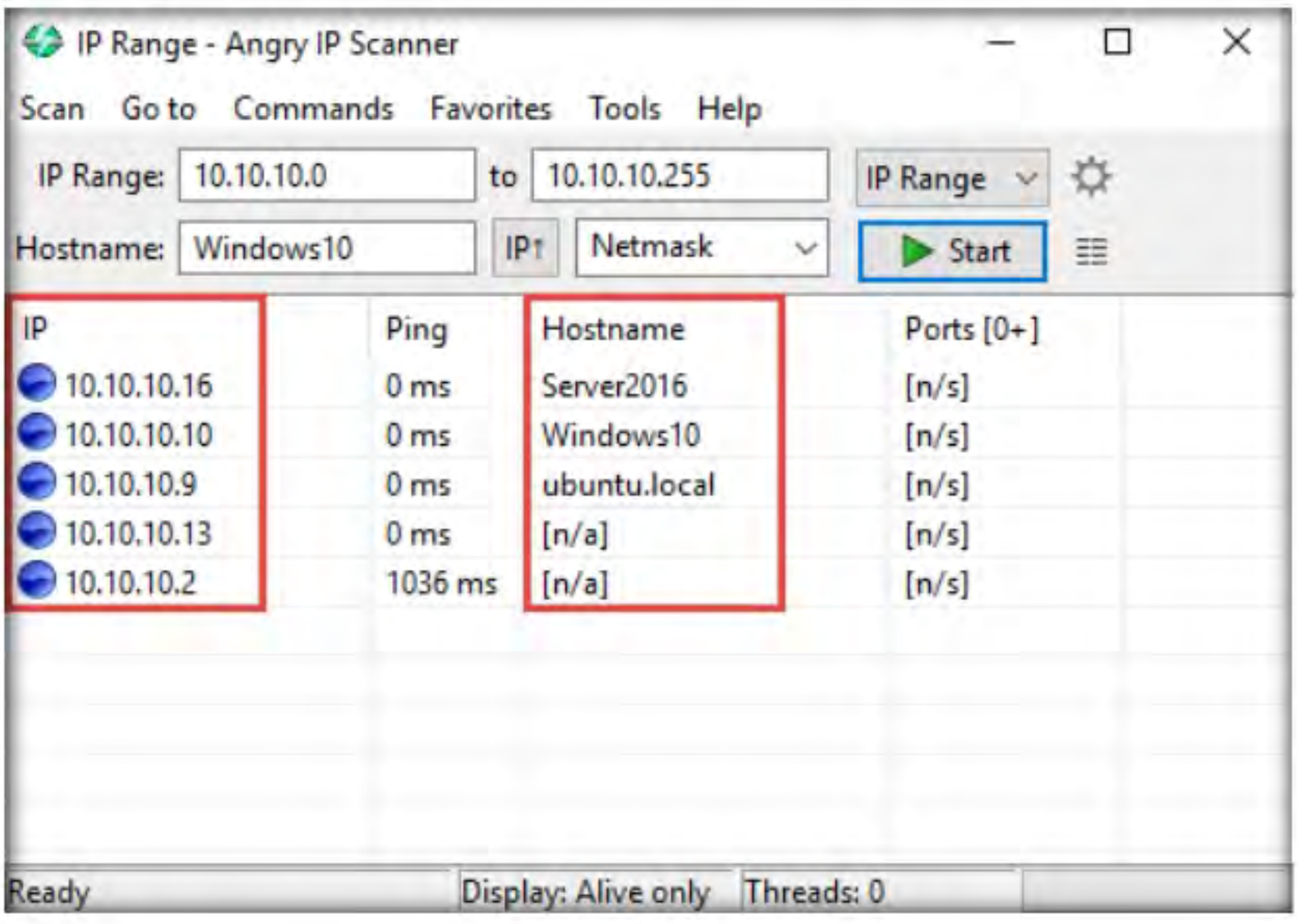


Figure 1.2.8: Scan results

12. This concludes the demonstration of discovering alive hosts in the target range of IP addresses using various Angry IP Scanner.
13. Close all open windows and document all the acquired information.
14. Turn off all the virtual machines.

Lab Analysis

Analyze and document all the results discovered in the lab exercise.

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS ABOUT THIS LAB.

Internet Connection Required	
☐ Yes	☑ No
Platform Supported	
☑ Classroom	☑ iLabs

Lab 2

Perform Port and Service Discovery

Port and service discovery is the process of identifying open ports and services running on the target IP addresses/active hosts.

ICON KEY

- Valuable information
- Test your knowledge
- Web exercise
- Workbook review

Lab Scenario

As a professional ethical hacker or a pen tester, the next step after discovering active hosts in the target network is to scan for open ports and services running on the target IP addresses in the target network. This discovery of open ports and services can be performed by using various port scanning tools and techniques.

Lab Objectives

- Perform port and service discovery using MegaPing
- Perform port and service discovery using NetScanTools Pro
- Explore various network scanning techniques using Nmap
- Explore various network scanning techniques using Hping3

Lab Environment

To carry out this lab, you need:

- Windows 10 virtual machine
- Windows Server 2016 virtual machine
- Parrot Security virtual machine
- Ubuntu virtual machine
- Web browsers with an Internet connection
- Administrator privileges to run the tools
- MegaPing located at **E:\CEH-Tools\CEHv11 Module 03 Scanning Networks\Scanning Tools\MegaPing**

- NetScanTools Pro located at **E:\CEH-Tools\CEHv11 Module 03 Scanning Networks\Scanning Tools\NetScanTools Pro**
- You can also download the latest version of the above-mentioned tools from their official websites. If you decide to download the latest version, the screenshots shown in the lab might differ.

Tools demonstrated in this lab are available in E:\CEH-Tools\CEHv11 Module 03 Scanning Networks

Lab Duration

Time: 40 Minutes

Overview of Port and Service Discovery

Port scanning techniques are categorized according to the type of protocol used for communication within the network.

- **TCP Scanning**
  - Open TCP scanning methods (TCP connect/full open scan)
  - Stealth TCP scanning methods (Half-open Scan, Inverse TCP Flag Scan, ACK flag probe scan, third party and spoofed TCP scanning methods)
- **UDP Scanning**
- **SCTP Scanning**
  - SCTP INIT Scanning
  - SCTP COOKIE/ECHO Scanning
- **SSDP and List Scanning**
- **IPv6 Scanning**

Lab Tasks

TASK 1

Perform Port and Service Discovery using MegaPing

Here, we will use the MegaPing tool to scan for open ports and services running on the target range of IP addresses.

1. Before beginning this lab, turn on the **Windows 10**, **Windows Server 2016**, **Parrot Security**, and **Ubuntu** virtual machines.

TASK 1.1

Install MegaPing

2. In the **Windows 10** virtual machine, log in with the credentials Username: **Admin** and Password: **Pa$$w0rd**.
3. Navigate to **E:\CEH-Tools\CEHv11 Module 03 Scanning Networks\Scanning Tools\MegaPing** and double-click **megaping_setup.exe**.

Note: If a **User Account Control** pop-up appears, click **Yes**.

4. The **MegaPing - InstallShield Wizard** window appears; click **Next** and follow the wizard-driven installation steps to install **MegaPing**.

MegaPing is a toolkit that provides essential utilities for Information System specialists, system administrators, IT solution providers, and individuals. It is used to detect live hosts and open ports of the system in the network, and can scan your entire network and provide information such as open shared resources, open ports, services/drivers active on the computer, key registry entries, users and groups, trusted domains, printers, etc.

5. After the completion of the installation, click on the **Launch the program** checkbox and click **Finish**.

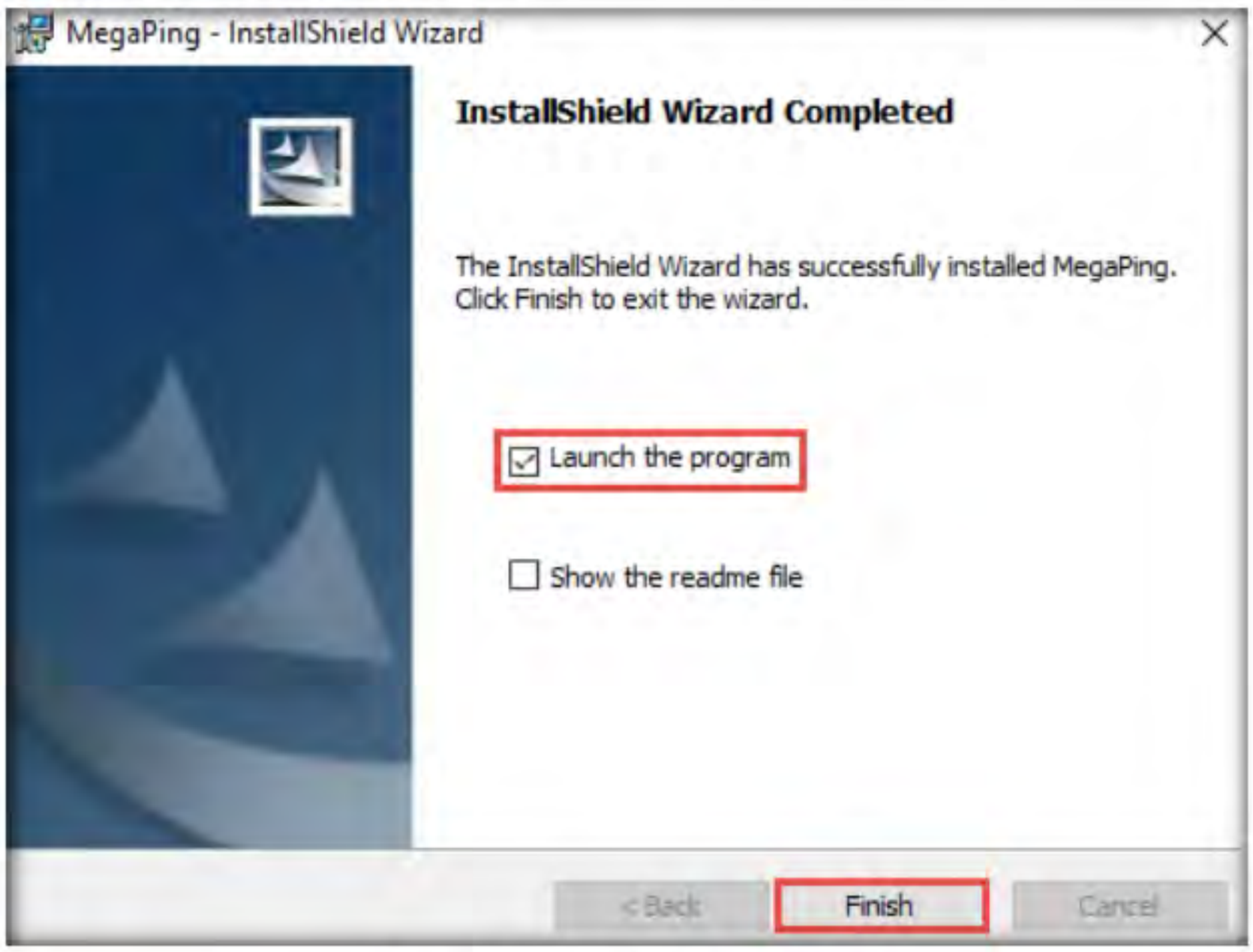


Figure 2.1.1: Launching MegaPing from Start menu

You can also perform various network troubleshooting activities with the help of integrated network utilities such as DNS lookup name, DNS list hosts, Finger, host monitor, IP scanner, NetBIOS scanner, ping, port scanner, share scanner, traceroute, and Whois.

6. The **About MegaPing** window appears; click the **I Agree** button.

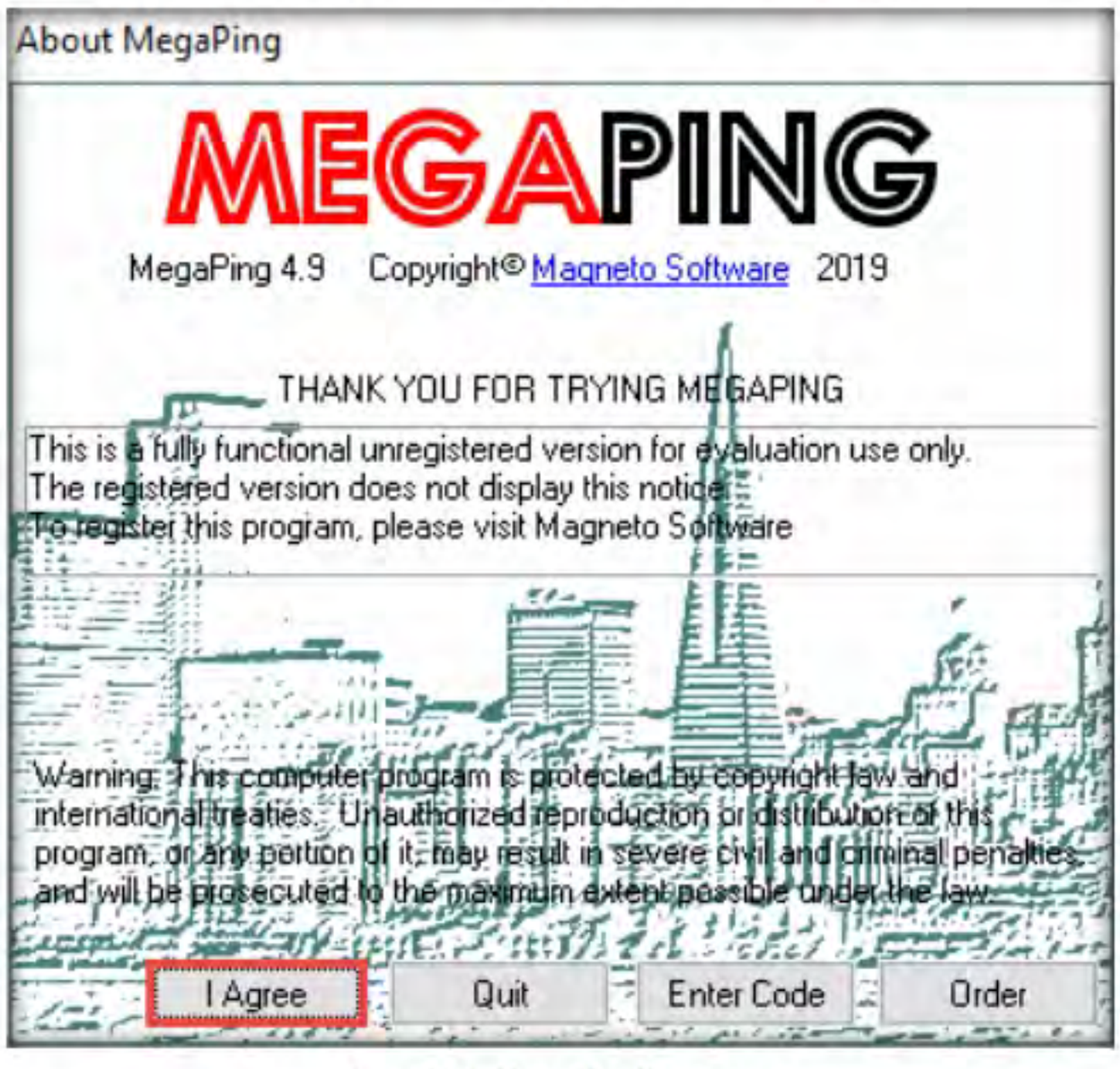


Figure 2.1.2: About MegaPing pop-up

7. The **MegaPing (Unregistered)** GUI appears displaying the **System Info**, as shown in the screenshot.

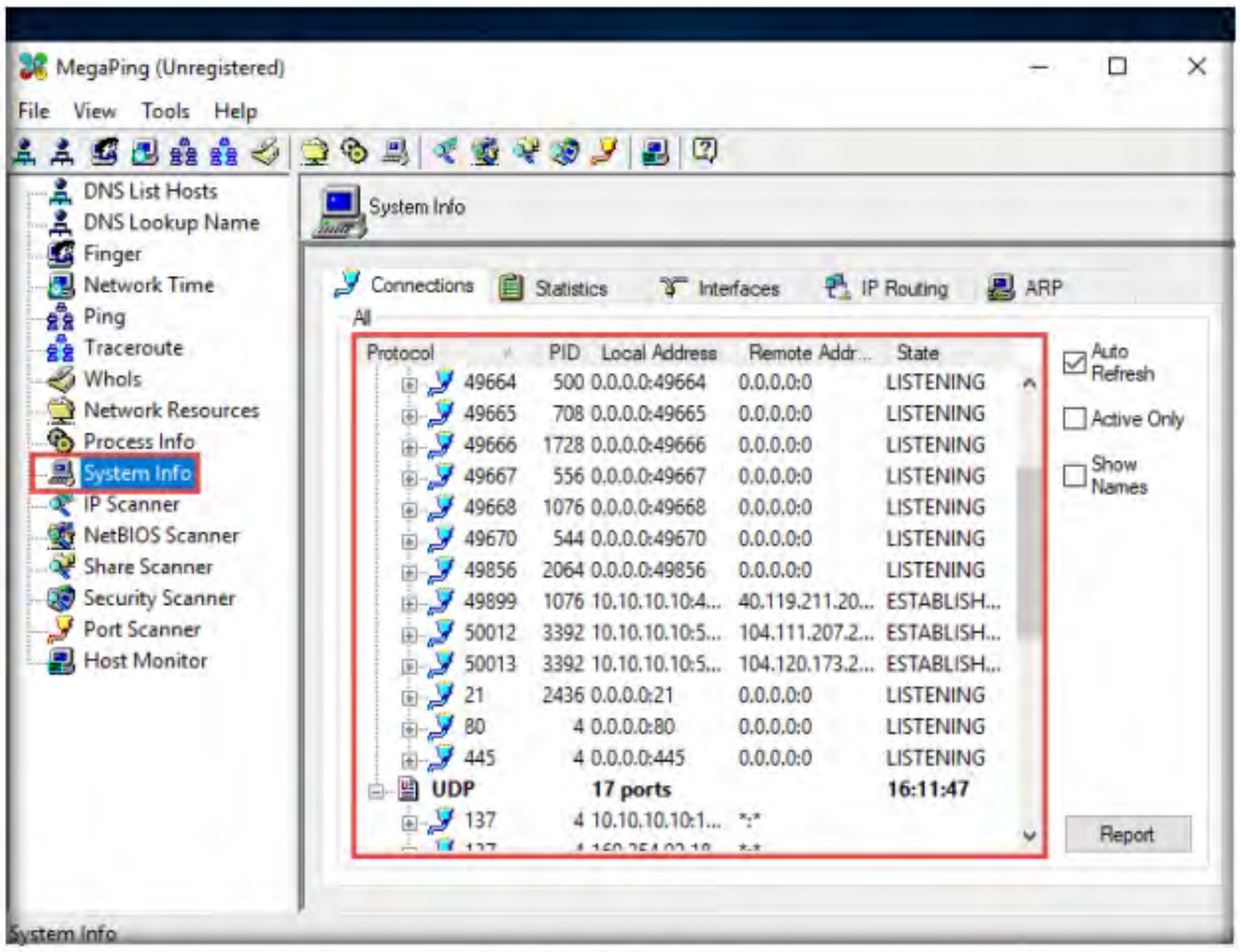


Figure 2.1.3: MegaPing GUI

TASK 1.2

Scan for Live Hosts

8. Select the **IP Scanner** option from the left pane. In the **IP Scanner** tab in the right-hand pane, enter the IP range in the **From** and **To** fields; in this lab, the IP range is **10.10.10.5** to **10.10.10.20**; then, click **Start**.

Note: You may specify the IP range depending on your network.

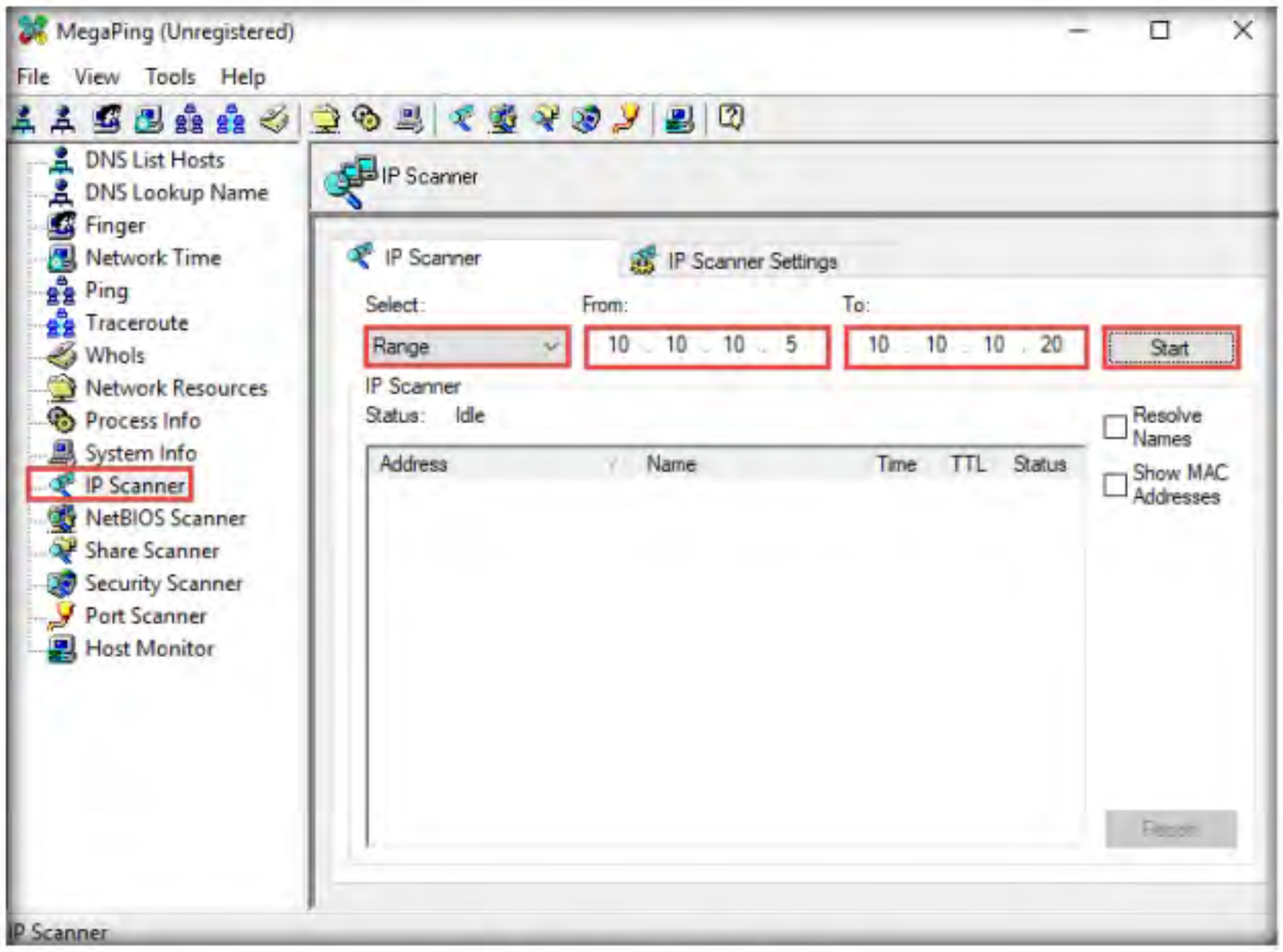


Figure 2.1.4: Configuring MegaPing

9. MegaPing lists all IP addresses under the specified target range with their TTL value, Status (dead or alive), and statistics of the dead and alive hosts, as shown in the screenshot.

Note: The results may vary in your lab environment.

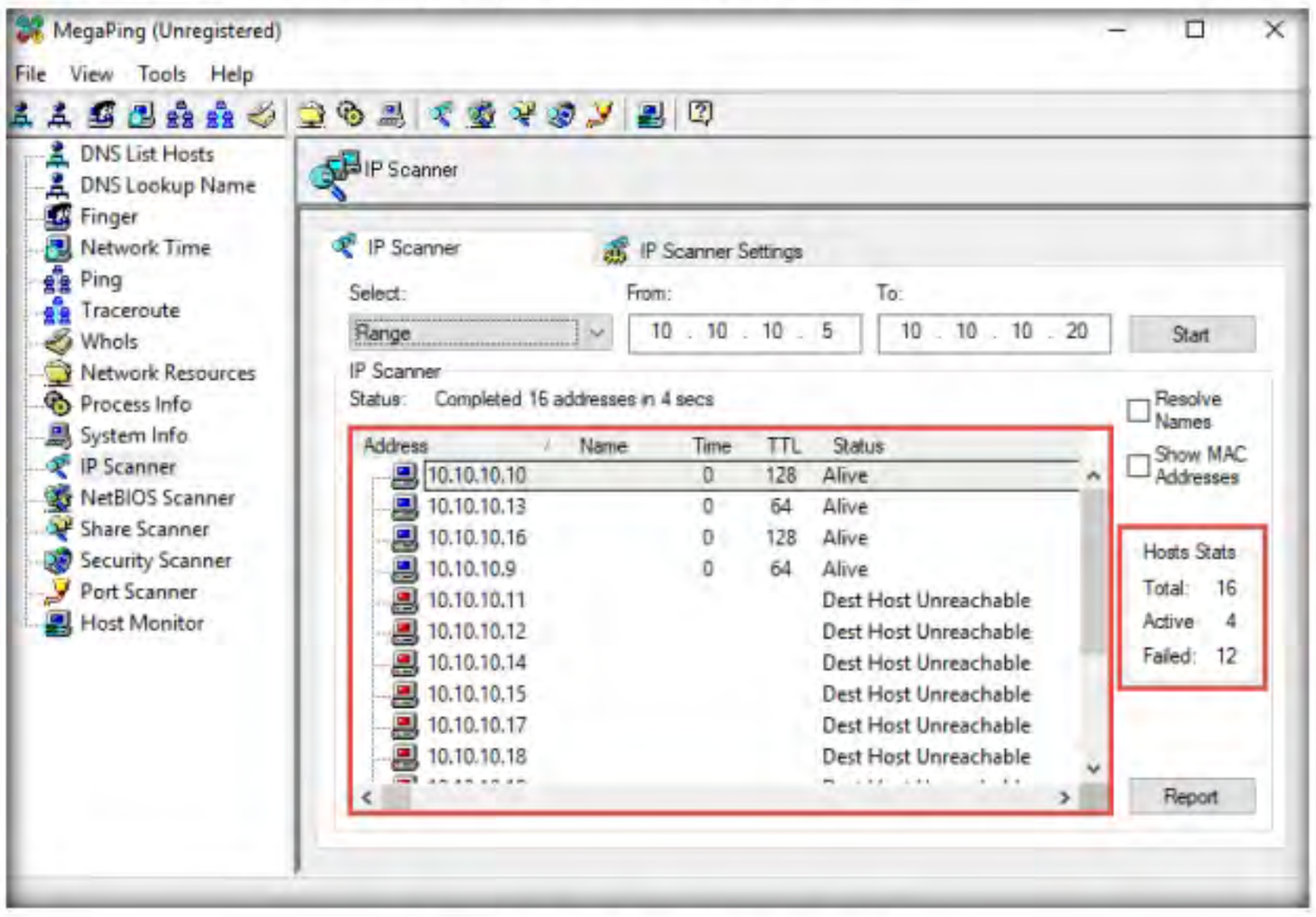


Figure 2.1.5: MegaPing IP Scanning Report

TASK 1.3

Perform Port Scan

10. Select the **Port Scanner** option from the left-hand pane. In the **Port Scanner** tab in the right-hand pane, enter the IP address of the **Windows Server 2016** (**10.10.10.16**) machine into the **Destination Address List** field and click **Add**.

Note: The IP address listed below might vary in your lab environment.

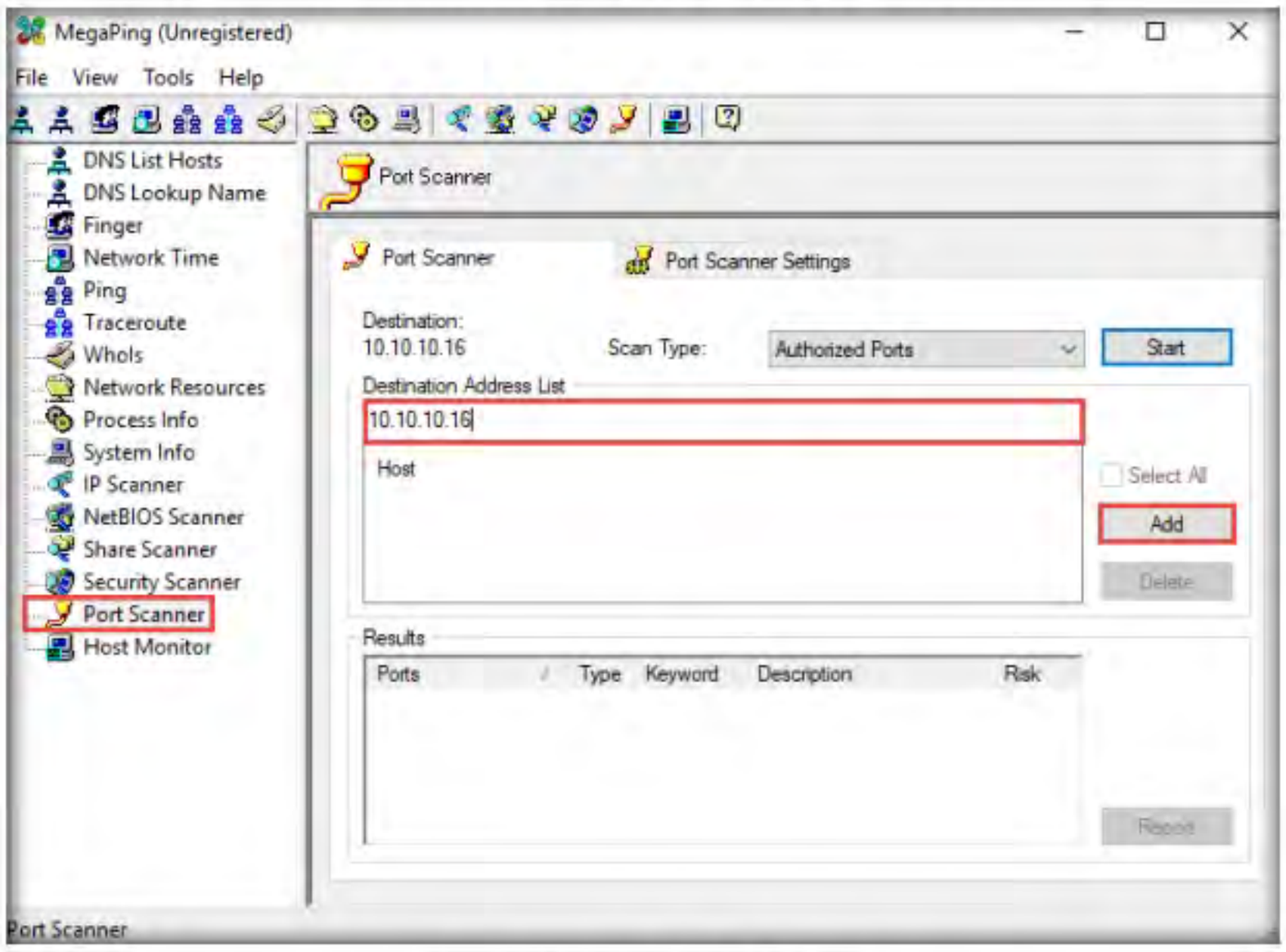


Figure 2.1.6: Adding a host in MegaPing

11. Select the **10.10.10.16** checkbox and click the **Start** button to start listening to the traffic on 10.10.10.16.

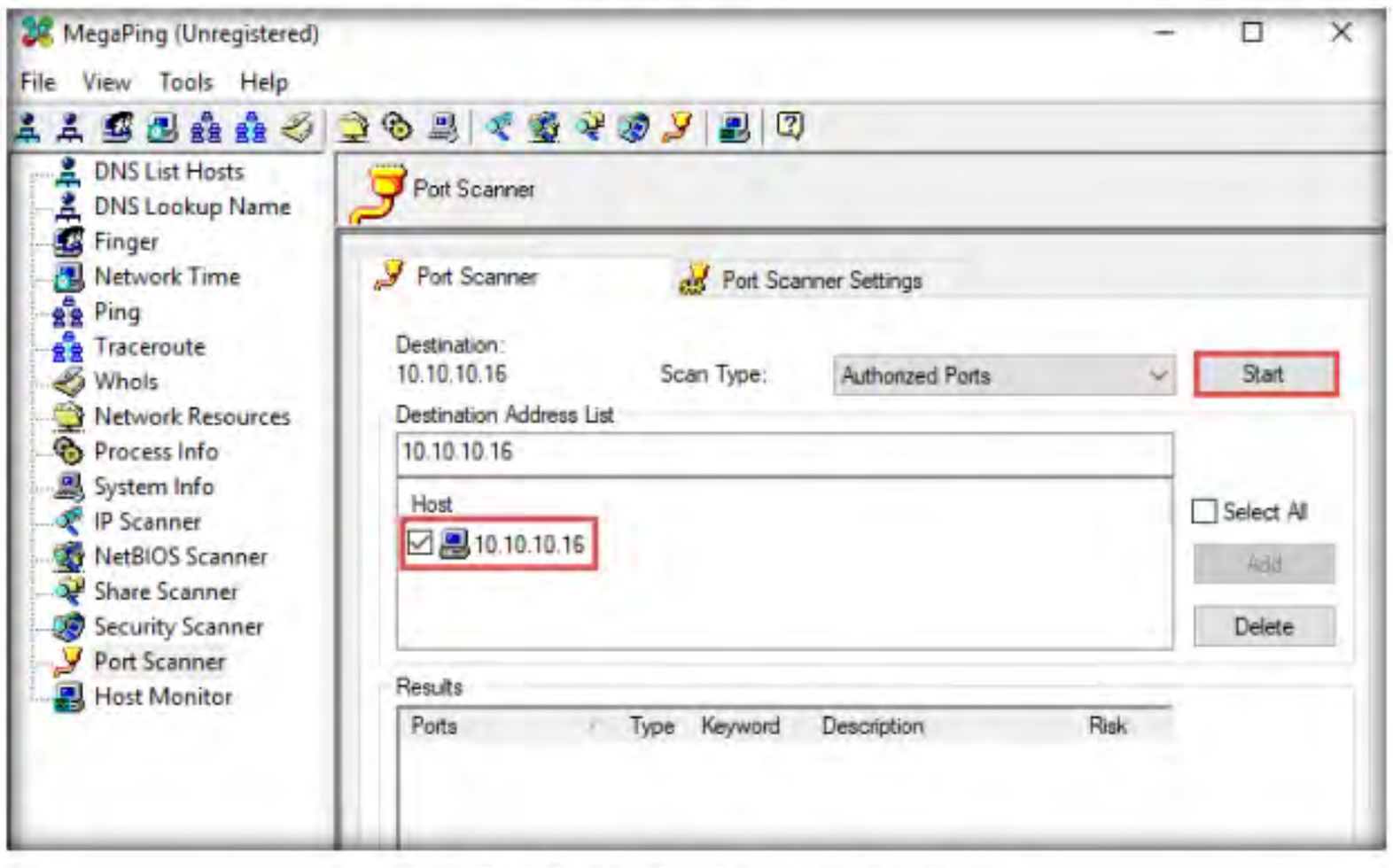


Figure 2.1.7: Starting MegaPing on the selected host

12. MegaPing lists the ports associated with **Windows Server 2016** (**10.10.10.16**), with detailed information on port number and type, service running on the port along with the description, and associated risk, as shown in the screenshot.

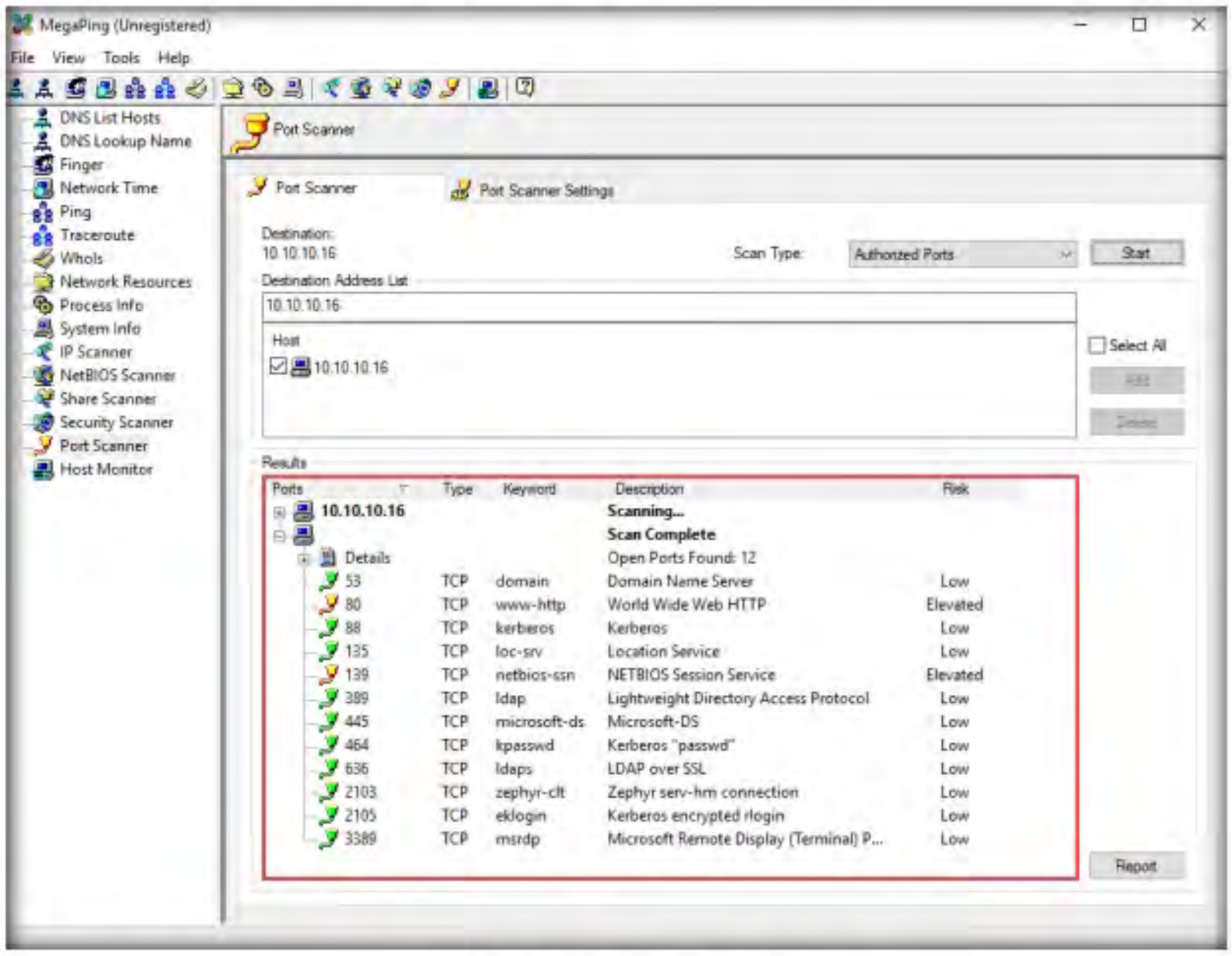


Figure 2.1.8: MegaPing Port Scanning Report

13. Similarly, you can perform port and service scanning on other target machines.
14. This concludes the demonstration of discovering open ports and services running on the target IP address using MegaPing.
15. Close all open windows and document all the acquired information.

TASK 2 Perform Port and Service Discovery using NetScanTools Pro

Here, we will use the NetScanTools Pro tool to discover open ports and services running on the target range of IP addresses.

TASK 2.1 Install NetScanTools Pro

1. In the **Windows 10** virtual machine, navigate to **E:\CEH-Tools\CEHv11 Module 03 Scanning Networks\Scanning Tools\NetScanTools Pro**, and double-click **nstp11demo.exe**.

Note: If a **User Account Control** pop-up appears, click **Yes**.

Note: If an **Open File - Security Warning** pop-up appears, click **Run**.

NetScanTools Pro is an integrated collection of utilities that gathers information on the Internet and troubleshoots networks for Network Professionals. With the available tools, you can research IPv4/IPv6 addresses, hostnames, domain names, e-mail addresses, and URLs on the target network.

2. The **Setup - NetScanTools Pro Demo** window appears; click **Next** and follow the wizard-driven installation steps to install **NetScanTools Pro**.

Note: If the **WinPcap Setup** window appears, click **Next** and install it.

3. After the completion of the installation, click **Finish**.

Figure 2.2.1: NetScanTools Setup window

4. The **Reminder** window appears; if you are using a demo version of NetScanTools Pro, click the **Start the DEMO** button.

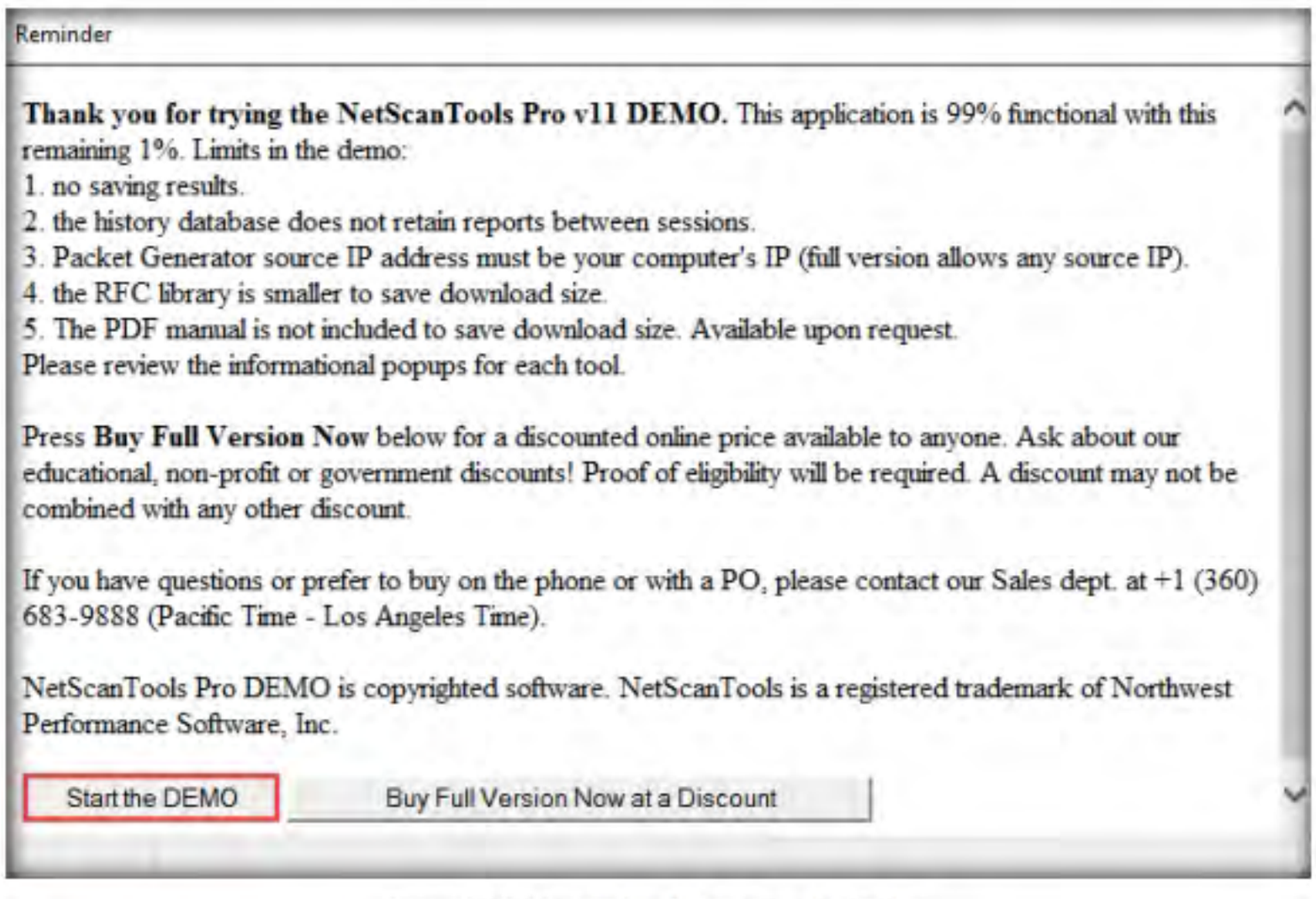


Figure 2.2.2: NetScan Tools Pro reminder window

5. A **DEMO Version** pop-up appears; click the **Start NetScanTools Pro Demo...** button.

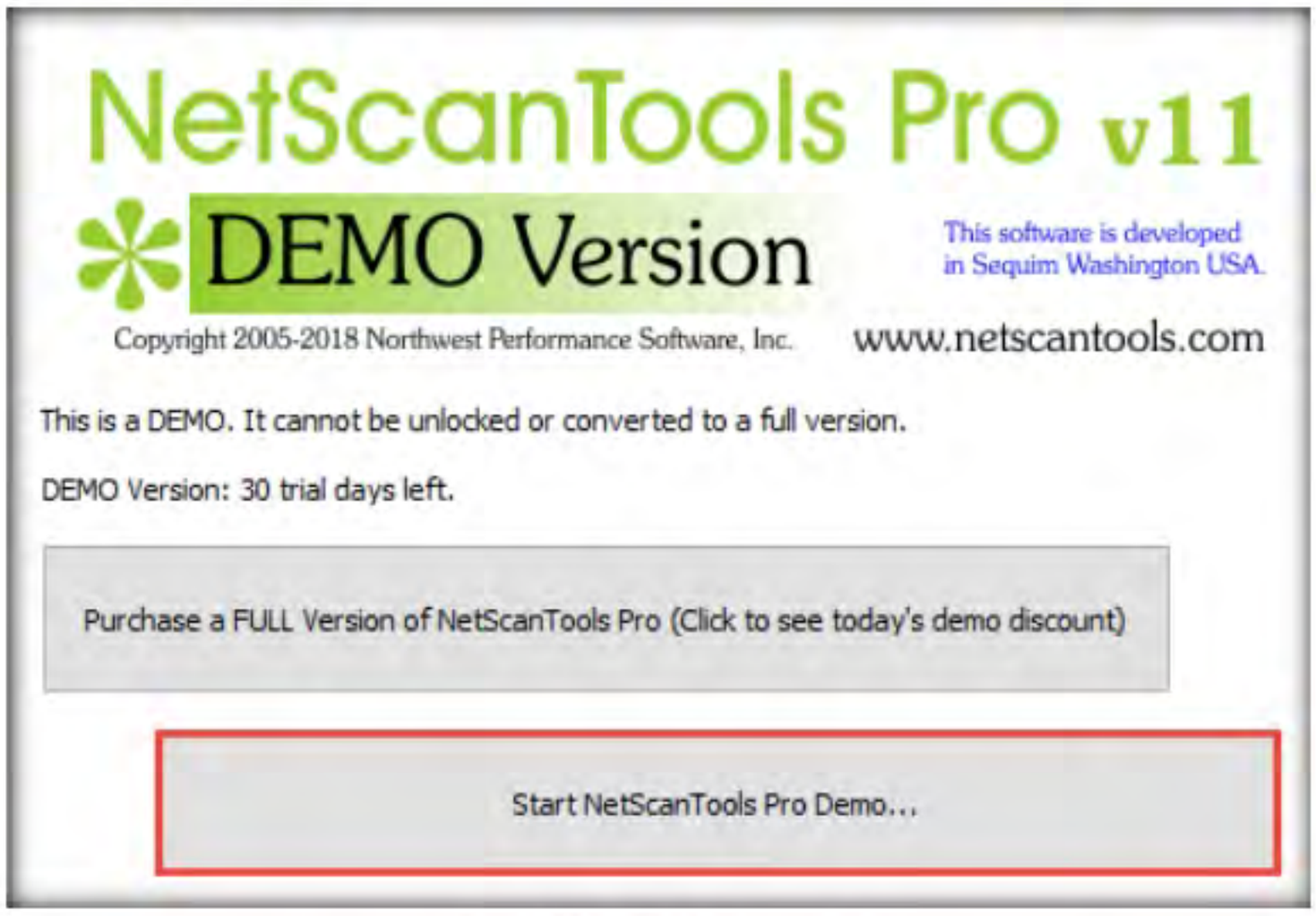


Figure 2.2.3: DEMO Version pop-up

6. The **NetScanTools Pro** main window appears, as shown in the screenshot.

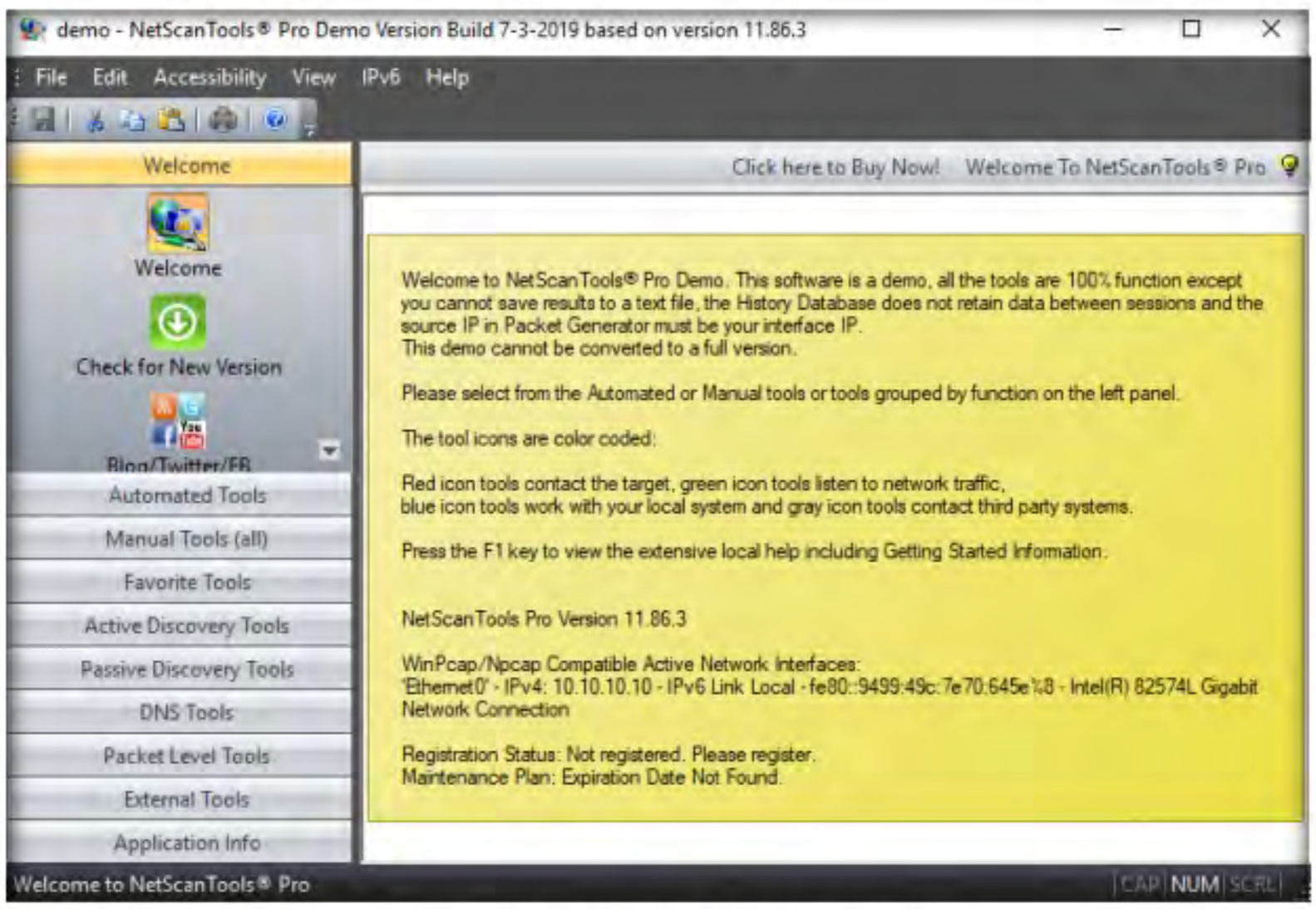


Figure 2.2.4: Main window of NetScan Tools Pro

TASK 2.2

Perform Ping Scan

7. In the left-hand pane, under the **Manual Tools (all)** section, scroll down and click the **Ping Scanner** option, as shown in the screenshot.

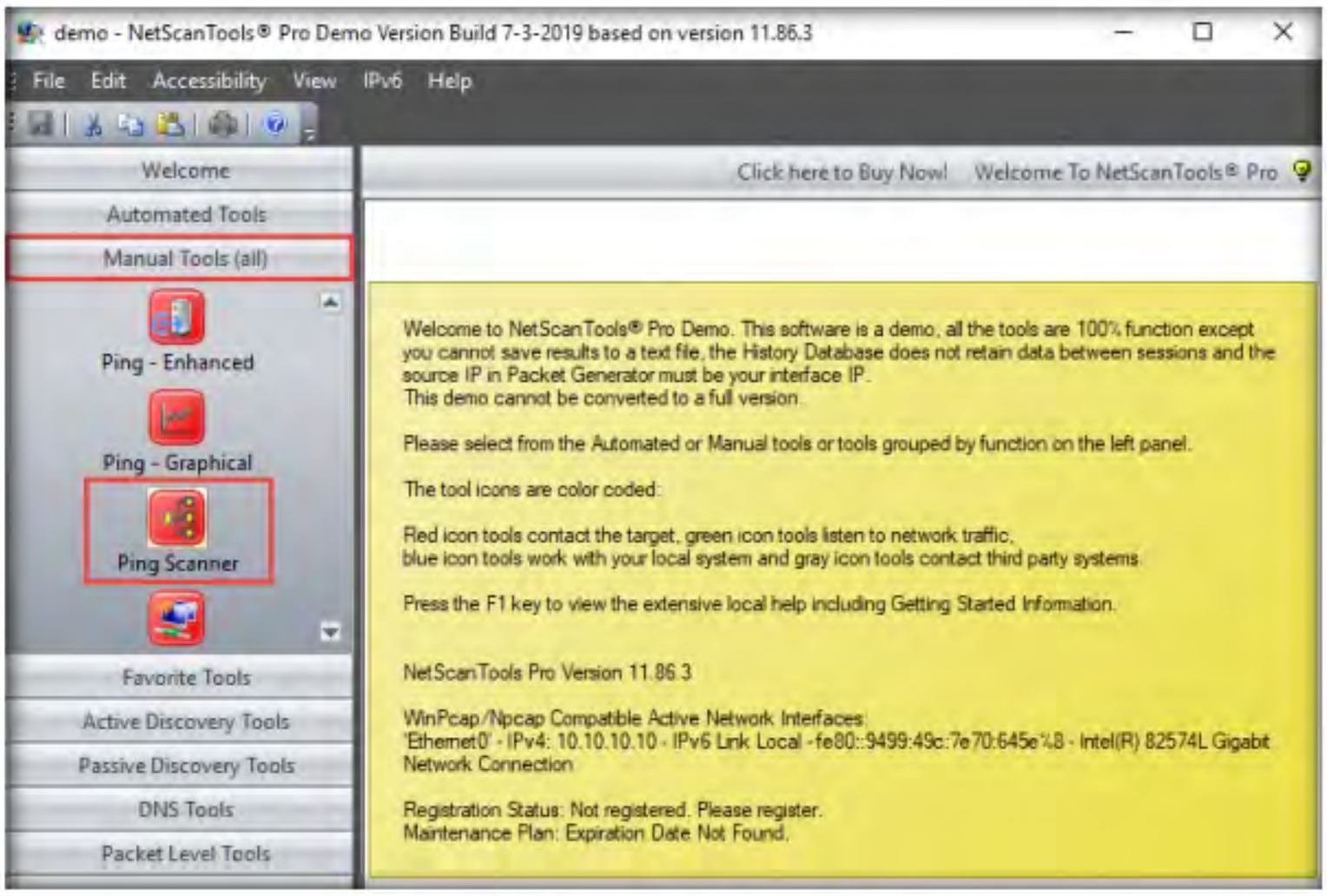


Figure 2.2.5: Selecting Ping Scanner option

8. A dialog box opens explaining the **Ping Scanner** tool; click **OK**.
9. Ensure that **Use Default System DNS** is selected. Enter the range of IP addresses into the **Start IP** and **End IP** fields (here, **10.10.10.5 - 10.10.10.20**); then, click **Start**.

Note: The IP address range might differ in your lab environment.

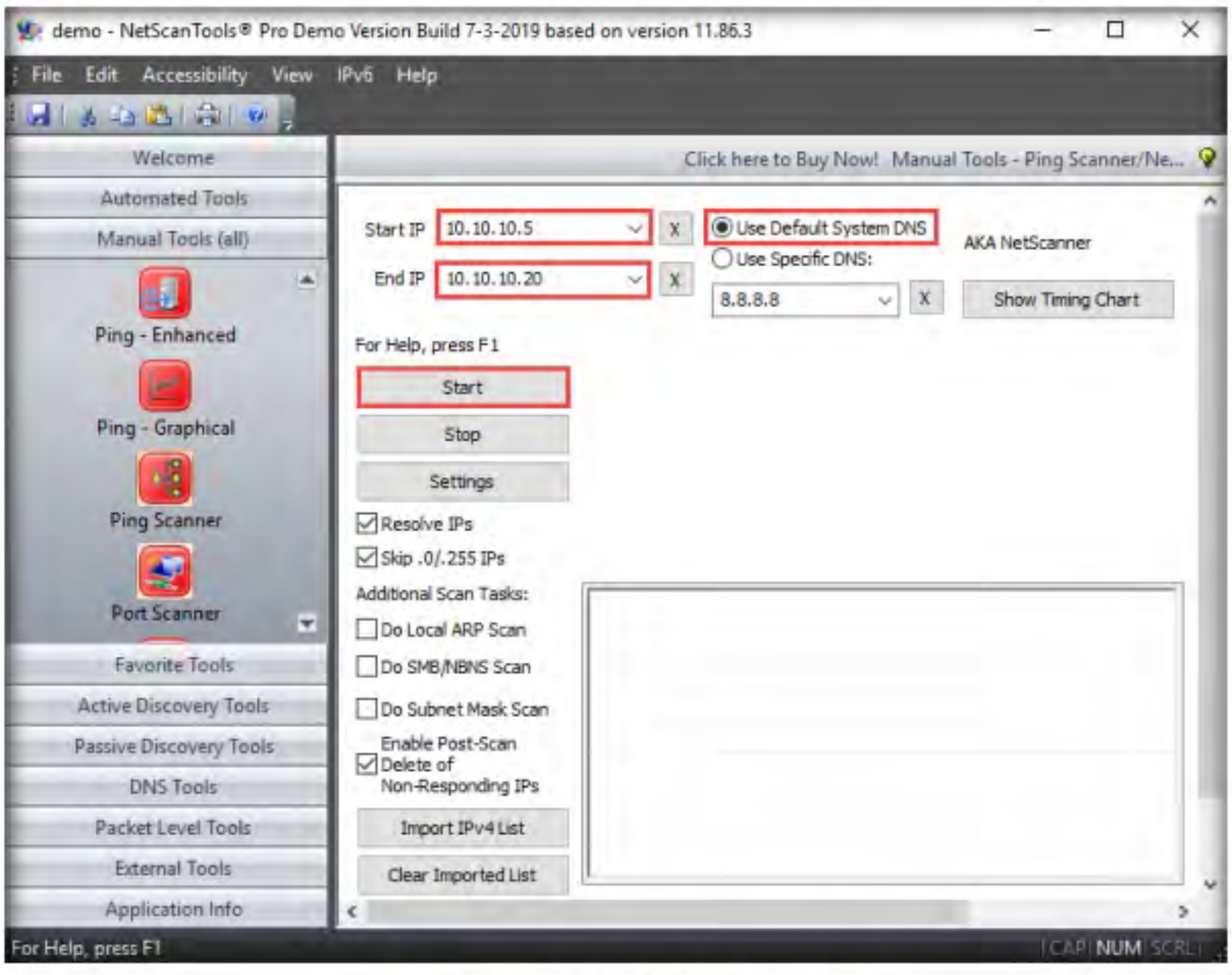


Figure 2.2.6: Configuring the Ping Scanner tool

10. A **Ping Scanner** notice pop-up appears; click **I Accept**.
11. After the completion of the scan, if a **How do you want to open this file?** pop-up appears, select any web browser (here, **Mozilla Firefox**) and click **OK**.

Note: If the browser opens automatically, skip to the next step.

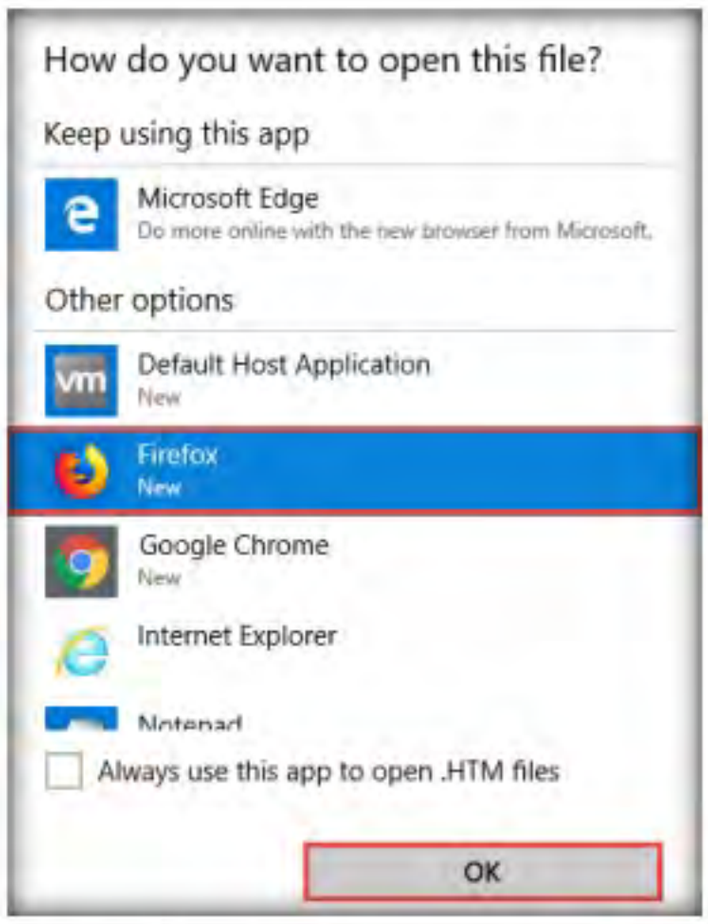
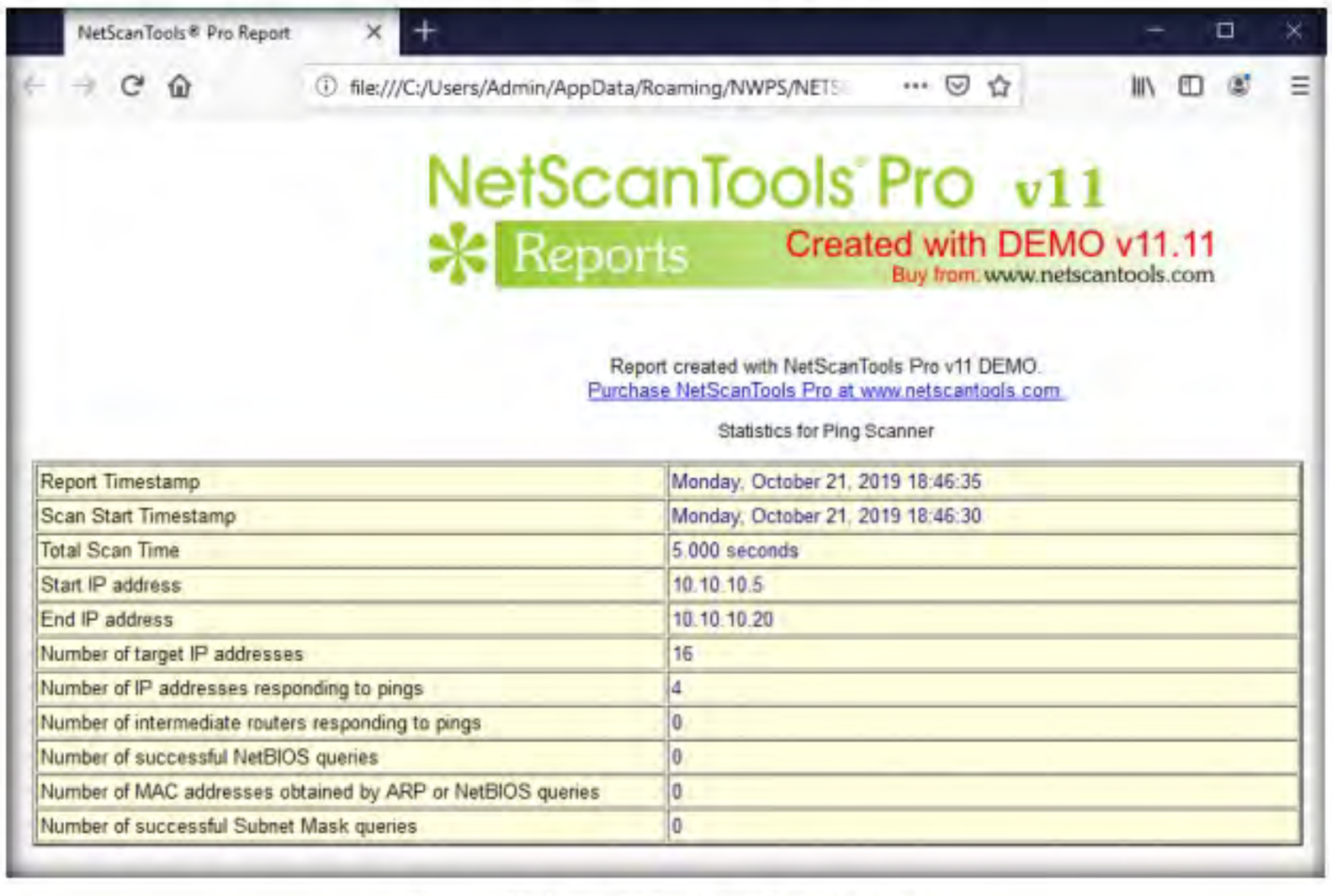


Figure 2.2.7: Choosing a browser to open the HTM file

12. A report appears in the browser displaying values such as **Start IP address**, **End IP address**, **Number of target IP addresses**, **Number of IP addresses responding to pings**, etc.

Note: The results might vary in your lab environment.



Report Timestamp	Monday, October 21, 2019 18:46:35
Scan Start Timestamp	Monday, October 21, 2019 18:46:30
Total Scan Time	5.000 seconds
Start IP address	10.10.10.5
End IP address	10.10.10.20
Number of target IP addresses	16
Number of IP addresses responding to pings	4
Number of intermediate routers responding to pings	0
Number of successful NetBIOS queries	0
Number of MAC addresses obtained by ARP or NetBIOS queries	0
Number of successful Subnet Mask queries	0

Figure 2.2.8: Browser displaying the scan report

TASK 2.3

Perform Port Scan

13. Close the browser and switch to the **NetScanTools Pro** window.
14. Now, click the **Port Scanner** option from the left-hand pane under the **Manual Tools (all)** section.

Note: If a dialog box appears explaining the **Port Scanner** tool, click **OK**.

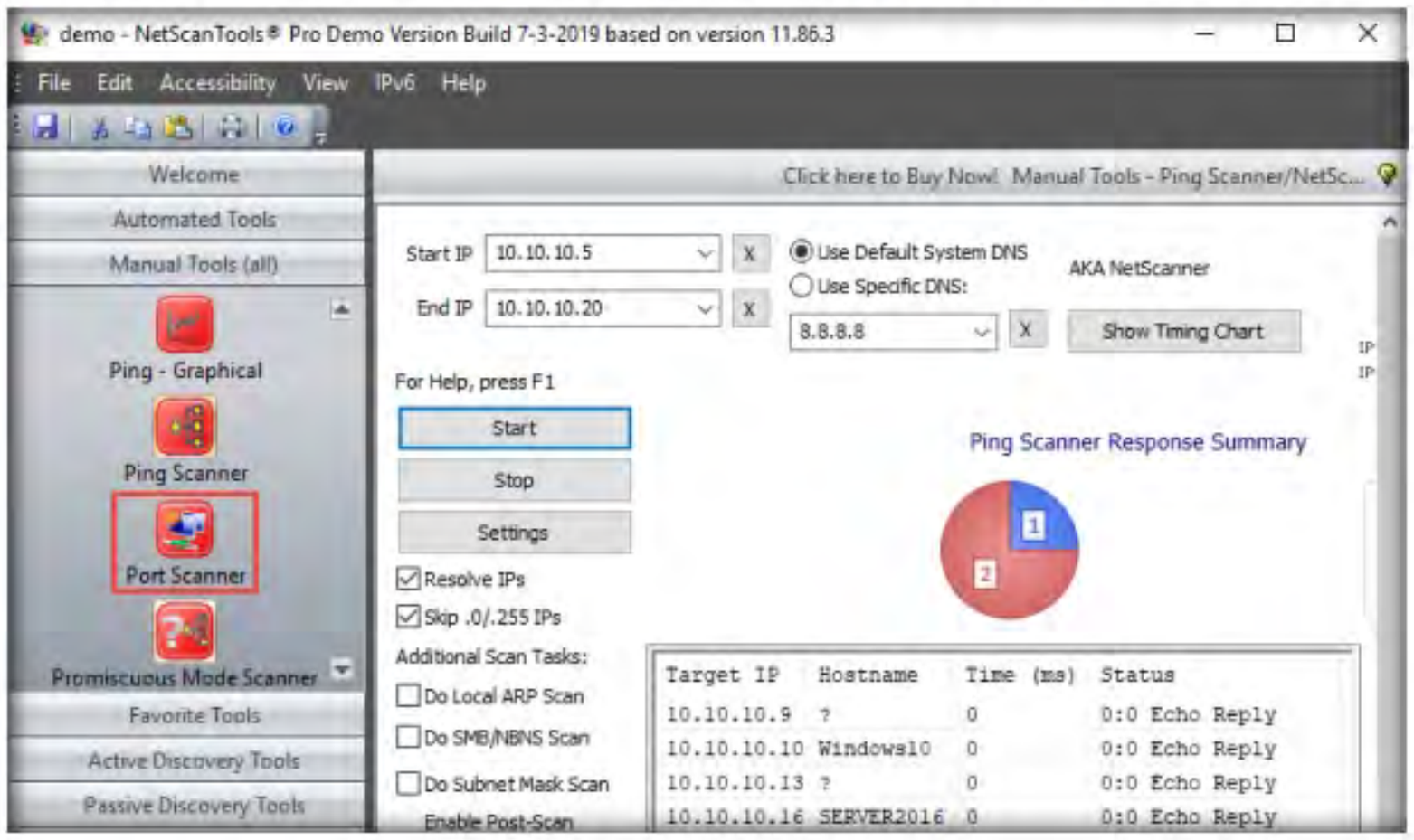


Figure 2.2.9: Selecting Port scanner option

15. In the **Target Hostname or IP Address** field, enter the IP address of the target (here, **10.10.10.16**). Ensure that **TCP Full Connect** is selected, and then click the **Scan Range of Ports** button.

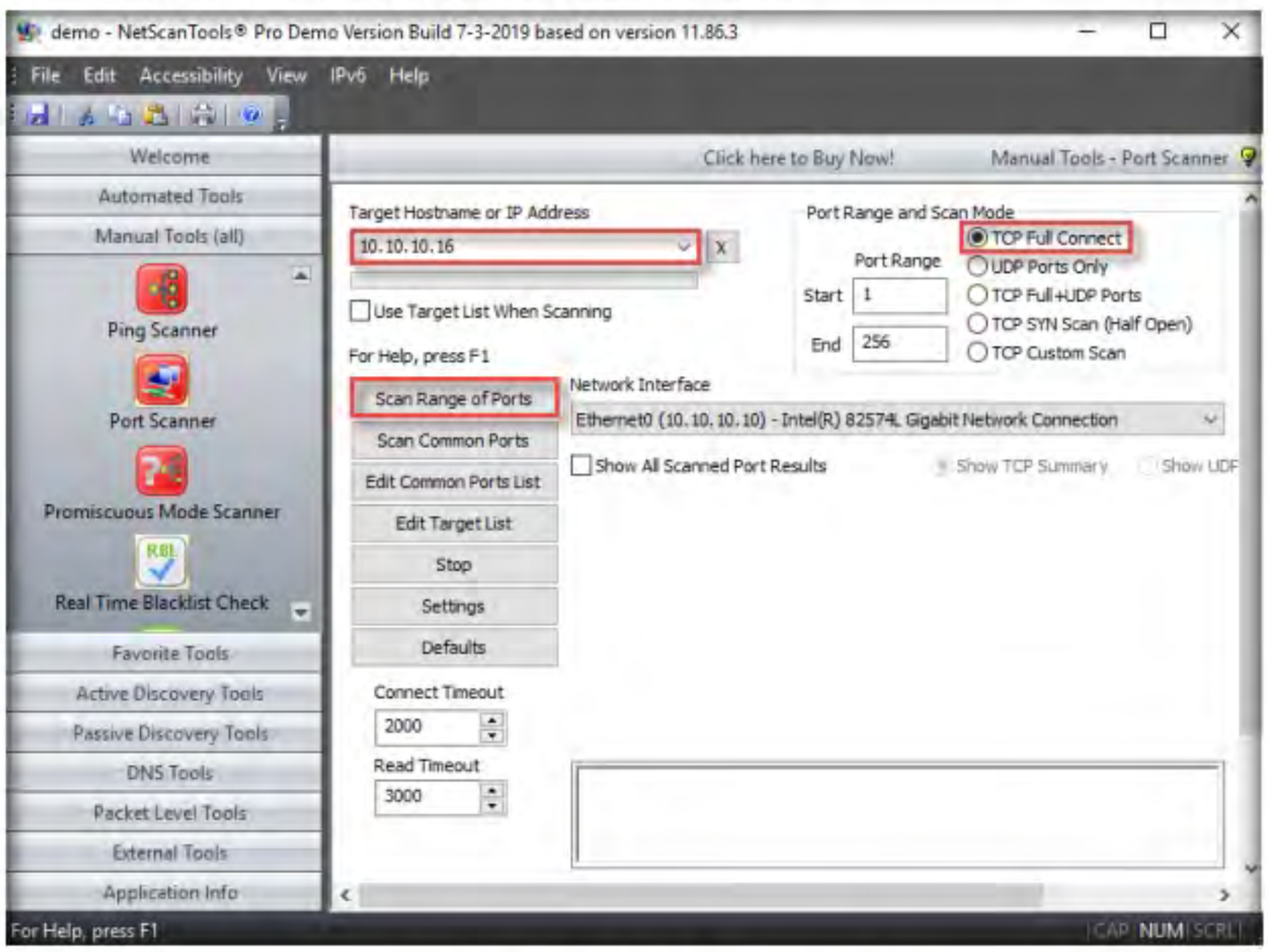


Figure 2.2.10: Configuring the Port Scanner tool

16. A **Port Scanner** notice pop-up appears; click **I Accept**.
17. A result appears displaying the active ports and their descriptions, as shown in the screenshot.

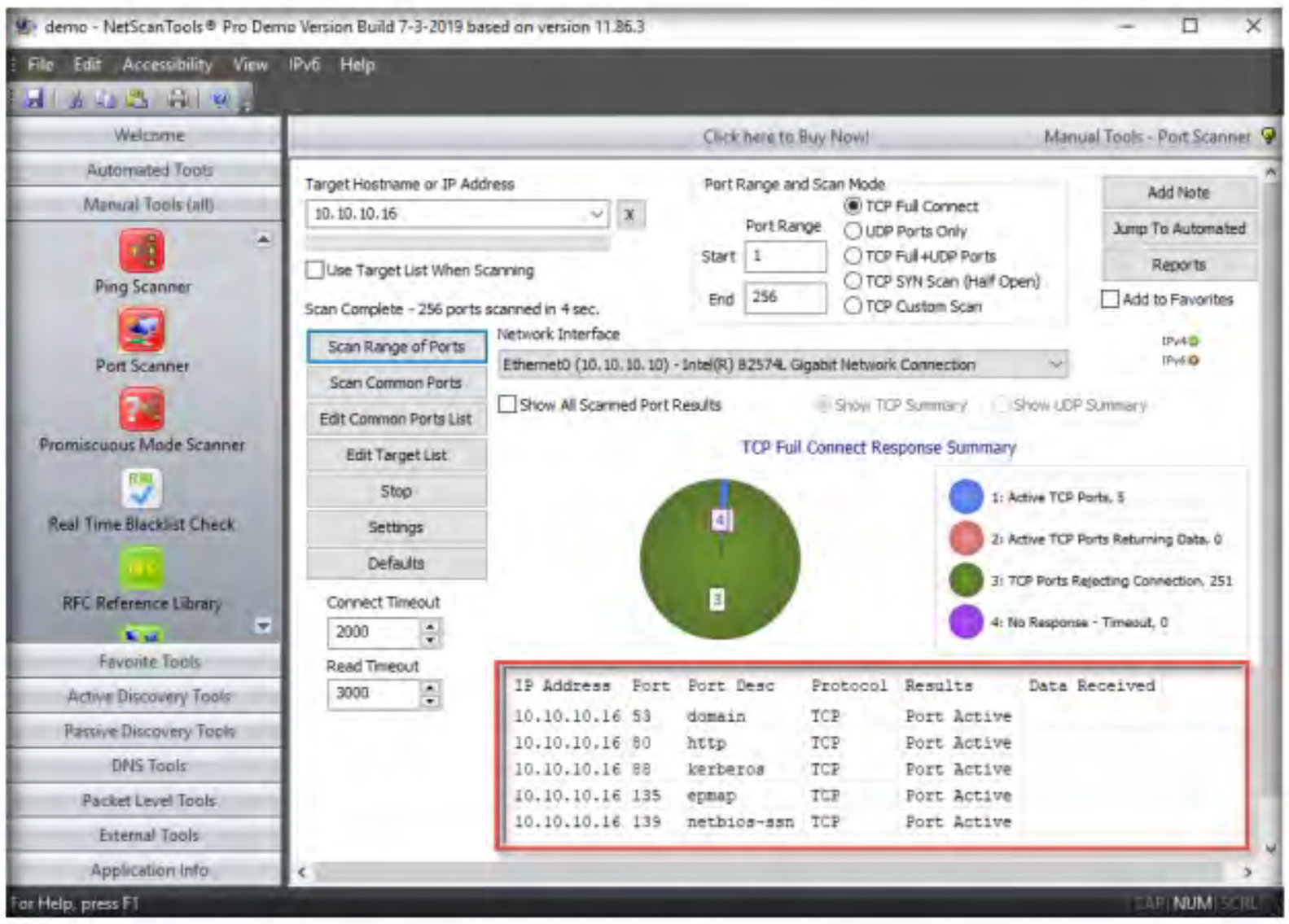


Figure 2.2.11: Port Scanner result

Note: By performing the above scans, you will be able to obtain a list of active machines in the network, their respective IP addresses and hostnames, and a list of all the open ports and services that will allow you to choose a target host in order to enter into its network and perform malicious activities such as ARP poisoning, sniffing, etc.

18. This concludes the demonstration of discovering open ports and services running on the target IP address using NetScanTools Pro.
19. Close all open windows and document all the acquired information.
20. Turn off the **Parrot Security** and **Ubuntu** virtual machines.

TASK 3 Explore Various Network Scanning Techniques using Nmap

Here, we will use Nmap to discover open ports and services running on the live hosts in the target network.

1. Before beginning this lab task, ensure that the **Windows Server 2016** virtual machine is turned on.

Note: In this lab task, we are using **Windows Server 2016** as a target machine.

2. In the **Windows 10** virtual machine, click on the **Start** menu and launch the **Nmap - Zenmap GUI** from the applications.

TASK 3.1

Perform TCP Connect/Full Open Scan

3. The **Nmap - Zenmap** GUI appears; in the **Command** field, type the command **nmap -sT -v <Target IP Address>** (here, the target IP address is **10.10.10.16**) and click **Scan**.

Note: **-sT**: performs the TCP connect/full open scan and **-v**: enables the verbose output (include all hosts and ports in the output).

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

Nmap comes with various inbuilt scripts that can be employed during a scanning process in an attempt to find the open ports and services running on the ports. It sends specially crafted packets to the target host, and then analyzes the responses to accomplish its goal. Nmap includes many port scanning mechanisms (TCP and UDP), OS detection, version detection, ping sweeps, etc.

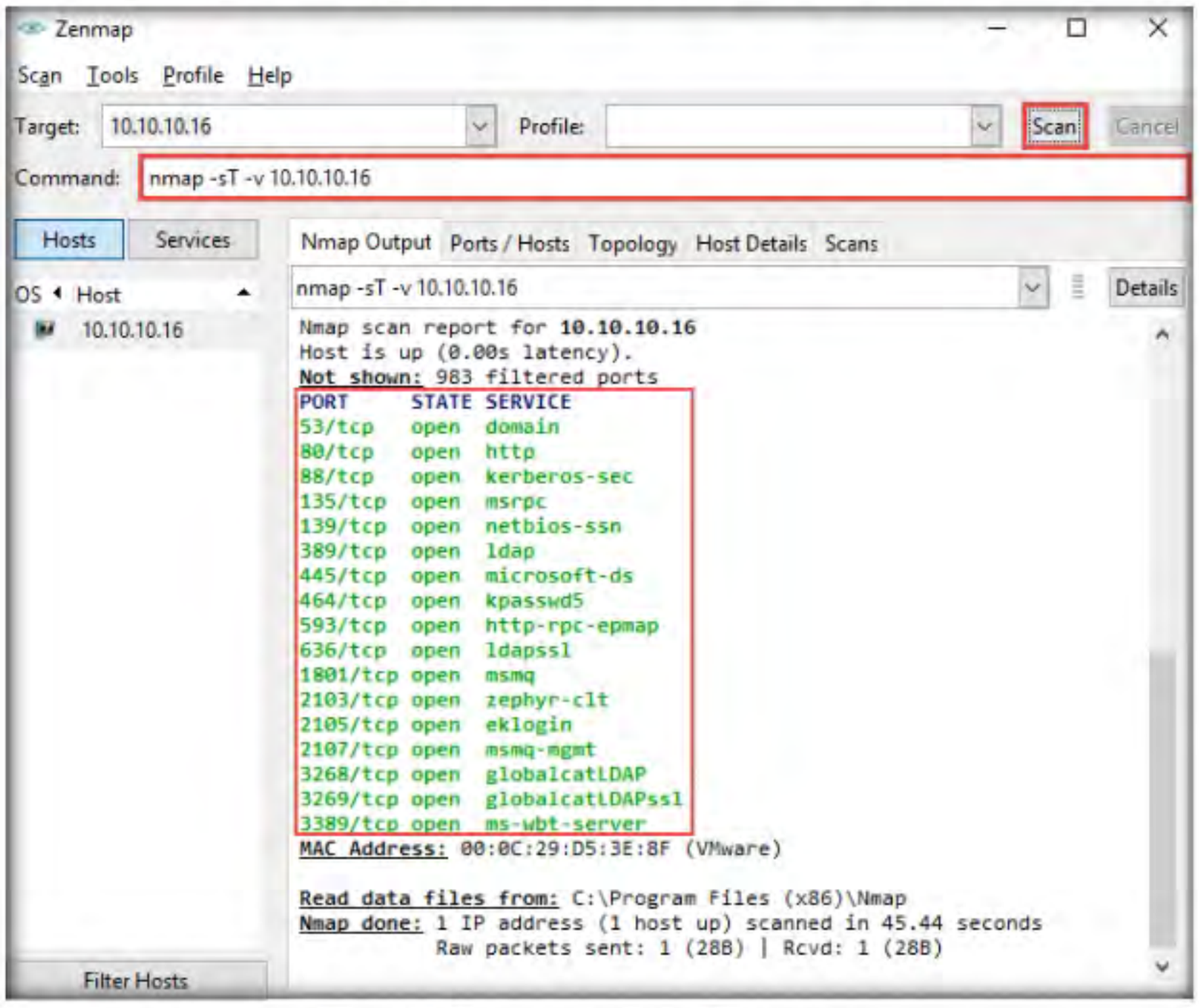


Figure 2.3.1: Zenmap scan results for TCP connect scan

Note: TCP connect scan completes a three-way handshake with the target machine. In the TCP three-way handshake, the client sends a SYN packet, which the recipient acknowledges with the SYN+ACK packet. In turn, the client acknowledges the SYN+ACK packet with an ACK packet to complete the connection. Once the handshake is completed, the client sends an RST packet to end the connection.

5. Click the **Ports/Hosts** tab to gather more information on the scan results. Nmap displays the Port, Protocol, State, Service, and Version of the scan.

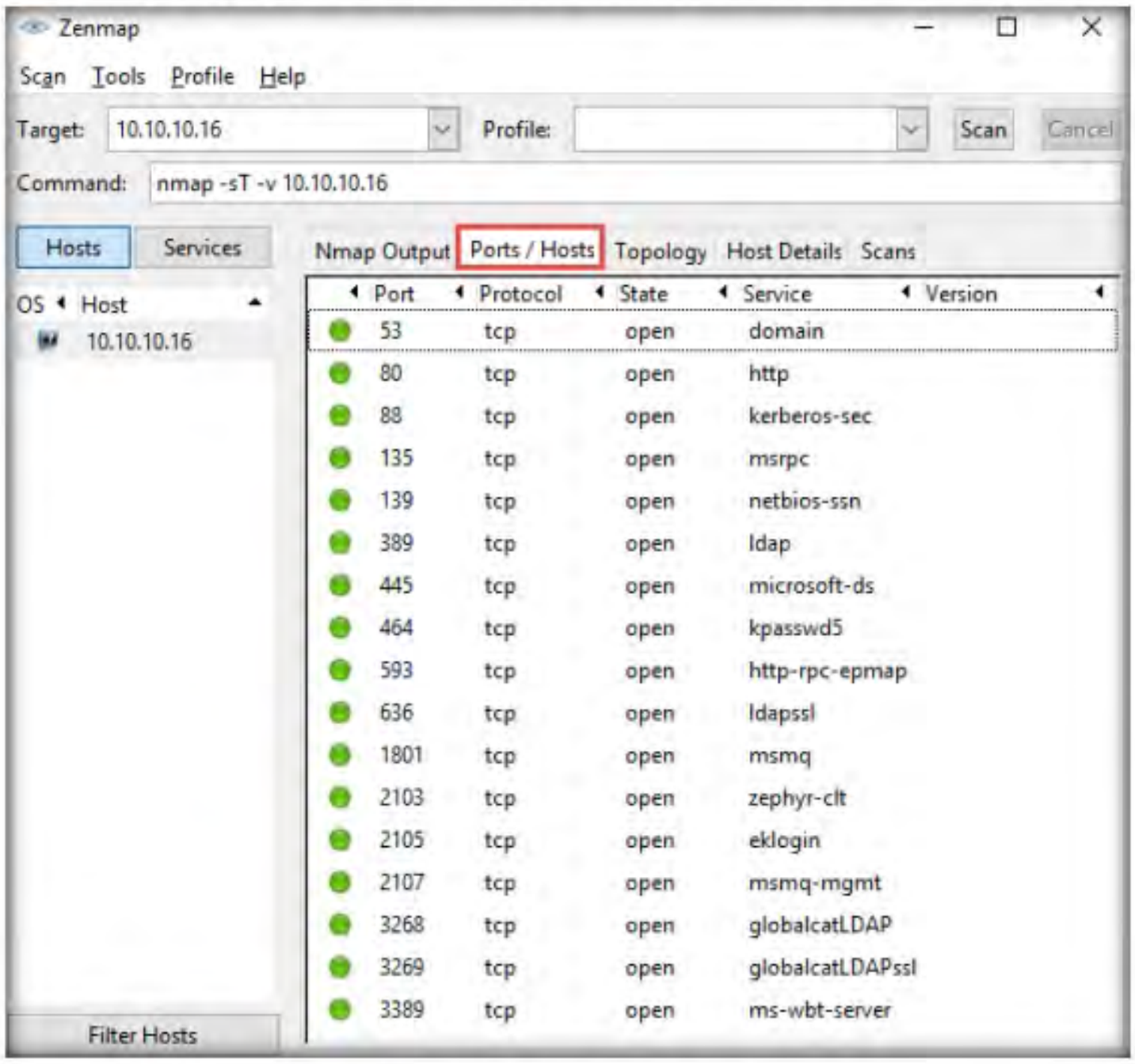


Figure 2.3.2: Zenmap Ports / Hosts tab

6. Click the **Topology** tab to view the topology of the target network that contains the provided IP address and click the **Fisheye** option to view the topology clearly.

Figure 2.3.3: Zenmap displaying topology of the selected target

7. In the same way, click the **Host Details** tab to view the details of the TCP connect scan.
8. Click the **Scans** tab to view the command used to perform TCP connect/full open scan.
9. Click the **Services** tab located in the right pane of the window. This tab displays a list of services.

Note: You can use any of these services and their open ports to enter into the target network/host and establish a connection.

10. In this lab, we shall be performing a stealth scan/TCP half-open scan, Xmas scan, TCP Maimon scan, and ACK flag probe scan on a firewall-enabled machine (i.e., **Windows Server 2016**) in order to observe the result. To do this, we need to enable **Windows Firewall** in the **Windows Server 2016** virtual machine.
11. Switch to the **Windows Server 2016** virtual machine and log in with the credentials **Administrator**/**Pa$$w0rd**.
12. 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.

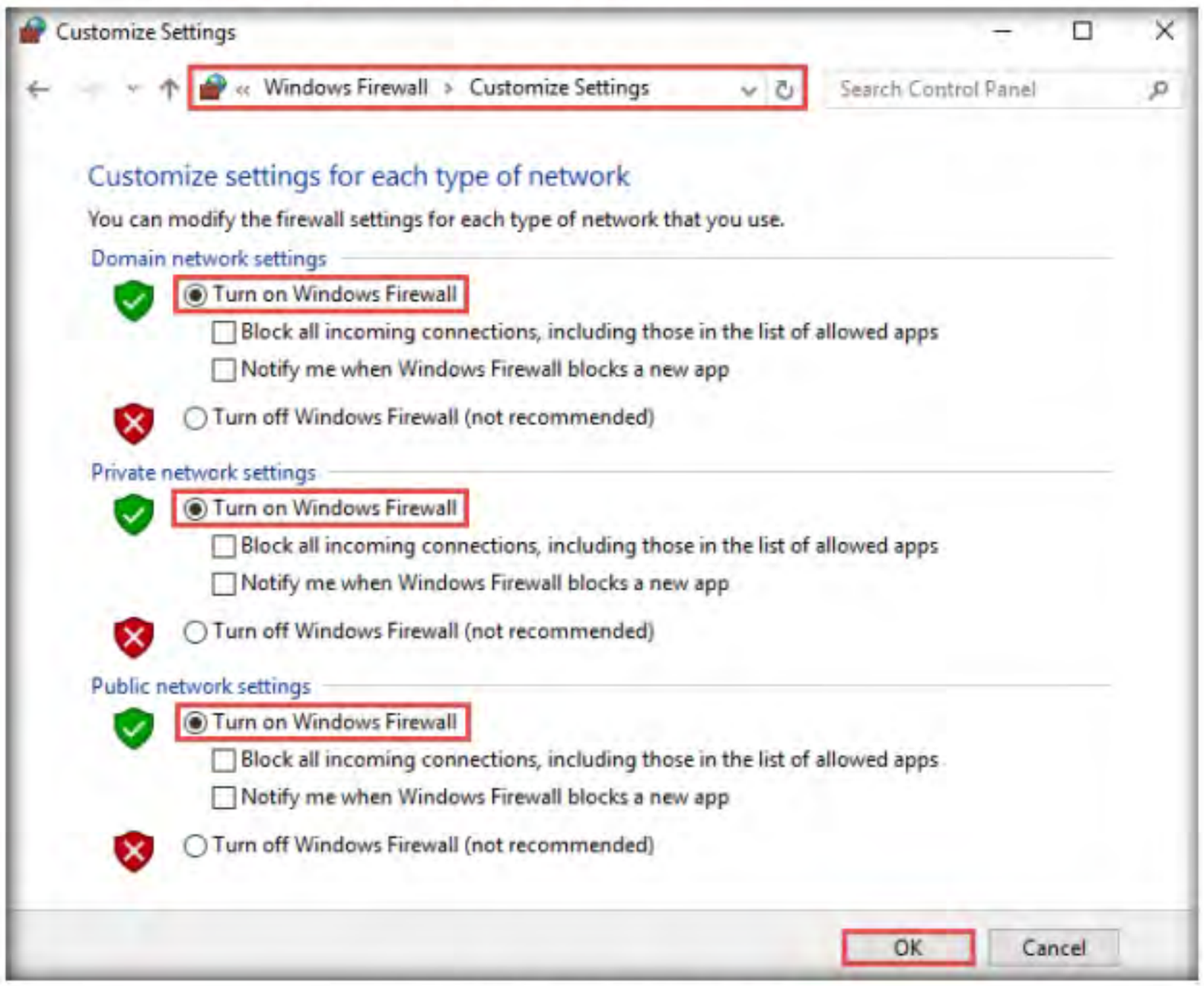


Figure 2.3.4: Turn on Windows Firewall

TASK 3.2

Perform Stealth Scan/TCP Half Open Scan

13. Now, switch to the **Windows 10** virtual machine. In the **Command** field of **Zenmap**, type the command **nmap -sS -v <Target IP Address>** (here, the target IP address is **10.10.10.16**) and click **Scan**.

Note: **-sS**: performs the stealth scan/TCP half-open scan and **-v**: enables the verbose output (include all hosts and ports in the output).

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

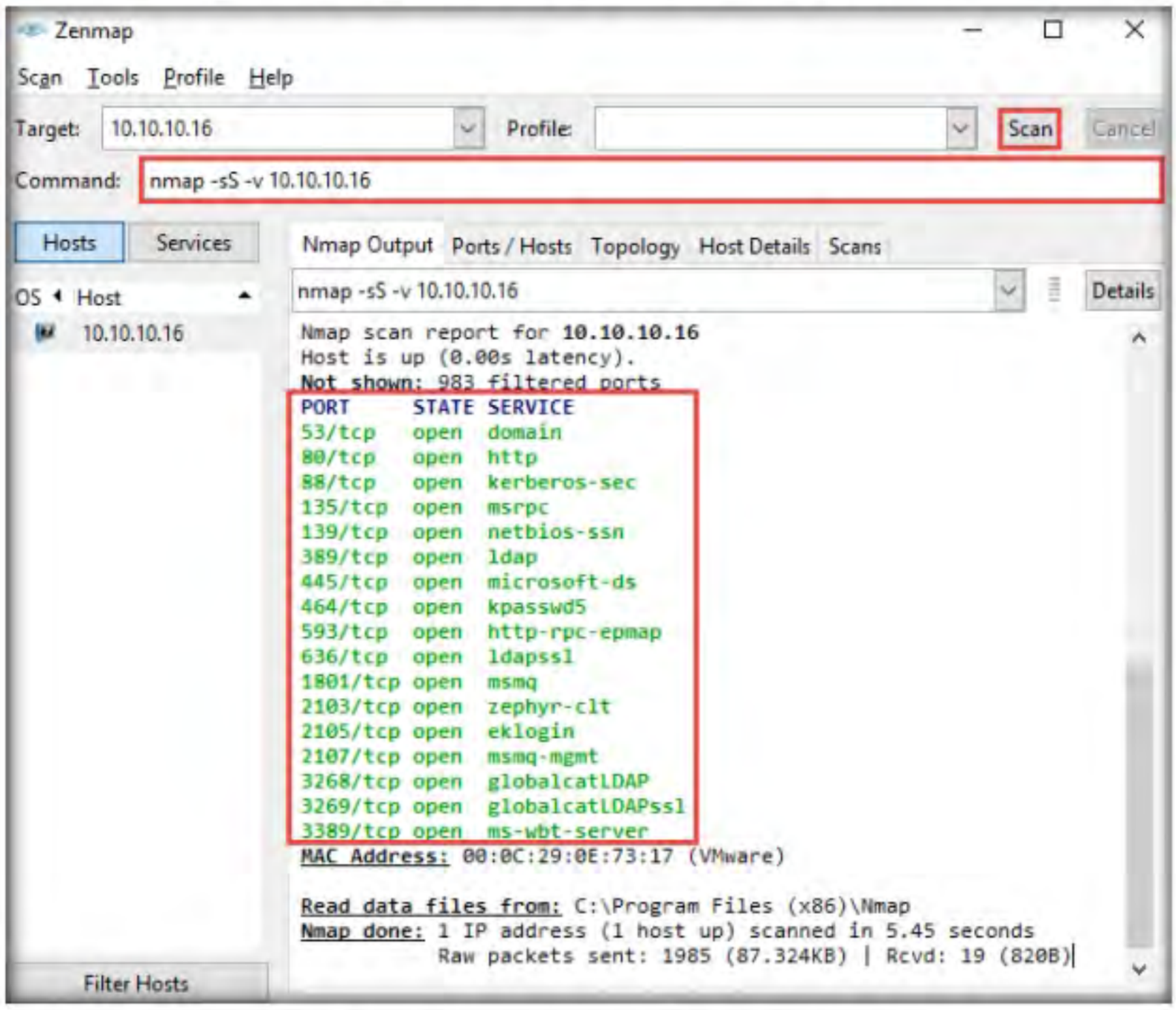


Figure 2.3.5: Zenmap scan results for Stealth scan

Note: The stealth scan involves resetting the TCP connection between the client and server abruptly before completion of three-way handshake signals, and hence leaving the connection half-open. This scanning technique can be used to bypass firewall rules, logging mechanisms, and hide under network traffic.

15. As shown in the last task, you can gather detailed information from the scan result in the **Ports/Hosts**, **Topology**, **Host Details**, and **Scan** tab.

TASK 3.3

Perform Xmas Scan

16. In the **Command** field of **Zenmap**, type the command **nmap -sX -v <Target IP Address>** (here, the target IP address is **10.10.10.16**) and click **Scan**.

Note: **-sX**: performs the Xmas scan and **-v**: enables the verbose output (include all hosts and ports in the output).

17. The scan results appear, displaying that the ports are either open or filtered on the target machine, which means a firewall has been configured on the target machine.

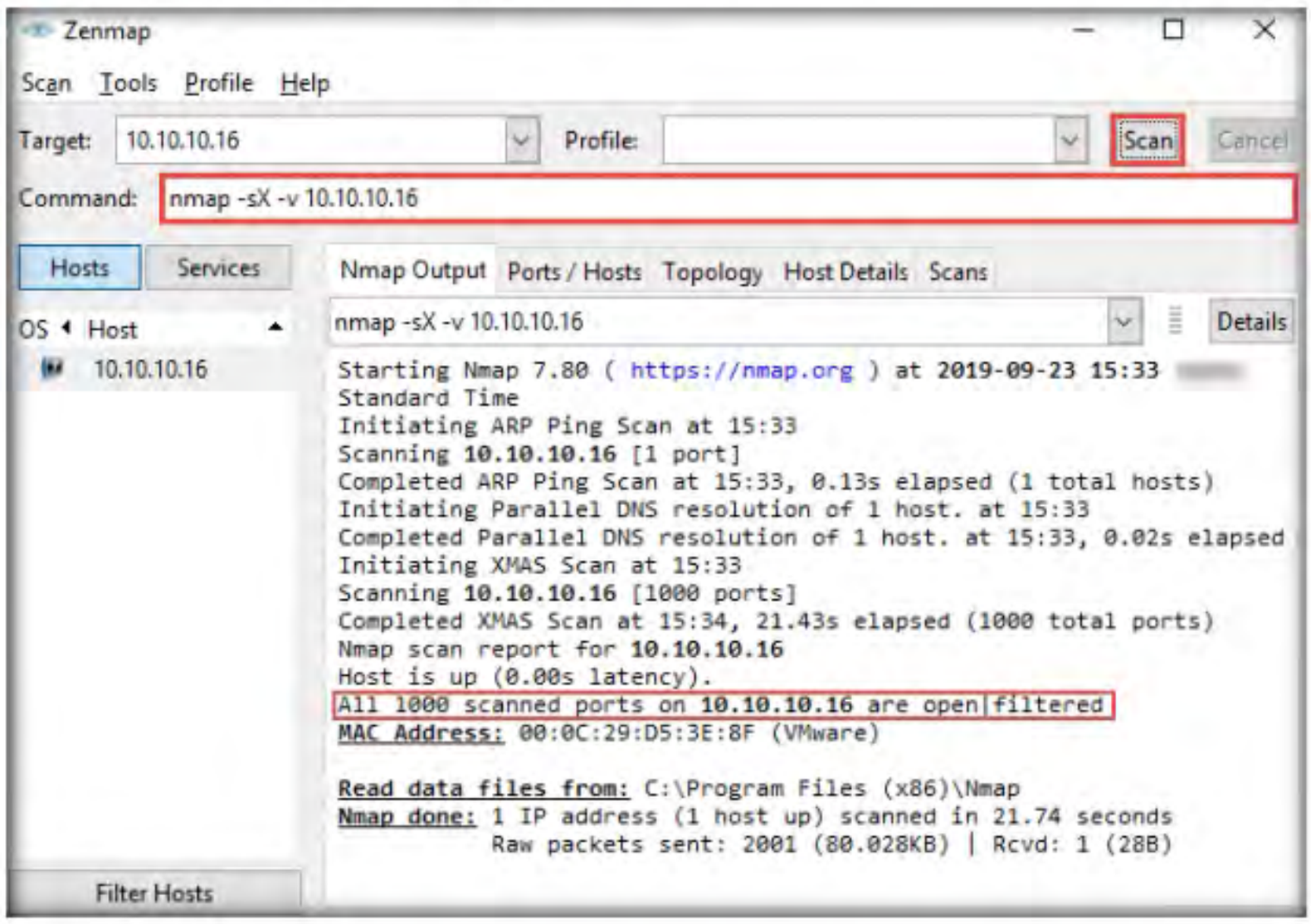


Figure 2.3.6: Zenmap scan results for Xmas scan

Note: Xmas scan sends a TCP frame to a target system with FIN, URG, and PUSH flags set. If the target has opened the port, then you will receive no response from the target system. If the target has closed the port, then you will receive a target system reply with an RST.

TASK 3.4

Perform TCP Maimon Scan

18. In the **Command** field, type the command **nmap -sM -v <Target IP Address>** (here, the target IP address is **10.10.10.16**) and click **Scan**

Note: **-sM**: performs the TCP Maimon scan and **-v**: enables the verbose output (include all hosts and ports in the output).

19. The scan results appear, displaying either the ports are open/filtered on the target machine, which means a firewall has been configured on the target machine.

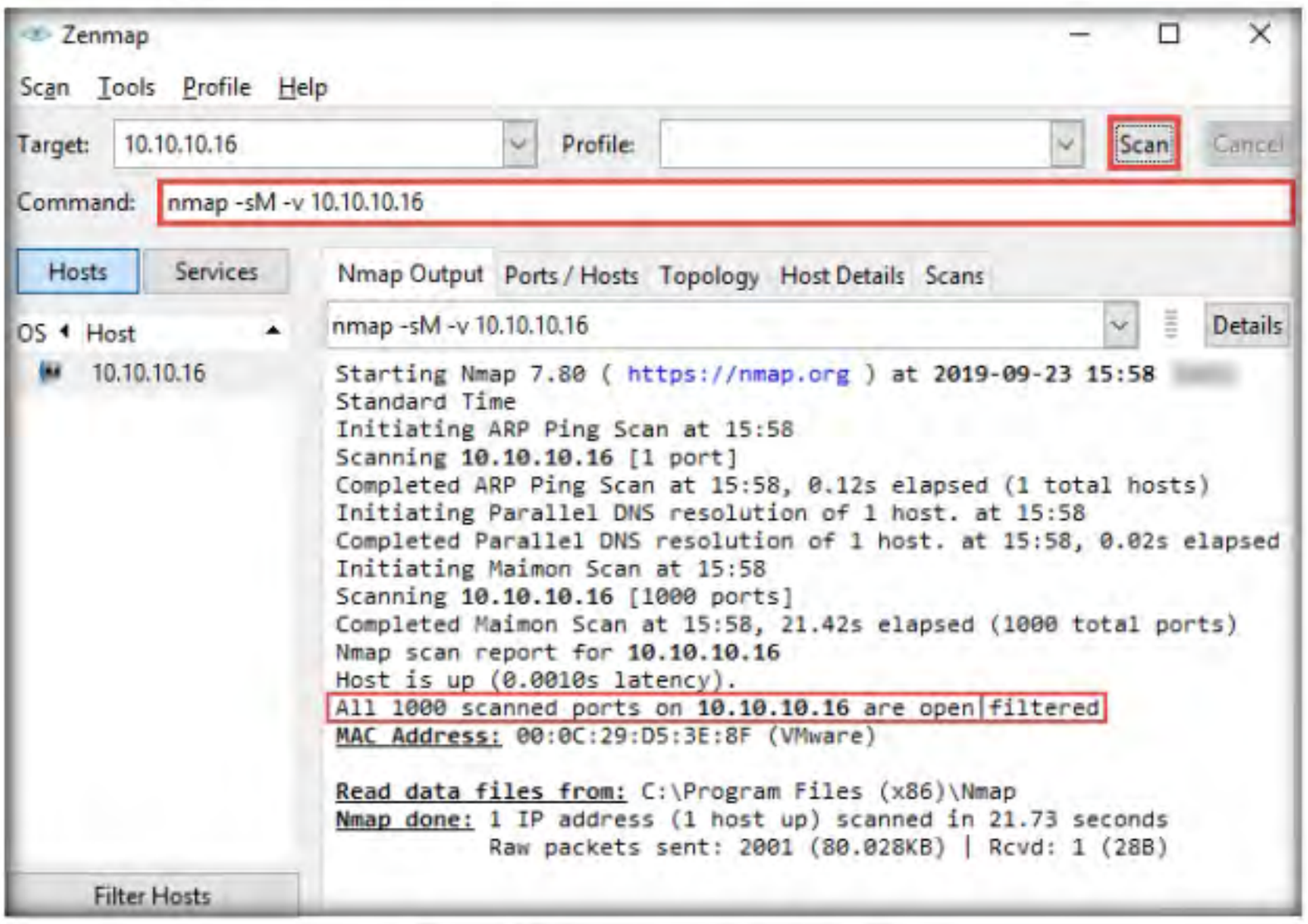


Figure 2.3.7: Zenmap scan results for Stealth scan

Note: In the TCP Maimon scan, a FIN/ACK probe is sent to the target; if there is no response, then the port is Open|Filtered, but if the RST packet is sent as a response, then the port is closed.

TASK 3.5

Perform ACK Flag Probe Scan

20. In the **Command** field, type the command **nmap -sA -v <Target IP Address>** (here, the target IP address is **10.10.10.16**) and click **Scan**.

Note: **-sA**: performs the ACK flag probe scan and **-v**: enables the verbose output (include all hosts and ports in the output).

21. The scan results appear, displaying that the ports are unfiltered on the target machine, as shown in the screenshot.

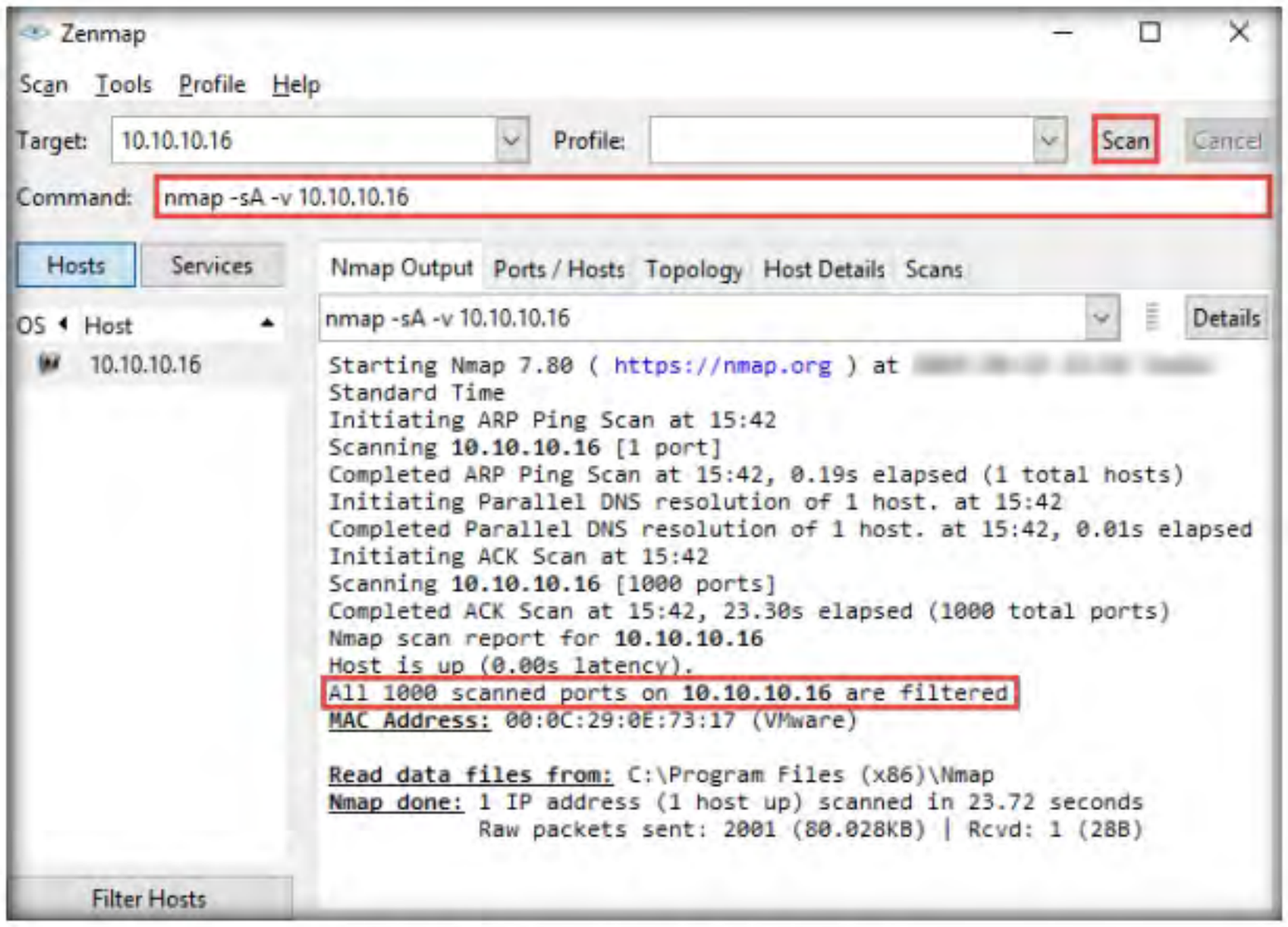


Figure 2.3.8: Zenmap scan results for Stealth scan

Note: The ACK flag probe scan sends an ACK probe packet with a random sequence number; no response implies that the port is filtered (stateful firewall is present), and an RST response means that the port is not filtered.

22. Now, switch to the **Windows Server 2016** virtual machine and turn off the **Windows Firewall** from **Control Panel**.

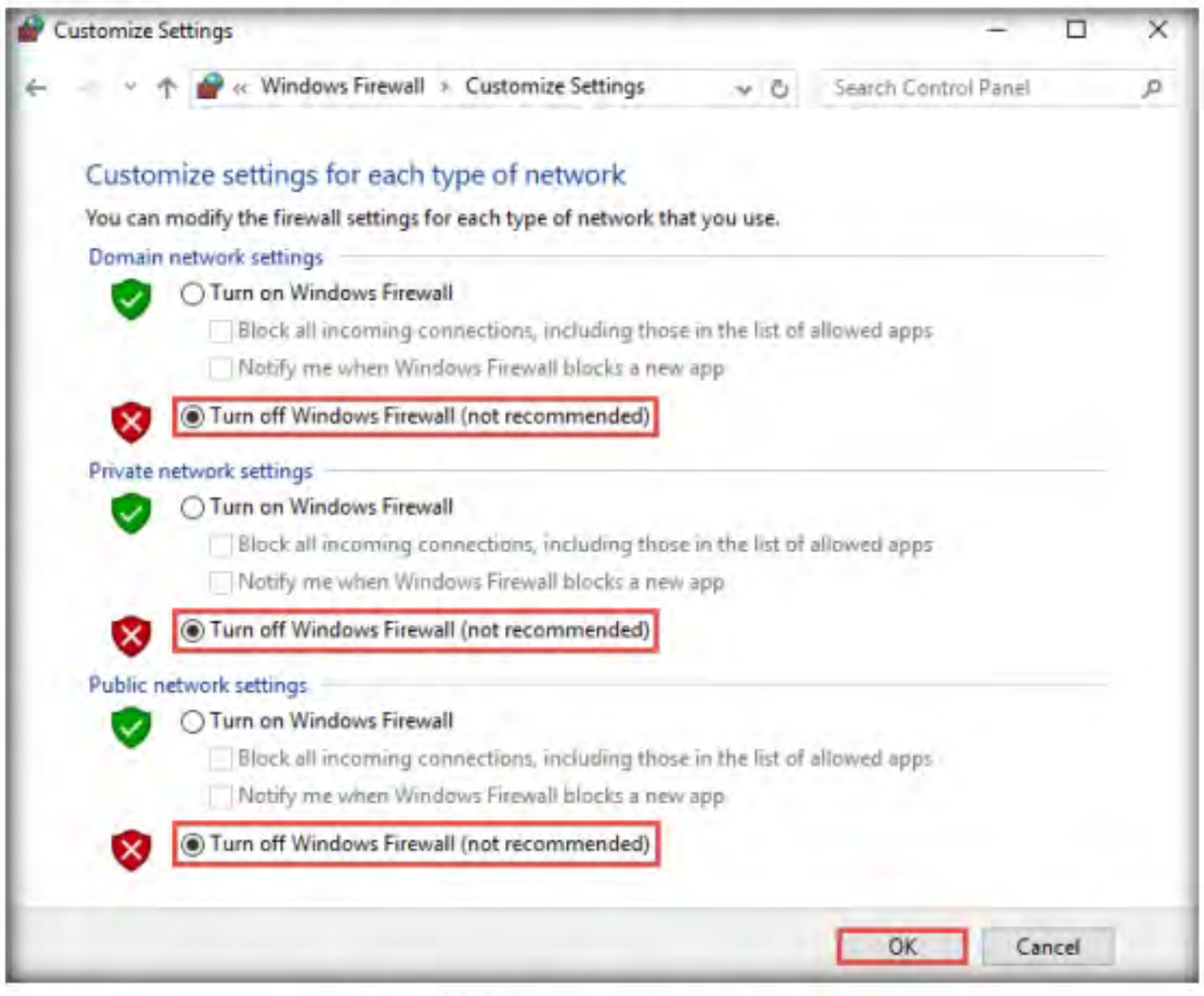


Figure 2.3.9: Turning off Windows Firewall

TASK 3.6

Perform UDP Scan

23. Now, return to the **Windows 10** virtual machine. In the **Command** field, type the command **nmap -sU -v <Target IP Address>** (here, the target IP address is **10.10.10.16**) and click **Scan**.

Note: **-sU**: performs the UDP scan and **-v**: enables the verbose output (include all hosts and ports in the output).

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

Note: This scan will take approximately 20 minutes to finish the scanning process and the results might differ in your lab environment.

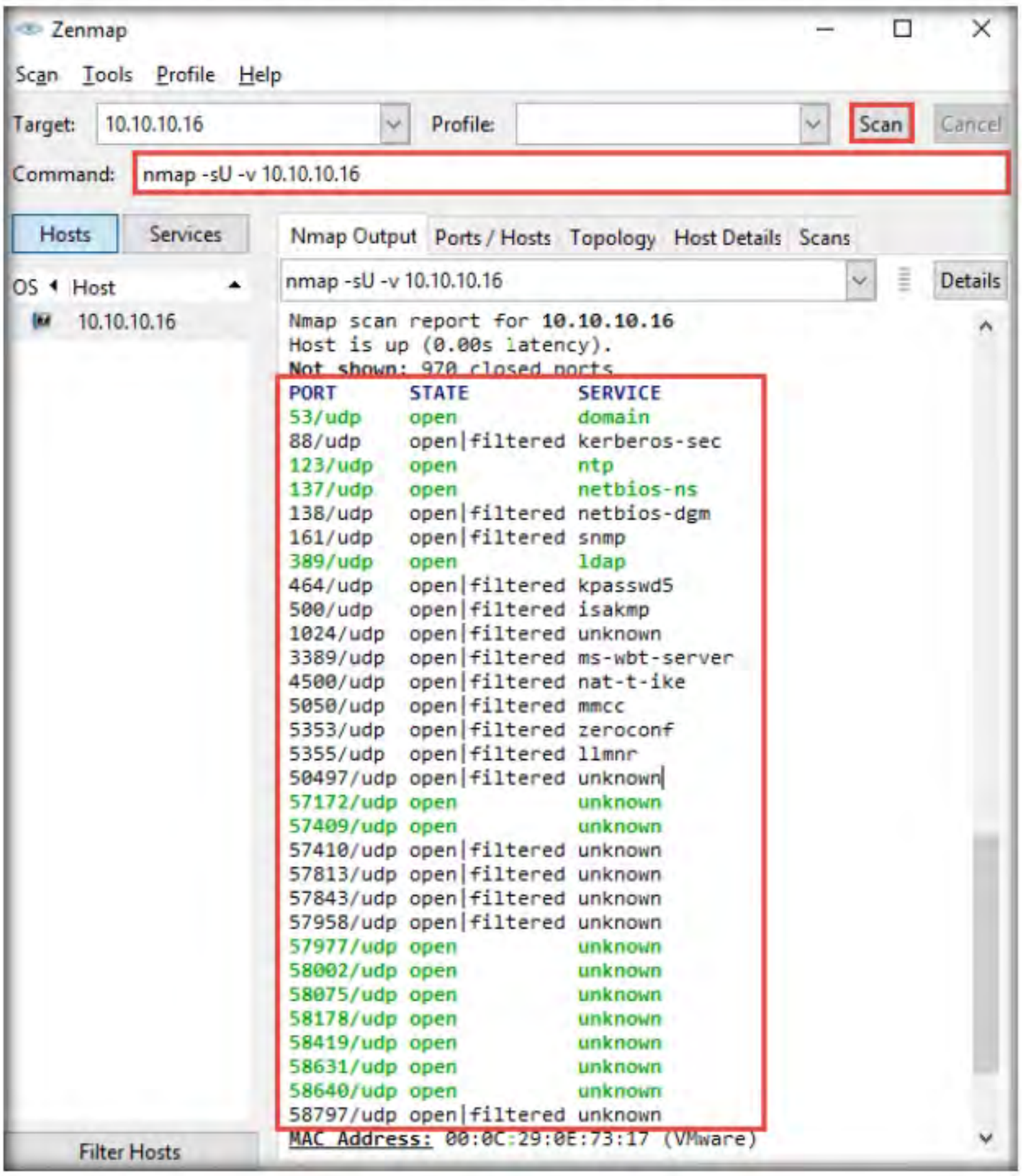


Figure 2.3.10: Zenmap- scan results for UDP scan

Note: The UDP scan uses UDP protocol instead of the TCP. There is no three-way handshake for the UDP scan. It sends UDP packets to the target host; no response means that the port is open. If the port is closed, an ICMP port unreachable message is received.

25. Close the **Zenmap** window.
26. You can create your scan profile, or you can also choose the default scan profiles available in Nmap to scan a network
27. Double-click the **Nmap - Zenmap GUI** shortcut from **Desktop** to launch Nmap.
28. To choose the default scan profiles available in Nmap, click on the drop-down icon in the **Profile** field and select the scanning technique you want to use.

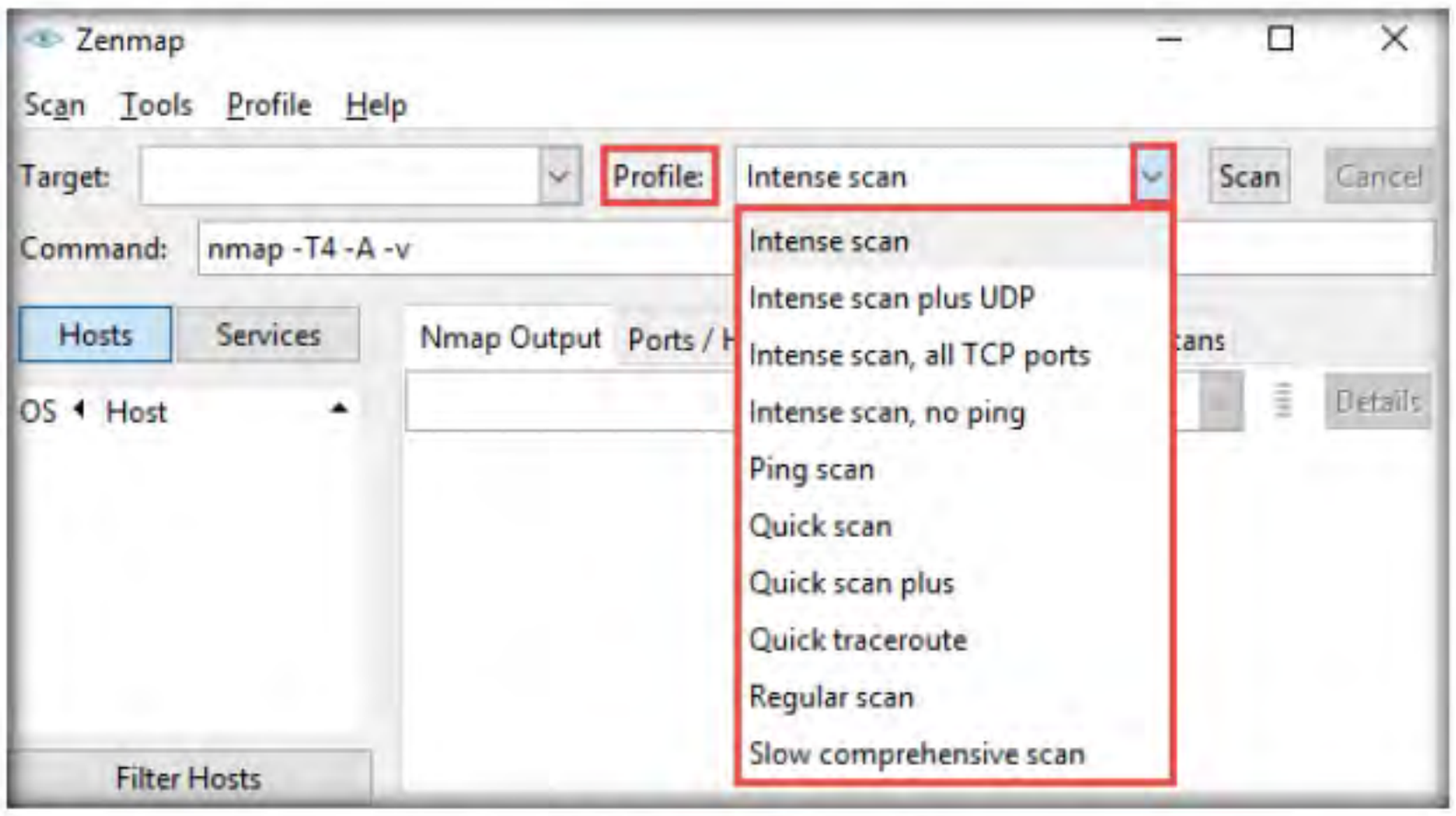


Figure 2.3.11: Zenmap Default Scan Options

TASK 3.7

Create a Scan Profile

29. To create a scan profile; click **Profile → New Profile or Command**.

Note: If a **User Account Control** pop-up appears, click **Yes**.

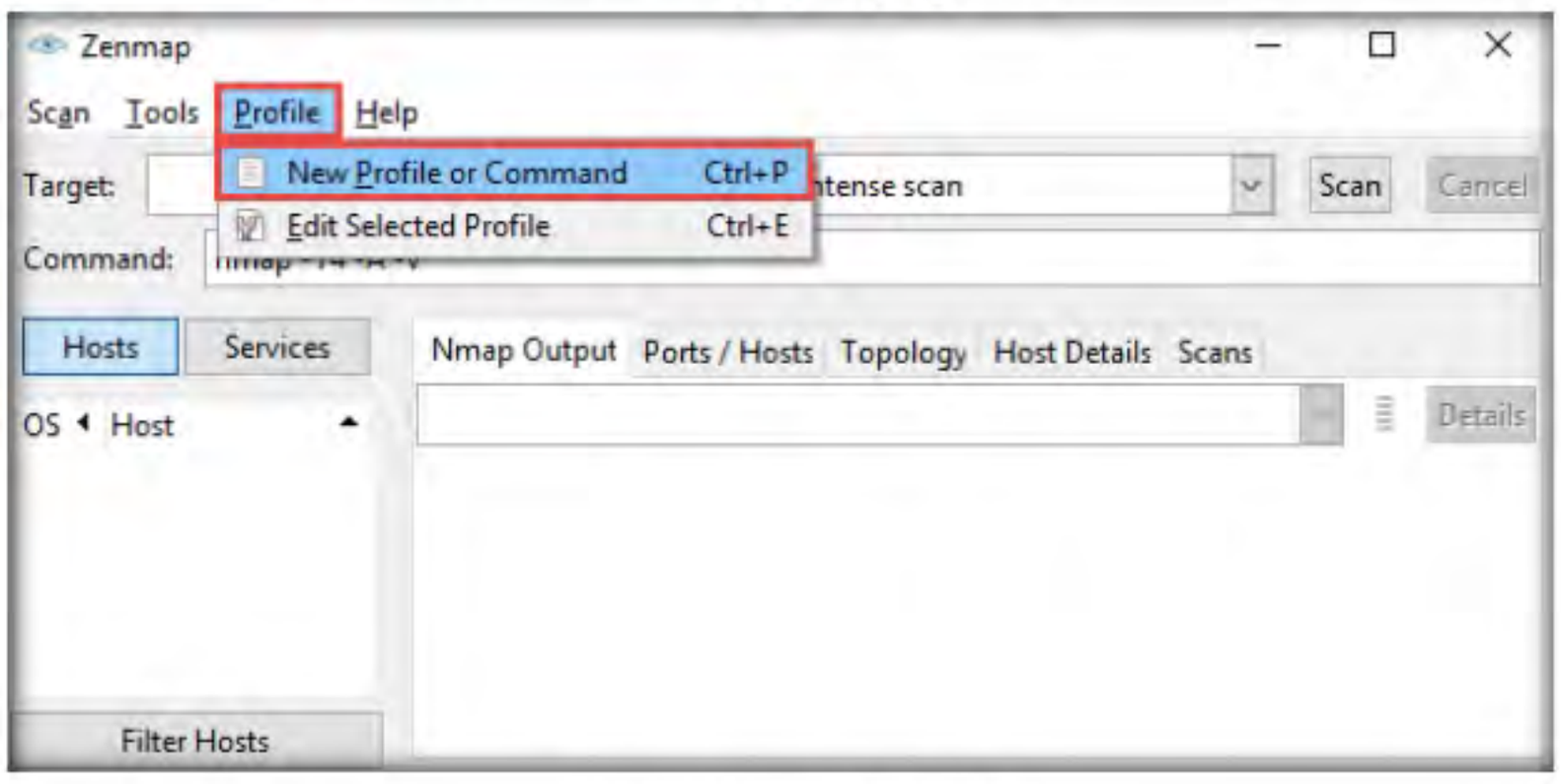


Figure 2.3.12: Creating a New Profile

30. The **Profile Editor** window appears. In the **Profile** tab, under the **Profile Information** section, input a profile name (here, **Null Scan**) into the **Profile name** field.

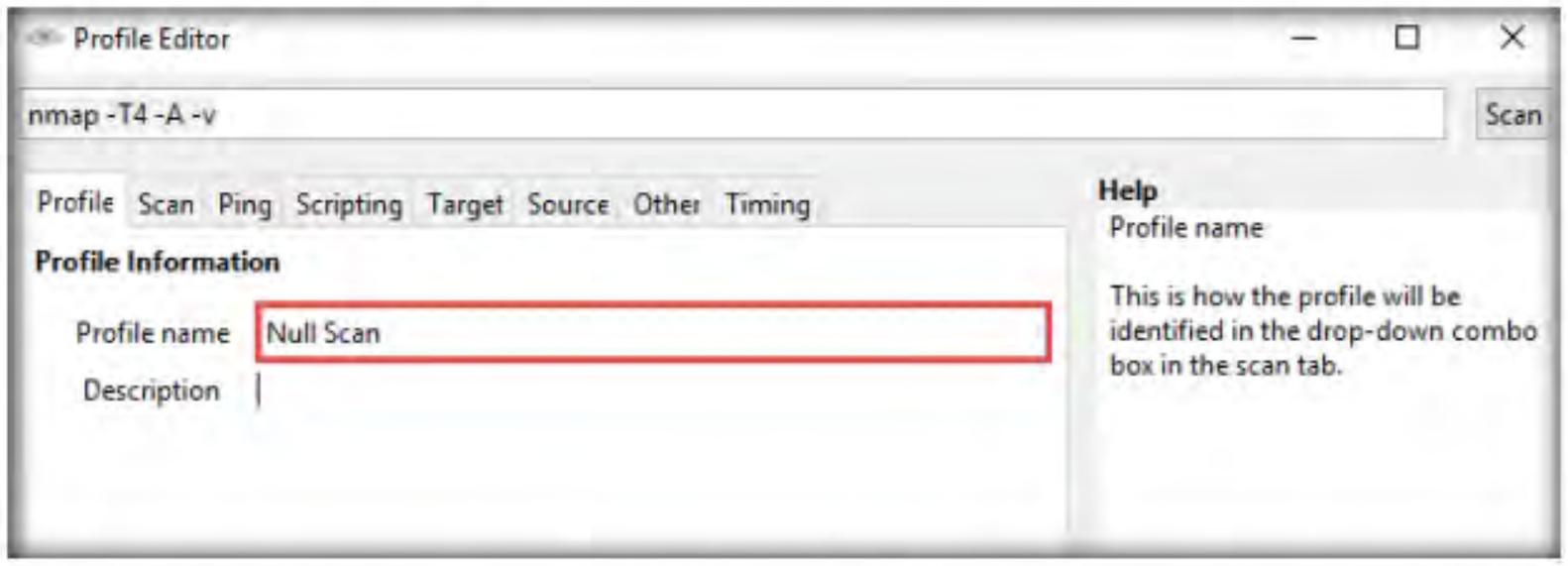


Figure 2.3.13: Entering Profile Name

31. Now, click the **Scan** tab and select the scan option (here, **Null scan (-sN)**) from the **TCP scan** drop-down list.
32. Select **None** in the **Non-TCP scans** drop-down list and **Aggressive (-T4)** in the **Timing template** list. Ensure that the **Enable all advanced/aggressive options (-A)** checkbox is selected and click **Save Changes**, as shown in the screenshot.

Note: Using this configuration, you are setting Nmap to perform a null scan with the time template as **-T4** and all **aggressive** options enabled.

33. This will create a new profile, and will thus be added to the profile list.

Note: You can select any other scan options as per your requirements.

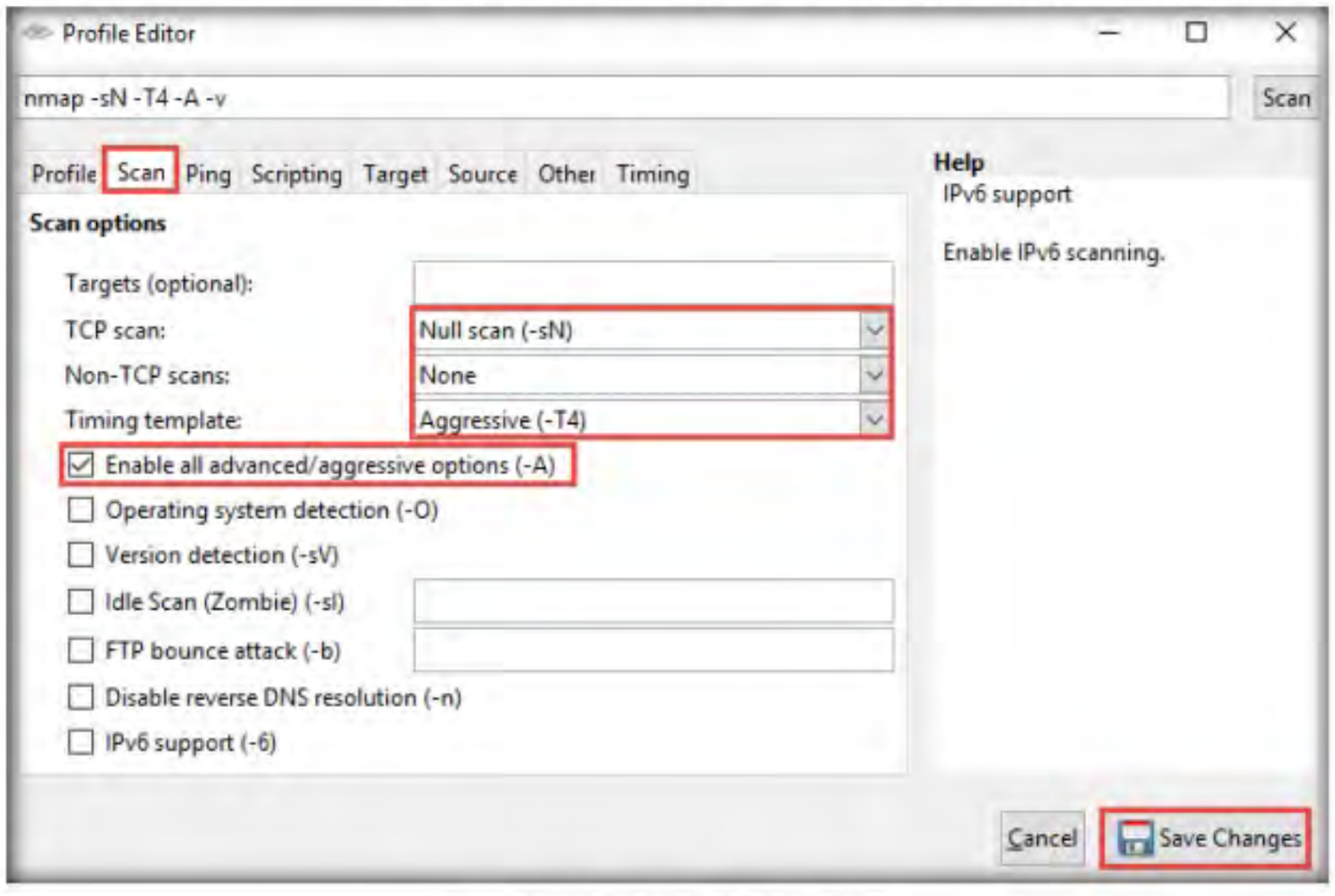


Figure 2.3.14: Configuring Null Scan Profile

34. In this task, we will be targeting the **Ubuntu** virtual machine (**10.10.10.9**). Therefore, before starting this task, turn on the **Ubuntu** virtual machine.

35. In the main window of **Zenmap**, enter the target IP address (here, **10.10.10.9**) in the **Target** field to scan. Select the **Null Scan** profile, which you created from the **Profile** drop-down list, and then click **Scan**.

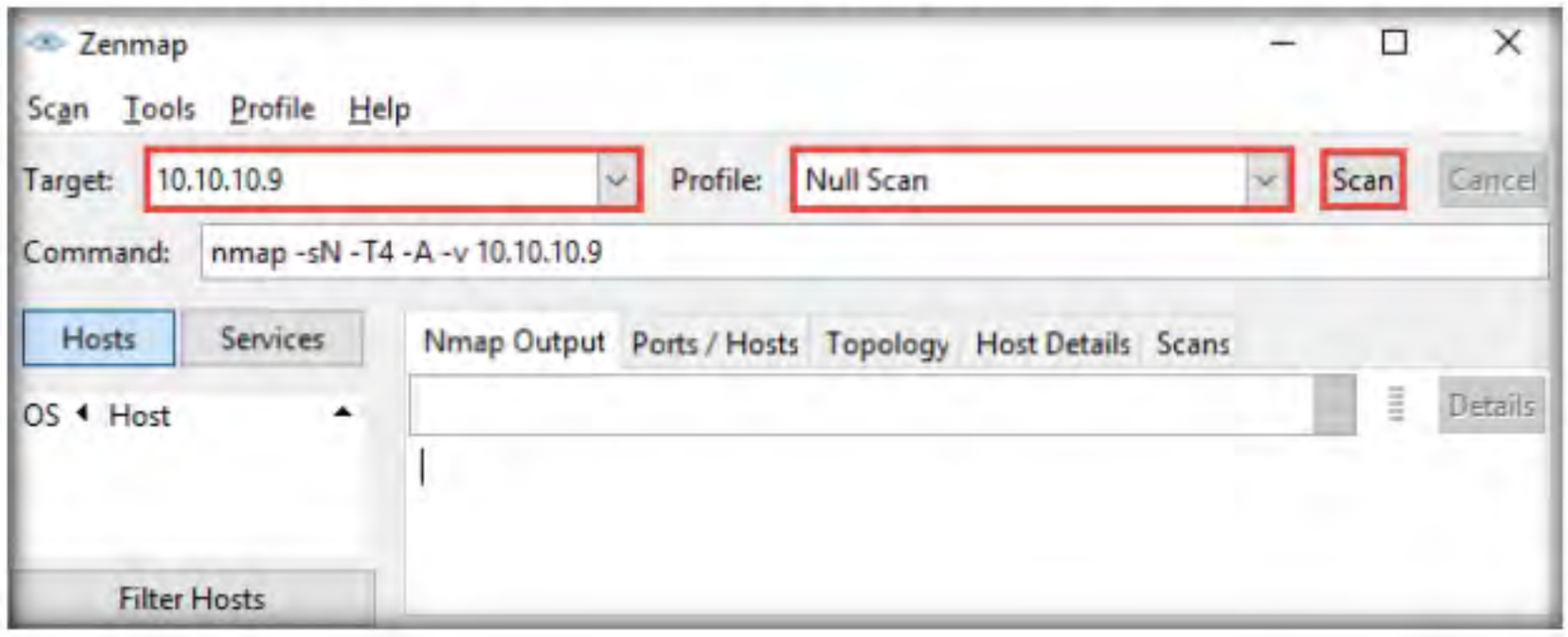


Figure 2.3.15: Initiating Null Scan

36. Nmap scans the target and displays results in the **Nmap Output** tab, as shown in the screenshot.

Note: The results obtained in your lab environment may differ.

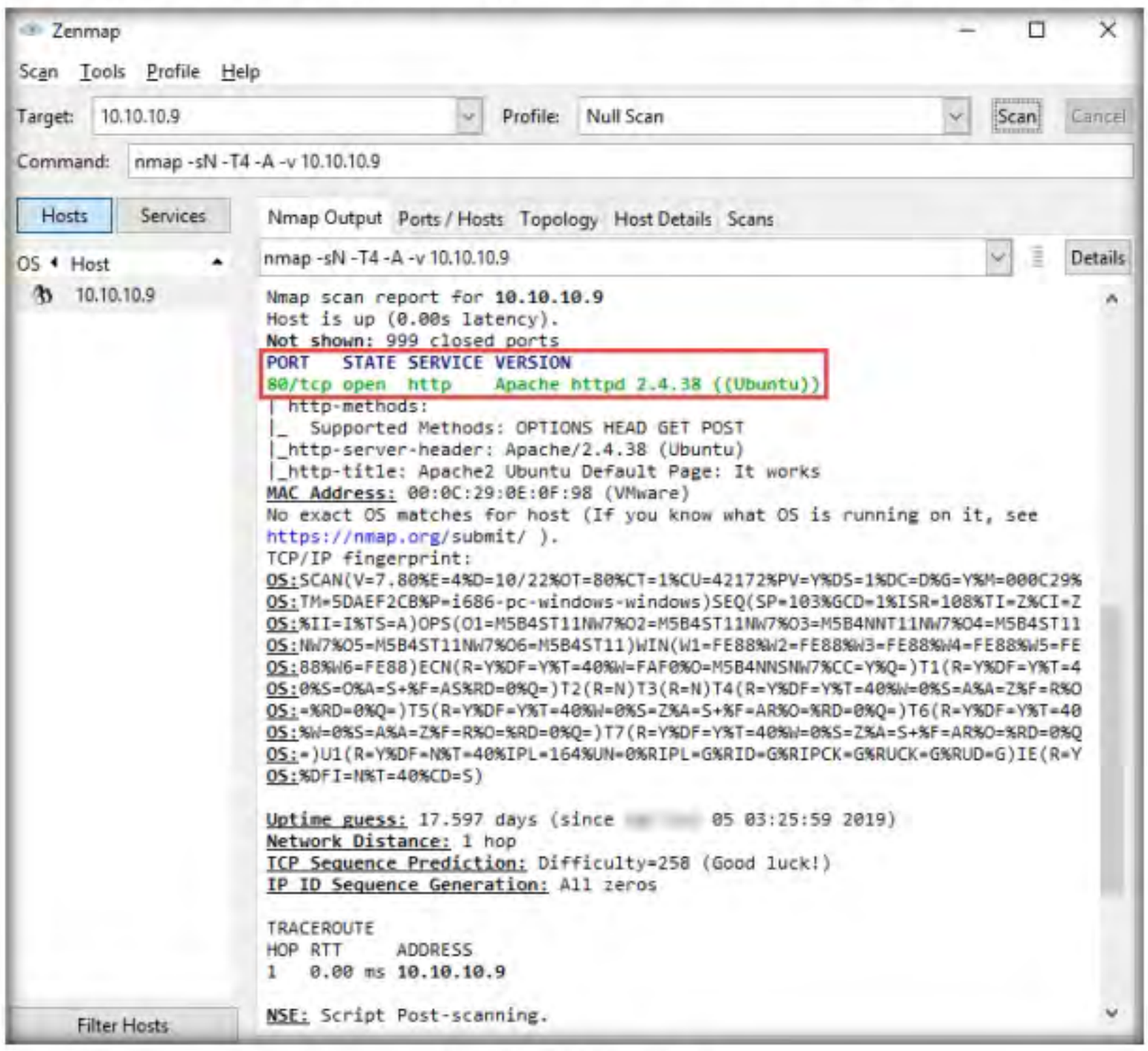


Figure 2.3.16: Null Scan Result

37. Apart from the aforementioned port scanning and service discovery techniques, you can also use the following scanning techniques to perform a port and service discovery on a target network using Nmap.

    - **IDLE/IPID Header Scan**: A TCP port scan method that can be used to send a spoofed source address to a computer to discover what services are available.

      # nmap -sI -v <target IP address>

    - **SCTP INIT Scan**: An INIT chunk is sent to the target host; an INIT+ACK chunk response implies that the port is open, and an ABORT Chunk response means that the port is closed.

      # nmap -sY -v <target IP address>

    - **SCTP COOKIE ECHO Scan**: A COOKIE ECHO chunk is sent to the target host; no response implies that the port is open and ABORT Chunk response means that the port is closed.

      # nmap -sZ -v <target IP address>

TASK 3.8

Perform Service Version Discovery

38. In the **Command** field, type the command **nmap -sV <Target IP Address>** (here, the target IP address is **10.10.10.16**) and click **Scan**.

Note: **-sV**: detects service versions.

39. The scan results appear, displaying that open ports and the version of services running on the ports, as shown in the screenshot.

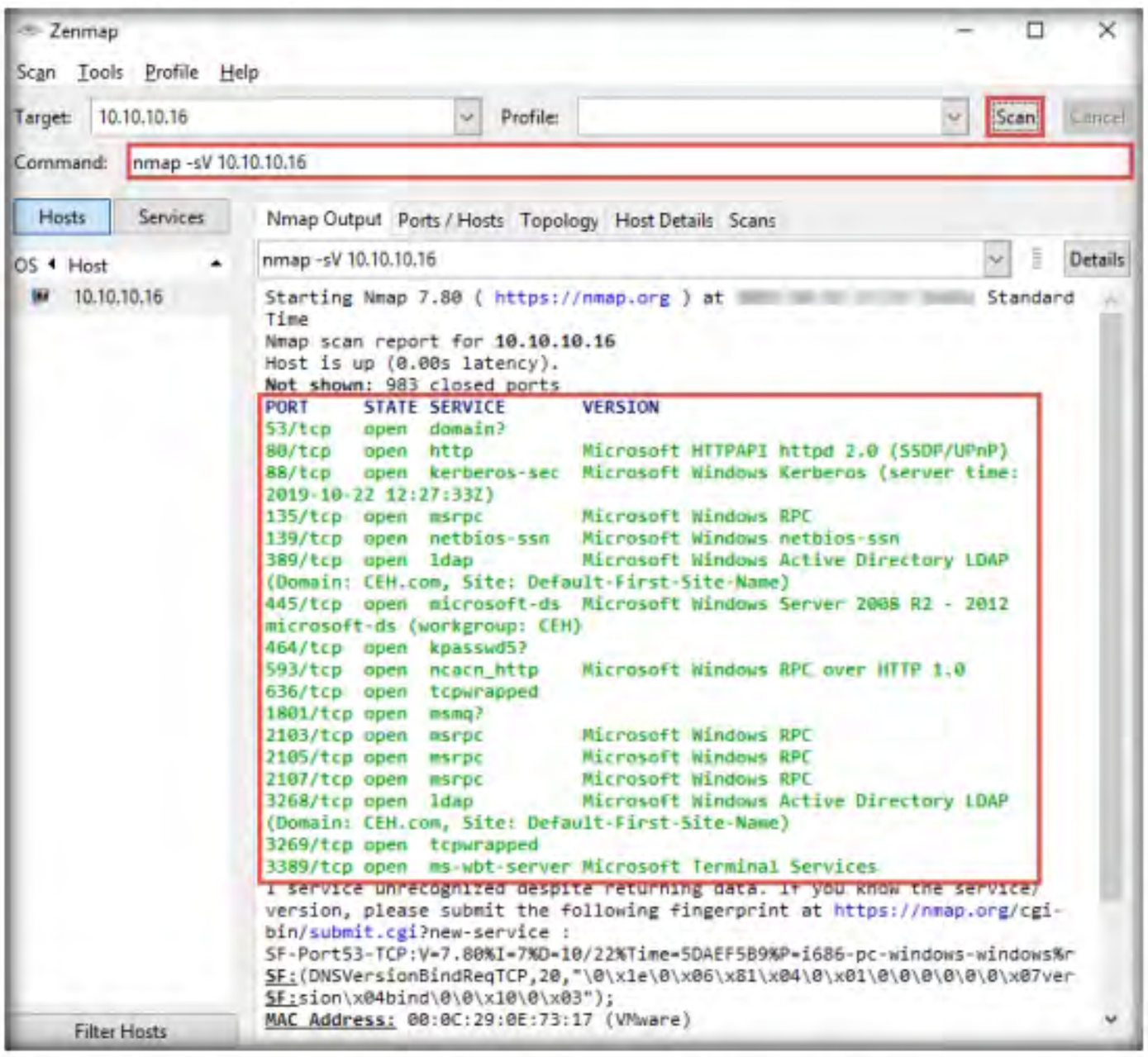


Figure 2.3.17: Zenmap scan results for Service Version Discovery

Note: Service version detection helps you to obtain information about the running services and their versions on a target system. Obtaining an accurate service version number allows you to determine which exploits the target system is vulnerable to.

40. Before starting the next sub-task, ensure that the **Parrot Security** and **Ubuntu** virtual machines are turned on.

TASK 3.9

Scan a Whole Subnet

41. In the **Command** field, type the command **nmap -A <Target Subnet>** (here, target subnet is **10.10.10.***) and click **Scan**. By providing the "*" (asterisk) wildcard, you can scan a whole subnet or IP range.

Note: **-A**: enables aggressive scan. The aggressive scan option supports OS detection (-O), version scanning (-sV), script scanning (-sC), and traceroute (--traceroute). You should not use -A against target networks without permission.

42. Nmap scans the entire network and displays information for all the hosts that were scanned, along with the open ports and services, device type, details of OS, etc. as shown in the screenshot.

Note: This scan will take some time to finish the scanning process. The results might differ in your lab environment.

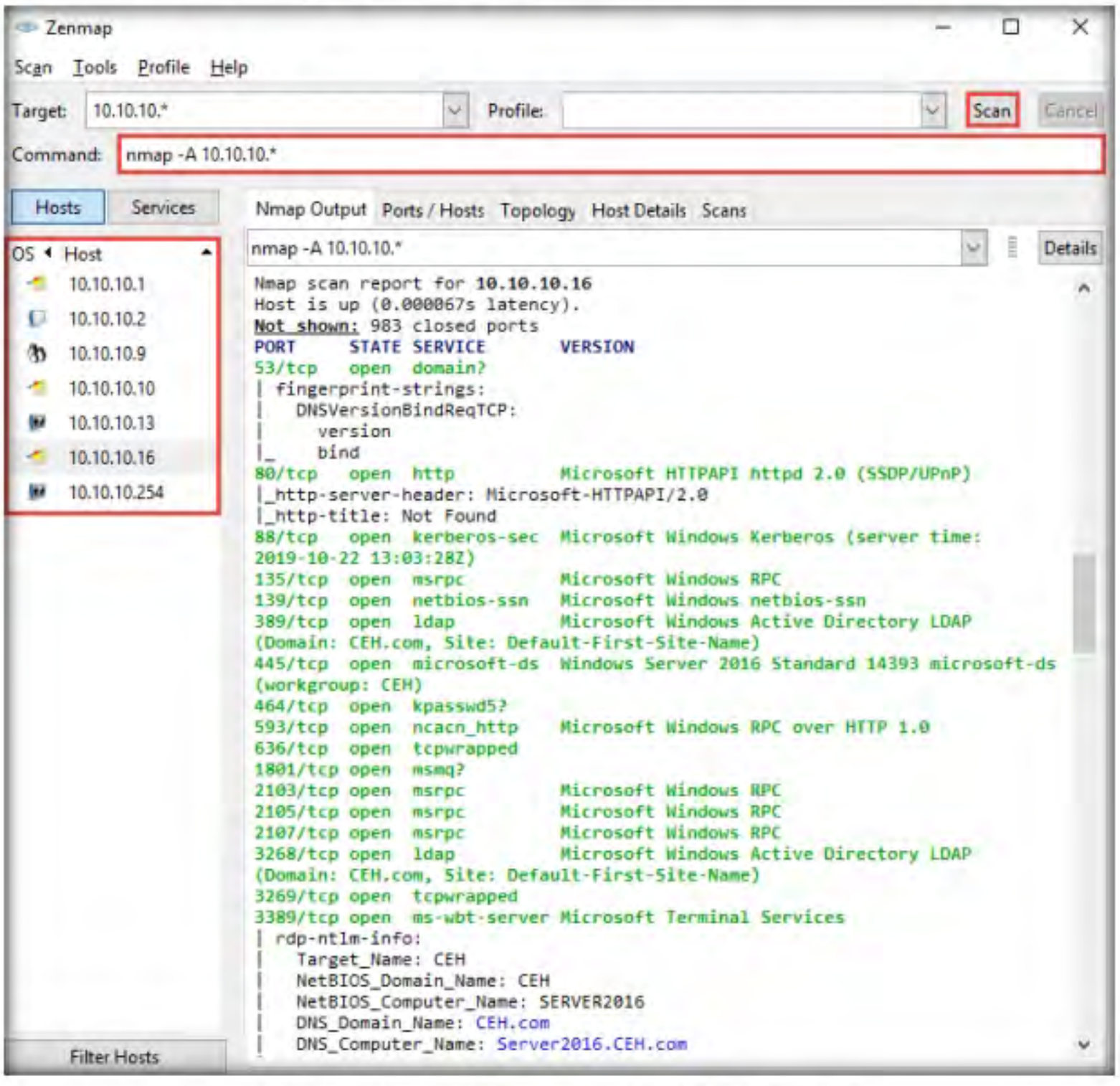


Figure 2.3.18: Zenmap displaying output for a Whole Subnet Scan

43. Choose an IP address **10.10.10.16** from the list of hosts in the left-pane and click the **Host Details** tab. This tab displays information such as **Host Status**, **Addresses**, **Operating System**, **Ports used**, **OS Classes**, etc. associated with the selected host.

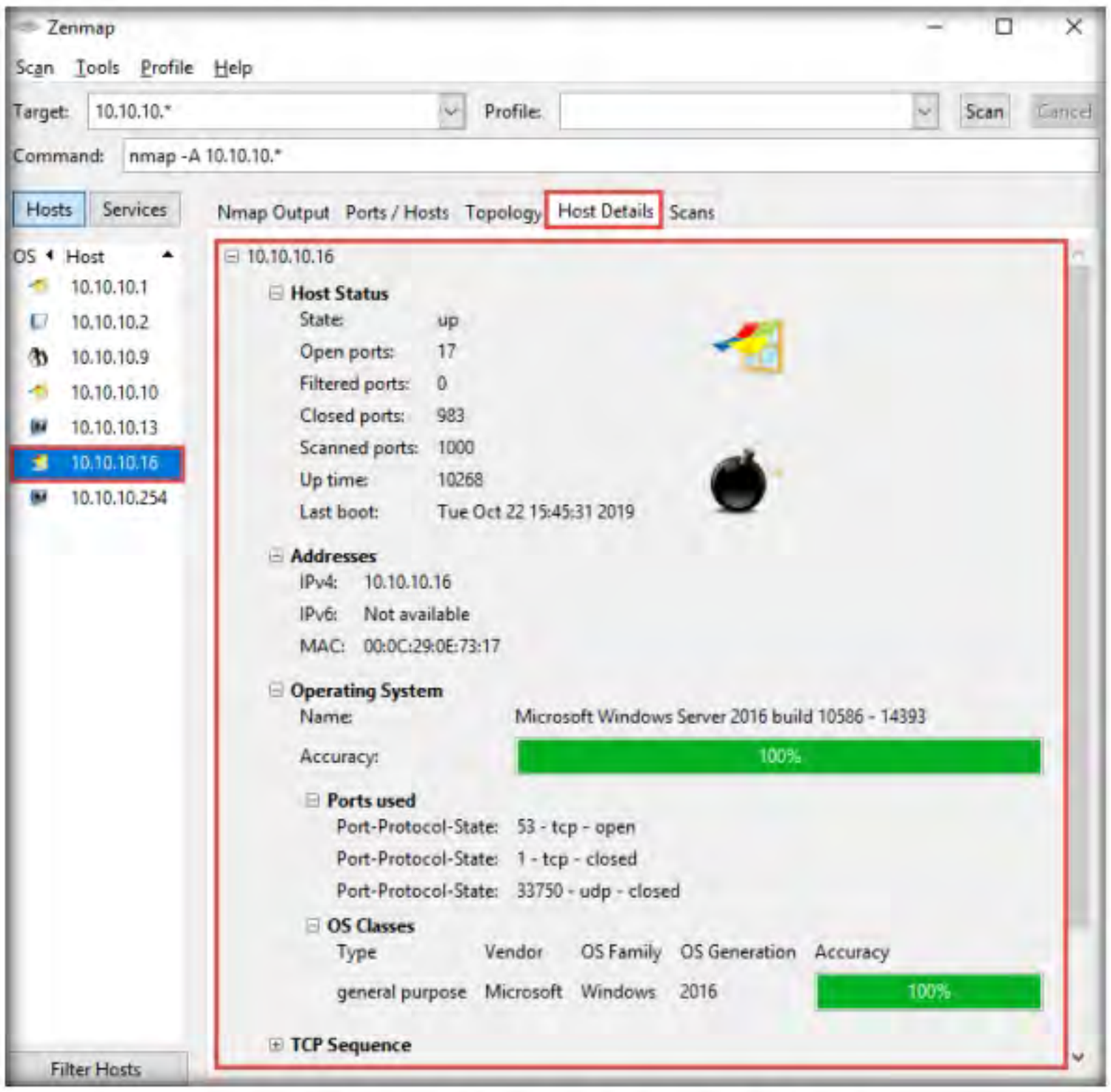


Figure 2.3.19: Zenmap displaying the Open Ports under Ports/Hosts tab

44. This concludes the demonstration of discovering target open ports, services, services versions, device type, OS details, etc. of the active hosts in the target network using various scanning techniques of Nmap.
45. Close all open windows and document all the acquired information.
46. Turn off the **Windows 10** and **Ubuntu** virtual machine.

TASK 4 Explore Various Network Scanning Techniques using Hping3

Here, we will use Hping3 to discover open ports and services running on the live hosts in the target network.

1. Switch to the **Parrot Security** virtual machine.
2. In the login page, the **attacker** username will be selected by default. Enter password as **toor** in the **Password** field and press **Enter** to log in to the machine.

Hping2/Hping3 is a command-line-oriented network scanning and packet crafting tool for the TCP/IP protocol that sends ICMP echo requests and supports TCP, UDP, ICMP, and raw-IP protocols. Using Hping, you can study the behavior of an idle host and gain information about the target such as the services that the host offers, the ports supporting the services, and the OS of the target.

Figure 2.4.1: Parrot Security login page

Note:

- If a **Parrot Updater** pop-up appears at the top-right corner of **Desktop**, ignore and close it.
- If a **Question** pop-up window appears asking you to update the machine, click **No** to close the window.

3. Click the **MATE Terminal** icon at the top of the **Desktop** window to open a **Terminal** window.

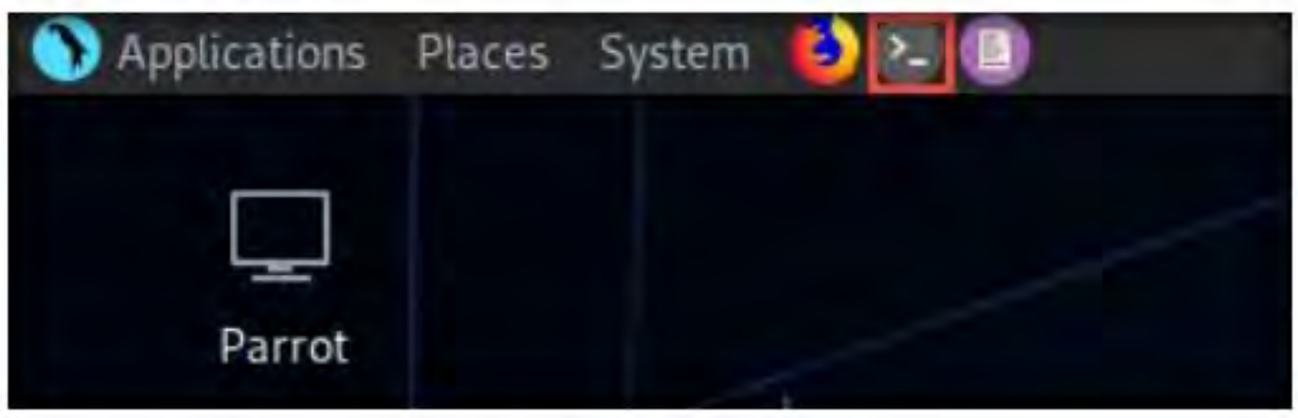


Figure 2.4.2: MATE Terminal Icon

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

Note: The password that you type will not be visible.

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

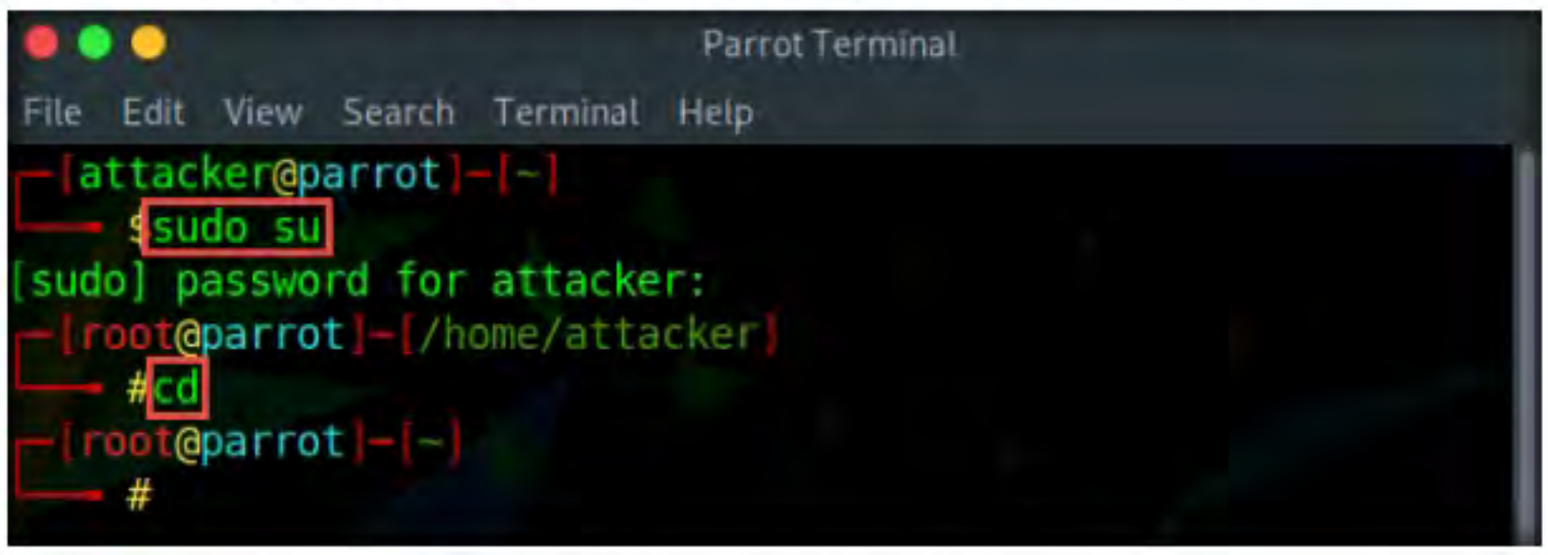


Figure 2.4.3: Running the programs as a root user

TASK 4.1

Perform ACK Scan

7. In the terminal window, type **hping3 -A <Target IP Address> -p 80 -c 5** (here, the target machine is Windows Server 2016 [**10.10.10.16**]) and press **Enter**.

Note: In this command, **-A** specifies setting the ACK flag, **-p** specifies the port to be scanned (here, **80**), and **-c** specifies the packet count (here, **5**).

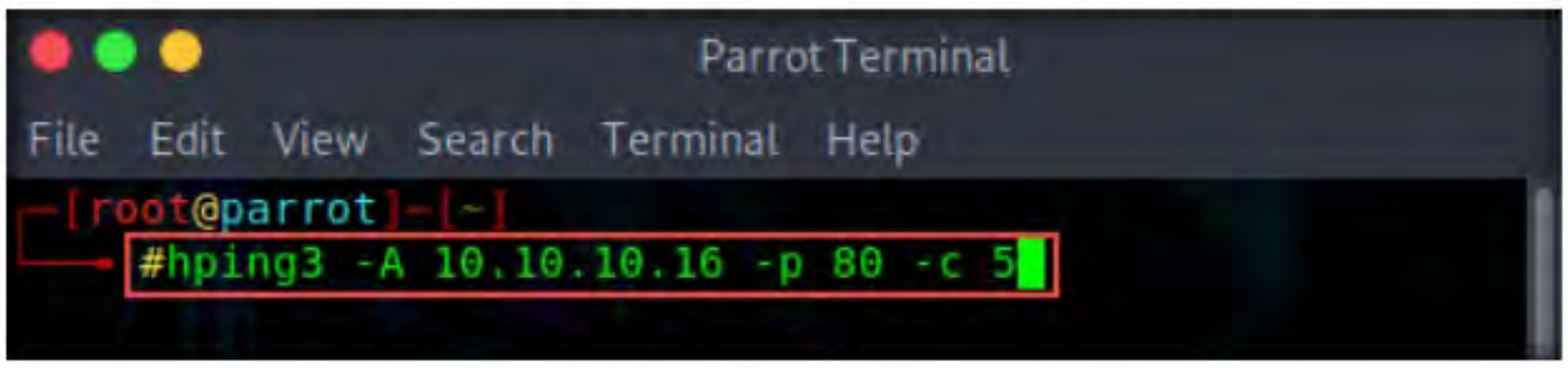


Figure 2.4.4: Hping3 TCP ACK scan command

8. In a result, the number of packets sent and received is equal, indicating that the respective port is open, as shown in the screenshot.

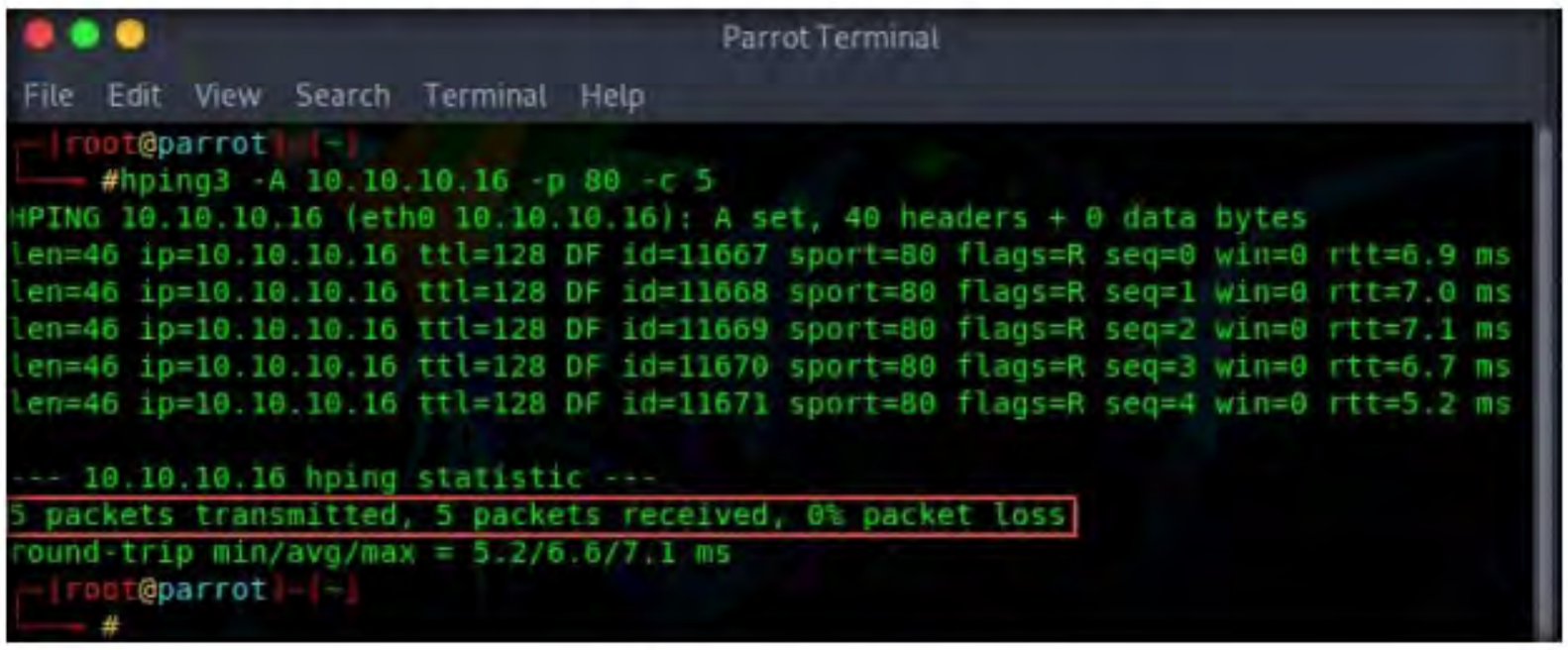


Figure 2.4.5: Hping3 ACK scan result

Note: The ACK scan sends an ACK probe packet to the target host; no response means that the port is filtered. If an RST response returns, this means that the port is closed.

TASK 4.2

Perform SYN Scan

9. In the terminal window, type **hping3 -8 0-100 -S <Target IP Address> -V** (here, the target machine is Windows Server 2016 [**10.10.10.16**]) and press **Enter**.

Note: In this command, **-8** specifies a scan mode, **-p** specifies the range of ports to be scanned (here, **0-100**), and **-V** specifies the verbose mode.

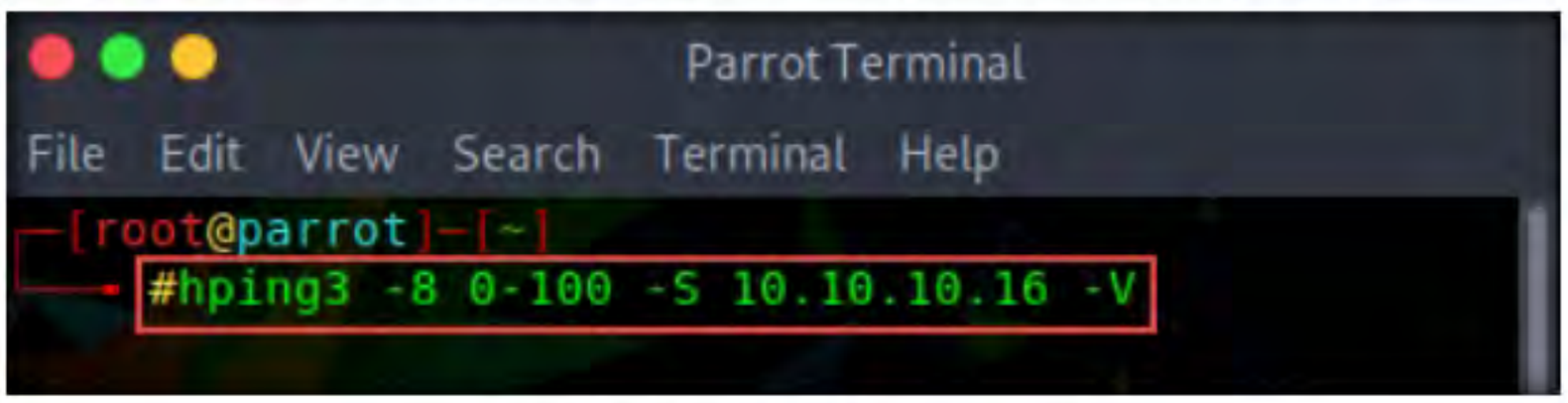


Figure 2.4.6: Hping3 TCP SYN scan command

10. The result appears, displaying the open ports along with the name of service running on each open port, as shown in the screenshot.

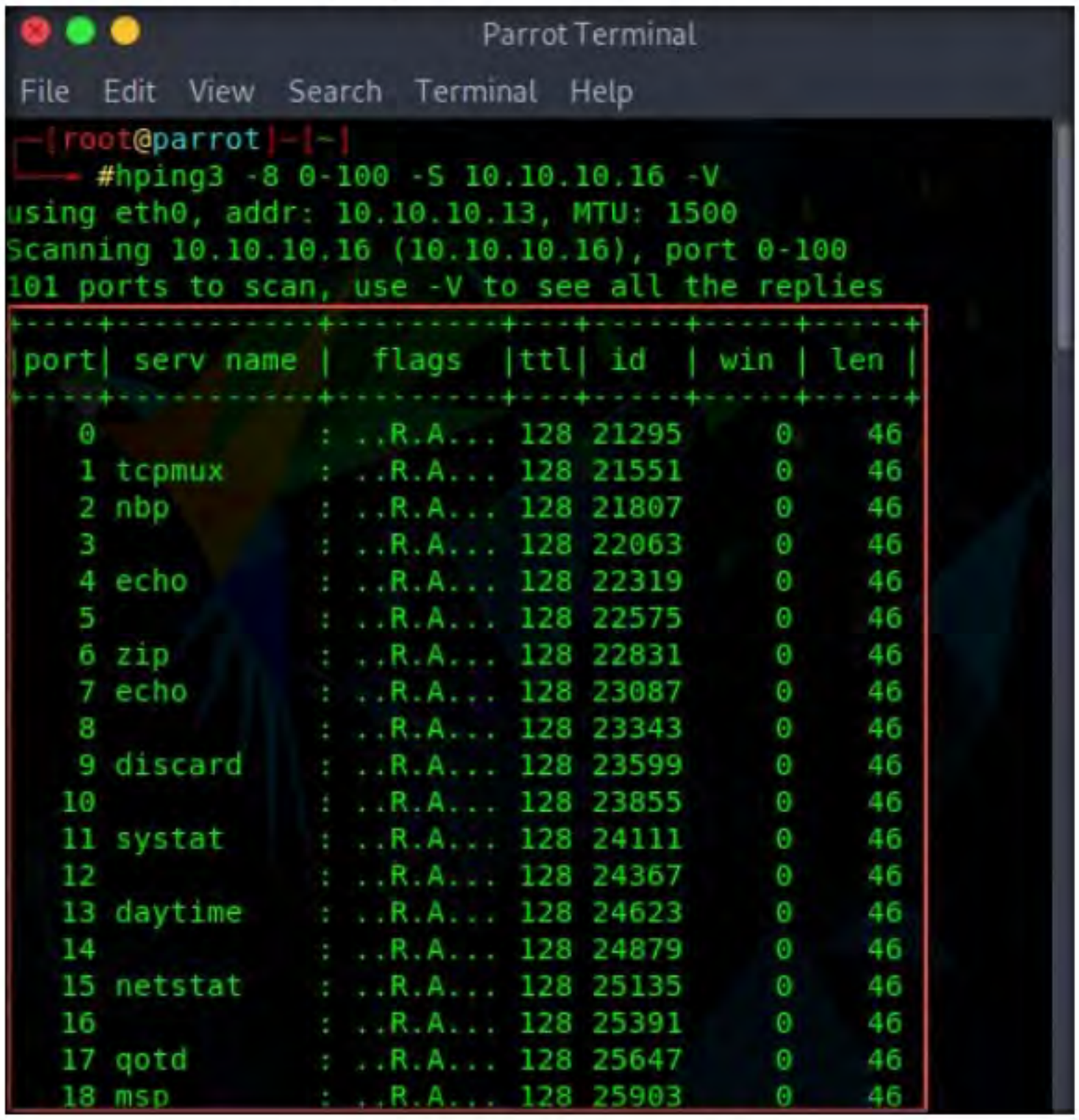


Figure 2.4.7: Hping3 SYN scan result

Note: The SYN scan principally deals with three of the flags: SYN, ACK, and RST. You can use these three flags for gathering illegal information from servers during the enumeration process.

TASK 4.3

Perform FIN, PUSH, and URG Scan

11. In the terminal window, type **hping3 -F -P -U <Target IP Address> -p 80 -c 5** (here, the target machine is Windows Server 2016 [**10.10.10.16**]) and press **Enter**.

Note: In this command, **-F** specifies setting the FIN flag, **-P** specifies setting the PUSH flag, **-U** specifies setting the URG flag, **-c** specifies the packet count (here, **5**), and **-p** specifies the port to be scanned (here, **80**).

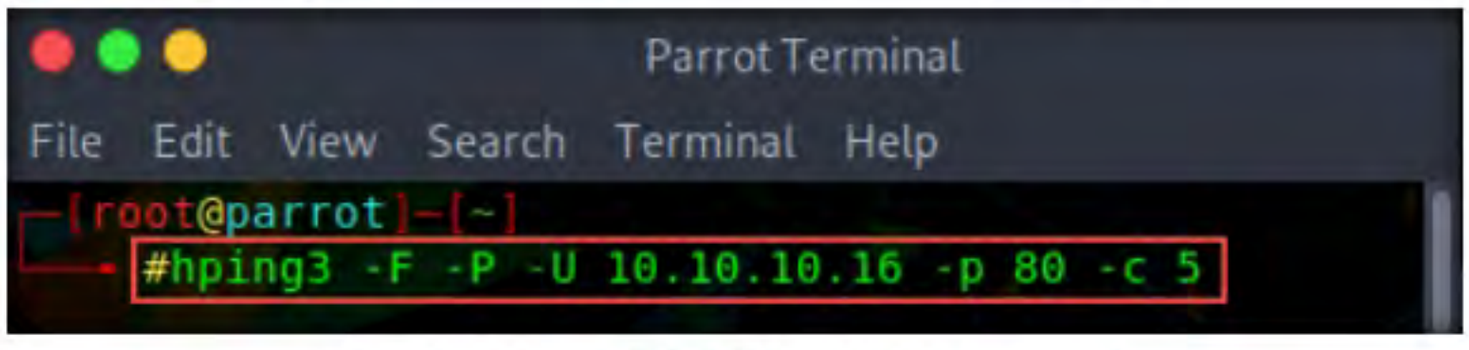


Figure 2.4.8: Hping3 FIN, PUSH, and URG scan command

12. The results demonstrate that the number of packets sent and received is equal, thereby indicating that the respective port is open, as shown in the screenshot.

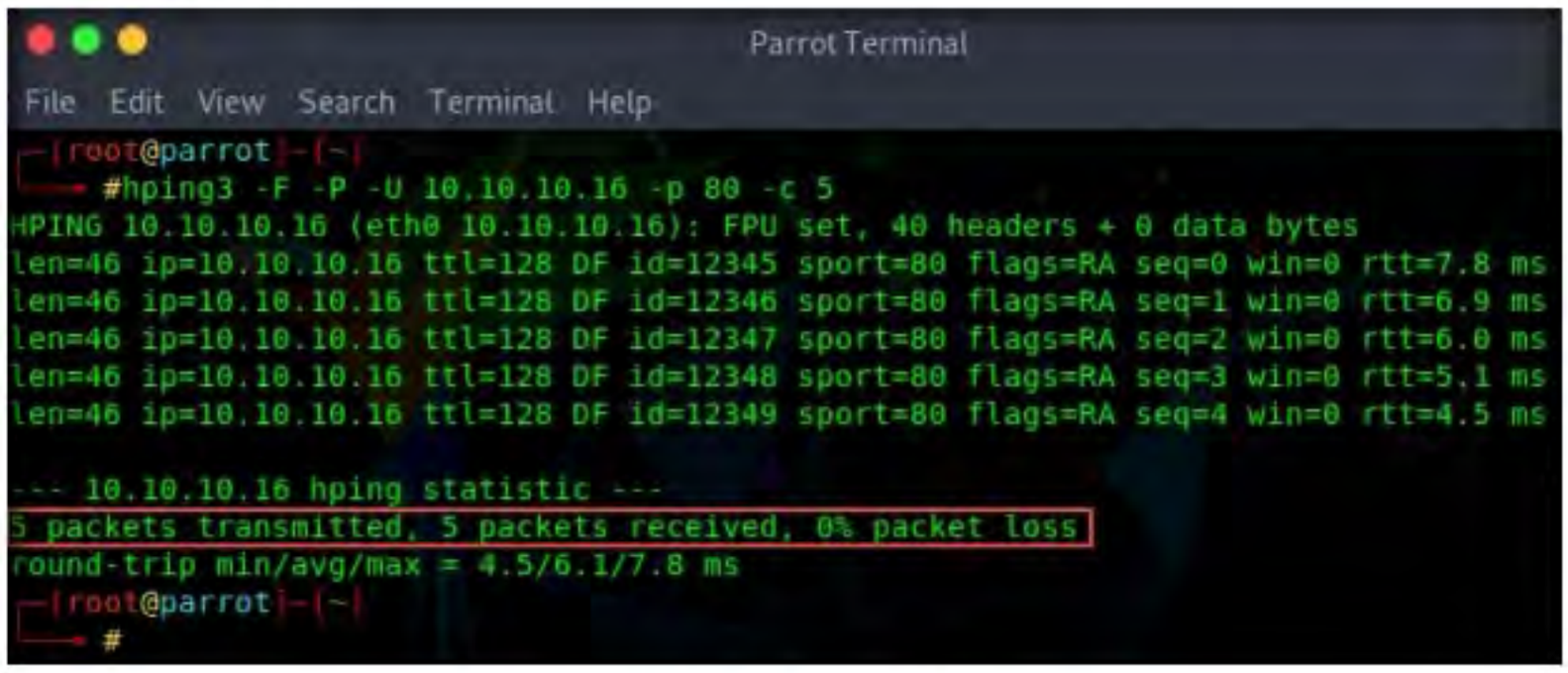


Figure 2.4.9: Hping3 FIN, PUSH, and URG scan result

Note: FIN, PUSH, and URG scan the port on the target IP address. If a port is open on the target, you will not receive a response. If the port is closed, Hping will return an RST response.

TASK 4.4

Perform TCP Stealth Scan

13. In the terminal window, type **hping3 --scan 0-100 -S <Target IP Address>** (here, the target machine is Windows Server 2016 [**10.10.10.16**]) and press **Enter**.

Note: In this command, **--scan** specifies the port range to scan, **0-100** specifies the range of ports to be scanned, and **-S** specifies setting the SYN flag.

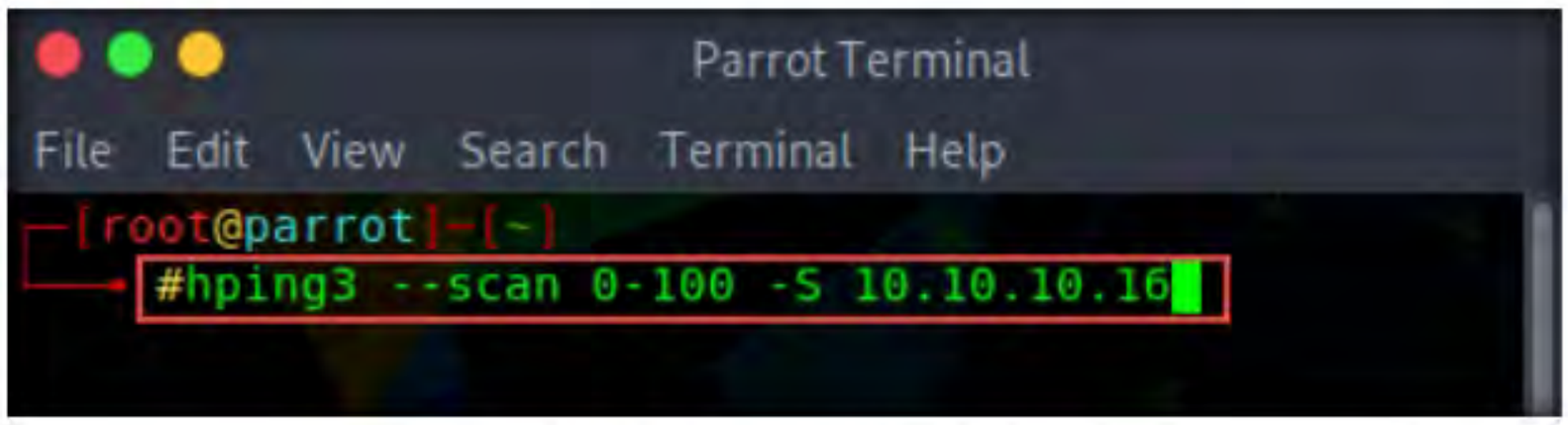


Figure 2.4.10: Hping3 TCP stealth scan command

14. The result appears displaying the open ports and names of the services running on the target IP address, as shown in the screenshot.

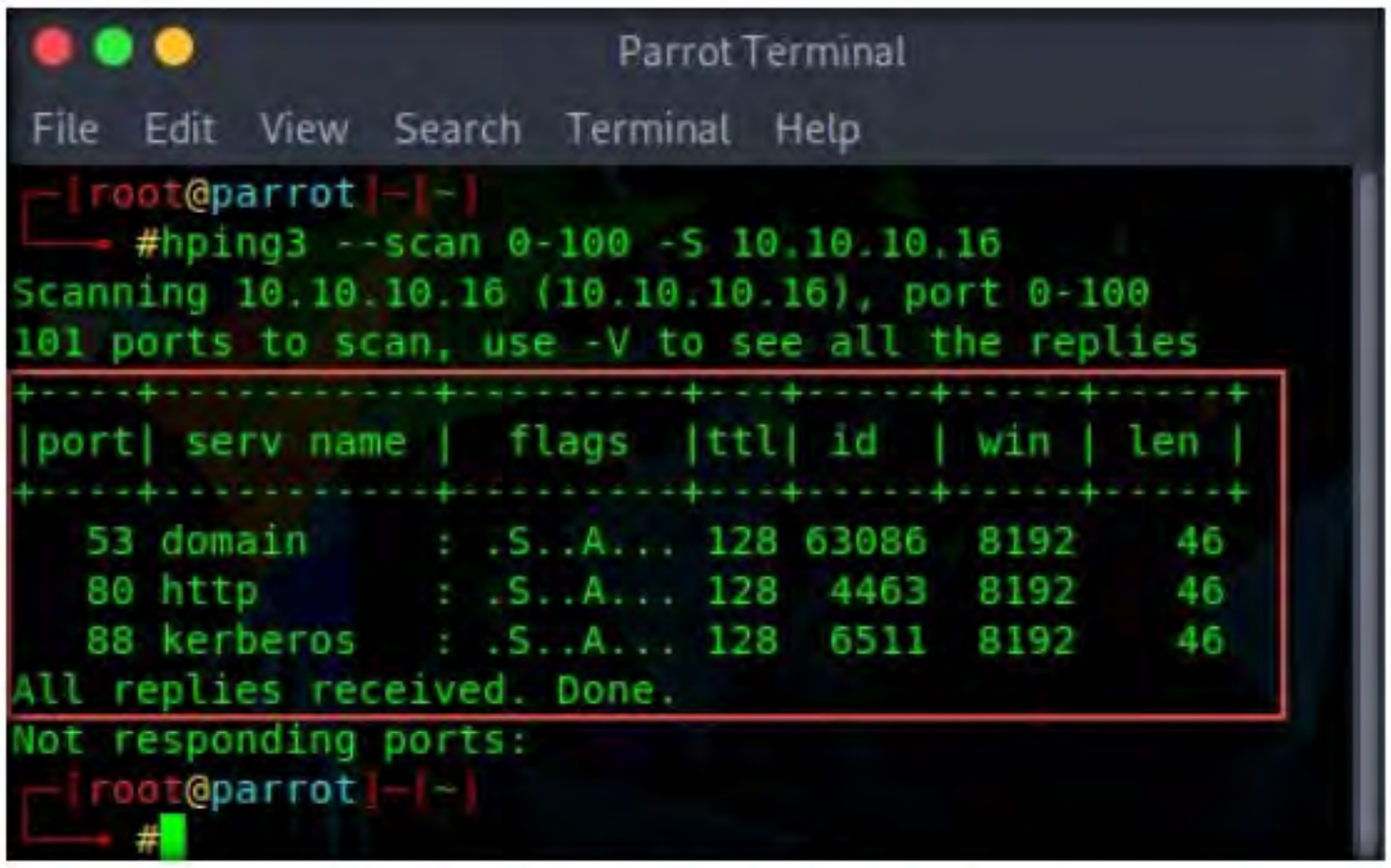


Figure 2.4.11: Hping3 TCP stealth scan result

Note: In the TCP stealth scan, the TCP packets are sent to the target host; if a SYN+ACK response is received, it indicates that the ports are open.

15. Apart from the aforementioned port scanning and service discovery techniques, you can also use the following scanning techniques to perform a port and service discovery on a target network using Hping3.
    - ICMP scan: **hping3 -1 <Target IP Address> -p 80 -c 5**
    - Entire subnet scan for live host: **hping3 -1 <Target Subnet> –rand-dest -I eth0**
    - UDP scan: **hping3 -2 <Target IP Address> -p 80 -c 5**
16. This concludes the demonstration of discovering open ports and services running on the live hosts in the target network using Hping3.
17. Close all open windows and document all the acquired information.
18. Turn off the **Parrot Security** and **Windows Server 2016** virtual machine.

Lab Analysis

Analyze and document all the results discovered in the lab exercise.

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS ABOUT THIS LAB.

Internet Connection Required	
☐ Yes	☑ No
Platform Supported	
☑ Classroom	☑ iLabs

Lab 3

Perform OS Discovery

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

ICON KEY

- Valuable information
- Test your knowledge
- Web exercise
- Workbook review

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

Lab Environment

To carry out this lab, you need:

- Windows 10 virtual machine
- Windows Server 2016 virtual machine
- Parrot Security virtual machine
- Ubuntu virtual machine
- Web browsers with an Internet connection
- Administrator privileges to run the tools
- Wireshark located at **E:\CEH-Tools\CEHv11 Module 03 Scanning Networks\Banner Grabbing Tools\Wireshark**
- You can also download the latest version of Wireshark from its official website. If you decide to download the latest version, the screenshots shown in the lab might differ.

Tools demonstrated in this lab are available in E:\CEH-Tools\CEHv11 Module 03 Scanning Networks

Lab Duration

Time: 15 Minutes

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
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

Lab Tasks

TASK 1

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

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

1. Before beginning this lab, turn on the **Windows 10**, **Windows Server 2016**, and **Ubuntu** virtual machine.
2. In the **Windows 10** virtual machine log in with the credentials Username: **Admin** and Password: **Pa$$w0rd**.

TASK 1.1

Install Wireshark

3. Navigate to **E:\CEH-Tools\CEHv11 Module 03 Scanning Networks\Banner Grabbing Tools\Wireshark** and double-click **Wireshark-win64-3.0.5.exe**.

Note: If a **User Account Control** window appears, click **Yes**.

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.

4. **Wireshark Setup** window appears; follow the steps to install Wireshark using all default settings.

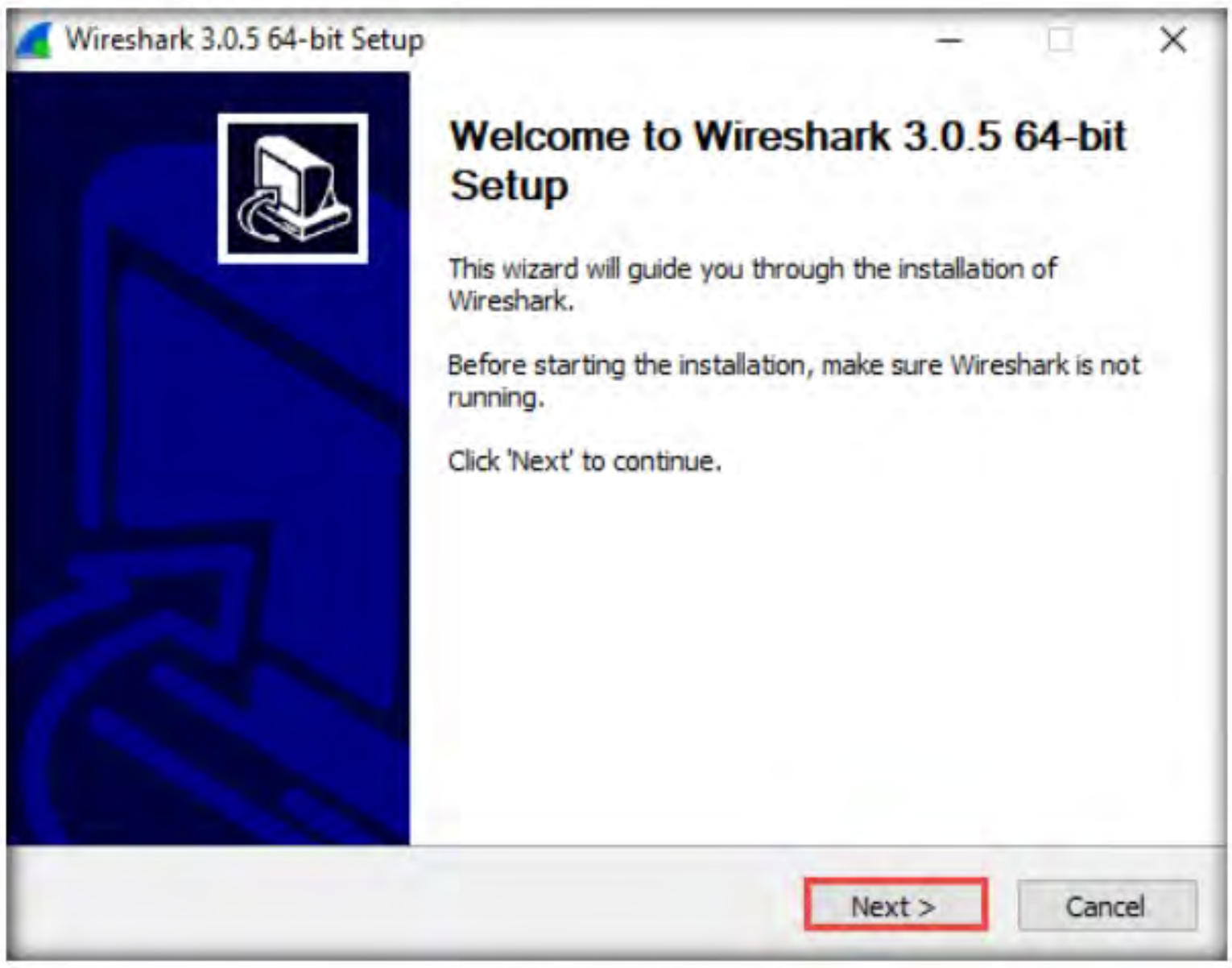


Figure 3.1.1: Wireshark Setup window

5. In the **Packet Capture** wizard, ensure that the **Install Npcap 0.9983** checkbox is selected and click **Next**.

Note: The version of Npcap may differ in your lab environment.

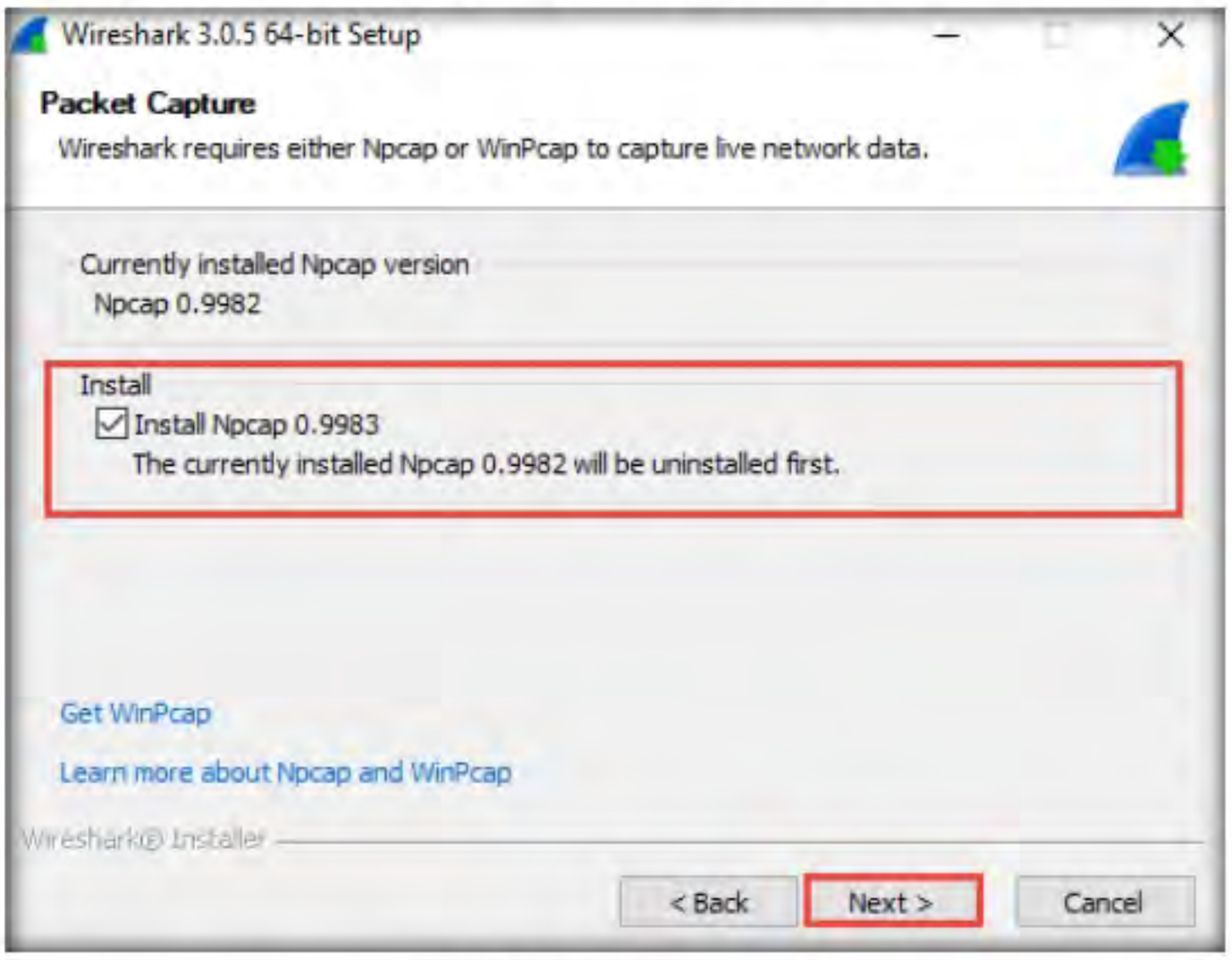


Figure 3.1.2: Packet Capture wizard

6. In the **USB Capture** wizard, select the **Install USBPcap 1.3.0.0** checkbox and click **Install**.

Note: The version of USBPcap may differ in your lab environment.

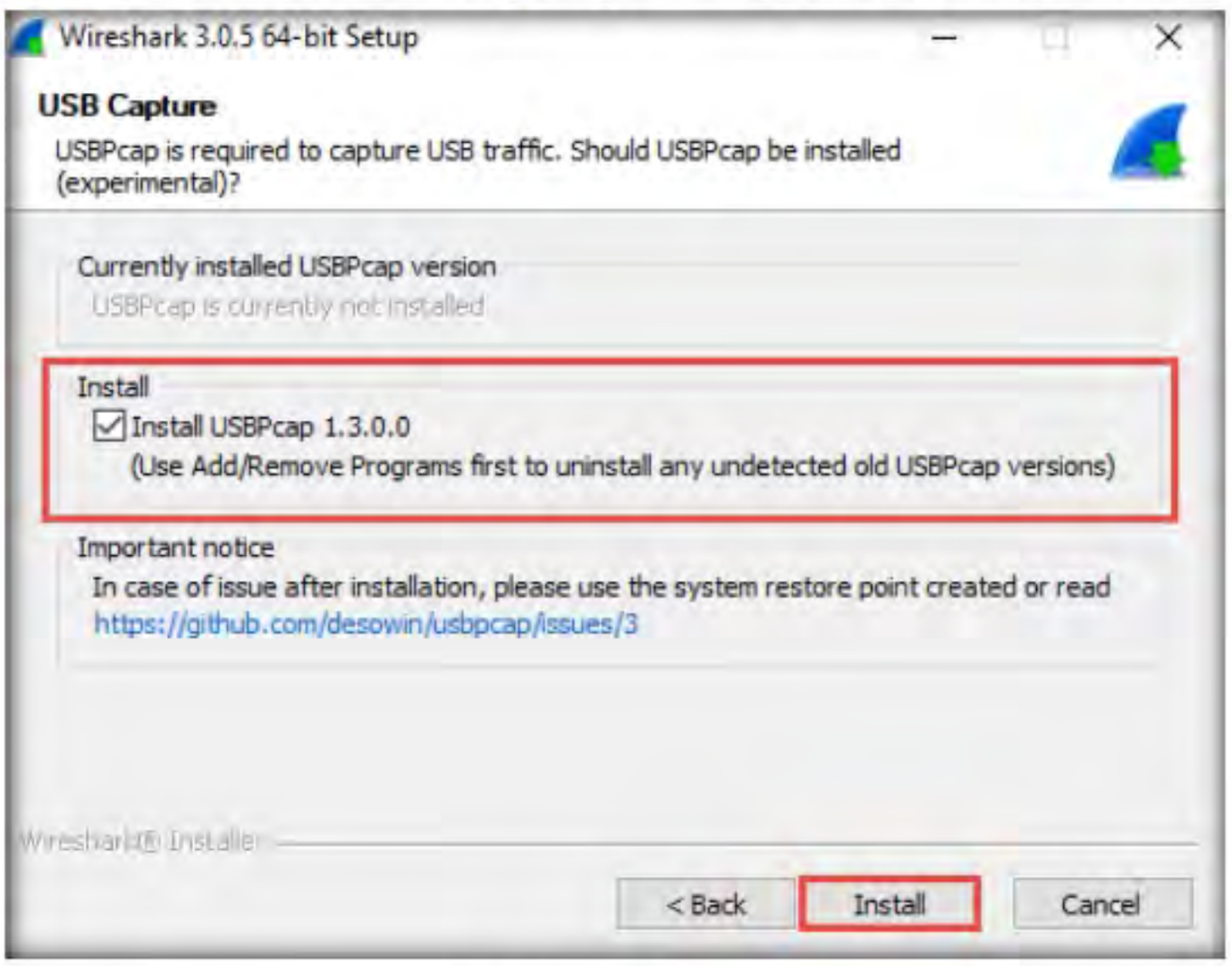


Figure 3.1.3: USB Capture wizard

7. Follow the wizard steps to install Npcap and USBPcap using all default settings.
8. After the completion of the installation, in the **Wireshark Setup** windows, ensure that the **Reboot now** radio button is selected, and click **Finish**.

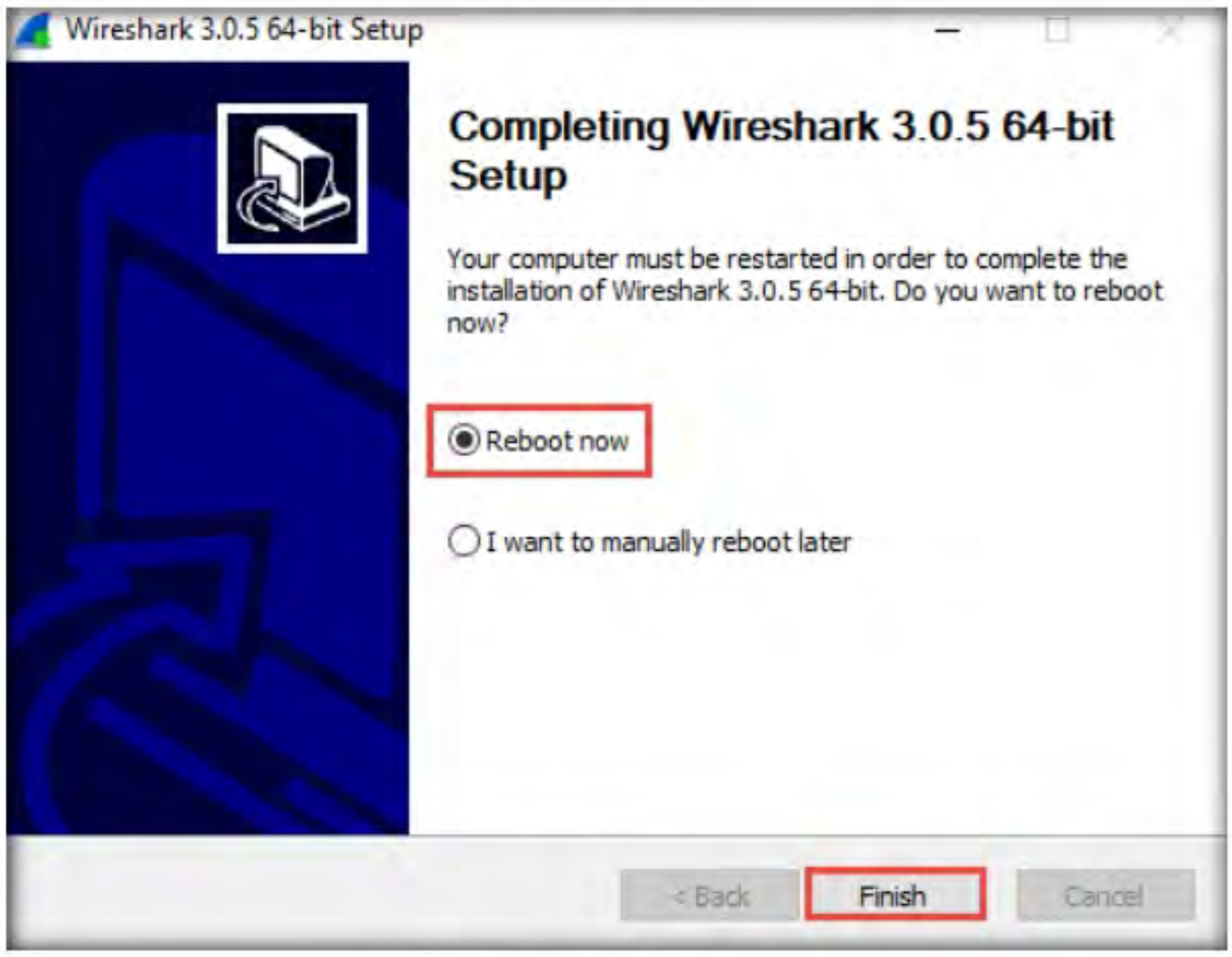


Figure 3.1.4: Wireshark setup window

9. The **Windows 10** virtual machine will restart. Log in with the credentials **Admin/Pa$$w0rd**.
10. Click on the **Start** menu and launch **Wireshark** from the applications.

TASK 1.2

Run Wireshark

11. The **Wireshark Network Analyzer** main window appears; double-click the available ethernet or interface (here, **Ethernet0**) to start the packet capture, as shown in the screenshot.

Note: The available interface might vary in your lab environment.

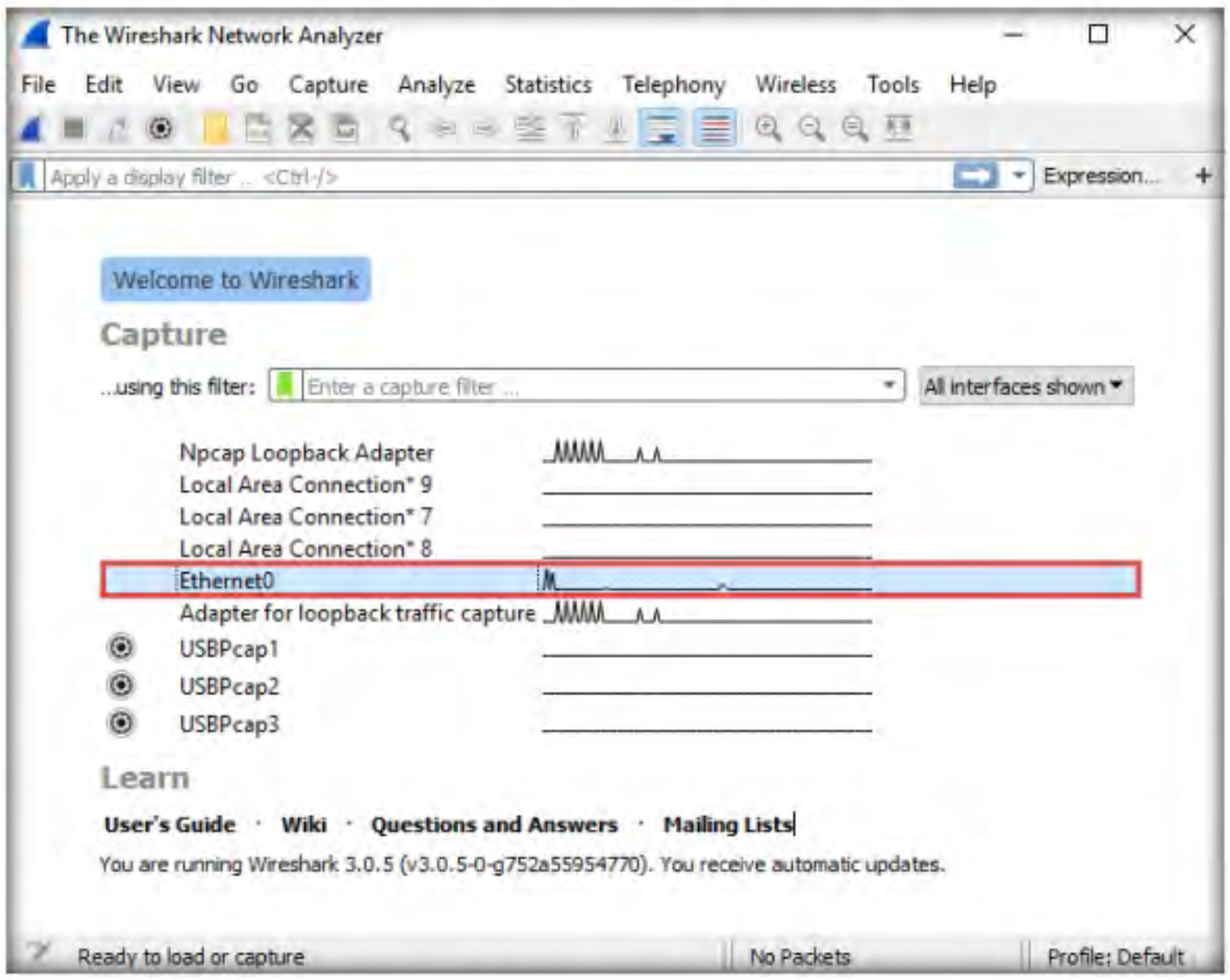


Figure 3.1.5: Wireshark main window, enabling to start the capture

TASK 1.3

Perform Ping on Target Machine (Windows)

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

Note: **10.10.10.16** is the IP address of the **Windows Server 2016** virtual machine; this may vary in your lab environment.

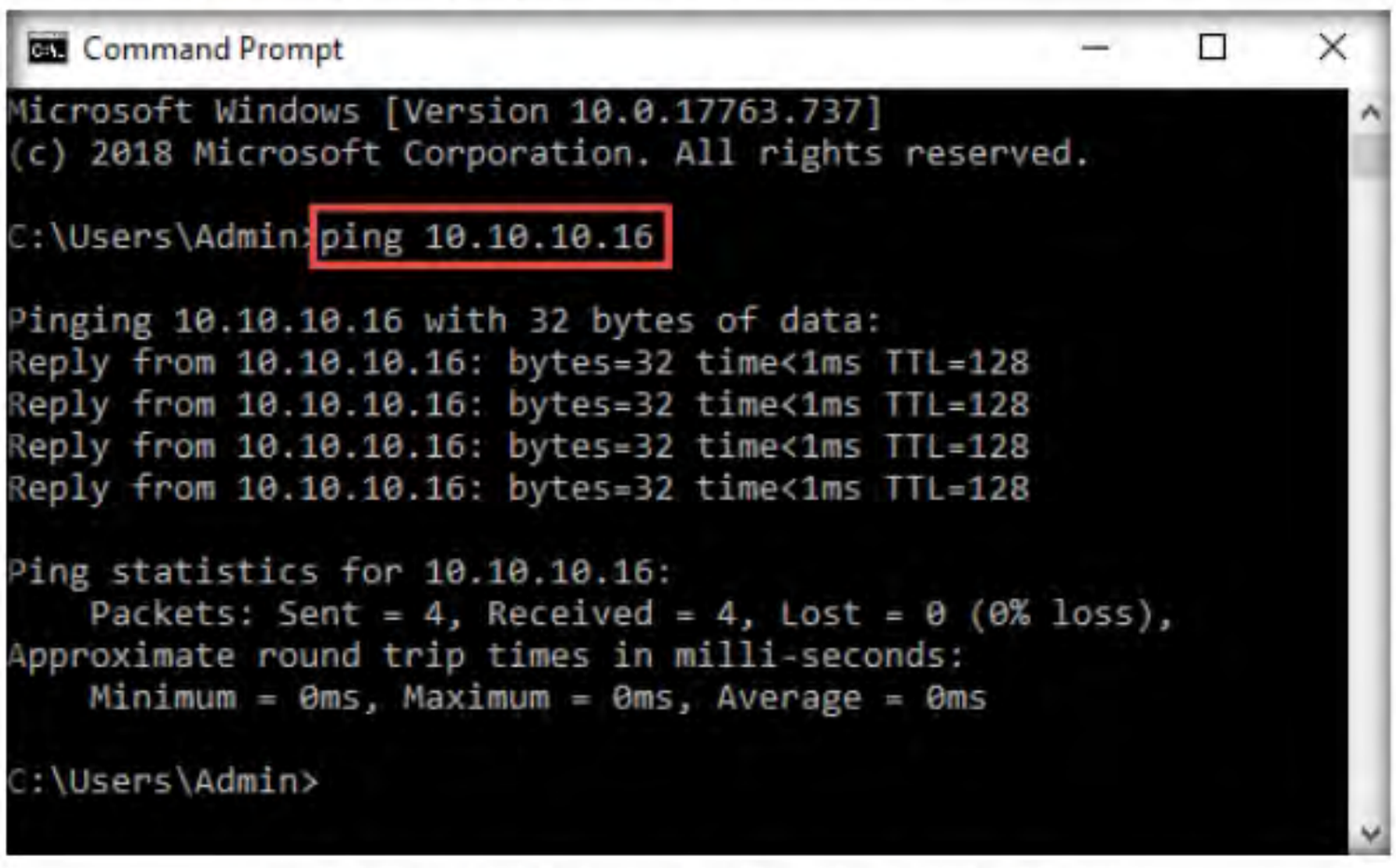


Figure 3.1.6: Sending ICMP requests to Windows Server 2016 virtual machine

TASK 1.4

Analyze the TTL Value

13. Observe the packets captured by **Wireshark**.

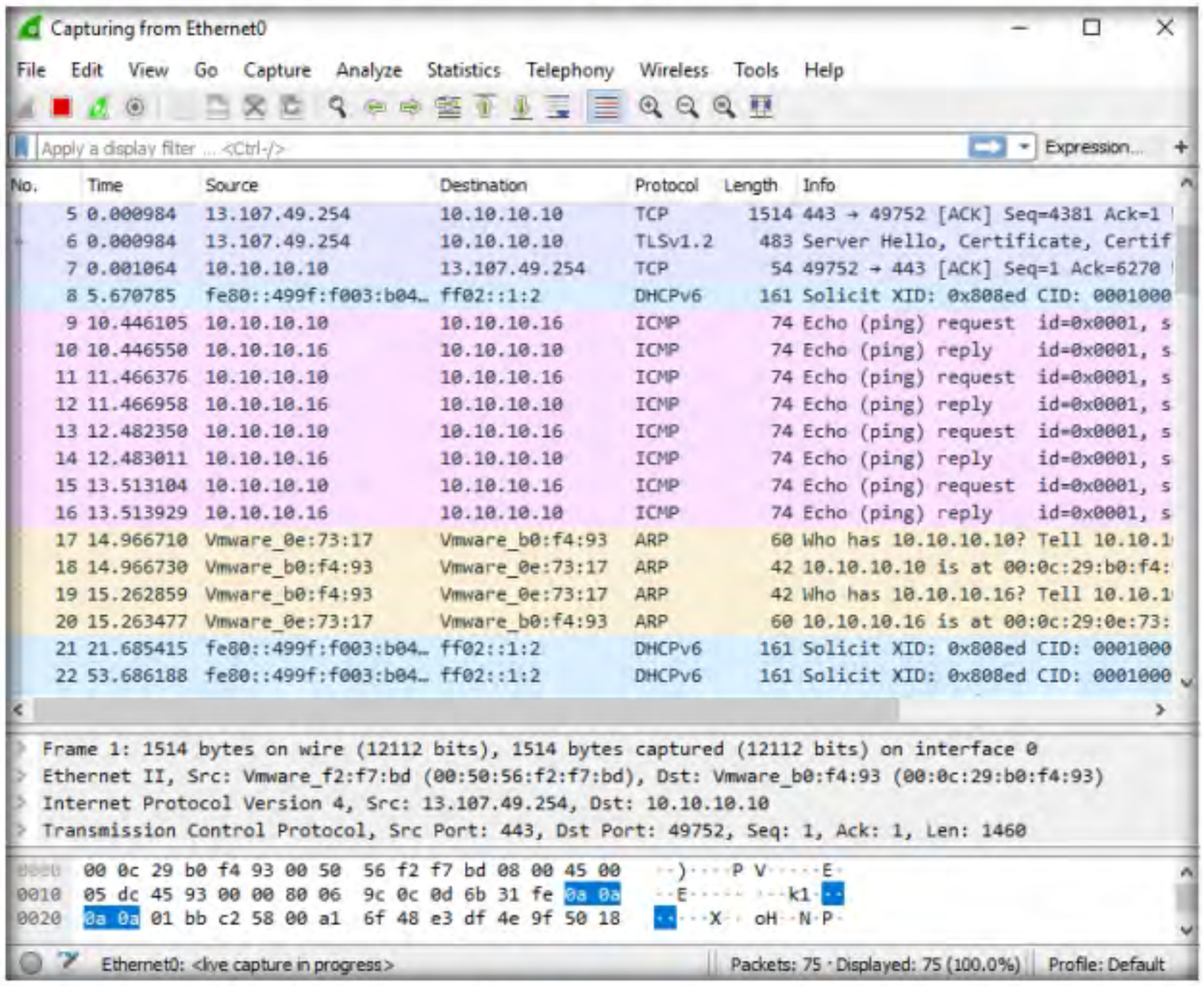


Figure 3.1.7: Packets Captured by Wireshark

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

Note: The IP address may vary in your lab environment.

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

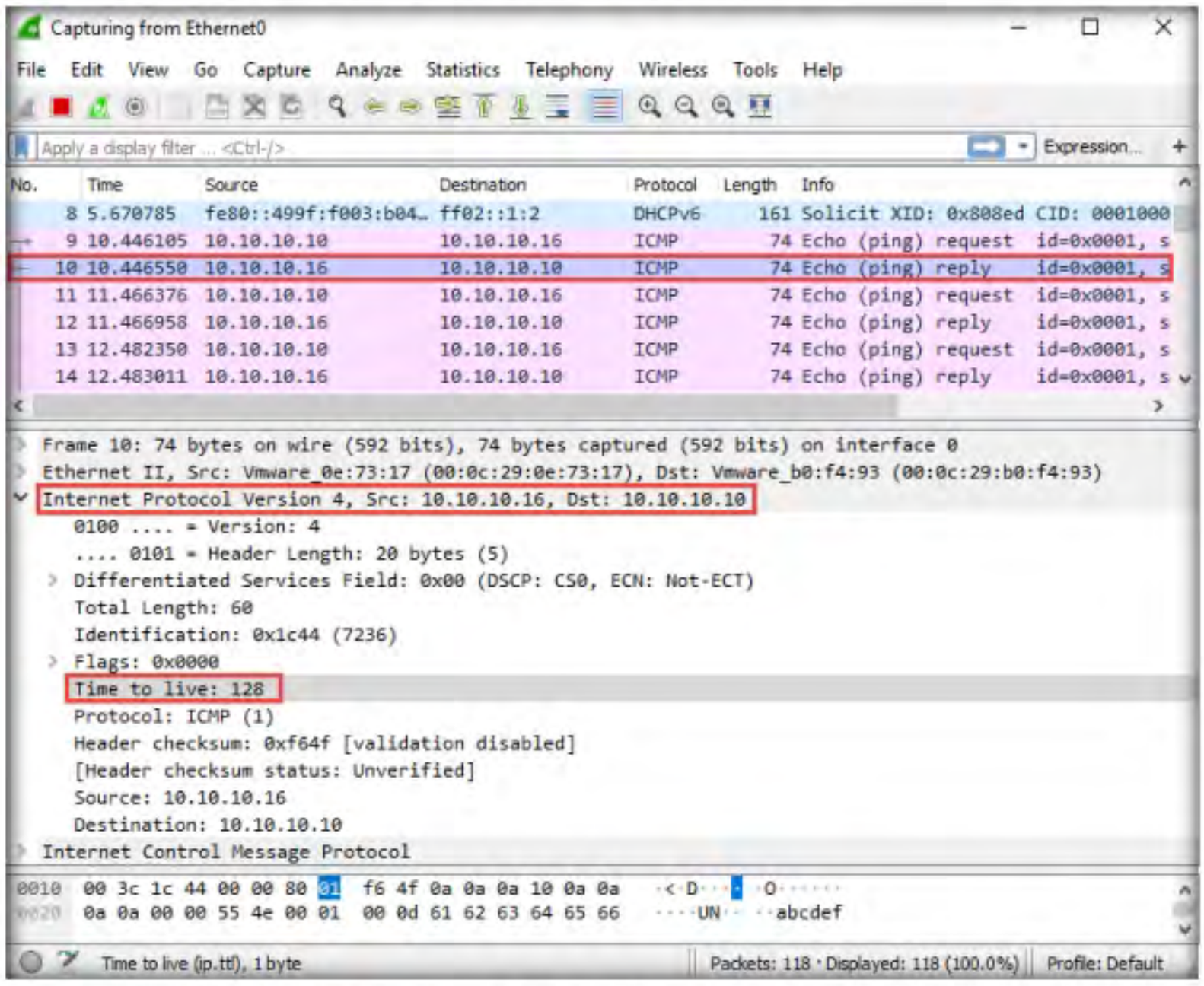


Figure 3.1.8: TTL value detected by Wireshark for Windows Server 2016 virtual machine

16. Now, stop the capture in the **Wireshark** window by clicking on the **Stop** () button.

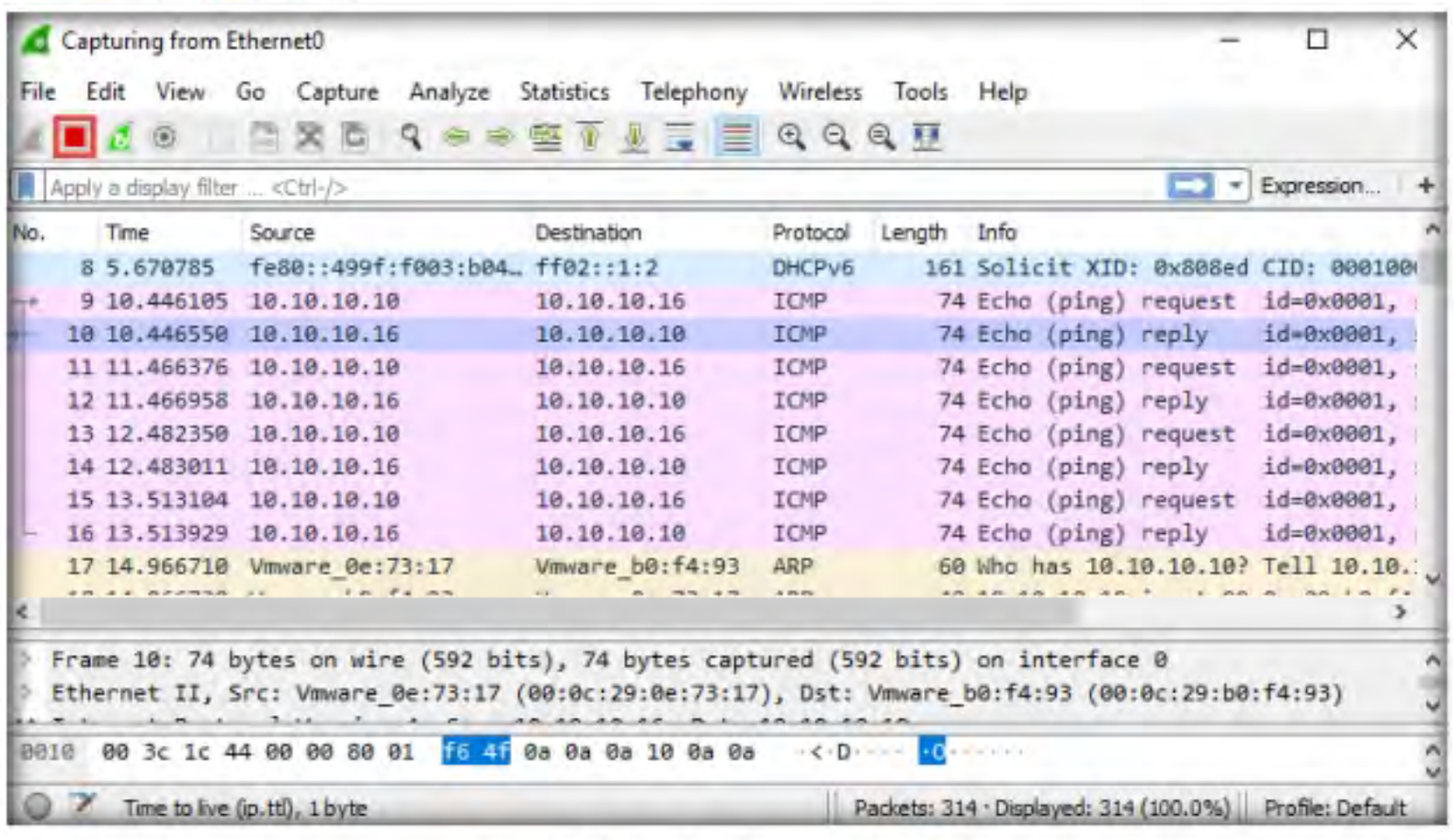


Figure 3.1.9: Stop Live Capture in Wireshark

17. Now, click the **Start capturing packets** ( ) button. If an **Unsaved packets...** pop-up appears, click **Continue without Saving**.

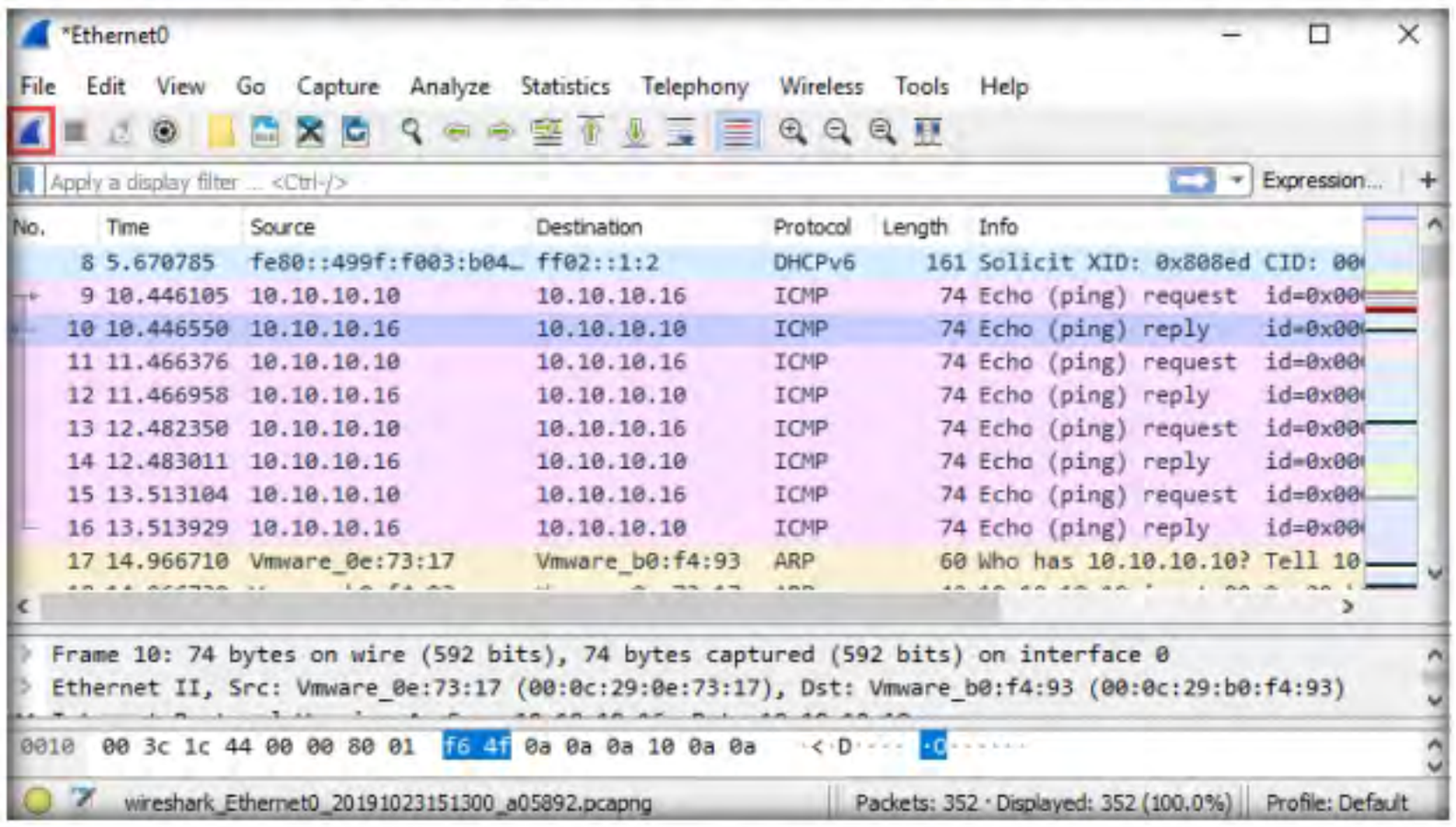


Figure 3.1.10: Start Live Capture in Wireshark

18. Wireshark will start capturing the new packets.

TASK 1.5

Perform Ping on Target Machine (Linux)

19. In the **Command Prompt** window, type **ping 10.10.10.9** and press **Enter**.

Note: **10.10.10.9** is the IP address of the **Ubuntu** virtual machine; this may vary in your lab environment.

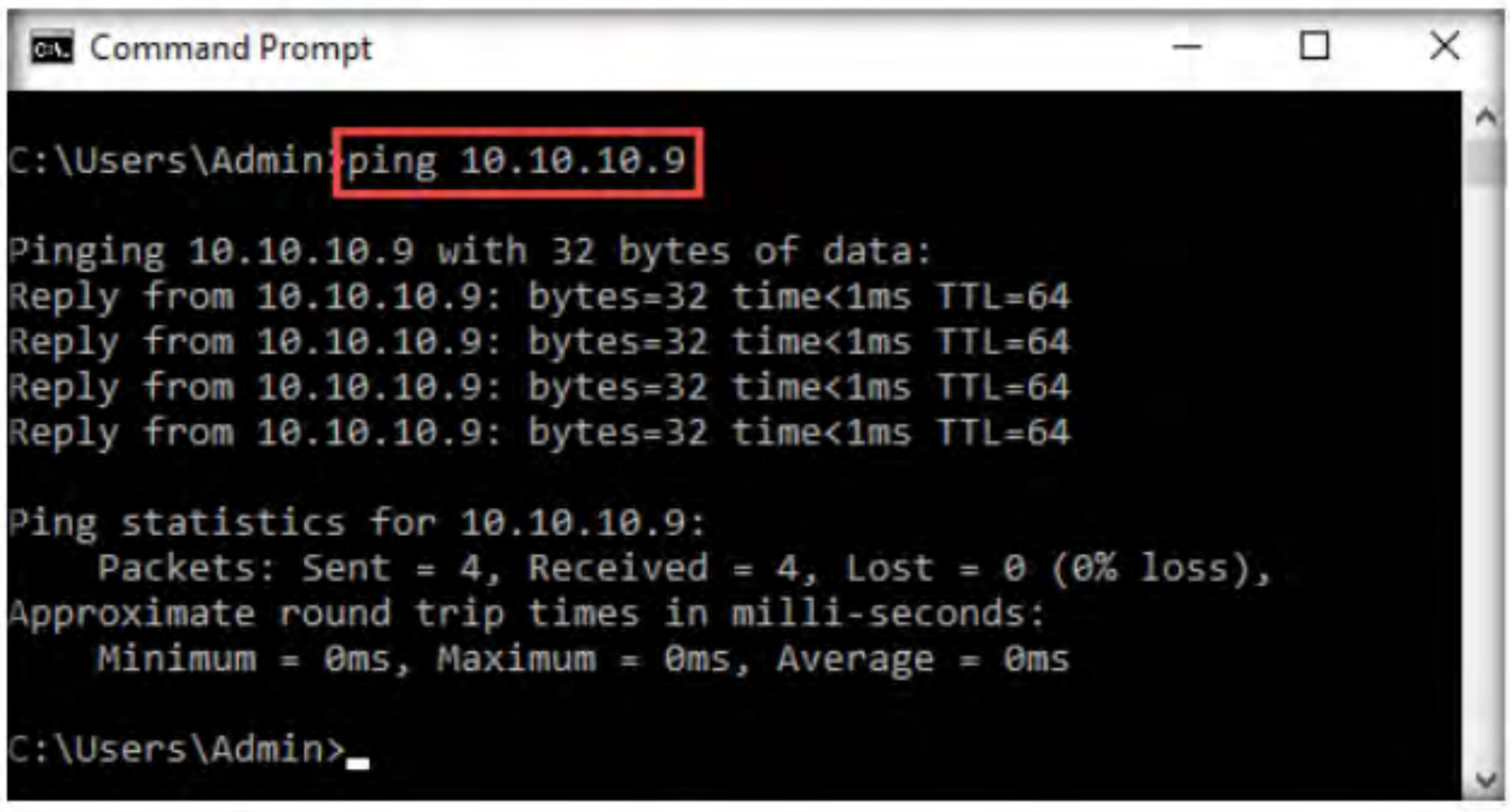


Figure 3.1.11: Sending ICMP requests to Ubuntu virtual machine

20. Observe the packets captured by **Wireshark**.

21. Choose any packet of ICMP reply from the **Ubuntu** (**10.10.10.9**) to **Windows 10** (**10.10.10.10**) virtual machine and expand the **Internet Protocol Version 4** node in the **Packet Details** pane.

Note: The IP address may vary in your lab environment.

TASK 1.6

Analyze the TTL Value

22. The TTL value is recorded as **64**, which means the ICMP reply possibly came from a Linux-based machine.

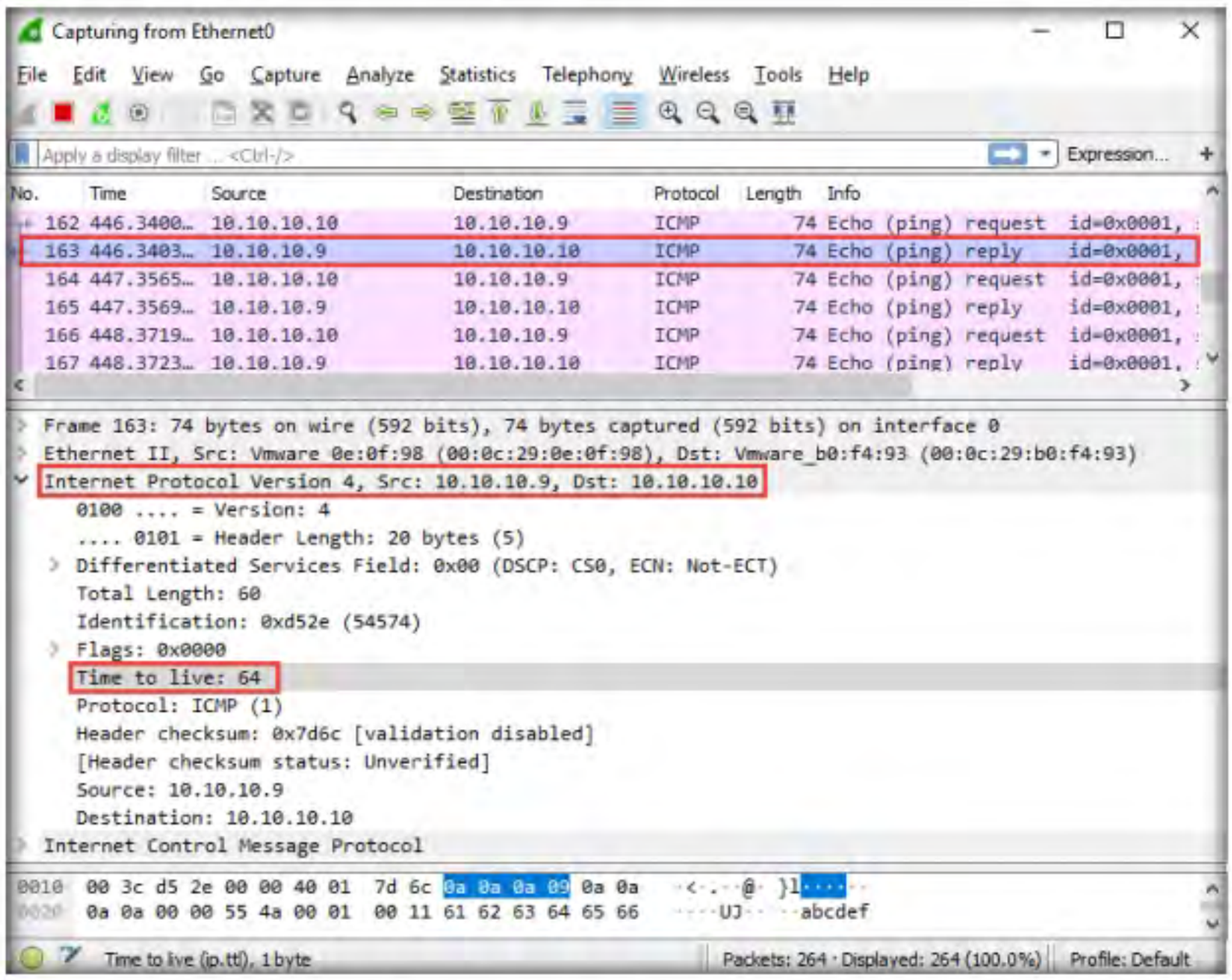


Figure 3.1.12: Time-to-Live value detected by Wireshark for Linux machine

23. Stop the capture in the **Wireshark** window by clicking on the **Stop** button.
24. This concludes the demonstration of identifying the OS of the target system using Wireshark.
25. Close all open windows and document all the acquired information.
26. Turn off the **Ubuntu** virtual machine.

TASK 2

Perform OS Discovery using Nmap Script Engine (NSE)

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

Note: Before beginning this lab, ensure that the **Windows 10** and **Windows Server 2016** virtual machines are turned on.

1. In the **Windows 10** virtual machine, click on the **Start** menu and launch **Nmap - Zenmap GUI** from the applications.

TASK 2.1

Perform OS Discovery Using -A Parameter (Aggressive Scan)

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**.

Note: **-A**: to perform an aggressive scan.

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.

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.

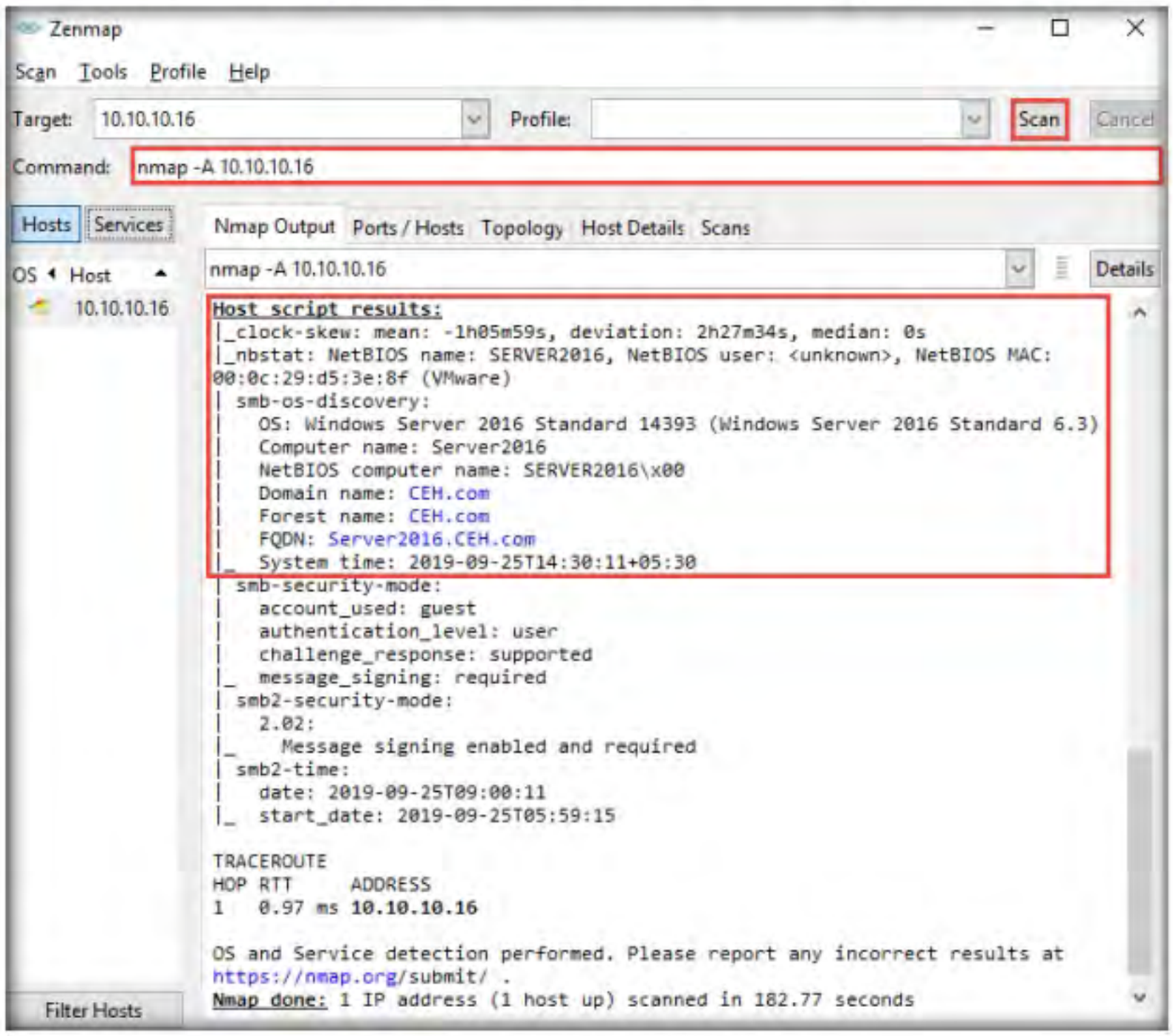


Figure 3.2.1: Zenmap scan results for -A parameter

TASK 2.2

Perform OS Discovery Using -O Parameter

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**.

Note: **-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.

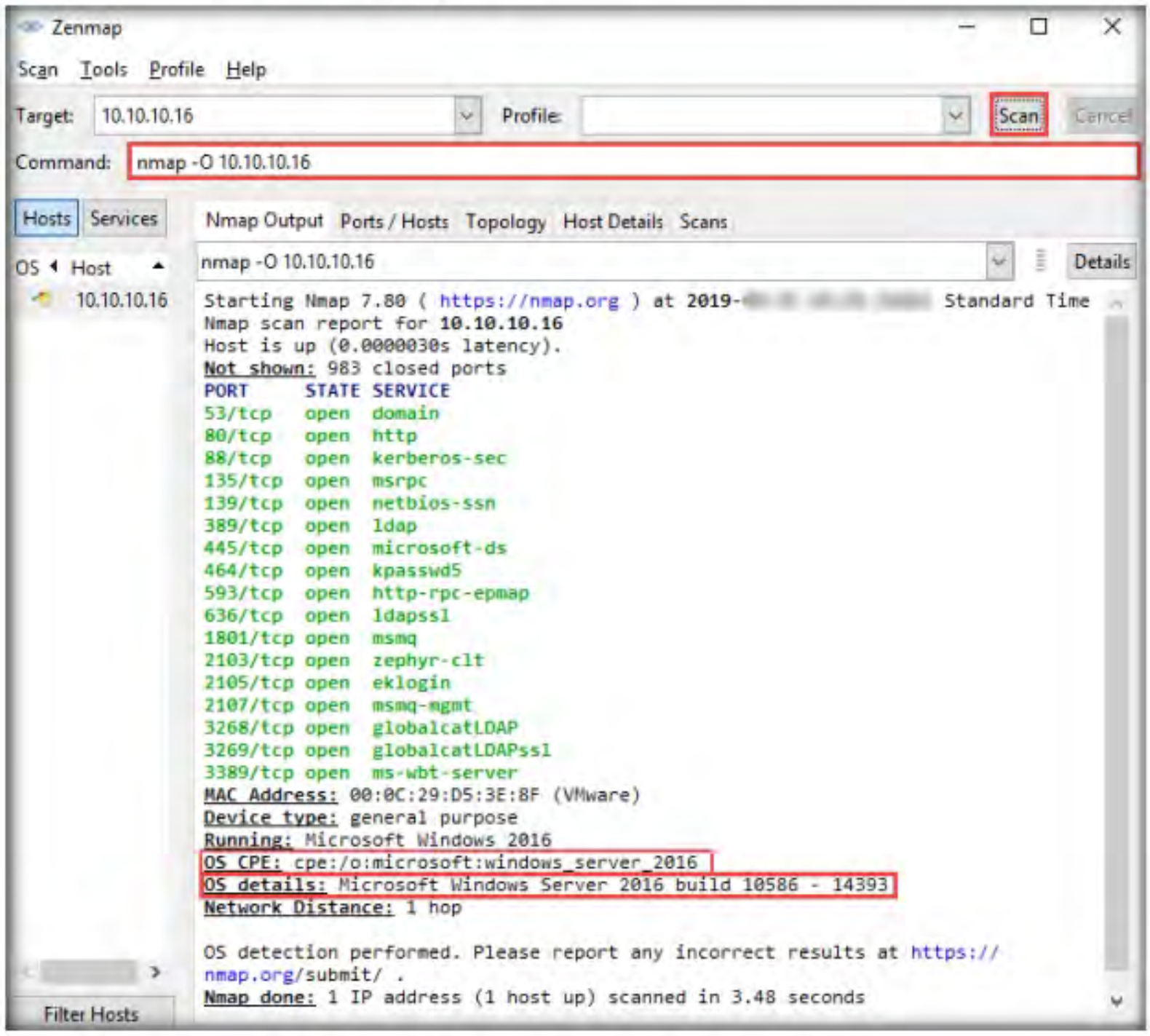


Figure 3.2.2: Zenmap scan results for -O parameter

TASK 2.3

Perform OS Discovery Using NSE

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**.

Note: **--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.

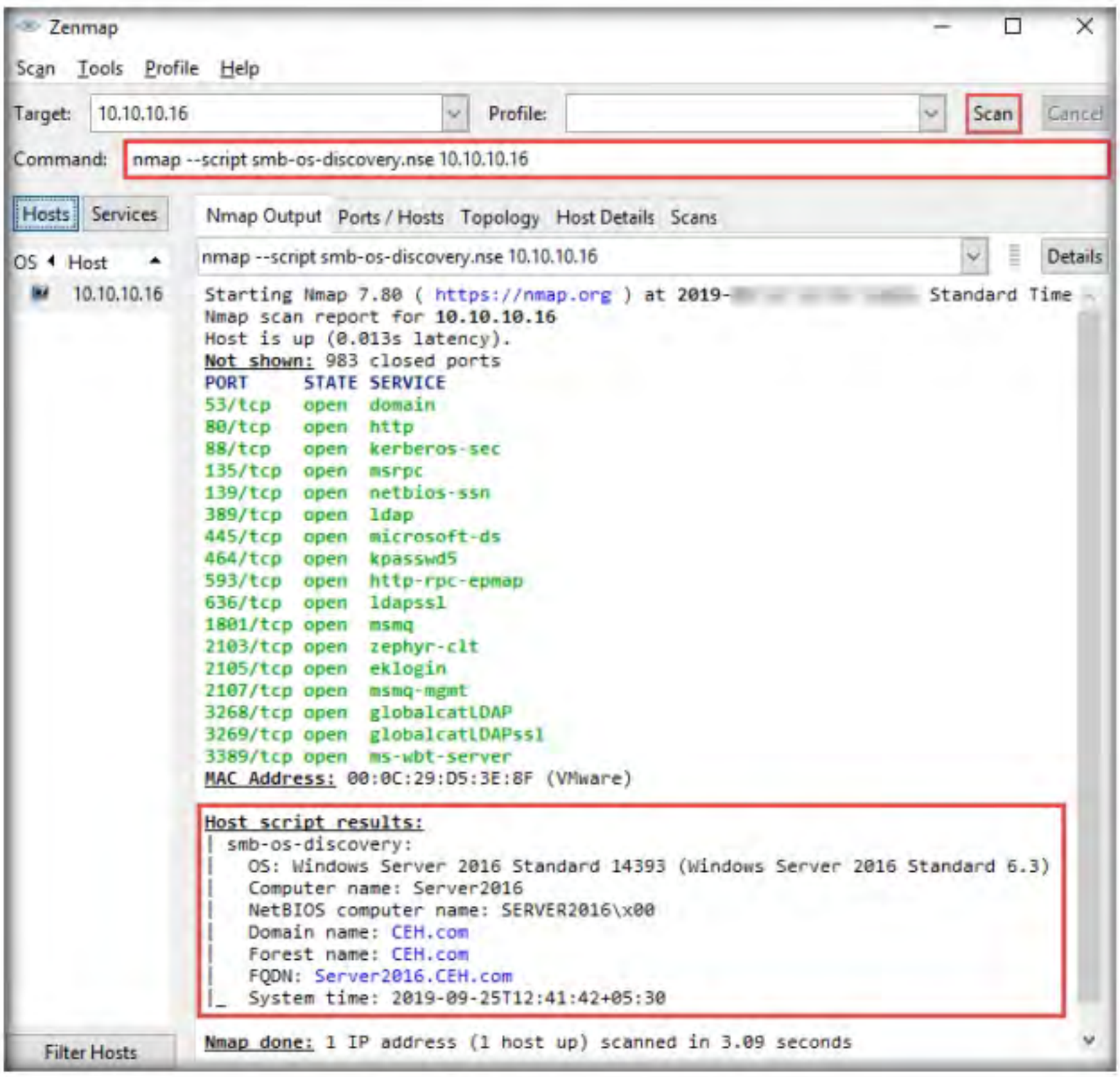


Figure 3.2.3: Zenmap scan results for NSE

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

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.

Perform OS Discovery using Unicornscan

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

1. Before beginning this lab, turn on the **Parrot Security** and **Ubuntu** virtual machine.

Note: Ensure that the **Windows Server 2016** virtual machine is also running.

2. In the login page, the **attacker** username will be selected by default. Enter password as **toor** in the **Password** field and press **Enter** to log in to the machine.

Note:

- If a **Parrot Updater** pop-up appears at the top-right corner of **Desktop**, ignore and close it.
- If a **Question** pop-up window appears asking you to update the machine, click **No** to close the window.

3. Click the **MATE Terminal** icon at the top of the **Desktop** window to open a **Terminal** window.
4. A **Parrot Terminal** window appears. In the terminal window, type **sudo su** and press **Enter** to run the programs as a root user.
5. In the **[sudo] password for attacker** field, type **toor** as a password and press **Enter**.

Note: The password that you type will not be visible.

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

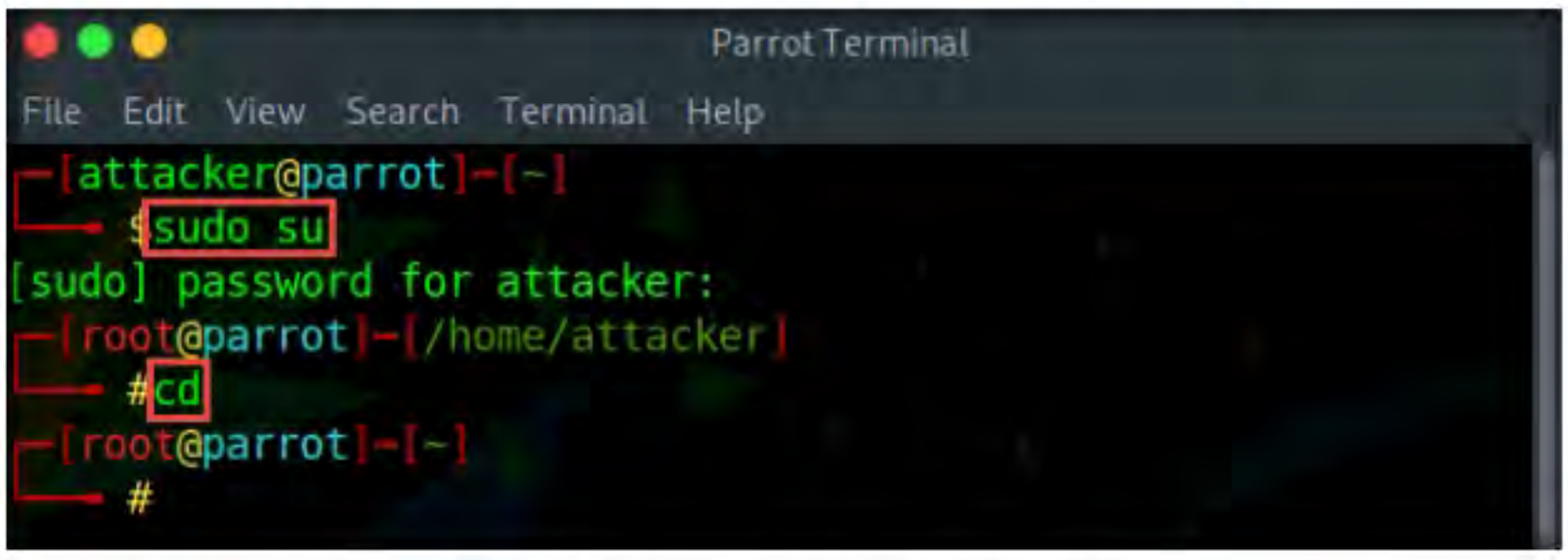


Figure 3.3.1: Running the programs as a root user

TASK 3.1

Perform OS Discovery on Windows-based Machine

7. 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**.

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

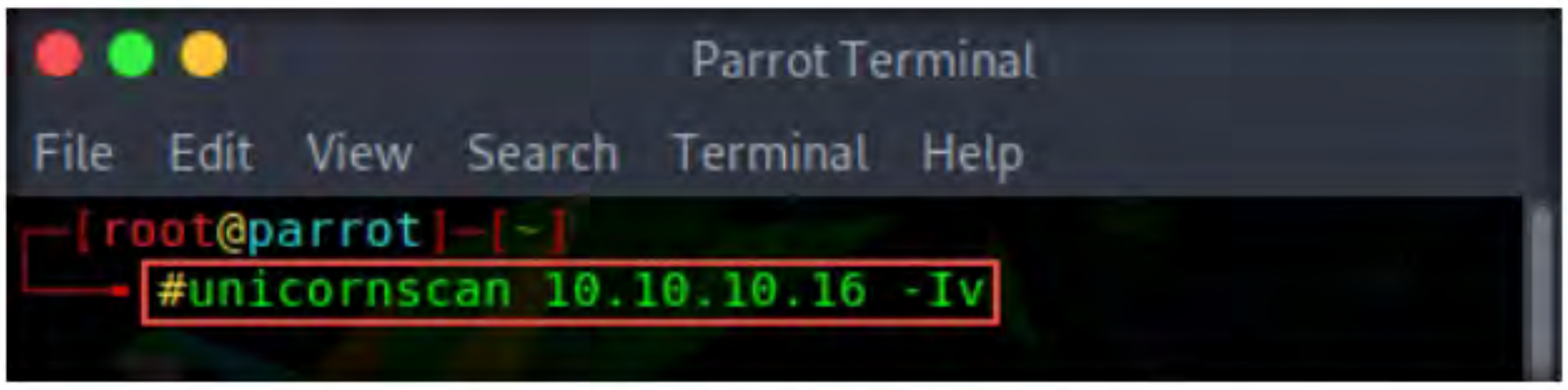


Figure 3.3.2: Unicornscan OS discovery command

8. 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).

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

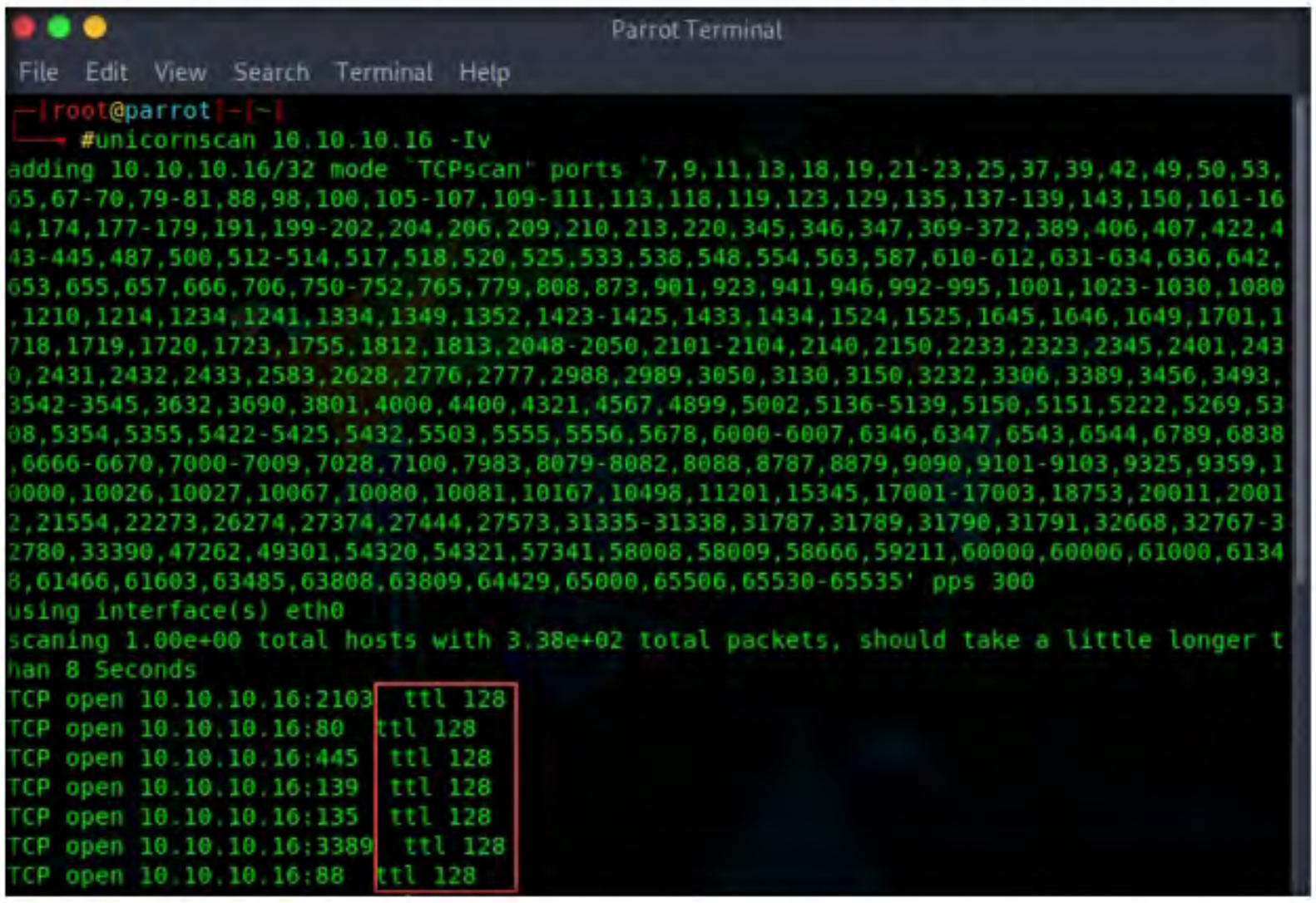


Figure 3.3.3: Unicornscan OS discovery command result Windows-based virtual machine

TASK 3.2

Perform OS Discovery on Linux-based Machine

9. In the **Parrot Terminal** window, type **unicornscan <Target IP Address> -Iv** (here, the target machine is Ubuntu [**10.10.10.9**]) and press **Enter**.
10. 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).

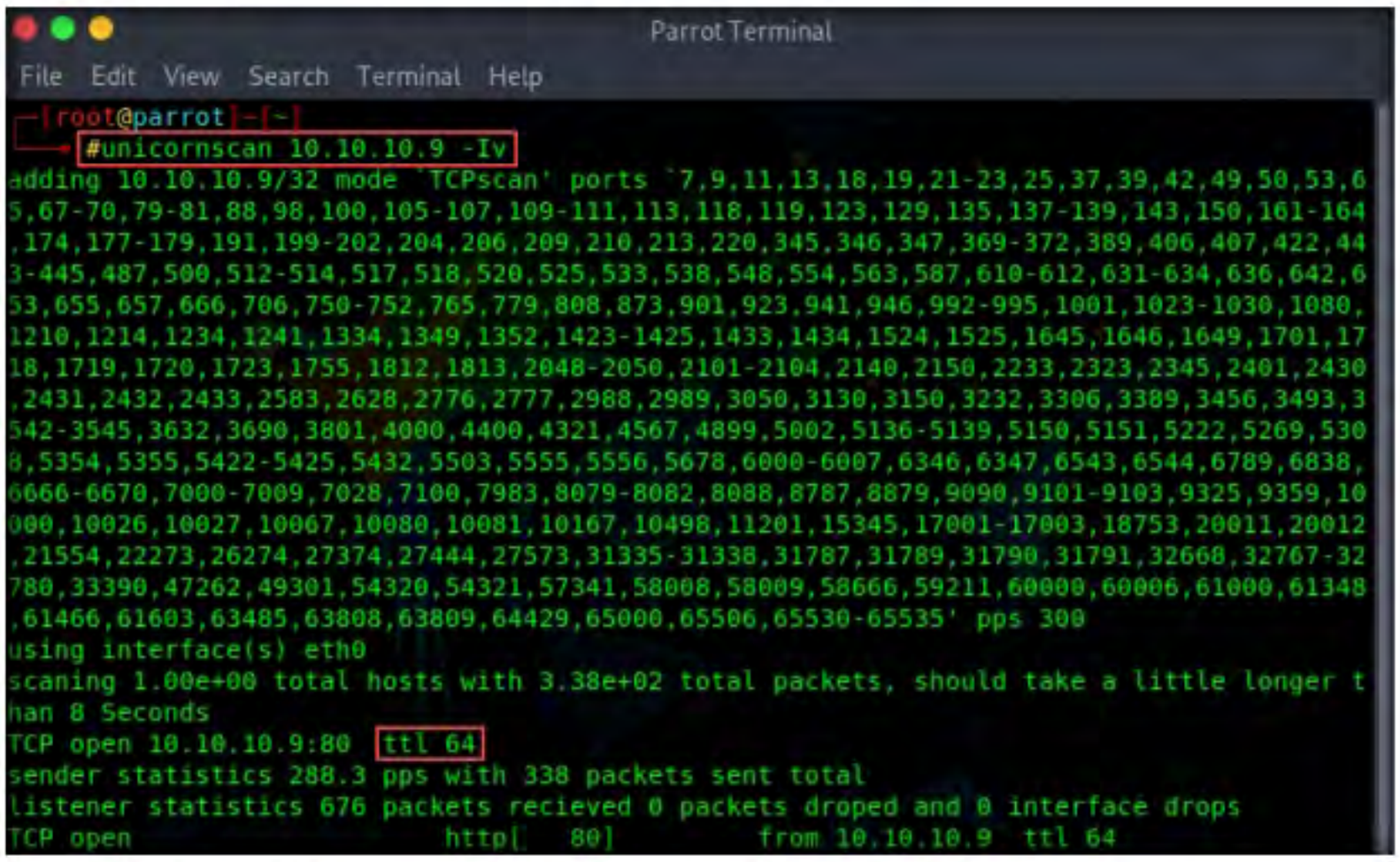


Figure 3.3.4: Unicornscan OS discovery command result for Linux-based virtual machine

11. This concludes the demonstration of discovering the OS of the target machine using Unicornscan.
12. Close all open windows and document all the acquired information.
13. Turn off the **Windows 10**, **Parrot Security**, **Windows Server 2016**, and **Ubuntu** virtual machines.

Lab Analysis

Analyze and document all the results discovered in the lab exercise.

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS ABOUT THIS LAB.

Internet Connection Required	
☐ Yes	☑ No
Platform Supported	
☑ Classroom	☑ iLabs

Lab 4

Scan beyond IDS and Firewall

Scanning beyond IDS and firewall is a process of sending intended packets to the target system in order to exploit IDS/firewall limitations.

ICON KEY

- Valuable information
- Test your knowledge
- Web exercise
- Workbook review

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
- Browse anonymously using Proxy Switcher
- Browse anonymously using CyberGhost VPN

Lab Environment

To carry out this lab, you need:

- Windows 10 virtual machine
- Windows Server 2016 virtual machine
- Parrot Security virtual machine
- Ubuntu virtual machine

- Web browsers with an Internet connection
- Administrator privileges to run the tools
- Colasoft Packet Builder located at **E:\CEH-Tools\CEHv11 Module 03 Scanning Networks\Packet Crafting Tools\Colasoft Packet Builder**
- Proxy Switcher located at **E:\CEH-Tools\CEHv11 Module 03 Scanning Networks\Proxy Tools\Proxy Switcher**
- CyberGhost VPN located at **E:\CEH-Tools\CEHv11 Module 03 Scanning Networks\Proxy Tools\CyberGhost VPN**
- You can also download the latest version of the above-mentioned tools from their official websites. If you decide to download the latest version, the screenshots shown in the lab might differ.

Tools demonstrated in this lab are available in E:\CEH-Tools\CEHv11 Module 03 Scanning Networks

Lab Duration

Time: 45 Minutes

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/UPD 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

Lab Tasks

TASK 1 — Scan beyond IDS/Firewall using various Evasion Techniques

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. Turn on the **Windows 10** and **Parrot Security** virtual machines.

Note: In this lab, we are using the **Parrot Security** (10.10.10.13) virtual machine as a host machine and the **Windows 10** (10.10.10.10) virtual machine as a target machine.

2. In the **Windows 10** virtual machine, log in with the credentials **Admin/Pa$$w0rd**.

TASK 1.1 — Turn on Windows Defender Firewall

3. 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.

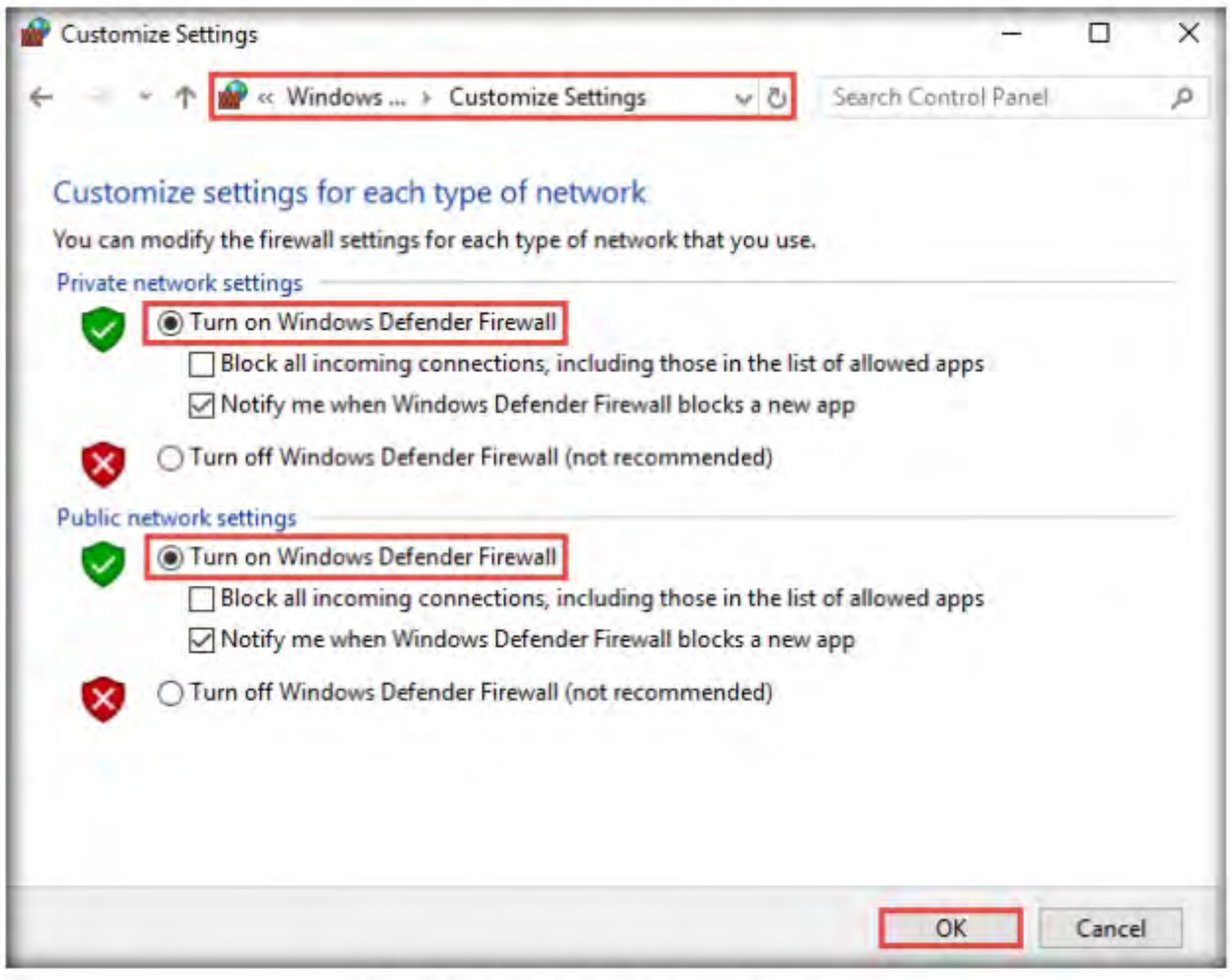


Figure 4.1.1: Turn on Windows Defender Firewall

4. Click on the **Start** menu and launch **Wireshark** from the applications. Start capturing packets by double-clicking the available ethernet or interface (here, **Ethernet0**).

Note: In this lab, the available interface might vary in your lab environment.

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.

5. Now, switch to the **Parrot Security** virtual machine. In the login page, the **attacker** username will be selected by default. Enter password as **toor** in the **Password** field and press **Enter** to log in to the machine.

Note:

- If a **Parrot Updater** pop-up appears at the top-right corner of **Desktop**, ignore and close it.
- If a **Question** pop-up window appears asking you to update the machine, click **No** to close the window.

6. Click the **MATE Terminal** icon at the top 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**.

Note: The password that you type will not be visible.

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

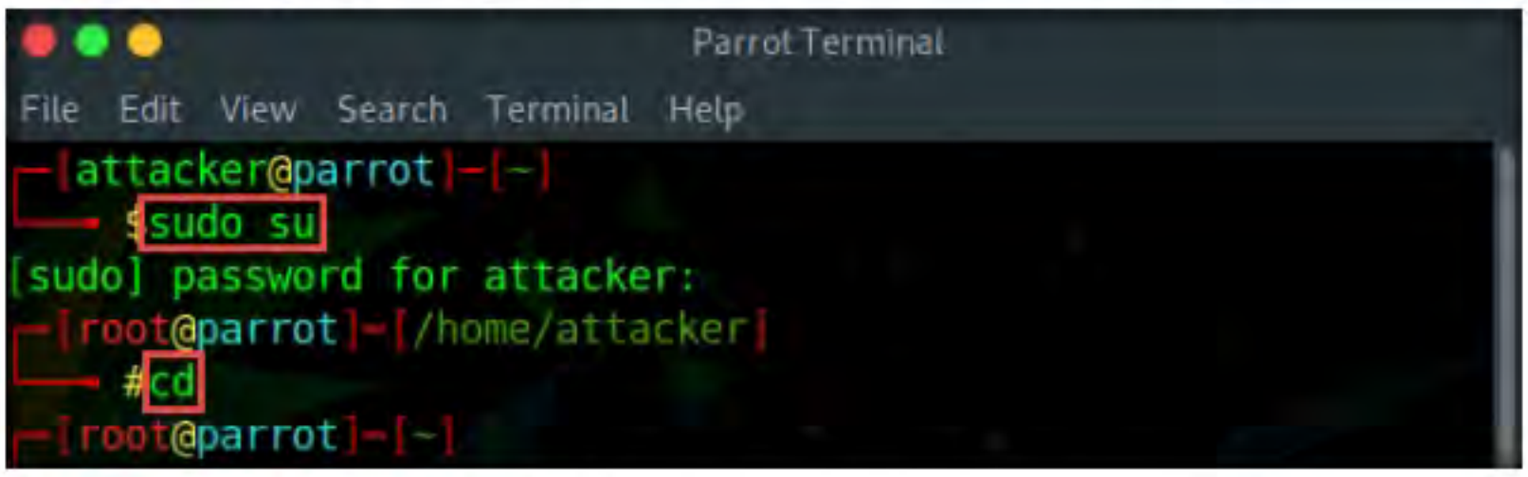


Figure 4.1.2: Running the programs as a root user

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

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

Note: 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.

TASK 1.2

Perform Packet Fragmentation

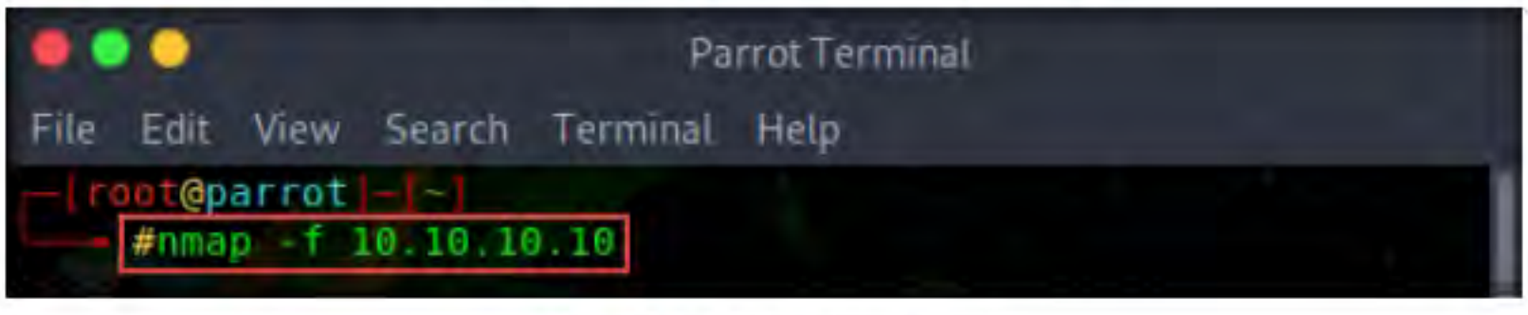


Figure 4.1.3: Nmap fragment scan

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.

Note: Depending upon the target machine that you are using, the results might vary in your lab environment.

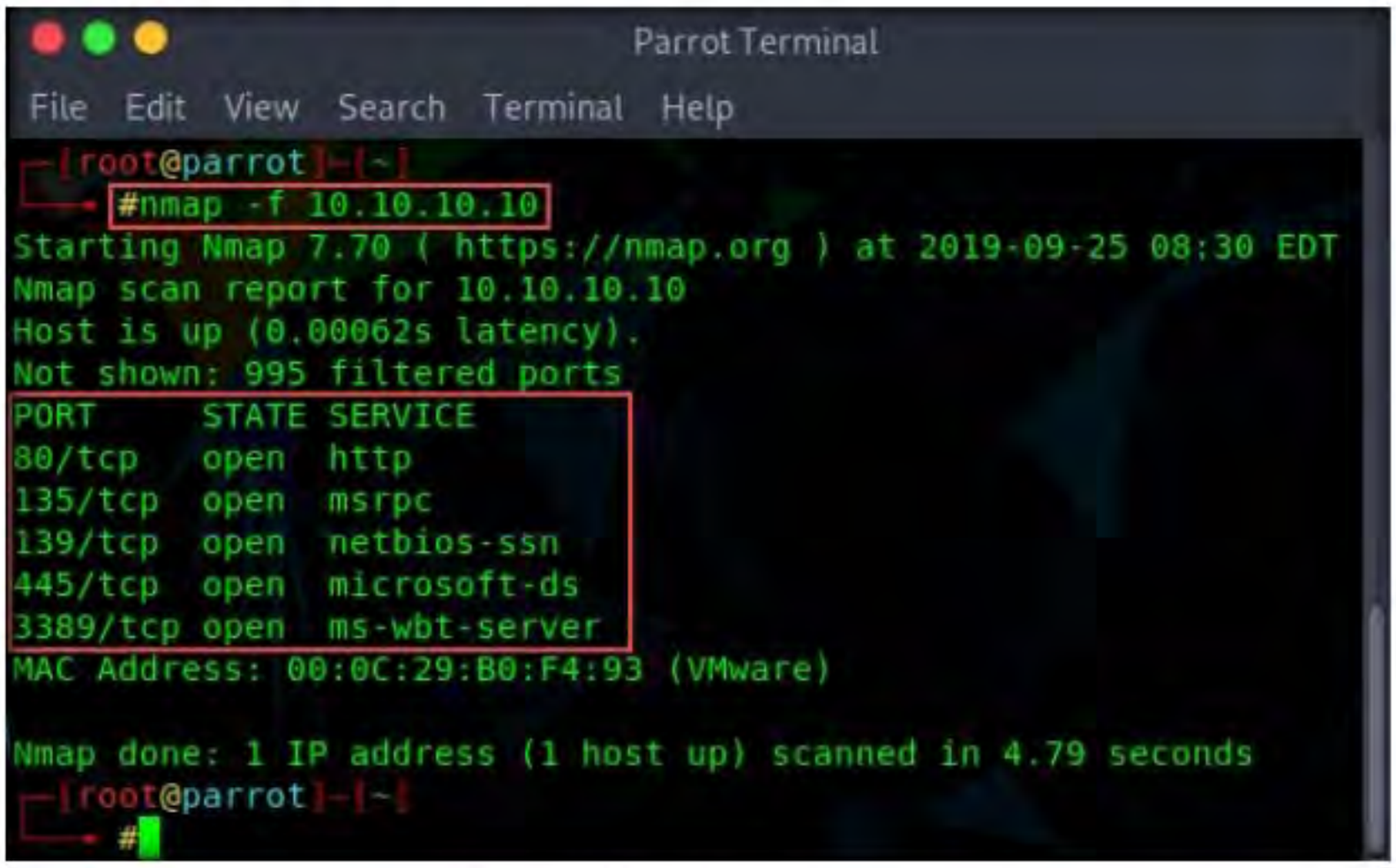


Figure 4.1.4: Nmap fragment scan output

TASK 1.3

Perform Source Port Manipulation

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**.

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

Note: 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.

Note: Depending upon the target machine that you are using, the results might vary in your lab environment.

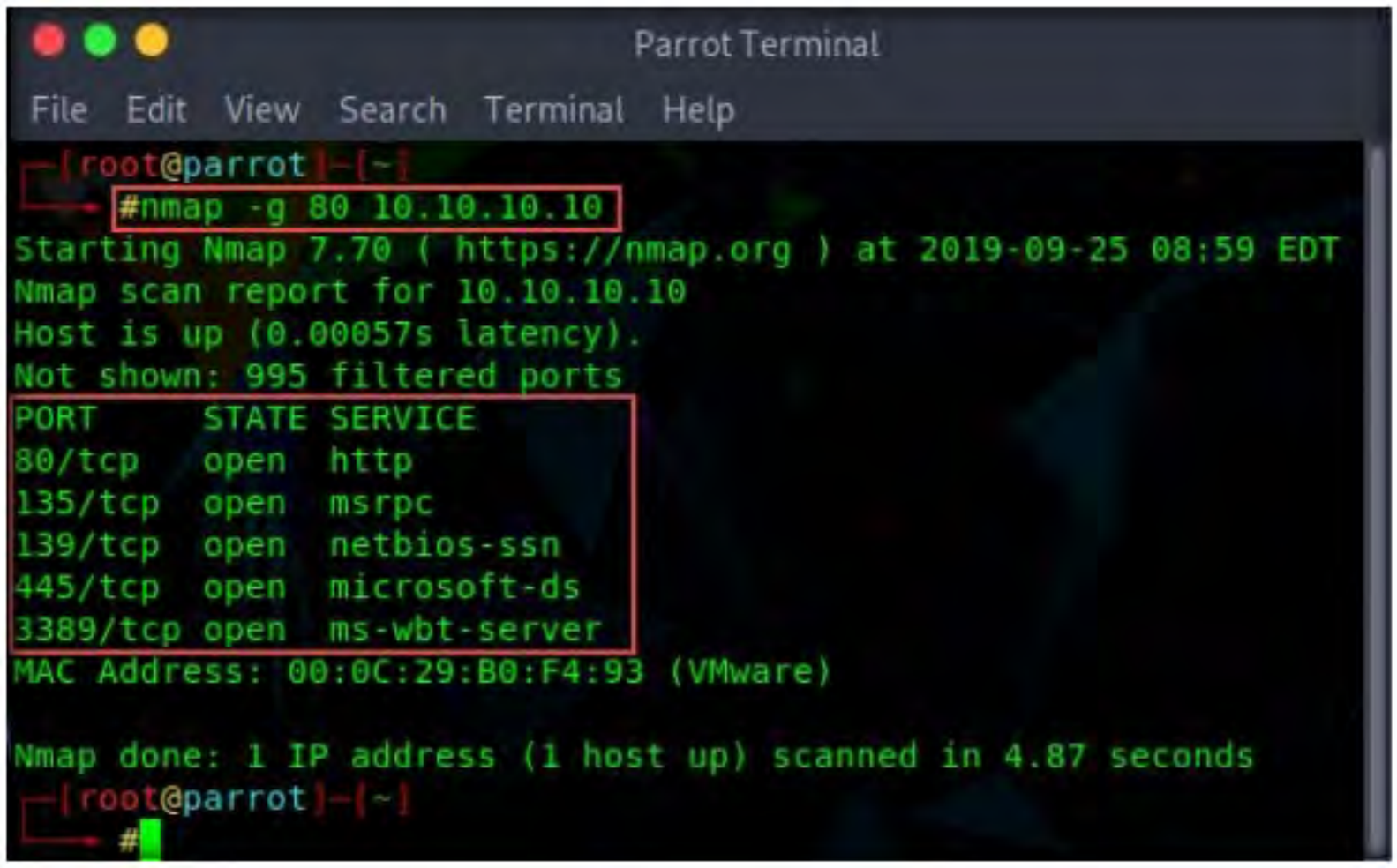


Figure 4.1.5: Nmap source port manipulation scan output

TASK 1.4

Perform Maximum Transmission Unit

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

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

Note: 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.

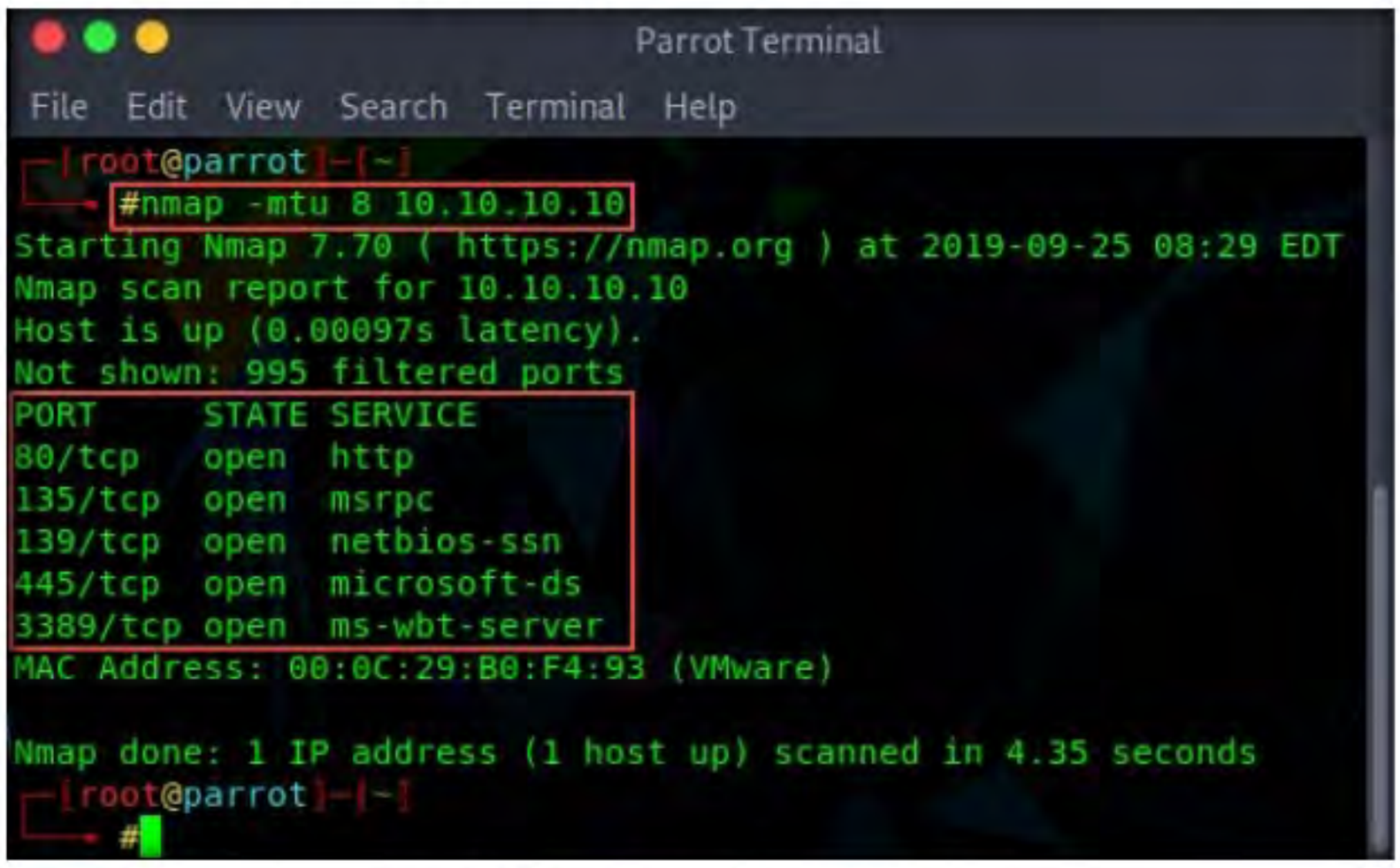


Figure 4.1.6: Nmap Maximum Transmission Unit scan

TASK 1.5

Perform IP Address Decoy

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

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

Note: 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.

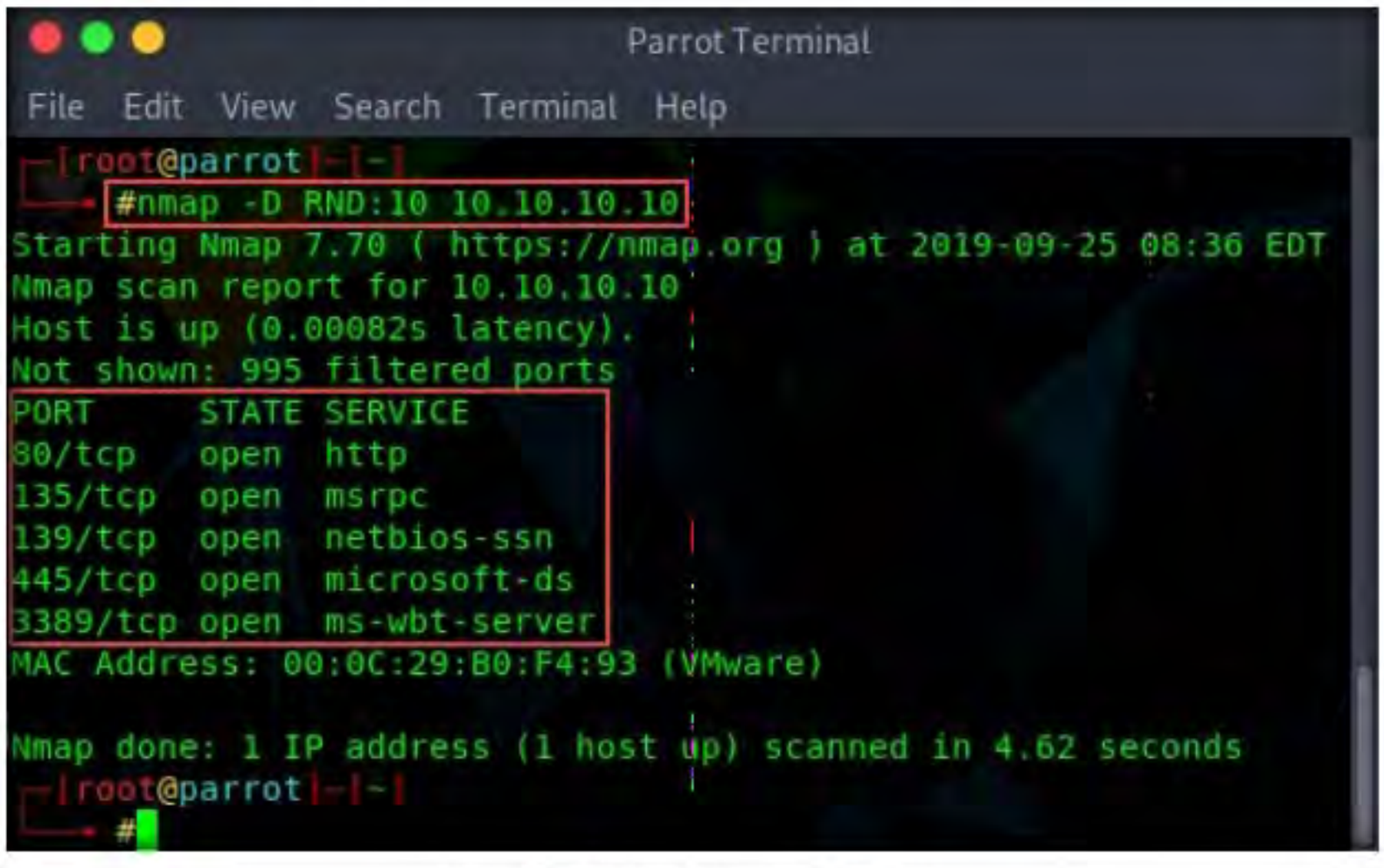


Figure 4.1.7: Nmap Decoying IP Addresses

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

Note: The results may vary in your lab environment.

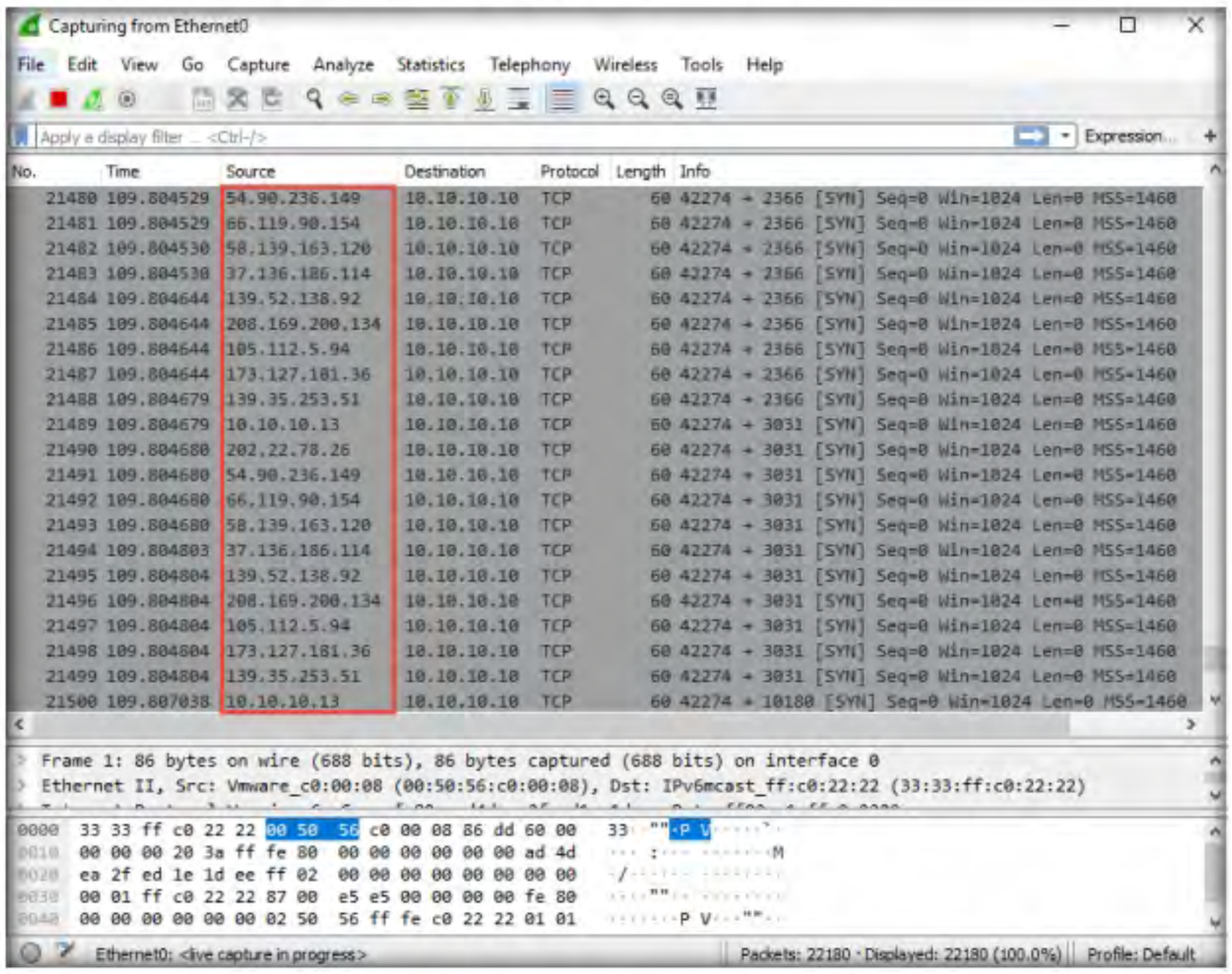


Figure 4.1.8: Decoyed IP Addresses in Windows 10 Wireshark

17. This concludes the demonstration of evading IDS and firewall using various evasion techniques in Nmap.
18. Close all open windows and document all the acquired information.
19. Turn off the **Parrot Security** virtual machine.

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

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. Turn on the **Windows Server 2019** virtual machine and log in with the credentials **Administrator/Pa$$w0rd**.
2. Install **Wireshark** available at **Z:\CEHv11 Module 03 Scanning Networks\Banner Grabbing Tools\Wireshark** with all the default settings.
3. **Windows Server 2019** virtual machine will restart. Log in with the credentials **Administrator/Pa$$w0rd**.

4. Click on the **Start** menu and launch **Wireshark** from the applications.
5. The **Wireshark Network Analyzer** main window appears; double-click the available ethernet or interface (here, **Ethernet0**) to start the packet capture.

TASK 2.1

Install Colasoft Packet Builder

6. Navigate to **Z:\CEHv11 Module 03 Scanning Networks\Packet Crafting Tools\Colasoft Packet Builder** and double-click **pktbuilder_2.0.0.212.exe**.
7. Follow the wizard-driven installation steps to install **Colasoft Packet Builder**.
8. After the completion of the installation, click on the **Launch Colasoft Packet Builder 2.0** checkbox and click **Finish**.

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.

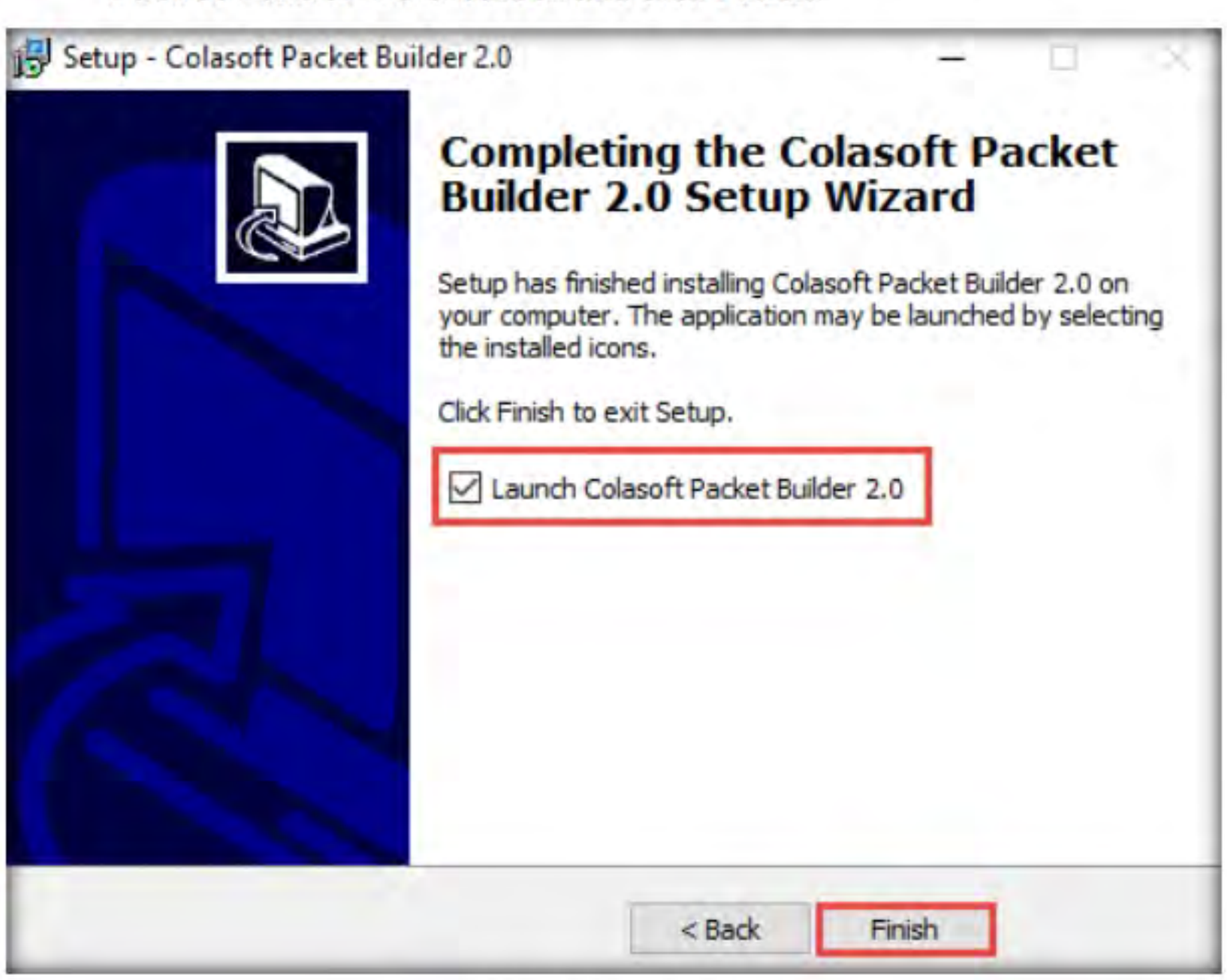


Figure 4.2.1: Colasoft Packet Builder Setup

TASK 2.2

Choose a Network Interface

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

Note: If a pop-up appears, close the window.

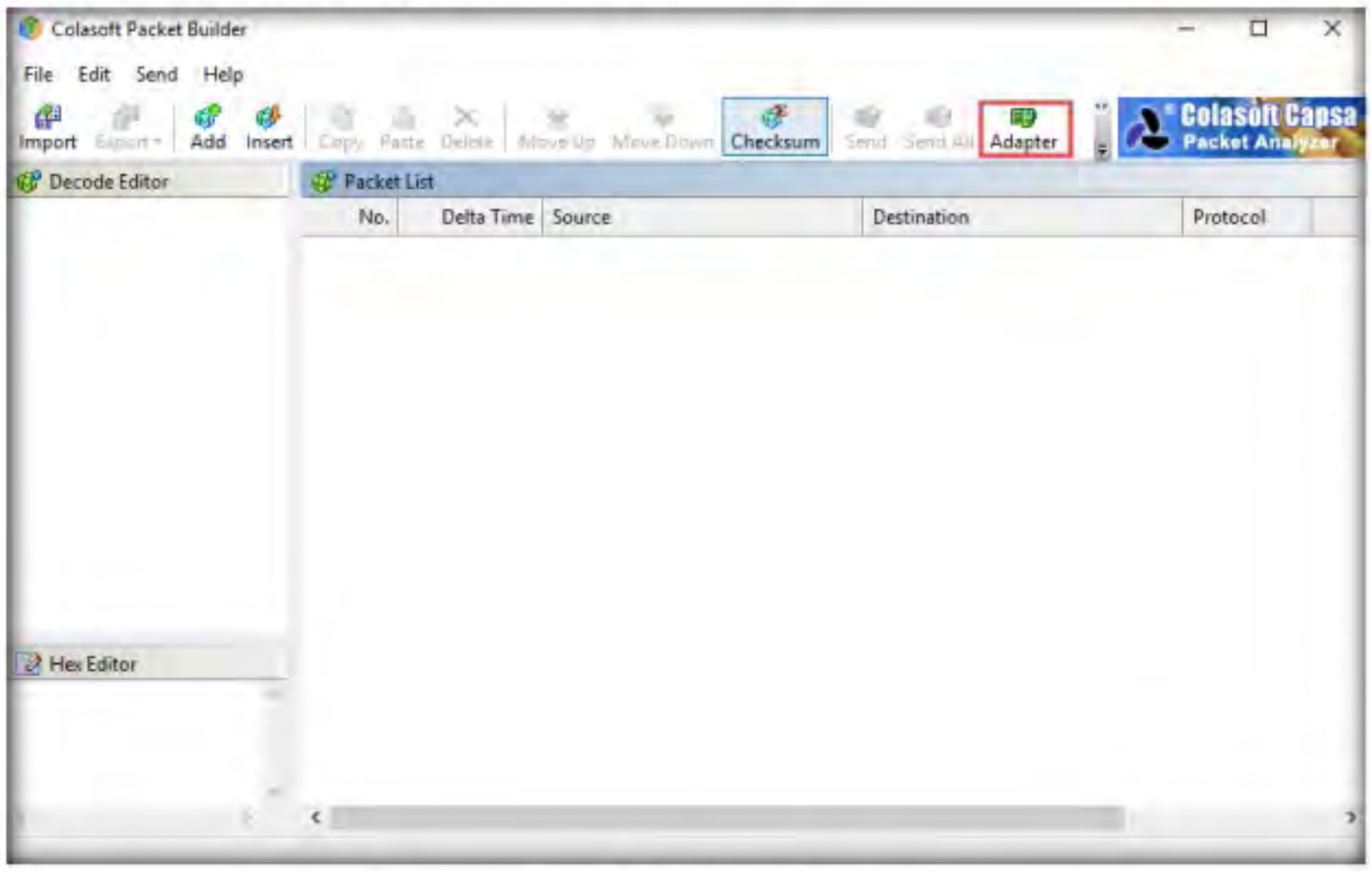


Figure 4.2.2: Colasoft Packet Builder GUI: click on adapter icon

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

Note: The adapter configuration might differ in your lab environment.

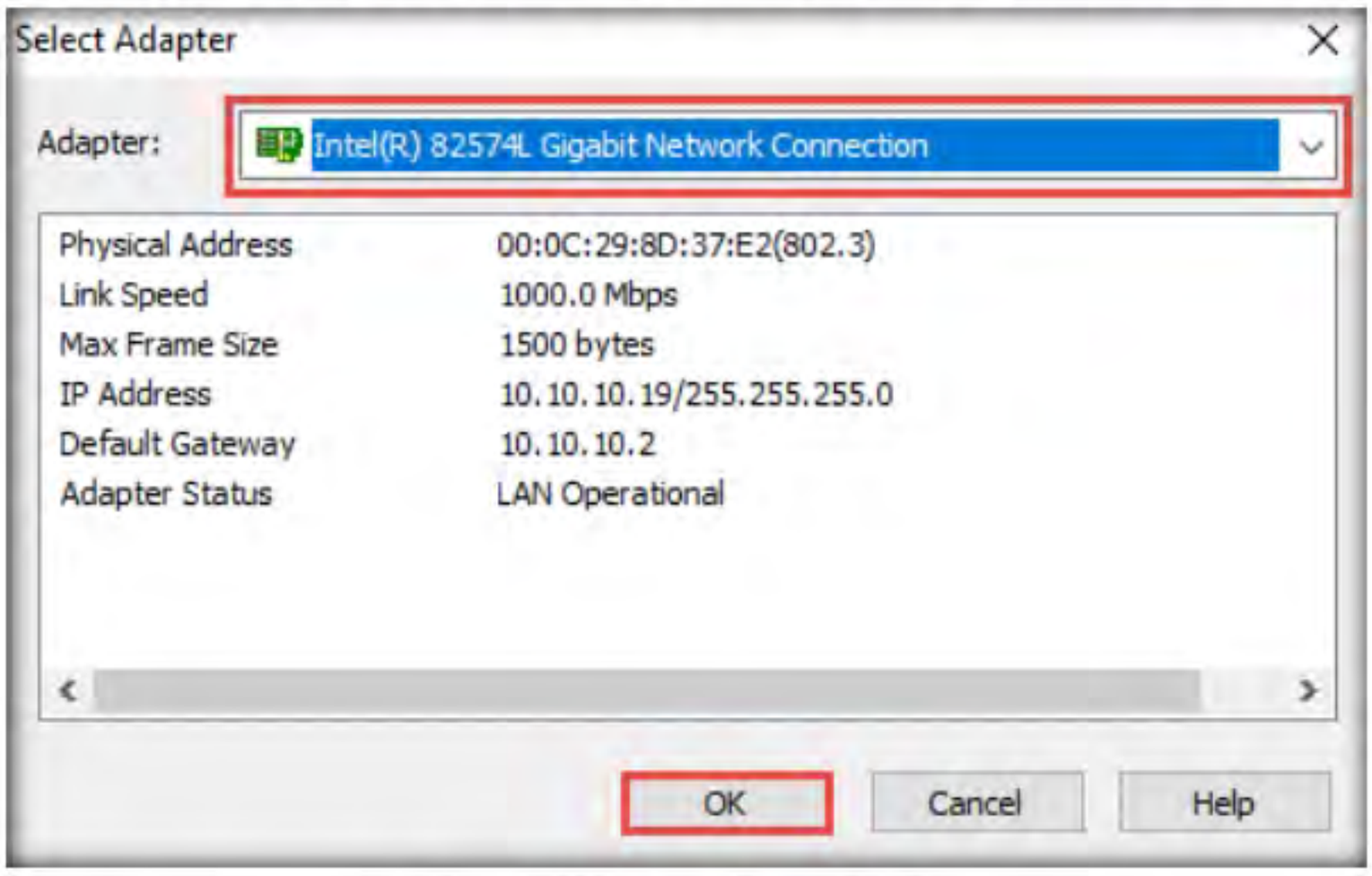


Figure 4.2.3: Choosing an adapter in Colasoft

TASK 2.3

Create an ARP Packet

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

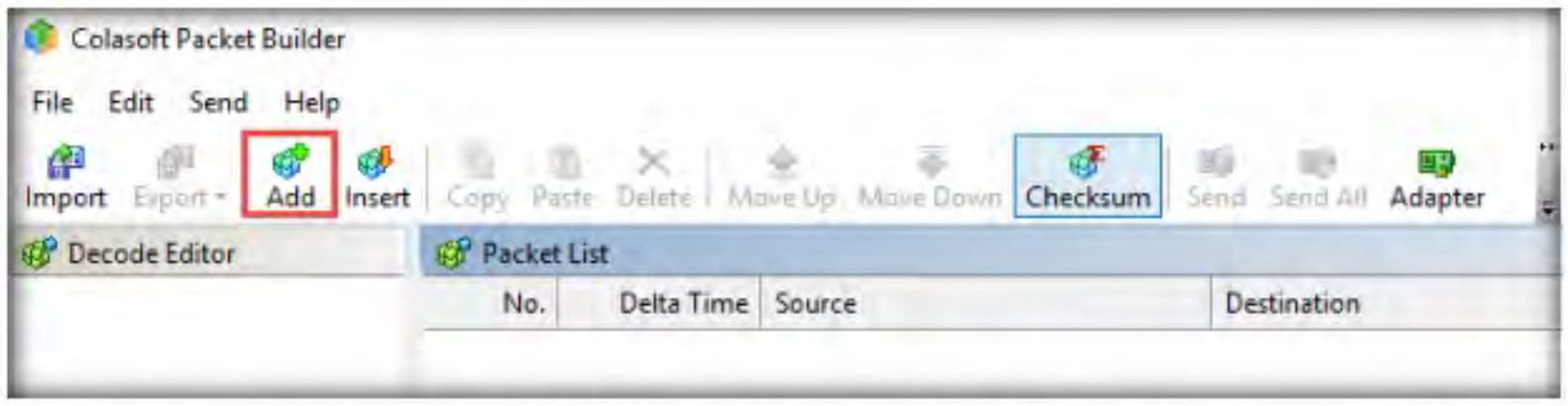


Figure 4.2.4: Adding a packet in Colasoft Packet Builder

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

Figure 4.2.5: Add Packet dialog box

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

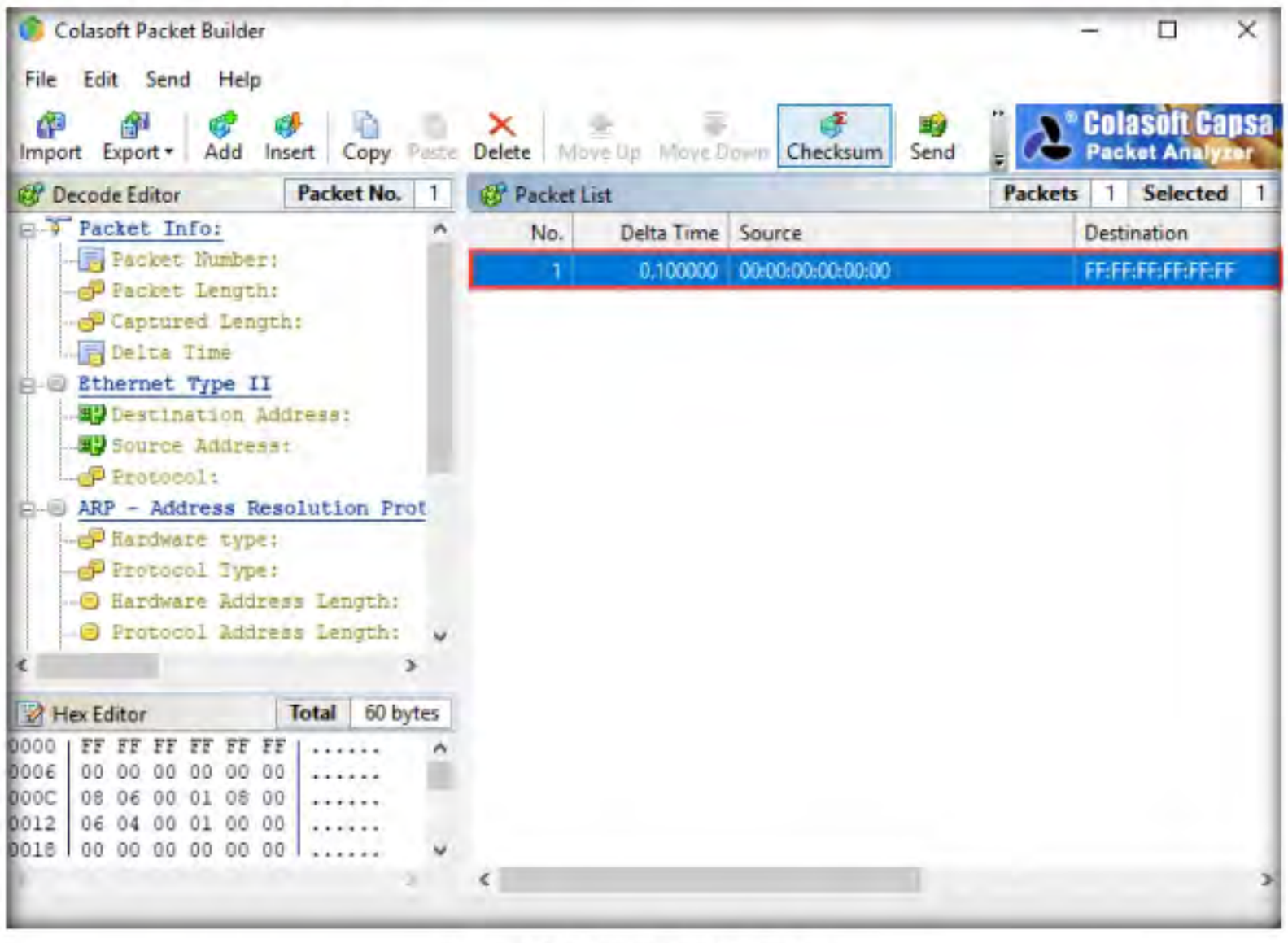


Figure 4.2.6: Viewing the added packets

14. **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.

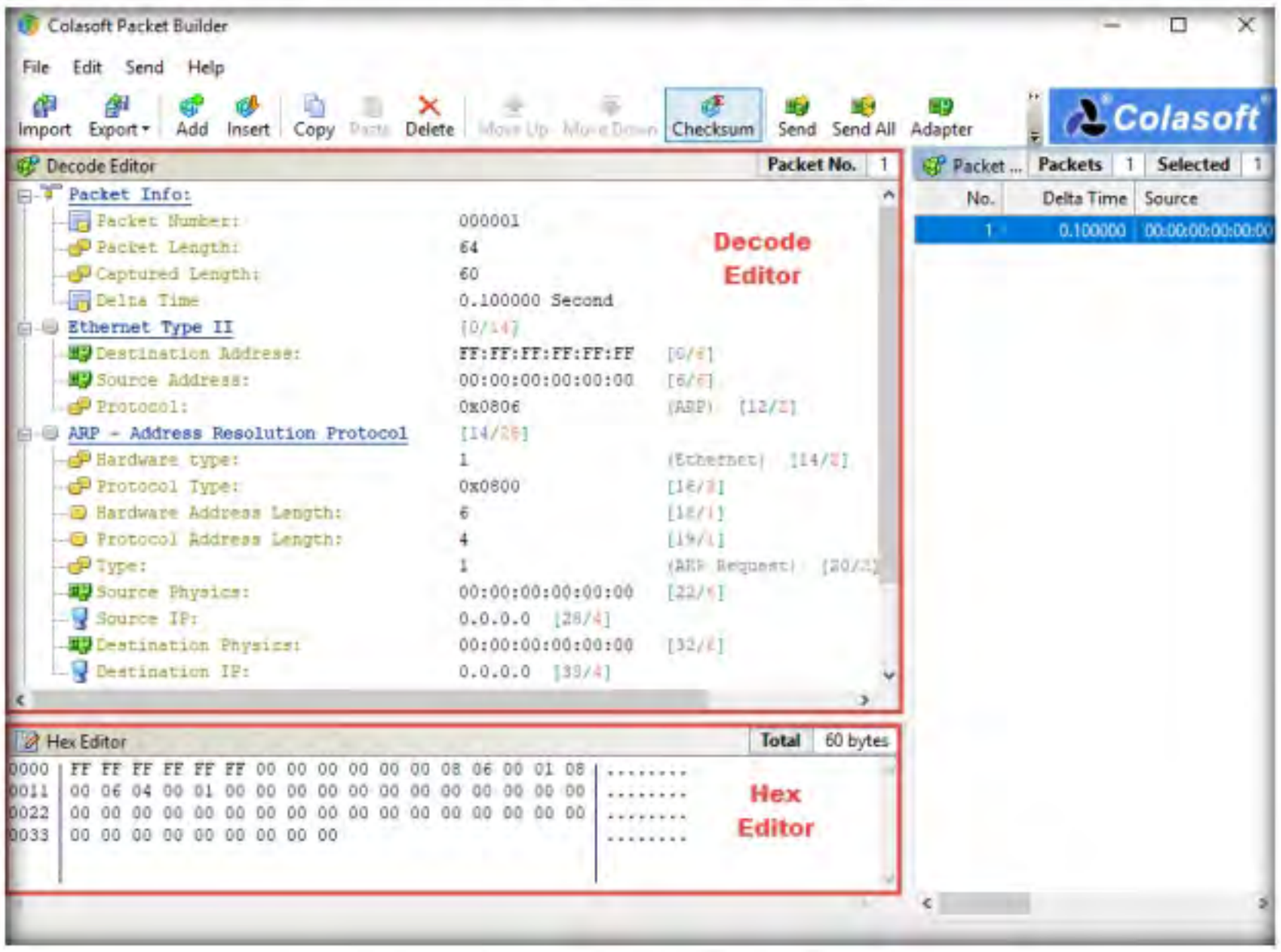


Figure 4.2.7: Colasoft Packet Builder Decode and Hex Editors

TASK 2.4

Send the Created ARP Packet to the Target

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

Figure 4.2.8: Sending all packets

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

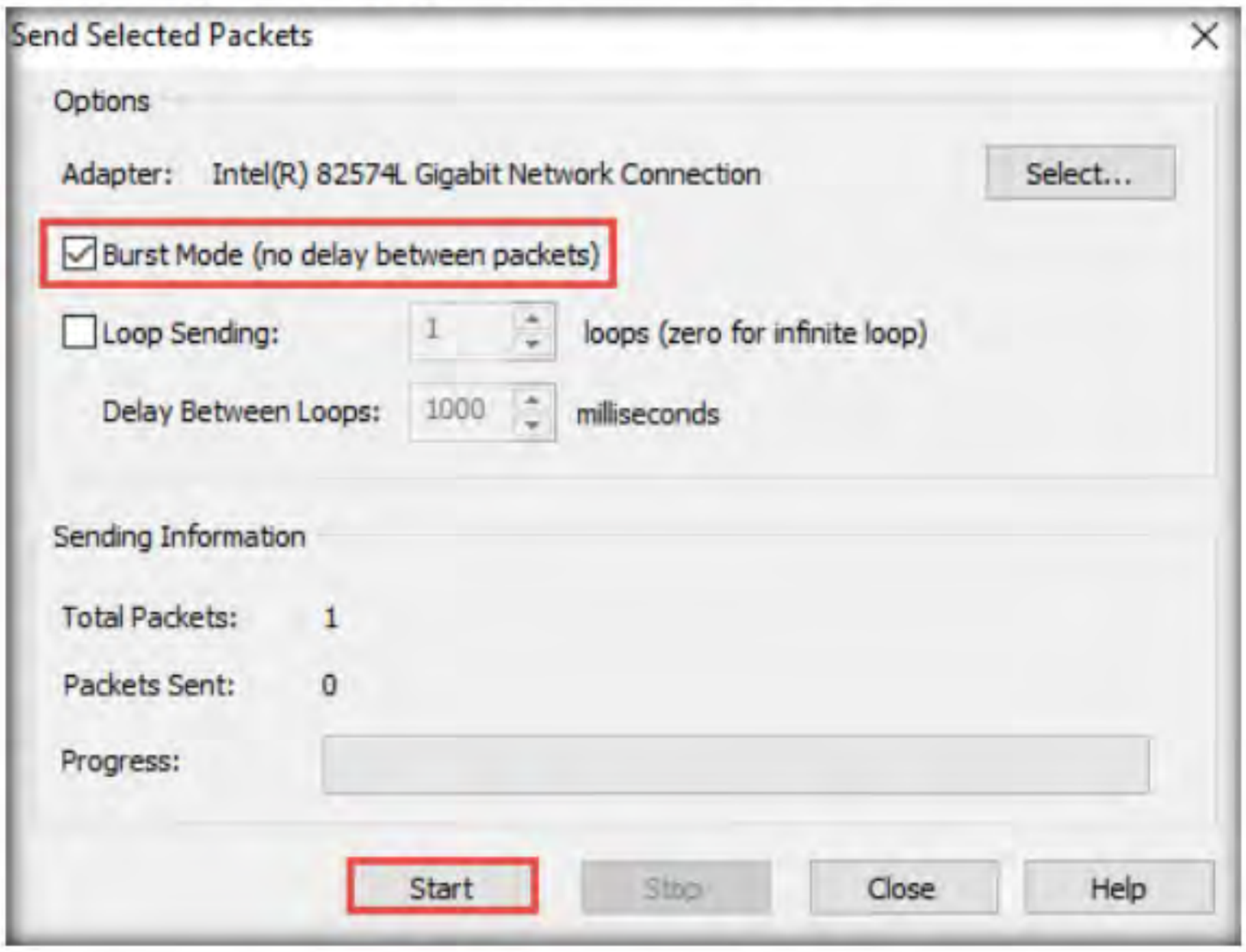


Figure 4.2.9: Setting Burst Mode option

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

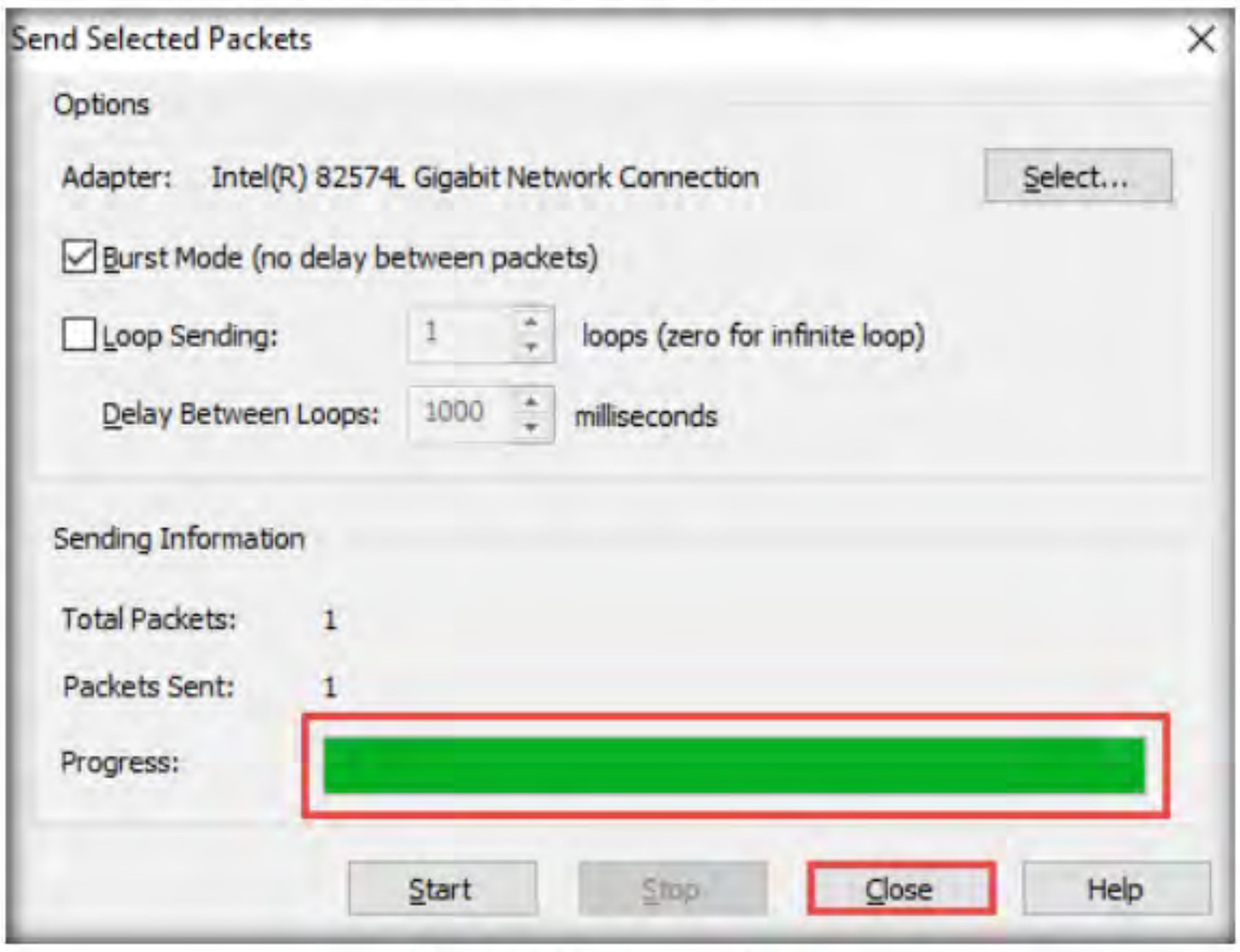


Figure 4.2.10: All packets successfully sent

18. 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.

Note: Here, the host machine (**10.10.10.19**) is broadcasting ARP packets, prompting the target machine (**10.10.10.10**) to reply to the message.

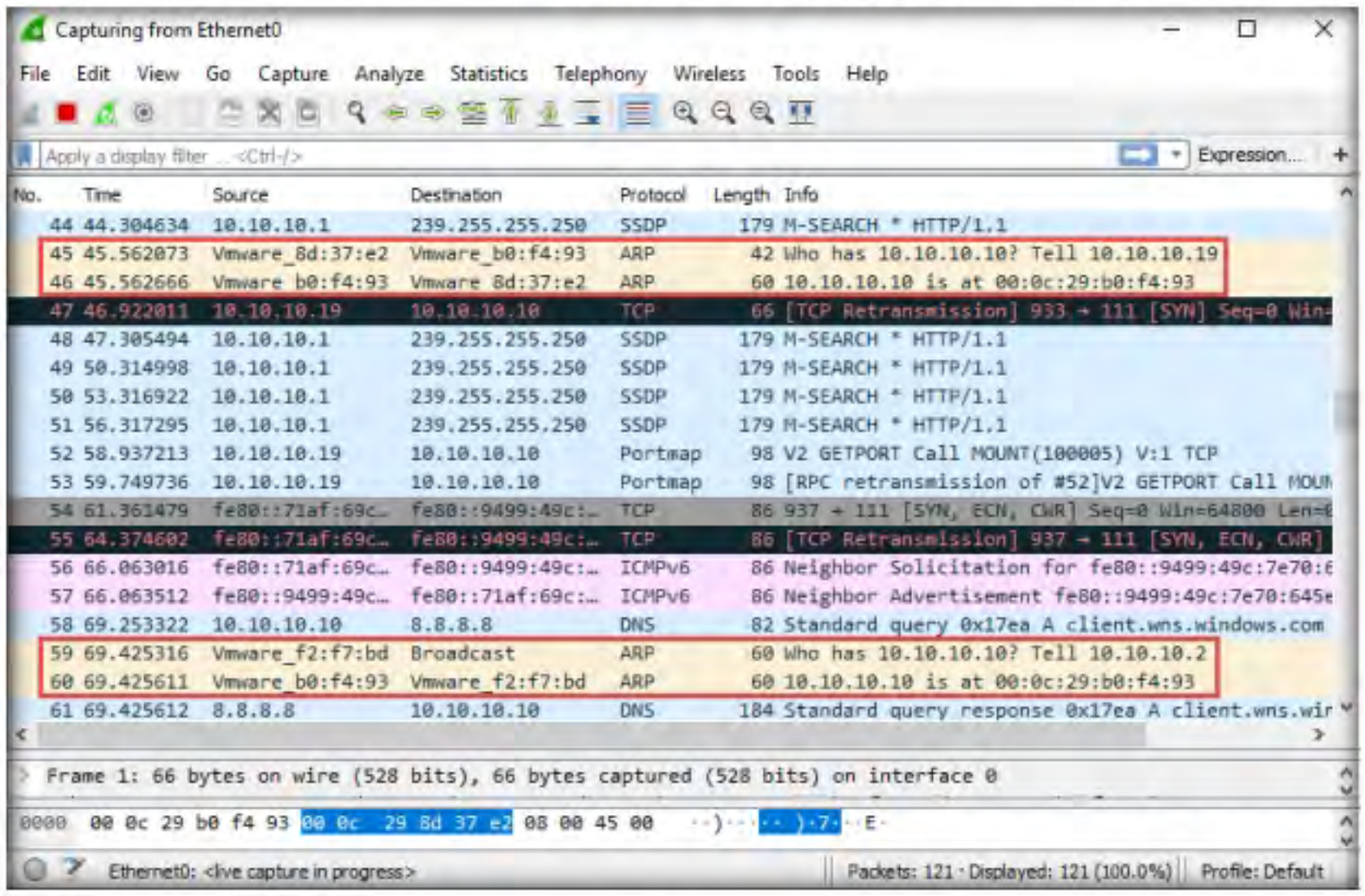


Figure 4.2.11: Wireshark captured packets

19. To export the packet, click **Export → Selected Packets...**.

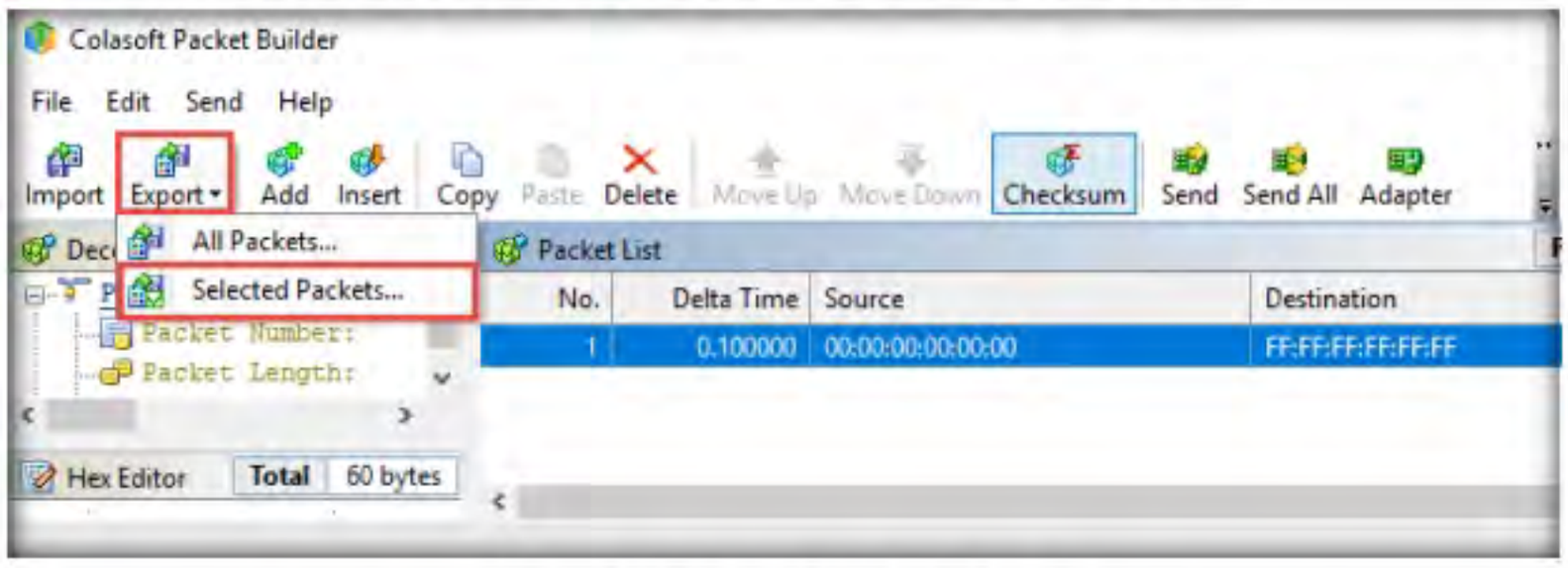


Figure 4.2.12: Exporting the packets in Colasoft

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.
24. Turn off the **Windows Server 2019** virtual machine.

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.

Note: Before beginning this task, ensure that the **Windows 10** virtual machine is running and that **Windows Defender Firewall** is enabled.

1. In the **Windows 10** virtual machine, launch **Wireshark** and double-click the available ethernet or interface (here, **Ethernet0**) to start the packet capture.
2. Turn on the **Parrot Security** virtual machine.
3. In the login page, the **attacker** username will be selected by default. Enter password as **toor** in the **Password** field and press **Enter** to log in to the machine.

Note:

- If a **Parrot Updater** pop-up appears at the top-right corner of **Desktop**, ignore and close it.
- If a **Question** pop-up window appears asking you to update the machine, click **No** to close the window.

4. Click the **MATE Terminal** icon at the top of the **Desktop** window to open a **Terminal** window.
5. A **Parrot Terminal** window appears. In the terminal window, type **sudo su** and press **Enter** to run the programs as a root user.
6. In the **[sudo] password for attacker** field, type **toor** as a password and press **Enter**.

Note: The password that you type will not be visible.

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

TASK 3.1

Perform UDP Packet Crafting

8. In the terminal window, type **hping3 <Target IP Address> --udp --rand-source --data 500** (here, the target machine is Windows 10 [**10.10.10.10**]) and press **Enter**.

Note: 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.

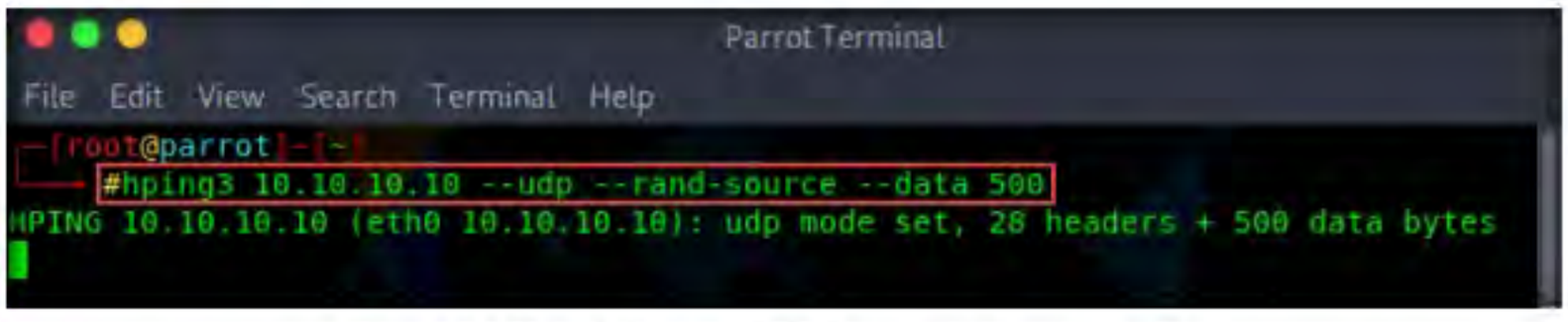


Figure 4.3.1: Hping3 Output of 3 Packets sent to the target machine

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

Note: You can double-click any UDP packet and observe the details.

10. Expand the **Data** node in the **Packet Details** pane and observe the size of **Data** and its **Length** (the length is the same as the size of the packet body that we specified in Hping3 command, i.e., **500**).

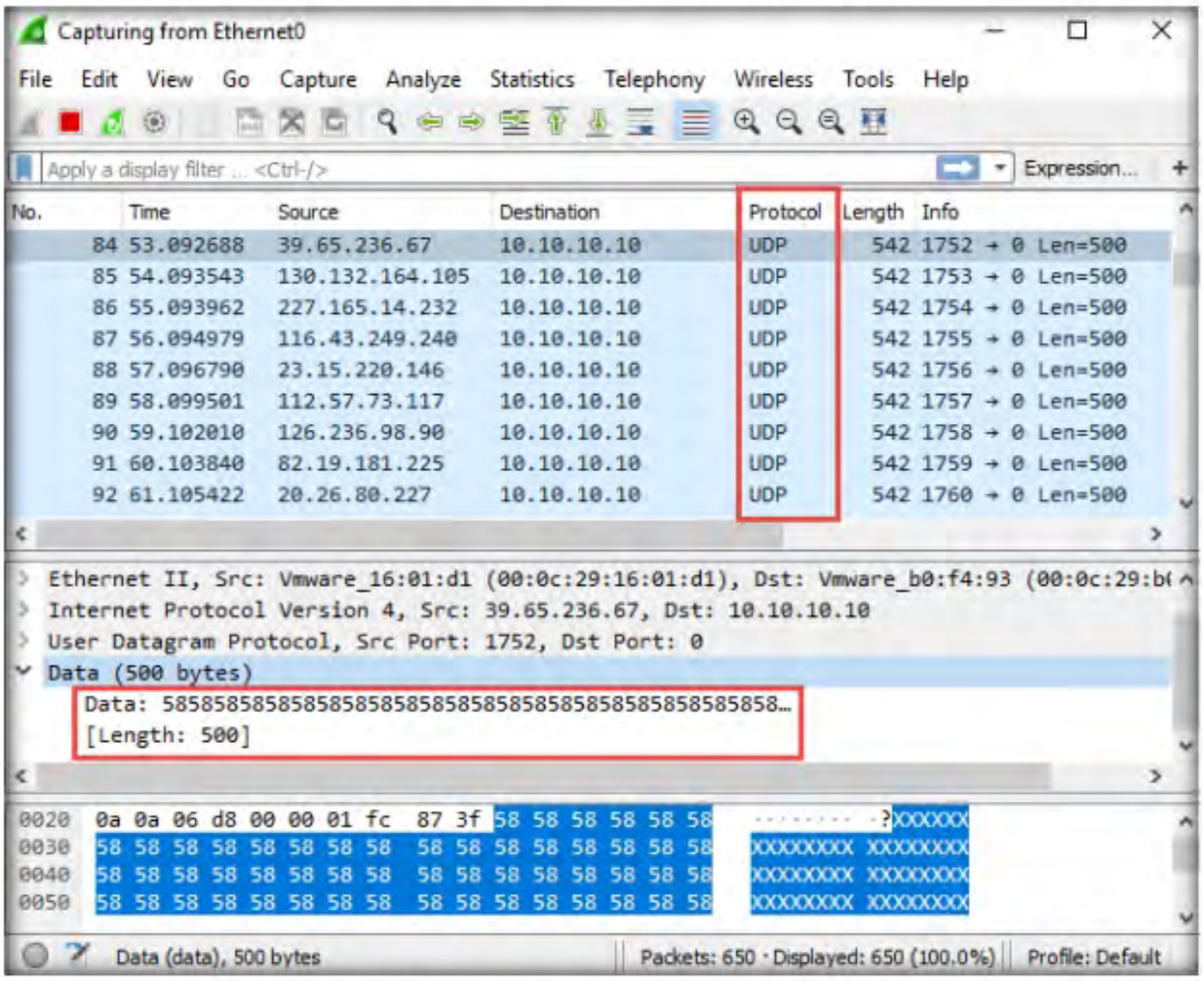


Figure 4.3.2: Wireshark capturing UDP packets in the target machine (Windows 10)

TASK 3.2

Send TCP SYN Request

11. Switch to the **Parrot Security** virtual 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.

Note: The IP addresses might vary in your lab environment.

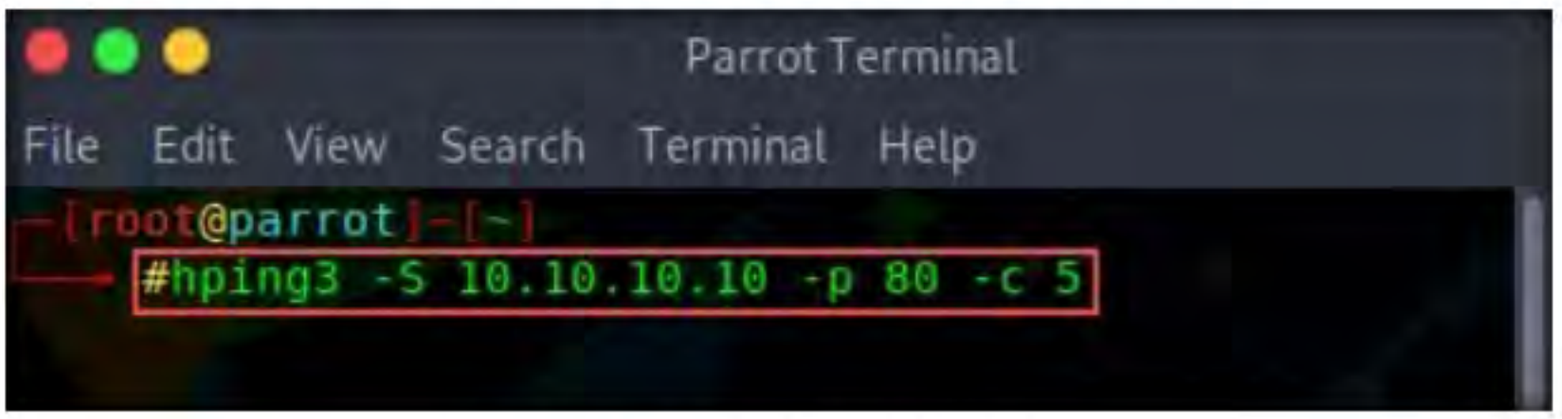


Figure 4.3.3: Hping3 sending TCP SYN packets

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

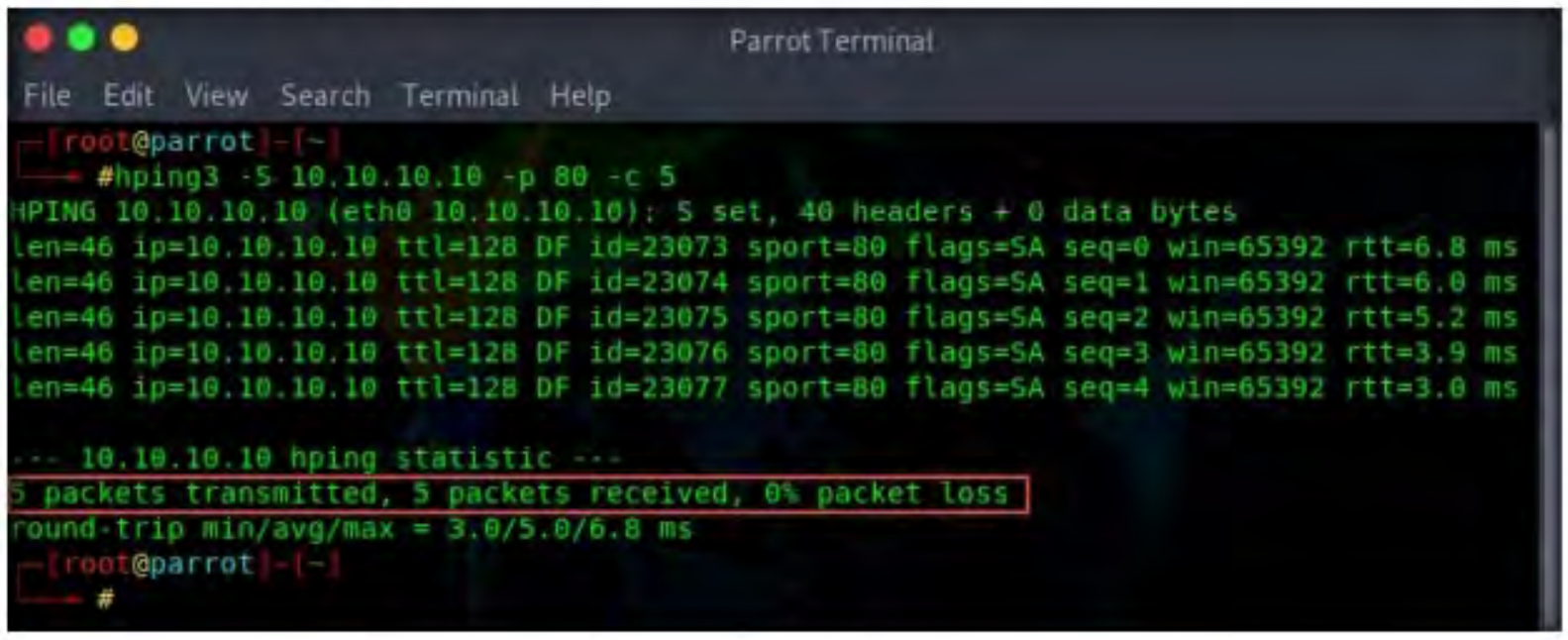


Figure 4.3.4: Hping3 sent TCP SYN packets to the target machine

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

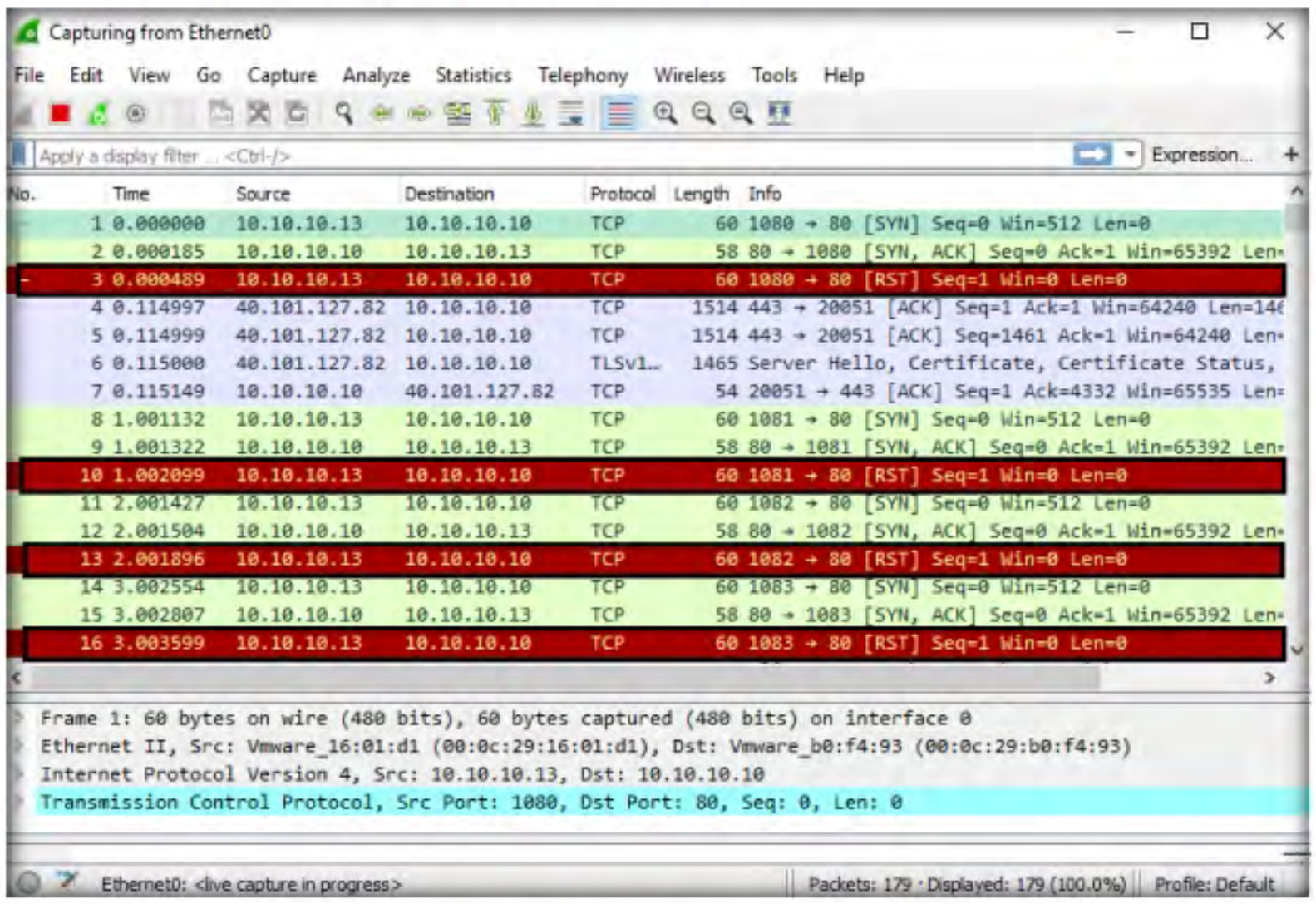


Figure 4.3.5: Wireshark TCP SYN Packets captured in the target machine

TASK 3.3

Perform TCP Flooding

14. Switch to the **Parrot Security** virtual machine and try to flood the target machine (here, **Windows 10**) with TCP packets.
15. In the **Parrot Terminal** window, type **hping3 <Target IP Address> --flood** (here, target IP address is **10.10.10.10**) and press **Enter**.

Note: **--flood**: performs the TCP flooding.

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

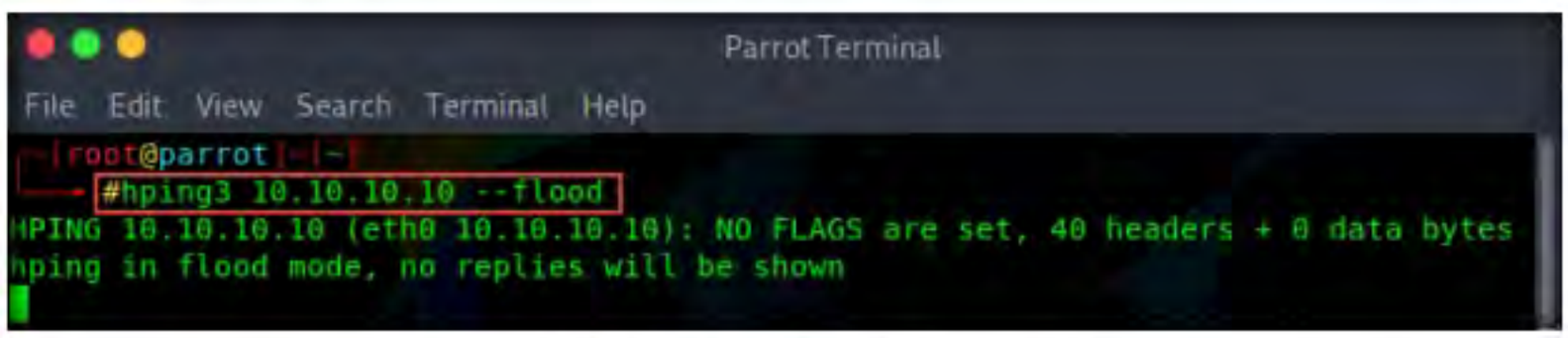


Figure 4.3.6: TCP Packets flooded to Target machine

17. Switch to **Windows 10** (target machine) and stop the packet capture in the Wireshark window after a while.
18. Observe the **Wireshark** window, which displays the TCP packet flooding from the host machine.

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

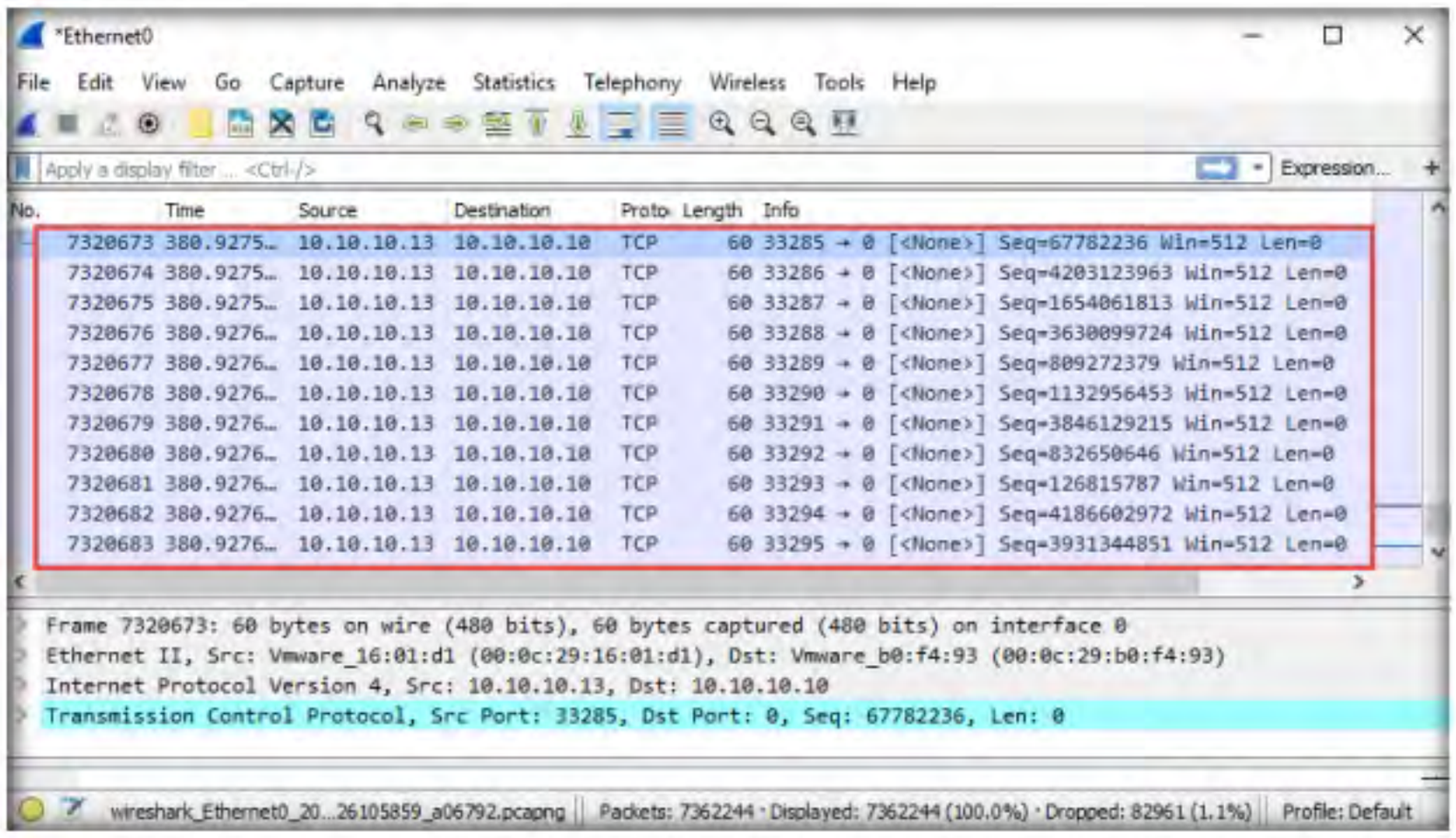


Figure 4.3.7: TCP Packets in Wireshark

19. 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.

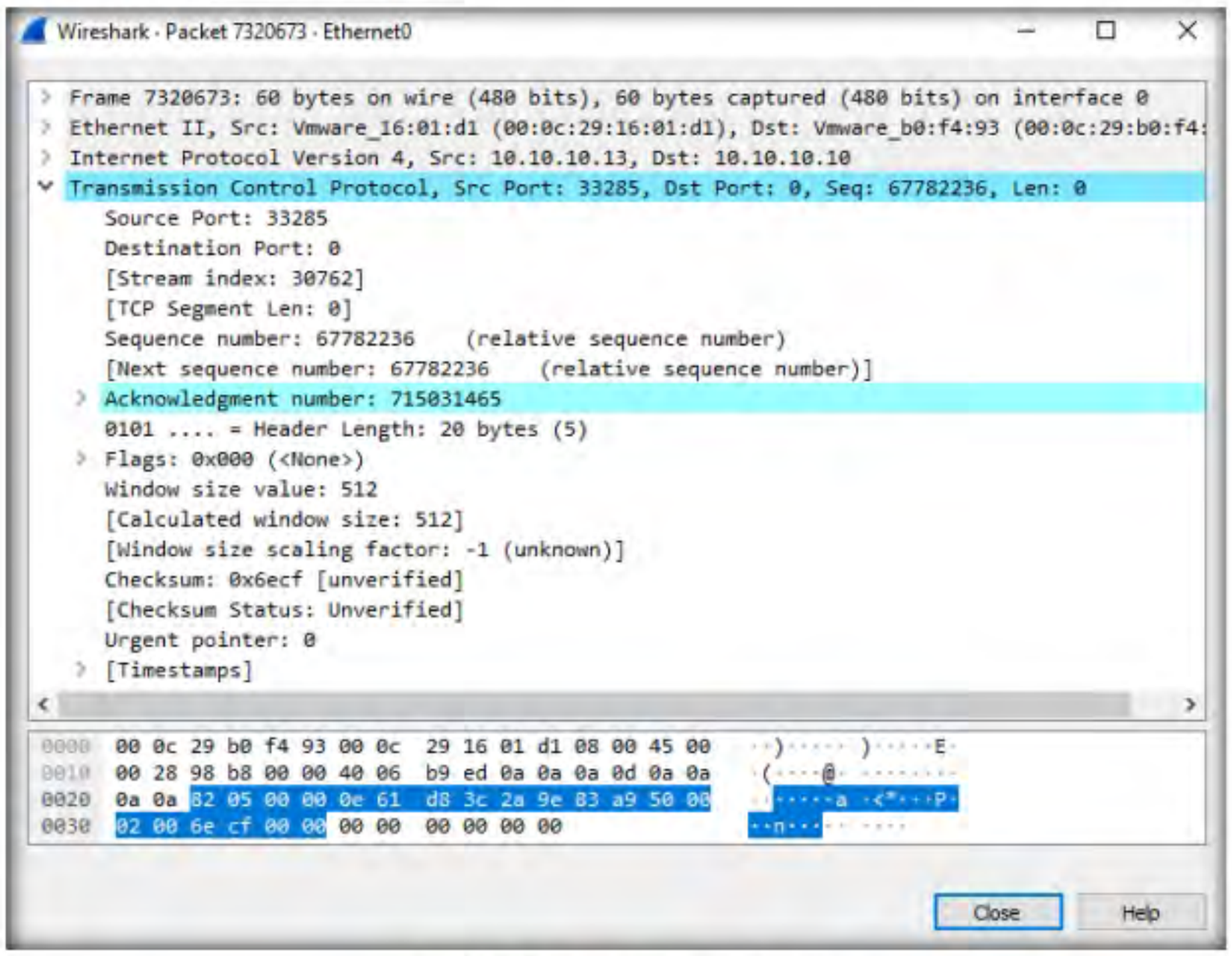


Figure 4.3.8: TCP packet Stream information

20. This concludes the demonstration of evading the IDS and firewall using various evasion techniques in Hping3.
21. Close all open windows and document all the acquired information.
22. Turn off the **Parrot Security** virtual machine.

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.

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

1. Turn on the **Windows Server 2016** virtual machine and log in the with credentials **Administrator/Pa$$w0rd**.
2. Navigate to **Control Panel → System and Security → Windows Firewall → Turn Windows Firewall on or off**, enable Windows Firewall, and click **OK**.

TASK 4.1

Append Custom Binary Data

3. Switch to the **Windows 10** virtual machine and launch **Nmap** by double-clicking on the **Nmap - Zenmap GUI** icon available on **Desktop**.
4. 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**.

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

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

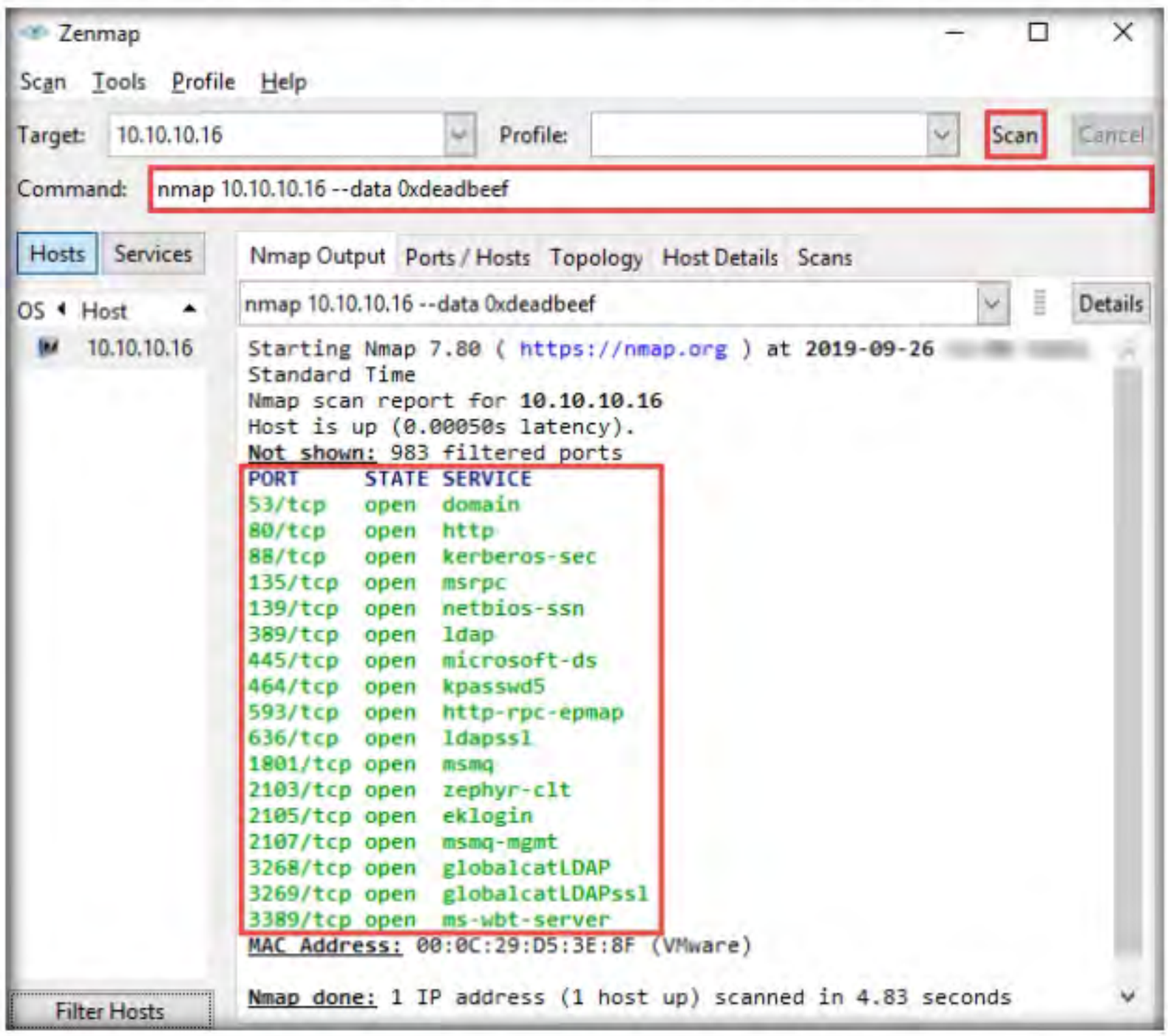


Figure 4.4.1: Zenmap scan results for Appending Custom Binary Data

TASK 4.2

Append Custom String

6. 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**.

Note: 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.

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

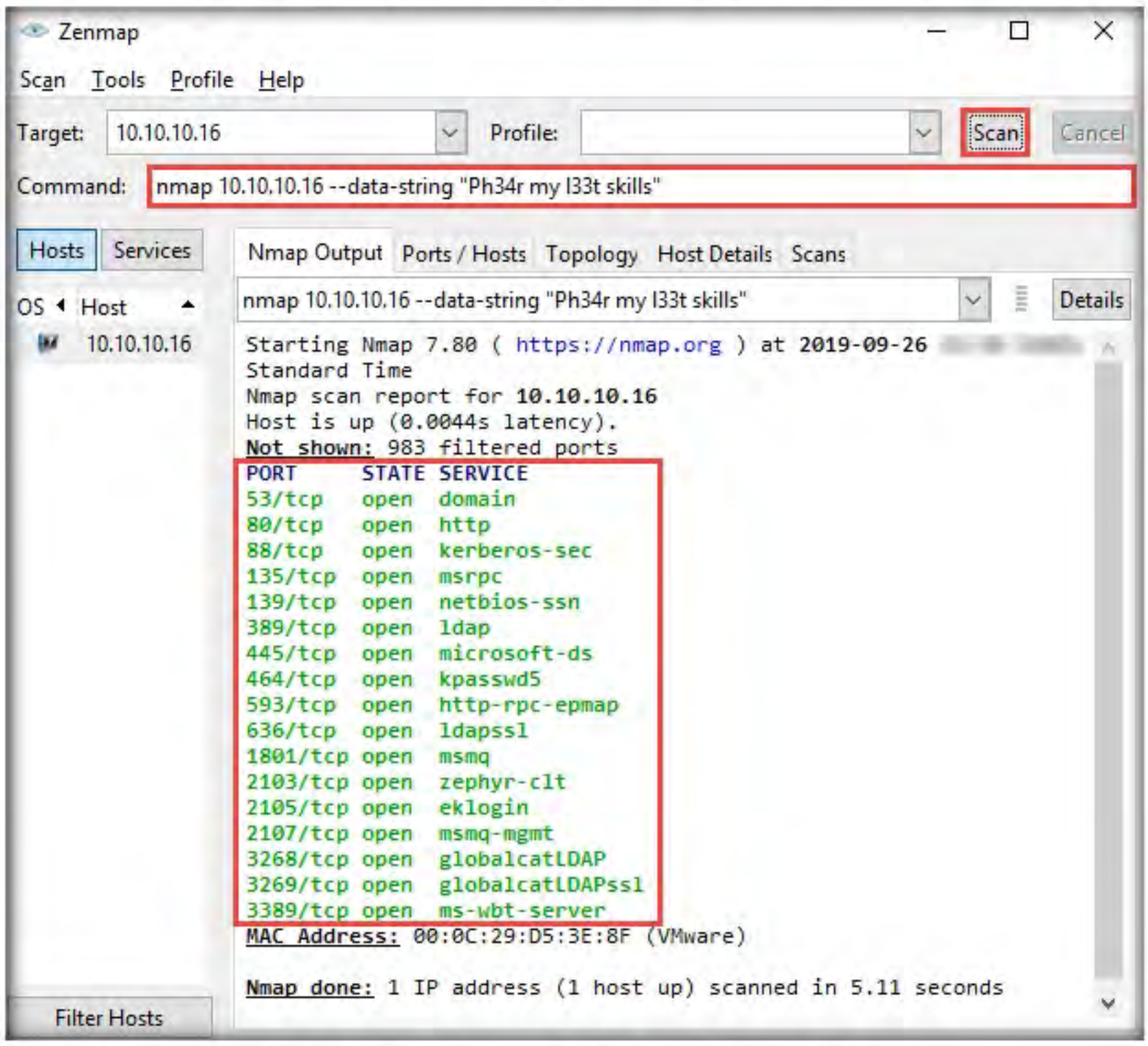


Figure 4.4.2: Zenmap scan results for Appending Custom String

TASK 4.3

Append Random Data

8. 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**.

Note: 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.

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

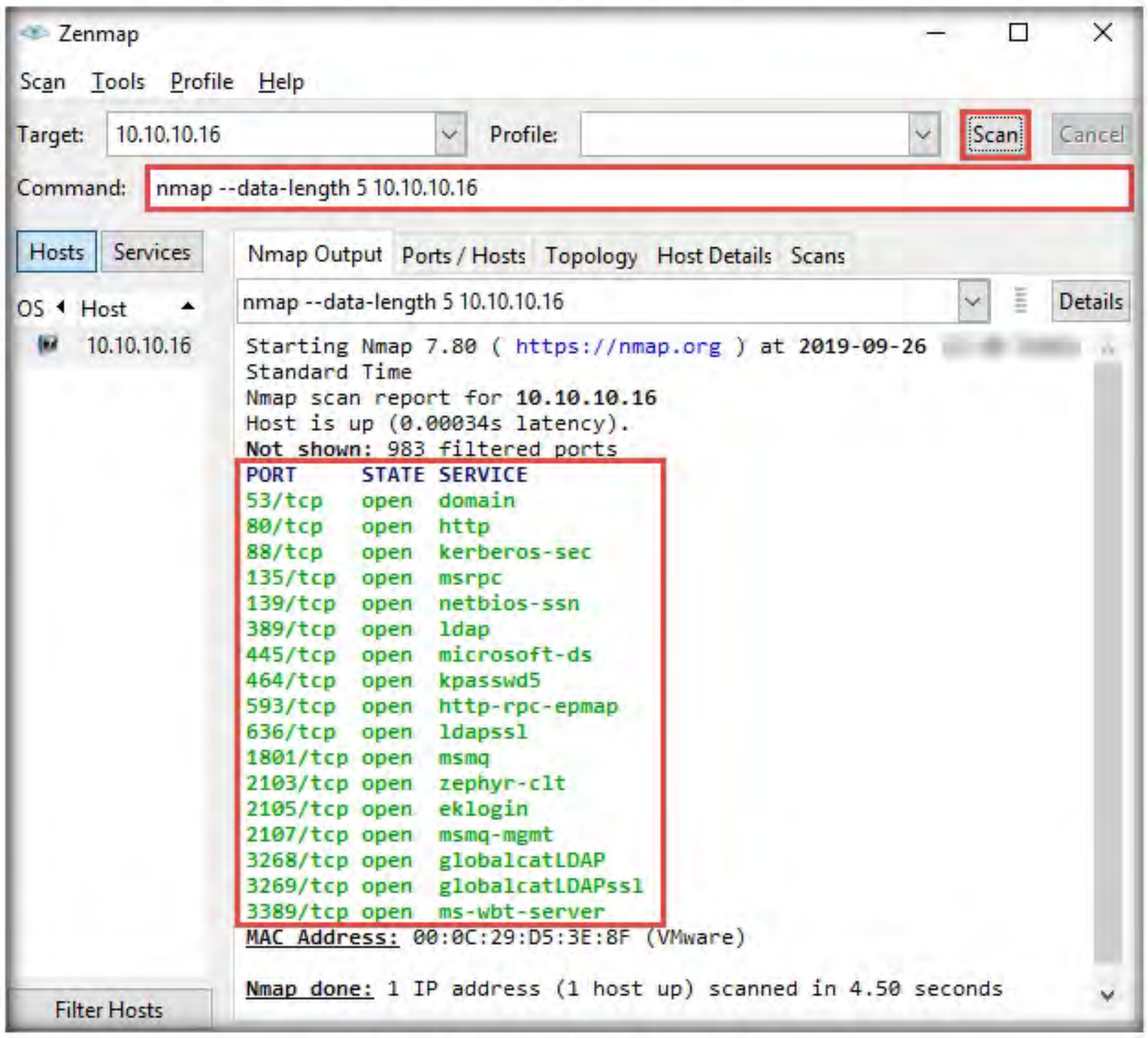


Figure 4.4.3: Zenmap scan results for Appending Random Data

TASK 4.4

Randomizing Host Order

10. 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**.

Note: 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.

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

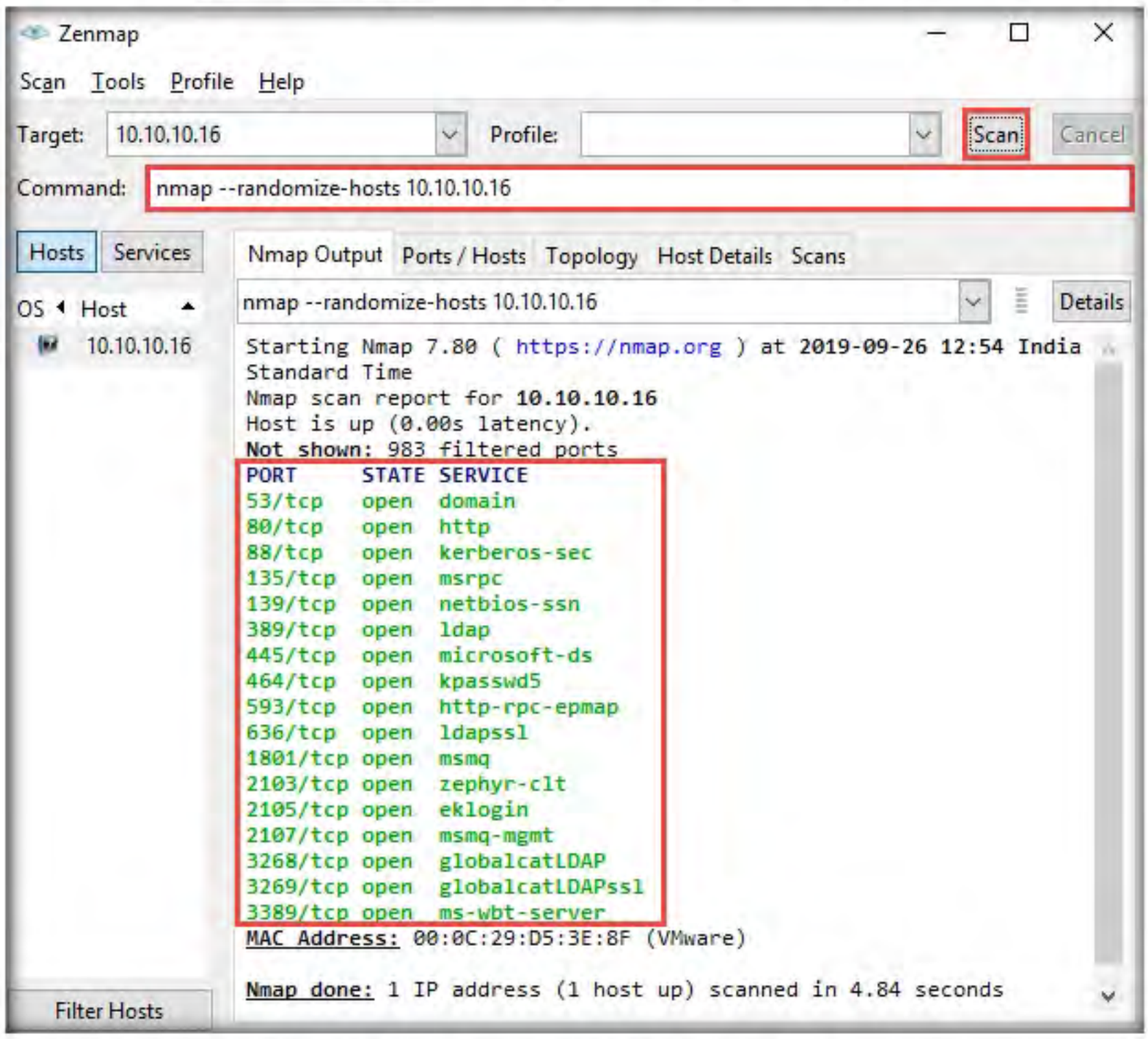


Figure 4.4.4: Zenmap scan results for Randomizing Host Order

TASK 4.5

Send Bad Checksums

12. 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**.

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

13. 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.

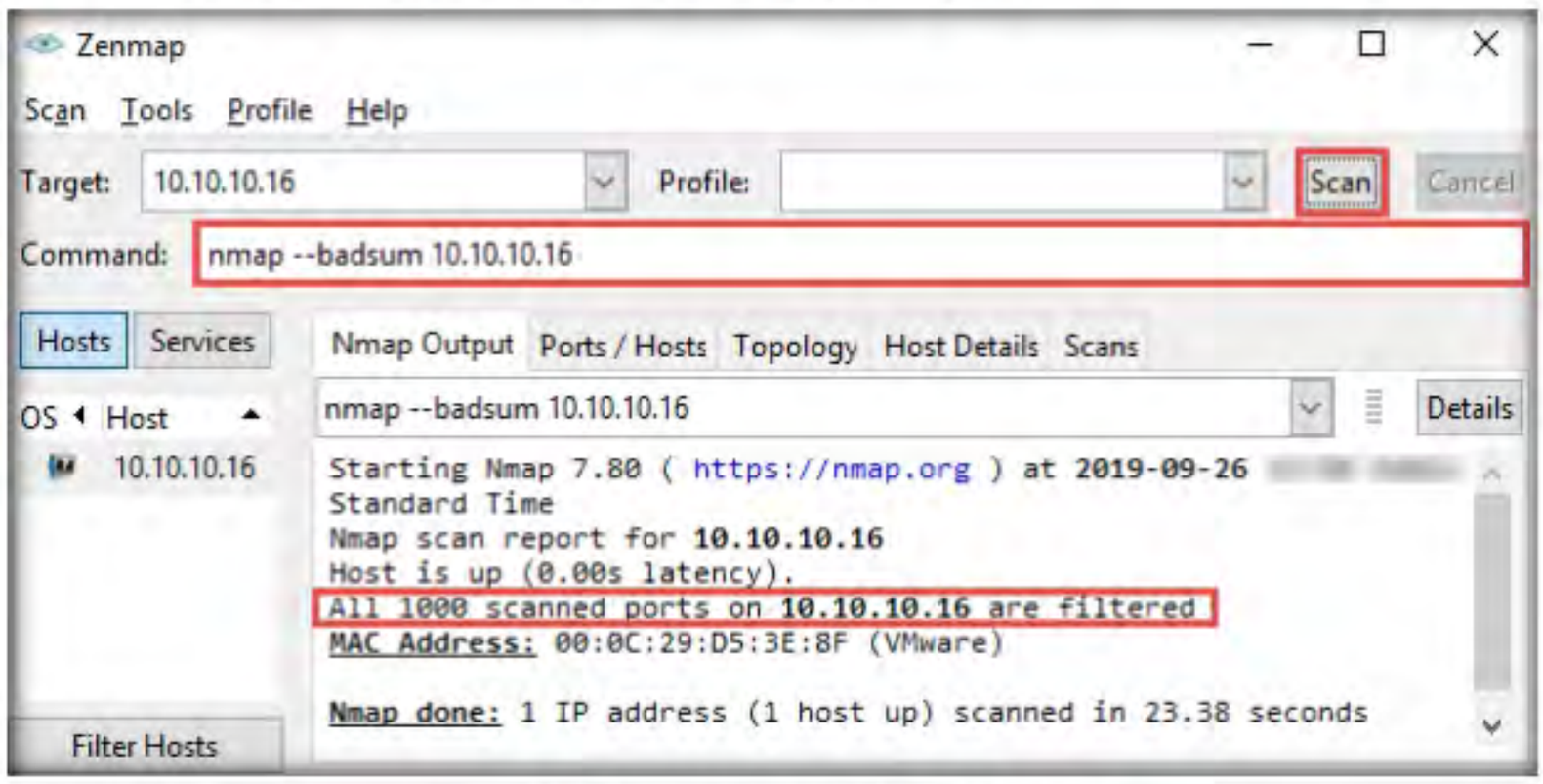


Figure 4.4.5: Zenmap scan results for Bad Checksum

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.

14. This concludes the demonstration of creating custom packets using Nmap to scan beyond the IDS and firewall.
15. Close all open windows and document all the acquired information.
16. Turn off the Windows Firewall in the **Windows 10** and **Windows Server 2016** virtual machines by navigating to **Control Panel → System and Security → Windows Firewall → Turn Windows Defender Firewall on or off**.
17. Turn off **Windows Server 2016** virtual machine.

TASK 5

Browse Anonymously using Proxy Switcher

Here, we will use Proxy Switcher to browse the Internet anonymously.

TASK 5.1

Install Proxy Switcher

1. In the **Windows 10** virtual machine, navigate to **E:\CEH-Tools\CEHv11 Module 03 Scanning Networks\Proxy Tools\Proxy Switcher** and double-click **ProxySwitcherStandard.exe**.

Note: If a **User Account Control** window appears, click **Yes**.

2. Follow the installation steps to install Proxy Switcher using all default settings.
3. Once the installation is complete, uncheck all options in the final step of the wizard, and click **Finish**.

Proxy Switcher allows you to surf the Internet anonymously without disclosing the IP address of your system, and helps to access various blocked sites in the organization. It avoids all types of limitations imposed by target sites.

Figure 4.5.1: ProxySwitcher finish wizard

TASK 5.2

Configure Local Proxy in a Web Browser

4. Now, launch the **Firefox** browser.

Note: If a **Default Browser** pop-up appears, click **Not now**.

5. Click the **Open menu** icon in the top-right corner of the browser window and click **Options**.

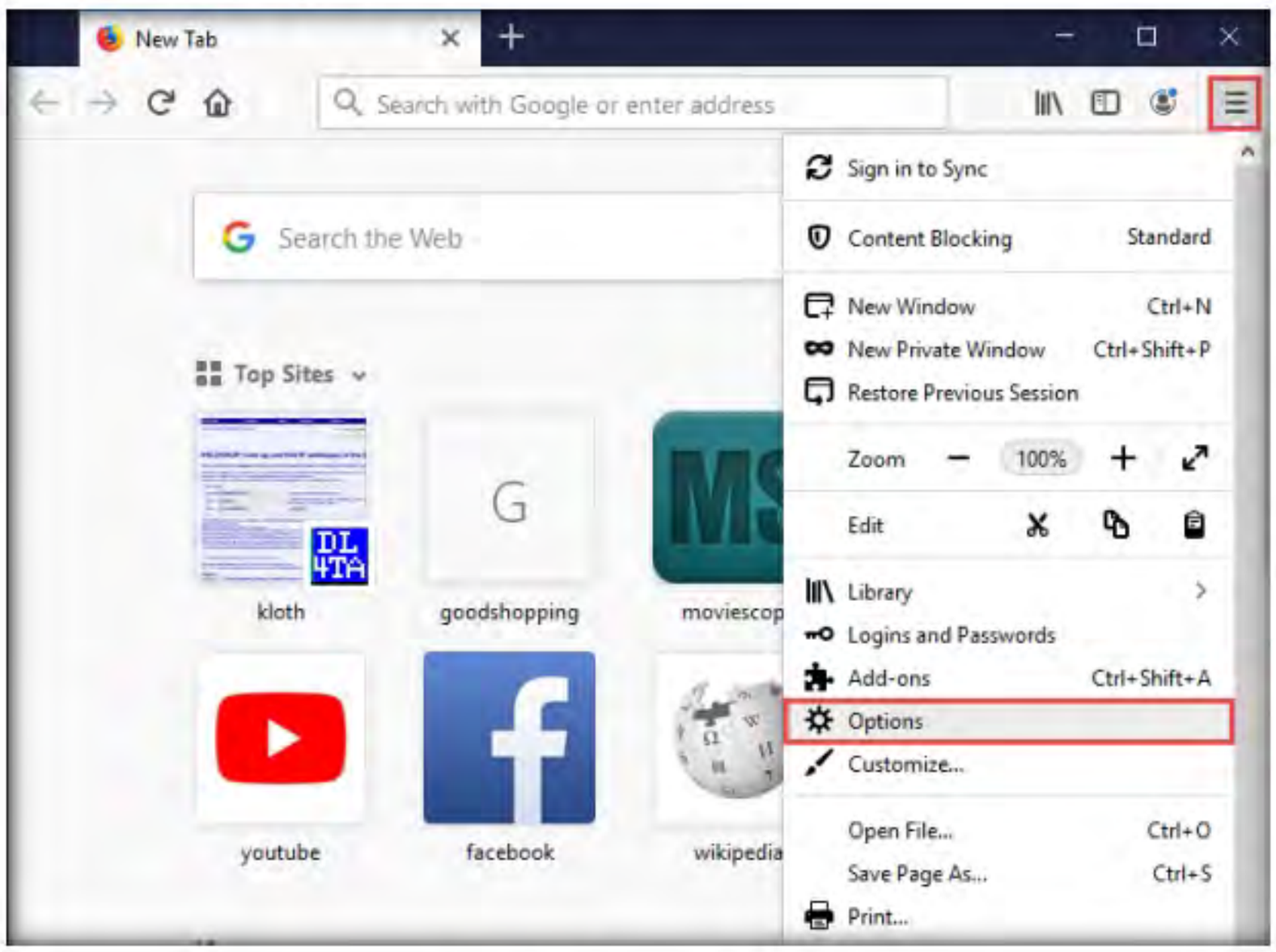


Figure 4.5.2: Firefox clicking on Options

6. In the **Options** wizard, scroll down to the end of the page and click **Settings...** under the **Network Settings** section.

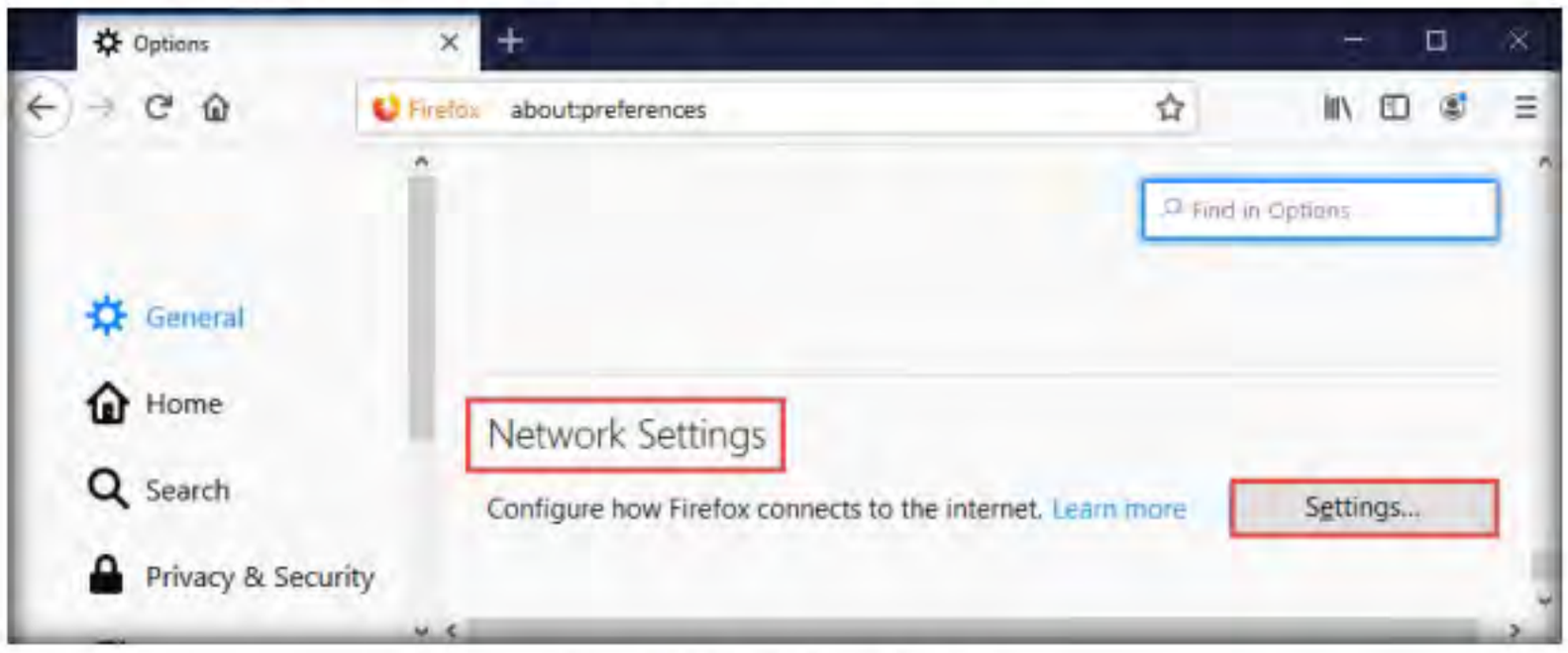


Figure 4.5.3: Firefox Network Settings

7. The **Connection Settings** window appears; under the **Configure Proxy Access to the Internet** section, ensure that the **Use system proxy settings** radio button is selected. Click **OK** and close the **Firefox** browser window.

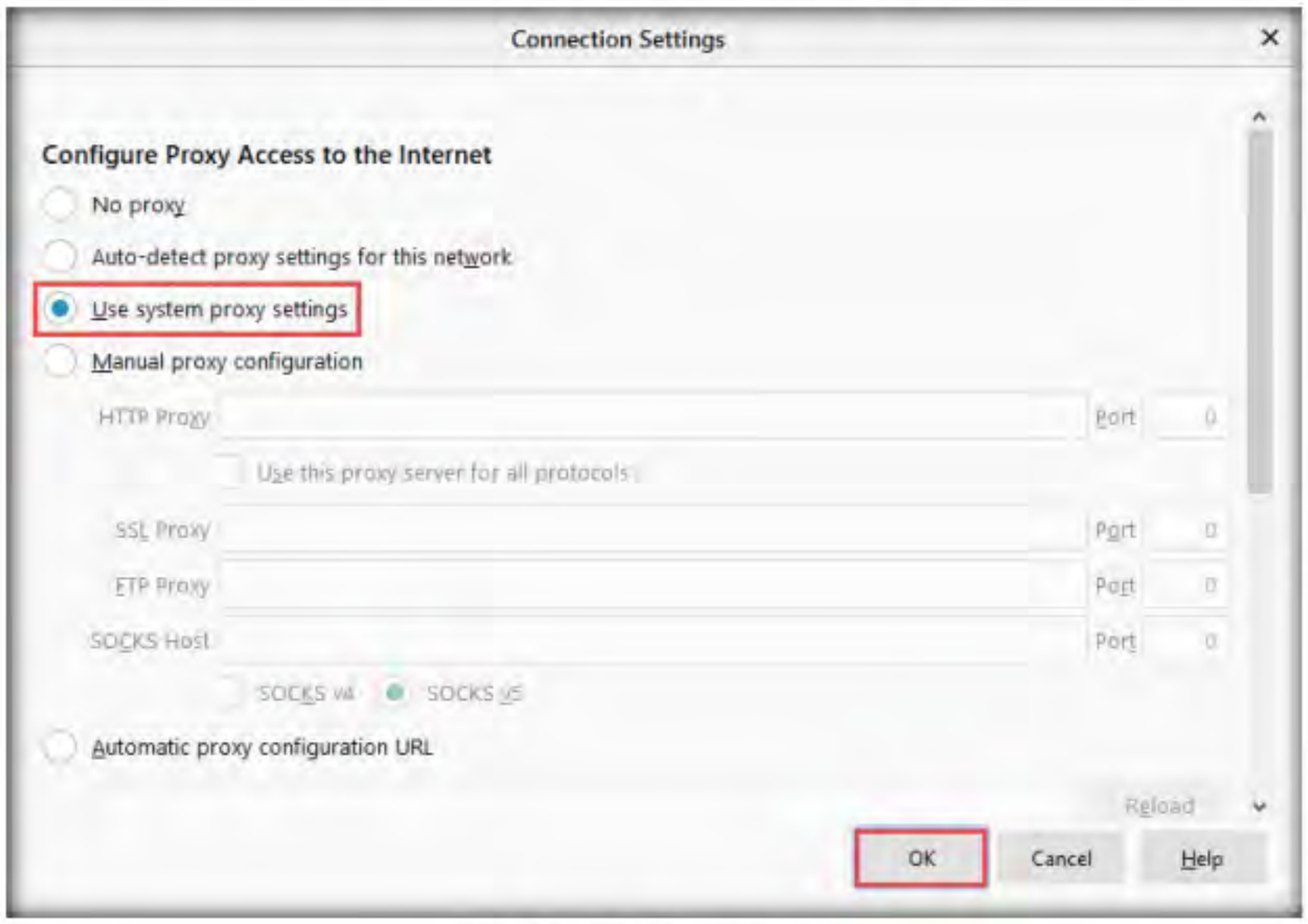


Figure 4.5.4: Firefox Connection Settings

8. Now, click on the **Start** menu and launch **ProxySwitcher Standard** from the applications.

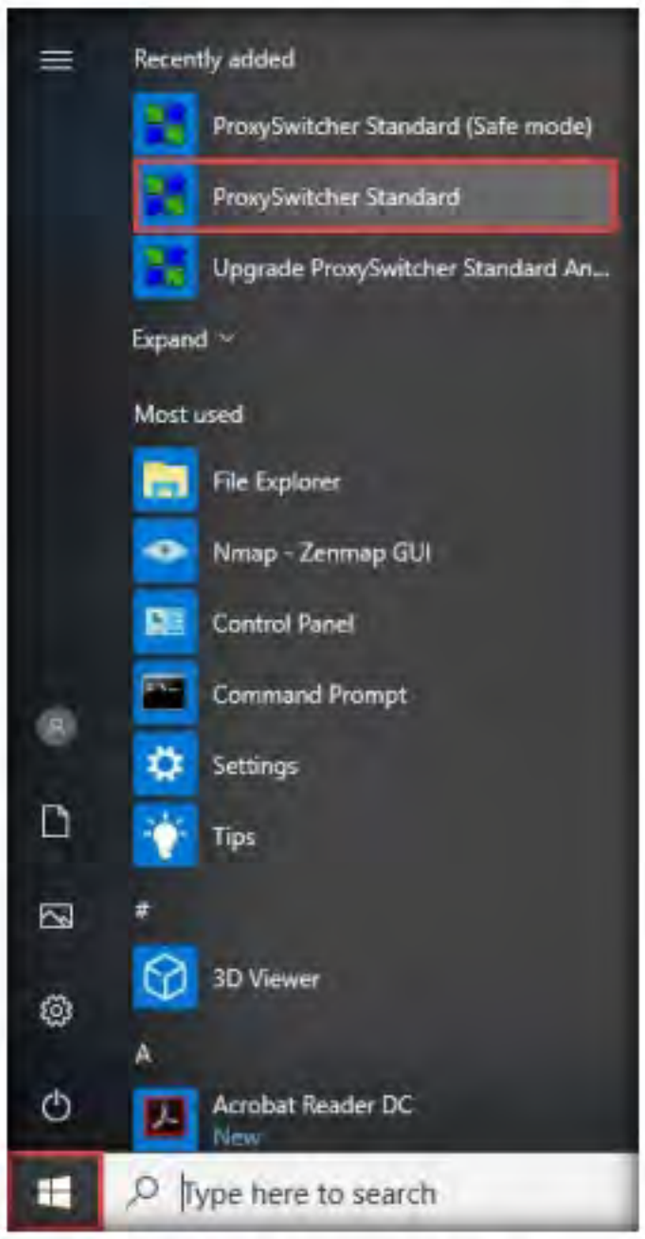


Figure 4.5.5: Windows 10 Apps list

9. The **ProxySwitcher Standard** loads, and its icon appears on **Taskbar**.
10. Click the **Taskbar** icon in the bottom-right corner of the desktop and the click **ProxySwitcher Standard** icon to launch the application.

Figure 4.5.6: Selecting ProxySwitcher Standard icon from the taskbar

TASK 5.3

Configure Proxy Switcher

11. The **Please Register** window appears; click the **Start 15 Day Trial** button to proceed.
12. The **Common Tasks Wizard** window appears; under **Welcome to the Proxy Switcher**, click **Next**.

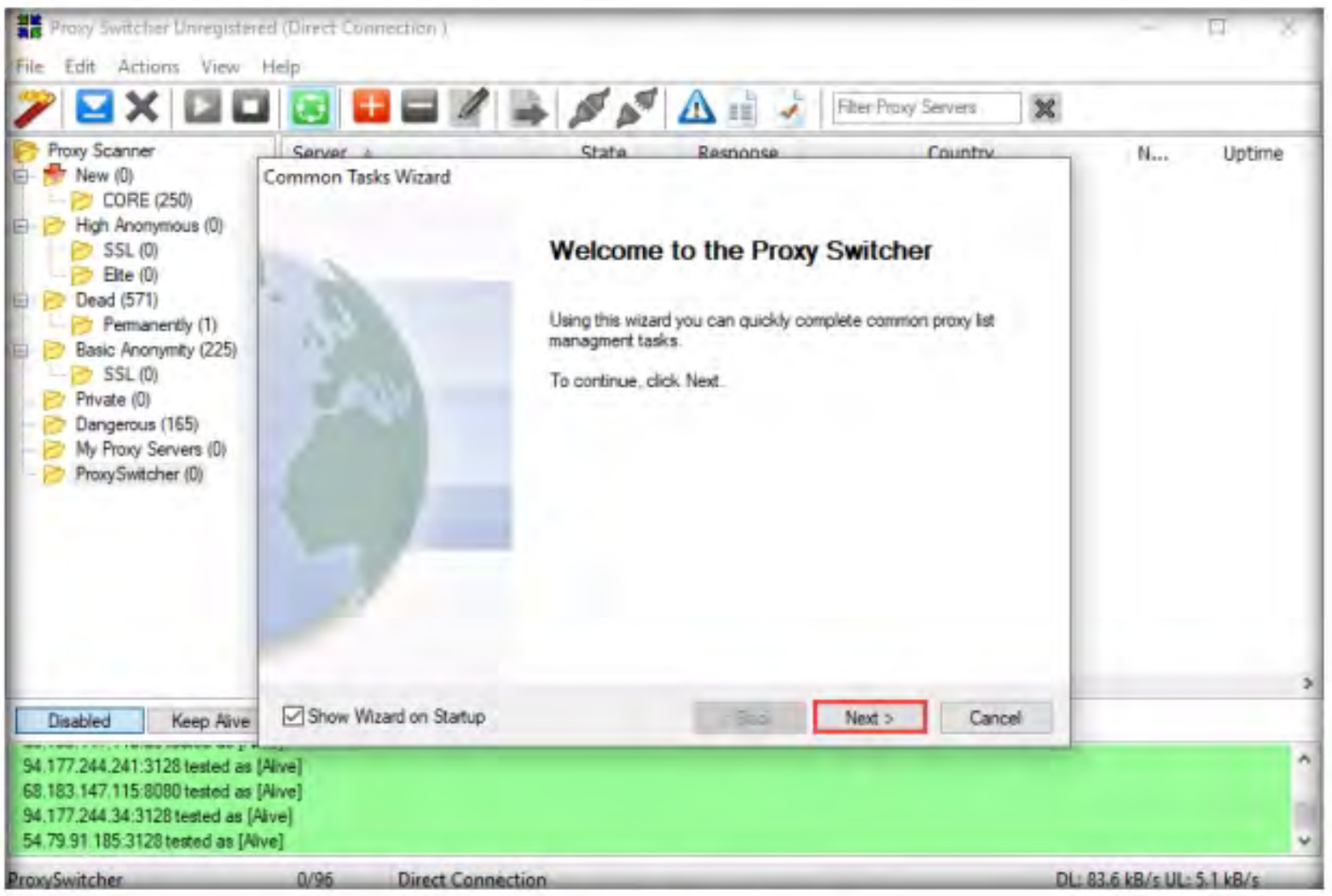


Figure 4.5.7: Proxy List wizard

13. Ensure that the **Find New Server, Rescan Servers, Recheck Dead** radio button is selected under the **Common Tasks** section, and click **Finish**.

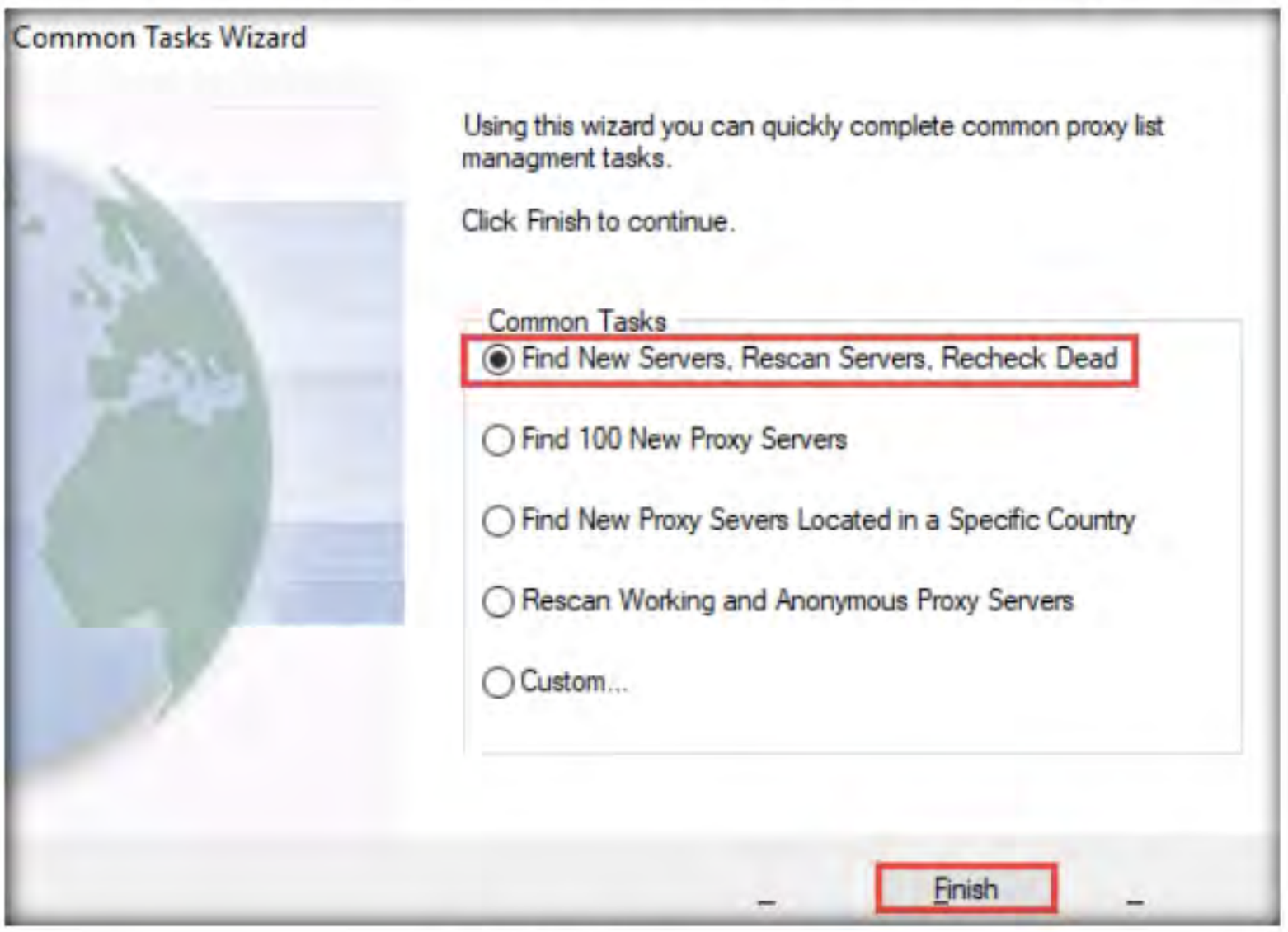


Figure 4.5.8: Selecting common tasks

14. **Proxy Switcher** window appears, showing a list of proxy servers in the right pane, as shown in the following screenshot.

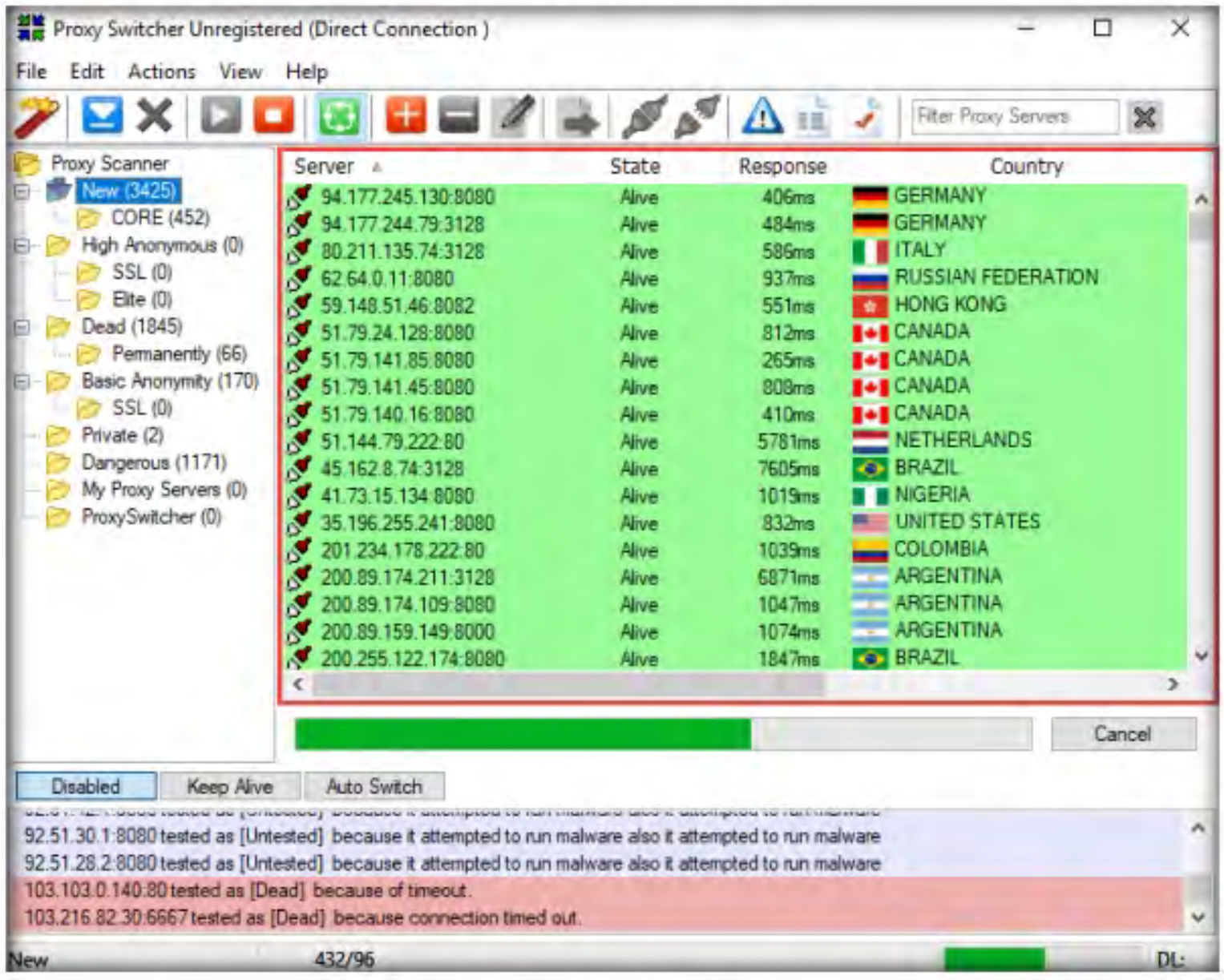


Figure 4.5.9: List of downloaded Proxy Servers

Note: The list of proxy servers might vary in your lab environment.

15. Observe the search bar below the server section; once it is completed, click the **Download Proxy Lists** icon ( ) to download the proxy list.

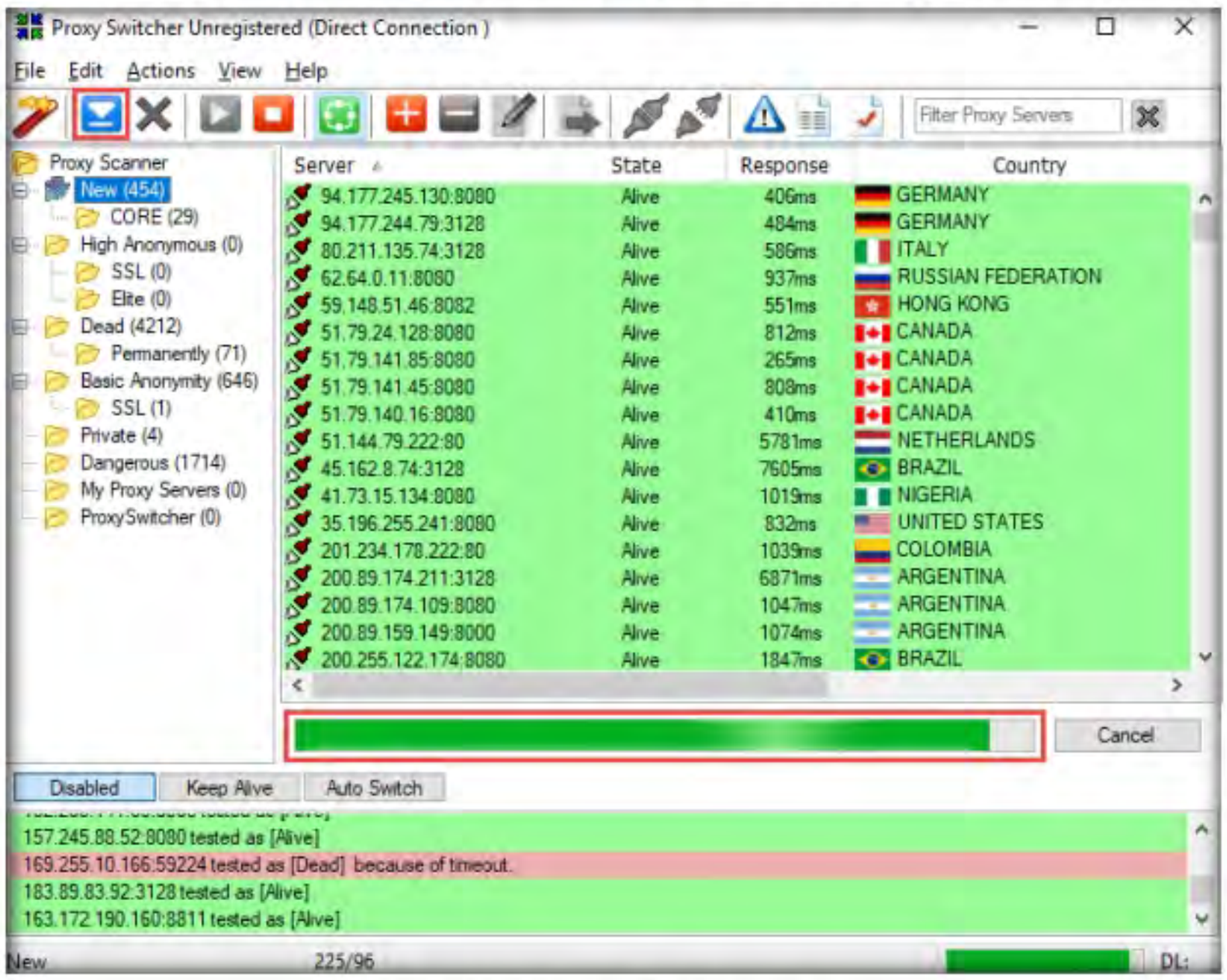


Figure 4.5.10: Downing a list of proxy servers

16. Wait until all the proxy servers are downloaded. This can take a significant amount of time.
17. If you have enough downloaded proxy servers, you can click the **Stop Download** ( ) icon to cancel the download.

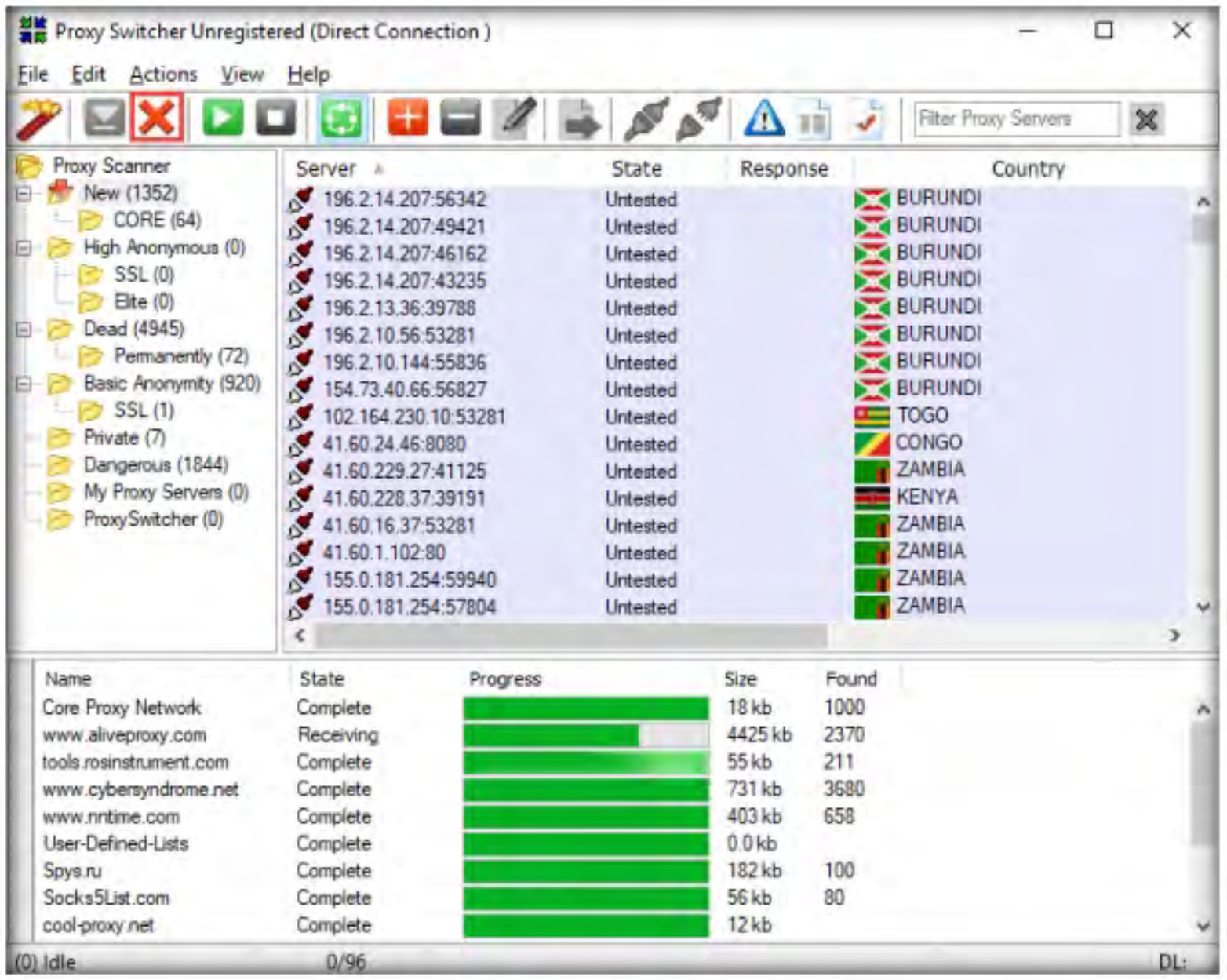


Figure 4.5.11: Proxies being downloaded

TASK 5.4

Assign Proxies

18. Click the **Basic Anonymity** folder in the left-hand pane to display a list of alive proxy servers, as shown in the screenshot.

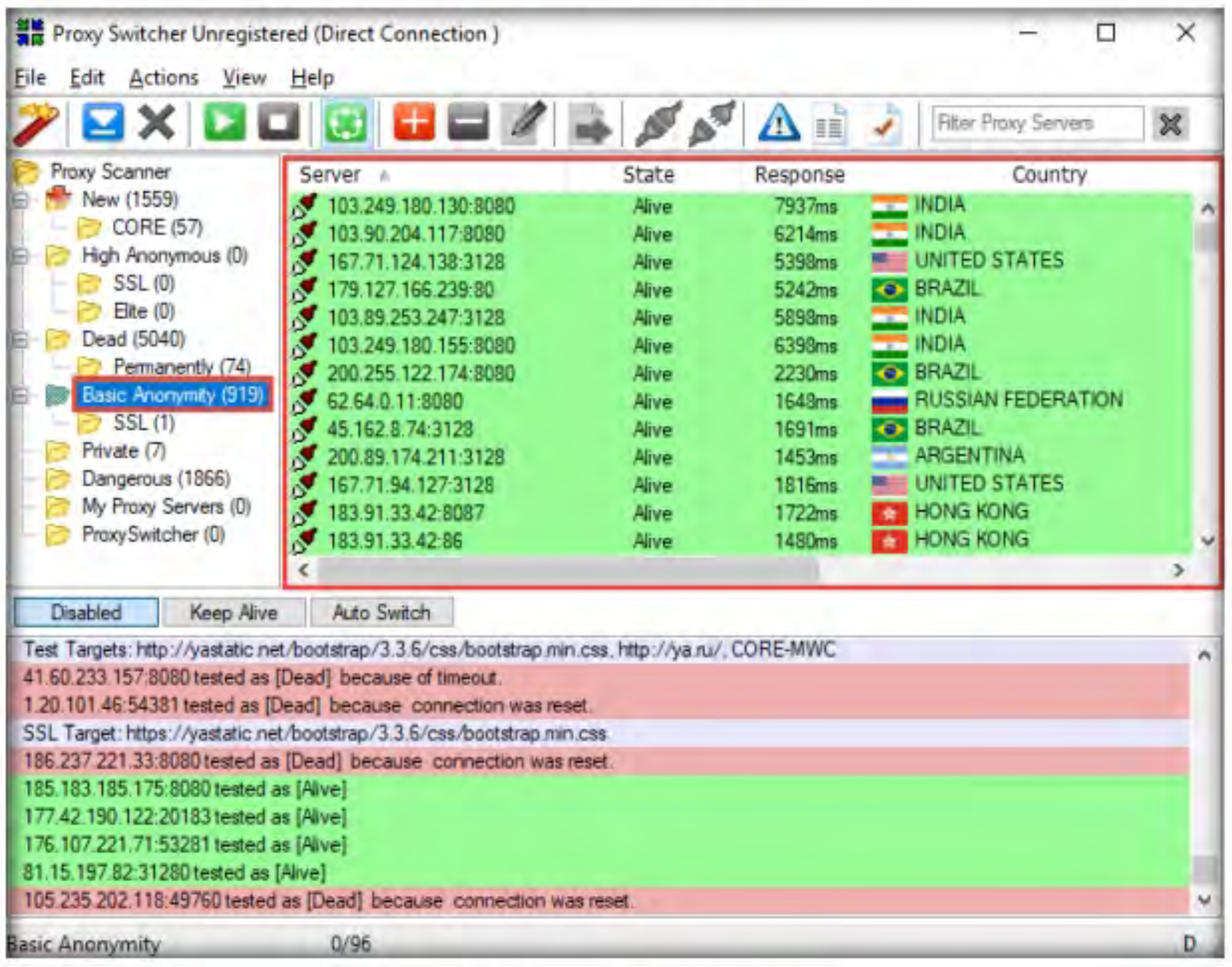


Figure 4.5.12: Searching for alive proxy servers

19. Select one proxy server IP address in the right-hand pane. To switch to the selected proxy server, click the **Switch to Selected Proxy Server** ( ) icon.

Note: The proxy selected in this lab might vary in your lab environment.

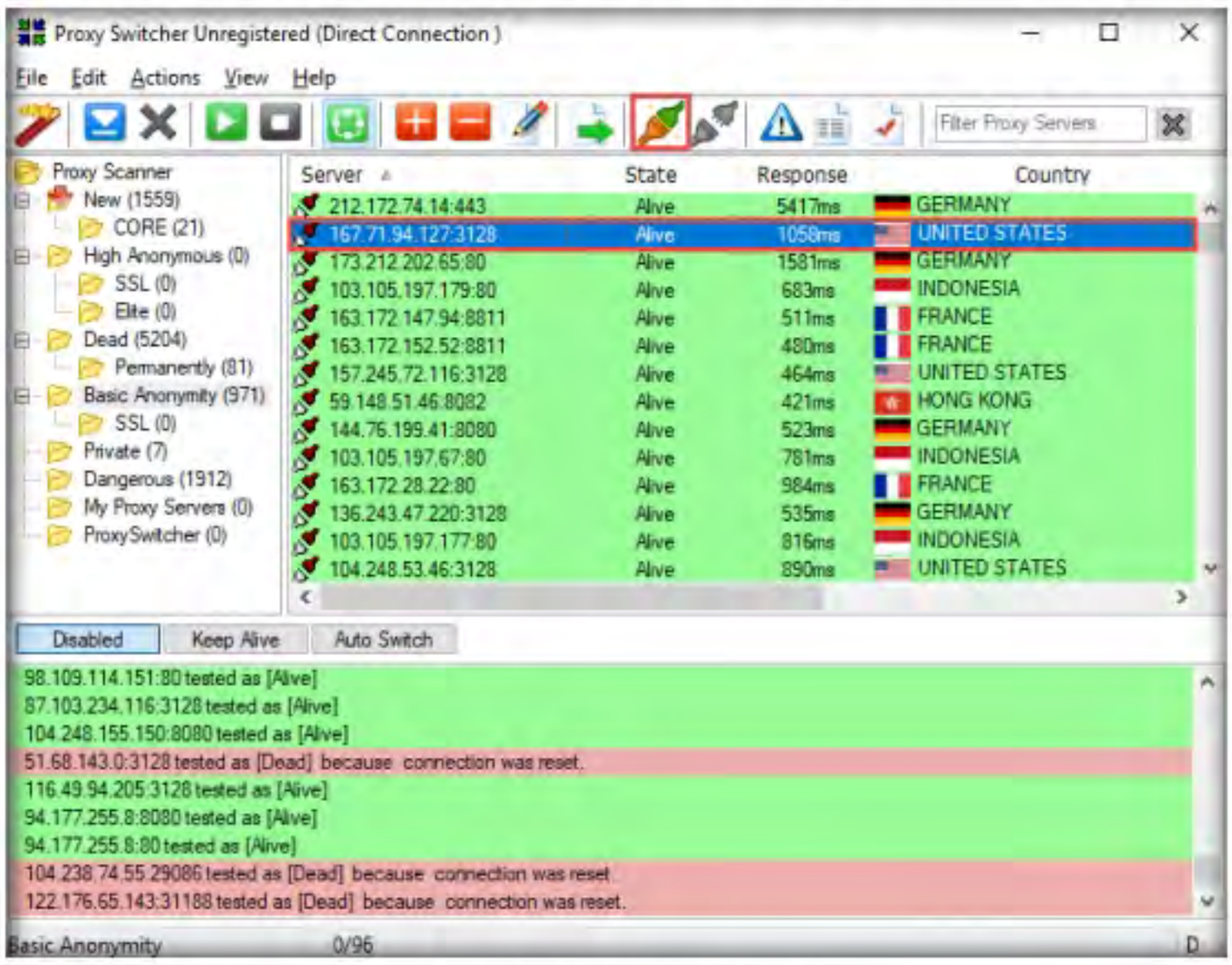


Figure 4.5.13: Selecting a proxy server

20. When the proxy server is connected, it will show the connection icon as .

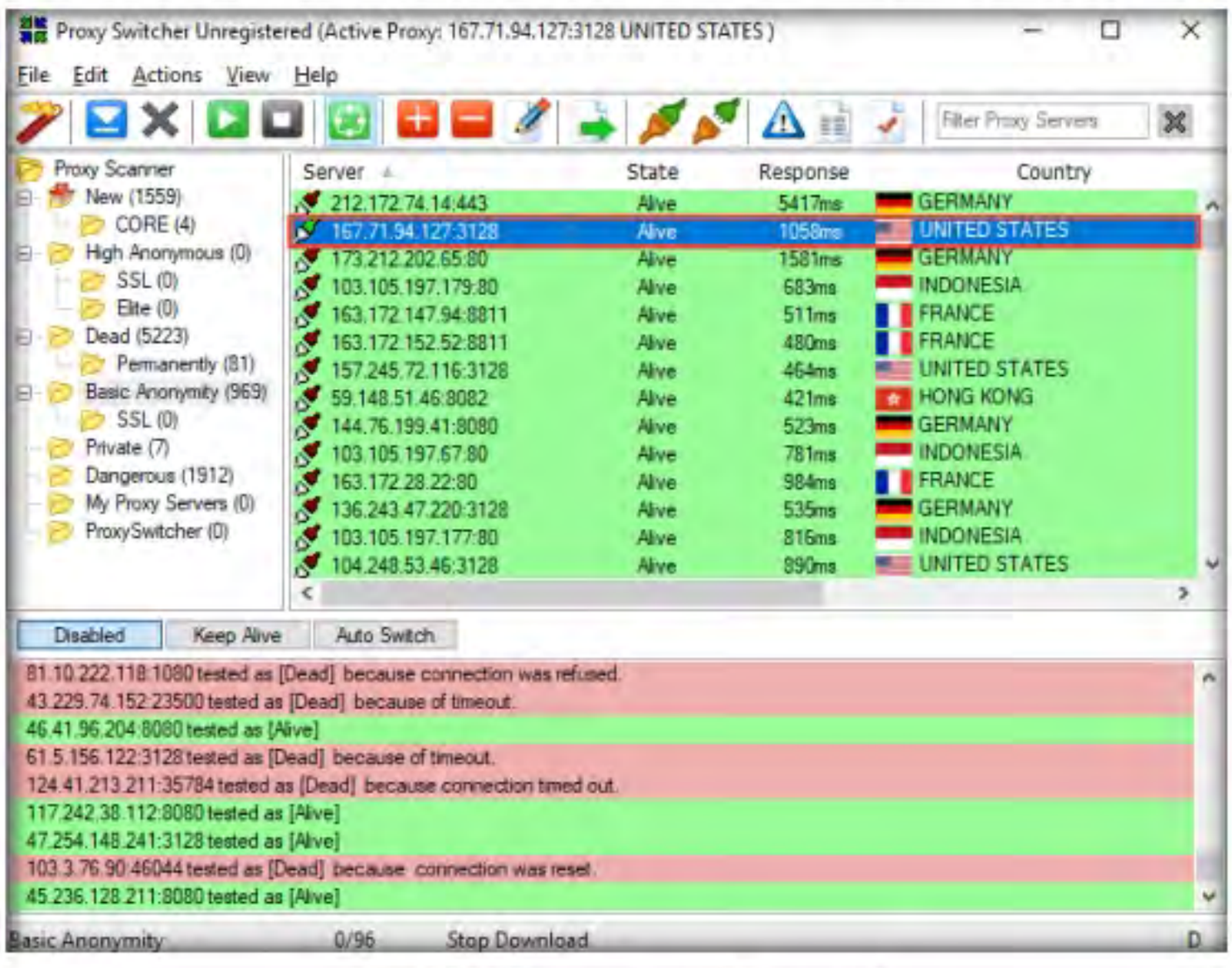


Figure 4.5.14: Proxy server successfully connected

TASK 5.5

Test Proxies

21. Launch the **Mozilla Firefox** web browser and enter the URL **http://www.proxyswitcher.com/check.php** to check the selected proxy-server connectivity. If the connection is successful, the following information is displayed in the browser:

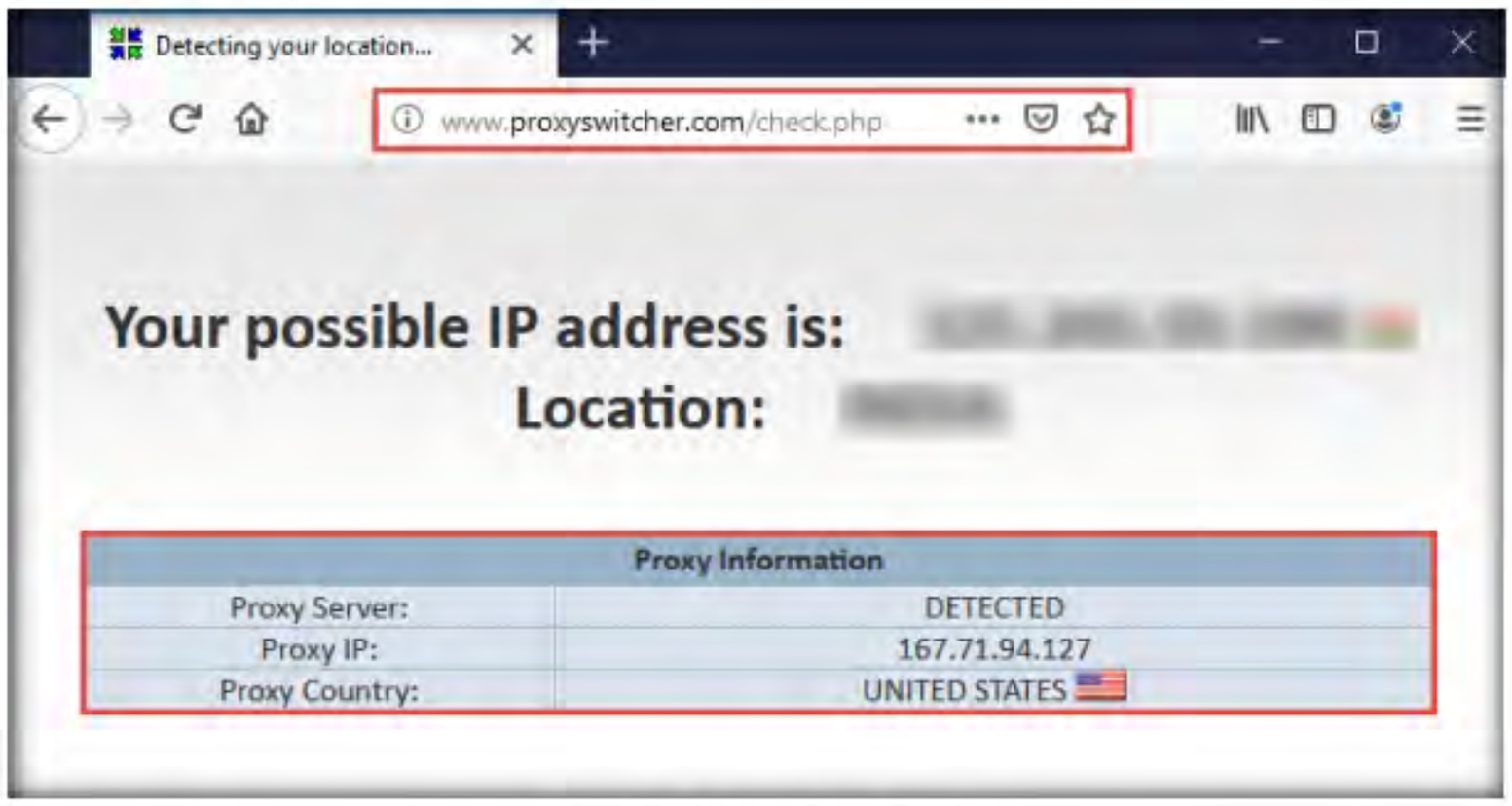


Figure 4.5.15: Detected Proxy Server

Note: The information displayed above may differ in your lab environment.

22. If the connection is unsuccessful, try selecting another proxy from **Proxy Switcher**, and repeat **Step 19**.

23. To ensure that the proxy is assigned, open a new tab and browse **https://www.google.com/**. In the search field, type **What is my ip** and press **Enter**.

24. The proxy IP address (**167.71.94.127**) is displayed, which infers that the legitimate address is masked and the proxy is in use.

Note: The displayed IP address might differ in your lab environment.

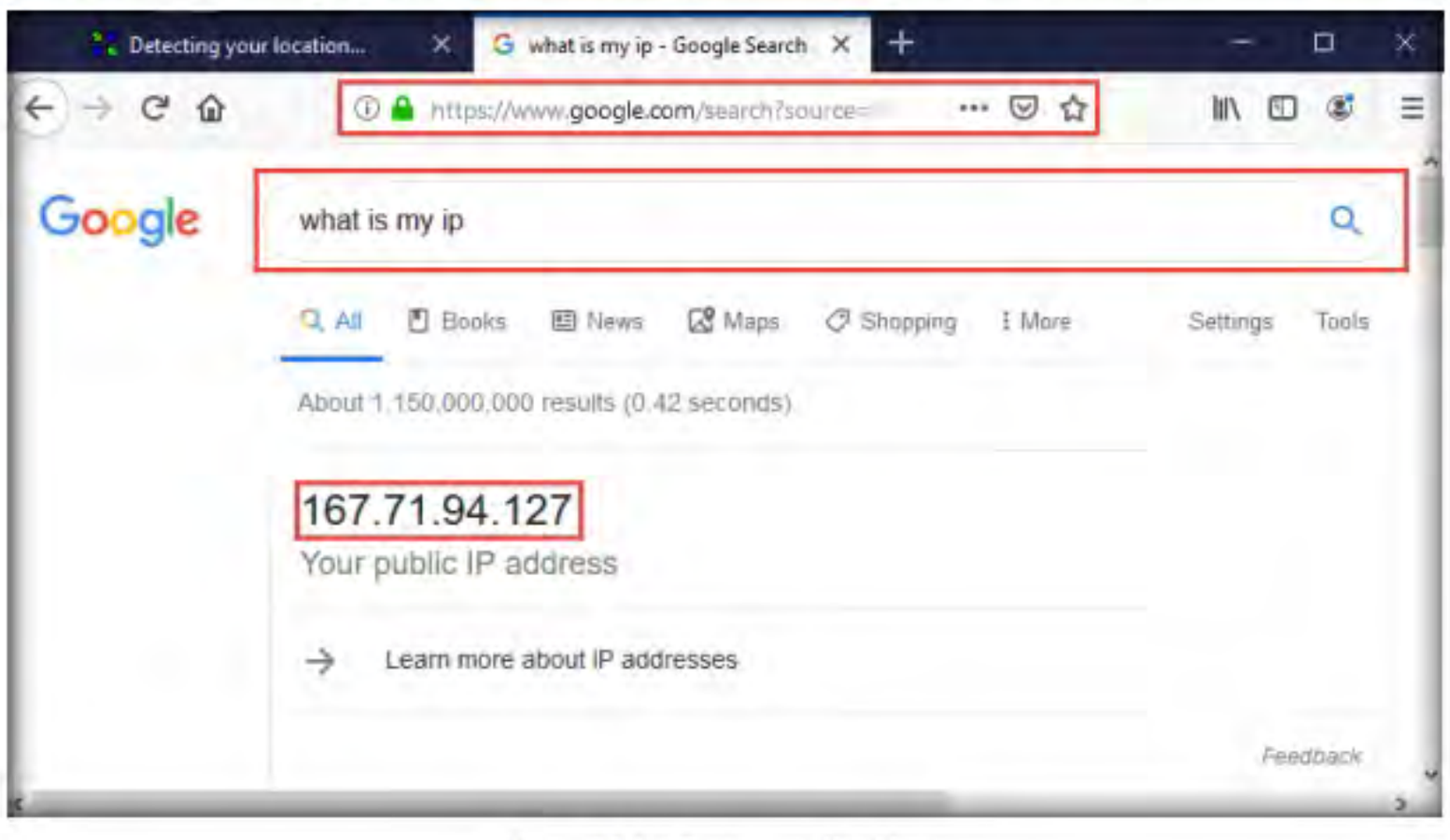


Figure 4.5.16: Testing your IP address

25. Open a new tab in your **web browser** and surf anonymously using this proxy.

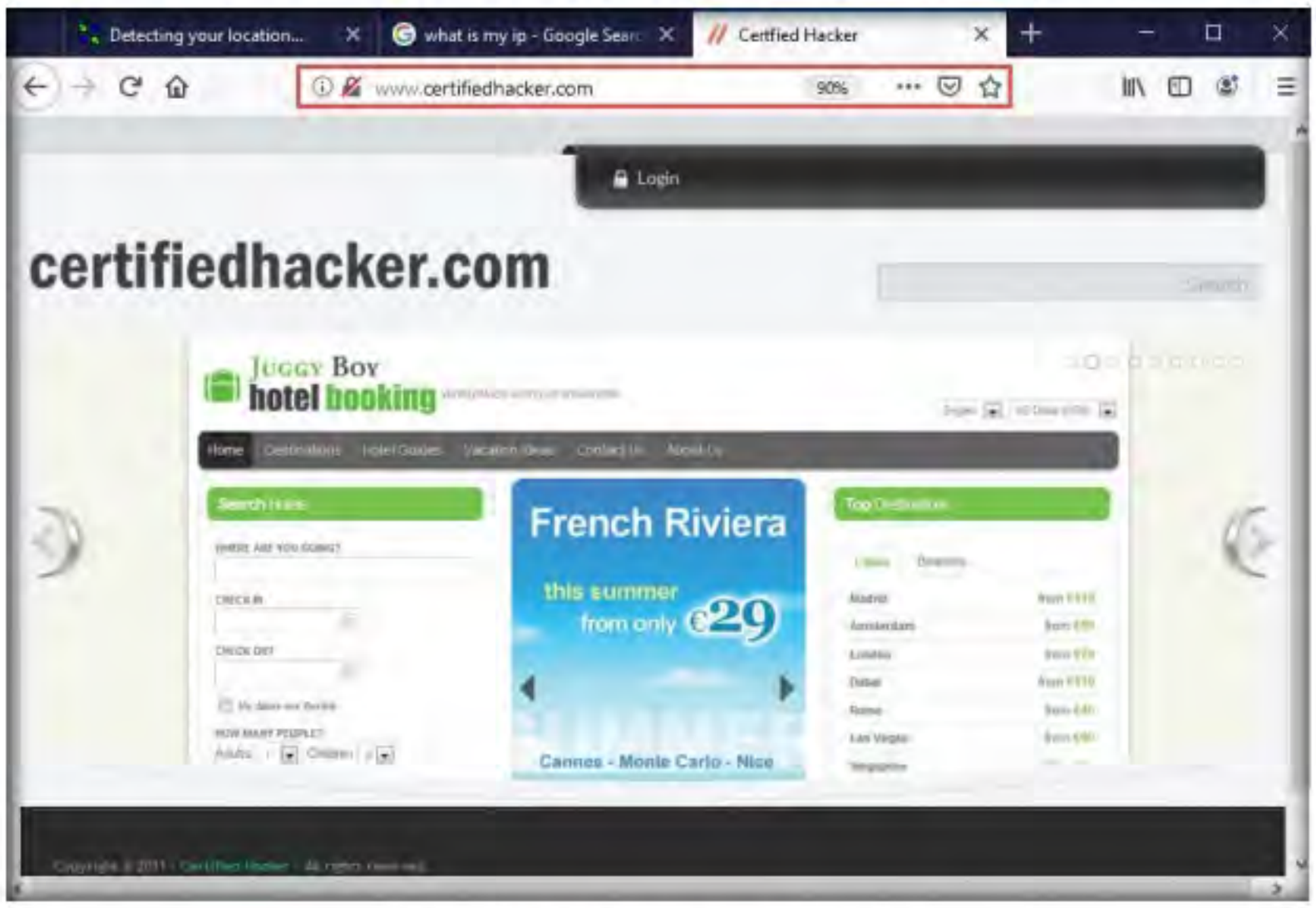


Figure 4.5.17: Surfing internet using Proxy server

26. This concludes the demonstration of anonymously surfing the Internet using Proxy Switcher.
27. Close all open windows and document all the acquired information.
28. Navigate to **Control Panel → Programs → Programs and Features** and uninstall the **Proxy Switcher** application

TASK 6 Browse Anonymously using CyberGhost VPN

Here, we will use CyberGhost VPN to browse the Internet anonymously.

TASK 6.1
Install CyberGhost VPN

1. In the **Windows 10** virtual machine, navigate to **E:\CEH-Tools\CEHv11 Module 03 Scanning Networks\Proxy Tools\CyberGhost VPN** and double-click **cgsetup_en_NMKh64apsGD93kapeBUc.exe**.

Note: If a **User Account Control** window appears, click **Yes**.

CyberGhost VPN hides your IP and replaces it with one of their choice, thereby allowing you to surf anonymously and access blocked or censored content. It encrypts the connection and does not keep logs, thus securing data.

2. **Downloading CyberGhost installer...** appears; once the **CyberGhost Setup** window appears, click **Accept**.

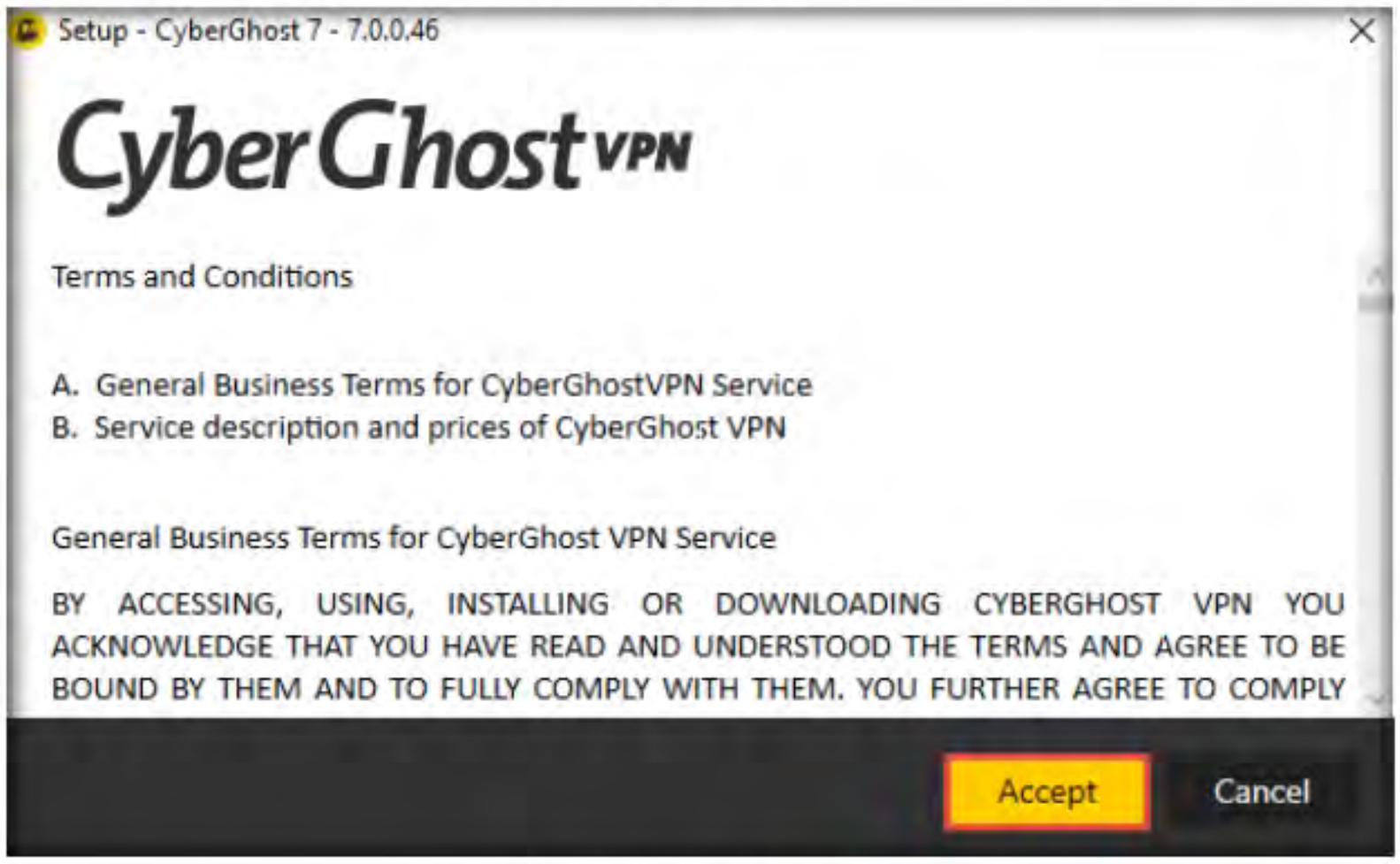


Figure 4.6.1: CyberGhost Setup window

3. Follow the installation steps to install **CyberGhost**.
4. If a **Windows Security** pop-up appears, click **Install**.

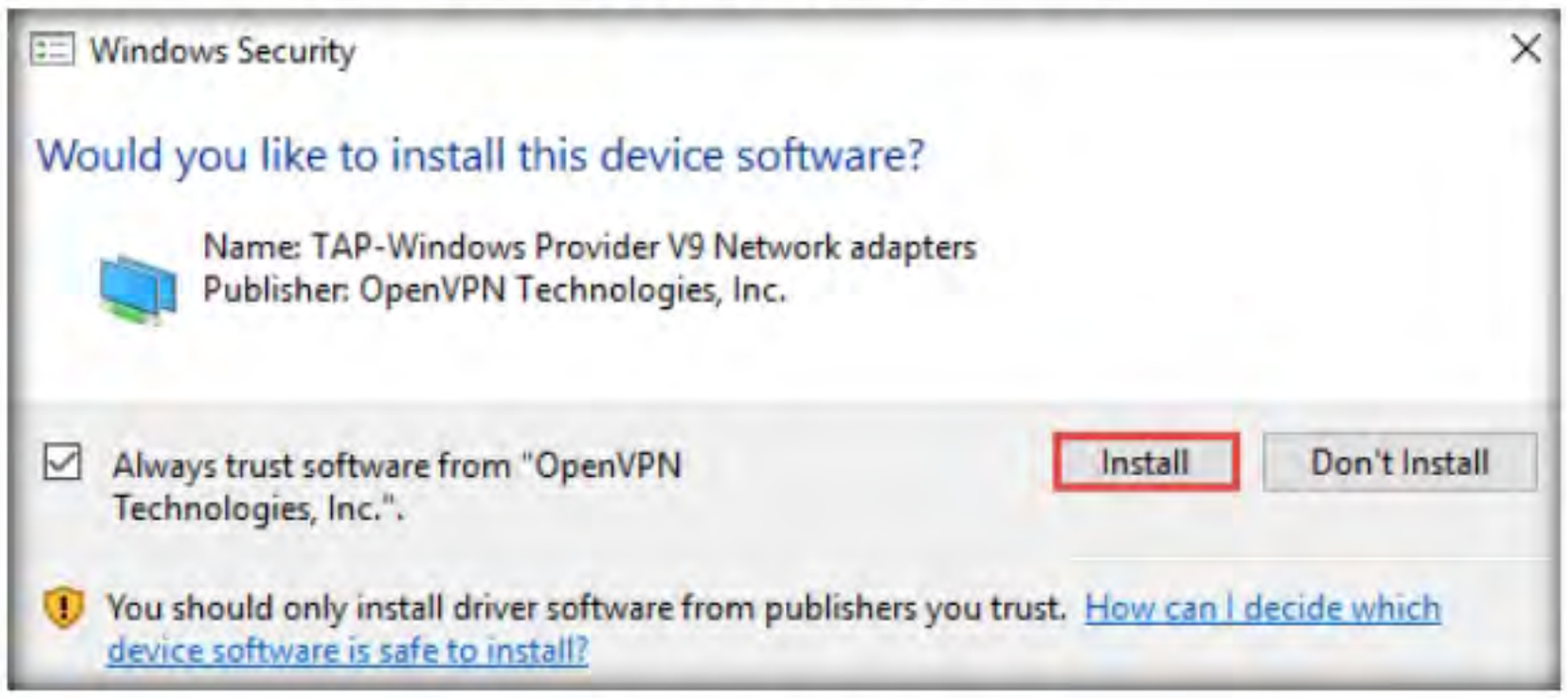


Figure 4.6.2: Windows Security pop-up

5. Once the installation is complete, the CyberGhost **Create account** window appears.

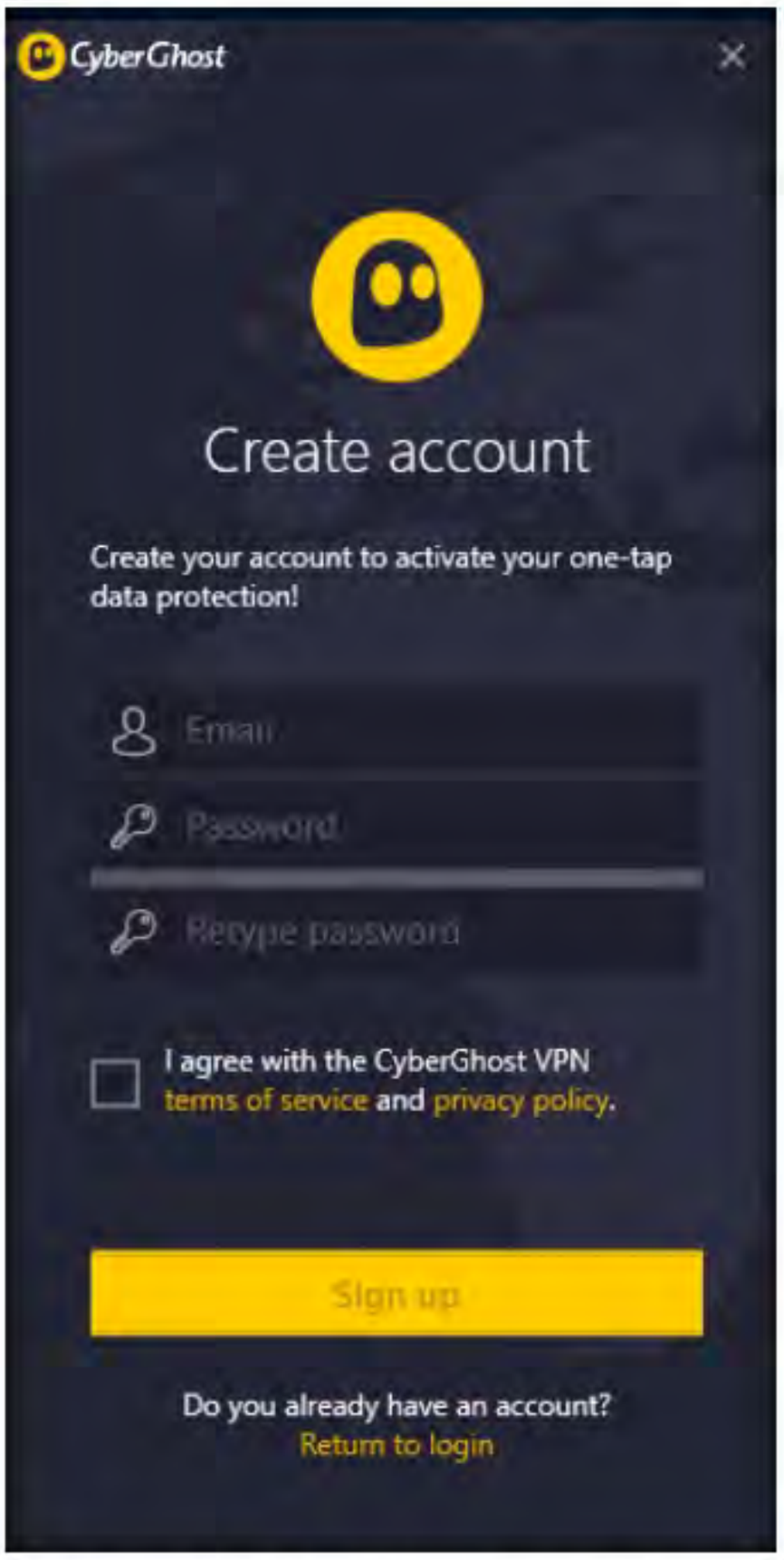


Figure 4.6.3: CyberGhost Create account window

TASK 6.2

Create Account

6. Create an account using your personal details and click on **Sign Up**.

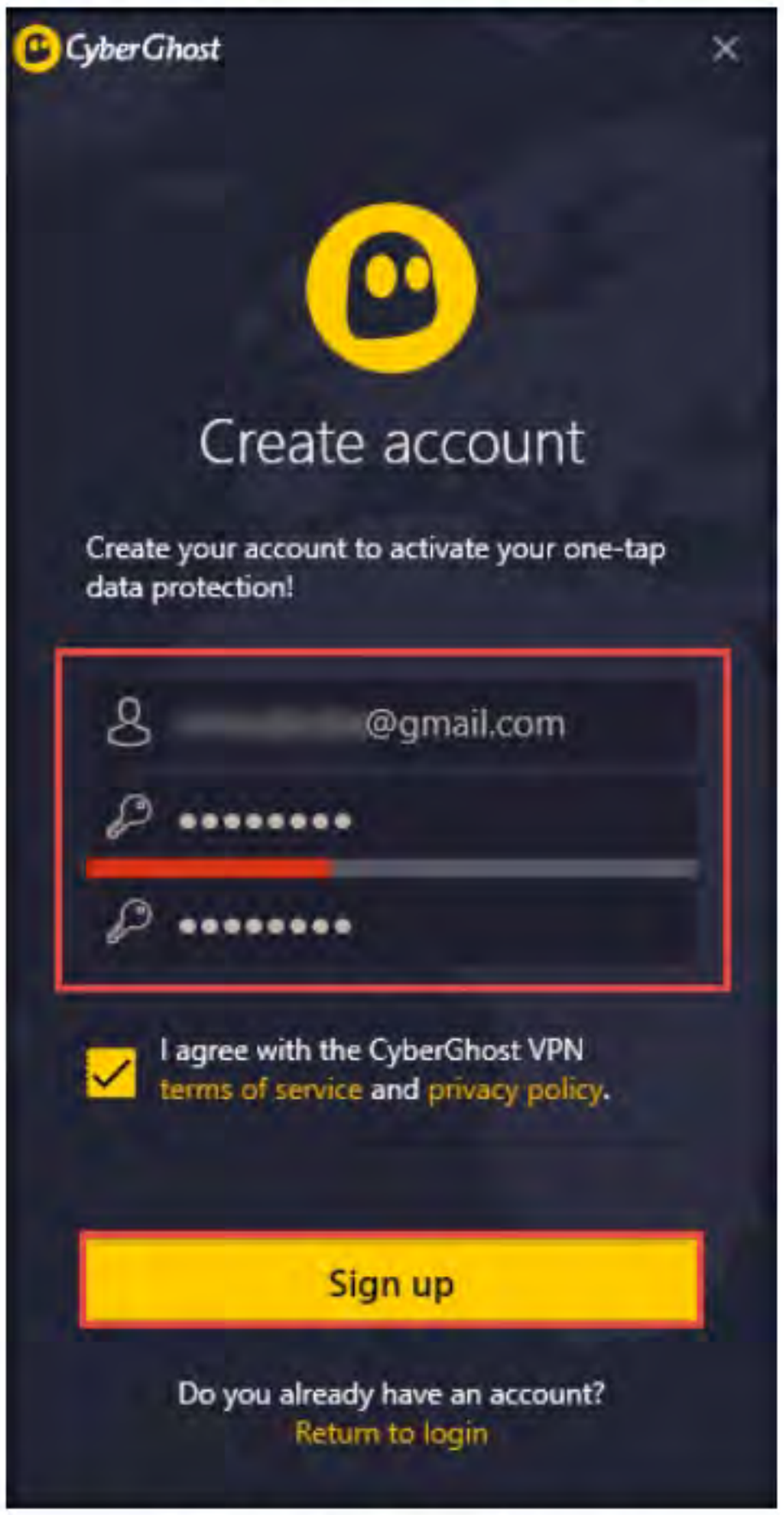


Figure 4.6.4: CyberGhost creating account

7. You will receive an activation email on your personal email. Open the email and click on **Activate Trial** to start your trial version of CyberGhost.

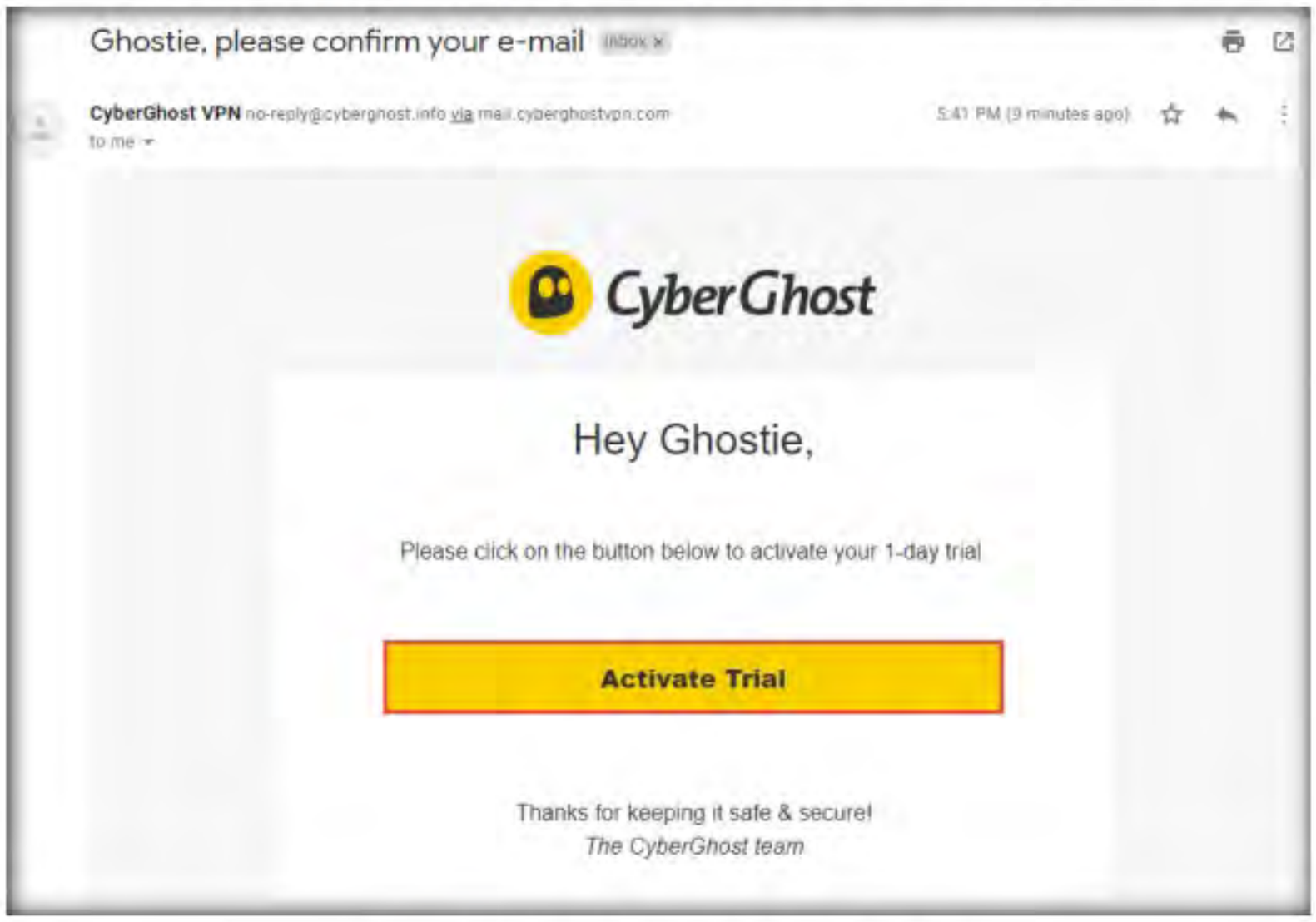


Figure 4.6.5: Email for Activation of a trial version

8. Now, switch to the **CyberGhost** login page and log in with the email address and password provided earlier; click **Log in**.

Figure 4.6.6: CyberGhost Log in

9. The CyberGhost window appears, displaying **VPN not connected!**

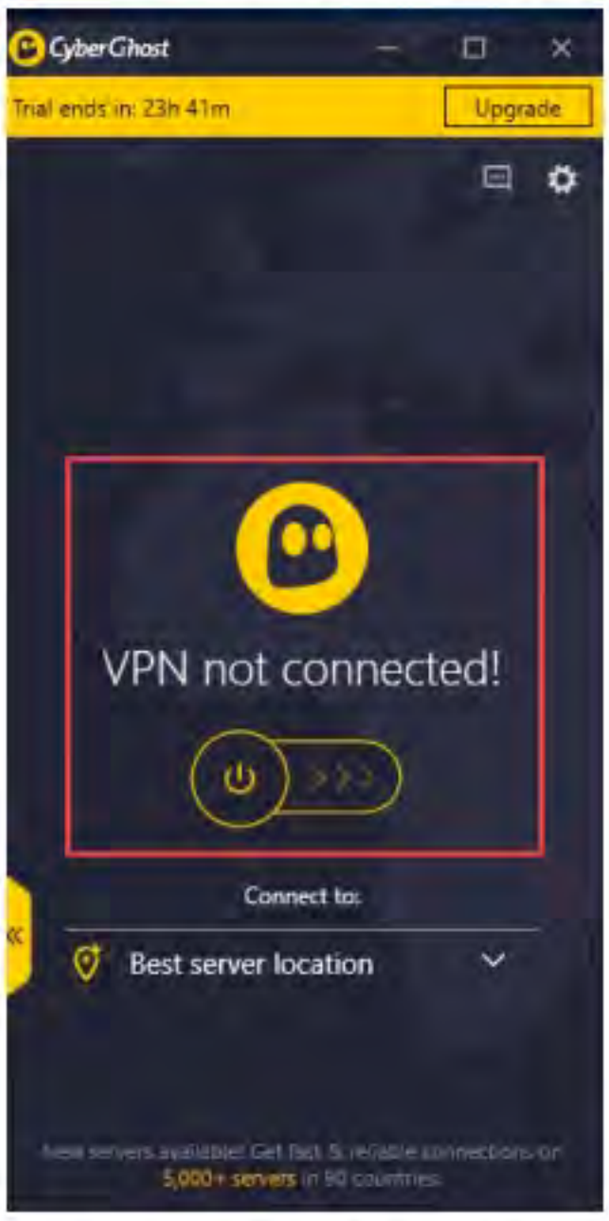


Figure 4.6.7: CyberGhost window

10. Now, click the **Settings** icon and select **Settings** from the options.

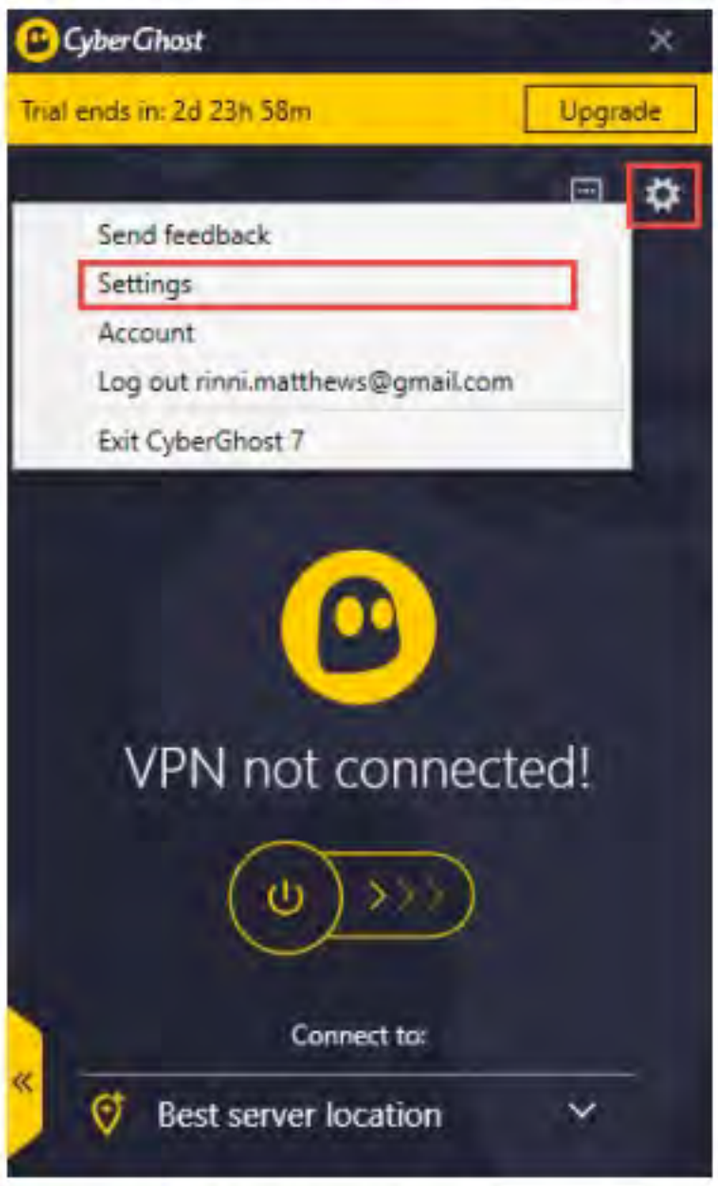


Figure 4.6.8: CyberGhost selecting the setting icon

TASK 6.3

Choose a Proxy from CyberGhost

11. The **CyberGhost Settings** window appears; click on **All server** from the left-hand pane.

Note: The list of the servers may vary in your lab environment

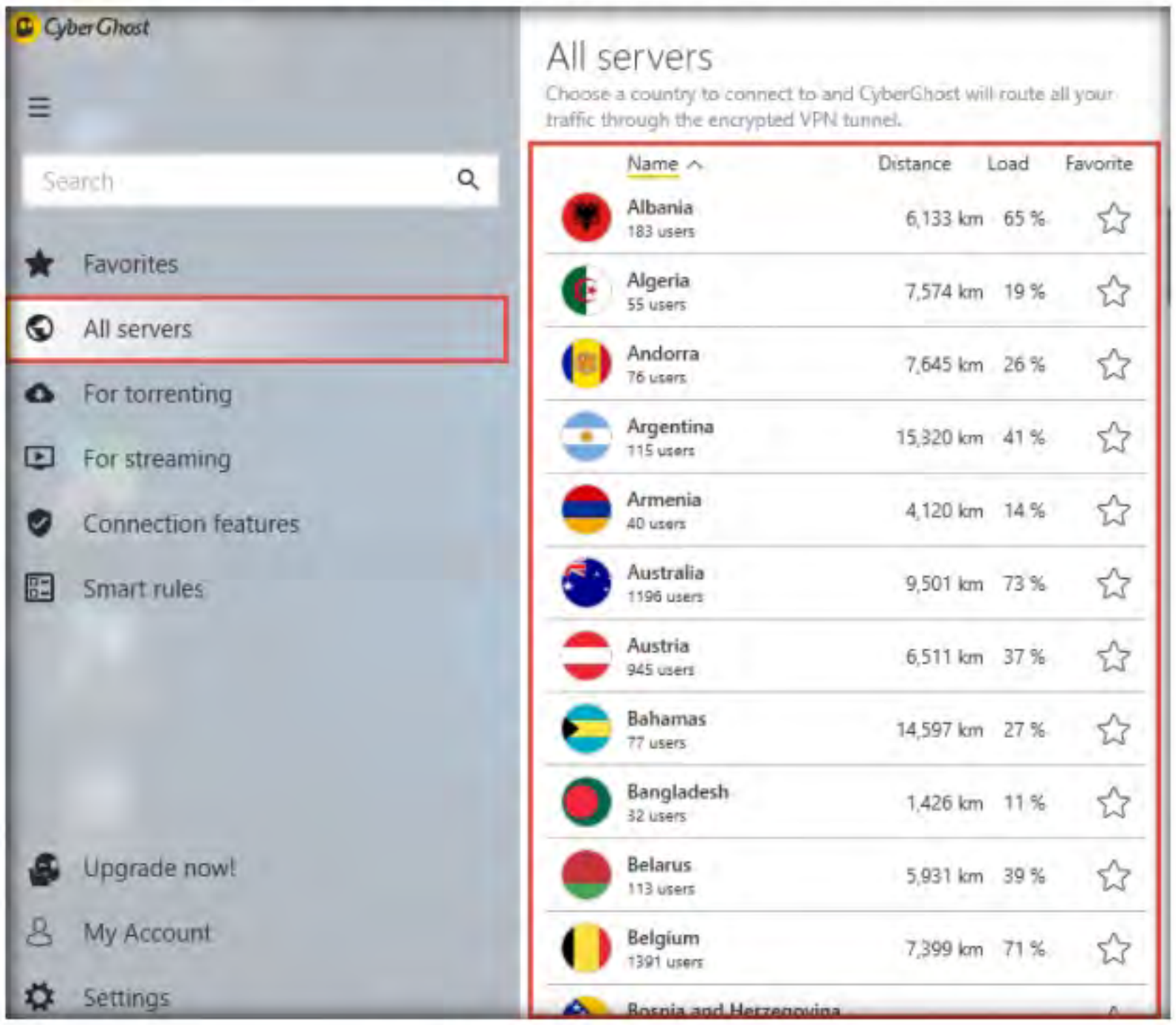


Figure 4.6.9: CyberGhost All servers window

12. Click to select any proxy server from the available options in the **All servers** section (here, **Albania**) and click on the power icon ( ) under **VPN now connected!**, as shown in the screenshot.

Note: If the CyberGhost window appears indicating that all free user slots are booked, then close the window and select another proxy server from the "all servers" list.

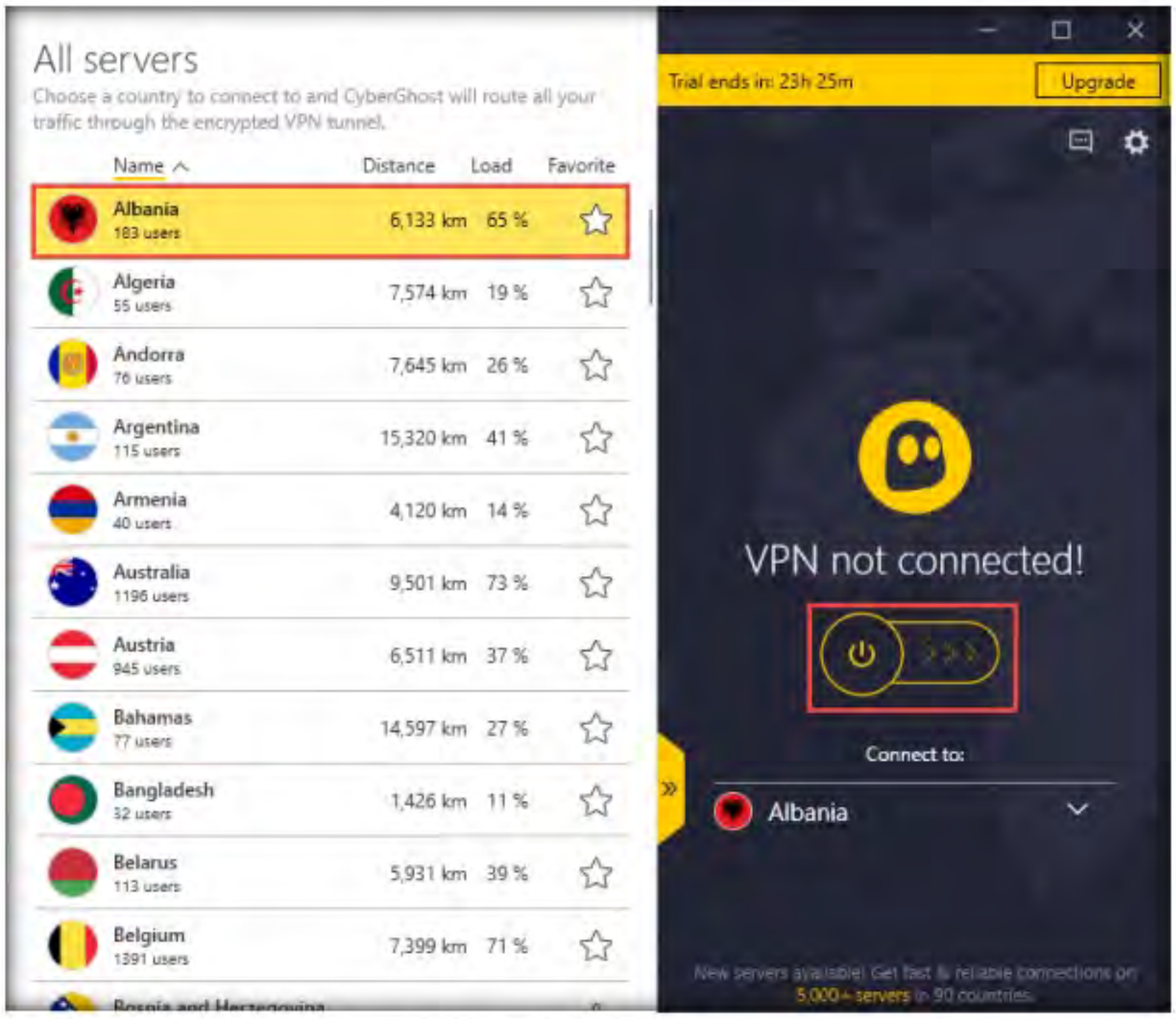


Figure 4.6.10: CyberGhost selecting proxy server

13. **CyberGhost** attempts to establish a connection to the proxy server. On successfully establishing a connection, **VPN connected!** appears.

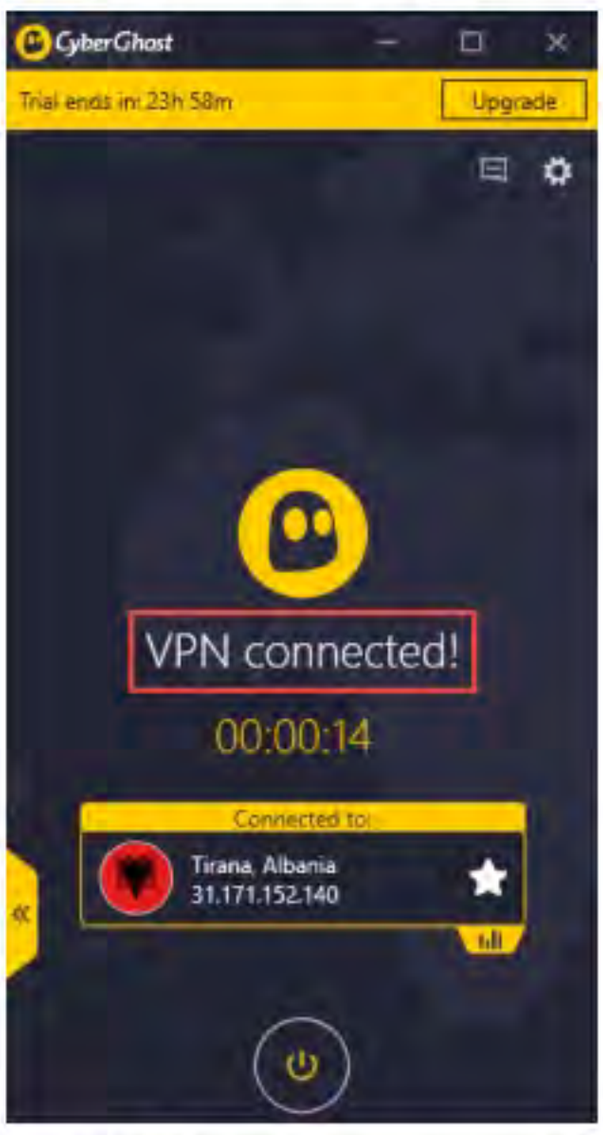


Figure 4.6.11: CyberGhost connection established

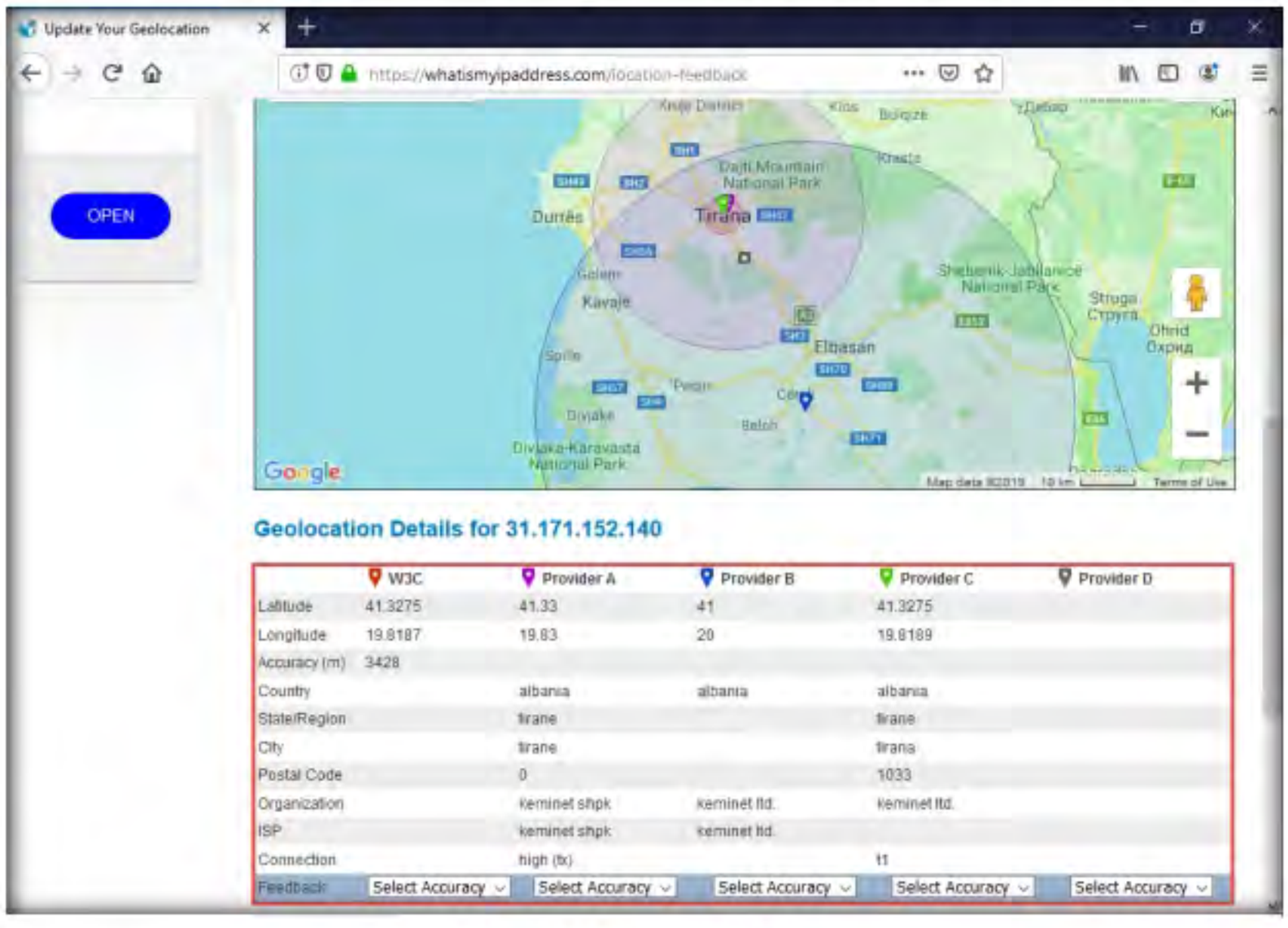
TASK 6.4

Browse Internet

14. Minimize the **CyberGhost** window and launch the **Mozilla Firefox** web browser; type the URL **https://whatismyipaddress.com/location-feedback** in the address bar and press **Enter**.

Note: If a **Will you allow whatismyipaddress.com to access your location?** pop-up appears, click **Allow Location Access**.

15. Scroll down to the **Geographical Details** section. Observe that the server IP address and location has changed to **31.171.152.140** and **Albania**.



	W3C	Provider A	Provider B	Provider C	Provider D
Latitude	41.3275	41.33	41	41.3275	
Longitude	19.8187	19.83	20	19.8189	
Accuracy (m)	3428				
Country		albania	albania	albania	
State/Region		tirane		tirane	
City		tirane		tirana	
Postal Code		0		1033	
Organization		keminet shpk	keminet ltd.	keminet ltd.	
ISP		keminet shpk	keminet ltd.		
Connection		high (fx)		t1	
Feedback	Select Accuracy	Select Accuracy	Select Accuracy	Select Accuracy	Select Accuracy

Figure 4.6.12: Testing your IP address

16. Open a new tab in the **web browser** and surf anonymously using this proxy.

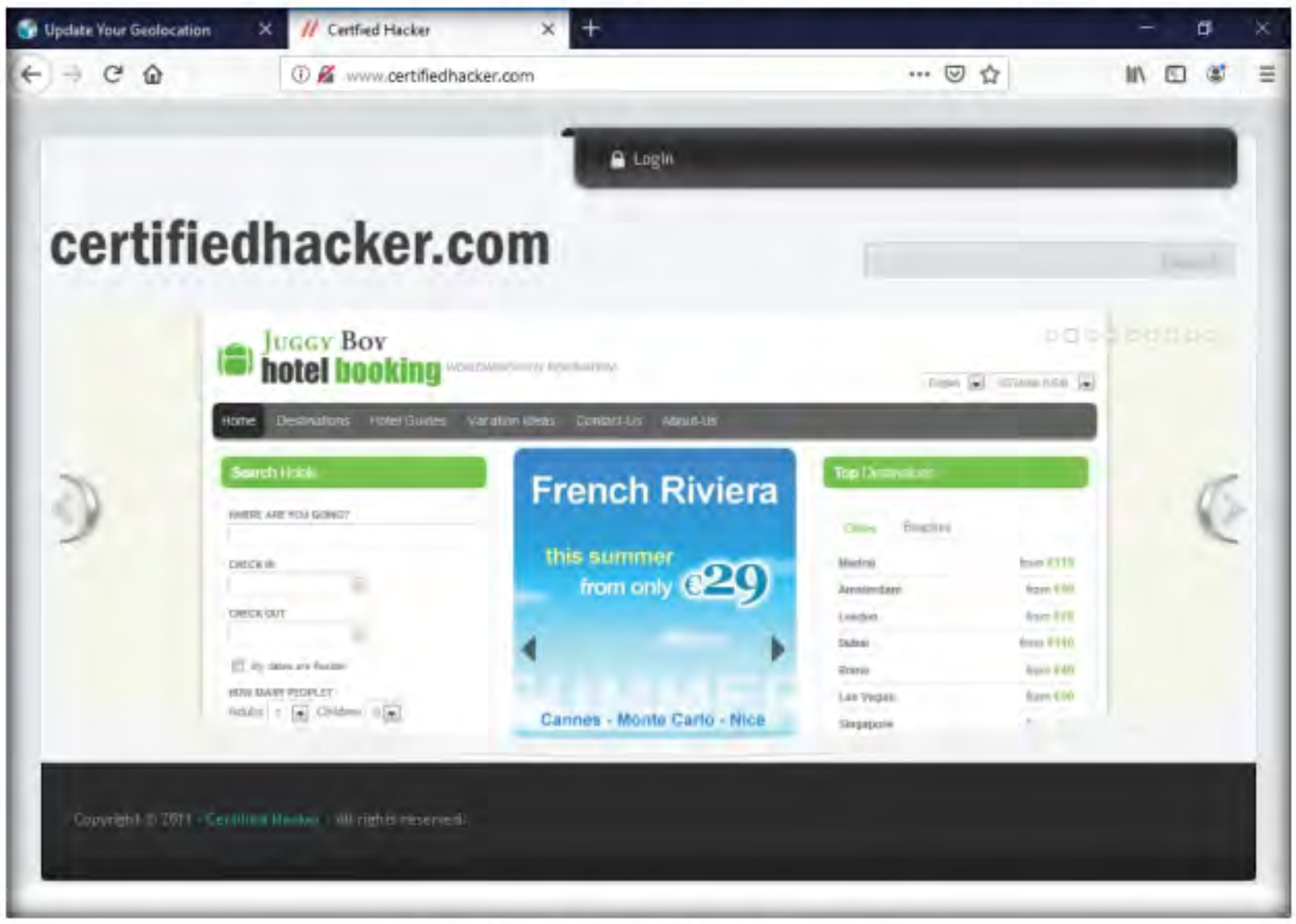


Figure 4.6.13: Surfing internet using Proxy server

17. Once you are done browsing, in the CyberGhost window, click the **Power** icon to disconnect the proxy, as shown in the screenshot.

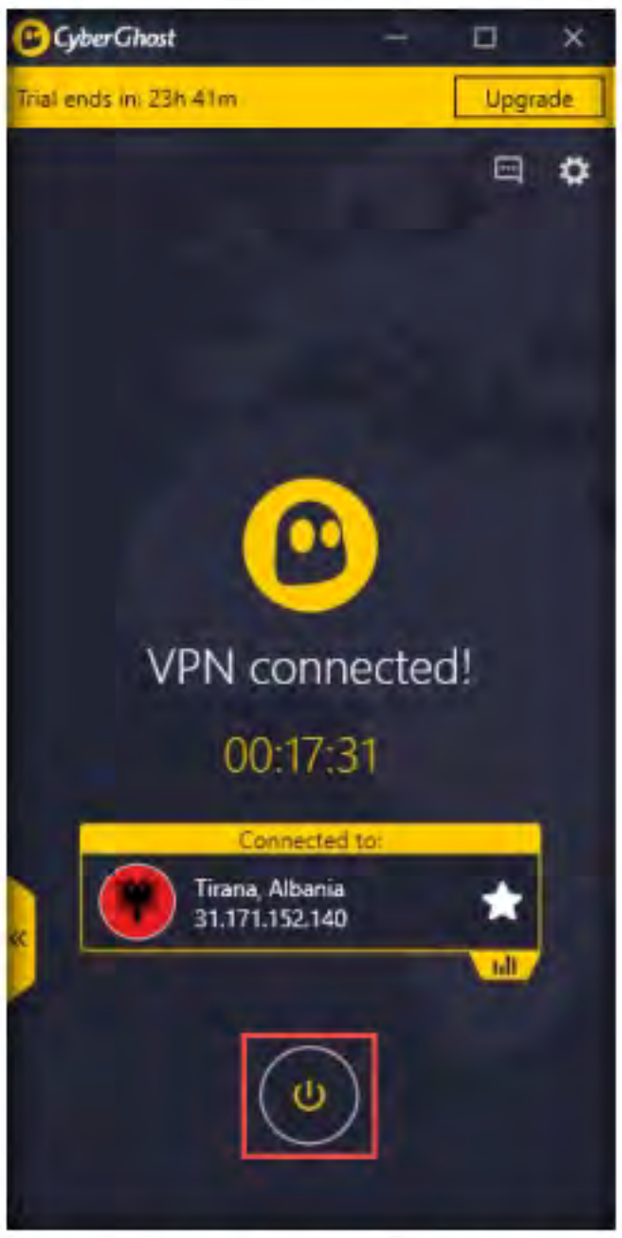


Figure 4.6.14: Turning Off the Proxy

18. This concludes the demonstration of anonymously surfing the Internet using CyberGhost.
19. Close all open windows and document all the acquired information.
20. Navigate to **Control Panel → Programs → Programs and Features** and uninstall the **CyberGhost VPN** application
21. Turn off the **Windows 10** virtual machine.

You can also use other proxy tools such as **Burp Suite** (https://www.portswigger.net), **Tor** (https://www.torproject.org), **CCProxy** (https://www.youngzsoft.net), and **Hotspot Shield** (https://www.hotspotshield.com) to browse the Internet anonymously.

Lab Analysis

Analyze and document all the results discovered in the lab exercise.

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS ABOUT THIS LAB.

Internet Connection Required	
☑ Yes	☐ No
Platform Supported	
☑ Classroom	☑ iLabs

Lab 5

Draw Network Diagrams

A network diagram helps to analyze complete network topology.

ICON KEY

- Valuable information
- Test your knowledge
- Web exercise
- Workbook review

Lab Scenario

Until now, you have gathered information about the open ports, services running on the ports, OS details, security mechanisms details, etc. of the target network using various port and network scanning techniques and tools.

As a professional ethical hacker or a pen tester, the last step in the penetration process is to draw a network diagram that assists in identifying the topology or architecture of a target network. The network diagram also helps to trace the path to the target host in the network and enables you to understand the position of firewalls, IDSs, routers, and other access control devices.

As a professional ethical hacker or pen tester, you should be able to create a pictorial representation of network topology used in the target network. The network diagrams can be used to launch further attacks on the target network.

Lab Objectives

- Draw network diagrams using Network Topology Mapper

Lab Environment

To carry out this lab, you need:

- Windows 10 virtual machine
- Windows Server 2016 virtual machine
- Parrot Security virtual machine
- Ubuntu virtual machine
- Web browsers with an Internet connection
- Administrator privileges to run the tools
- Network Topology Mapper located at **E:\CEH-Tools\CEHv11 Module 03 Scanning Networks\Network Discovery Tools\Network Topology Mapper**
- You can also download the latest version of the above-mentioned tool from its official website. If you decide to download the latest version, the screenshots shown in the lab might differ.

Tools demonstrated in this lab are available in E:\CEH-Tools\CEHv11 Module 03 Scanning Networks

Lab Duration

Time: 10 Minutes

Overview of Network Diagrams

Drawing a network diagram assists in the identification of the topology or architecture of a target network, and further assists you in finding the vulnerabilities or weak points of security mechanisms. These vulnerabilities can then be exploited to bypass the target's network. The network diagram also helps the network administrators to manage their networks.

Lab Tasks

Draw Network Diagrams using Network Topology Mapper

Here, we will use Network Topology Mapper to draw network diagrams of the target network.

TASK 1.1

Install Network Topology Mapper

1. Turn on the **Windows 10**, **Windows Server 2016**, **Parrot Security**, and **Ubuntu** virtual machines
2. In the **Windows 10** virtual machine, navigate to **E:\CEH-Tools\CEHv11 Module 03 Scanning Networks\Network Discovery Tools\Network Topology Mapper**, and then double-click **SolarWinds Network Topology Mapper.exe**.
3. The **SolarWinds Registration** dialog-box opens. Enter a working email address, and then click **Continue**.

Network Topology Mapper discovers a network and produces a comprehensive network diagram that integrates OSI Layer 2 and Layer 3 topology data. It automatically detects new devices and changes to network topology, simplifies inventory management for hardware and software assets, and addresses reporting needs for PCI compliance and other regulatory requirements.

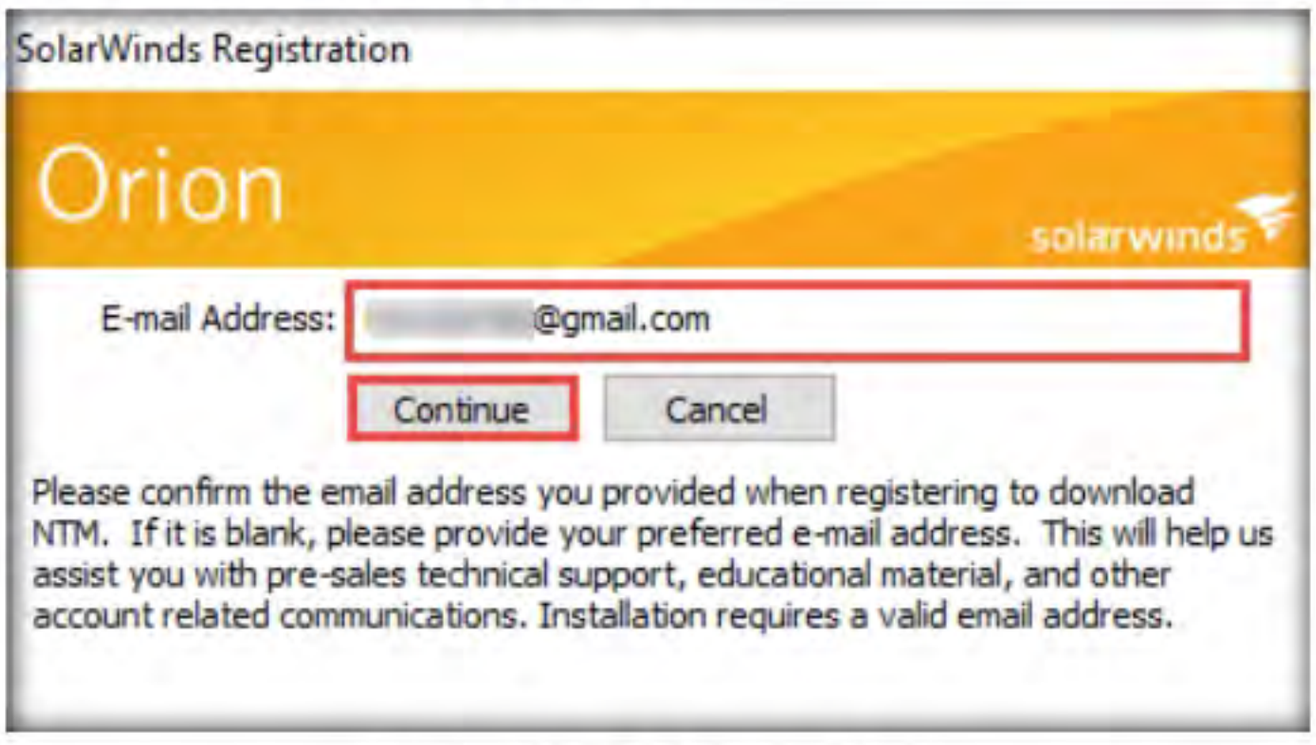


Figure 5.1.1: SolarWinds Registration dialog-box

4. In the next window, accept the license agreement and click **Install**.

Note: If a **User Account Control** window appears, click **Yes**.

5. The SolarWinds license pop-up appears; click **Continue Evaluation**.

Figure 5.1.2: Solarwinds license pop-up

6. The **Help SolarWinds Improve** window appears. Click the **No, I would not like to participate** radio button, and then click **OK**.

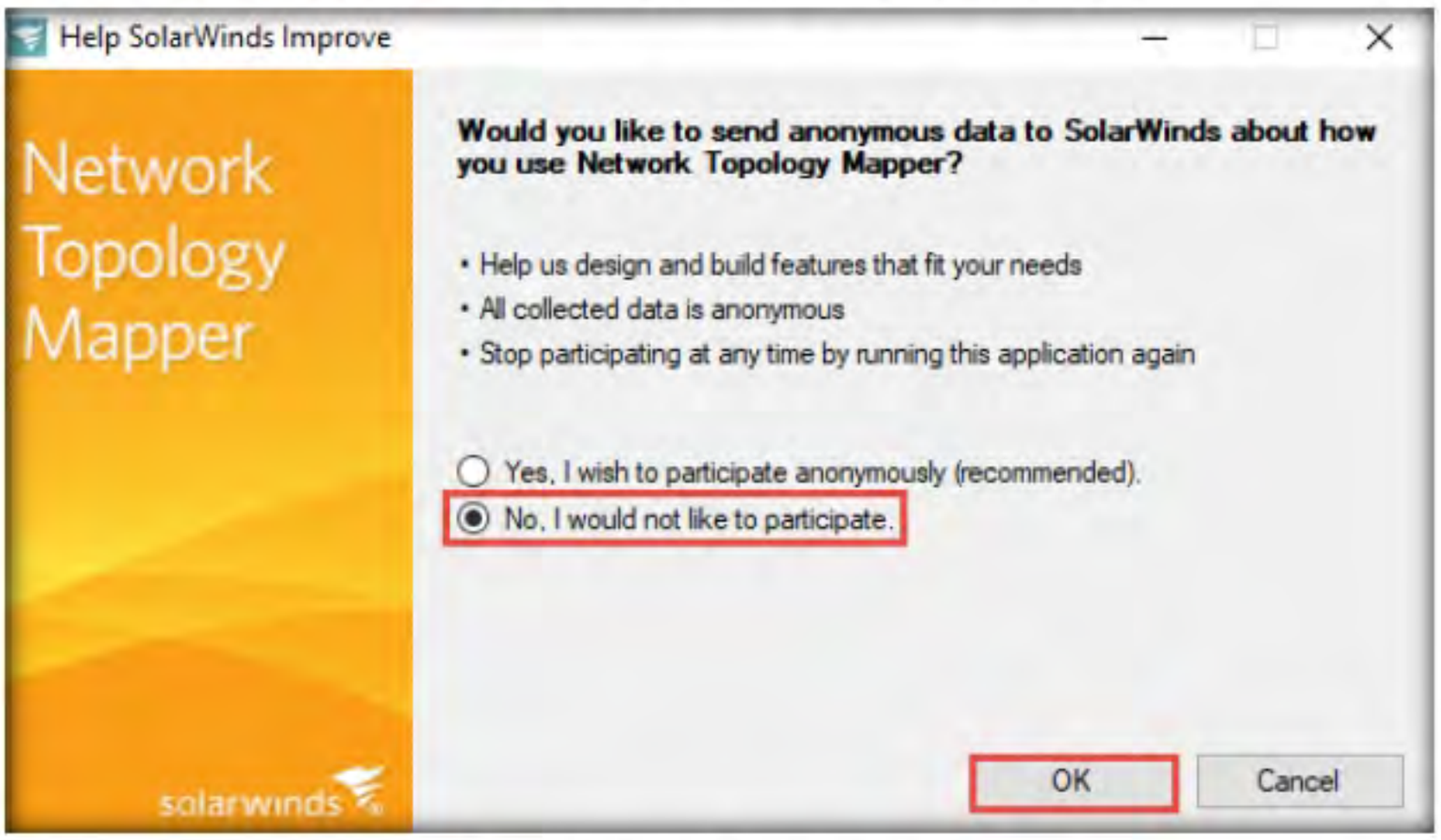


Figure 5.1.3: Help SolarWinds Improve window

7. Once the installation is complete, and the **SolarWinds Network Topology Mapper** window opens, click **Close**.

Note: Ensure that the **Run SolarWinds Network Topology Mapper now** option is selected.

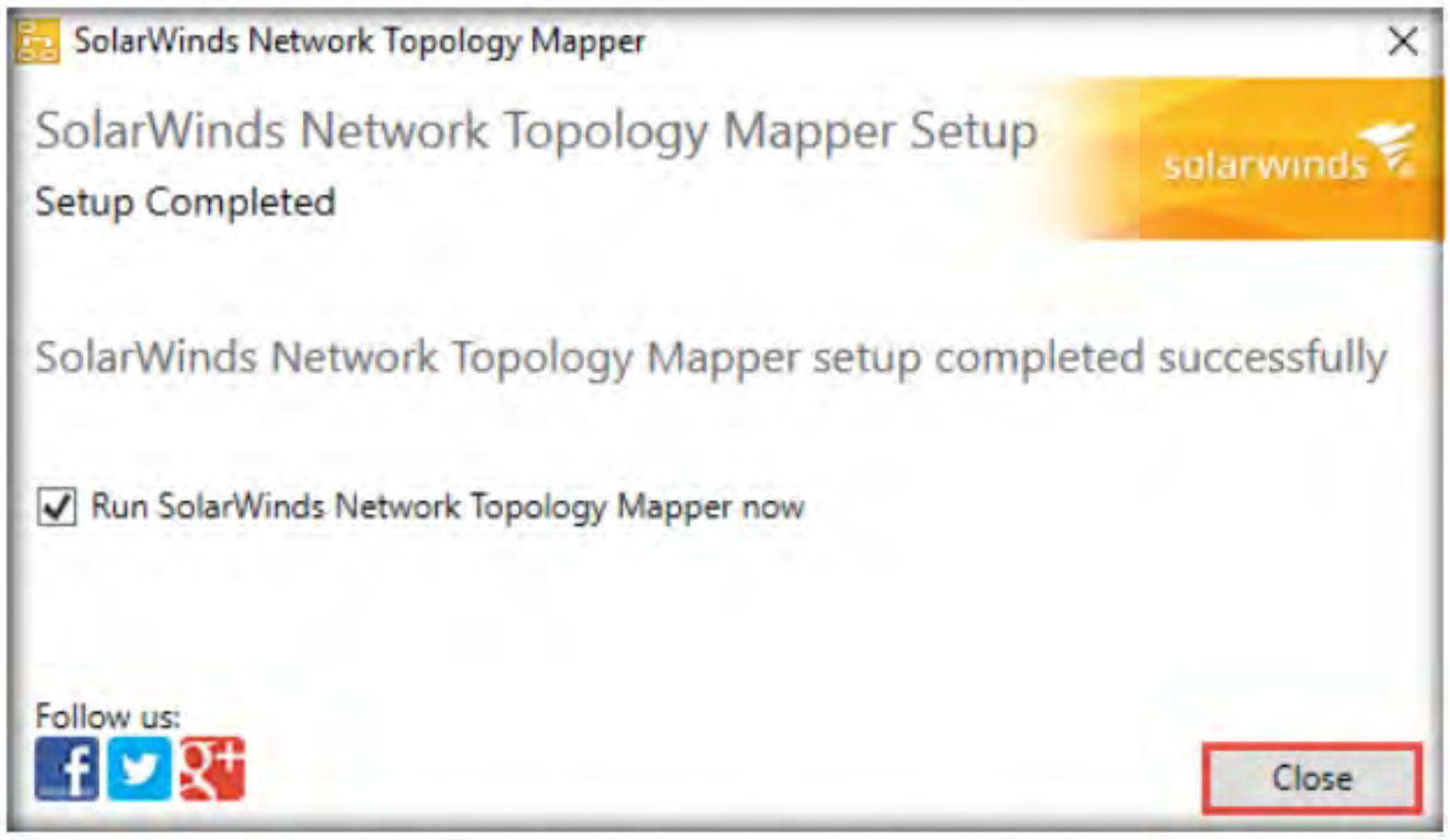


Figure 5.1.4: SolarWinds setup completed window

8. The **Solarwinds** pop-up opens; click **Continue Evaluation**.
9. The **SolarWinds Network Topology Mapper** main window appears, along with the **Welcome Screen...**. Click **New Scan** in the left-hand pane of the **Welcome Screen**.

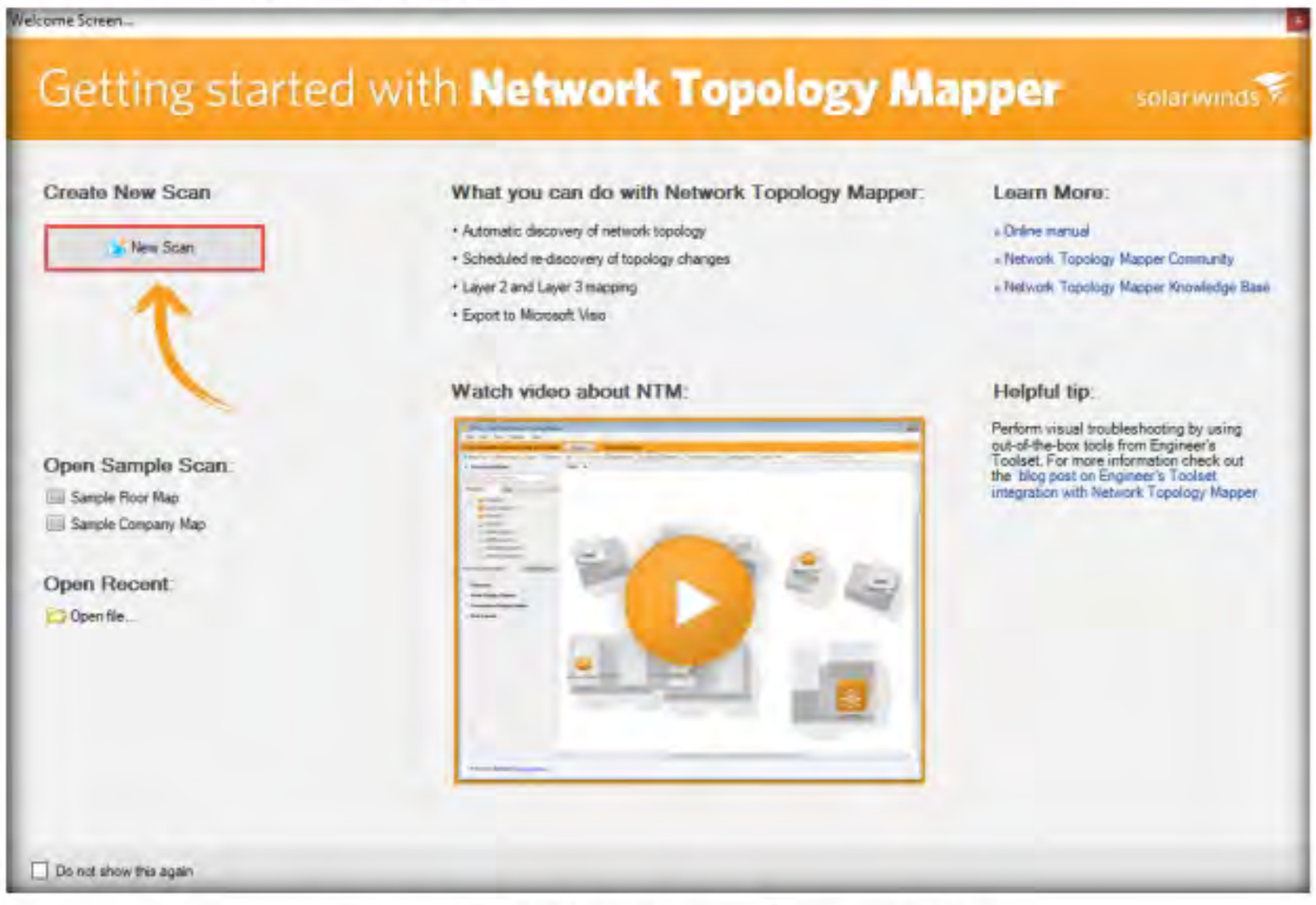


Figure 5.1.5: SolarWinds Network Topology Mapper main window

10. The **Set a Maps Password** pop-up appears. Enter a password (here **qwerty@123**) of your choice in the **New Password** field, re-enter the same password in the **Confirm Password** field, and click **Save**.

Figure 5.1.6: Set a Maps Password window

TASK 1.2

Configure Network Topology Mapper

11. The **Network Topology Scan** window appears. In the **SNMP Credentials** section, select the **private** credential under the **Stored Credentials** section and **public** credential under the **Discovery Credentials** section, and then click **Next**.

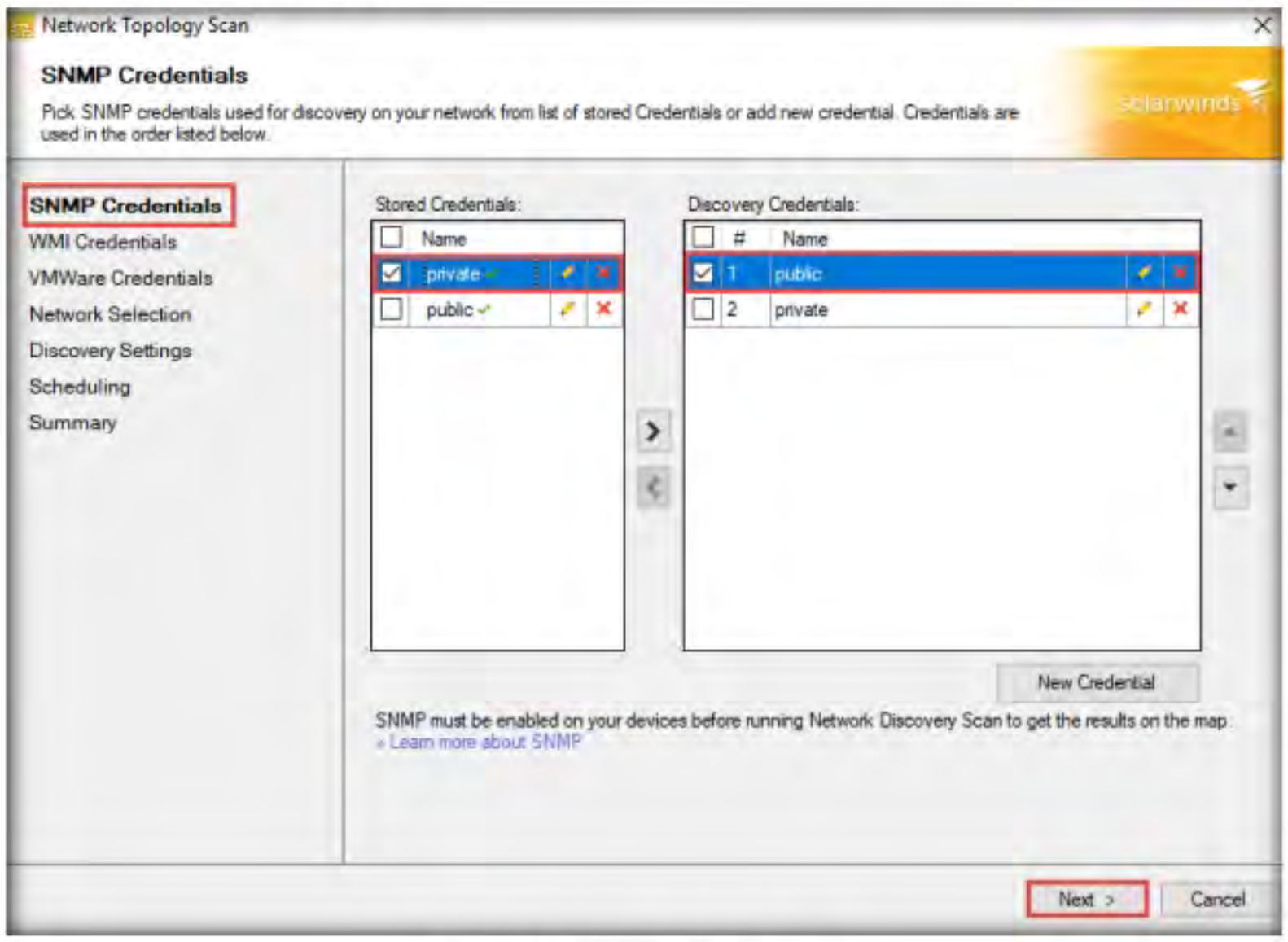


Figure 5.1.7: SNMP Credentials section

12. Leave the **WMI Credentials** and **VMWare Credentials** section to default and click **Next**.
13. The **Network Selection** section appears. Click the **IP Ranges** tab in the right-pane, enter the IP address range (**10.10.10.3 - 10.10.10.254**) in the **Start Address** and **End Address** fields, and click **Next**.

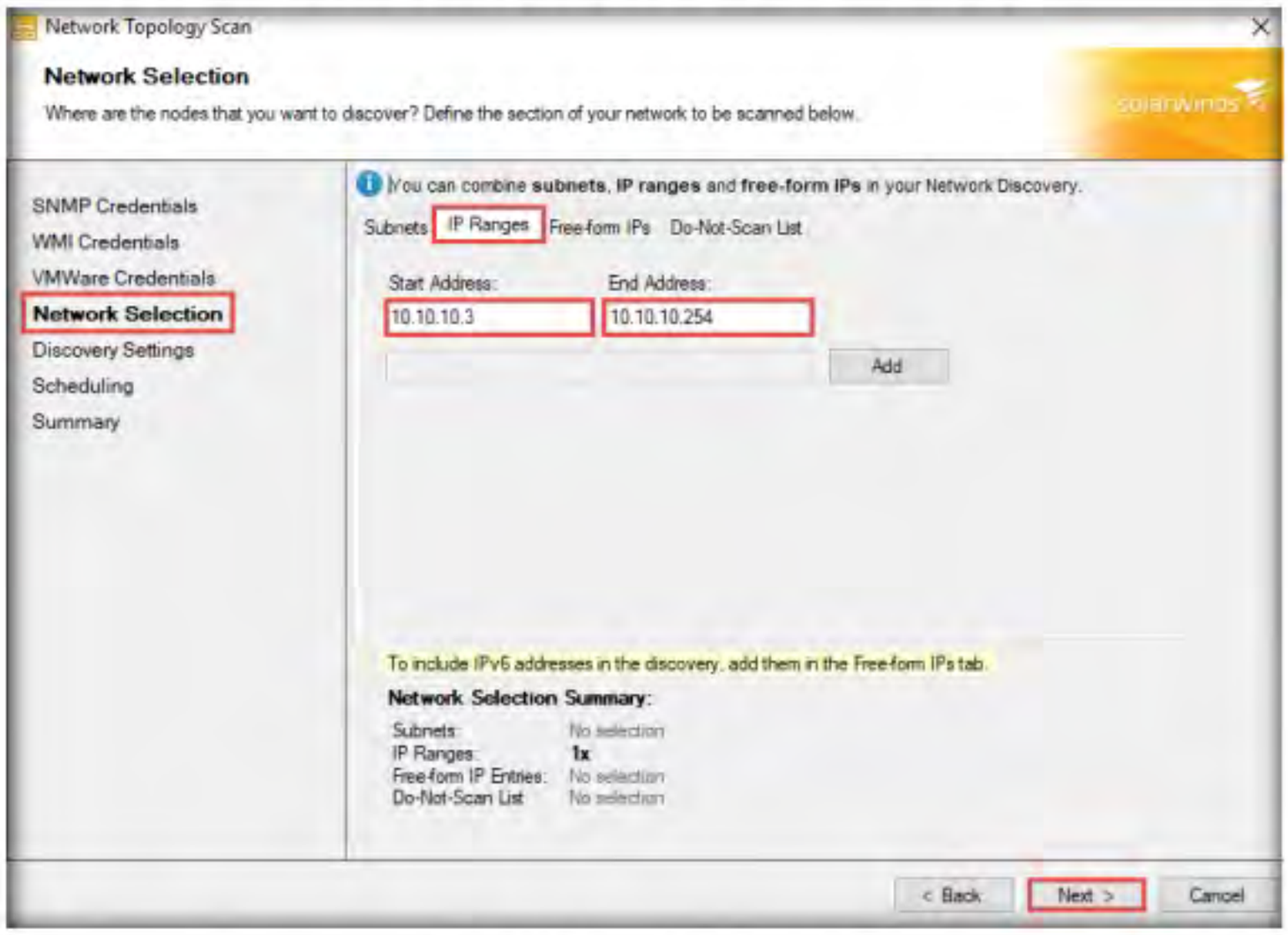


Figure 5.1.8: Network Selection section

14. The **Discovery Settings** section appears. Enter a name under the **Scan name** field (here, "**Network Topology**") and click **Next**.

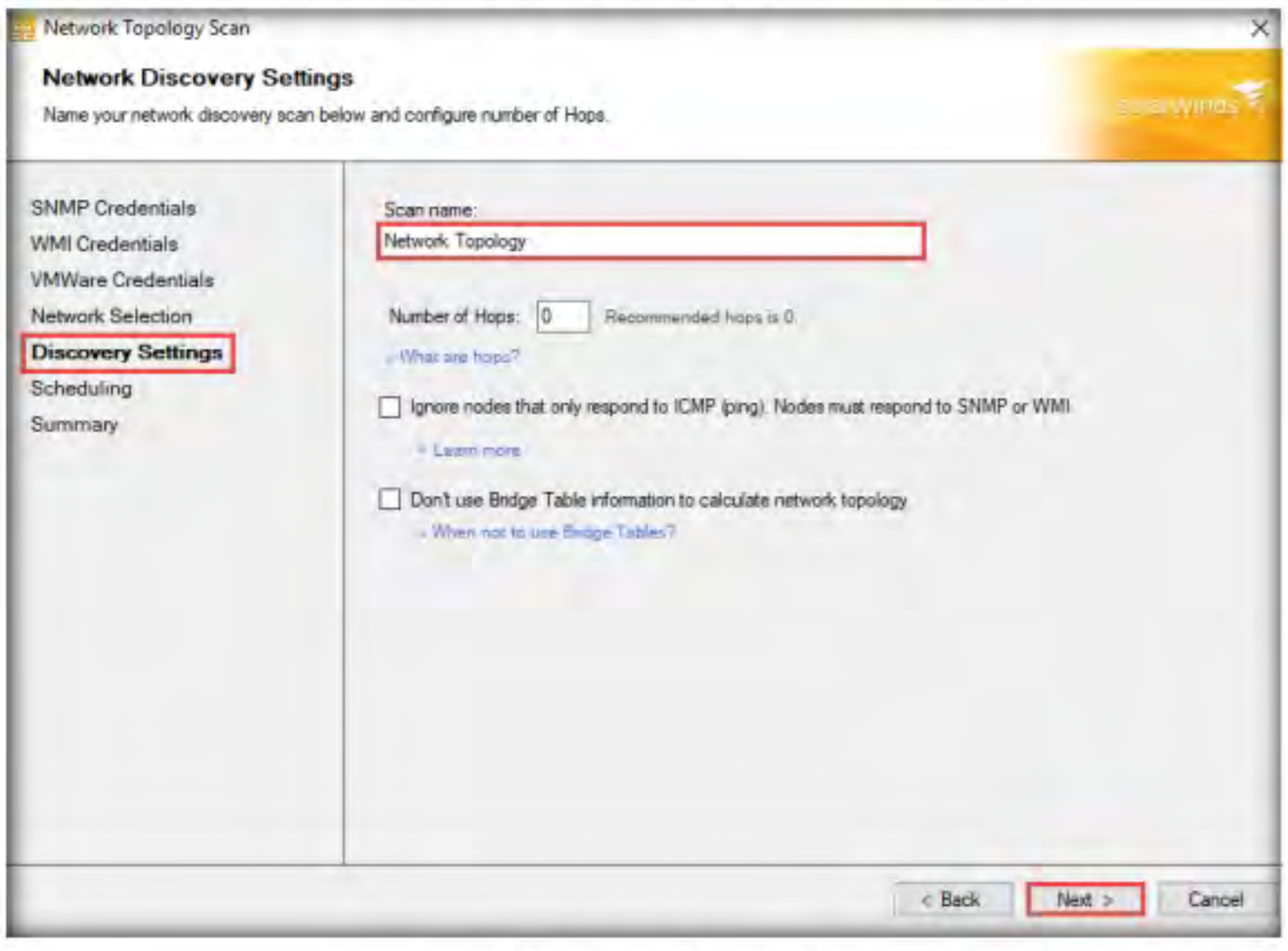


Figure 5.1.9: Discovery Settings section

15. The **Scheduling** section appears. Ensure that **Once** is selected in the **Frequency** drop-down menu; under the **Execute immediately** radio button **Yes, run this discovery now** is selected; then, click **Next**.

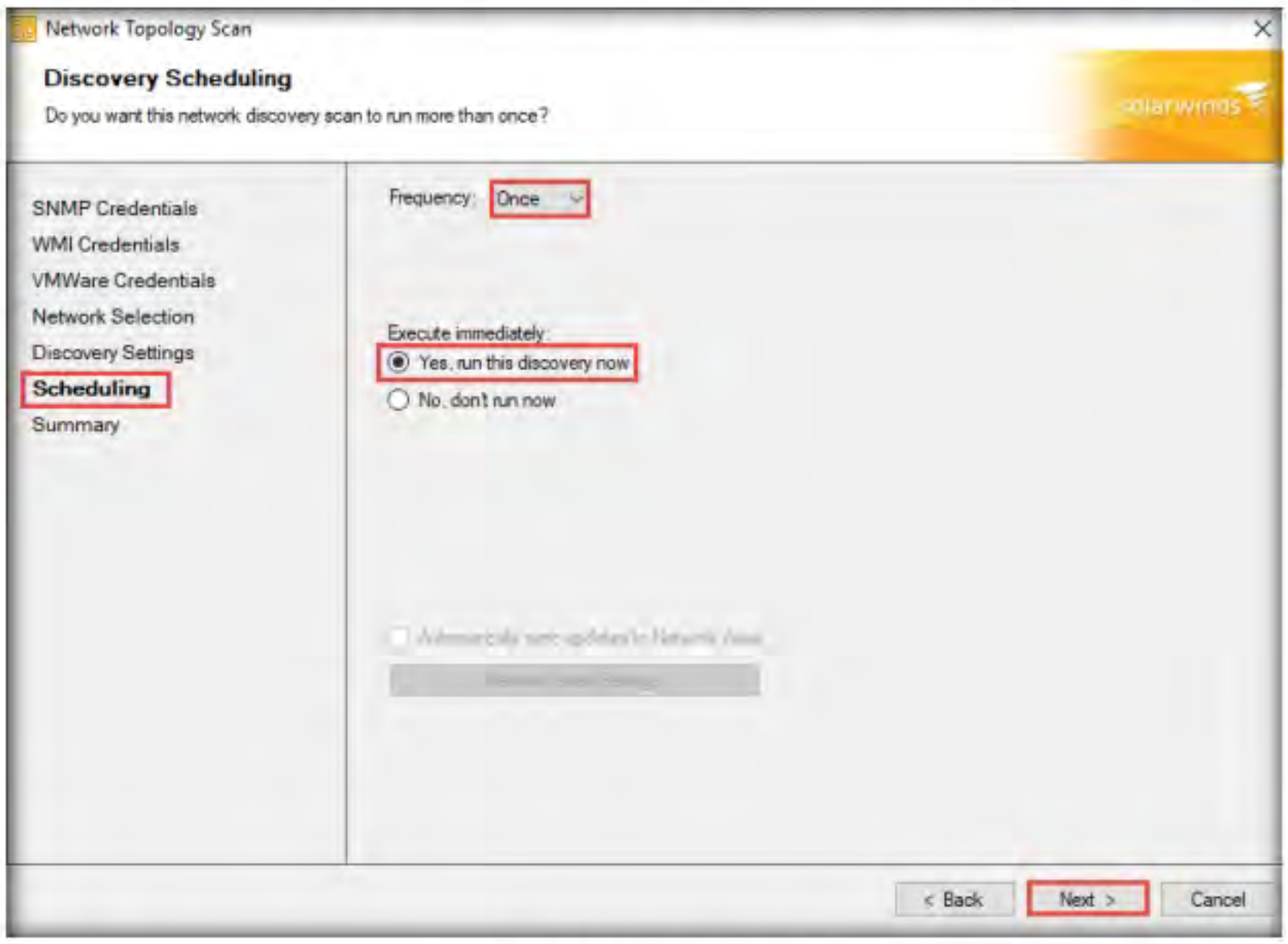


Figure 5.1.10: Scheduling section

16. The **Summary** section appears; click **Discover**.

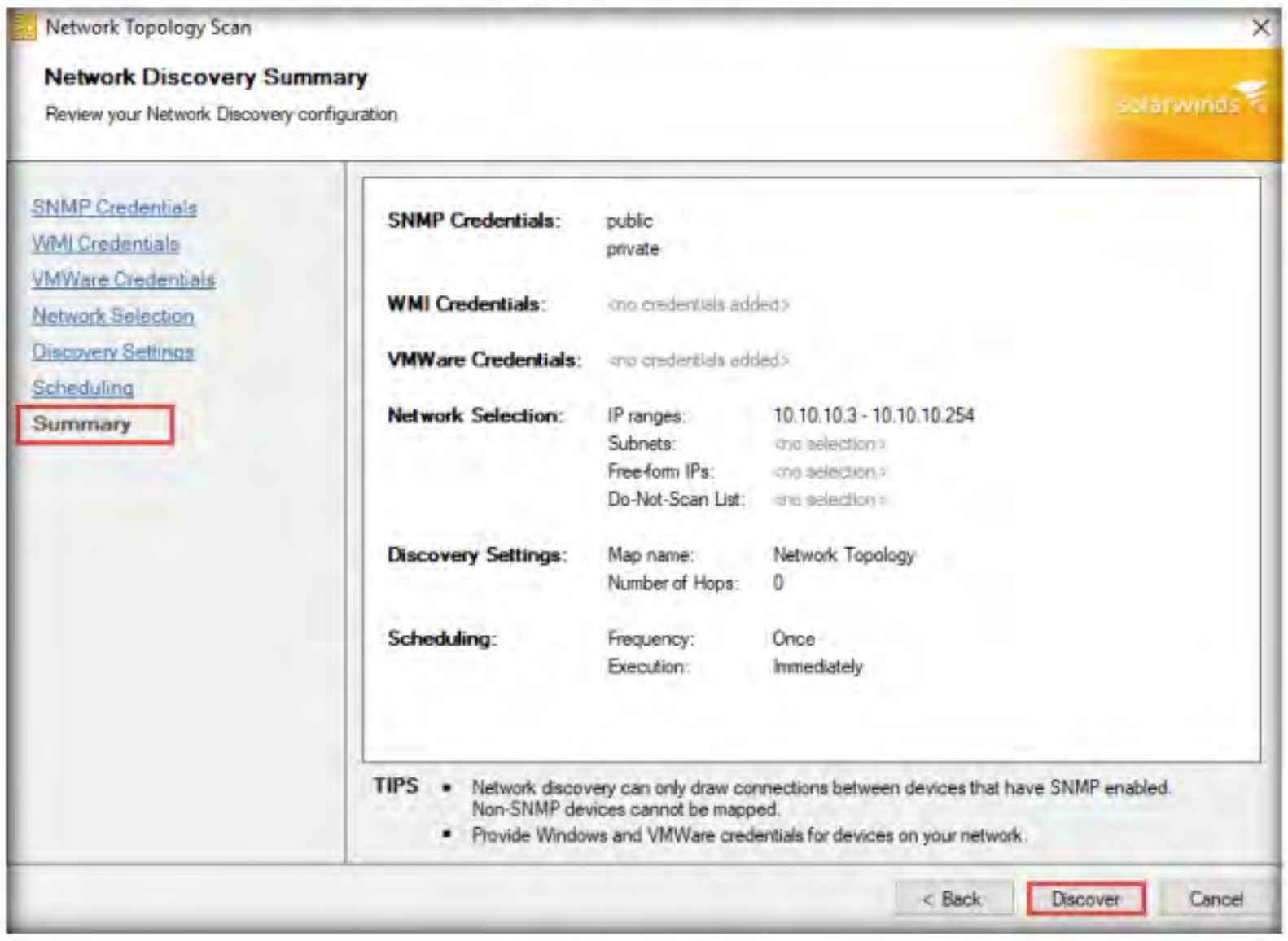


Figure 5.1.11: Summary section

17. The **New Network Scan** window appears; the Network Topology Mapper starts scanning the network for live hosts.

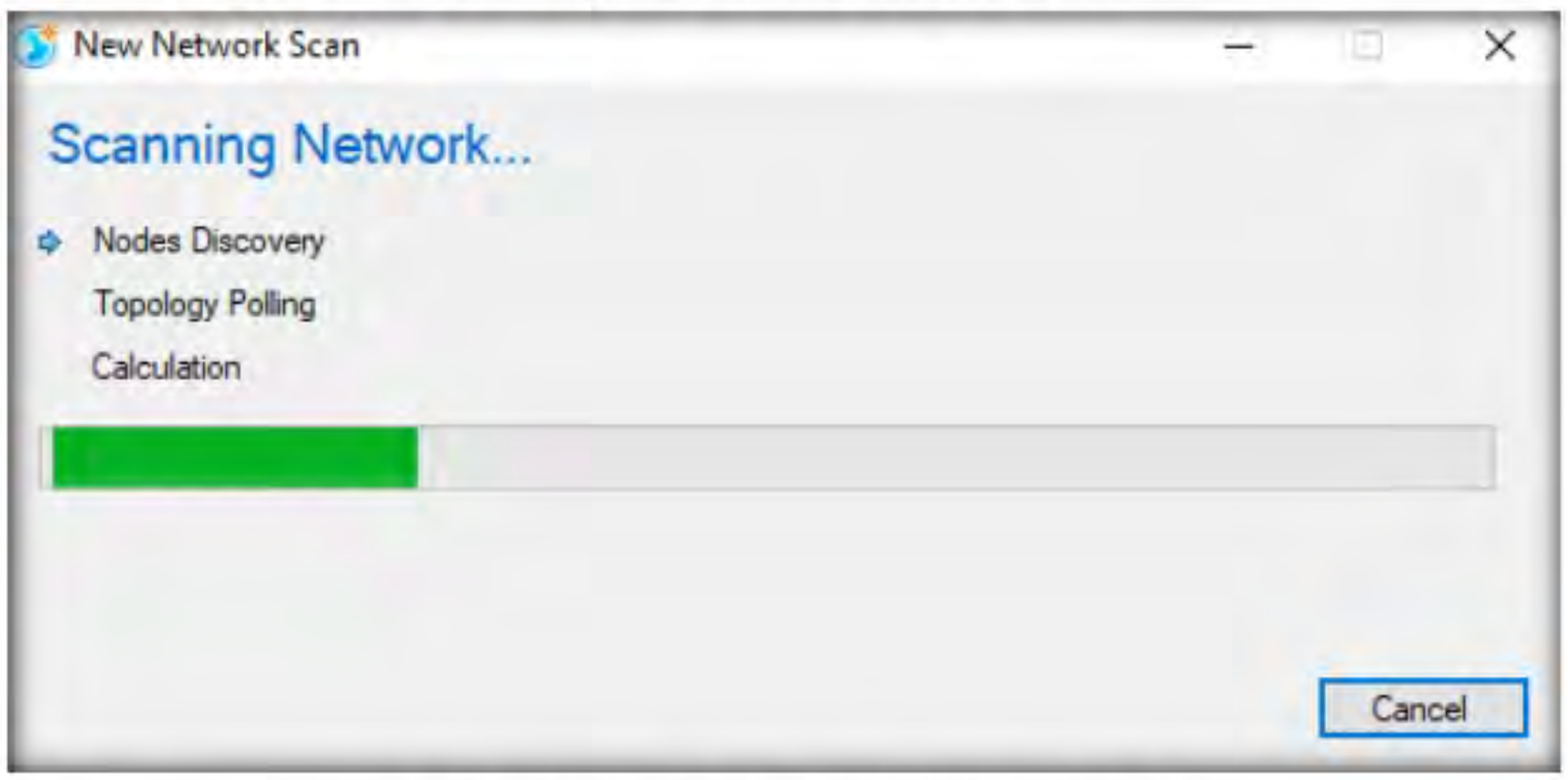


Figure 5.1.12: Network Topology Mapper scanning the network

TASK 1.3

Draw Network Diagram

18. The **Network Topology - SolarWinds Network Topology Mapper** window appears. Close the **Map Navigator** window.

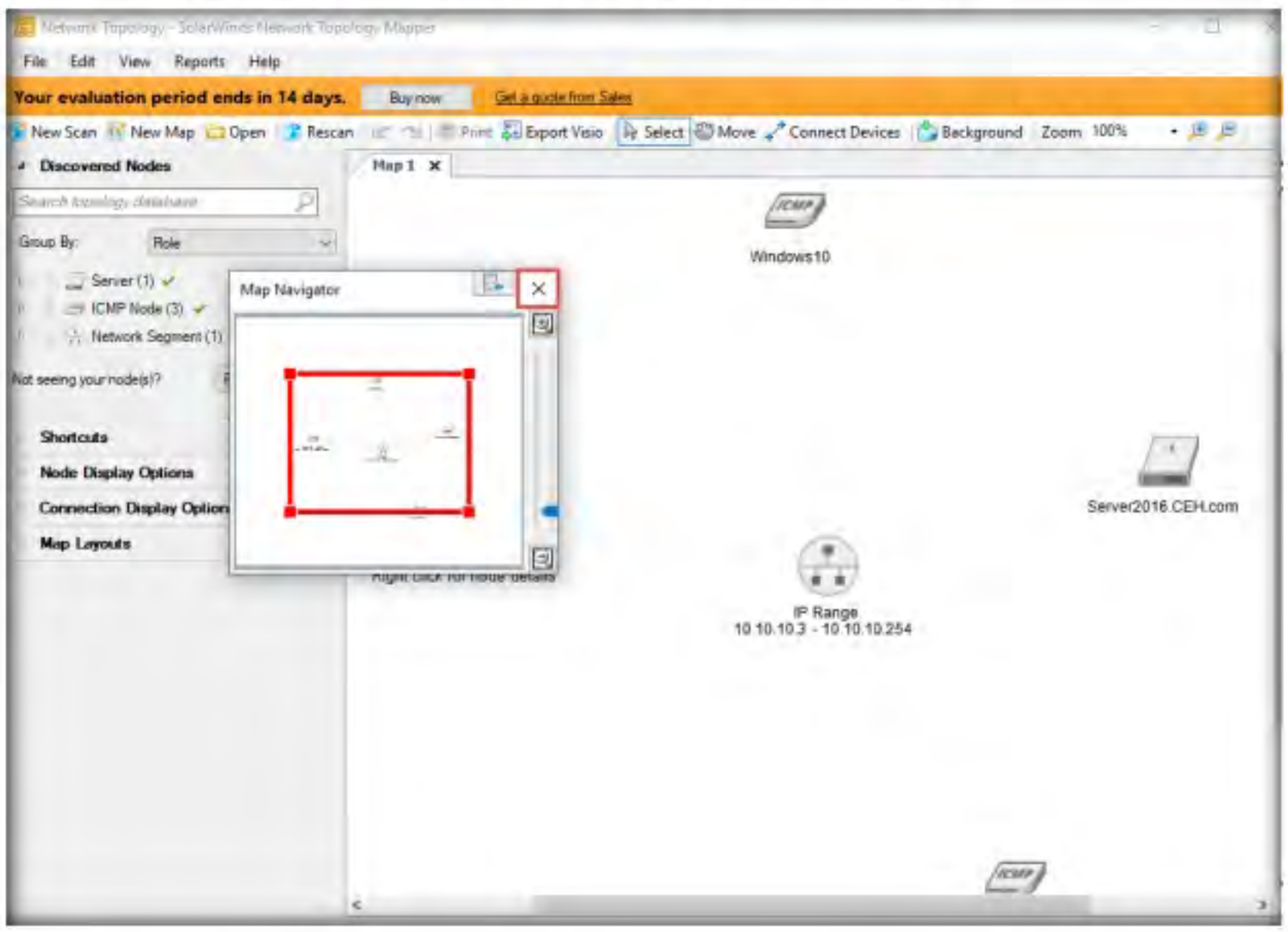


Figure 5.1.13: Network Scan results window

19. The **Network Topology Mapper** displays a network topology diagram for the provided IP address range, as shown in the following screenshot.

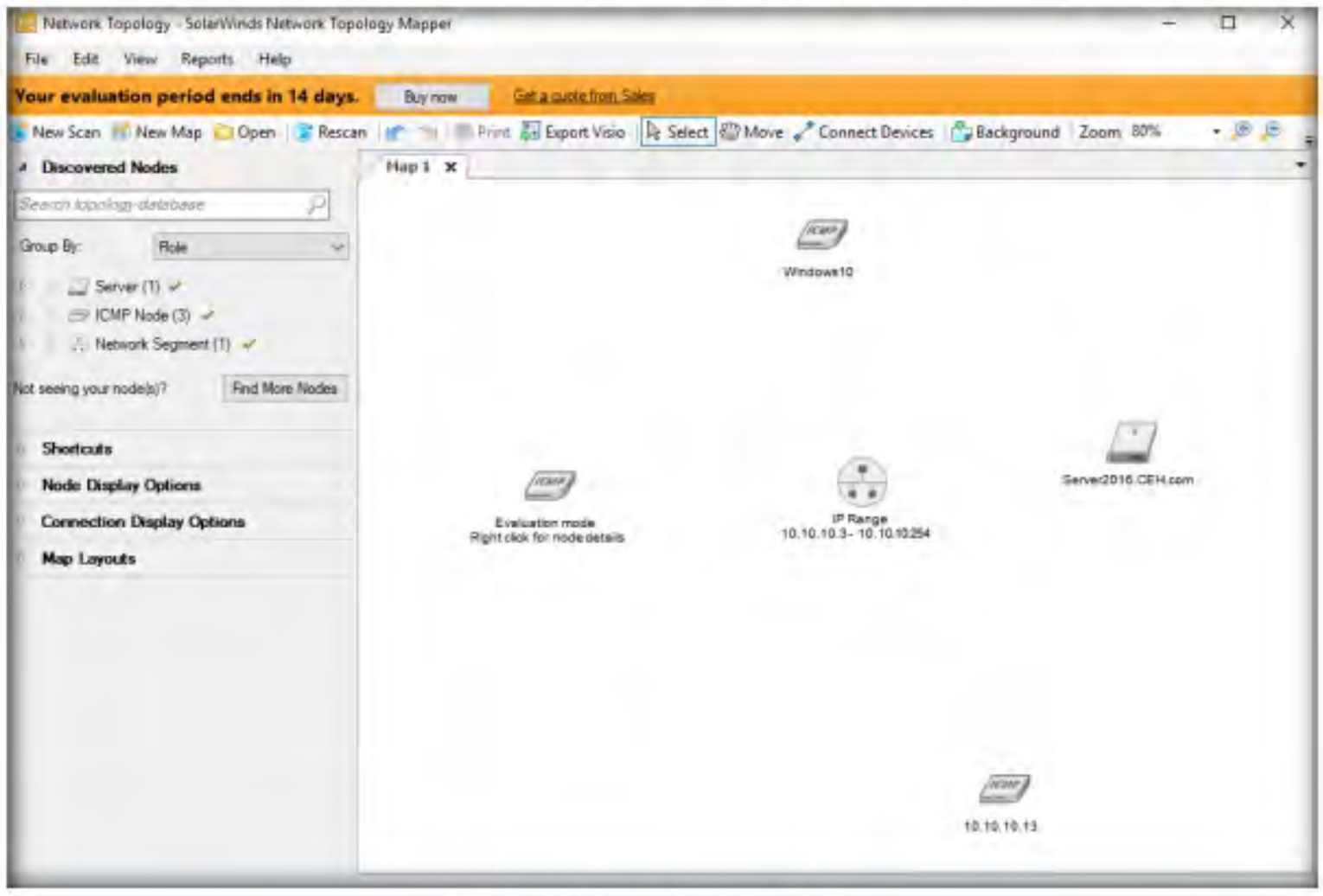


Figure 5.1.14: Network topology diagram

20. Expand **Node Display Options** in the right-hand pane and select the **IP address** checkbox. This displays IP addresses for all nodes in the layout.

Figure 5.1.15: Node Display Options: selecting the IP address

21. Now, expand the **Map Layouts** node, and select **Symmetrical** under the **Auto Arrange** section to change the topology layout of the mapped network. Each time you click **Symmetrical**, all nodes are rearranged randomly.

Note: You may select the node display options of your choice: whichever options you choose, they are added to the topology map. These topology maps are saved automatically to **C:\ProgramData\Solarwinds\Network Topology Mapper\UserMaps**.

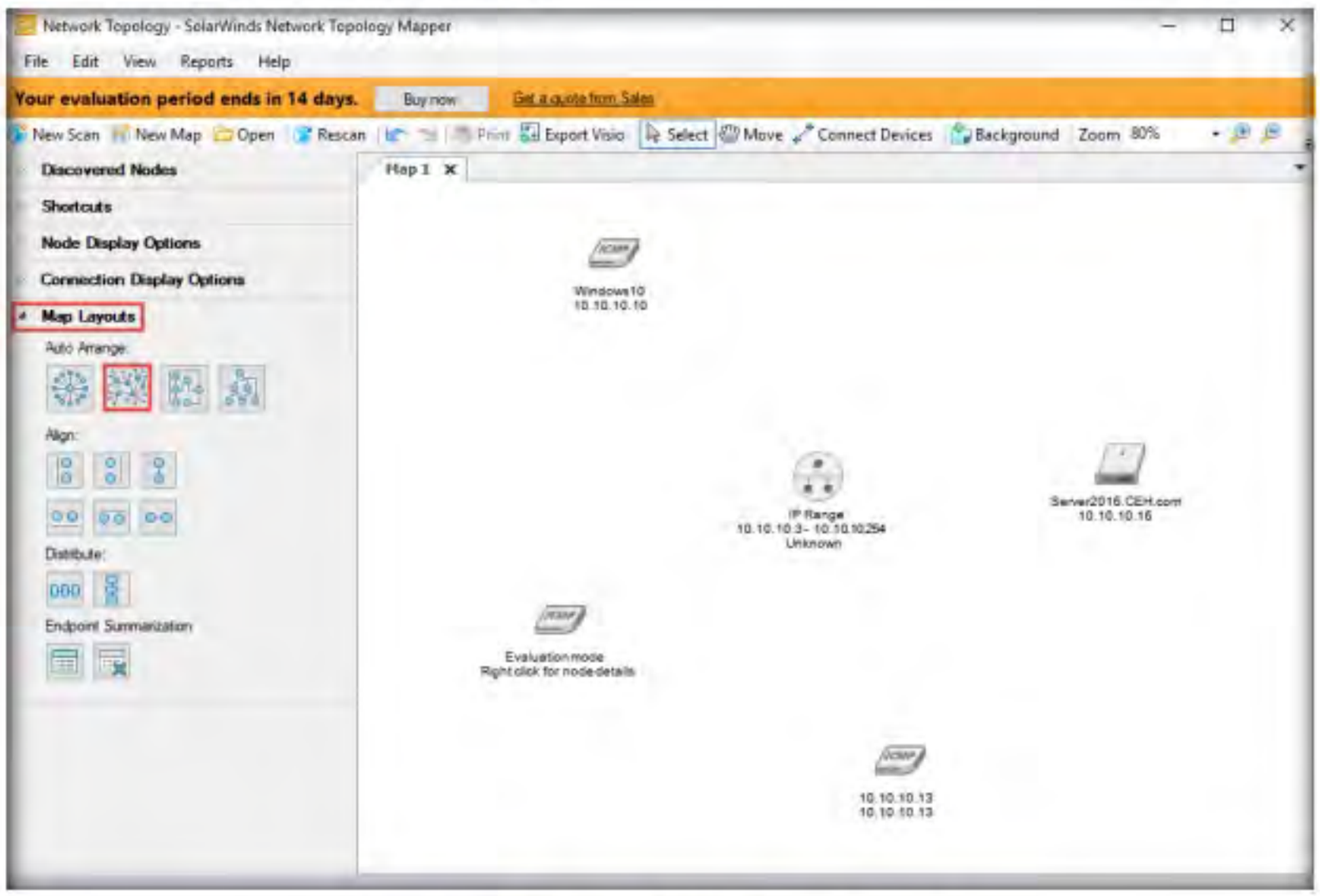


Figure 5.1.16: Map Layouts Options: selecting Symmetrical

22. Right-click on a node (here **Server2016** with IP address **10.10.10.16**) and select **Node Properties** to view information about the selected node.

Note: The network diagram and the IP addresses might differ in your lab environment

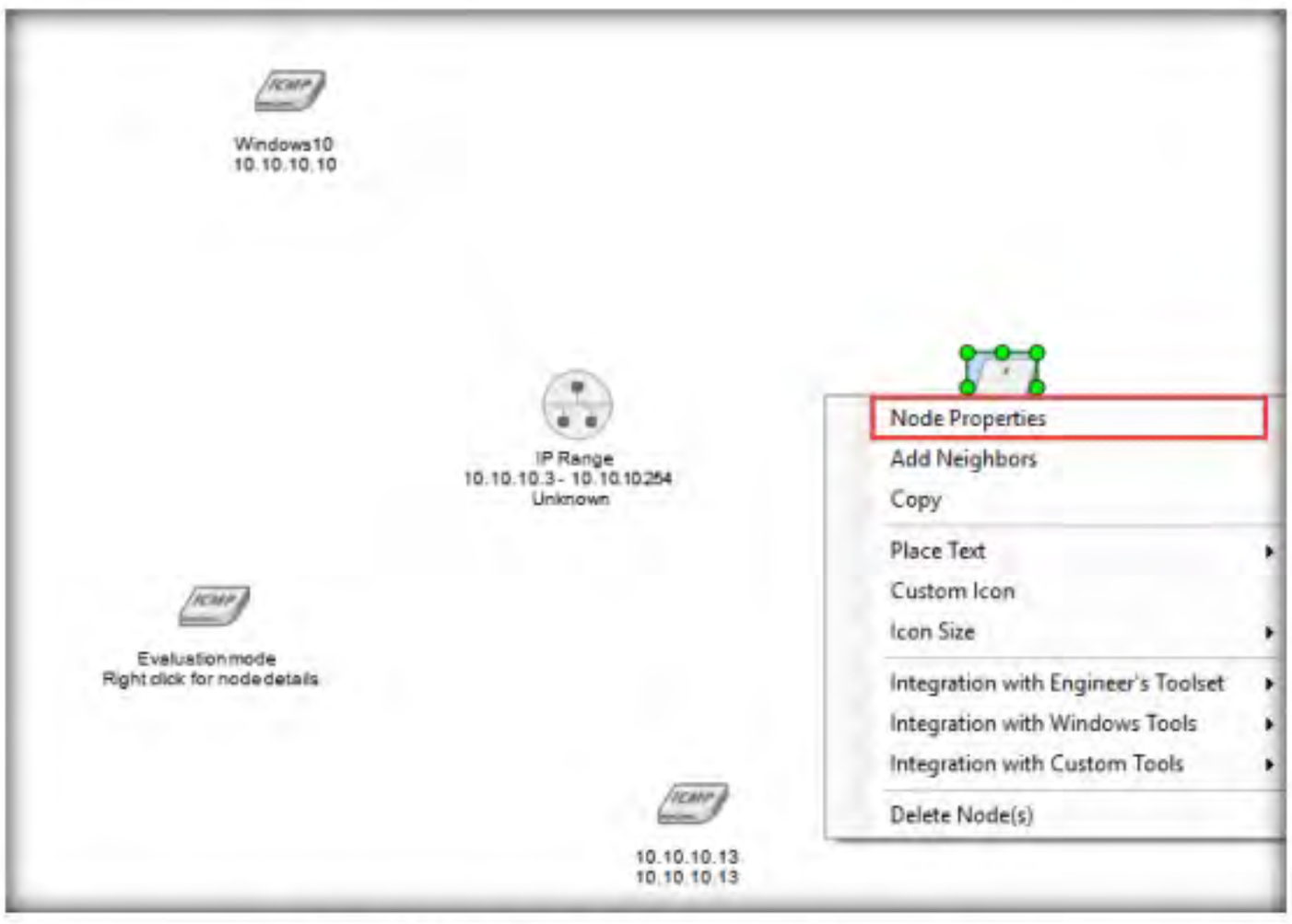


Figure 5.1.17: Viewing the details of a selected target machine

23. The **Node Details** window appears, displaying information about the selected node. Click **Close** to close the window.

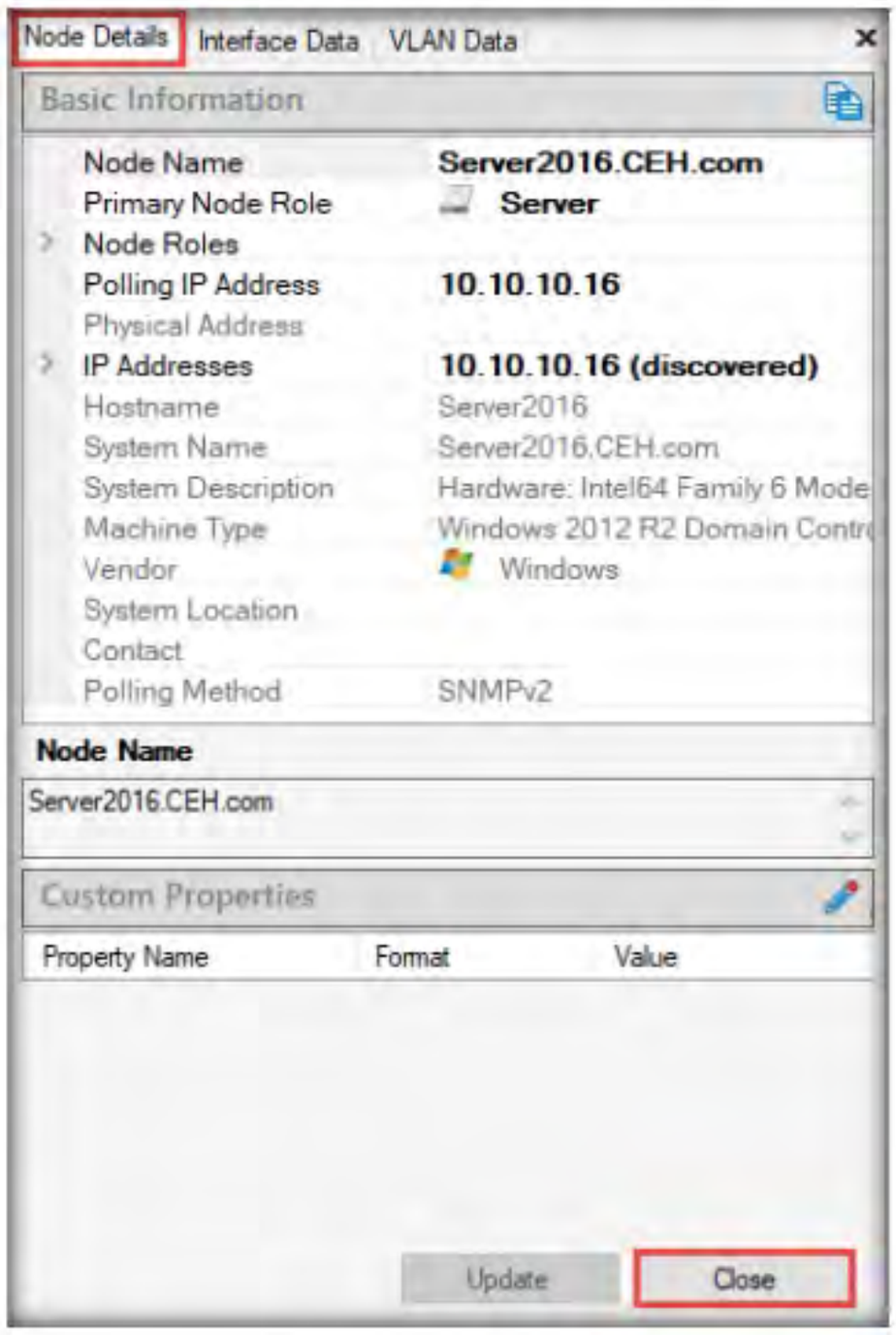


Figure 5.1.18: Details window

TASK 1.4

Additional Features in Network Topology Mapper

24. Right-click on a node (**Server2016** with IP address **10.10.10.16**) and select **Integration with Windows Tools** and click **Remote Desktop**.

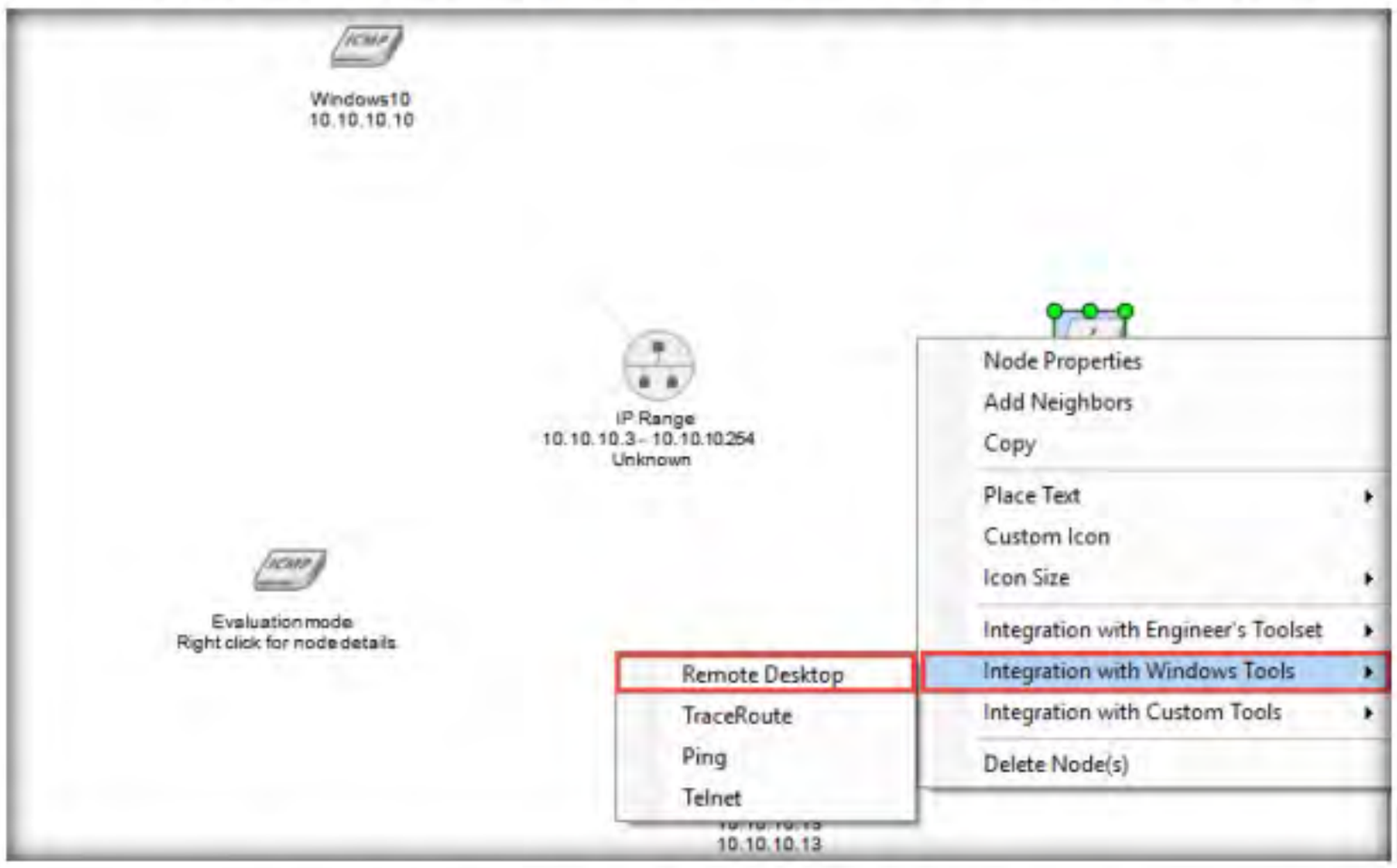


Figure 5.1.19: Establishing a Remote Desktop Connection with the target machine

25. The **Windows Security** dialog box appears. Enter **Username** as **Administrator** and **Password** as **Pa$$w0rd** for **Windows Server 2016**, and click **OK**.

Figure 5.1.20: Establishing a Remote Desktop Connection with the target machine

26. The **Remote Desktop Connection** pop-up appears; click **Yes**.

Figure 5.1.21: Establishing a remote desktop connection with the target machine

27. The **Remote Desktop Connection** is successfully set to the target virtual machine (here, **Windows Server 2016**), as shown in the following screenshot.

Figure 5.1.22: Remote Desktop Connection established with the target machine

28. You can use other options such as **Ping**, **Telnet**, and **Traceroute**. Similarly, an attacker can use this application to draw network diagrams, find the active hosts on the network, perform Ping, Telnet, etc.
29. This concludes the demonstration of drawing network diagram of the target network using Network Topology Mapper.
30. Close all open windows and document all the acquired information.
31. Turn off all the virtual machines.

You can also use other network discovery tools such as **OpManager** (https://www.manageengine.com), **The Dude** (https://mikrotik.com), **NetSurveyor** (http://nutsaboutnets.com), **NetBrain** (https://www.netbraintech.com), and **Spiceworks Network Mapping Tool** (https://www.spiceworks.com) to draw network diagram of the target network.

Lab Analysis

Analyze and document all the results discovered in the lab exercise.

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS ABOUT THIS LAB.

Internet Connection Required	
☐ Yes	☑ No
Platform Supported	
☑ Classroom	☑ iLabs

Lab 6

Perform Network Scanning using Various Scanning Tools

Ethical hackers and pen testers are aided in network scanning with the help of various scanning tools, which make scanning a target network an easy task.

ICON KEY

- Valuable information
- Test your knowledge
- Web exercise
- Workbook review

Lab Scenario

The information obtained in the previous steps might be insufficient to reveal potential vulnerabilities in the target network: there may be more information available that could help in finding loopholes in the target network. As an ethical hacker and pen tester, you should look for as much information as possible about systems in the target network using various network scanning tools when needed. This lab will demonstrate other techniques/commands/methods that can assist you in extracting information about the systems in the target network using various scanning tools.

Lab Objectives

- Scan a target network using Metasploit

Tools demonstrated in this lab are available in E:\CEH-Tools\CEHv11 Module 03 Scanning Networks

Lab Environment

To carry out this lab, you need:

- Windows 10 virtual machine
- Windows Server 2016 virtual machine
- Parrot Security virtual machine
- Ubuntu virtual machine
- Web browsers with an Internet connection
- Administrator privileges to run the tools

Lab Duration

Time: 20 Minutes

Overview of Network Scanning Tools

Scanning tools are used to scan and identify live hosts, open ports, running services on a target network, location-info, NetBIOS info, and information about all TCP/IP and UDP open ports. Information obtained from these tools will assist an ethical hacker in creating the profile of the target organization and to scan the network for open ports of the devices connected.

Lab Tasks

TASK 1

Scan a Target Network using Metasploit

Here, we will use Metasploit to discover active hosts, open ports, services running, and OS details of systems present in the target network.

TASK 1.1

Link Metasploit Framework to Database

Metasploit Framework is a tool that provides information about security vulnerabilities in the target organization's system, and aids in penetration testing and IDS signature development. It facilitates the tasks of attackers, exploit writers, and payload writers. A major advantage of the framework is the modular approach, that is, allowing the combination of any exploit with any payload.

1. Before starting this task, turn on the **Windows 10**, **Windows Server 2016**, **Parrot Security**, and **Ubuntu** virtual machine.
2. Switch to the **Parrot Security** virtual machine.
3. In the login page, the **attacker** username will be selected by default. Enter password as **toor** in the **Password** field and press **Enter** to log in to the machine.

Note:

- If a **Parrot Updater** pop-up appears at the top-right corner of **Desktop**, ignore and close it.
- If a **Question** pop-up window appears asking you to update the machine, click **No** to close the window.

4. Click the **MATE Terminal** icon at the top of the **Desktop** window to open a **Terminal** window.
5. A **Parrot Terminal** window appears. In the terminal window, type **sudo su** and press **Enter** to run the programs as a root user.
6. In the **[sudo] password for attacker** field, type **toor** as a password and press **Enter**.

Note: The password that you type will not be visible.

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

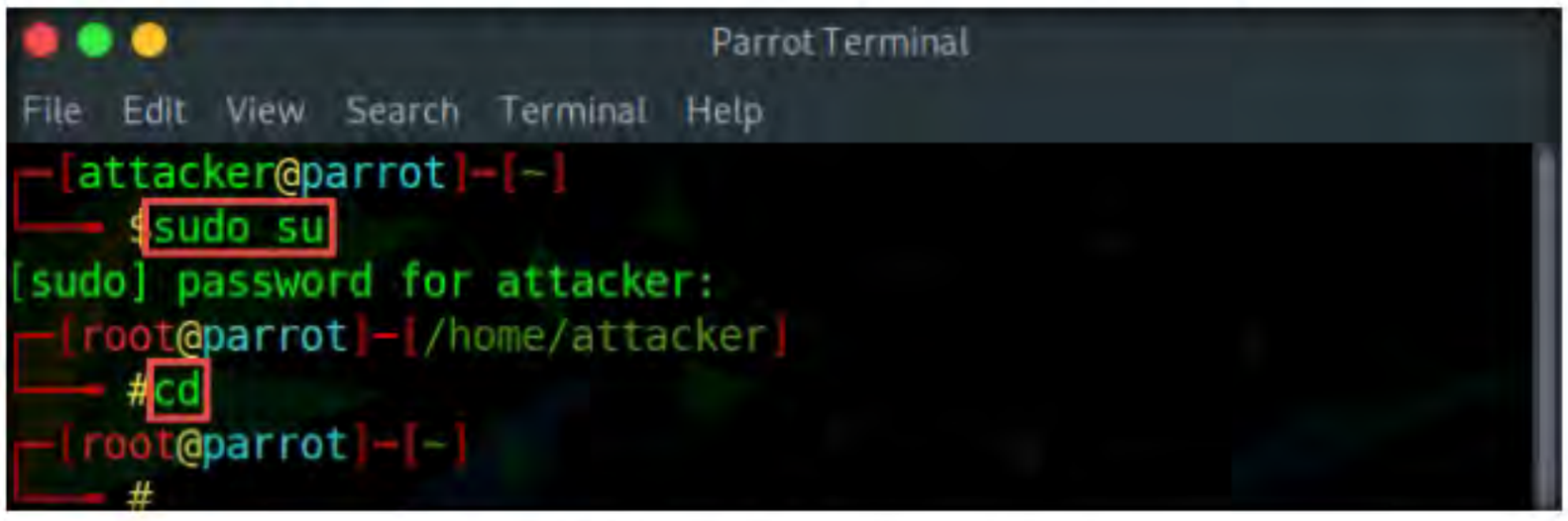


Figure 6.1.1: Running the programs as a root user

8. In the **Parrot Terminal** window, type **service postgresql start** and hit **Enter**.

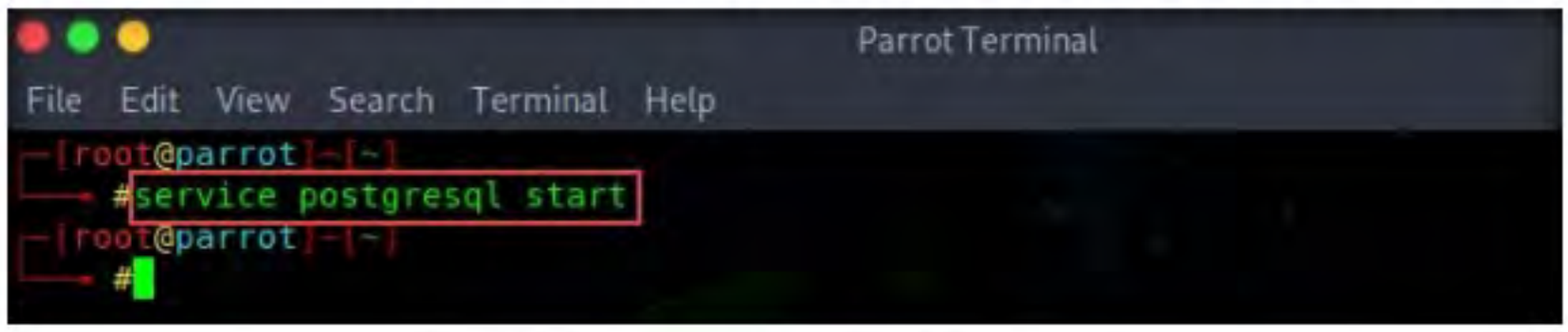


Figure 6.1.2: Start postgresql service through the command line

9. Now, type **msfconsole** and hit **Enter** to launch Metasploit.

Note: The version of Metasploit might differ in your lab environment.

Figure 6.1.3: Launch Metasploit framework

10. An msf command line appears. Type **db_status** and hit **Enter** to check if Metasploit has connected to the database successfully. If you receive the message "**postgersql selected, no connection**," then the database did not connect to msf.

Note: If the message you receive is "**connected to msf**," then skip to **Step #14**.

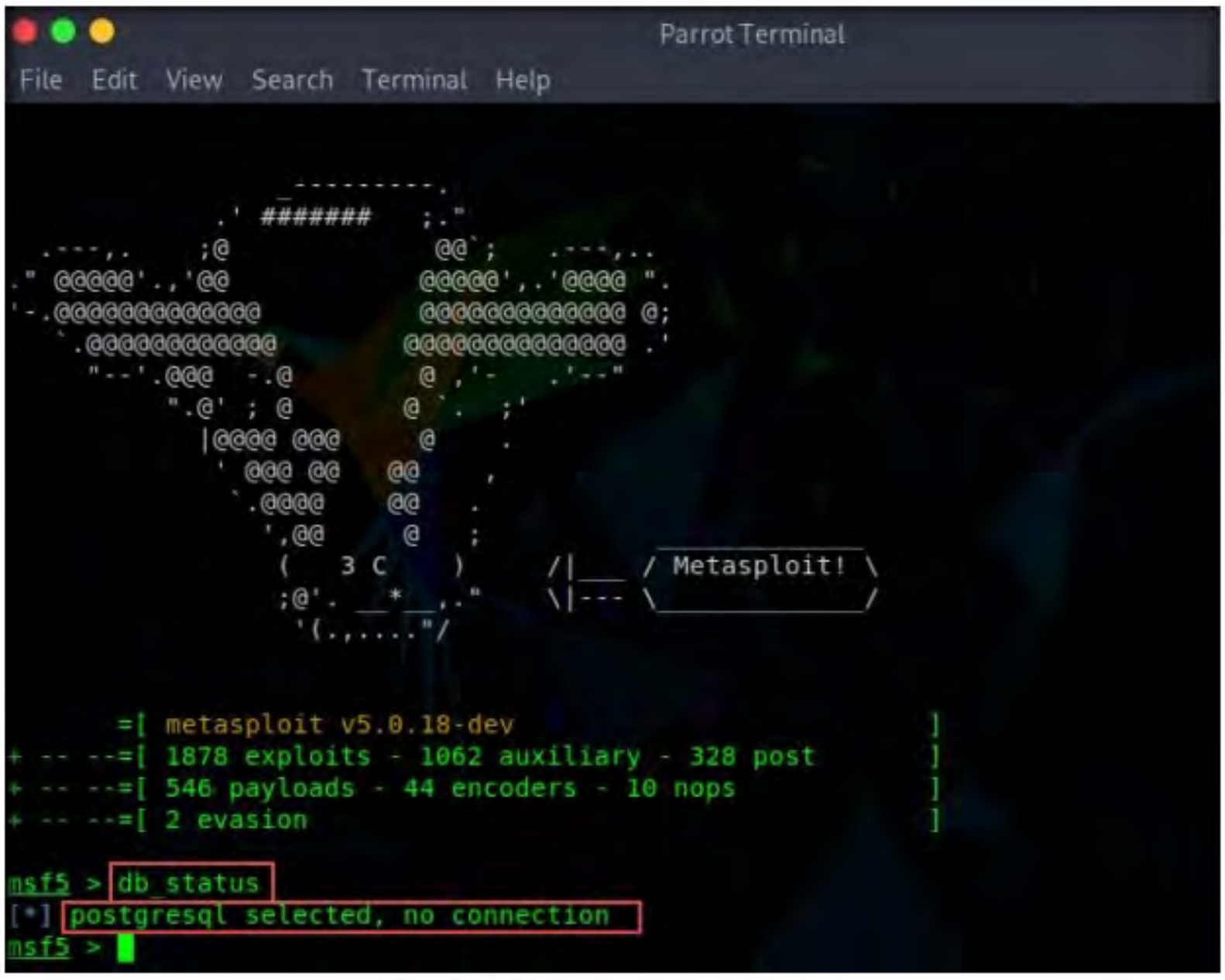


Figure 6.1.4: Database not connected to msf

11. Exit the Metasploit framework by typing **exit** and press **Enter**. Then, to initiate the database, type **msfdb init**, and press **Enter**.

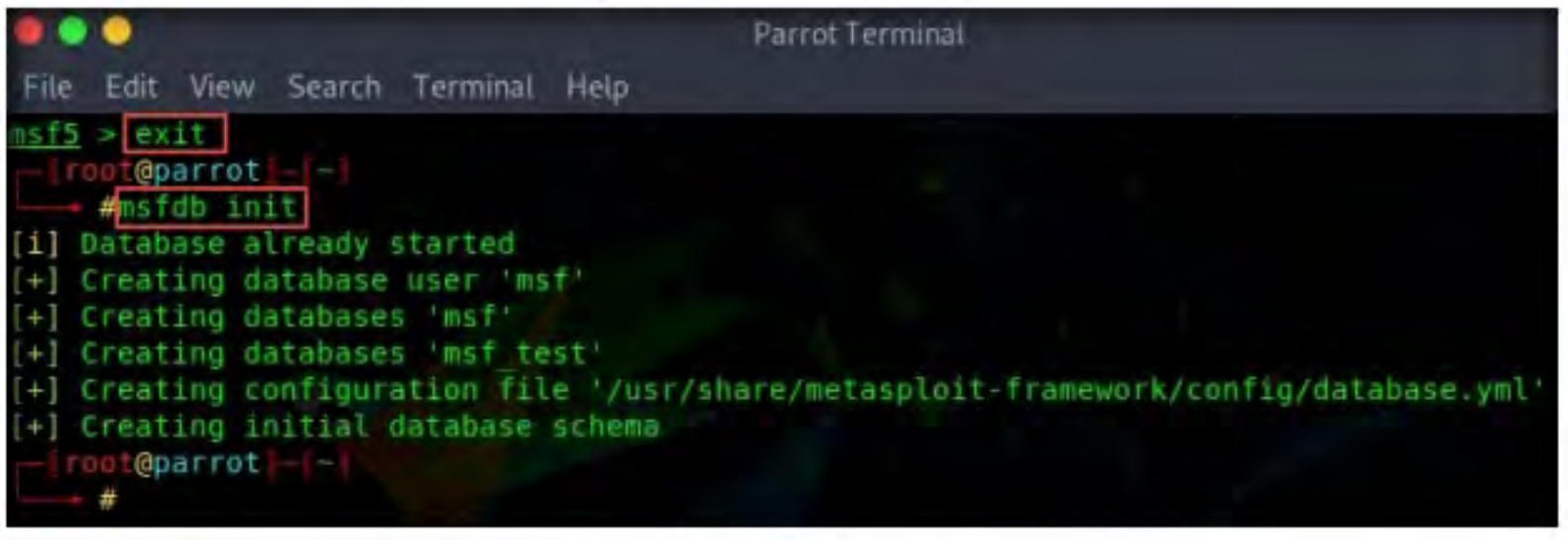


Figure 6.1.5: Initialize the database

12. To restart the postgresql service, type **service postgresql restart** and press **Enter**. Now, start the Metasploit Framework again by typing **msfconsole** and pressing **Enter**.

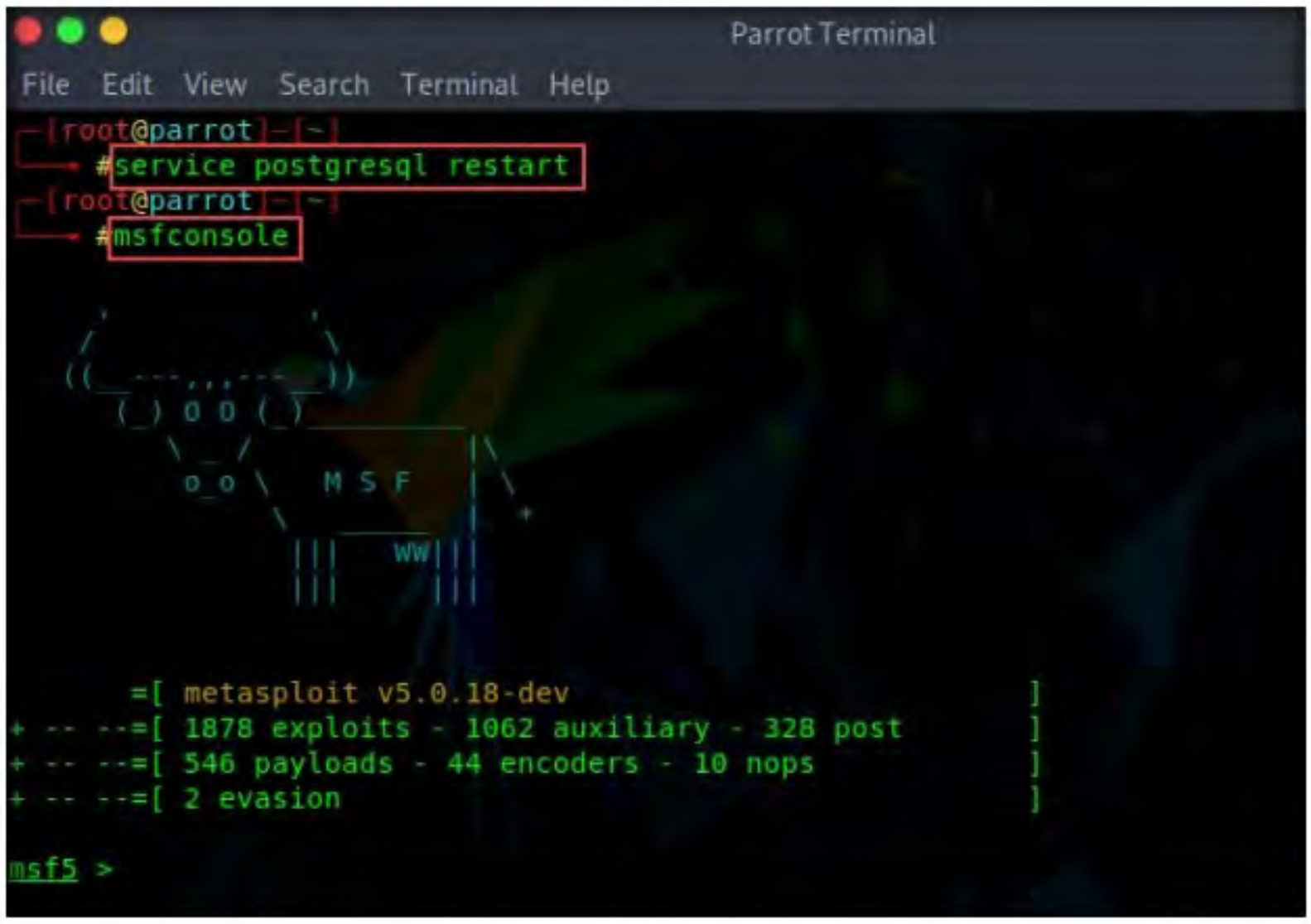


Figure 6.1.6: Launch Metasploit framework

13. Check the database status by typing **db_status** and press **Enter**. This time, the database should successfully connect to msf, as shown in the screenshot.

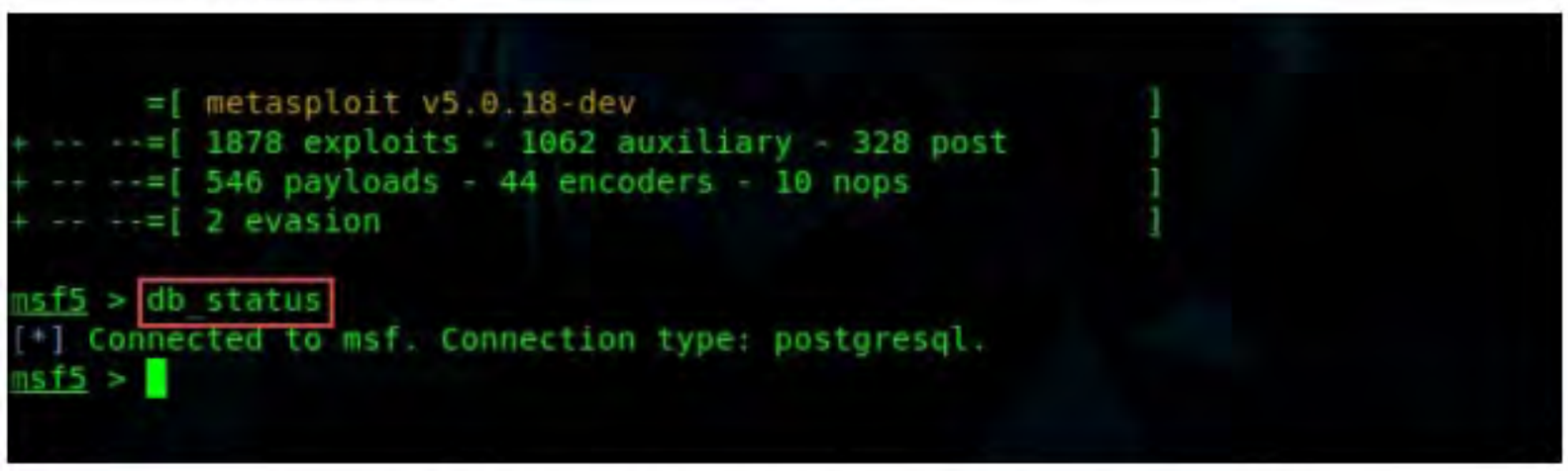


Figure 6.1.7: Verify that database is successfully connected to Metasploit framework

TASK 1.2

Information Gathering Using Metasploit

14. Type **nmap -Pn -sS -A -oX Test 10.10.10.0/24** and hit **Enter** to scan the subnet, as shown in the screenshot.

Here, we are scanning the whole subnet 10.10.10.0/24 for active hosts.

Note: The target IP address subnet might differ in your lab environment.

15. Nmap begins scanning the subnet and displays the results. It takes approximately 5 minutes for the scan to complete.

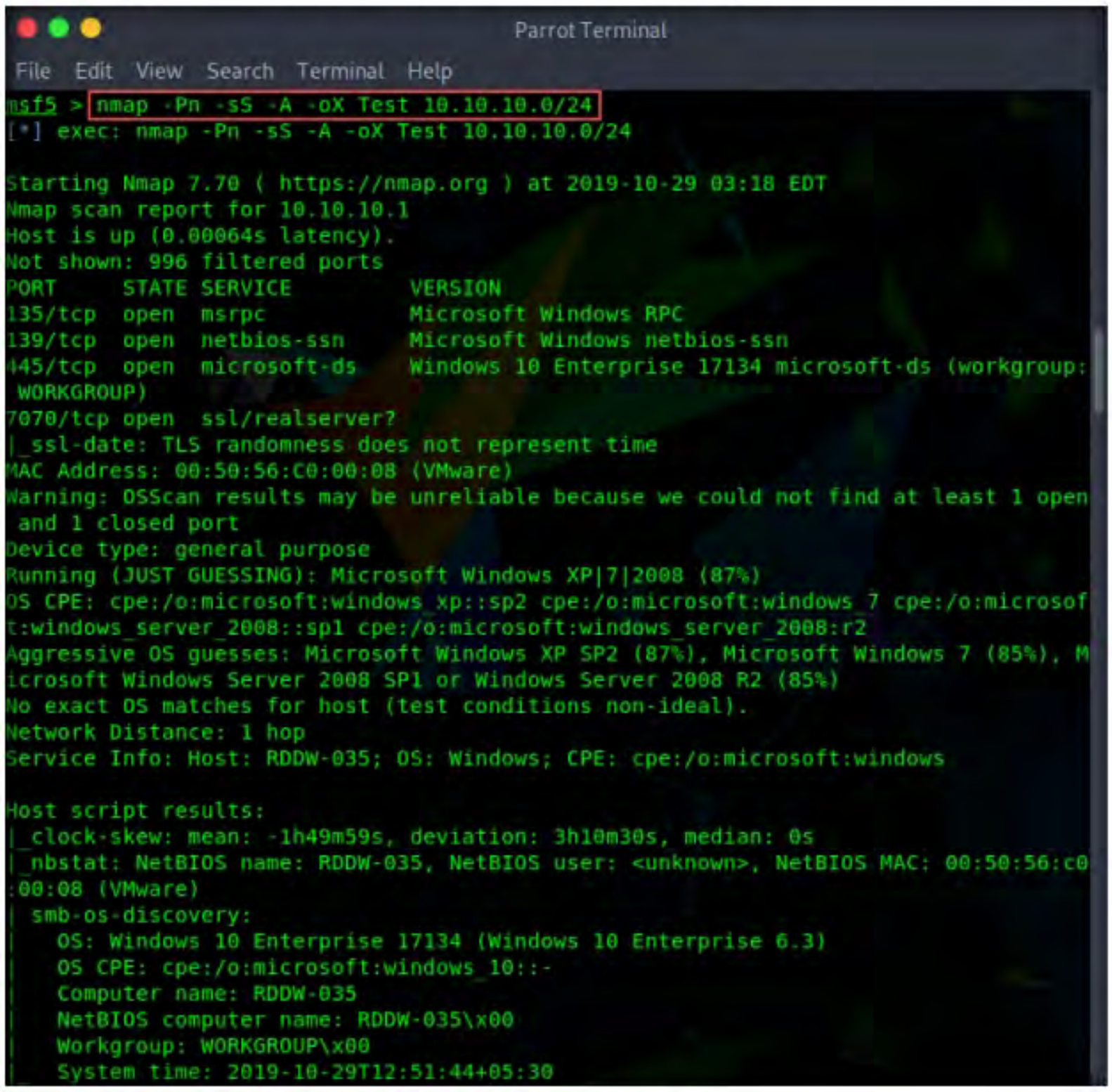


Figure 6.1.8: Nmap scan result

16. After the scan completes, Nmap displays the number of active hosts in the target network (here, **6**).

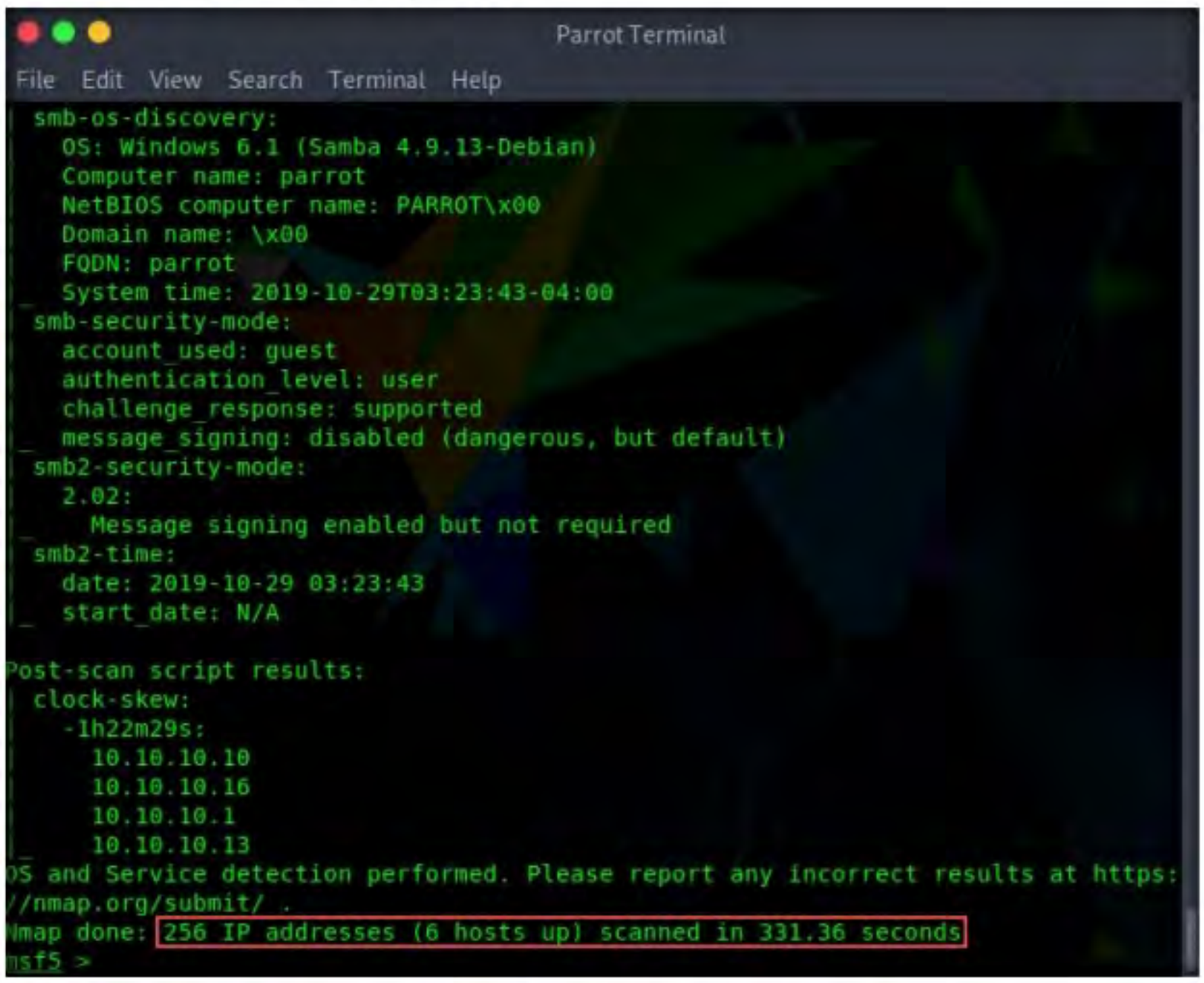


Figure 6.1.9: Nmap showing new found hosts in the subnet

17. Now, type **db_import Test** and hit **Enter** to import the Nmap results from the database.

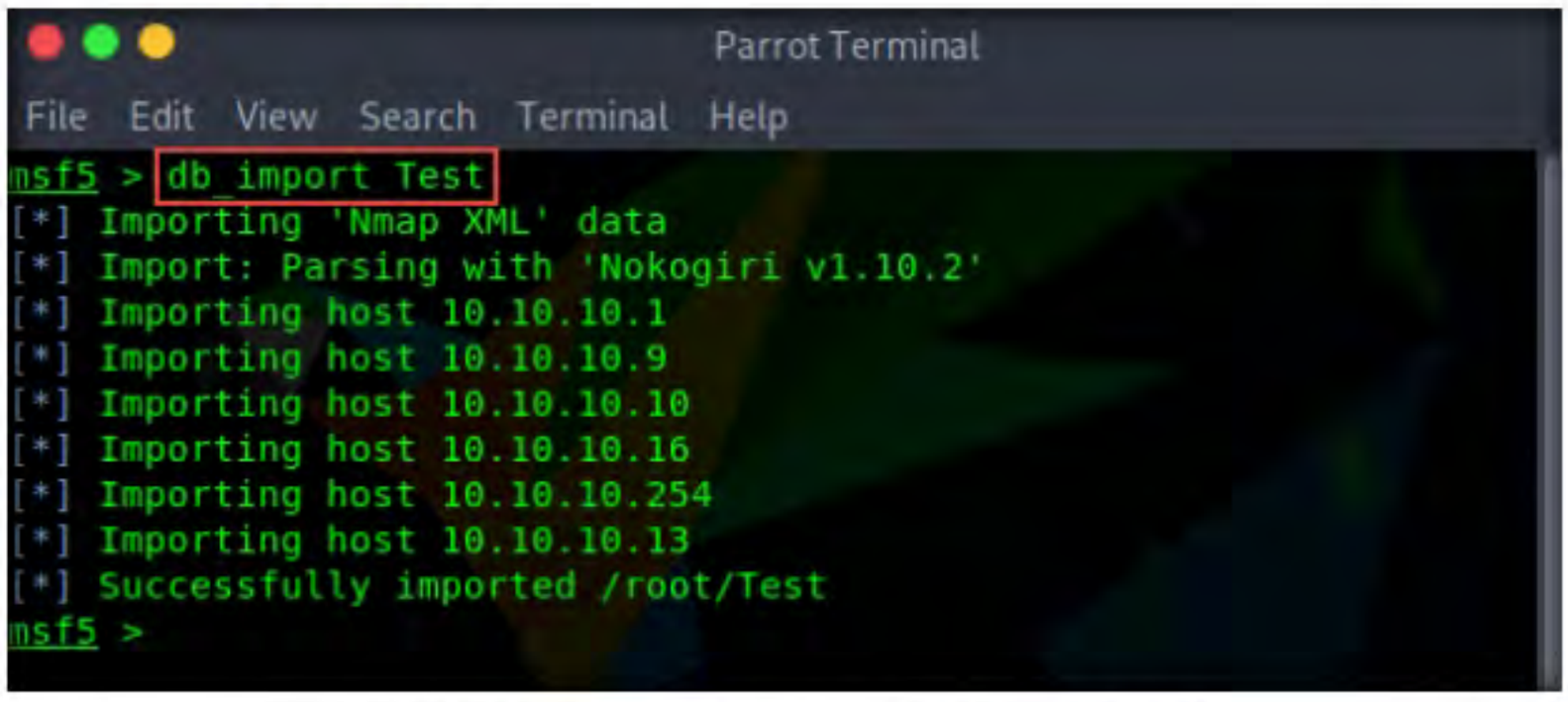


Figure 6.1.10: Importing Nmap results from the database using Metasploit

18. Type **hosts** and hit **Enter** to view the list of active hosts along with their MAC addresses, OS names, etc. as shown in the screenshot.

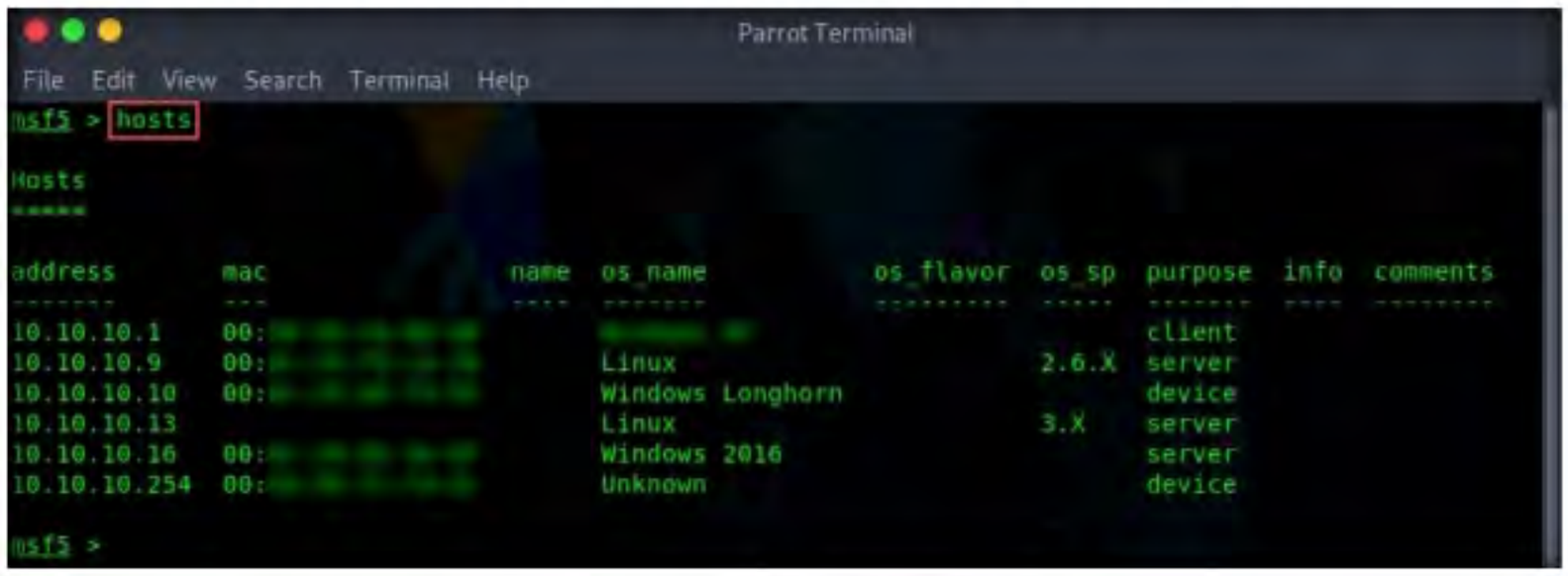


Figure 6.1.11: List of live hosts in the subnet

19. Type **services** or **db_services** and hit **Enter** to receive a list of the services running on the active hosts, as shown in the screenshot.

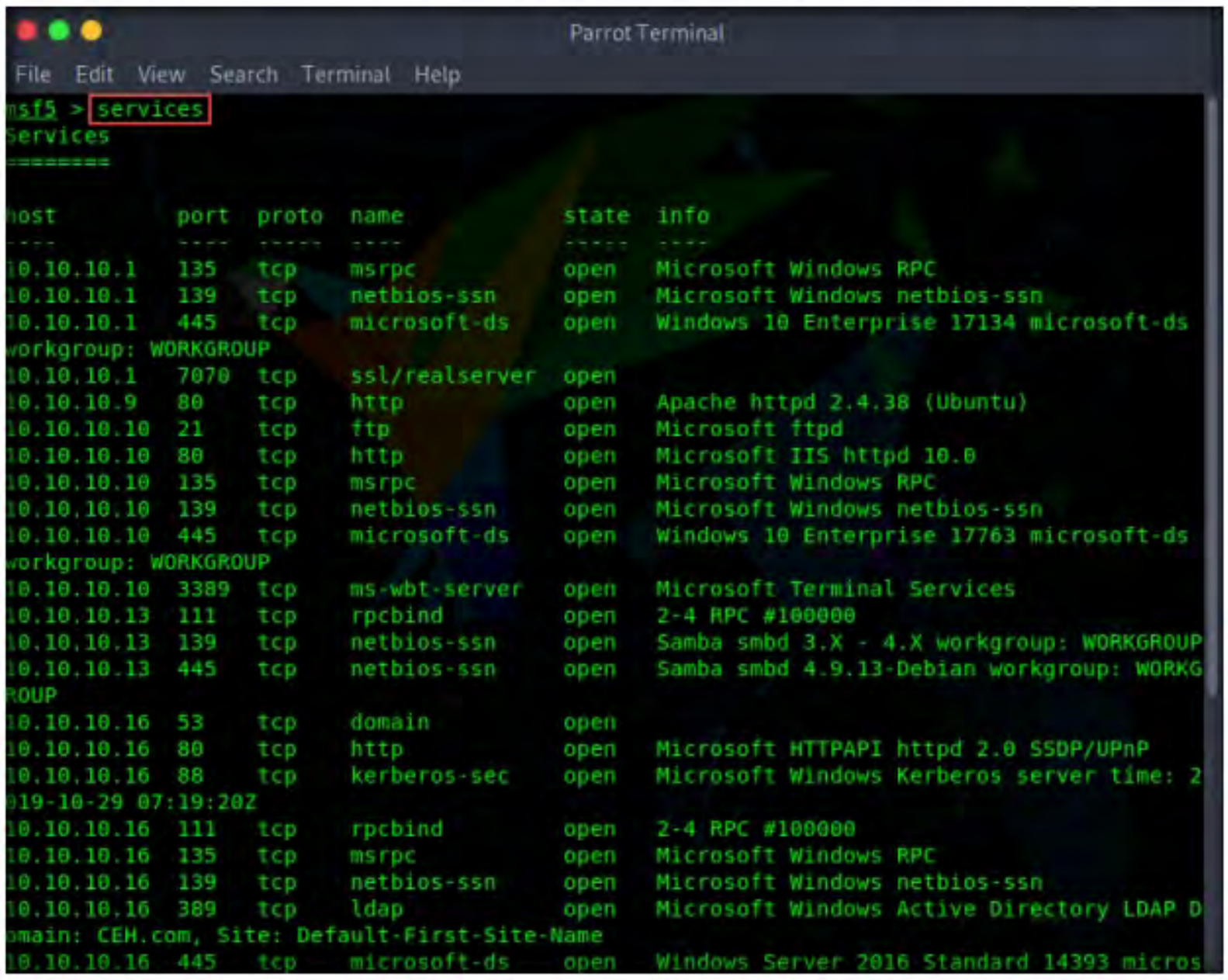


Figure 6.1.12: List of services running on the hosts in the subnet

Note: In addition to running Nmap, there are a variety of other port scanners that are available within the Metasploit framework to scan the target systems.

TASK 1.3

Perform Port Scanning Using MSF Modules

20. Type **search portscan** and hit **Enter**. The Metasploit port scanning modules appear, as shown in the screenshot.

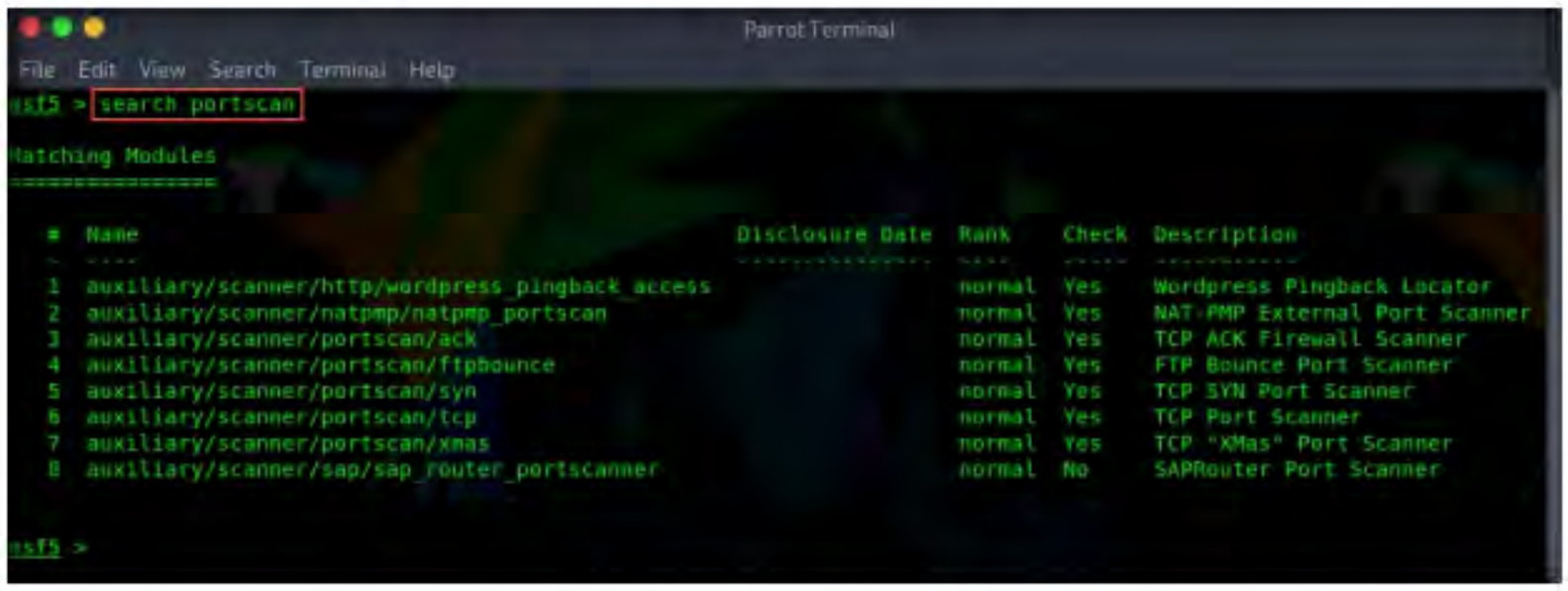


Figure 6.1.13: List portscan modules

21. Here, we will use the **auxiliary/scanner/portscan/syn** module to perform an SYN scan on the target systems. To do so, type **use auxiliary/scanner/portscan/syn** and press **Enter**.

22. We will use this module to perform an SYN scan against the target IP address range (**10.10.10.5-20**) to look for open port 80 through the eth0 interface.

    To do so, issue the below commands:

    - **set INTERFACE eth0**
    - **set PORTS 80**
    - **set RHOSTS 10.10.10.5-20**
    - **set THREADS 50**

Note: **PORTS**: specifies the ports to scan (e.g., 22-25, 80, 110-900), **RHOSTS**: specifies the target address range or CIDR identifier, and **THREADS**: specifies the number of concurrent threads (default 1).

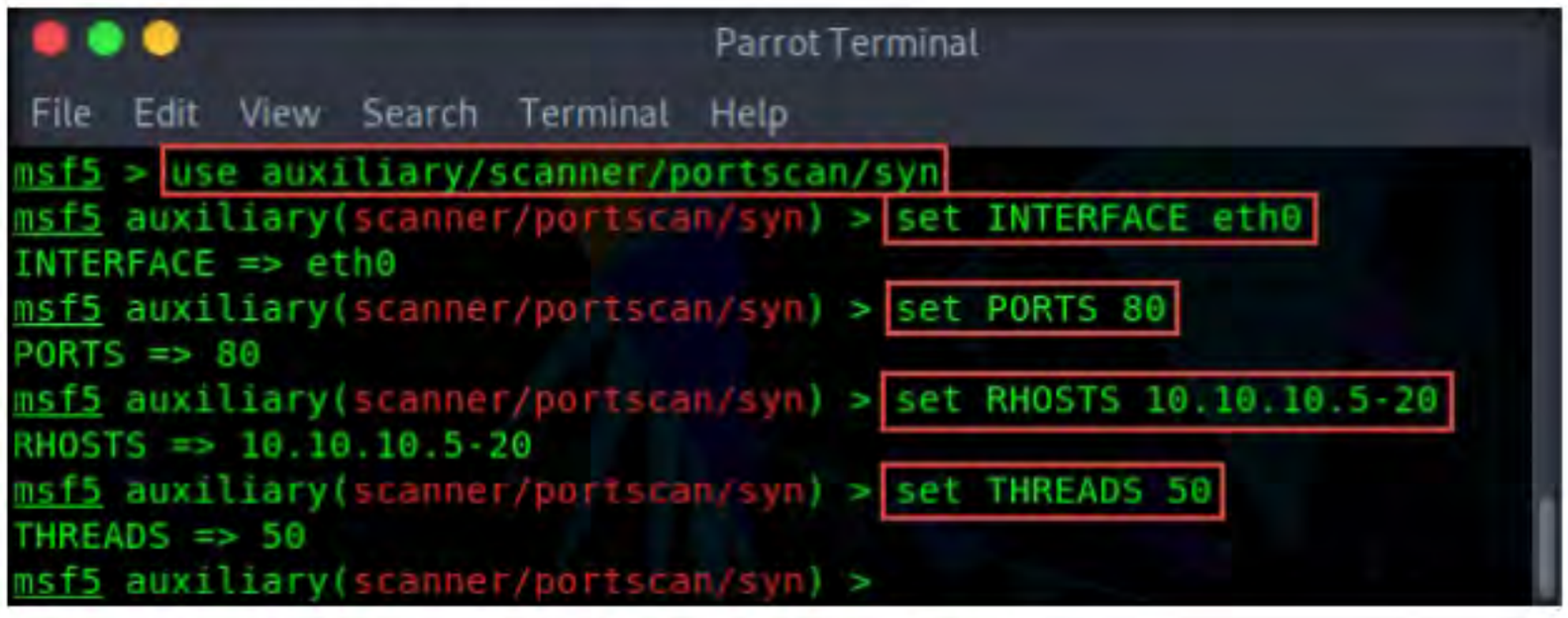


Figure 6.1.14: Setting values to perform a port scan

23. After specifying the above values, type **run**, and press **Enter** to initiate the scan against the target IP address range.

Note: Similarly, you can also specify a range of ports to be scanned against the target IP address range.

24. The result appears, displaying open port 80 in active hosts, as shown in the screenshot.

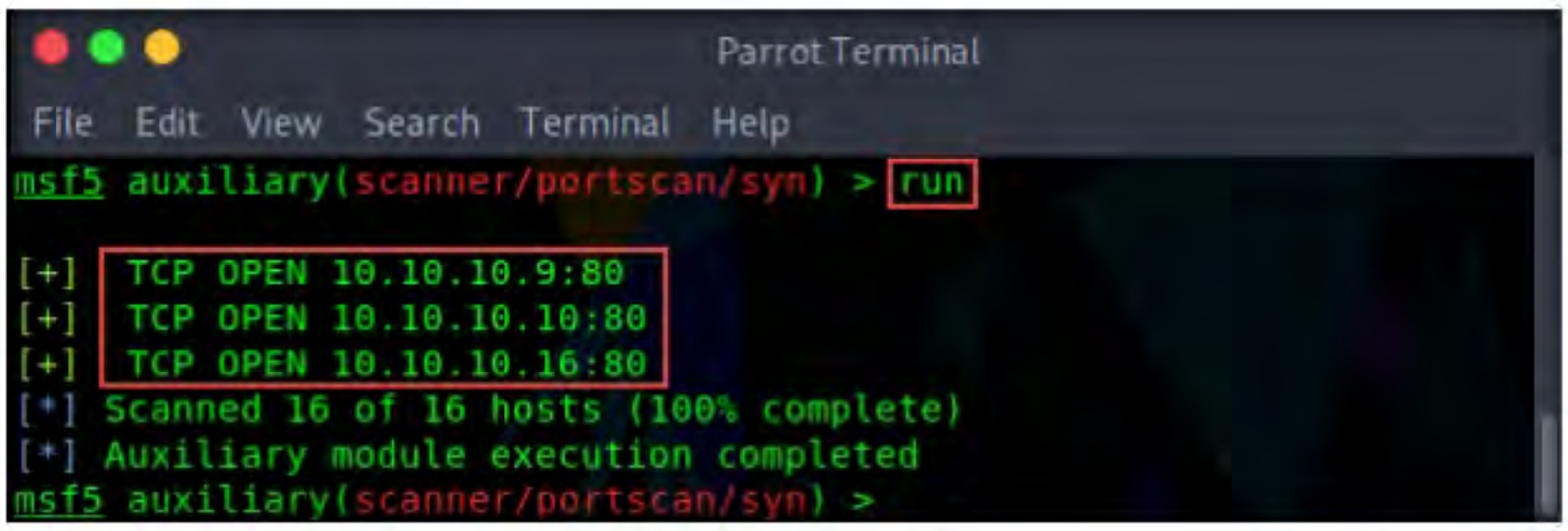


Figure 6.1.15: Port scan result

25. Now, we will perform a TCP scan for open ports on the target systems. To do so, first type **back**, and then press **Enter** to revert to the msf command line.

26. To load the auxiliary/scanner/portscan/tcp module, type **use auxiliary/scanner/portscan/tcp** and press **Enter**.

27. Type **hosts -R** and press **Enter** to automatically set this option with the discovered hosts present in our database.

Or

Type **set RHOSTS <Target IP Address>** and press **Enter**.

Note: Here, we will perform a TCP scan for open ports on a single IP address (**10.10.10.16**), as scanning multiple IP addresses consumes much time.

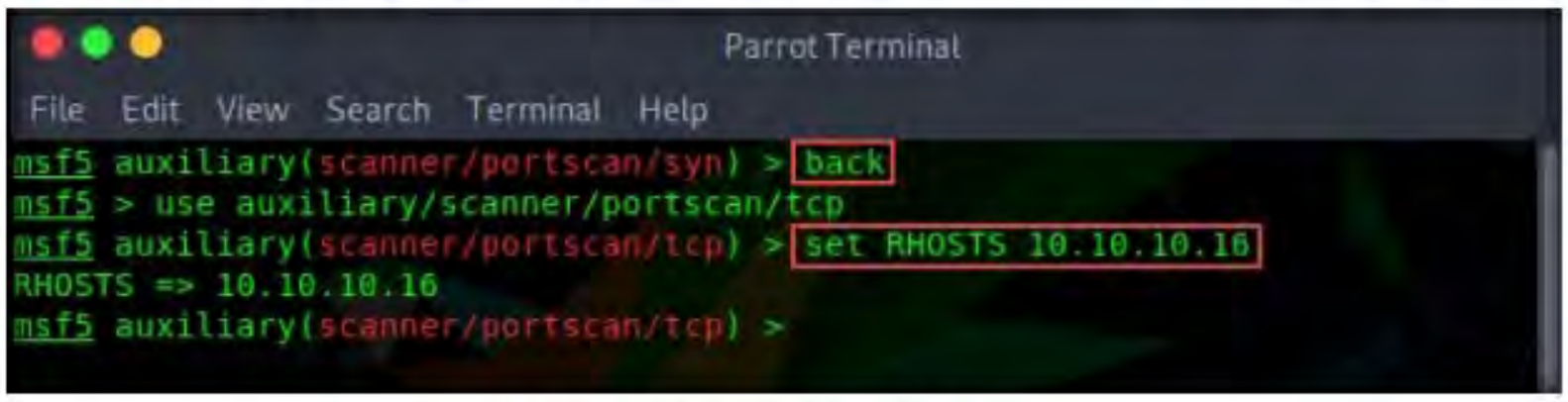


Figure 6.1.16: Port scanning using TCP scan

28. Type **run** and press **Enter** to discover open TCP ports in the target system.

29. The results appear, displaying all open TCP ports in the target IP address (10.10.10.16).