Computer Network and Defense Fundamentals

Module 01

Lab 1

Basic Network Administration and Troubleshooting Using Windows Command Line Utilities

Windows offers several powerful command line utilities that help administrators in troubleshooting their network connections.

ICON KEY

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

Lab Scenario

Network troubleshooting is becoming the most common task that a network admin needs to perform in large or medium organizations. As a network administrator, you are often required to troubleshoot the network problems as a part of your role and responsibilities. Administrators should have basic knowledge of network troubleshooting required to diagnose, monitor, and repair network connections. There are various basic Windows commands available to diagnose a network problem that every network admin needs to know.

Lab Objectives

This lab demonstrates the use of basic Windows command-line utilities to perform troubleshooting in the network

Lab Environment

To carry out this lab, you need:

- Windows Server 2012 and Windows 10 VMs
- Administrator privileges to run the tools

Lab Duration

Time: 25 Minutes

Overview of the Lab

Windows Command utilities such as ipconfig, Ping, tracert, nslookup, netstat, arp, etc., allows you to administer, diagnose, monitor, and repair network connections.

Note: Before starting this lab, login to Windows 10 VM (User: Admin, Password: Pa$$w0rd) and disable the network adapter:

- Go to Control Panel → Network and Internet → Network and Sharing Center, and click Change adapter settings

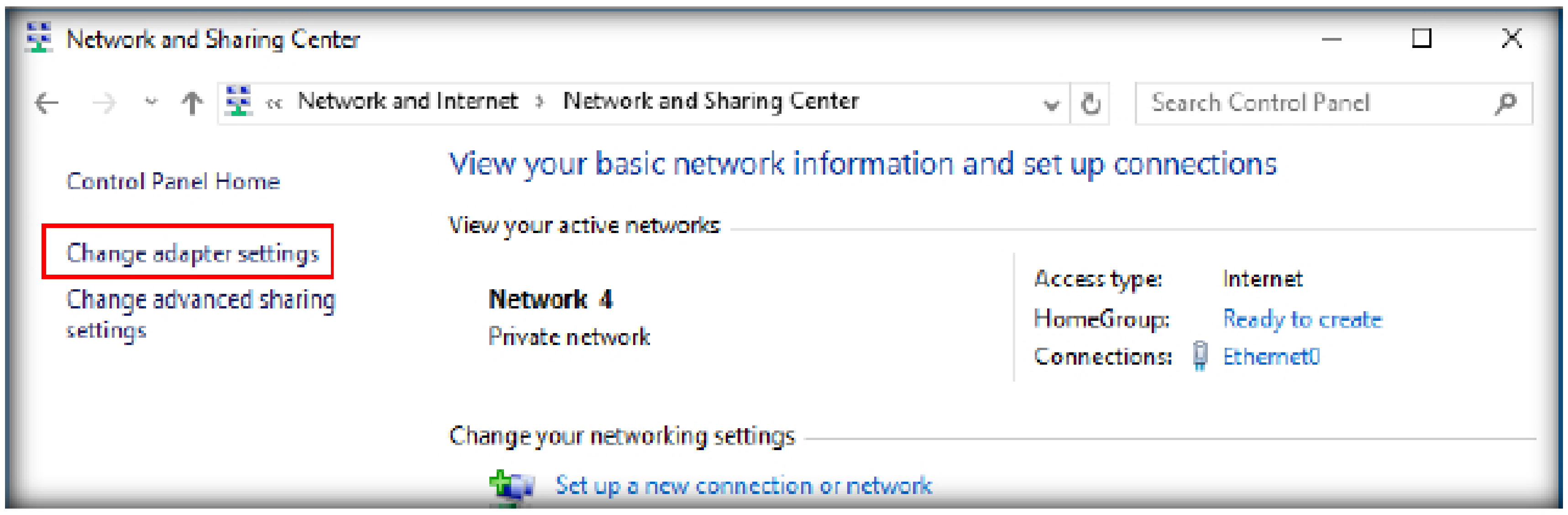

Network and Sharing Center
Network and Internet › Network and Sharing Center
Search Control Panel
Control Panel Home
Change adapter settings
Change advanced sharing settings
View your basic network information and set up connections
View your active networks
Network 4
Private network
Access type: Internet
HomeGroup: Ready to create
Connections: Ethernet0
Change your networking settings
Set up a new connection or network

FIGURE 1.1: Change Adapter Settings

- Select and right-click the Ethernet adapter, and click Disable from the context menu.

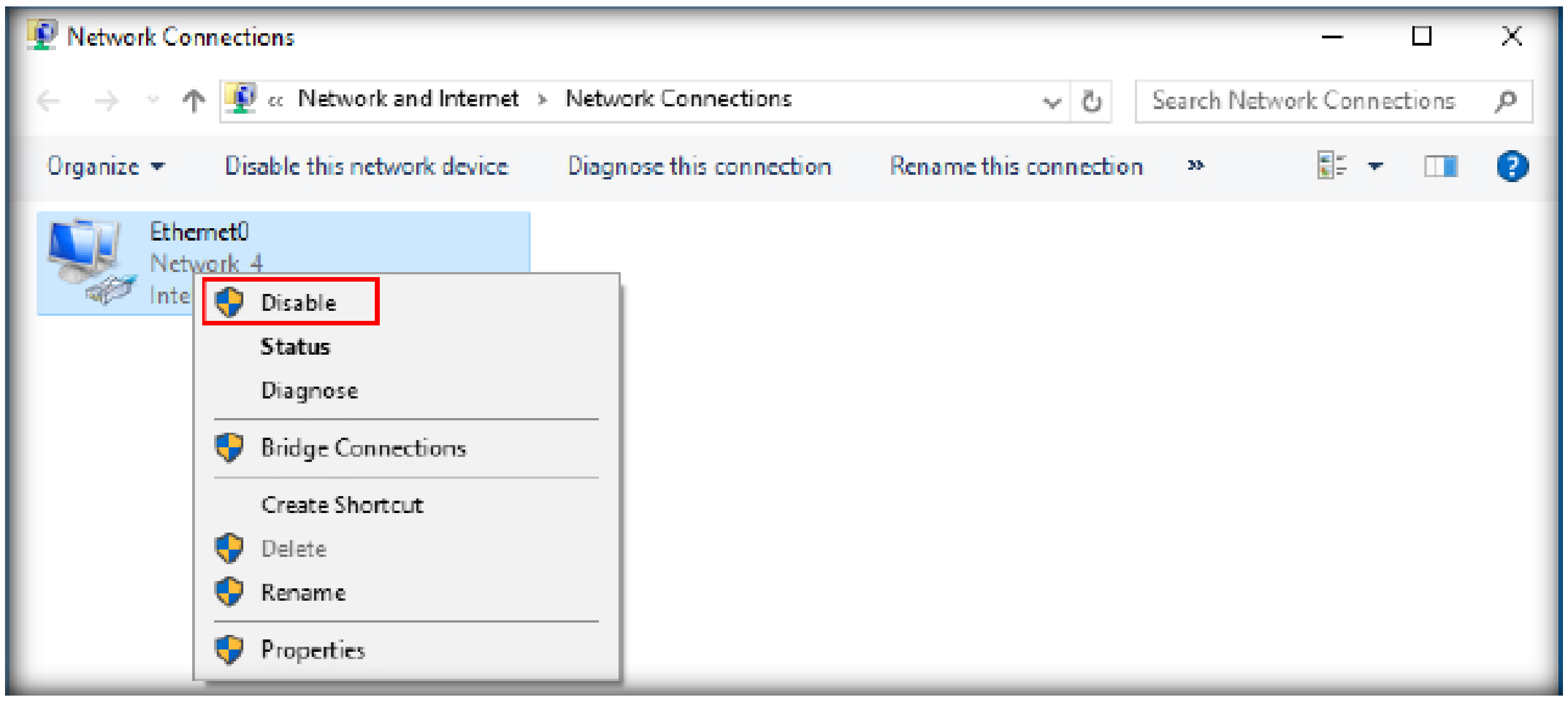


FIGURE 1.2: Disabling Network Adapter

- It will disable Ethernet adapter as shown below:

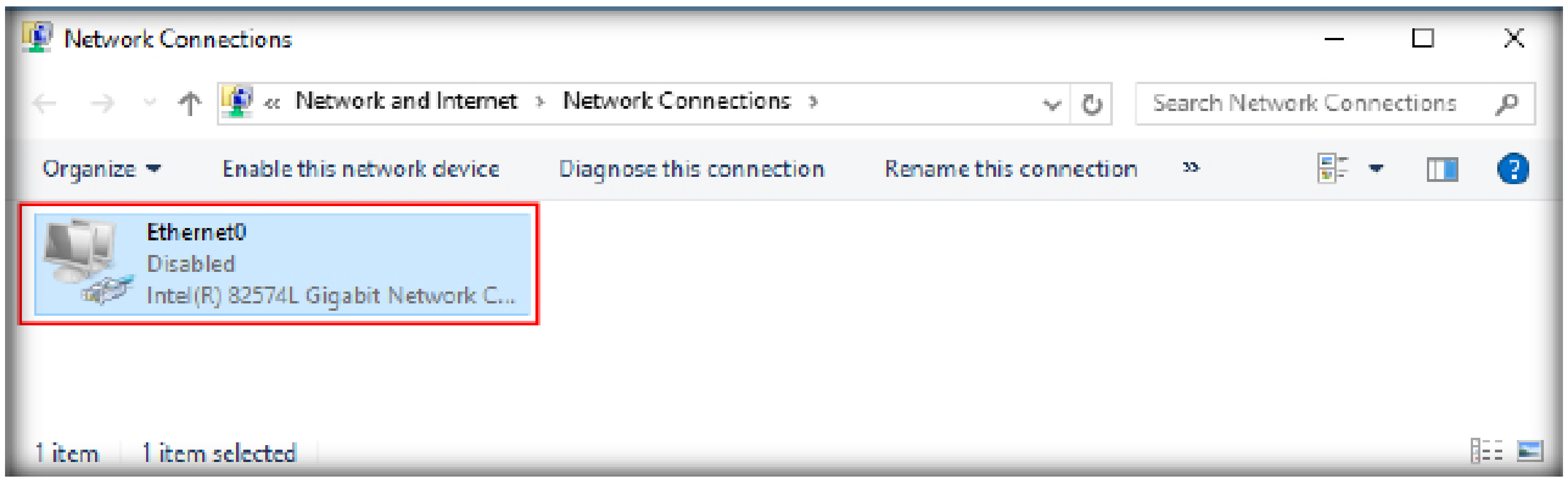

Network Connections
Network and Internet › Network Connections ›
Search Network Connections
Organize
Enable this network device
Diagnose this connection
Rename this connection
Ethernet0
Disabled
Intel(R) 82574L Gigabit Network C...
1 item
1 item selected

FIGURE 1.3: Network Adapter Disabled

Lab Tasks

TASK 1

Verifying IP Configuration Settings

1. Launch **Windows Server 2012** VM, and login to the local administrator account (username: **Administrator** and password: **Pa$$w0rd)**.

2. Open a command prompt in Admin mode by right-clicking on the **Start** icon and then click on **Command Prompt (Admin)** from the context menu.

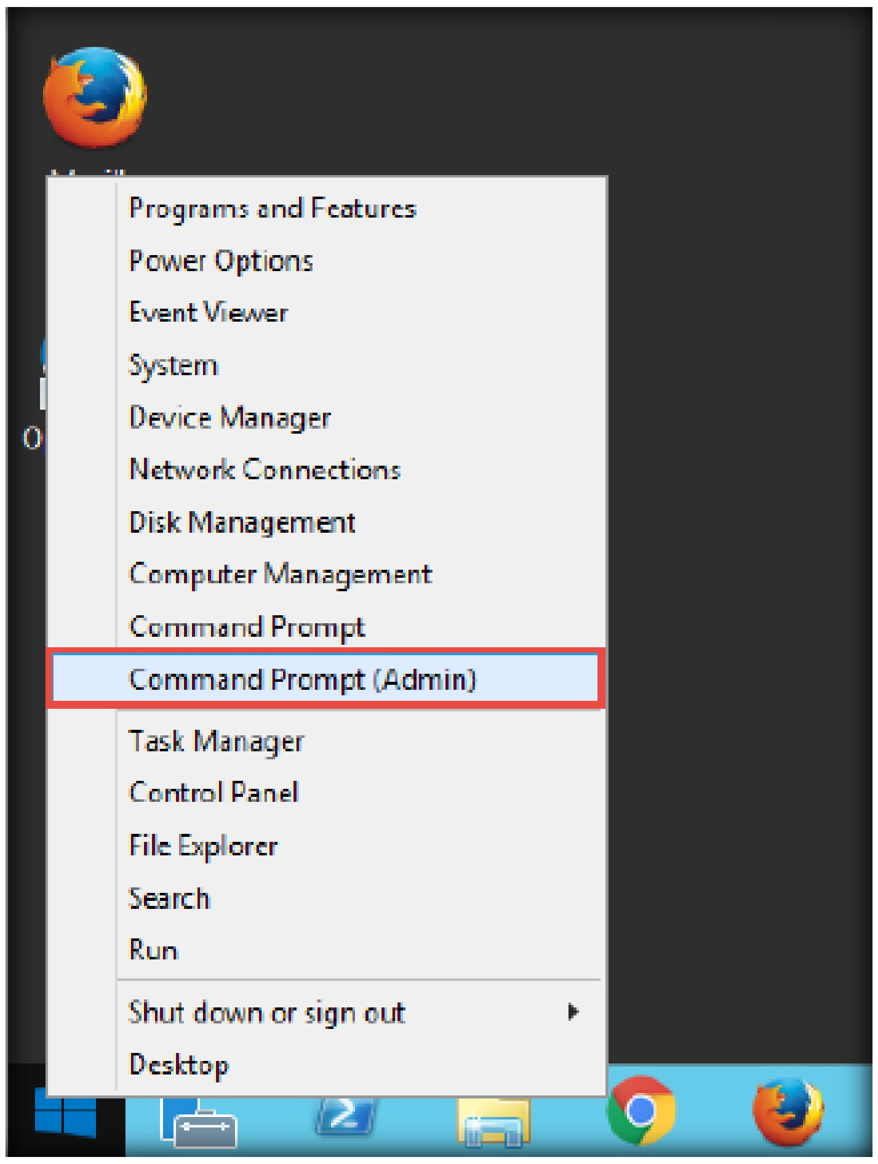
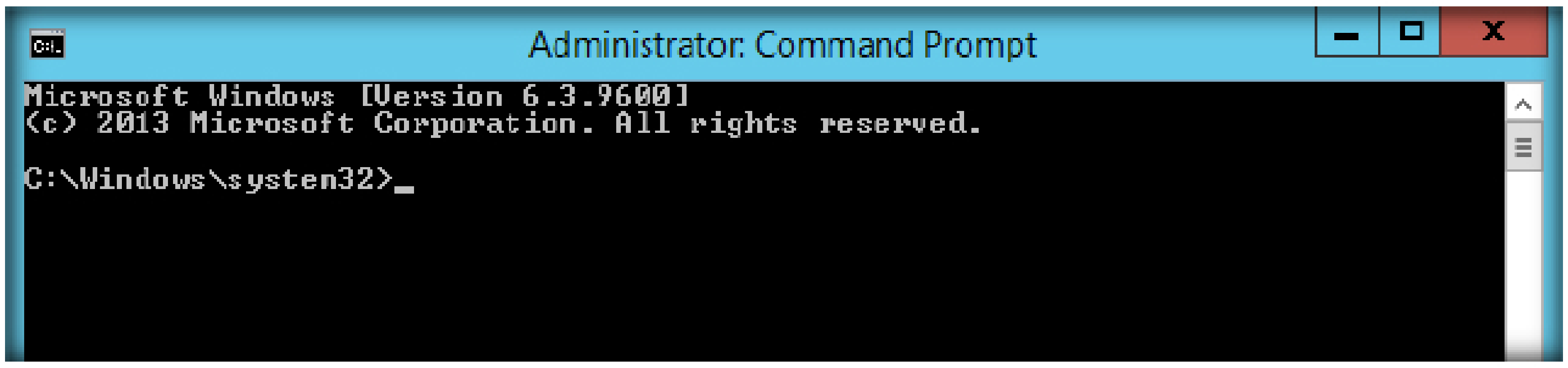


FIGURE 1.4: Launching Command Prompt

3. The command prompt appears on the screen



```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Windows\system32>_
```

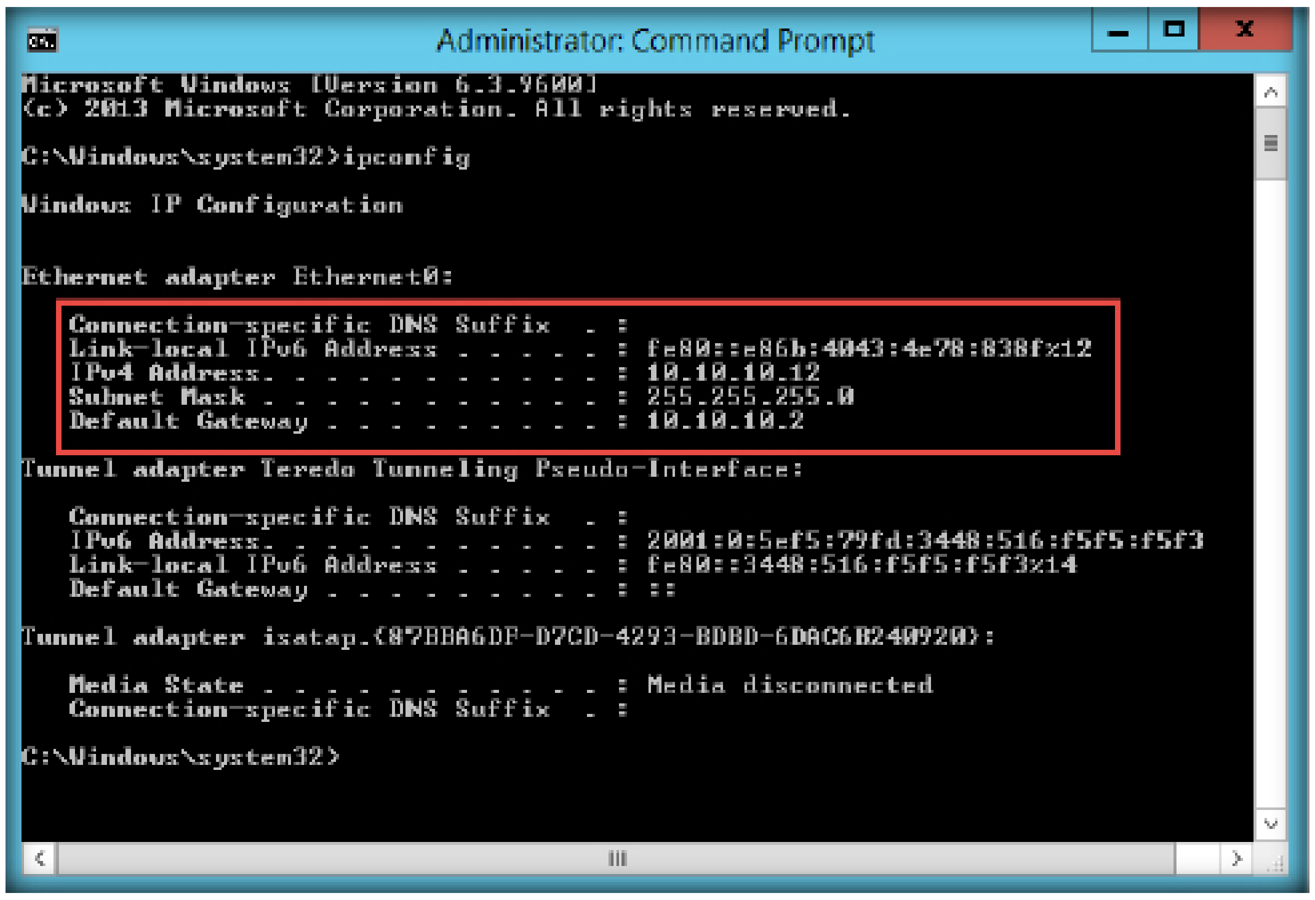
FIGURE 1.5: User Account Control

4. Type **ipconfig** in the command prompt and press **Enter** to verify the IP configuration settings of the machine.

5. The IP Configuration details of the system will be displayed. As a network admin you should know the IP configuration details of all the systems in the network.

ipconfig Syntax

ipconfig [/all] [/renew [Adapter]] [/release [Adapter]] [/flushdns] [/displaydns] [/registerdns] [/showclassid Adapter] [/setclassid Adapter [ClassID]].



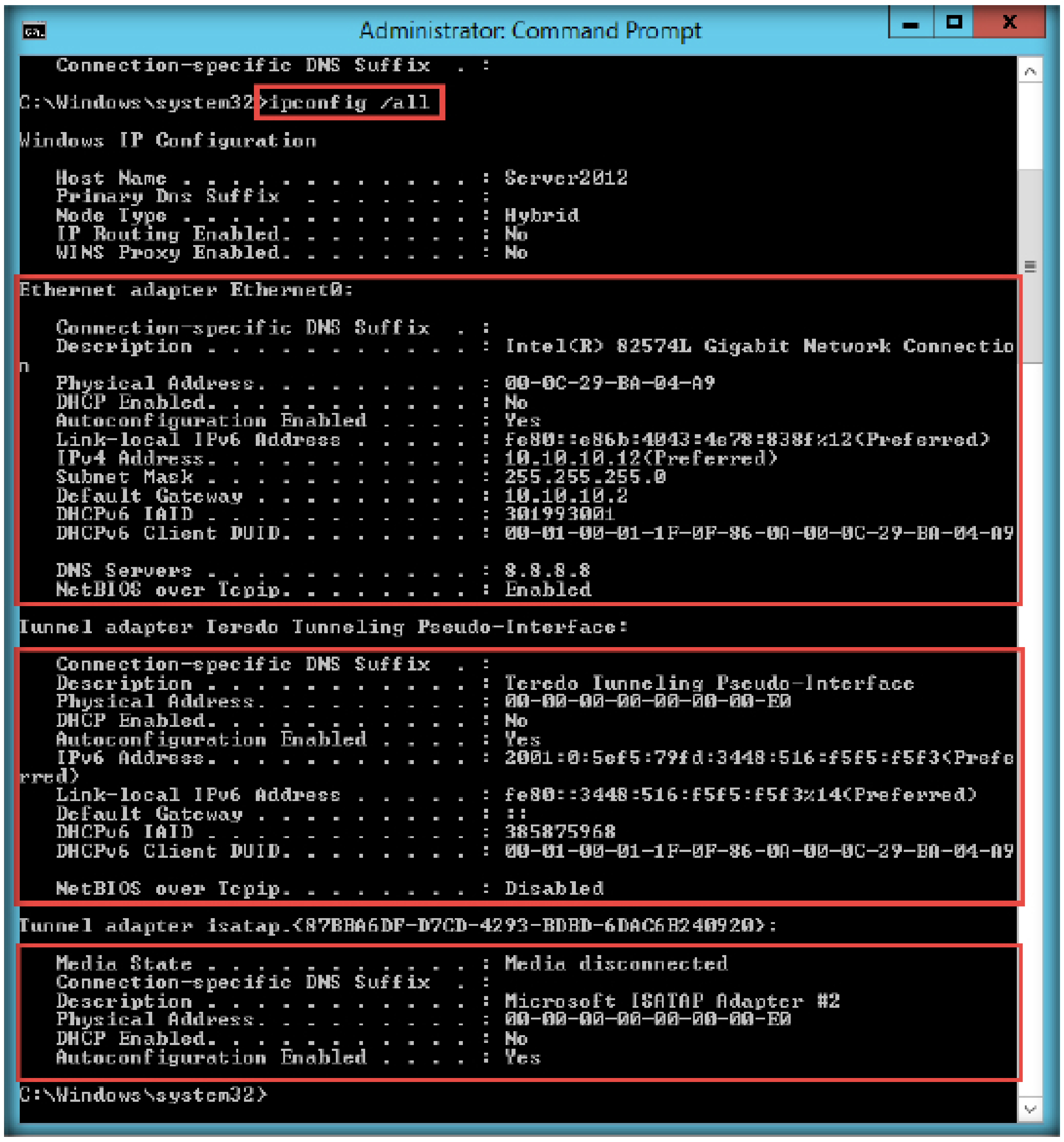
```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Windows\system32>ipconfig

Windows IP Configuration


Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::e86b:4043:4e78:838f%12
   IPv4 Address. . . . . . . . . . . : 10.10.10.12
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.10.10.2
Tunnel adapter Teredo Tunneling Pseudo-Interface:

   Connection-specific DNS Suffix  . :
   IPv6 Address. . . . . . . . . . . : 2001:0:5ef5:79fd:3448:516:f5f5:f5f3
   Link-local IPv6 Address . . . . . : fe80::3448:516:f5f5:f5f3%14
   Default Gateway . . . . . . . . . : ::

Tunnel adapter isatap.{87BBA6DF-D7CD-4293-BDBD-6DAC6B240920}:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

C:\Windows\system32>
```

FIGURE 1.6: Checking IP Configuration

6. You can use different ipconfig parameters to perform various network troubleshooting activities.

ipconfig Parameters	
/all	Displays the full TCP/IP configuration for all adapters.
/renew [Adapter]	Renews DHCP configuration for all adapters
/release [Adapter]	Sends a DHCPRELEASE message to the DHCP server to release the current DHCP configuration and discard the IP address configuration for either all adapters (if an adapter is not specified) or for a specific adapter
/flushdns	Flushes and resets the contents of the DNS client resolver cache.
/displaydns	Displays the contents of the DNS client resolver cache, which includes both entries preloaded from the local Hosts file and any recently obtained resource records for name queries resolved by the computer.
/registerdns	Initiates manual dynamic registration for the DNS names and IP addresses that are configured at a computer.
/showclassid Adapter	Displays the DHCP class ID for a specified adapter.
/setclassid Adapter [ClassID]	Configures the DHCP class ID for a specified adapter.
/?	Displays help at the command prompt.

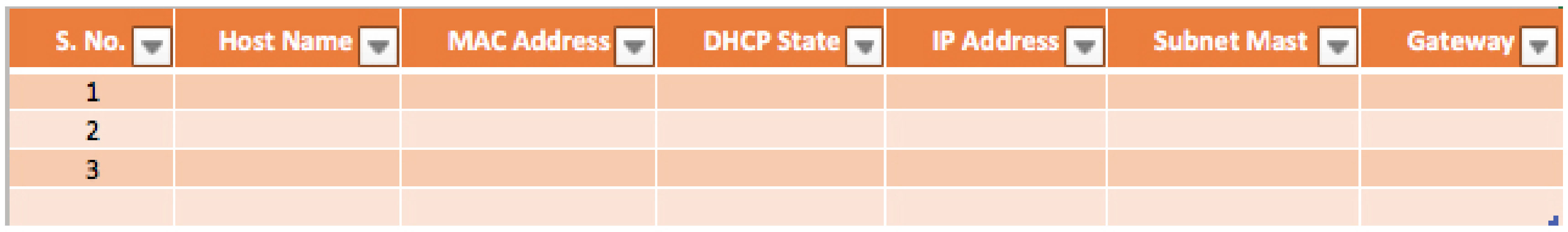
7. Now, type **ipconfig /all** and press **Enter**. This command will list out the System's IP configuration, host name, Ethernet Adapter installed and its MAC Address (Physical Address) and so on, as shown in the screenshot.



```
   Connection-specific DNS Suffix  . :

C:\Windows\system32>ipconfig /all

Windows IP Configuration

   Host Name . . . . . . . . . . . . : Server2012
   Primary Dns Suffix  . . . . . . . :
   Node Type . . . . . . . . . . . . : Hybrid
   IP Routing Enabled. . . . . . . . : No
   WINS Proxy Enabled. . . . . . . . : No

Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Intel(R) 82574L Gigabit Network Connectio
n
   Physical Address. . . . . . . . . : 00-0C-29-BA-04-A9
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::e86b:4043:4e78:838f%12(Preferred)
   IPv4 Address. . . . . . . . . . . : 10.10.10.12(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.10.10.2
   DHCPv6 IAID . . . . . . . . . . . : 301993001
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-1F-0F-86-0A-00-0C-29-BA-04-A9

   DNS Servers . . . . . . . . . . . : 8.8.8.8
   NetBIOS over Tcpip. . . . . . . . : Enabled

Tunnel adapter Teredo Tunneling Pseudo-Interface:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Teredo Tunneling Pseudo-Interface
   Physical Address. . . . . . . . . : 00-00-00-00-00-00-00-E0
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes
   IPv6 Address. . . . . . . . . . . : 2001:0:5ef5:79fd:3448:516:f5f5:f5f3(Prefe
rred)
   Link-local IPv6 Address . . . . . : fe80::3448:516:f5f5:f5f3%14(Preferred)
   Default Gateway . . . . . . . . . : ::
   DHCPv6 IAID . . . . . . . . . . . : 385875968
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-1F-0F-86-0A-00-0C-29-BA-04-A9

   NetBIOS over Tcpip. . . . . . . . : Disabled

Tunnel adapter isatap.{87BBA6DF-D7CD-4293-BDBD-6DAC6B240920}:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Microsoft ISATAP Adapter #2
   Physical Address. . . . . . . . . : 00-00-00-00-00-00-00-E0
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes
C:\Windows\system32>
```

FIGURE 1.7: Complete IP Configuration

8. You can use the information obtained from the above steps to create an Inventory List of all the computing devices in the network. In later modules we will look at better and more sophisticated techniques to create a Network Inventory but this could be an ideal starting point.



S. No.	Host Name	MAC Address	DHCP State	IP Address	Subnet Mast	Gateway
1						
2						
3						

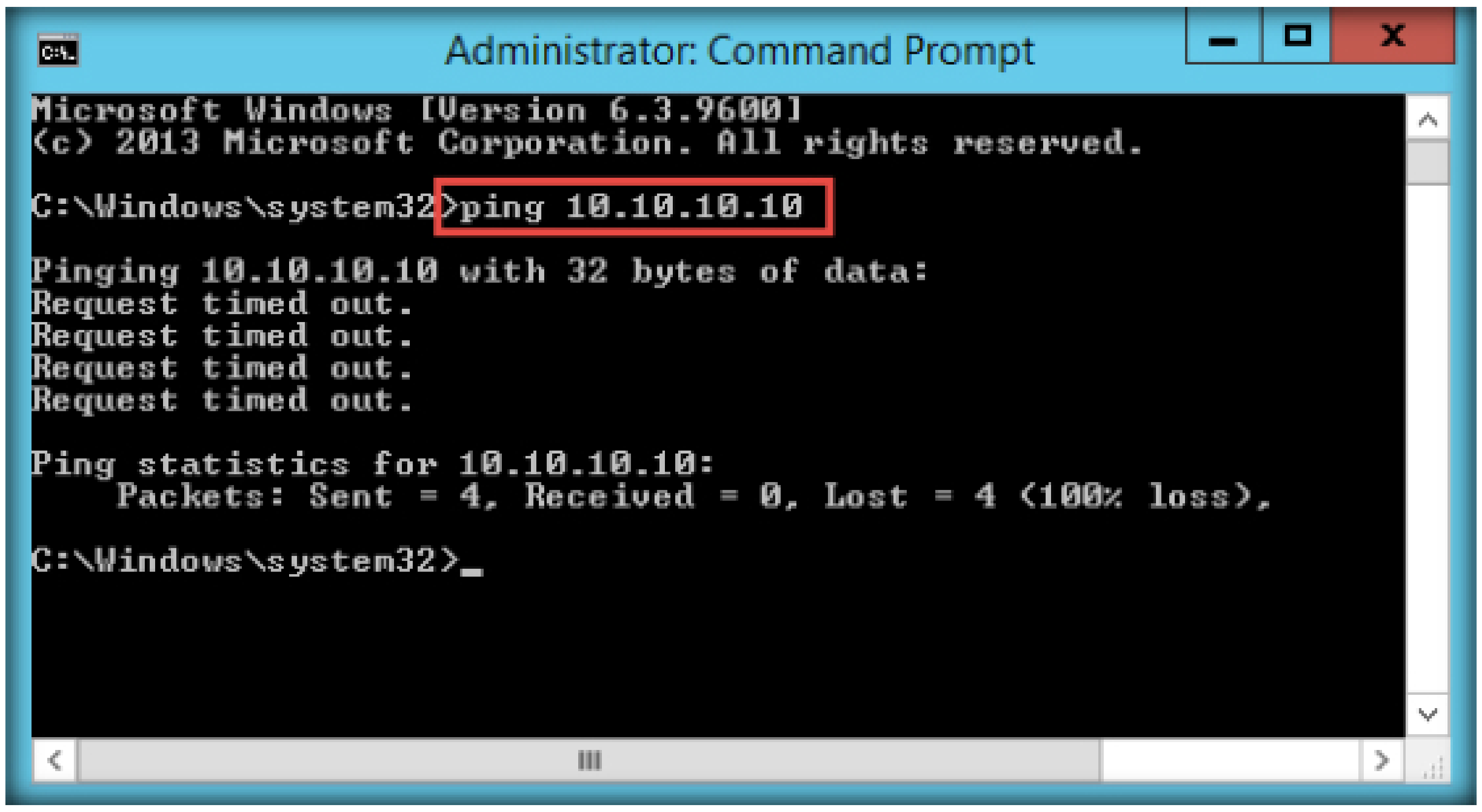
9. Close the command prompt after noting down all the information.

TASK 2

Checking IP level Connectivity Using Ping command

10. Now, we will explore the usage of the **Ping** command. Network administrators always encounter IP level Connectivity errors in the network such as **Request timed out**, **Destination host unreachable**, etc. With the help of the Ping command, they can ensure the reachability of a host to other hosts connected in the network

11. Open a command prompt in the Admin mode by right-clicking on the **Start** icon and then clicking on **Command Prompt (Admin)** from the context menu. Type **ping** followed by the IP address of the Windows 10 machine (it is 10.10.10.10 for this lab setup)



```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Windows\system32>ping 10.10.10.10

Pinging 10.10.10.10 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 10.10.10.10:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\Windows\system32>_
```

FIGURE 1.8: Demonstration of the Ping command

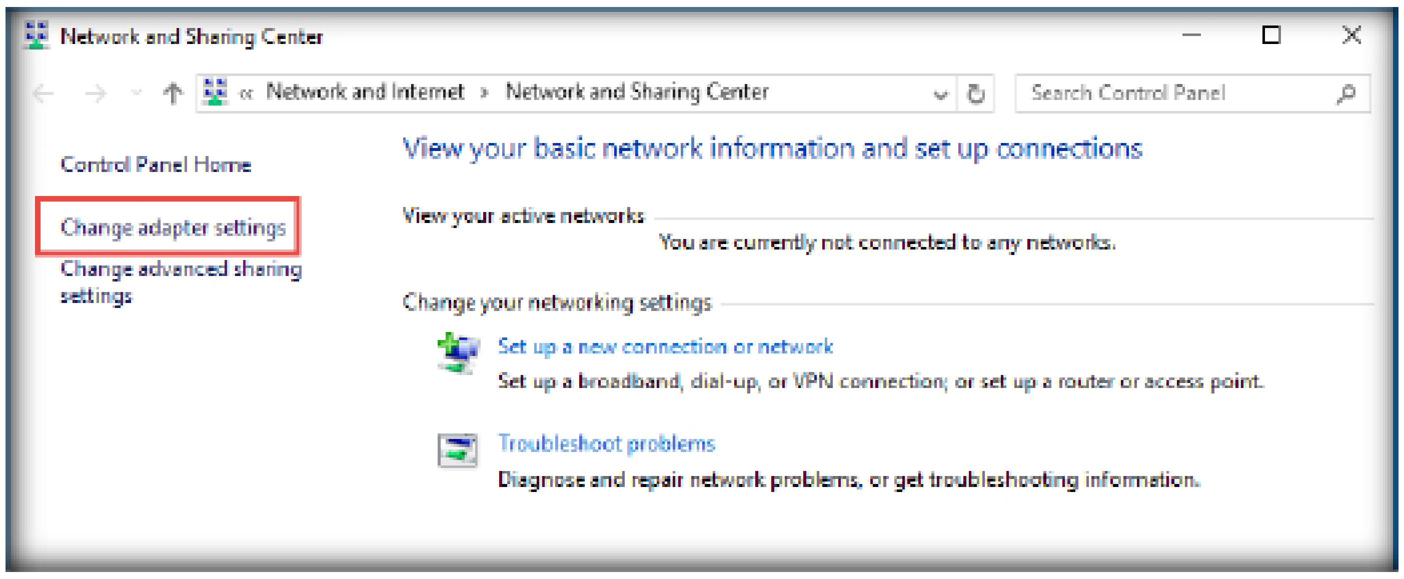
12. You can see that the “**Request timed out**” error. It means that the target system did not reply within the stipulated time frame. It implies that the target device is out of reach. The cause of this is either due to the target machine is turned off or the Network adapter is disabled on the target machine.

Option	Use
-n *Count*	Determines the number of echo requests to send. The default is 4 requests.
-w *Timeout*	Enables you to adjust the time-out (in milliseconds). The default is 1,000 (a 1-second time-out).
-l *Size*	Enables you to adjust the size of the ping packet. The default size is 32 bytes.
-f	Sets the Do Not Fragment bit on the ping packet. By default, the ping packet allows fragmentation.

13. Now, switch to the Windows 10 machine to troubleshoot the issue.

14. Go to **Control Panel -> Network and Internet -> Network and Sharing Center**. Check for the Network adapter status

15. Now you can see that Ethernet 2 adapter is showing up “No internet access”. Click on **Change adapter settings** in the left pane


Network and Sharing Center
« Network and Internet › Network and Sharing Center
Search Control Panel
Control Panel Home
Change adapter settings
Change advanced sharing settings
Set up a new connection or network
Troubleshoot problems

View your basic network information and set up connections

View your active networks

You are currently not connected to any networks.

Change your networking settings

Set up a broadband, dial-up, or VPN connection; or set up a router or access point.

Diagnose and repair network problems, or get troubleshooting information.

FIGURE 1.9: Ethernet 2 Network adapter error

16. Now you can see that the **Ethernet 2** adapter is disabled.

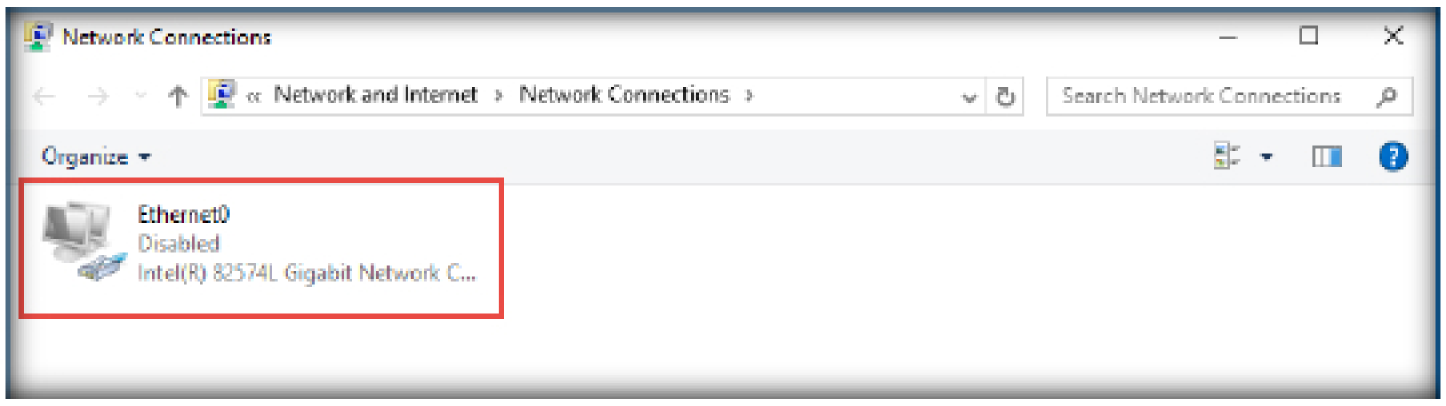

Network Connections
« Network and Internet › Network Connections ›
Search Network Connections
Organize ▾
Ethernet0
Disabled
Intel(R) 82574L Gigabit Network C...

FIGURE 1.10: Disabled Ethernet adapter

17. Right click on it and select **Enable** from the context menu.

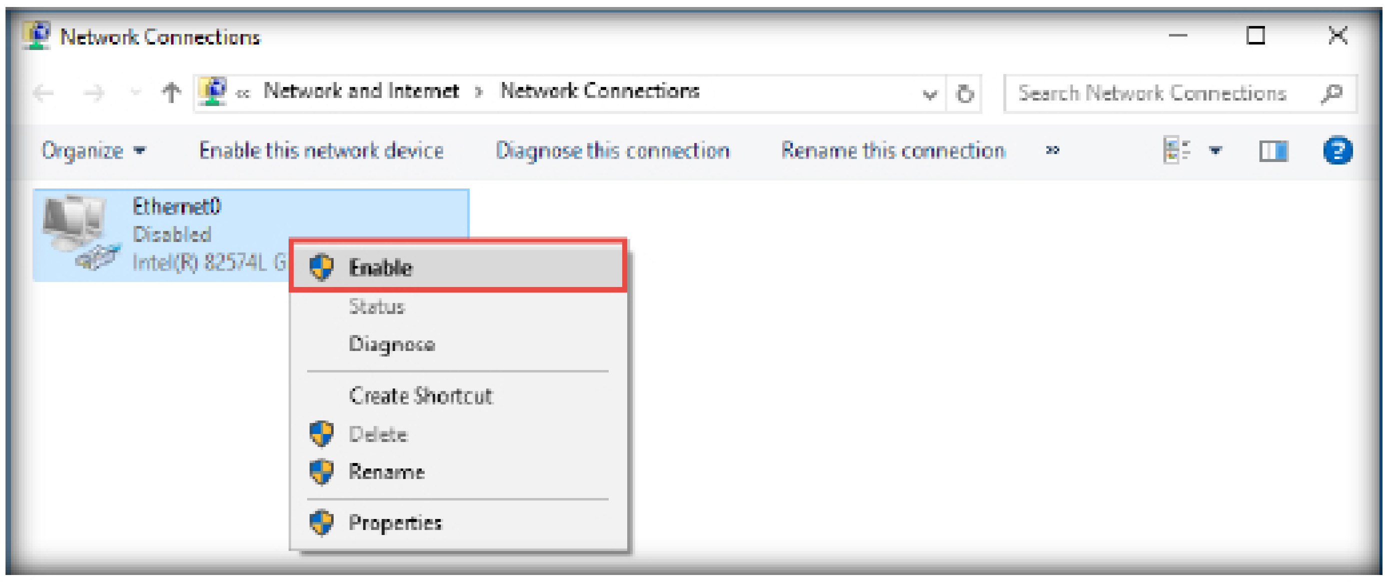
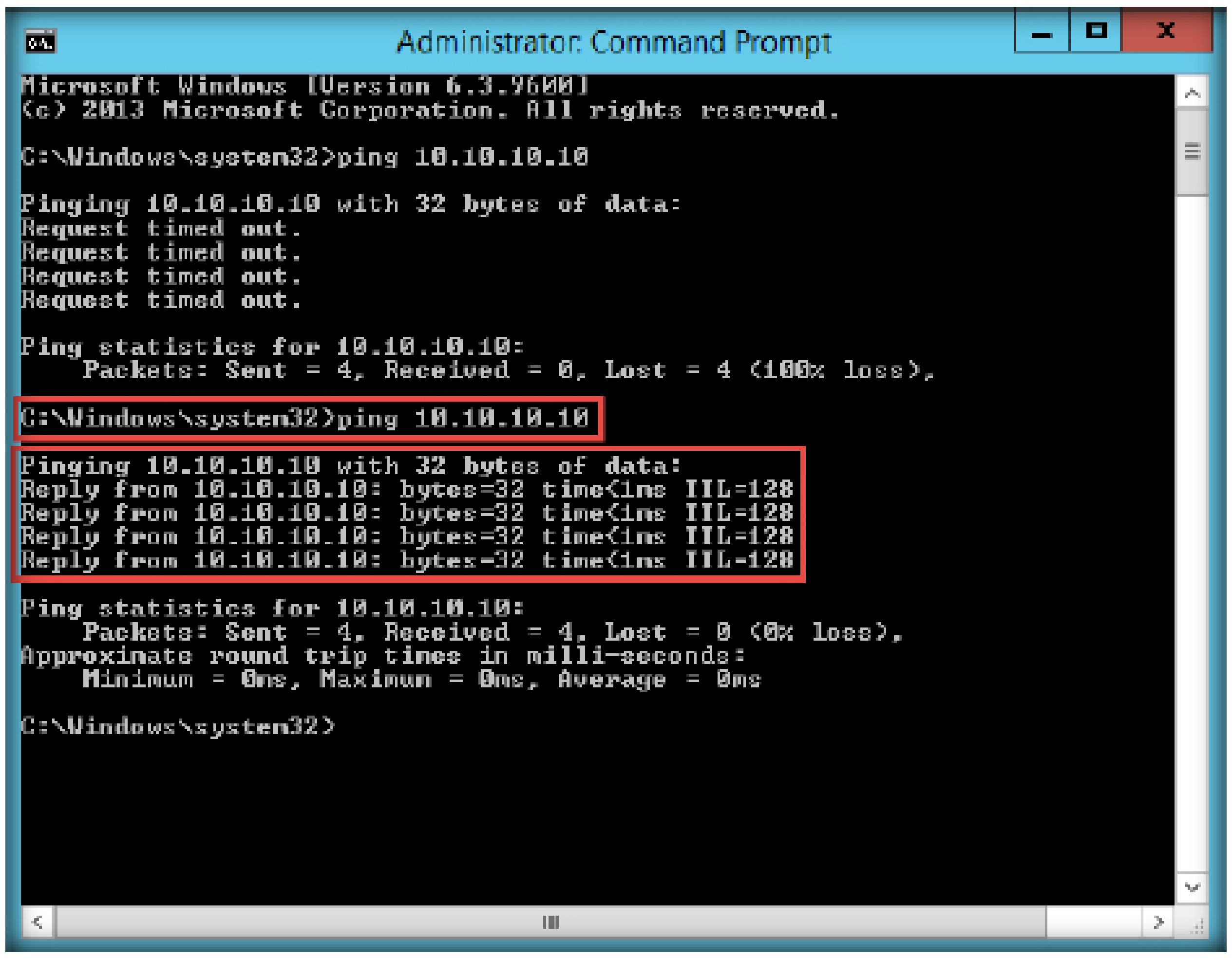


FIGURE 1.11: Enabling the disabled adapter

18. Now, switch back to **Windows Server 2012** machine and ping the target machine again



```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Windows\system32>ping 10.10.10.10

Pinging 10.10.10.10 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 10.10.10.10:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\Windows\system32>ping 10.10.10.10

Pinging 10.10.10.10 with 32 bytes of data:
Reply from 10.10.10.10: bytes=32 time<1ms TTL=128
Reply from 10.10.10.10: bytes=32 time<1ms TTL=128
Reply from 10.10.10.10: bytes=32 time<1ms TTL=128
Reply from 10.10.10.10: bytes=32 time<1ms TTL=128

Ping statistics for 10.10.10.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Windows\system32>
```

FIGURE 1.12: Ping request successfully executed

19. This time, you will be able to ping Window 10 machine successfully.

Note: Sometimes even after enabling the adapter, the ping request might not be successful due to firewall restrictions. In such cases, you need to temporarily disable the firewall on the target machine to check its reachability

TASK 3

Tracing the route of packets using tracert command

20. Now, we will see the usage of the **tracert** command to know the number of hops between a source and a destination node in a network. **tracert** is useful for troubleshooting large networks where several paths can lead to the same point or where many intermediate components (routers or bridges) are involved.

    About tracert:

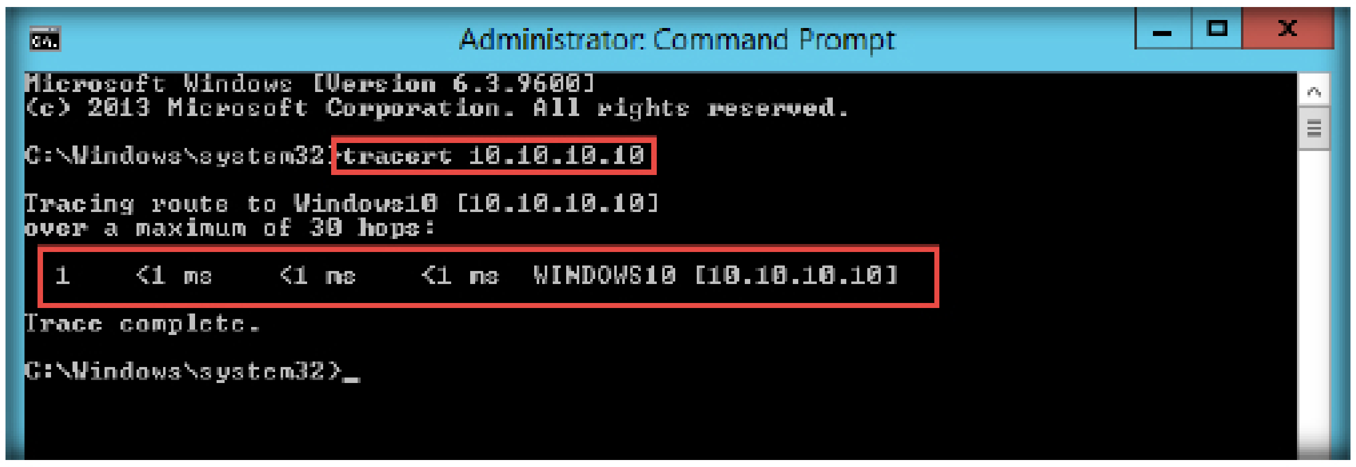
    Source: *https://support.microsoft.com*

    The **tracert** diagnostic utility determines the route to a destination by sending Internet Control Message Protocol (ICMP) echo packets to the destination. In these packets, **tracert** uses varying IP Time-To-Live (TTL) values. Because each router along the path is required to decrement the packet's TTL by at least 1 before forwarding the packet, the TTL is effectively a hop counter. When the TTL on a packet reaches zero (0), the router sends an ICMP "Time Exceeded" message back to the source computer.

    tracert sends the first echo packet with a TTL of 1 and increments the TTL by 1 on each subsequent transmission, until the destination responds or until the maximum TTL is reached. The ICMP "Time Exceeded" messages that intermediate routers send back show the route. Note however that some routers silently drop packets that have expired TTLs, and these packets are invisible to **tracert**.

    tracert prints out an ordered list of the intermediate routers that return ICMP "Time Exceeded" messages. Using the **-d** option with the tracert command instructs **tracert** not to perform a DNS lookup on each IP address, so that tracert reports the IP address of the near-side interface of the routers.

21. On the Windows Server 2012 machine. Open a command prompt in the Admin mode by right-clicking on **Start** icon and then clicking **Command Prompt (Admin)** from the context menu. Type **tracert** followed by the target system IP address the command prompt and press Enter.



```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Windows\system32>tracert 10.10.10.10

Tracing route to Windows10 [10.10.10.10]
over a maximum of 30 hops:

  1    <1 ms    <1 ms    <1 ms  WINDOWS10 [10.10.10.10]

Trace complete.

C:\Windows\system32>_
```

FIGURE 1.13: Demonstration of Tracert command

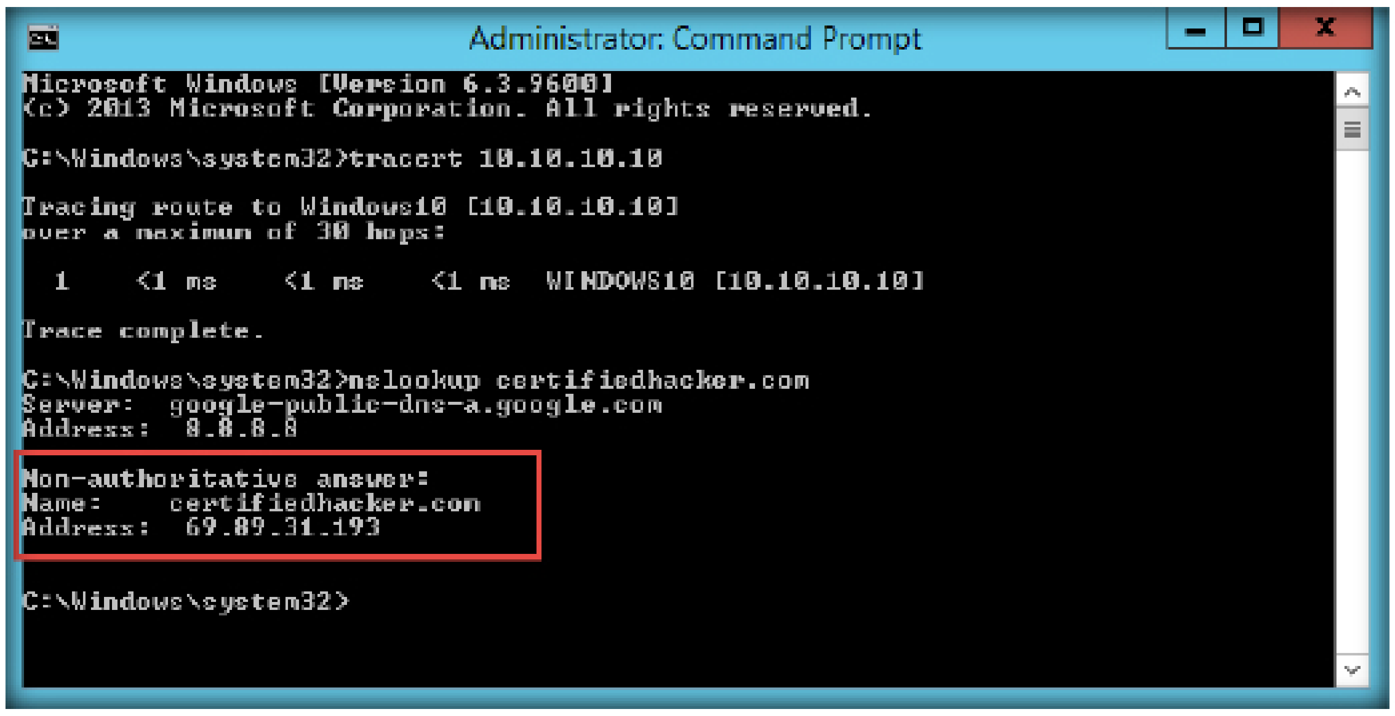
22. From the above screenshot, we can see that the destination was reached in the first hop itself.

TASK 4

Resolving Domain names with Using nslookup command

23. Now we will demonstrate the use of **nslookup** command. Nslookup stands for name server lookup. It is used to query a DNS server to obtain its domain name and associated IP address. It can be used with the domain name as an argument or independently

24. On the Windows Server 2012 machine, type **nslookup** followed by the domain name which you want to resolve (here, certifiedhacker.com) in the command prompt and press Enter.

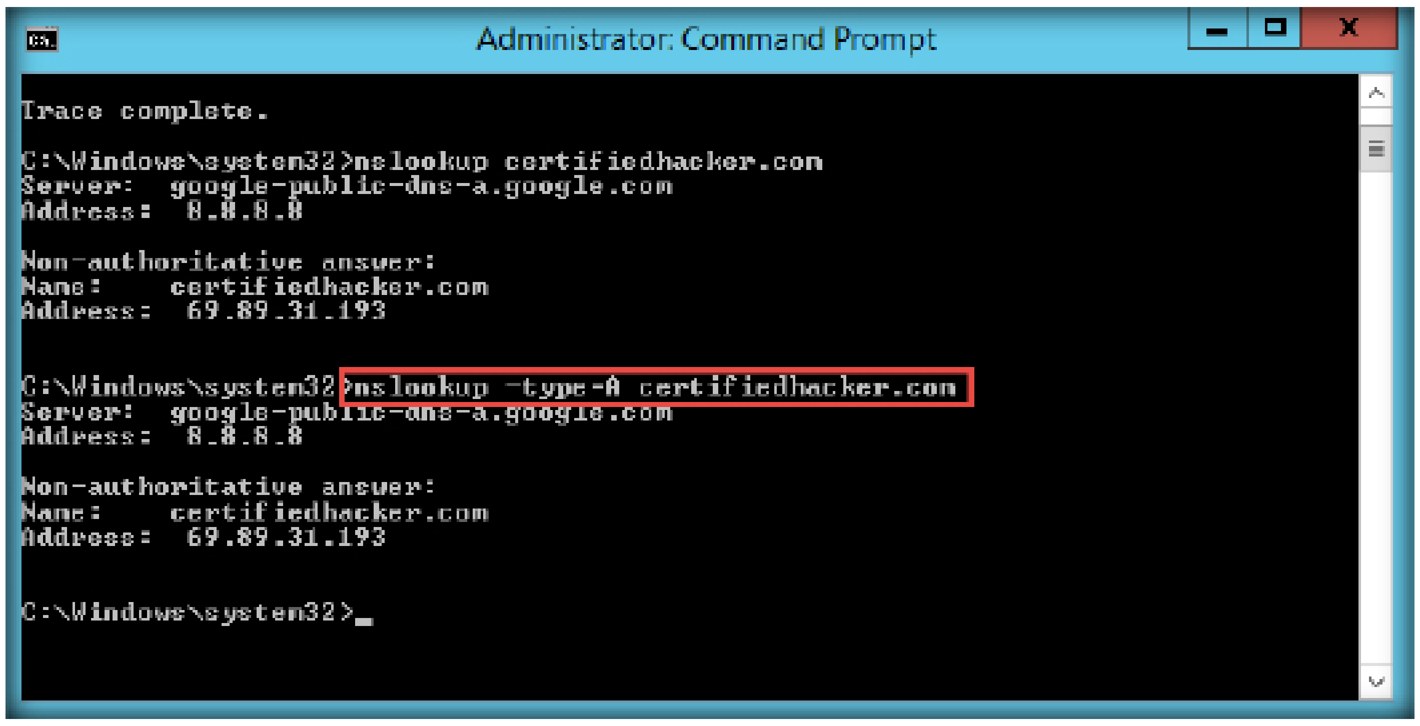


```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Windows\system32>tracert 10.10.10.10

Tracing route to Windows10 [10.10.10.10]
over a maximum of 30 hops:

  1    <1 ms    <1 ms    <1 ms  WINDOWS10 [10.10.10.10]

Trace complete.

C:\Windows\system32>nslookup certifiedhacker.com
Server:  google-public-dns-a.google.com
Address:  8.8.8.8

Non-authoritative answer:
Name:    certifiedhacker.com
Address:  69.89.31.193


C:\Windows\system32>
```

FIGURE 1.14: Demonstration of nslookup command

25. From the above screenshot, you will see that the domain name (certifiedhacker.com) resolves to its corresponding IP address (69.89.31.193)

26. You can also use the nslookup command with type parameters to get non-authoritative name server (NS) information as shown in the screenshot below:

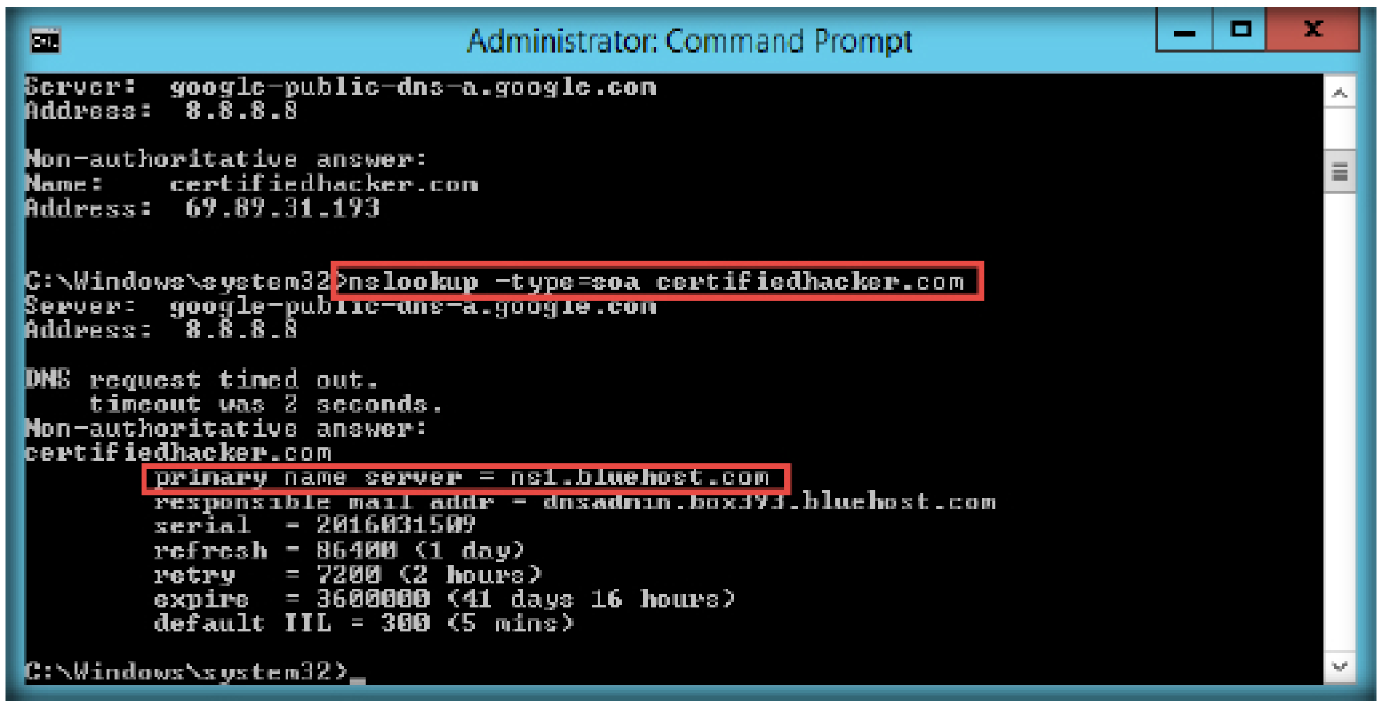


```
Trace complete.

C:\Windows\system32>nslookup certifiedhacker.com
Server:  google-public-dns-a.google.com
Address:  8.8.8.8

Non-authoritative answer:
Name:    certifiedhacker.com
Address:  69.89.31.193


C:\Windows\system32>nslookup -type-A certifiedhacker.com
Server:  google-public-dns-a.google.com
Address:  8.8.8.8

Non-authoritative answer:
Name:    certifiedhacker.com
Address:  69.89.31.193


C:\Windows\system32>_
```

FIGURE 1.15: nslookup command with type parameter

An Authoritative or Primary Nameserver is a nameserver (DNS Server) that holds the actual DNS records (A, CNAME, PTR, etc) for a particular domain/ address.

27. To get an authoritative NS information, you can use –type=soa parameter with nslookup.



```
Server:  google-public-dns-a.google.com
Address:  8.8.8.8

Non-authoritative answer:
Name:    certifiedhacker.com
Address:  69.89.31.193


C:\Windows\system32>nslookup -type=soa certifiedhacker.com
Server:  google-public-dns-a.google.com
Address:  8.8.8.8

DNS request timed out.
    timeout was 2 seconds.
Non-authoritative answer:
certifiedhacker.com
        primary name server = ns1.bluehost.com
        responsible mail addr = dnsadmin.box393.bluehost.com
        serial  = 2016031509
        refresh = 86400 (1 day)
        retry   = 7200 (2 hours)
        expire  = 3600000 (41 days 16 hours)
        default TTL = 300 (5 mins)

C:\Windows\system32>_
```

FIGURE 1.16: nslookup command with type parameter

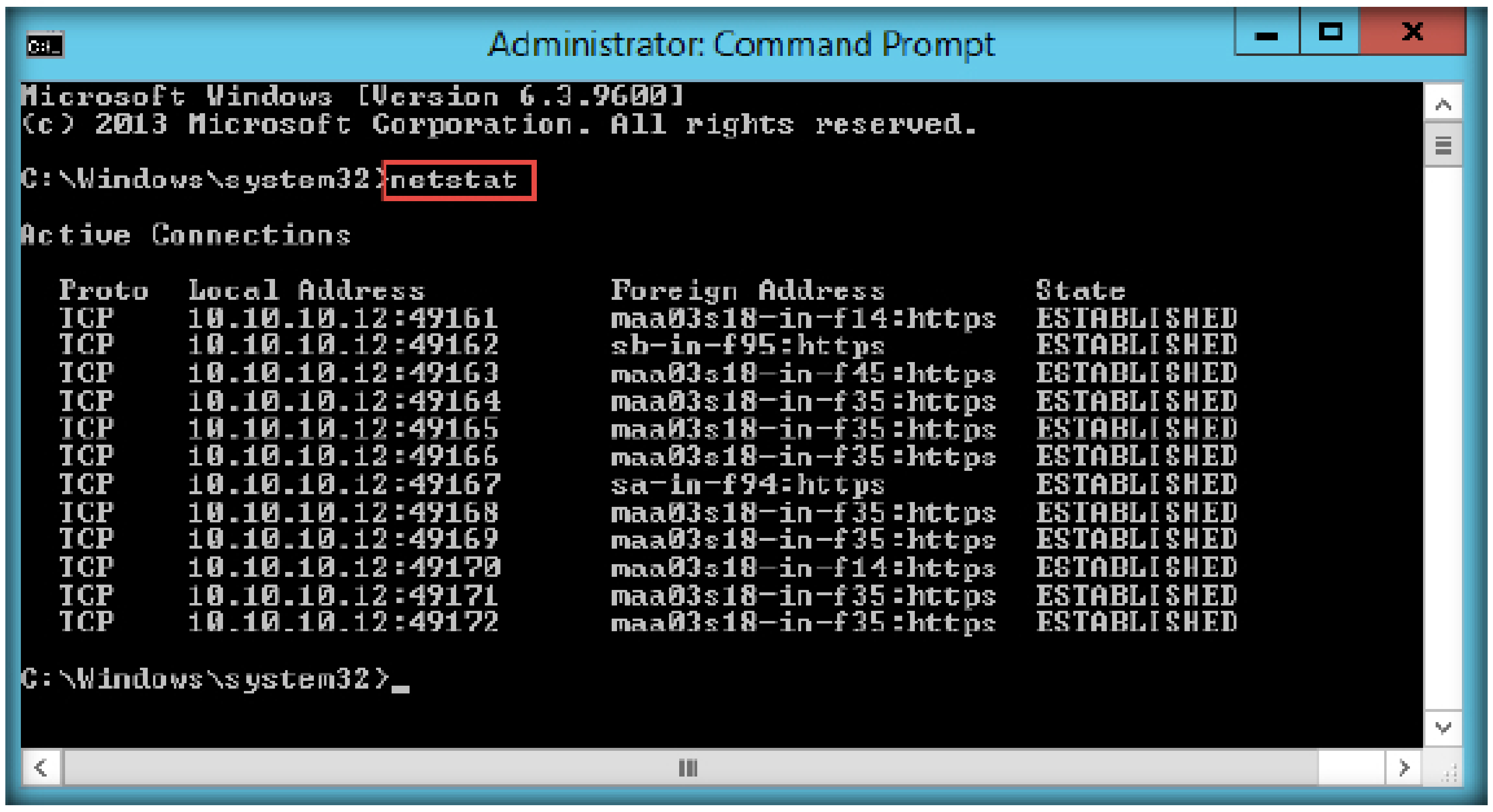
28. The address labelled as primary name server in the above screenshot is the DNS authority for the domain.

TASK 5

Checking your network configuration and statistics netstat command

29. Now we will see the use of the **netstat** command. Netstat stands for Network statistics. Displays active TCP connections, ports on which the computer is listening, Ethernet statistics, the IP routing table, IPv4 statistics (for the IP, ICMP, TCP, and UDP protocols), and IPv6 statistics (for the IPv6, ICMPv6, TCP over IPv6, and UDP over IPv6 protocols). Used without parameters, netstat displays active TCP connections.

30. Type the **netstat** command to check your network statistics as shown in following screenshot



```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Windows\system32>netstat

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    10.10.10.12:49161      maa03s18-in-f14:https  ESTABLISHED
  TCP    10.10.10.12:49162      sb-in-f95:https        ESTABLISHED
  TCP    10.10.10.12:49163      maa03s18-in-f45:https  ESTABLISHED
  TCP    10.10.10.12:49164      maa03s18-in-f35:https  ESTABLISHED
  TCP    10.10.10.12:49165      maa03s18-in-f35:https  ESTABLISHED
  TCP    10.10.10.12:49166      maa03s18-in-f35:https  ESTABLISHED
  TCP    10.10.10.12:49167      sa-in-f94:https        ESTABLISHED
  TCP    10.10.10.12:49168      maa03s18-in-f35:https  ESTABLISHED
  TCP    10.10.10.12:49169      maa03s18-in-f35:https  ESTABLISHED
  TCP    10.10.10.12:49170      maa03s18-in-f14:https  ESTABLISHED
  TCP    10.10.10.12:49171      maa03s18-in-f35:https  ESTABLISHED
  TCP    10.10.10.12:49172      maa03s18-in-f35:https  ESTABLISHED

C:\Windows\system32>_
```

FIGURE 1.17: Demonstration of netstat command

31. You can use different **nestat** parameters to obtain important connection information

Parameters	Use
-a	Displays all active TCP connections and the TCP and UDP ports on which the computer is listening.
-e	Displays Ethernet statistics, such as the number of bytes and packets sent and received. This parameter can be combined with –s

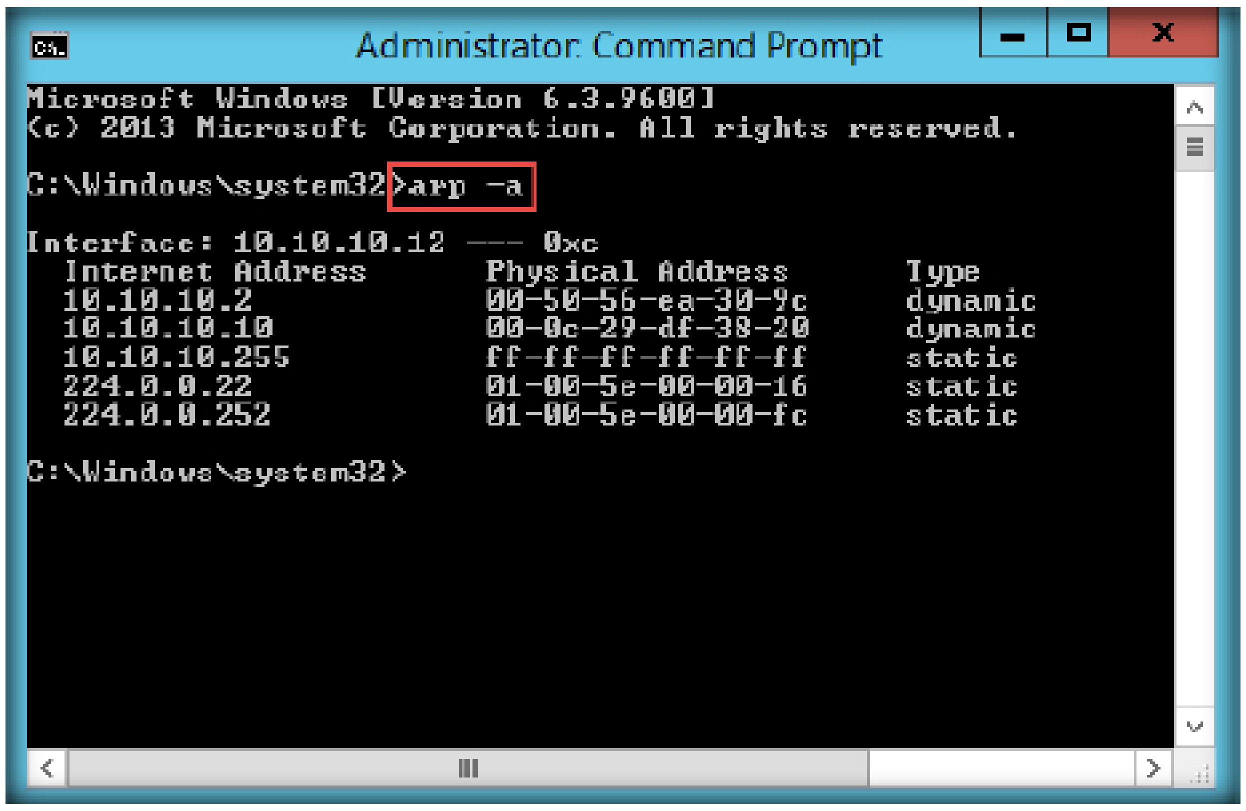
-n	Displays active TCP connections, however, addresses and port numbers are expressed numerically and no attempt is made to determine names.
-o	Displays active TCP connections and includes the process ID (PID) for each connection. You can find the application based on the PID on the **Processes** tab in Windows Task Manager. This parameter can be combined with **-a**, **-n**, and **-p**.
-p *Protocol*	Shows connections for the protocol specified by *Protocol*. In this case, the *Protocol* can be **tcp**, **udp**, **tcpv6**, or **udpv6**. If this parameter is used with **-s** to display statistics by protocol, *Protocol* can be **tcp**, **udp**, **icmp**, **ip**, **tcpv6**, **udpv6**, **icmpv6**, or **ipv6**.
-s	Displays statistics by protocol. By default, statistics are shown for the TCP, UDP, ICMP, and IP protocols. If the IPv6 protocol for Windows XP is installed, statistics are shown for the TCP over IPv6, UDP over IPv6, ICMPv6, and IPv6 protocols. The **-p** parameter can be used to specify a set of protocols.
-r	Displays the contents of the IP routing table. This is equivalent to the **route print** command.
Interval	Redisplays the selected information every *Interval* seconds. Press CTRL+C to stop the redisplay. If this parameter is omitted, **netstat** prints the selected information only once.
/?	Displays help at the command prompt.

TASK 6

Displaying Address Resolution Protocol (ARP) cache using arp command

32. The **arp –a** command displays ARP cache. The cache has a mapping of IP addresses with their respective MAC addresses. It has many options and if you use ARP without any option it displays the available options

33. Type **arp –a** command and press Enter to display the ARP cache entries.



```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Windows\system32>arp -a

Interface: 10.10.10.12 --- 0xc
  Internet Address      Physical Address      Type
  10.10.10.2            00-50-56-ea-30-9c     dynamic
  10.10.10.10           00-0c-29-df-38-20     dynamic
  10.10.10.255          ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.252           01-00-5e-00-00-fc     static

C:\Windows\system32>
```

FIGURE 1.18: Using arp –a command

Note: If you want to view the MAC address of only a particular IP address, type the IP address after **arp –a** command and press Enter.

Similarly, you can use the following useful commands for network administration and troubleshooting

Commands	Objectives
Gpresult	Starts the Operating System Group Policy Result tool
ipconfig /flushdns	Flushes the DNS resolver cache. Helpful when troubleshooting DNS name resolution problems
nbtstat -a <MachineName>	Obtains info from WINS or LMHOST (discovers who is logged on)
nbtstst -A <IP>	Gets info from WINS or LMHOST (discovers who is logged on)
nbtstat –R	Purges and reloads the remote cache name table
nbtstat –n	Lists local NetBIOS names.
nbtstat –r	Useful for detecting errors when browsing WINS or NetBIOS
netstat –ab	The **b** switch links each used port with its application
netstat –an	Shows open ports
netstat -an 1 | find "15868"	Locates only lines with the number 15868 and redisplays every one second
netstat -an | find "LISTENING"	Shows open ports with LISTENING status
net use	Retrieves a list of network connections
net user	Shows user account for the computer
net user /domain	Displays user accounts for the domain
net user /domain <UserName>	Shows account details for specific user
net group /domain	Shows group accounts for the domain
net view	Displays domains in the network
net view /domain	Specifies computers available in a specific domain
net view /domain: <DomainName> | more	Shows user accounts from specific domain
net view /cache	Shows workstation names
ping -a <IP>	Resolves IP to Hostname
ping -t <IP>	Pings host until stopped
Pathping	Displays the route and ping information when performing queries such as –n and –h options representing hostnames and maximum hops respectively.
set U	Shows which user is logged on
set L	Shows the logon server
telnet <IP> <port>	Confirms whether the port is open

Lab Analysis

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

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

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

Lab

2

Analyzing and Examining Various Network Packet Headers

We shall analyze various packets like TCP, HTTP, ICMP, DNS with Wireshark

ICON KEY

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

Lab Scenario

Each packet in a network contains control information and user data, which is also known as the payload. The control information contains data for delivering the payload, which includes source and destination IP and MAC addresses, sequencing information, etc. The header part of the packet stores this control information. So, being a network admin, you need to know how to examine the packet headers while examining the data packets.

Lab Objectives

The objective of this lab is to help students learn how to inspect TCP/IP packet header fields of different network packets.

Tools demonstrated in this lab are available in Z:\CND-Tools\CND Module 11 Network Traffic Monitoring and Analysis

Lab Environment

In this lab, you need:

- Wireshark, located at **Z:\CND-Tools\CND Module 11 Network Traffic Monitoring and Analysis\Packet Sniffing Tools\Wireshark**
- You can also download the latest version of Wireshark from the link https://www.wireshark.org/download.html
- If you decide to download the latest version, the screenshots shown in the lab may differ
- A virtual machine running Windows Server 2012
- A Web browser with Internet connection
- **Administrative** privileges to run tools

You can download Wireshark from http://www.wireshark.org.

Lab Duration

Time: 30 Minutes

Overview of Wireshark Packet Capture

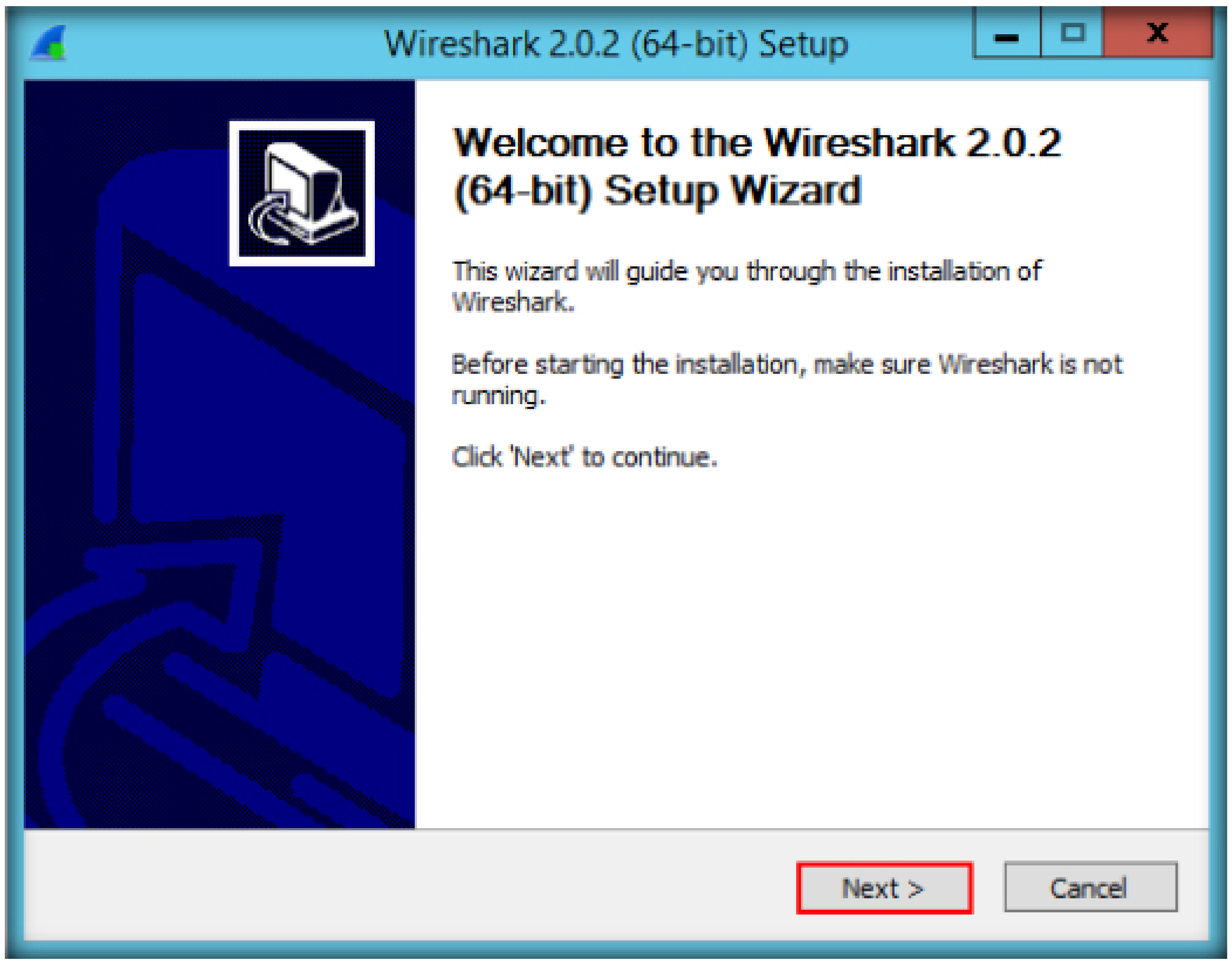
Packet capture is the intercepting of data packets traversing over a network using packet capture tools like Wireshark. These captured packets are analyzed in order to determine whether proper network security policies are being followed.

Lab Tasks

TASK 1

Installing and Launching Wireshark

1. Log on to **Windows Server 2012** virtual machine in Hyper-V Manager
2. Before beginning this lab, ensure that WinPcap is installed.
3. Navigate to **Z:\CND-Tools\CND Module 11 Network Traffic Monitoring and Analysis\Packet Sniffing Tools\Wireshark** and double-click **Wireshark-win64-2.0.2.exe**
4. If **Open File - Security Warning** pop-up appears, click **Run**.
5. Follow the wizard-driven installation steps to install Wireshark



Wireshark 2.0.2 (64-bit) Setup

Welcome to the Wireshark 2.0.2 (64-bit) Setup Wizard

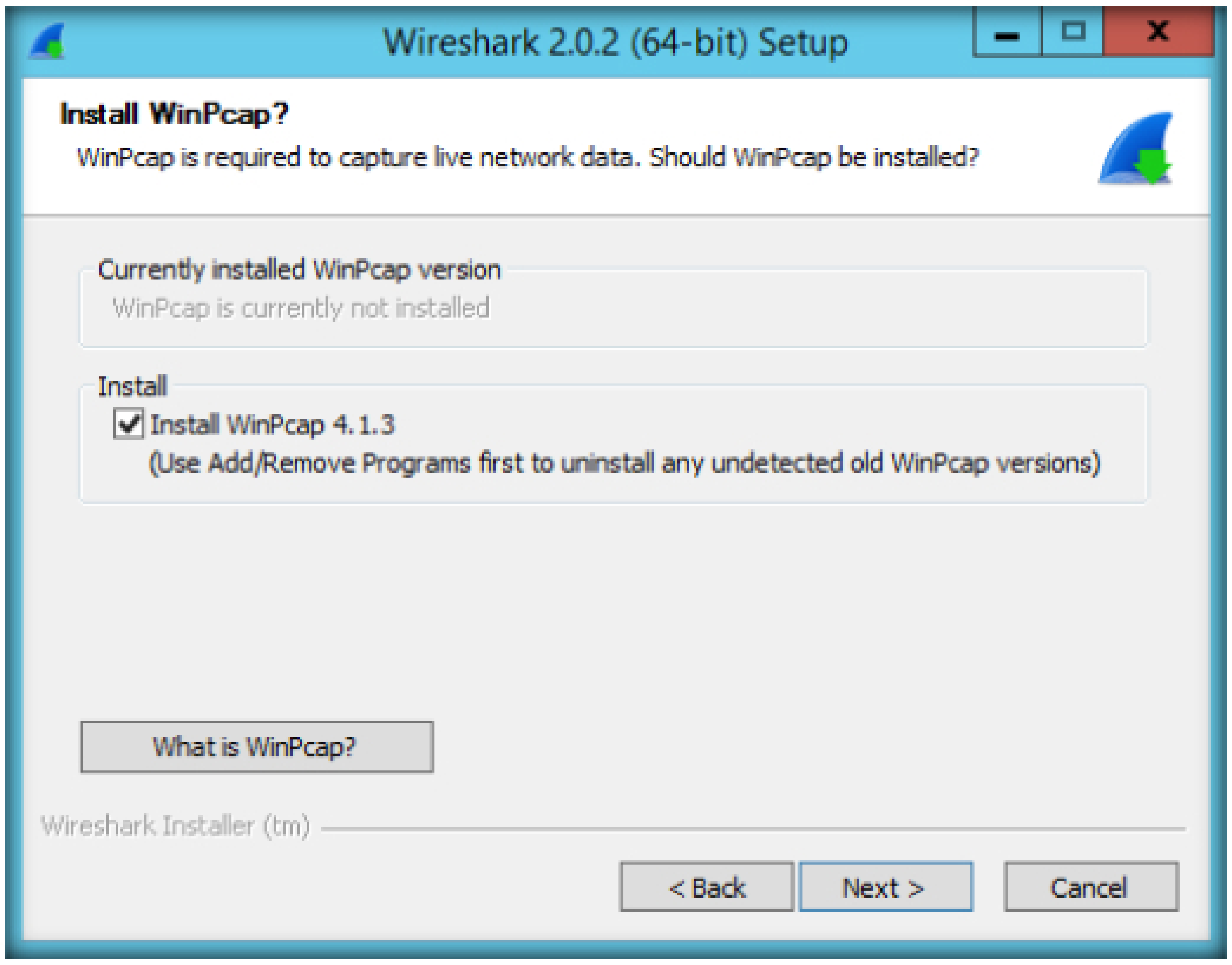
This wizard will guide you through the installation of Wireshark.

Before starting the installation, make sure Wireshark is not running.

Click 'Next' to continue.

FIGURE 2.1: Wireshark installation wizard

6. During the installation, a window appears asking you to install WinPcap. If you have already installed the application, click **Cancel**; else, click **Next** if you have not installed WinPcap.



Install WinPcap?

WinPcap is required to capture live network data. Should WinPcap be installed?

Currently installed WinPcap version

WinPcap is currently not installed

Install

☑ Install WinPcap 4.1.3

(Use Add/Remove Programs first to uninstall any undetected old WinPcap versions)

FIGURE 2.2: WinPcap installation wizard

7. On completing the installation, launch **Wireshark** from the **Apps** screen

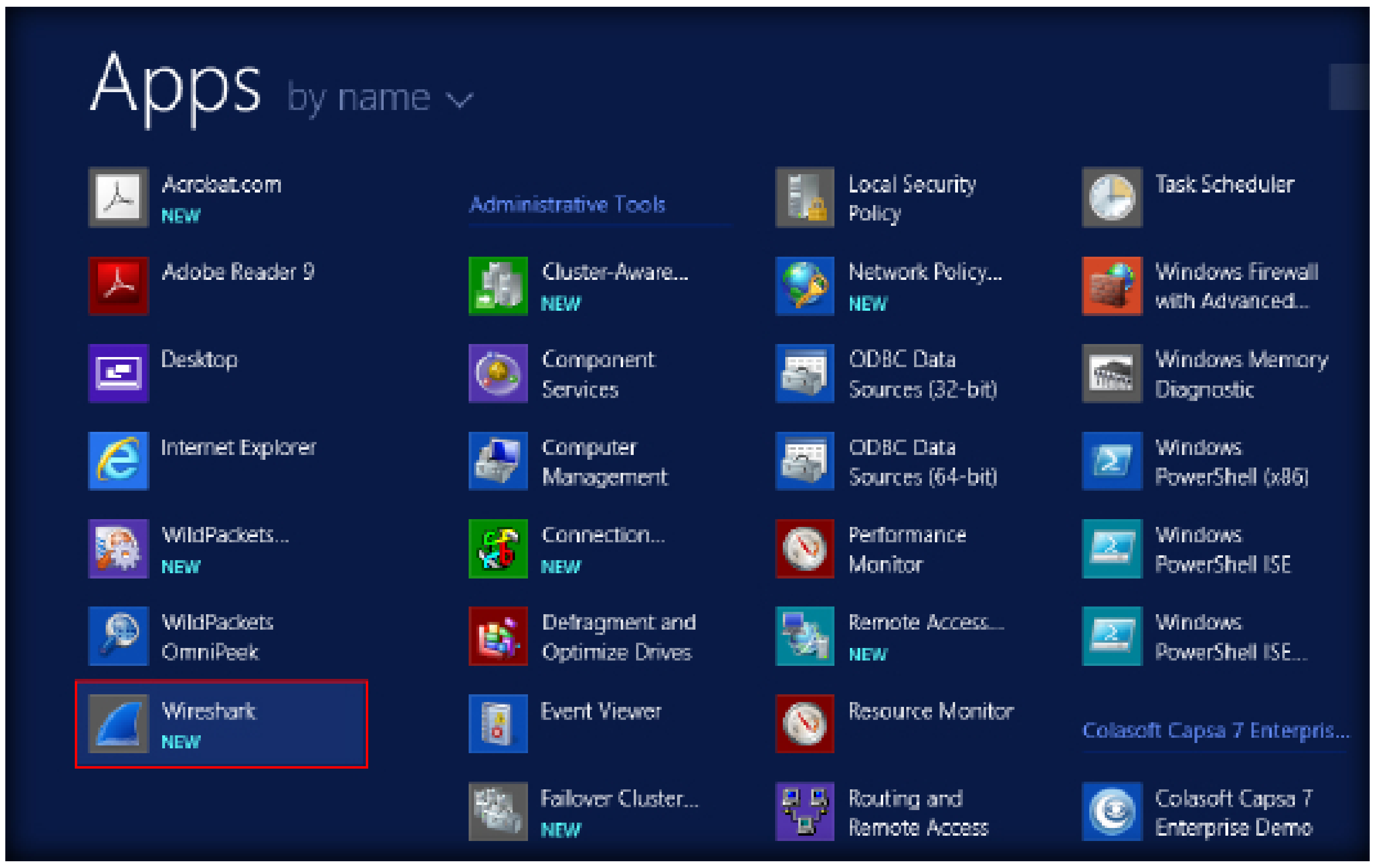


FIGURE 2.3: Windows Server 2012 Apps Screen

8. The main window of Wireshark appears as shown in following screenshot:

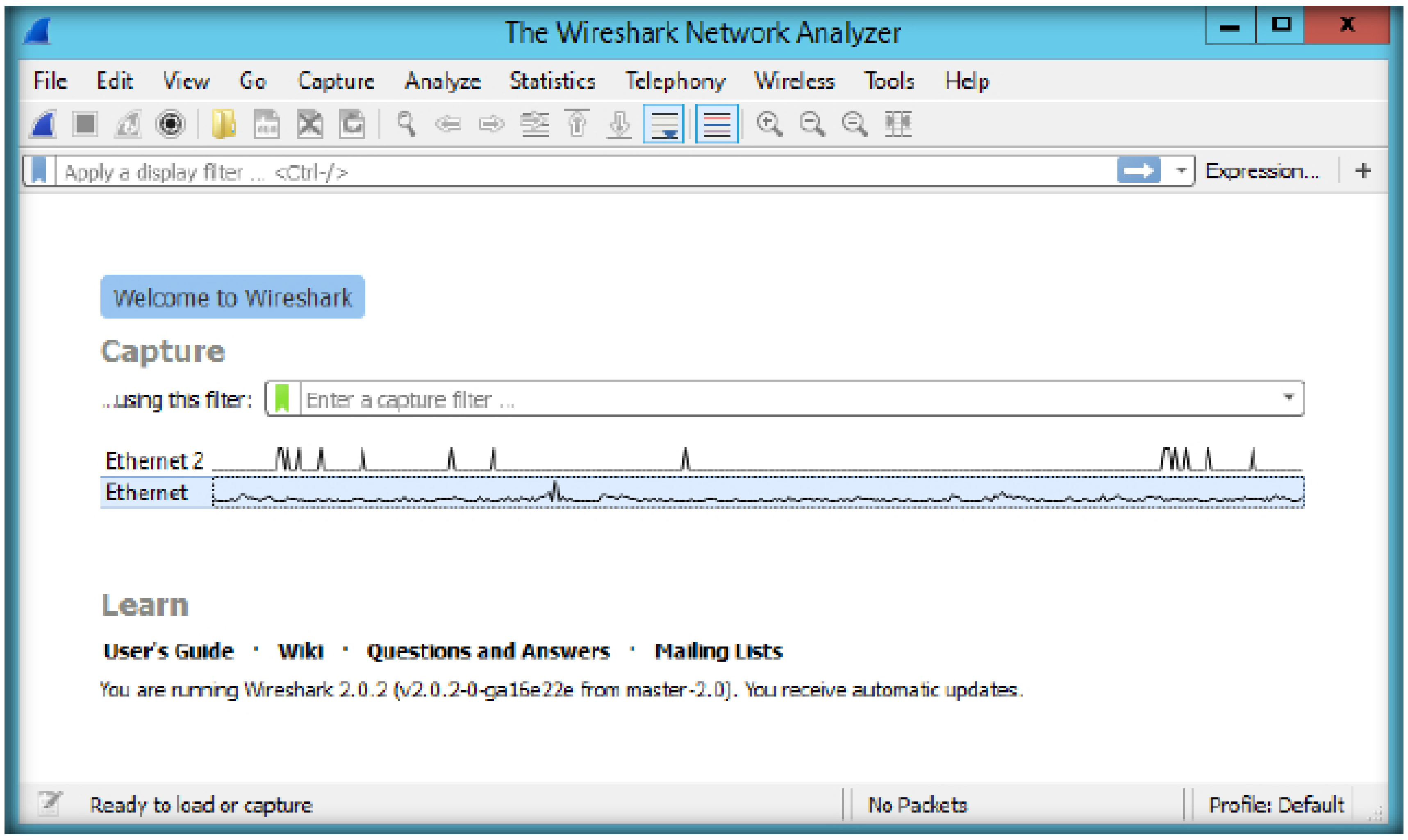

The Wireshark Network Analyzer
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Expression...
Welcome to Wireshark
Capture
...using this filter:
Enter a capture filter ...
Ethernet 2
Ethernet
Learn
User's Guide · Wiki · Questions and Answers · Mailing Lists
You are running Wireshark 2.0.2 (v2.0.2-0-ga16e22e from master-2.0). You receive automatic updates.
Ready to load or capture
No Packets
Profile: Default

FIGURE 2.4: Wireshark Window

9. Go to the **File** menu and click **Open**, to open a packet capture file

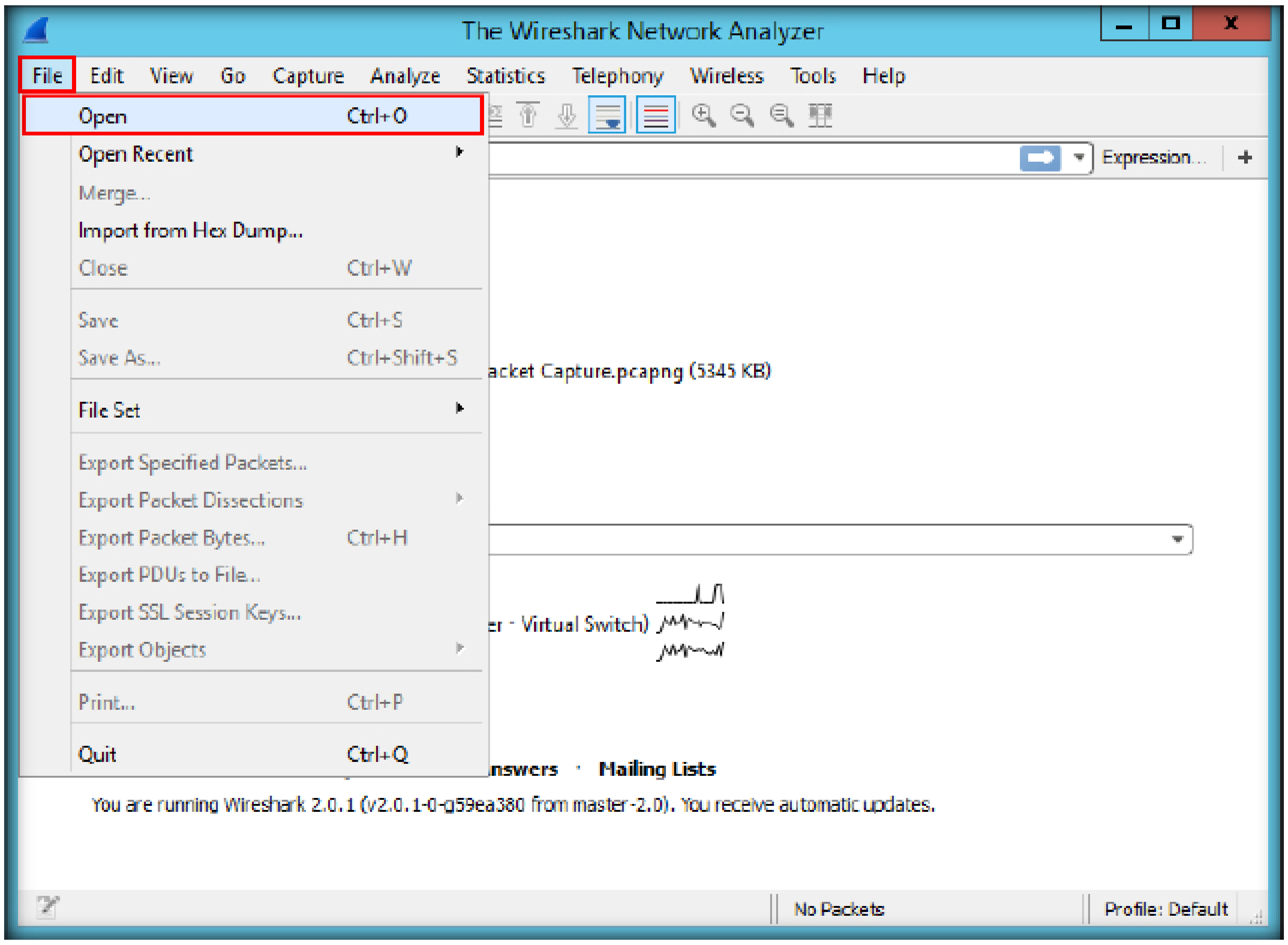
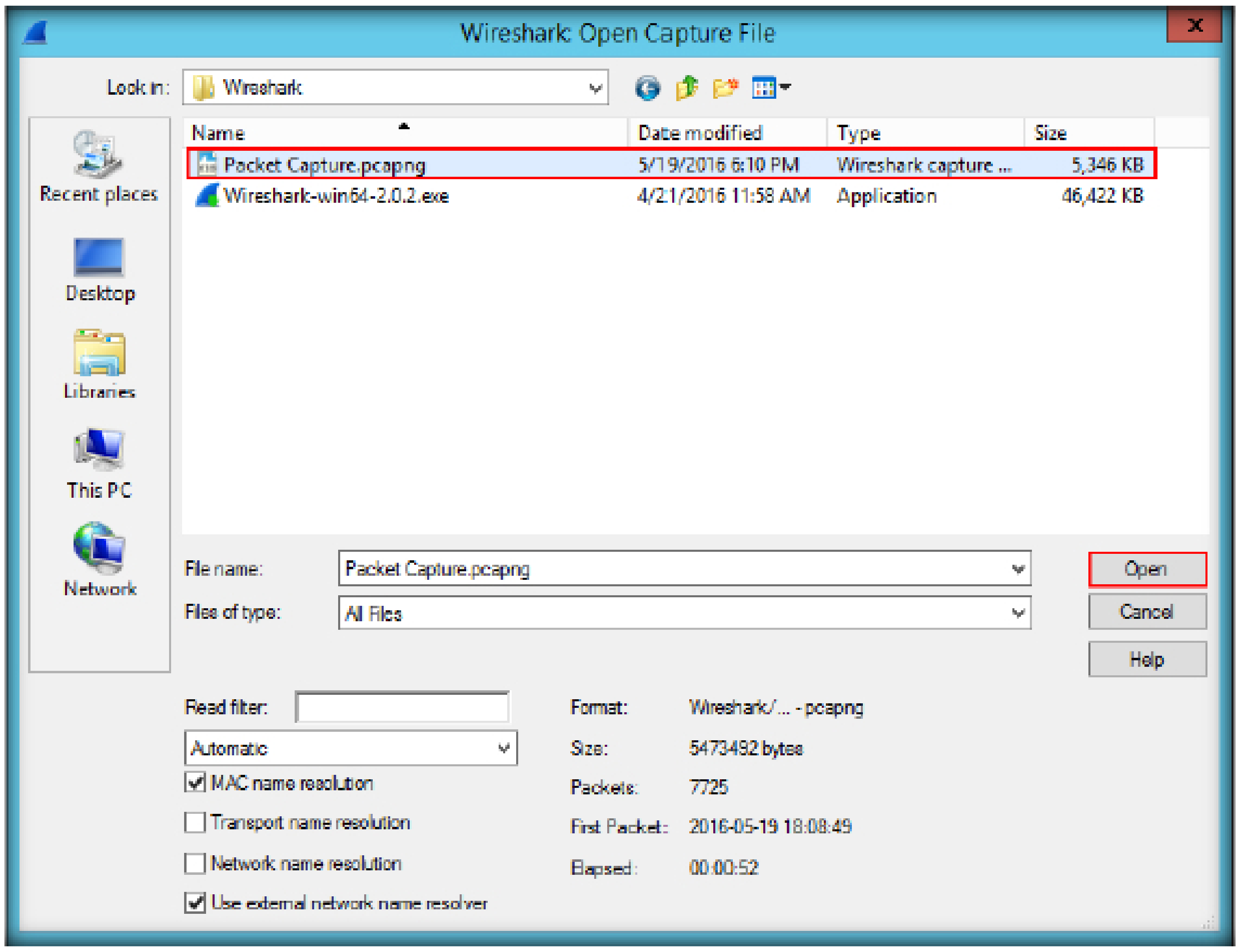

The Wireshark Network Analyzer
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Open Ctrl+O
Open Recent
Merge...
Import from Hex Dump...
Close Ctrl+W
Save Ctrl+S
Save As... Ctrl+Shift+S
File Set
Export Specified Packets...
Export Packet Dissections
Export Packet Bytes... Ctrl+H
Export PDUs to File...
Export SSL Session Keys...
Export Objects
Print... Ctrl+P
Quit Ctrl+Q
Expression...
acket Capture.pcapng (5345 KB)
er - Virtual Switch)
nswers · Mailing Lists
You are running Wireshark 2.0.1 (v2.0.1-0-g59ea380 from master-2.0). You receive automatic updates.
No Packets
Profile: Default

FIGURE 2.5: Opening a captured file

10. Wireshark: The Open Capture File window appears, navigate to **Z:\CND-Tools\CND Module 11 Network Traffic Monitoring and Analysis\Packet Sniffing Tools\Wireshark**, select **Packet Capture.pcapng** and click **Open**



Name	Date modified	Type	Size
Packet Capture.pcapng	5/19/2016 6:10 PM	Wireshark capture ...	5,346 KB
Wireshark-win64-2.0.2.exe	4/21/2016 11:58 AM	Application	46,422 KB

File name: Packet Capture.pcapng

Files of type: All Files

Read filter:

Automatic

☑ MAC name resolution

□ Transport name resolution

□ Network name resolution

☑ Use external network name resolver

Format: Wireshark/... - pcapng

Size: 5473492 bytes

Packets: 7725

First Packet: 2016-05-19 18:08:49

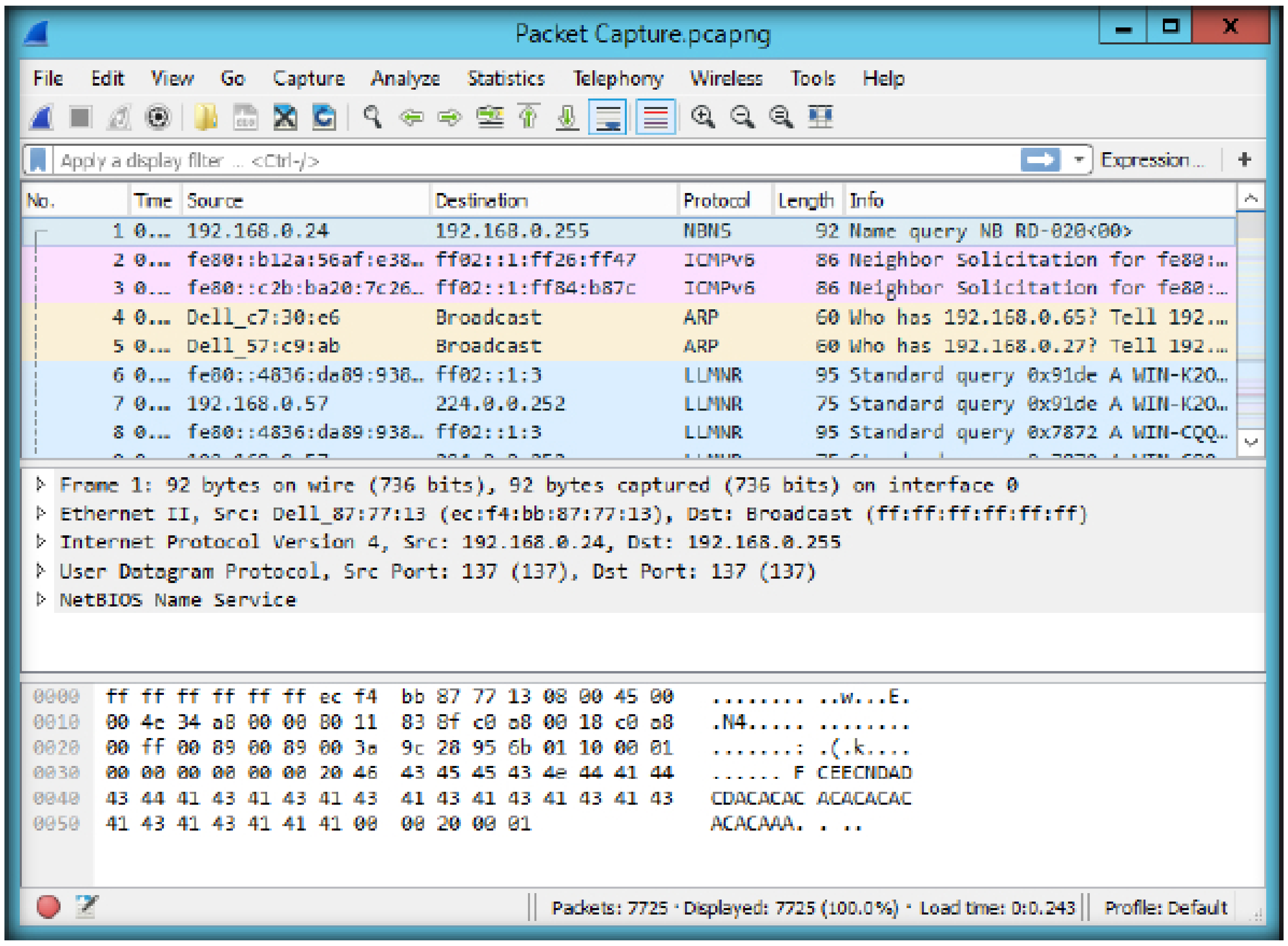
Elapsed: 00:00:52

FIGURE 2.6: Selecting the captured file

TASK 2

Opening a Packet Capture File

11. Wireshark displays the captured packets associated with the file as shown in the following screenshot:



```
No.     Time  Source                    Destination         Protocol  Length Info
      1 0...  192.168.0.24              192.168.0.255       NBNS          92 Name query NB RD-020<00>
      2 0...  fe80::b12a:56af:e38..     ff02::1:ff26:ff47   ICMPv6        86 Neighbor Solicitation for fe80:...
      3 0...  fe80::c2b:ba20:7c26..     ff02::1:ff84:b87c   ICMPv6        86 Neighbor Solicitation for fe80:...
      4 0...  Dell_c7:30:e6             Broadcast           ARP           60 Who has 192.168.0.65? Tell 192....
      5 0...  Dell_57:c9:ab             Broadcast           ARP           60 Who has 192.168.0.27? Tell 192....
      6 0...  fe80::4836:da89:938..     ff02::1:3           LLMNR         95 Standard query 0x91de A WIN-K20...
      7 0...  192.168.0.57              224.0.0.252         LLMNR         75 Standard query 0x91de A WIN-K20...
      8 0...  fe80::4836:da89:938..     ff02::1:3           LLMNR         95 Standard query 0x7872 A WIN-CQQ...
```

```
▷ Frame 1: 92 bytes on wire (736 bits), 92 bytes captured (736 bits) on interface 0
▷ Ethernet II, Src: Dell_87:77:13 (ec:f4:bb:87:77:13), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
▷ Internet Protocol Version 4, Src: 192.168.0.24, Dst: 192.168.0.255
▷ User Datagram Protocol, Src Port: 137 (137), Dst Port: 137 (137)
▷ NetBIOS Name Service
```

```
0000  ff ff ff ff ff ff ec f4  bb 87 77 13 08 00 45 00   ........ ..w...E.
0010  00 4e 34 a8 00 00 80 11  83 8f c0 a8 00 18 c0 a8   .N4..... ........
0020  00 ff 00 89 00 89 00 3a  9c 28 95 6b 01 10 00 01   .......: .(.k....
0030  00 00 00 00 00 00 20 46  43 45 45 43 4e 44 41 44   ...... F CEECNDAD
0040  43 44 41 43 41 43 41 43  41 43 41 43 41 43 41 43   CDACACAC ACACACAC
0050  41 43 41 43 41 41 41 00  00 20 00 01               ACACAAA. . ..
```

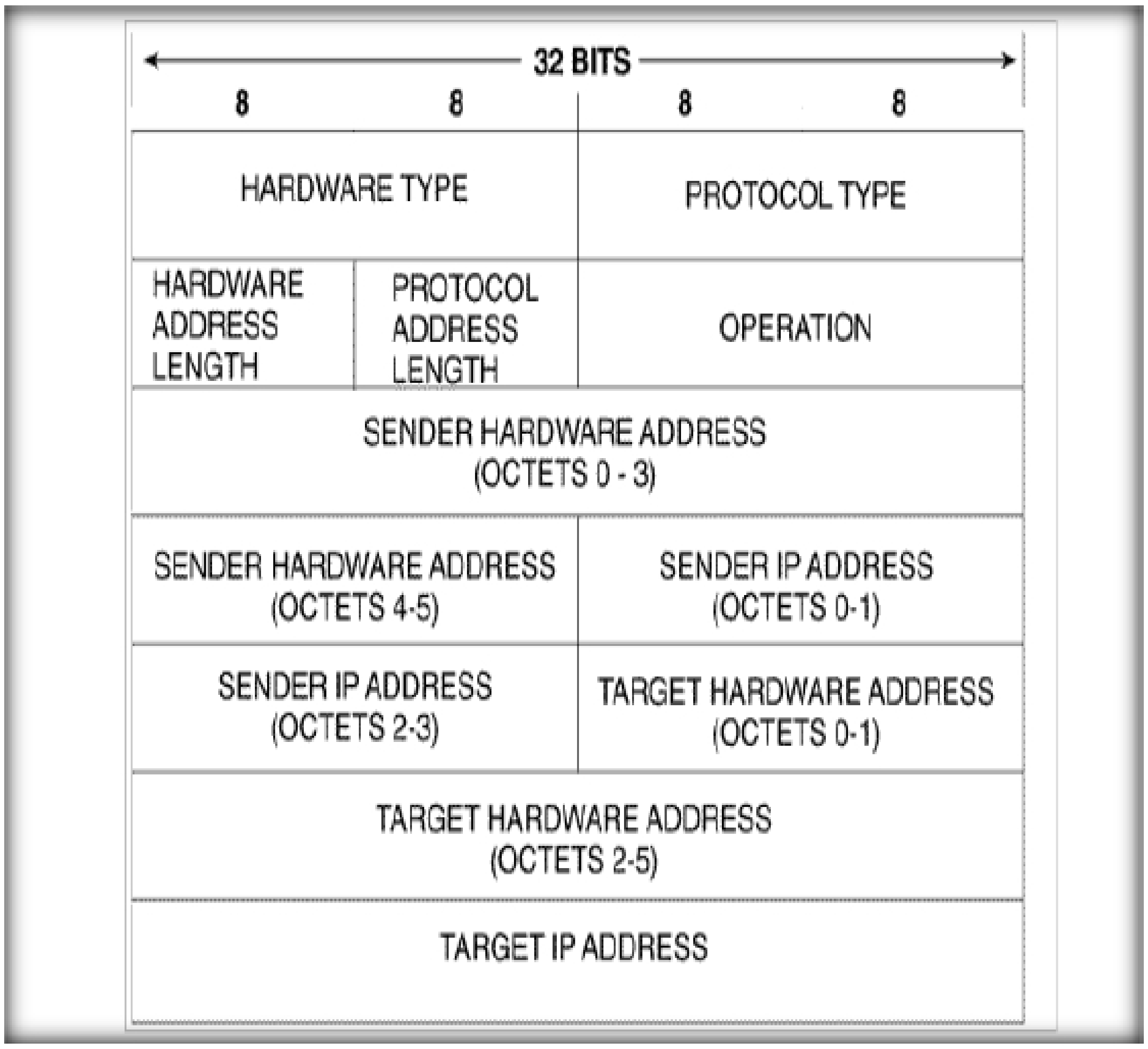
Packets: 7725 · Displayed: 7725 (100.0%) · Load time: 0:0.243 | Profile: Default

FIGURE 2.7: Wireshark GUI with Stop Button Highlighted

TASK 3

Inspecting ARP Packet Header

12. Typical **format of an ARP** packet **header is as shown in following figure**



32 BITS			
8	8	8	8
HARDWARE TYPE		PROTOCOL TYPE	
HARDWARE ADDRESS LENGTH	PROTOCOL ADDRESS LENGTH	OPERATION	
SENDER HARDWARE ADDRESS (OCTETS 0 - 3)			
SENDER HARDWARE ADDRESS (OCTETS 4-5)		SENDER IP ADDRESS (OCTETS 0-1)	
SENDER IP ADDRESS (OCTETS 2-3)		TARGET HARDWARE ADDRESS (OCTETS 0-1)	
TARGET HARDWARE ADDRESS (OCTETS 2-5)			
TARGET IP ADDRESS			

FIGURE 2.8: Typical Structure of an ARP Packet

The various fields are explained below

I. **Hardware Type (HTYPE):** Describes the Network adapter used, which is Ethernet.

II. **Protocol Type(PTYPE):** Describes the inter network protocol, for which the ARP packet was sent

III. **Hardware size (HLEN):** Describes the hardware address length

IV. **Protocol size (PSIZE):** Describes the protocol address length

V. **Opcode:** The operational code of an ARP packet, which describes if it is a request or a reply packet

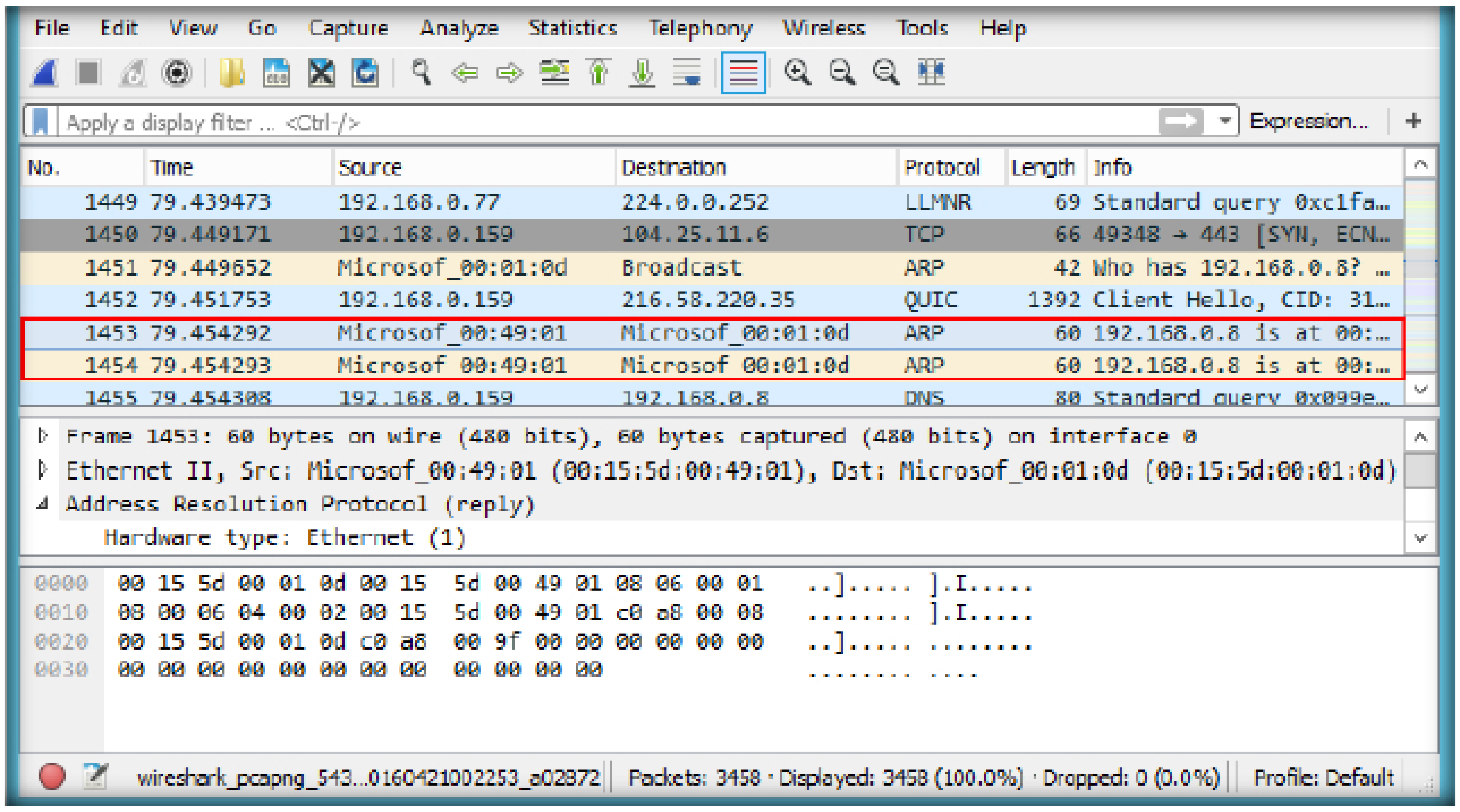
VI. **Sender MAC address:** Source system's MAC address

VII. **Sender IP address:** Source system's IP address.

VIII. **Target MAC address:** Destination system's MAC address

IX. **Target IP address:** Destination system's IP address.

13. Now, click on an **ARP packet** to analyze its various header fields


File
Edit
View
Go
Capture
Analyze
Statistics
Telephony
Wireless
Tools
Help
Apply a display filter ... <Ctrl-/>
Expression...
+
wireshark_pcapng_543...0160421002253_a02872
Packets: 3458 · Displayed: 3458 (100.0%) · Dropped: 0 (0.0%)
Profile: Default

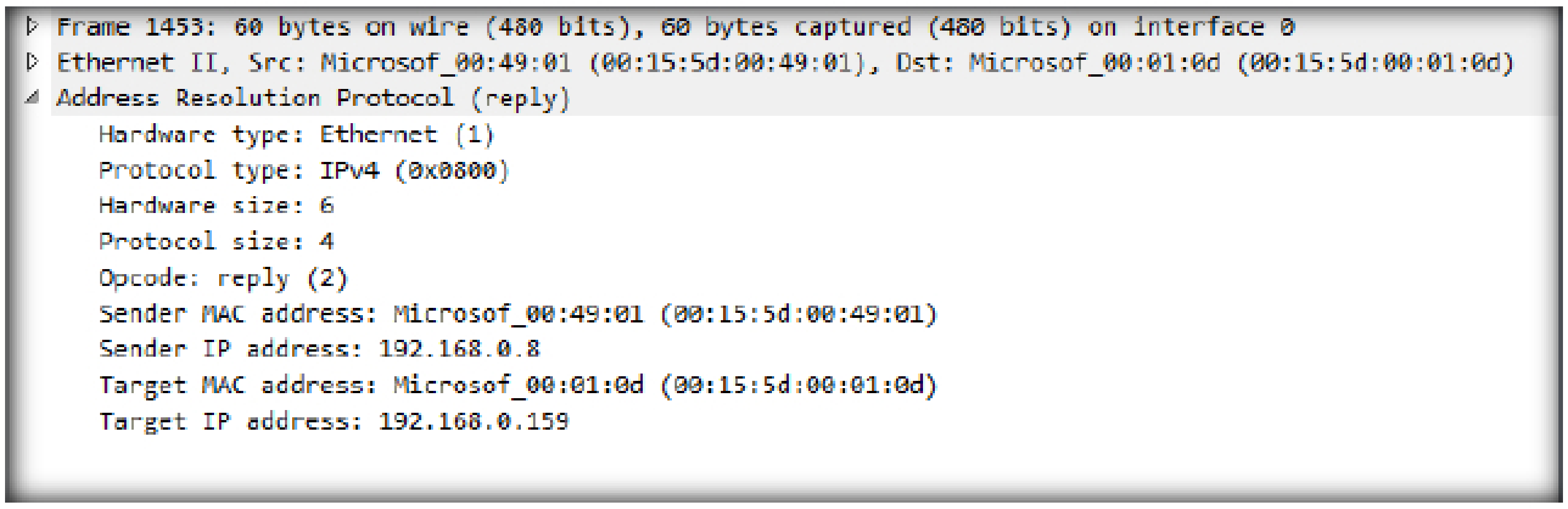
No.	Time	Source	Destination	Protocol	Length	Info
1449	79.439473	192.168.0.77	224.0.0.252	LLMNR	69	Standard query 0xc1fa…
1450	79.449171	192.168.0.159	104.25.11.6	TCP	66	49348 → 443 [SYN, ECN…
1451	79.449652	Microsof_00:01:0d	Broadcast	ARP	42	Who has 192.168.0.8? …
1452	79.451753	192.168.0.159	216.58.220.35	QUIC	1392	Client Hello, CID: 31…
1453	79.454292	Microsof_00:49:01	Microsof_00:01:0d	ARP	60	192.168.0.8 is at 00:…
1454	79.454293	Microsof 00:49:01	Microsof 00:01:0d	ARP	60	192.168.0.8 is at 00:…
1455	79.454308	192.168.0.159	192.168.0.8	DNS	80	Standard query 0x099e…

```
▷ Frame 1453: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
▷ Ethernet II, Src: Microsof_00:49:01 (00:15:5d:00:49:01), Dst: Microsof_00:01:0d (00:15:5d:00:01:0d)
◢ Address Resolution Protocol (reply)
    Hardware type: Ethernet (1)
```

```
0000  00 15 5d 00 01 0d 00 15  5d 00 49 01 08 06 00 01   ..]..... ].I.....
0010  08 00 06 04 00 02 00 15  5d 00 49 01 c0 a8 00 08   ........ ].I.....
0020  00 15 5d 00 01 0d c0 a8  00 9f 00 00 00 00 00 00   ..]..... ........
0030  00 00 00 00 00 00 00 00  00 00 00 00               ........ ....
```

FIGURE 2.9: Wireshark Captured Packets

14. Expand the **Address Resolution Protocol** node (here, **ARP Reply** node) under the **Packet details** Pane



```
▷ Frame 1453: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
▷ Ethernet II, Src: Microsof_00:49:01 (00:15:5d:00:49:01), Dst: Microsof_00:01:0d (00:15:5d:00:01:0d)
◢ Address Resolution Protocol (reply)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
    Sender MAC address: Microsof_00:49:01 (00:15:5d:00:49:01)
    Sender IP address: 192.168.0.8
    Target MAC address: Microsof_00:01:0d (00:15:5d:00:01:0d)
    Target IP address: 192.168.0.159
```

FIGURE 2.10 Captured ARP Packet

15. Compare and analyze the various fields in an ARP packet with the ARP packet header format

TASK 4

Inspecting TCP Packet Header

16. Typical TCP packet header format is as shown in following figure

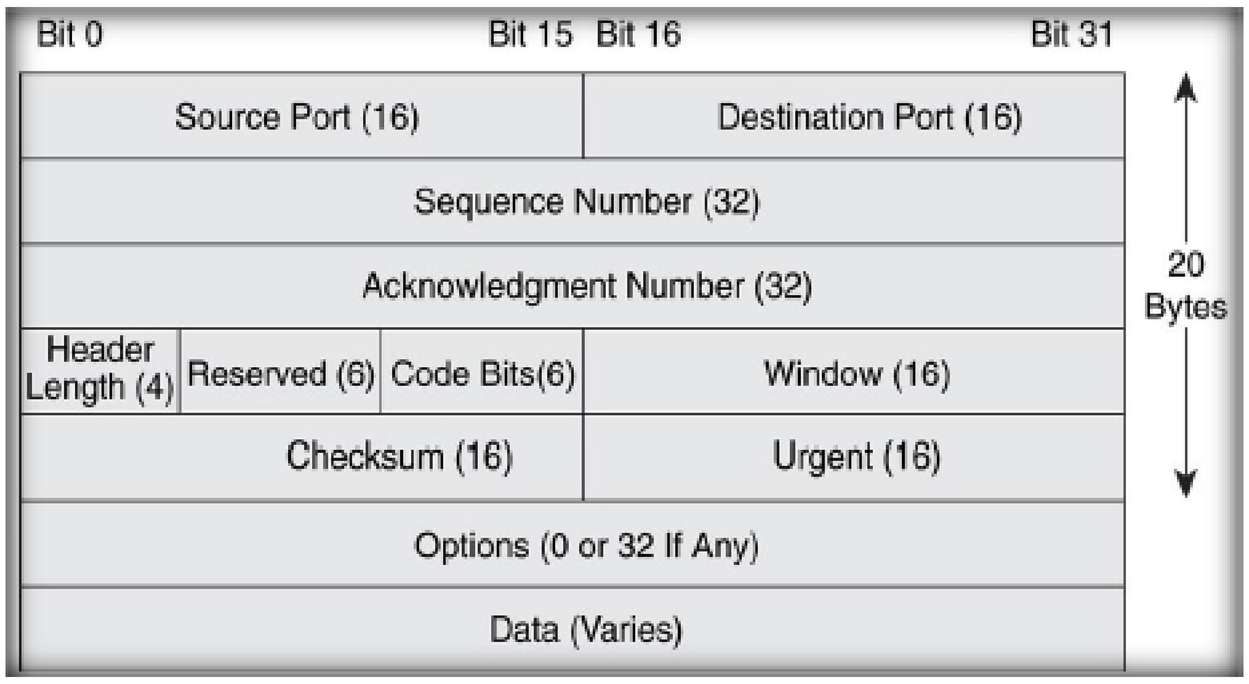


FIGURE 2.11: Typical Structure of a TCP Packet

The various fields are explained below

I. **Source port:** Port number of the source machine.

II. **Destination port:** Port number of the destination

III. **Sequence number:** The sequence number of the segment

IV. **Acknowledgement number:** The acknowledgement number of the segment

V. **Header length:** Specifies segment's total header length

VI. **Reserved:** Reserved bits for future use

VII. **Code bits (flags):** Specifies which flags are set based on nature of segment.

VIII. **Window size:** Maximum length of segment which the sender can receive as a reply to this segment and starts from acknowledgement number

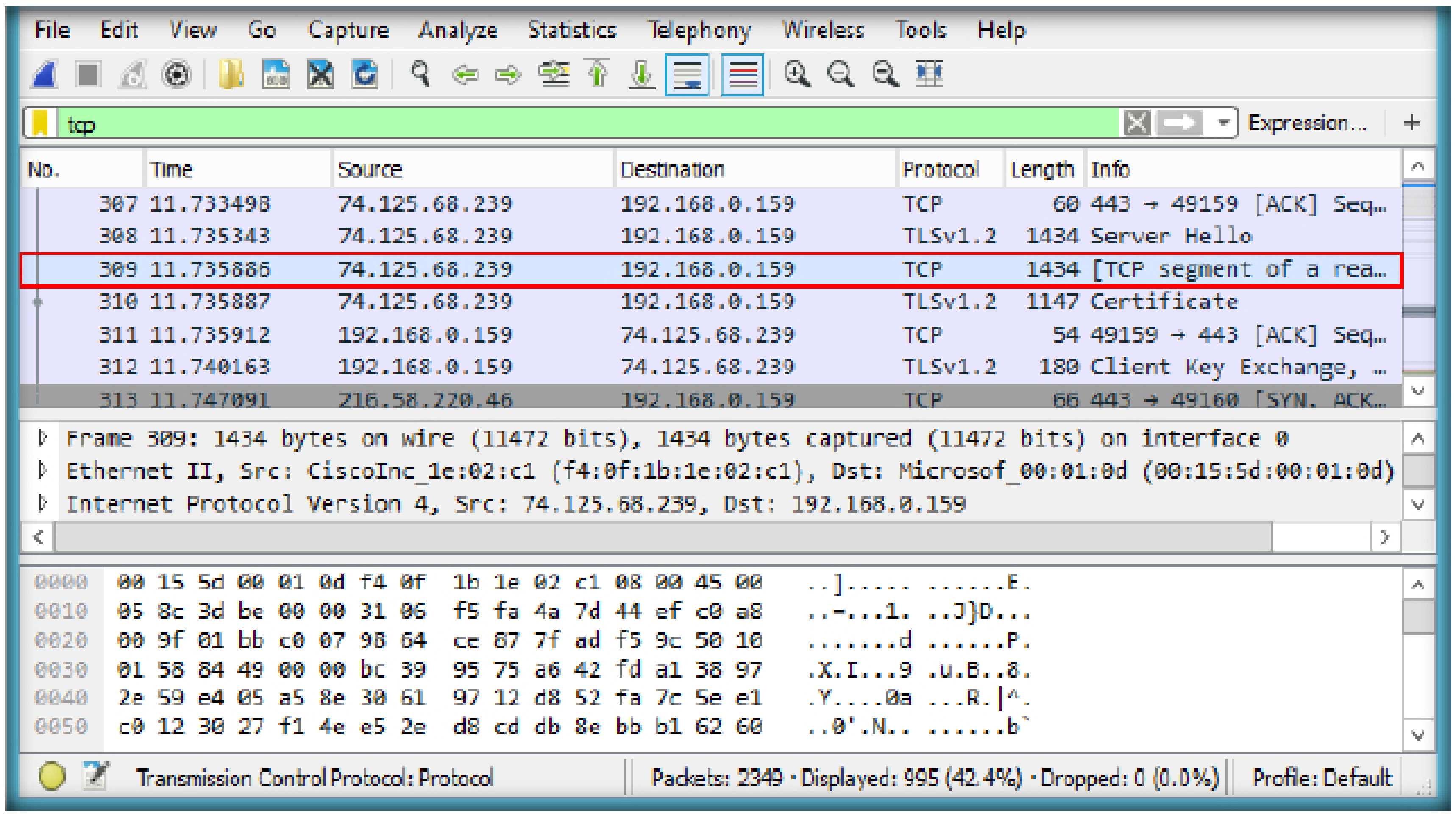
IX. **Checksum:** Specifies the error detection data

X. **Urgent:** If set, implies that urgent reply needed from recipient.

XI. **Options:** Can be from 0-32 bits in multiples of eight, used optionally in checksum calculation.

XII. **Segment data:** total data length of the segment

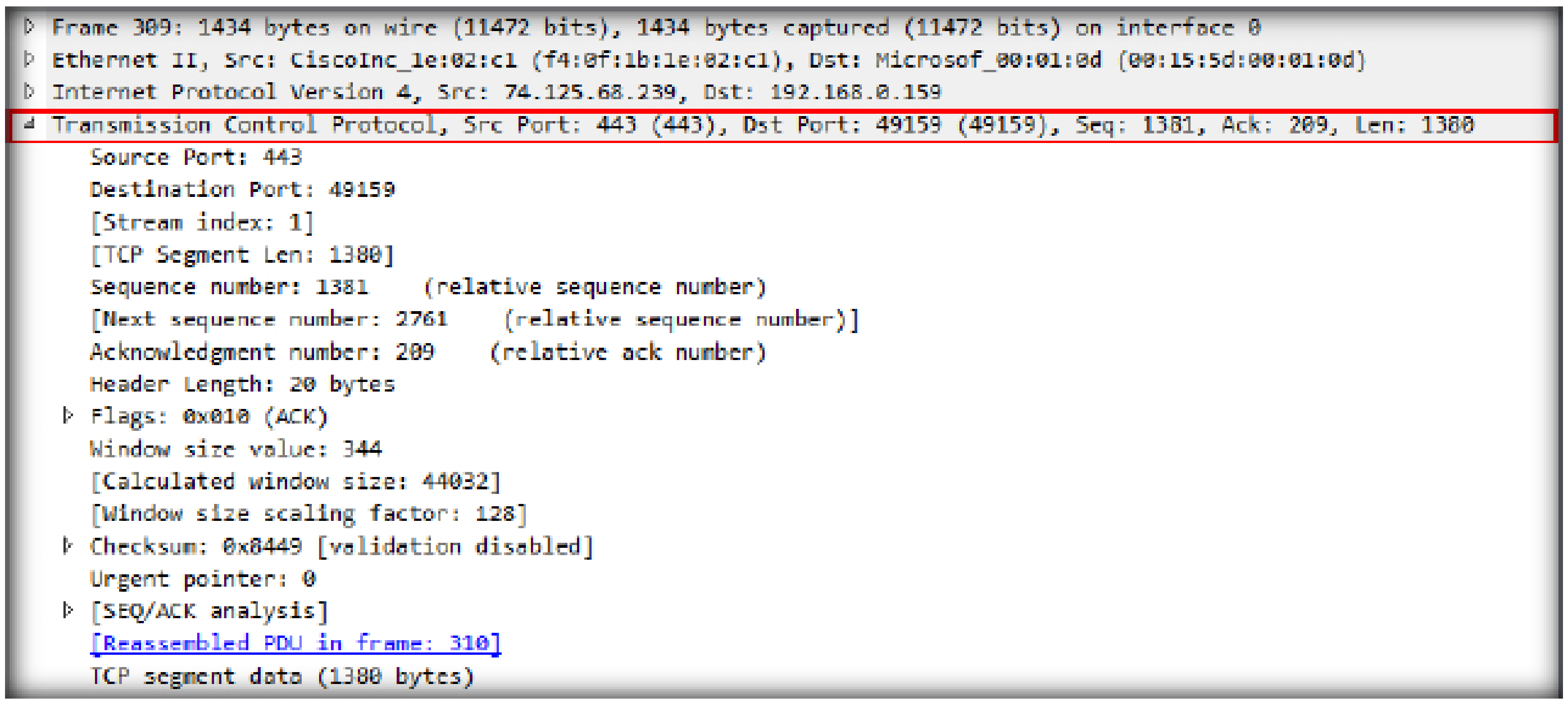
17. Now, **click** on a **TCP packet** to analyze its various header fields.


File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Expression...
+

```
tcp

No.   Time          Source            Destination       Protocol  Length Info
  307 11.733498     74.125.68.239     192.168.0.159     TCP           60 443 → 49159 [ACK] Seq…
  308 11.735343     74.125.68.239     192.168.0.159     TLSv1.2     1434 Server Hello
  309 11.735886     74.125.68.239     192.168.0.159     TCP         1434 [TCP segment of a rea…
  310 11.735887     74.125.68.239     192.168.0.159     TLSv1.2     1147 Certificate
  311 11.735912     192.168.0.159     74.125.68.239     TCP           54 49159 → 443 [ACK] Seq…
  312 11.740163     192.168.0.159     74.125.68.239     TLSv1.2      180 Client Key Exchange, …
  313 11.747091     216.58.220.46     192.168.0.159     TCP           66 443 → 49160 [SYN, ACK…

▷ Frame 309: 1434 bytes on wire (11472 bits), 1434 bytes captured (11472 bits) on interface 0
▷ Ethernet II, Src: CiscoInc_1e:02:c1 (f4:0f:1b:1e:02:c1), Dst: Microsof_00:01:0d (00:15:5d:00:01:0d)
▷ Internet Protocol Version 4, Src: 74.125.68.239, Dst: 192.168.0.159

0000  00 15 5d 00 01 0d f4 0f  1b 1e 02 c1 08 00 45 00   ..]..... ......E.
0010  05 8c 3d be 00 00 31 06  f5 fa 4a 7d 44 ef c0 a8   ..=...1. ..J}D...
0020  00 9f 01 bb c0 07 98 64  ce 87 7f ad f5 9c 50 10   .......d ......P.
0030  01 58 84 49 00 00 bc 39  95 75 a6 42 fd a1 38 97   .X.I...9 .u.B..8.
0040  2e 59 e4 05 a5 8e 30 61  97 12 d8 52 fa 7c 5e e1   .Y....0a ...R.|^.
0050  c0 12 30 27 f1 4e e5 2e  d8 cd db 8e bb b1 62 60   ..0'.N.. ......b`

Transmission Control Protocol: Protocol | Packets: 2349 · Displayed: 995 (42.4%) · Dropped: 0 (0.0%) | Profile: Default
```

FIGURE 2.12: TCP Packets in Wireshark

18. Expand the **Transmission Control Protocol** node in the Packet Details Pane



```
▷ Frame 309: 1434 bytes on wire (11472 bits), 1434 bytes captured (11472 bits) on interface 0
▷ Ethernet II, Src: CiscoInc_1e:02:c1 (f4:0f:1b:1e:02:c1), Dst: Microsof_00:01:0d (00:15:5d:00:01:0d)
▷ Internet Protocol Version 4, Src: 74.125.68.239, Dst: 192.168.0.159
◢ Transmission Control Protocol, Src Port: 443 (443), Dst Port: 49159 (49159), Seq: 1381, Ack: 209, Len: 1380
    Source Port: 443
    Destination Port: 49159
    [Stream index: 1]
    [TCP Segment Len: 1380]
    Sequence number: 1381    (relative sequence number)
    [Next sequence number: 2761    (relative sequence number)]
    Acknowledgment number: 209    (relative ack number)
    Header Length: 20 bytes
  ▷ Flags: 0x010 (ACK)
    Window size value: 344
    [Calculated window size: 44032]
    [Window size scaling factor: 128]
  ▷ Checksum: 0x8449 [validation disabled]
    Urgent pointer: 0
  ▷ [SEQ/ACK analysis]
    [Reassembled PDU in frame: 310]
    TCP segment data (1380 bytes)
```

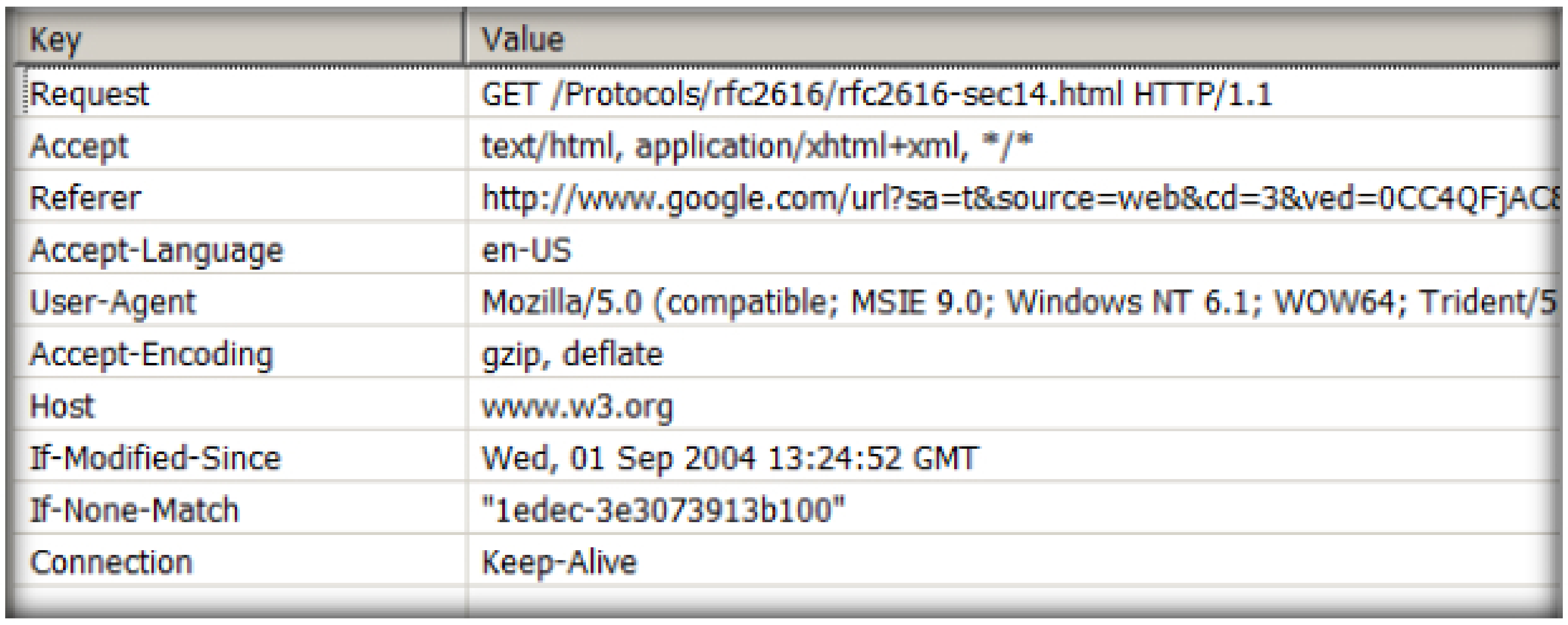
FIGURE 2.13: TCP Packet

19. Compare and analyze the various fields in a TCP packet with the TCP packet header format

TASK 5

Inspecting HTTP Packet Header

20. Typical structure of a HTTP request header is shown in the following figure:



Key	Value
Request	GET /Protocols/rfc2616/rfc2616-sec14.html HTTP/1.1
Accept	text/html, application/xhtml+xml, */*
Referer	http://www.google.com/url?sa=t&source=web&cd=3&ved=0CC4QFjAC
Accept-Language	en-US
User-Agent	Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5
Accept-Encoding	gzip, deflate
Host	www.w3.org
If-Modified-Since	Wed, 01 Sep 2004 13:24:52 GMT
If-None-Match	"1edec-3e3073913b100"
Connection	Keep-Alive

FIGURE 2.14: Typical Structure of a HTTP Request Header

The various fields are explained below

I. **HTTP type**: There are two types of HTTP messages. Request and response. All HTTP packets belong to one of the above two formats.

II. **Request type**: Specifies the type of request. GET, POST, HEAD and so on

III. **Accept:** Uses wild cards to specify the acceptable media

IV. **Referer:** Points to the site which requested a resource on behalf of the client

V. **Accept language:** Language in which the requested response is preferred

VI. **User agent:** Client software which requested the resource on the client's behalf

VII. **Accept encoding:** Specifies the content coding acceptable in the response packet

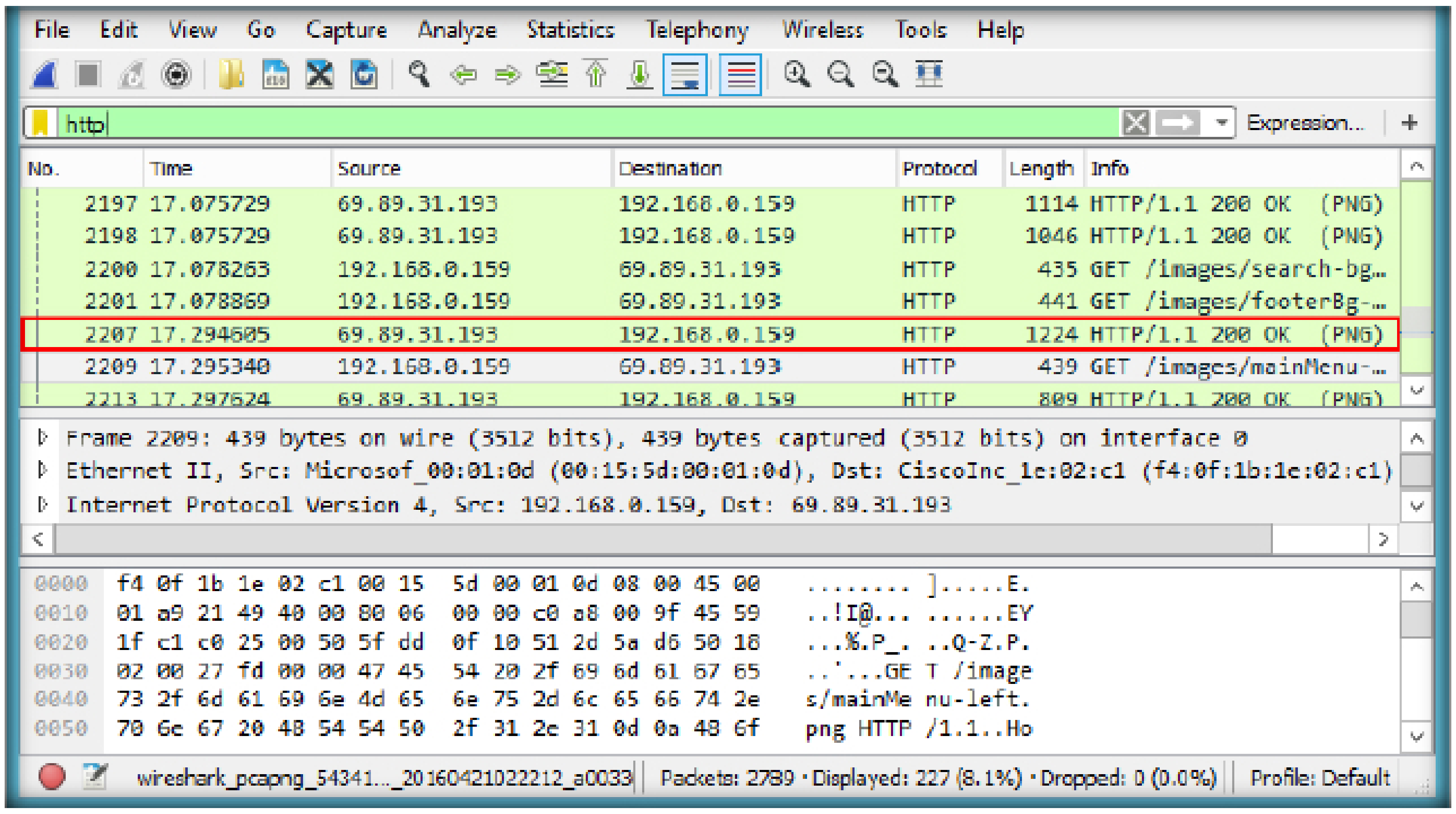
VIII. **Host:** The URL at which the actual resource requested or part of resource is available.

IX. **If-modified-since:** Used to impose a condition. If the requested response is not modified after the time period specified then the resource is not returned. A *304- not modified* message is returned

X. **If-none-match:** Used to impose a condition. Client specifies a set of resources it already has, in this field. If requested information is same as the information in this field server does not need to send the same information again. It sends a *304 message.*

XI. **Connection:** Defines the type of connection to be established.

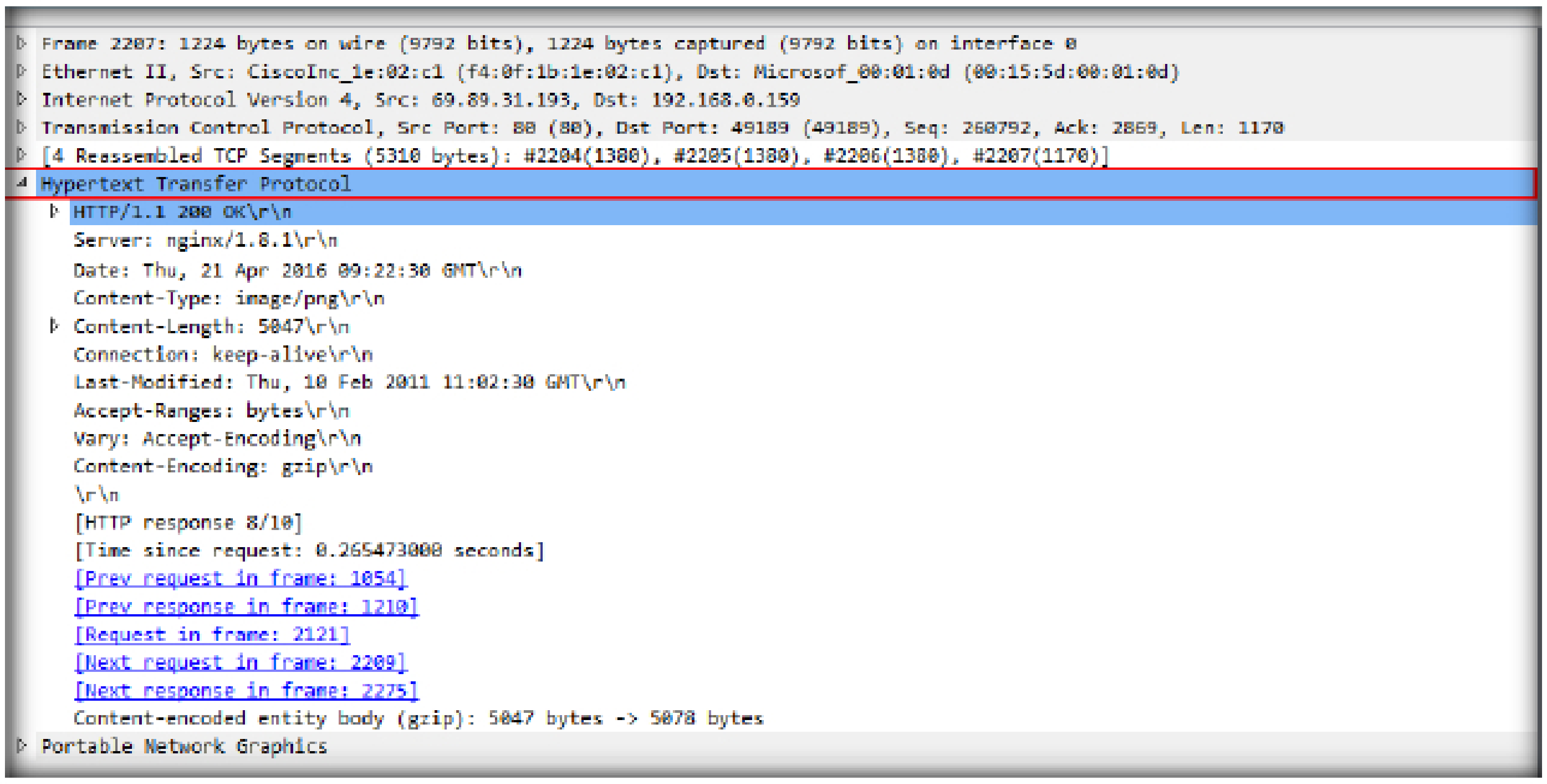
21. Now, click on a HTTP packet to analyze various fields


File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Expression...
+

```
http

No.      Time        Source          Destination     Protocol  Length Info
   2197 17.075729   69.89.31.193    192.168.0.159   HTTP      1114 HTTP/1.1 200 OK  (PNG)
   2198 17.075729   69.89.31.193    192.168.0.159   HTTP      1046 HTTP/1.1 200 OK  (PNG)
   2200 17.078263   192.168.0.159   69.89.31.193    HTTP       435 GET /images/search-bg…
   2201 17.078869   192.168.0.159   69.89.31.193    HTTP       441 GET /images/footerBg-…
   2207 17.294605   69.89.31.193    192.168.0.159   HTTP      1224 HTTP/1.1 200 OK  (PNG)
   2209 17.295340   192.168.0.159   69.89.31.193    HTTP       439 GET /images/mainMenu-…
   2213 17.297624   69.89.31.193    192.168.0.159   HTTP       809 HTTP/1.1 200 OK  (PNG)

▷ Frame 2209: 439 bytes on wire (3512 bits), 439 bytes captured (3512 bits) on interface 0
▷ Ethernet II, Src: Microsof_00:01:0d (00:15:5d:00:01:0d), Dst: CiscoInc_1e:02:c1 (f4:0f:1b:1e:02:c1)
▷ Internet Protocol Version 4, Src: 192.168.0.159, Dst: 69.89.31.193

0000  f4 0f 1b 1e 02 c1 00 15  5d 00 01 0d 08 00 45 00   ........ ].....E.
0010  01 a9 21 49 40 00 80 06  00 00 c0 a8 00 9f 45 59   ..!I@... ......EY
0020  1f c1 c0 25 00 50 5f dd  0f 10 51 2d 5a d6 50 18   ...%.P_. ..Q-Z.P.
0030  02 00 27 fd 00 00 47 45  54 20 2f 69 6d 61 67 65   ..'...GE T /image
0040  73 2f 6d 61 69 6e 4d 65  6e 75 2d 6c 65 66 74 2e   s/mainMe nu-left.
0050  70 6e 67 20 48 54 54 50  2f 31 2e 31 0d 0a 48 6f   png HTTP /1.1..Ho

wireshark_pcapng_54341..._20160421022212_a0033 | Packets: 2789 · Displayed: 227 (8.1%) · Dropped: 0 (0.0%) | Profile: Default
```

FIGURE 2.15: HTTP Packets in Wireshark

22. Expand the **Hypertext Transfer Protocol** node in the Packet Details Pane



```
▷ Frame 2207: 1224 bytes on wire (9792 bits), 1224 bytes captured (9792 bits) on interface 0
▷ Ethernet II, Src: CiscoInc_1e:02:c1 (f4:0f:1b:1e:02:c1), Dst: Microsof_00:01:0d (00:15:5d:00:01:0d)
▷ Internet Protocol Version 4, Src: 69.89.31.193, Dst: 192.168.0.159
▷ Transmission Control Protocol, Src Port: 80 (80), Dst Port: 49189 (49189), Seq: 260792, Ack: 2869, Len: 1170
▷ [4 Reassembled TCP Segments (5310 bytes): #2204(1380), #2205(1380), #2206(1380), #2207(1170)]
◢ Hypertext Transfer Protocol
  ▷ HTTP/1.1 200 OK\r\n
    Server: nginx/1.8.1\r\n
    Date: Thu, 21 Apr 2016 09:22:30 GMT\r\n
    Content-Type: image/png\r\n
  ▷ Content-Length: 5047\r\n
    Connection: keep-alive\r\n
    Last-Modified: Thu, 10 Feb 2011 11:02:30 GMT\r\n
    Accept-Ranges: bytes\r\n
    Vary: Accept-Encoding\r\n
    Content-Encoding: gzip\r\n
    \r\n
    [HTTP response 8/10]
    [Time since request: 0.265473000 seconds]
    [Prev request in frame: 1054]
    [Prev response in frame: 1210]
    [Request in frame: 2121]
    [Next request in frame: 2209]
    [Next response in frame: 2275]
    Content-encoded entity body (gzip): 5047 bytes -> 5078 bytes
▷ Portable Network Graphics
```

FIGURE 2.16: HTTP Packet

23. Compare and analyze the various fields in the HTTP packet with the HTTP request header format

TASK 6

Inspecting ICMP Packet Header

24. Typical structure of an ICMP packet is as shown in following figure

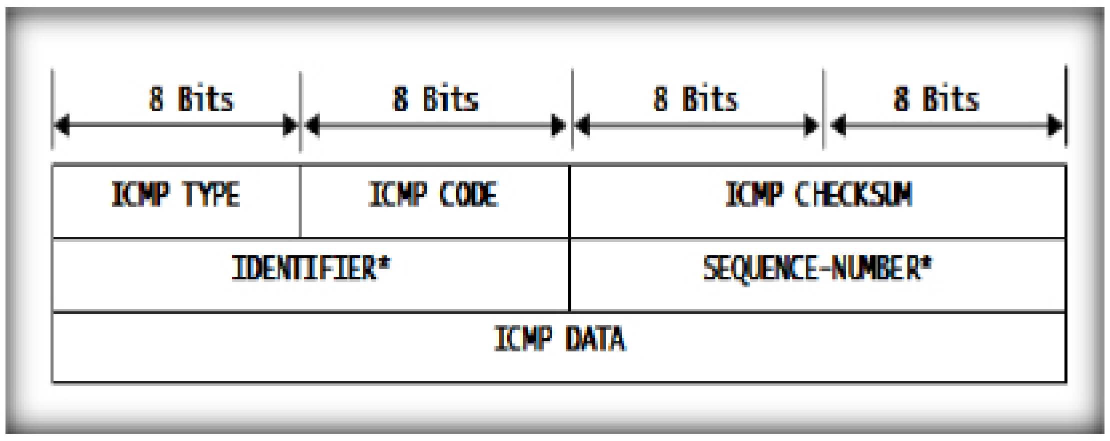


FIGURE 2.17: Typical structure of an ICMP packet

The various fields are explained below

I. **Type**: It defines the ICMP message type. It has two values. 0 or 8. 0 implies that it is an echo reply ICMP packet and 8 implies that it is an echo request packet.

II. **Code:** Code is zero for both ICMP request and reply packets. Its value varies in other types of ICMP packets.

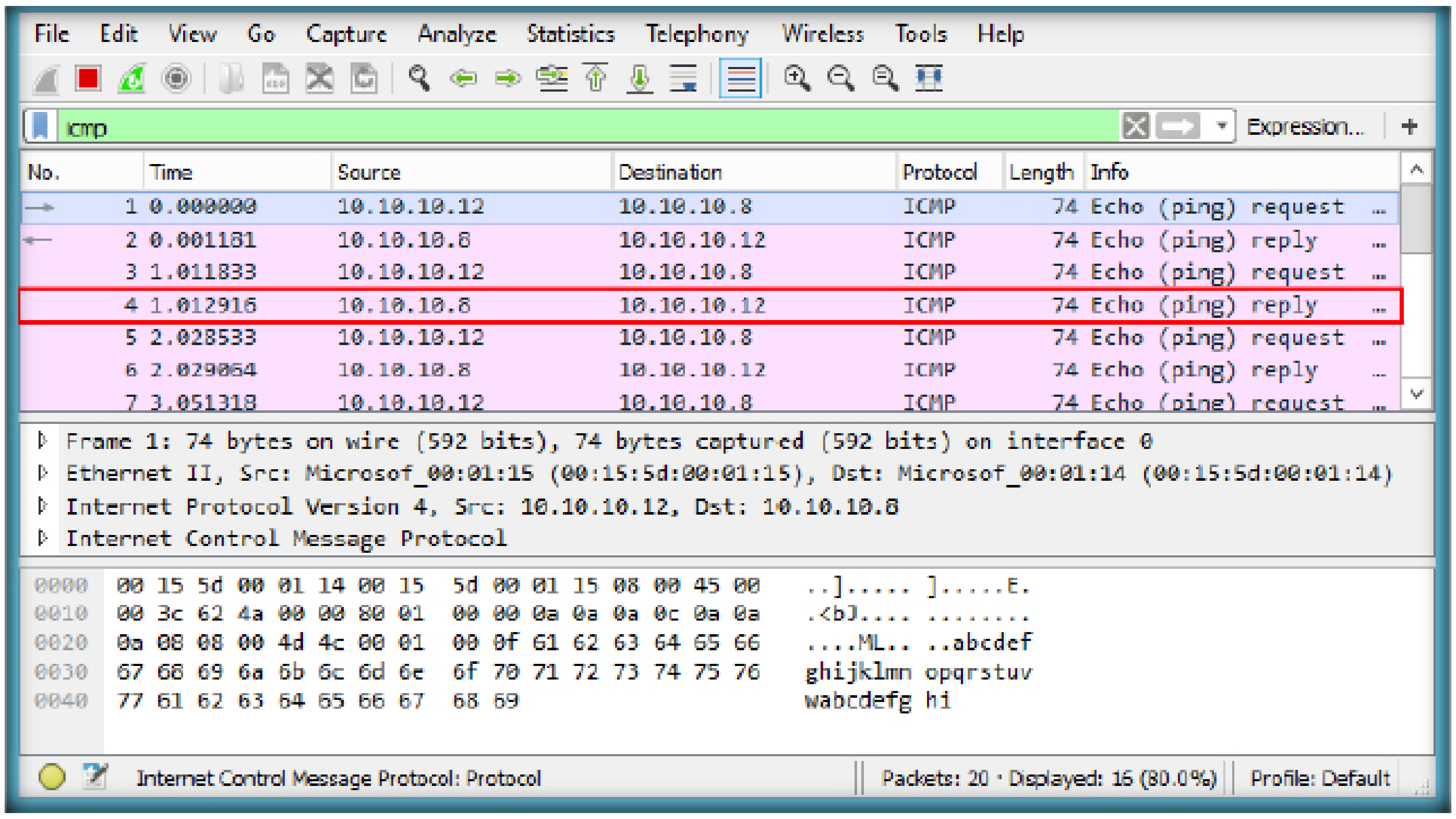
III. **Checksum**: Checksum data is used for error detection.

IV. **Identifier**: It is set to the process ID of the sender

V. **Sequence Number**: It begins with zero and increments by one, with every ICMP echo request packet being sent.

VI. **Data**: Contains ICMP data

25. Now click on the ICMP packet to analyze its various fields



```
No.     Time        Source          Destination     Protocol  Length  Info
      1 0.000000    10.10.10.12     10.10.10.8      ICMP          74  Echo (ping) request  ...
      2 0.001181    10.10.10.8      10.10.10.12     ICMP          74  Echo (ping) reply    ...
      3 1.011833    10.10.10.12     10.10.10.8      ICMP          74  Echo (ping) request  ...
      4 1.012916    10.10.10.8      10.10.10.12     ICMP          74  Echo (ping) reply    ...
      5 2.028533    10.10.10.12     10.10.10.8      ICMP          74  Echo (ping) request  ...
      6 2.029064    10.10.10.8      10.10.10.12     ICMP          74  Echo (ping) reply    ...
      7 3.051318    10.10.10.12     10.10.10.8      ICMP          74  Echo (ping) request  ...
```

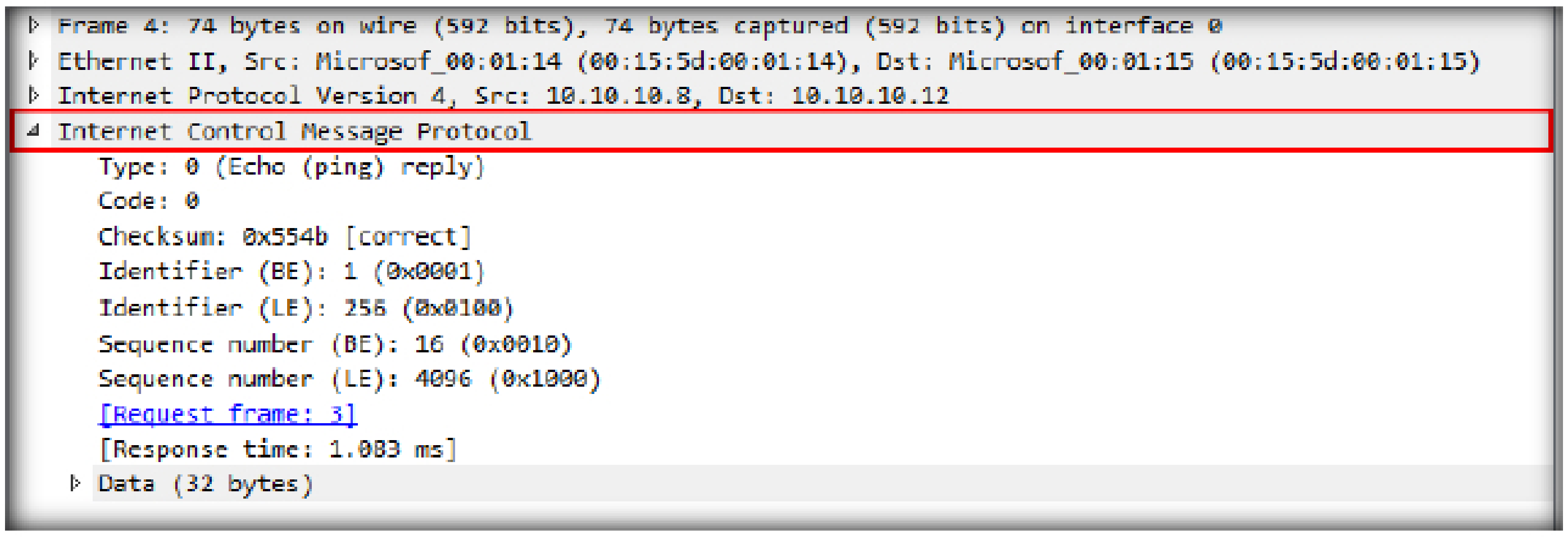
```
▷ Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
▷ Ethernet II, Src: Microsof_00:01:15 (00:15:5d:00:01:15), Dst: Microsof_00:01:14 (00:15:5d:00:01:14)
▷ Internet Protocol Version 4, Src: 10.10.10.12, Dst: 10.10.10.8
▷ Internet Control Message Protocol
```

```
0000  00 15 5d 00 01 14 00 15  5d 00 01 15 08 00 45 00   ..]..... ].....E.
0010  00 3c 62 4a 00 00 80 01  00 00 0a 0a 0a 0c 0a 0a   .<bJ.... ........
0020  0a 08 08 00 4d 4c 00 01  00 0f 61 62 63 64 65 66   ....ML.. ..abcdef
0030  67 68 69 6a 6b 6c 6d 6e  6f 70 71 72 73 74 75 76   ghijklmn opqrstuv
0040  77 61 62 63 64 65 66 67  68 69                     wabcdefg hi
```

Internet Control Message Protocol: Protocol | Packets: 20 · Displayed: 16 (80.0%) | Profile: Default

FIGURE 2.18: ICMP Packets in Wireshark

26. Expand the **Internet Control Message Protocol** node in the **Packet Details** Pane



```
▷ Frame 4: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
▷ Ethernet II, Src: Microsof_00:01:14 (00:15:5d:00:01:14), Dst: Microsof_00:01:15 (00:15:5d:00:01:15)
▷ Internet Protocol Version 4, Src: 10.10.10.8, Dst: 10.10.10.12
◢ Internet Control Message Protocol
    Type: 0 (Echo (ping) reply)
    Code: 0
    Checksum: 0x554b [correct]
    Identifier (BE): 1 (0x0001)
    Identifier (LE): 256 (0x0100)
    Sequence number (BE): 16 (0x0010)
    Sequence number (LE): 4096 (0x1000)
    [Request frame: 3]
    [Response time: 1.083 ms]
  ▷ Data (32 bytes)
```

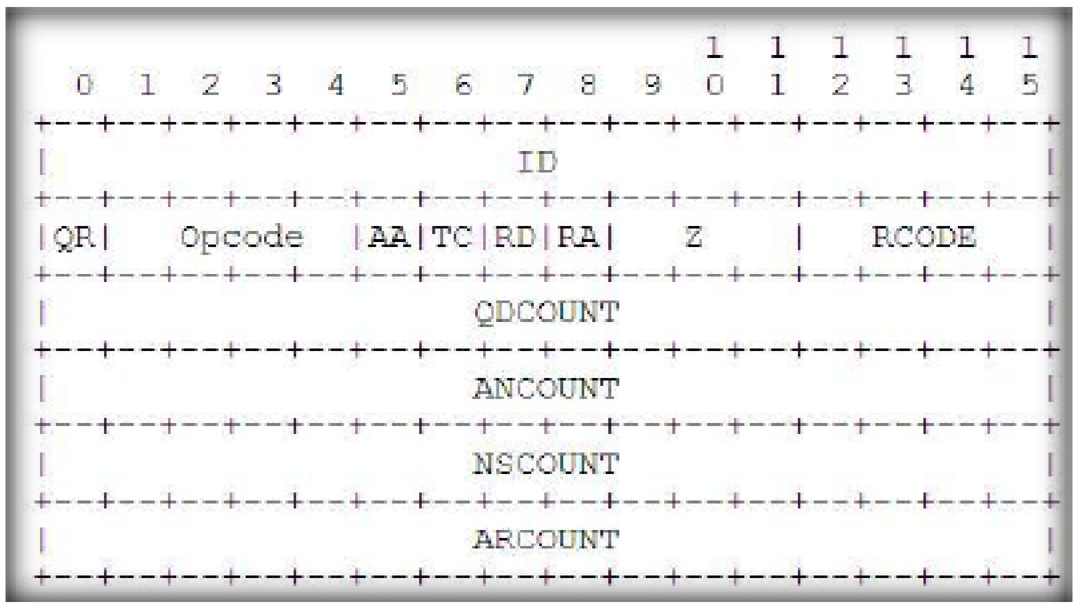
FIGURE 2.19: Captured ICMP packet

27. Compare and analyze the various fields in an ICMP packet with the ICMP header format

TASK 7

Inspecting DNS Packet Header

28. Typical structure of a DNS packet header is as shown in following figure



```
                                  1  1  1  1  1  1
    0  1  2  3  4  5  6  7  8  9  0  1  2  3  4  5
  +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
  |                      ID                       |
  +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
  |QR|   Opcode  |AA|TC|RD|RA|   Z    |   RCODE   |
  +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
  |                    QDCOUNT                    |
  +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
  |                    ANCOUNT                    |
  +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
  |                    NSCOUNT                    |
  +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
  |                    ARCOUNT                    |
  +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
```

FIGURE 2.20: Typical structure of a DNS Packet Header

The various fields are explained as follows:

I. **Identifier:** A 2-byte ID created by the system that created the DNS query.

II. **Query/response flag:** Defines the type of DNS packet, if it is a Query type or response type flag.

III. **Operational code:** Specifies the type of query present in the message. It is created by the sender and is the same in the response message as well

IV. **Authoritative answer flag (AA):** Applies to response messages. A non-authoritative response is indicated by zero and a 1 implies that the server is responsive

V. **Truncation flag (TC):** If set to 1 implies the message was truncated as it was long, and a zero indicates no truncation.

VI. **Recursion Desired (RD):** Applies to request type packets. If set to 1, implies that server needs to recursively reply to the client.

VII. **Recursion available (RA):** Applies to response queries. If set to 1, implies that server supports recursive reply.

VIII. **Zero(Z):** Three bits are reserved and are always set to zero

IX. **Response code (Rcode):** Set to zero in a request query. Server does not alter the query if response was successful. (zero). Any other value returned by server implies an error in response

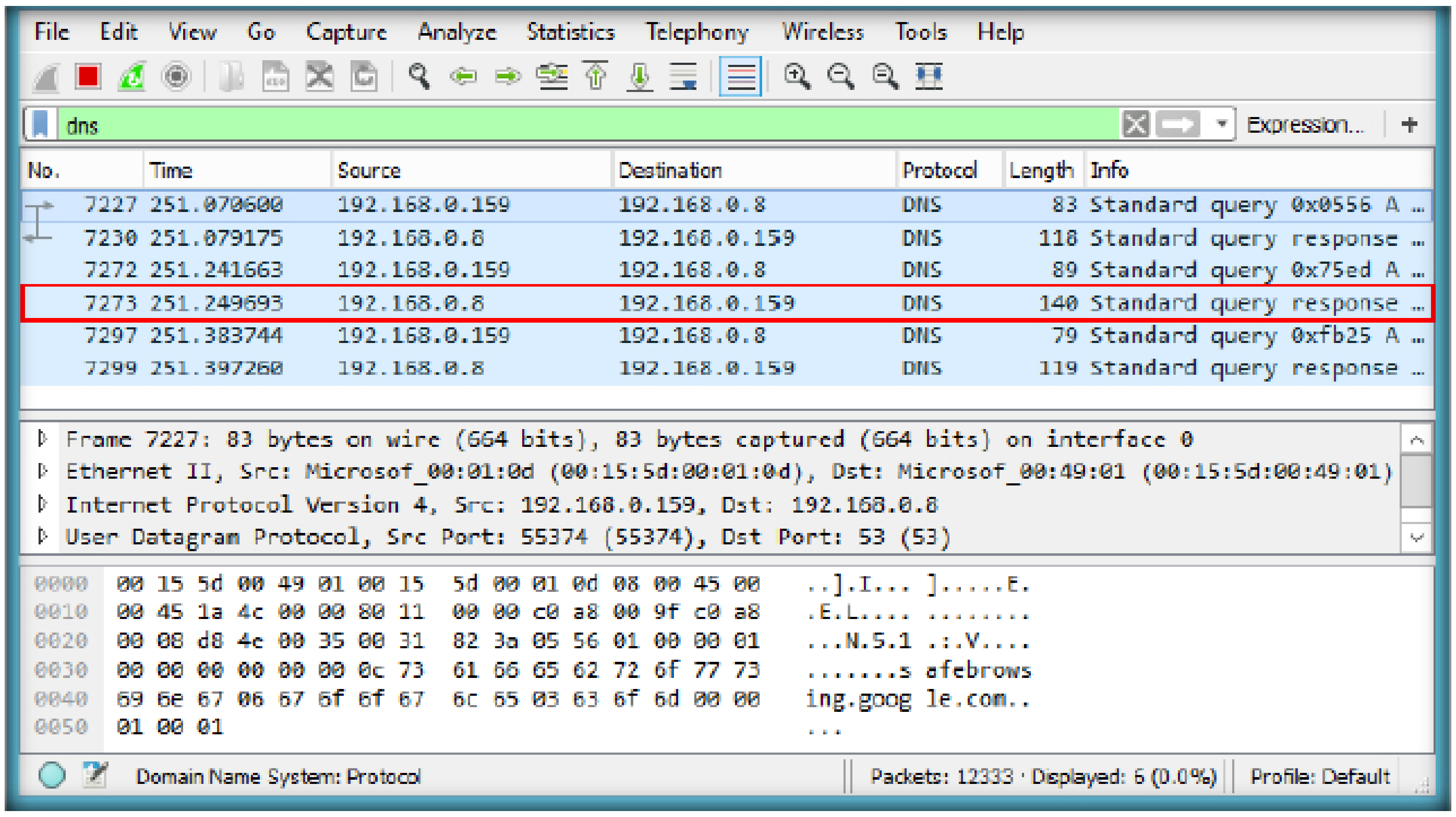
X. **Question count (QD count):** Specifies the number of questions in question segment

XI. **Answer record count (AN count):** Specifies number of resource records in answer section.

XII. **Authority record count(NS count):** Specifies number of records in authority section

XIII. **Additional record count (AR count):** Specifies number of resource records in the additional section

29. Now, click on any DNS packet capture to analyze it various fields



```
No.    Time        Source          Destination     Protocol Length Info
  7227 251.070600  192.168.0.159   192.168.0.8     DNS          83 Standard query 0x0556 A …
  7230 251.079175  192.168.0.8     192.168.0.159   DNS         118 Standard query response …
  7272 251.241663  192.168.0.159   192.168.0.8     DNS          89 Standard query 0x75ed A …
  7273 251.249693  192.168.0.8     192.168.0.159   DNS         140 Standard query response …
  7297 251.383744  192.168.0.159   192.168.0.8     DNS          79 Standard query 0xfb25 A …
  7299 251.397260  192.168.0.8     192.168.0.159   DNS         119 Standard query response …
```

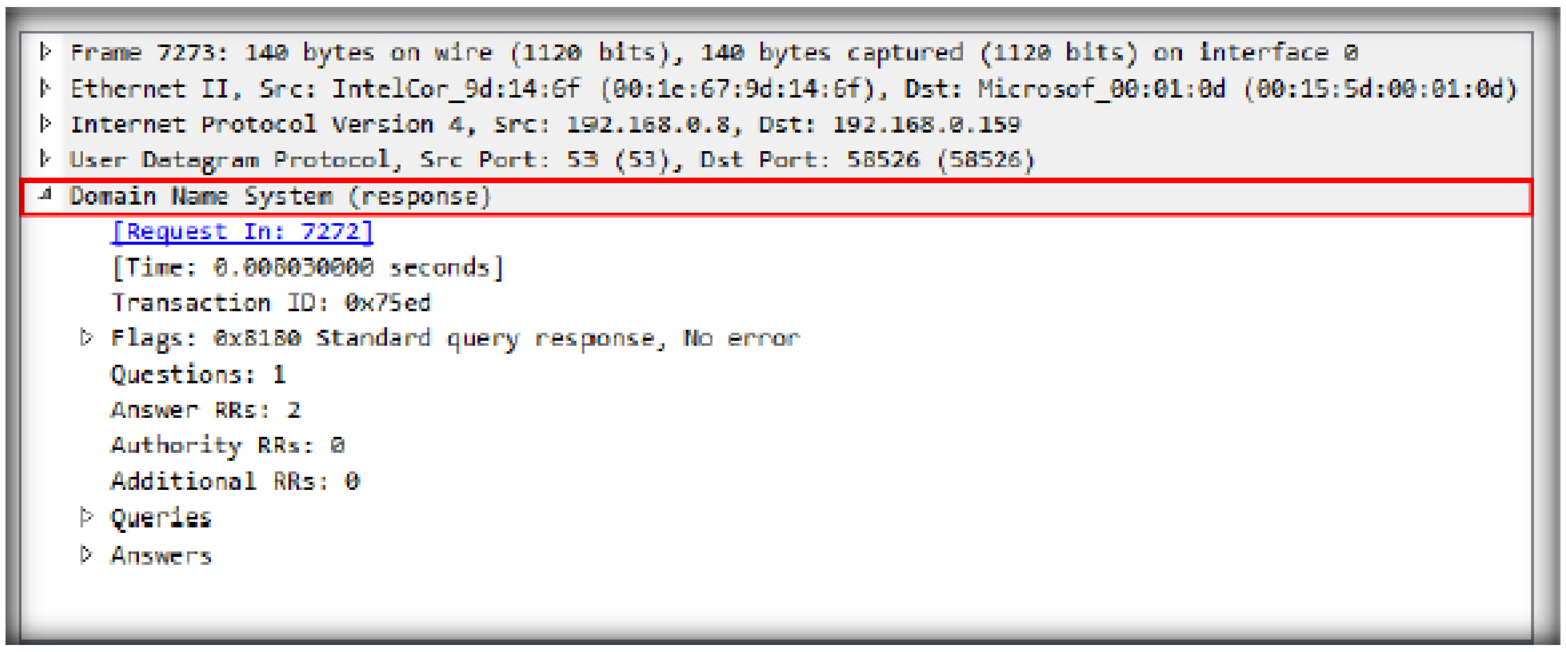
```
▷ Frame 7227: 83 bytes on wire (664 bits), 83 bytes captured (664 bits) on interface 0
▷ Ethernet II, Src: Microsof_00:01:0d (00:15:5d:00:01:0d), Dst: Microsof_00:49:01 (00:15:5d:00:49:01)
▷ Internet Protocol Version 4, Src: 192.168.0.159, Dst: 192.168.0.8
▷ User Datagram Protocol, Src Port: 55374 (55374), Dst Port: 53 (53)
```

```
0000  00 15 5d 00 49 01 00 15  5d 00 01 0d 08 00 45 00   ..].I... ].....E.
0010  00 45 1a 4c 00 00 80 11  00 00 c0 a8 00 9f c0 a8   .E.L.... ........
0020  00 08 d8 4e 00 35 00 31  82 3a 05 56 01 00 00 01   ...N.5.1 .:.V....
0030  00 00 00 00 00 00 0c 73  61 66 65 62 72 6f 77 73   .......s afebrows
0040  69 6e 67 06 67 6f 6f 67  6c 65 03 63 6f 6d 00 00   ing.goog le.com..
0050  01 00 01                                           ...
```

Domain Name System: Protocol | Packets: 12333 · Displayed: 6 (0.0%) | Profile: Default

FIGURE 2.21: DNS Packets in Wireshark

30. Expand the **Domain Name System(Query)** node in the Packet details Pane



```
▷ Frame 7273: 140 bytes on wire (1120 bits), 140 bytes captured (1120 bits) on interface 0
▷ Ethernet II, Src: IntelCor_9d:14:6f (00:1c:67:9d:14:6f), Dst: Microsof_00:01:0d (00:15:5d:00:01:0d)
▷ Internet Protocol Version 4, Src: 192.168.0.8, Dst: 192.168.0.159
▷ User Datagram Protocol, Src Port: 53 (53), Dst Port: 58526 (58526)
◢ Domain Name System (response)
     [Request In: 7272]
     [Time: 0.008030000 seconds]
     Transaction ID: 0x75ed
   ▷ Flags: 0x8180 Standard query response, No error
     Questions: 1
     Answer RRs: 2
     Authority RRs: 0
     Additional RRs: 0
   ▷ Queries
   ▷ Answers
```

FIGURE 2.22: Captured DNS Packet

31. Compare and analyze the various fields in the DNS packet with the DNS header format

TASK 8

Inspecting UDP Packet Header

32. Typical structure of a UDP packet is as shown in following figure

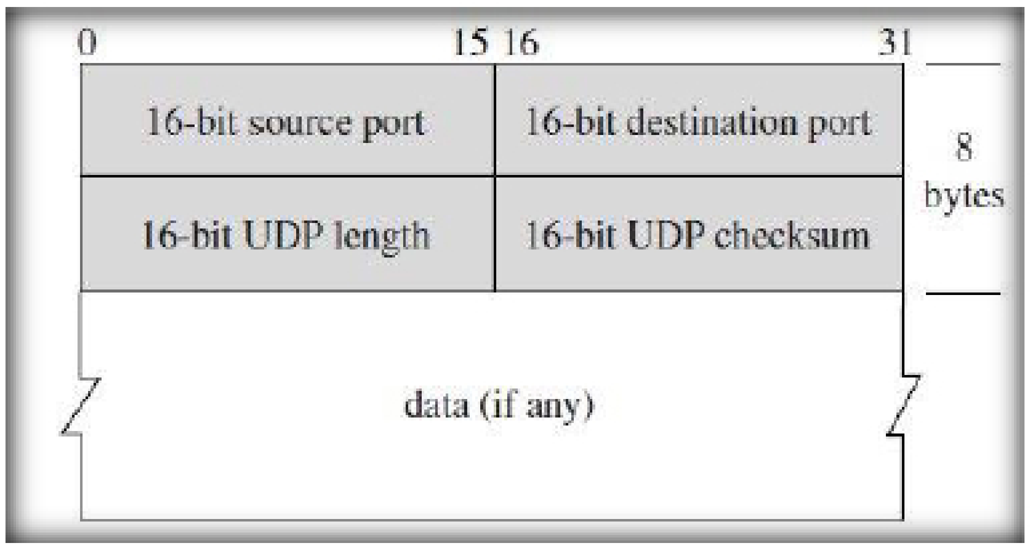


FIGURE 2.23: Typical Structure of a UDP Packet

The various fields are described as follows:

I. **Source port:** Contains the port number of the process that originated the UDP datagram

II. **Destination port:** Port number of the process towards which the datagram is destined.

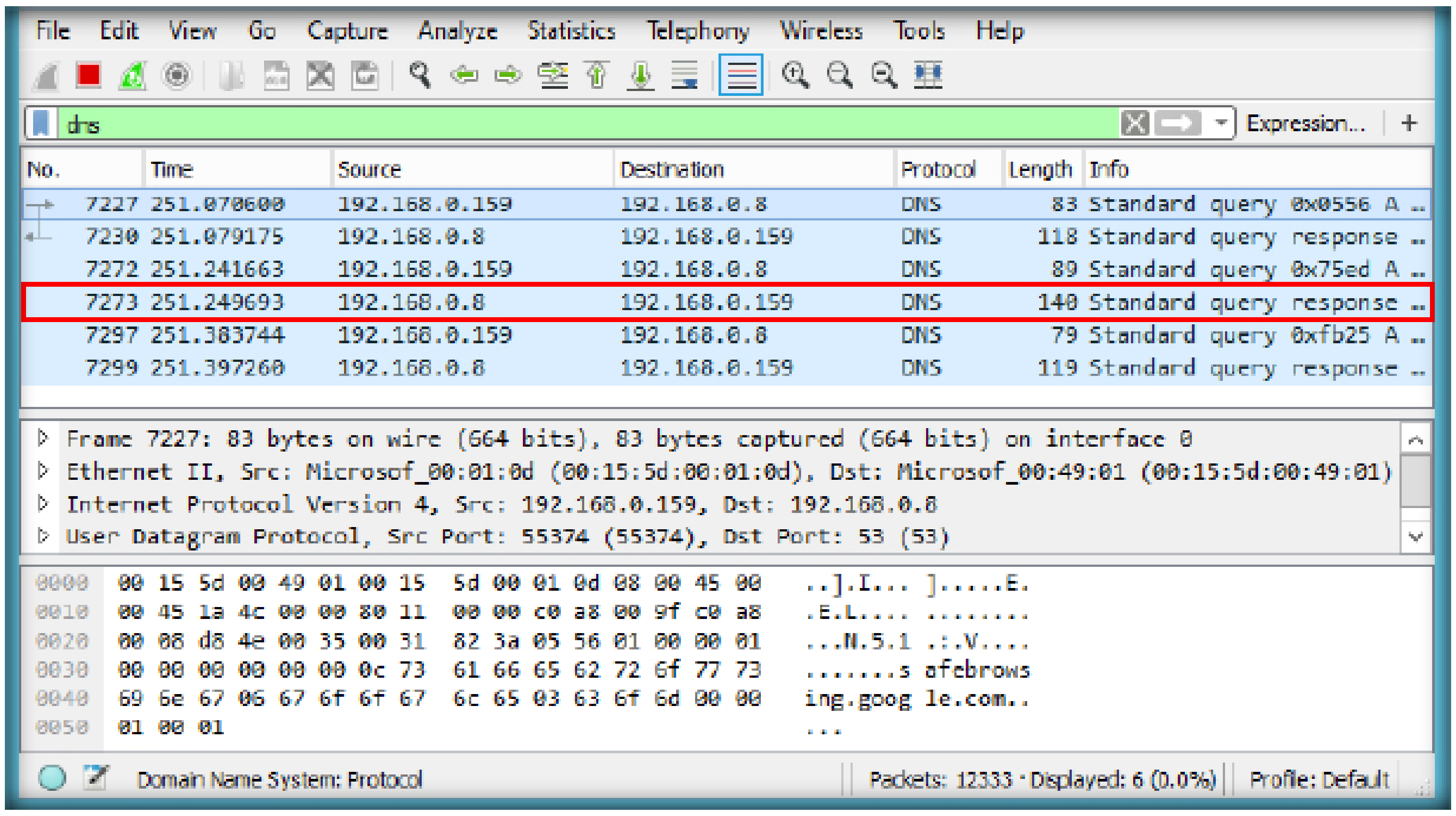
III. **UDP length:** Length of the datagram, which includes header size and data size

IV. **UDP Checksum:** An optional field used in error detection

V. **Data:** An optional field, contains higher layer message in encapsulated format

33. Since UDP works in conjunction with higher level protocols to help manage data transmission services. Common application-layer protocols that are built atop UDP are Domain Name System (DNS), Trivial File Transfer Protocol (TFTP), Real Time Streaming Protocol (RTSP), Simple Network Protocol (SNP), etc.

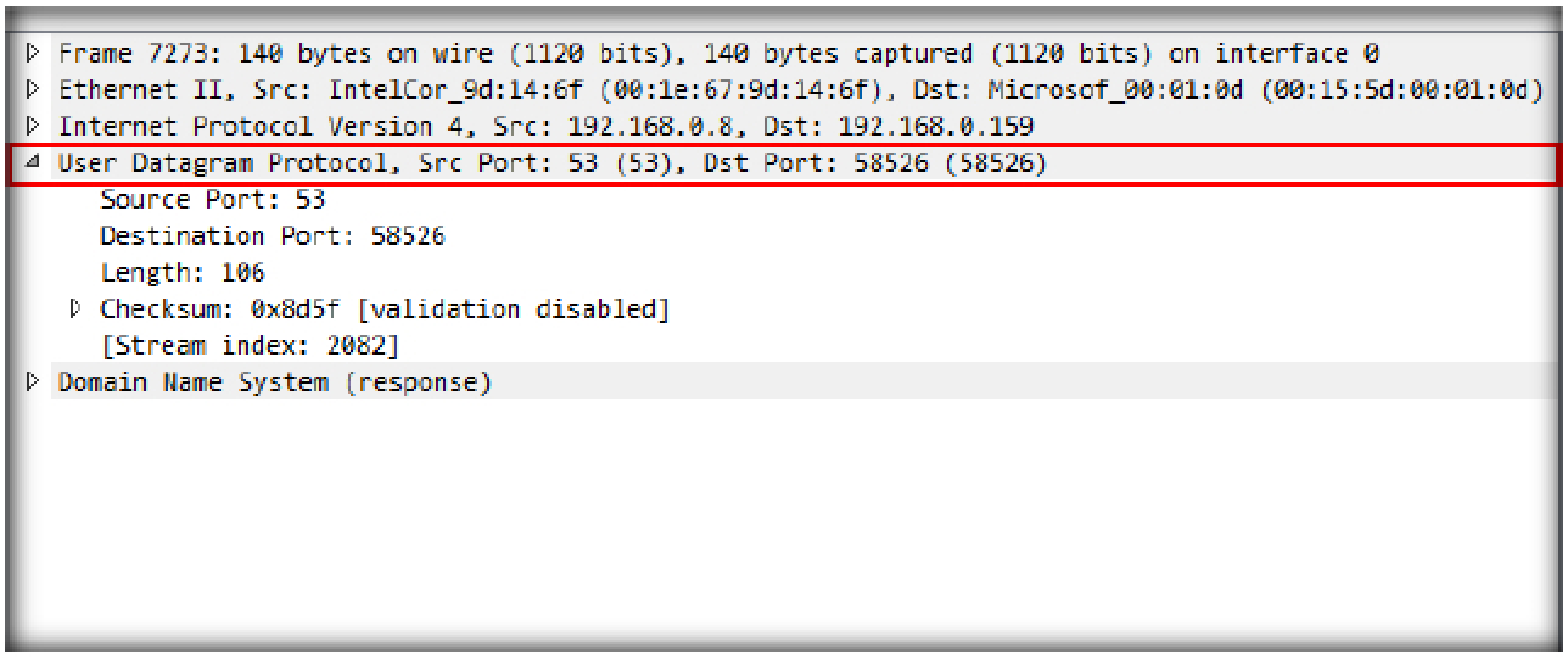
34. Now, click on a DNS packet capture to analyze various fields of an UDP header


File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
dns
Expression...
+

```
No.    Time        Source          Destination     Protocol  Length Info
 7227 251.070600  192.168.0.159   192.168.0.8     DNS         83 Standard query 0x0556 A …
 7230 251.079175  192.168.0.8     192.168.0.159   DNS        118 Standard query response …
 7272 251.241663  192.168.0.159   192.168.0.8     DNS         89 Standard query 0x75ed A …
 7273 251.249693  192.168.0.8     192.168.0.159   DNS        140 Standard query response …
 7297 251.383744  192.168.0.159   192.168.0.8     DNS         79 Standard query 0xfb25 A …
 7299 251.397260  192.168.0.8     192.168.0.159   DNS        119 Standard query response …

▷ Frame 7227: 83 bytes on wire (664 bits), 83 bytes captured (664 bits) on interface 0
▷ Ethernet II, Src: Microsof_00:01:0d (00:15:5d:00:01:0d), Dst: Microsof_00:49:01 (00:15:5d:00:49:01)
▷ Internet Protocol Version 4, Src: 192.168.0.159, Dst: 192.168.0.8
▷ User Datagram Protocol, Src Port: 55374 (55374), Dst Port: 53 (53)

0000  00 15 5d 00 49 01 00 15  5d 00 01 0d 08 00 45 00   ..].I... ].....E.
0010  00 45 1a 4c 00 00 80 11  00 00 c0 a8 00 9f c0 a8   .E.L.... ........
0020  00 08 d8 4e 00 35 00 31  82 3a 05 56 01 00 00 01   ...N.5.1 .:.V....
0030  00 00 00 00 00 00 0c 73  61 66 65 62 72 6f 77 73   .......s afebrows
0040  69 6e 67 06 67 6f 6f 67  6c 65 03 63 6f 6d 00 00   ing.goog le.com..
0050  01 00 01                                           ...

Domain Name System: Protocol        Packets: 12333 · Displayed: 6 (0.0%)    Profile: Default
```

FIGURE 2.24: DNS Packets in Wireshark

35. Expand the **User Datagram Protocol** node in the Packet Details Pane



```
▷ Frame 7273: 140 bytes on wire (1120 bits), 140 bytes captured (1120 bits) on interface 0
▷ Ethernet II, Src: IntelCor_9d:14:6f (00:1e:67:9d:14:6f), Dst: Microsof_00:01:0d (00:15:5d:00:01:0d)
▷ Internet Protocol Version 4, Src: 192.168.0.8, Dst: 192.168.0.159
◢ User Datagram Protocol, Src Port: 53 (53), Dst Port: 58526 (58526)
    Source Port: 53
    Destination Port: 58526
    Length: 106
  ▷ Checksum: 0x8d5f [validation disabled]
    [Stream index: 2082]
▷ Domain Name System (response)
```

FIGURE 2.25: Captured UDP Packet

36. Compare and analyze the various fields in an UDP packet with the UDP header format

Lab Analysis

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

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

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