

CHAPTER 17

NETWORK TRAFFIC MONITORING

CERTIFIED CYBERSECURITY TECHNICIAN

INDEX

Chapter 17:

Network Traffic Monitoring

Exercise 1:

Configure, View, and Analyze Windows Event Logs

05

Exercise 2:

View and Analyze Windows Logs

18

Exercise 3:

View and Analyze Linux Logs

23

SCENARIO

Network monitoring helps security professionals identify possible issues before they affect business continuity. If an issue occurs in the network, the root cause can be determined easily through network monitoring. Subsequently, using network automation tools, the problem can be fixed automatically. Networking monitoring not only prevents outages but also provides visibility to potential issues. Continuous network monitoring minimizes downtime and increases the performance of the network.

Even when security tools are in place, attackers can find ways to bypass such security mechanisms to enter the network. Security tools generally use signature-based detection techniques, and it is difficult to identify continuously changing attack signatures/patterns. These tools are not designed to identify behavioural anomalies and are unable to detect attackers' activities that are initiated before and during attacks.

Network monitoring tools provide the first level of security and help identify anomalous conditions in the network, that indicate attacker activity.

OBJECTIVE

The objective of this lab is to provide expert knowledge in network traffic monitoring. This includes knowledge of the following tasks:

- Intercepting network traffic using various tools such as Wireshark and tcpdump
- Exploring various filters in Wireshark
- Analyzing and examining various network packet headers in Linux using tools such as tcpdump
- Performing scan on network to identify machines in the local network

OVERVIEW OF NETWORK TRAFFIC MONITORING

Network monitoring is a retrospective security approach that involves monitoring a network for abnormal activities, performance issues, bandwidth issues, etc. It is an integral part of network security and is a demanding task within the network security operations of organizations. Continuous network traffic monitoring and analysis are critical for effective threat detection.

A proper analysis of log data enables actionable information to be identified, which helps the security professional in detecting and monitoring potential security breaches, internal misuse of information, operational issues, and other long-term issues. It also helps validate whether the end-user has followed all documented protocols to detect fraudulent activities and policy violations. It is also useful for internal investigations, security auditing and forensic analysis, determination of operational trends, and implementation of baselines.

LAB TASKS

Cyber security professional or a security professional use numerous tools and techniques to monitor network traffic. The recommended labs that will assist you in learning various aspects of network traffic monitoring include the following:

01 Intercept Network Traffic using Wireshark and tcpdump

03 Analyze and Examine Various Network Packet Headers in Linux using tcpdump

02 Apply Various Filters in Wireshark

04 Scan Network to Identify Hosts in the Local Network

Note: Turn on PfSense Firewall virtual machine and keep it running throughout the lab exercises.

EXERCISE 1: INTERCEPT NETWORK TRAFFIC USING WIRESHARK AND tcpdump

Network traffic monitoring is the process of capturing network traffic and inspecting it closely to determine what is happening on the network.

LAB SCENARIO

A security professional must have the required knowledge to intercept and interpret network traffic using various packet sniffing tools. The captured traffic can be used to identify malicious or suspicious packets hiding within traffic.

OBJECTIVE

This lab will demonstrate how to capture network traffic using Wireshark and tcpdump.

OVERVIEW OF TROJAN

The network monitoring process involves sniffing the traffic flowing through the network. For this purpose, network packets must be captured, and a signature analysis must be conducted to identify any malicious activity. Security professionals should constantly strive to maintain smooth network operation by monitoring network traffic. If the network goes down even for a small period, productivity within a company may decline. To be proactive rather than reactive, the traffic movement and performance must be monitored to ensure that no security breach occurs within the network.

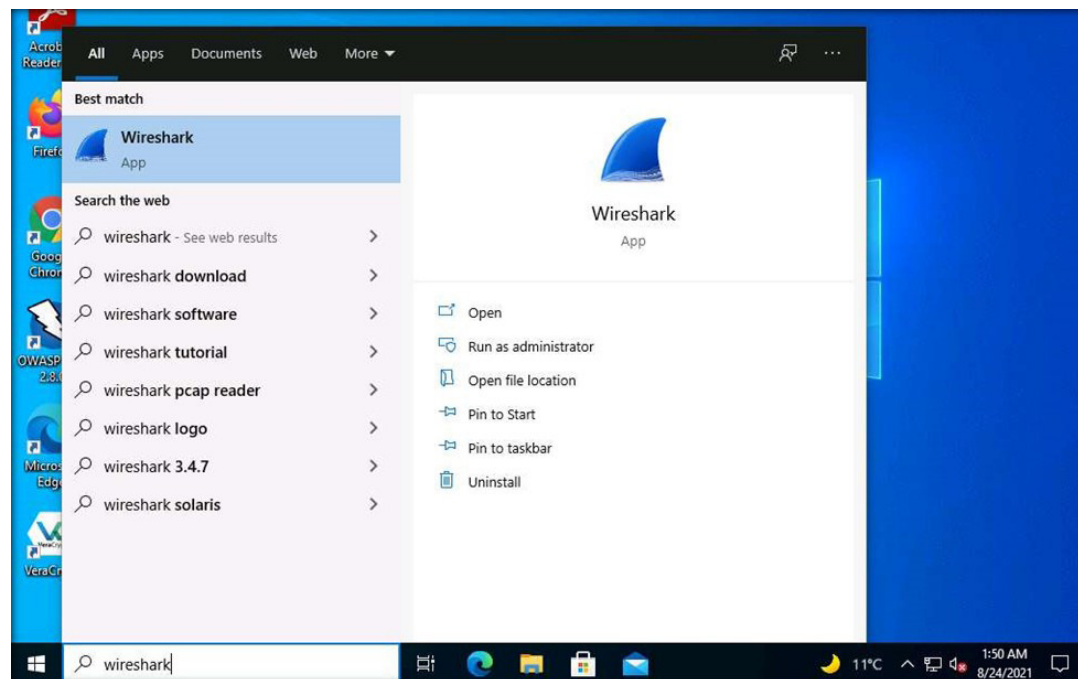
Note: Ensure that the PfSense Firewall virtual machine is running.

1. Turn on Admin Machine-1, Web Server and Attacker Machine-2 virtual machines.
2. Log in with the credentials Admin and admin@123.

Note: If the network screen appears, click Yes.

3. Click on Type here to search field at the bottom right of the Desktop, type wireshark and select Wireshark from the results. The Wireshark app will appear. Click to open the Wireshark.

EXERCISE 1:
INTERCEPT
NETWORK TRAFFIC
USING WIRESHARK
AND TCPDUMP



4. The main window of Wireshark appears.

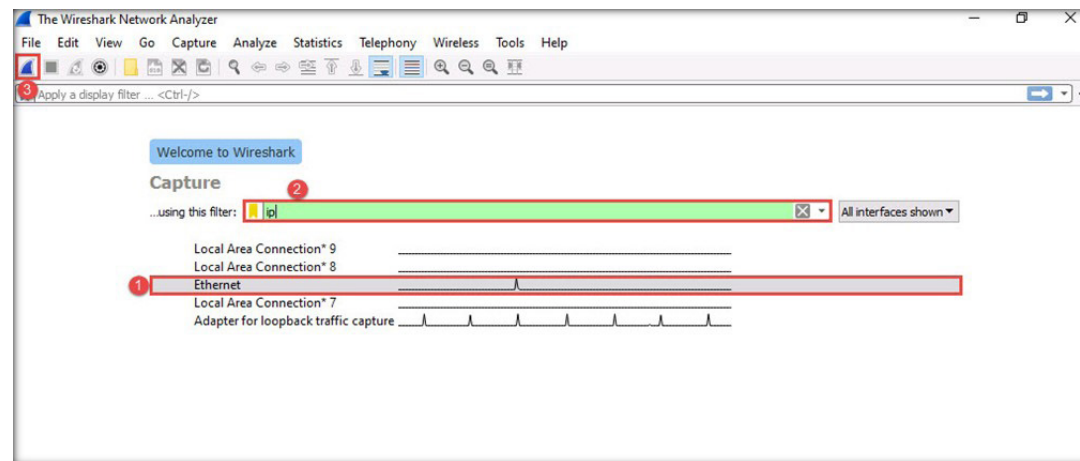
Note: If Software Update Window appears, click on Skip this version.

5. Next, you need to select the interface of which you want Wireshark to capture traffic. To begin packet capture, select the Ethernet interface from the list and in the Enter a captured filter... field, enter ip.

Note: The ip filter captures only IPv4 traffic not IPV6.

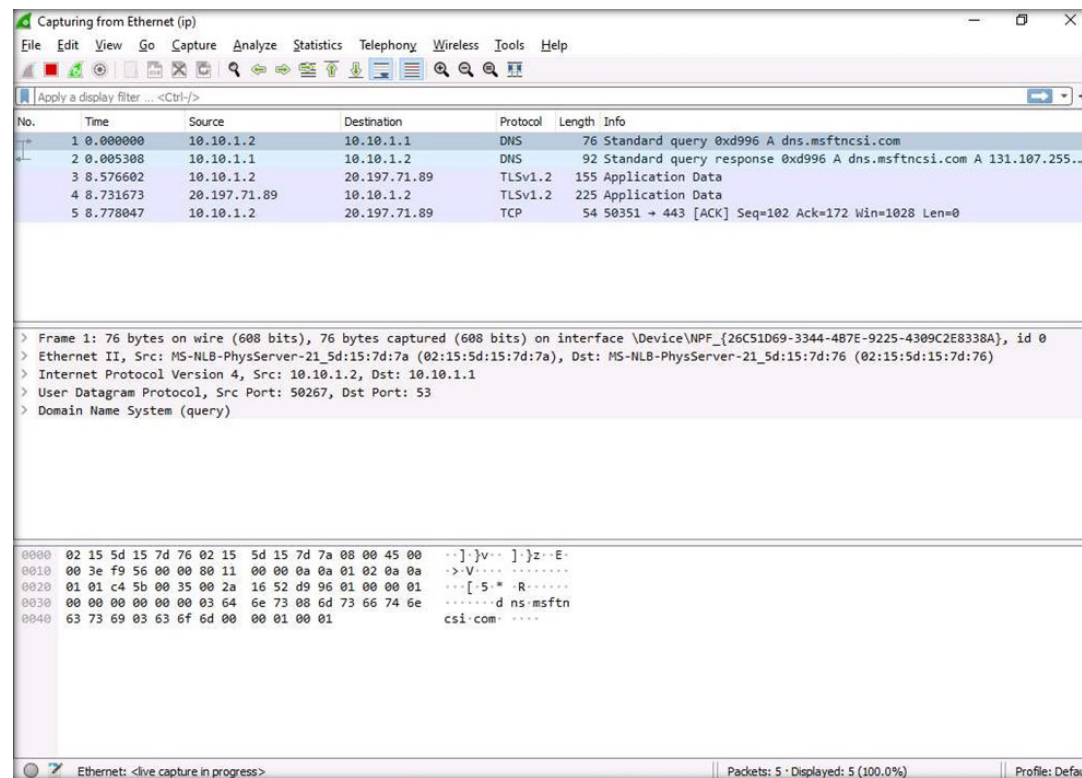
6. Now, to begin packet capturing, click Start capturing packets icon (blue color shark fin icon) from the tool bar.

EXERCISE 1:
INTERCEPT
NETWORK TRAFFIC
USING WIRESHARK
AND TCPDUMP



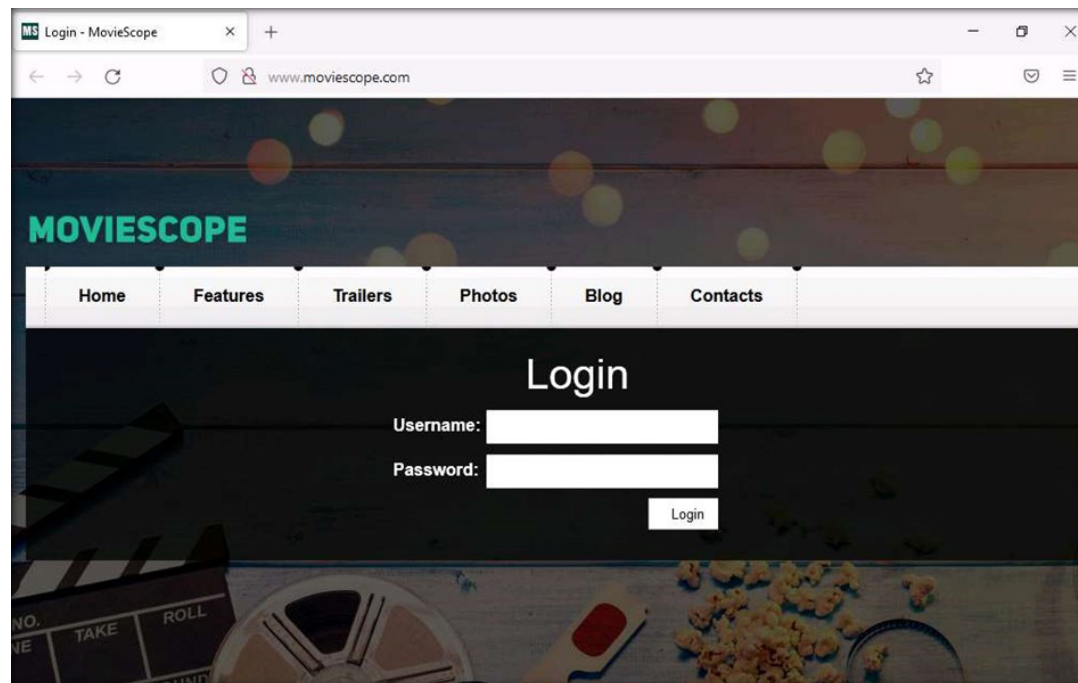
7. Wireshark begins to capture the traffic of the selected interface, as shown in the screenshot below.

EXERCISE 1:
INTERCEPT
NETWORK TRAFFIC
USING WIRESHARK
AND TCPDUMP



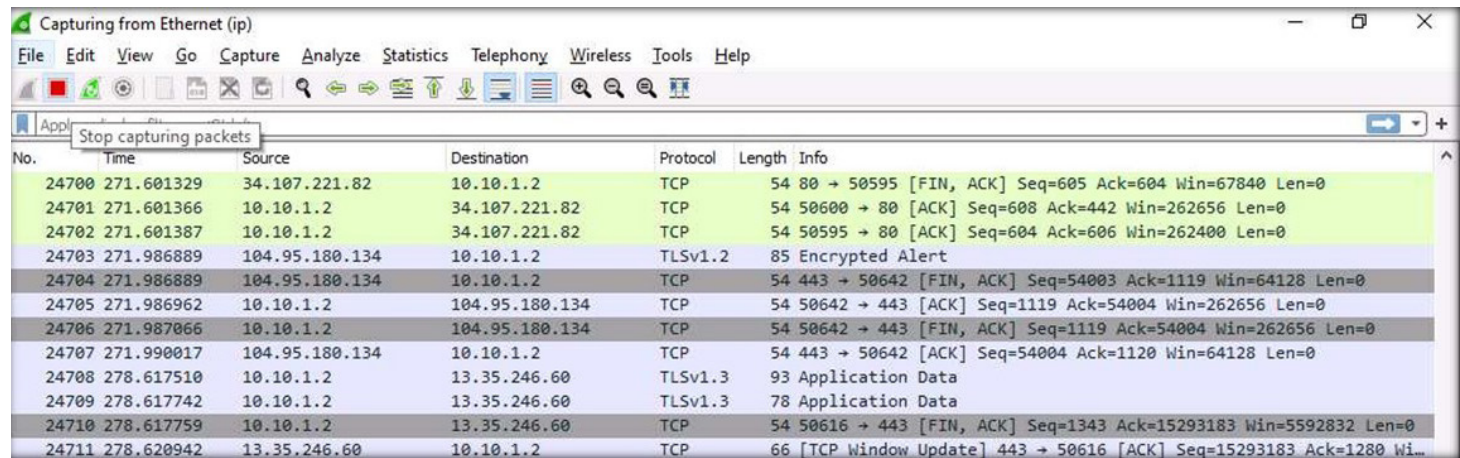
8. Minimize the Wireshark window.
9. Open any web browser (here, Mozilla Firefox) and type `http://www.moviescope.com` in the url field and press Enter.

EXERCISE 1:
INTERCEPT
NETWORK TRAFFIC
USING WIRESHARK
AND TCPDUMP



10. Now, switch back to the Wireshark window and click Stop capturing packets icon (red square icon) from the tool bar to stop capturing packets.

EXERCISE 1:
INTERCEPT
NETWORK TRAFFIC
USING WIRESHARK
AND TCPDUMP

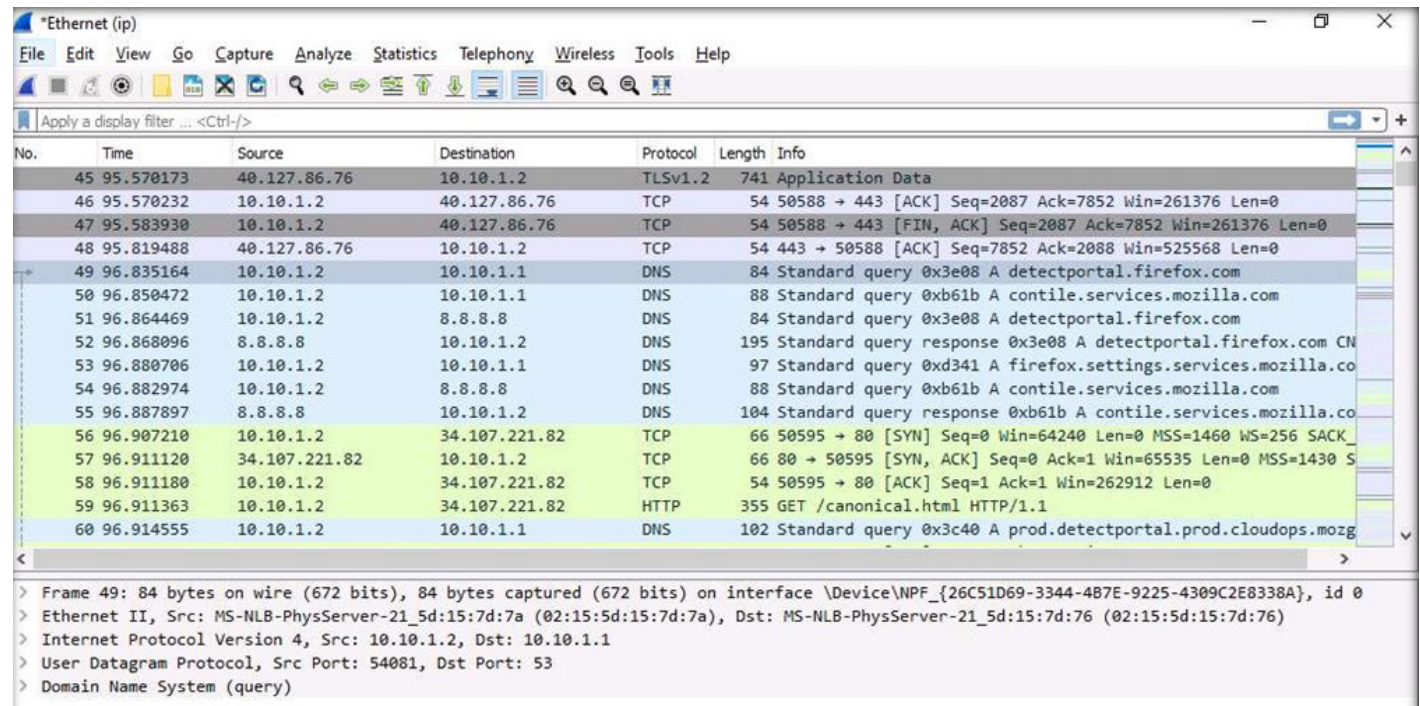


11. From the captured packets, select any DNS frame (they are light blue in colour), and observe the packet content displayed in the middle section, as shown in the screenshot below.

Note: Frame: Displays details regarding captured bytes.

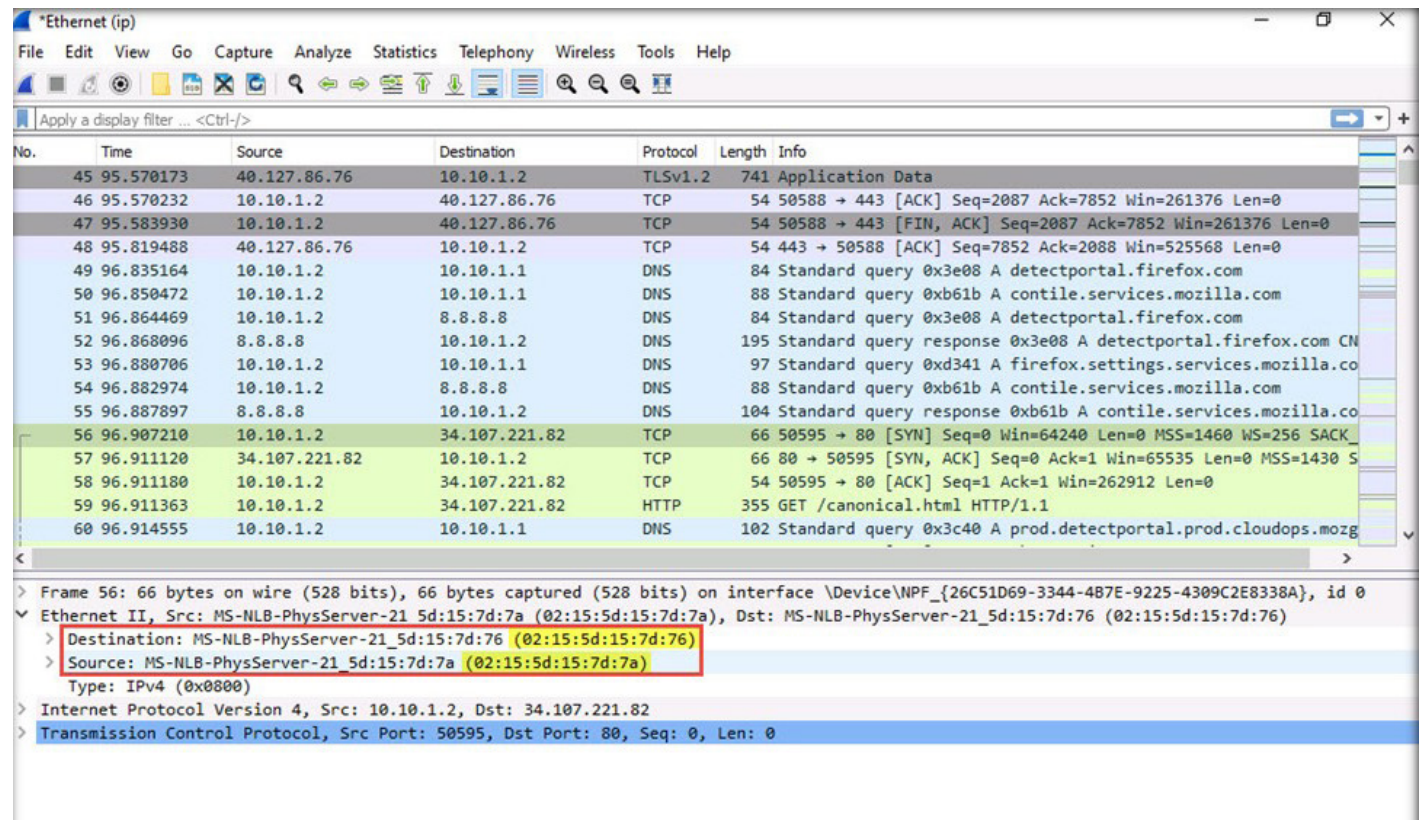
- Ethernet II: Displays details such as destination and source MAC addresses and type of network protocol used in the captured packet such as IPv4.
- Internet Protocol Version 4: Displays details such as source and destination IP addresses.
- User Datagram Protocol: Displays source and destination ports, length of the frame and checksum values.
- Domain Name System: Refers to the application protocol, there are two types of frame query and response.

EXERCISE 1:
INTERCEPT
NETWORK TRAFFIC
USING WIRESHARK
AND TCPDUMP



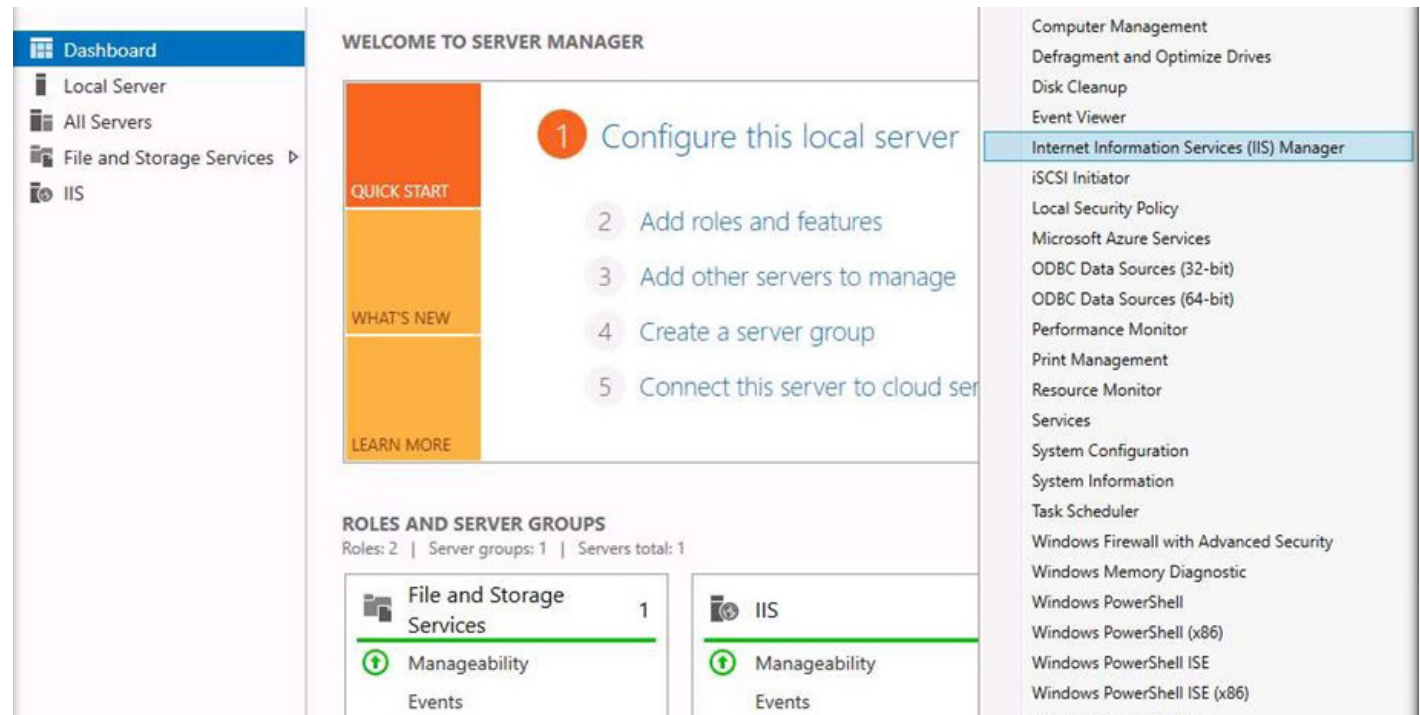
12. Now, select first TCP packet (with light green color), to observe the packet content.
13. In the middle section, you can observe source and destination MAC addresses under Ethernet II, as shown in the screenshot below.

EXERCISE 1:
INTERCEPT
NETWORK TRAFFIC
USING WIRESHARK
AND TCPDUMP



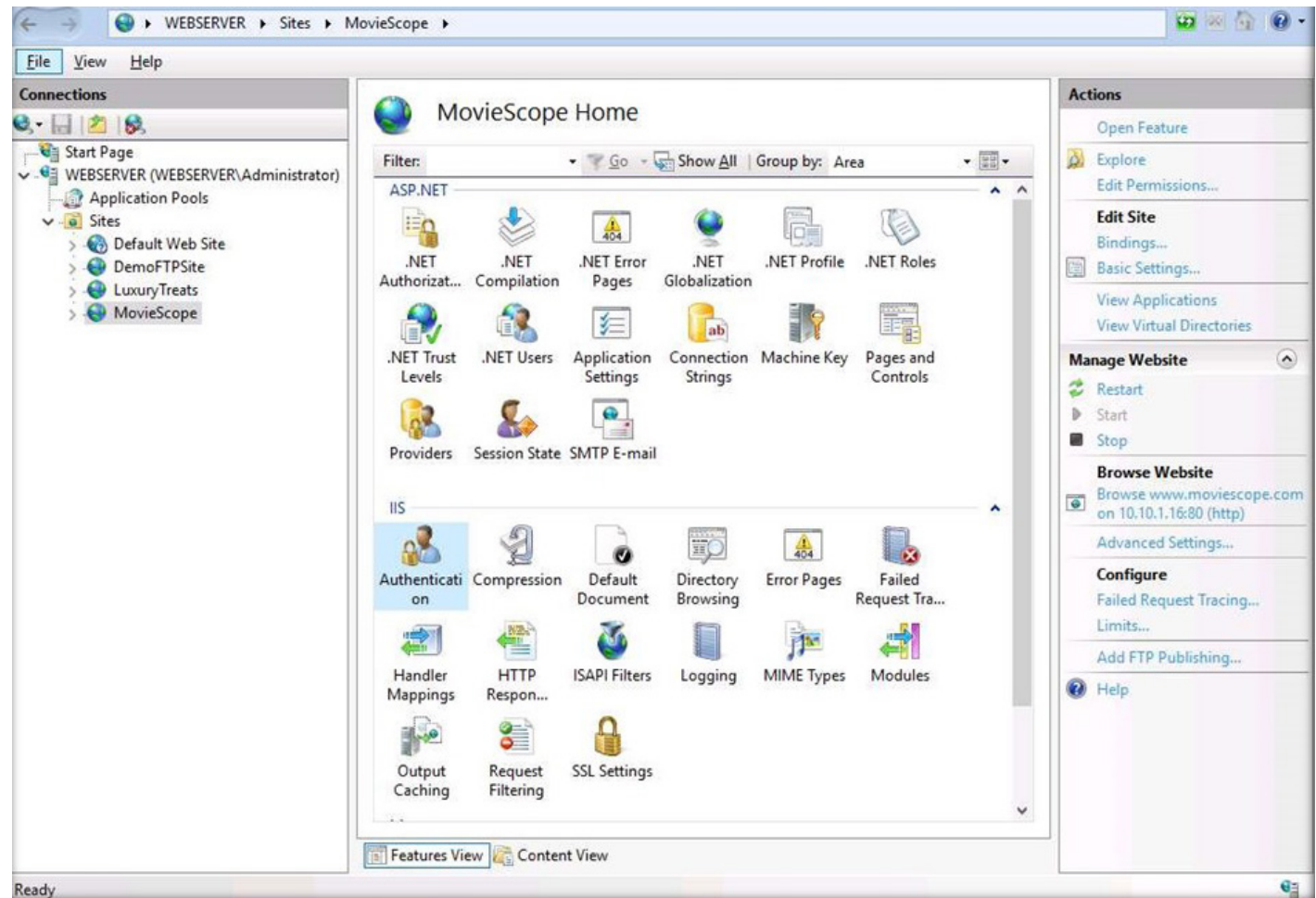
14. Similarly, you can view all the other information under different sections such as Frame, Internet Protocol Version 4, Transmission Control Protocol.
 15. Close all open windows.
 16. Now, we will use tcpdump tool to intercept HTTP traffic.
 17. Switch to the Web Server virtual machine.
 18. Log in with the credentials Administrator and admin@123.
- Note: The network screen appears, click Yes.
19. Click Start icon from the lower-left corner of the Desktop and from the options, select Server Manager.
 20. The Server Manager window appears. Click Tools and select Internet Information Services (IIS) Manager option.

EXERCISE 1:
INTERCEPT
NETWORK TRAFFIC
USING WIRESHARK
AND TCPDUMP



21. The Internet Information Services (IIS) Manager window appears; expand WEBSERVER (WEBSERVER\Administrator) node and Sites node under the Connections section from the left-hand pane. Select MovieScope site.
22. From the middle-pane, double-click on Authentication applet under IIS section.

EXERCISE 1:
INTERCEPT
NETWORK TRAFFIC
USING WIRESHARK
AND TCPDUMP

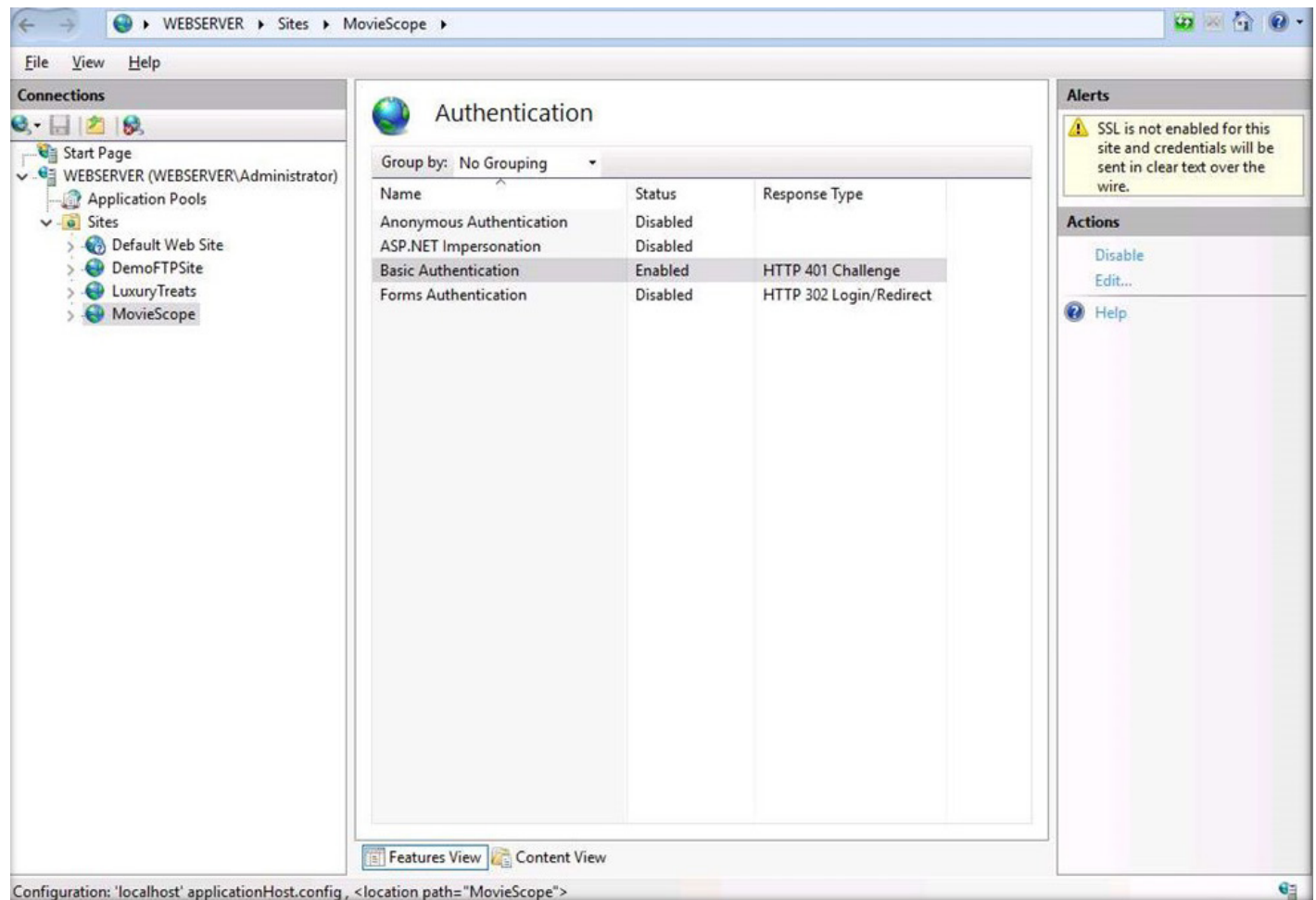


23. Authentication wizard appears, select Anonymous Authentication and click Disable from the right-pane under Actions section.

24. Similarly, select Basic Authentication and click Enable from the right-pane under Actions section.

Note: For demonstration purposes, here, we are using Basic authentication mechanism where plaintext credentials are used to authenticate and access the website which is not a safe practice. In real practice, it is advised to use Windows authentication which is considerably more secure than basic authentication.

EXERCISE 1:
INTERCEPT
NETWORK TRAFFIC
USING WIRESHARK
AND TCPDUMP



25. Switch to the Attacker Machine-2 virtual machine.

26. 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.

Note: If a Question pop-up window appears asking you to update the machine, click No to close the window.

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

28. A Parrot Terminal window appears. In the terminal window, type `sudo su` and press Enter to run the programs as a root user.

29. In the `[sudo]` password for attacker field, type `toor` as a password and press Enter.

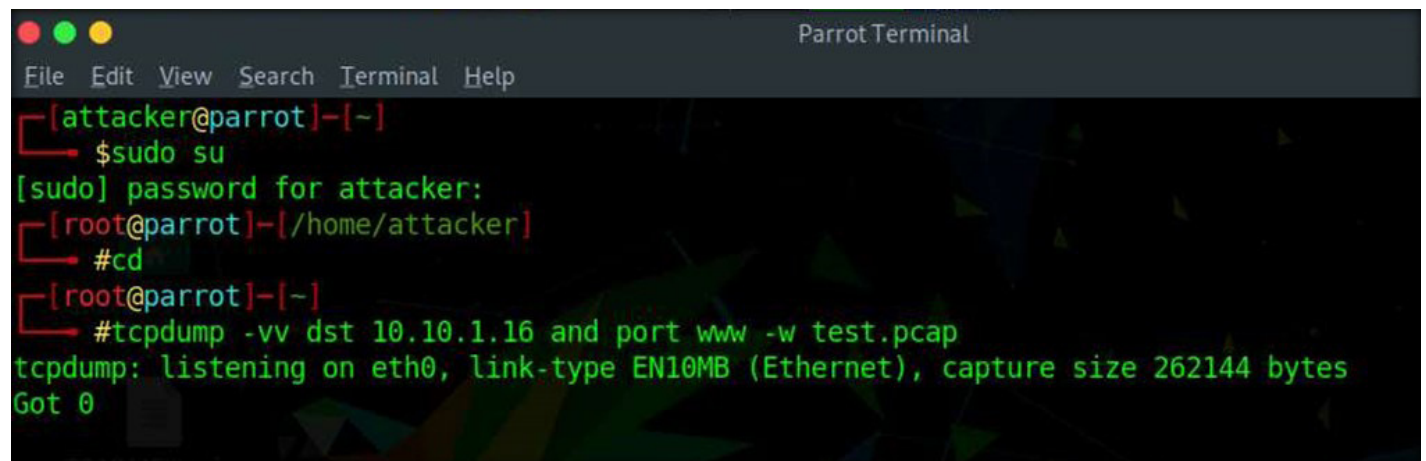
Note: The password that you type will not be visible.

30. Now, type `cd` and press Enter to jump to the root directory.

31. In the Terminal window, type `tcpdump -vv dst 10.10.1.16 and port www -w test.pcap` and press Enter to capture HTTP traffic of the target machine Web Server (10.10.1.16).

Note: `--vv`: Indicate a verbose output, `dst`: Indicate the destination, `-w`: To write raw packets to a file (here, `test.pcap`)

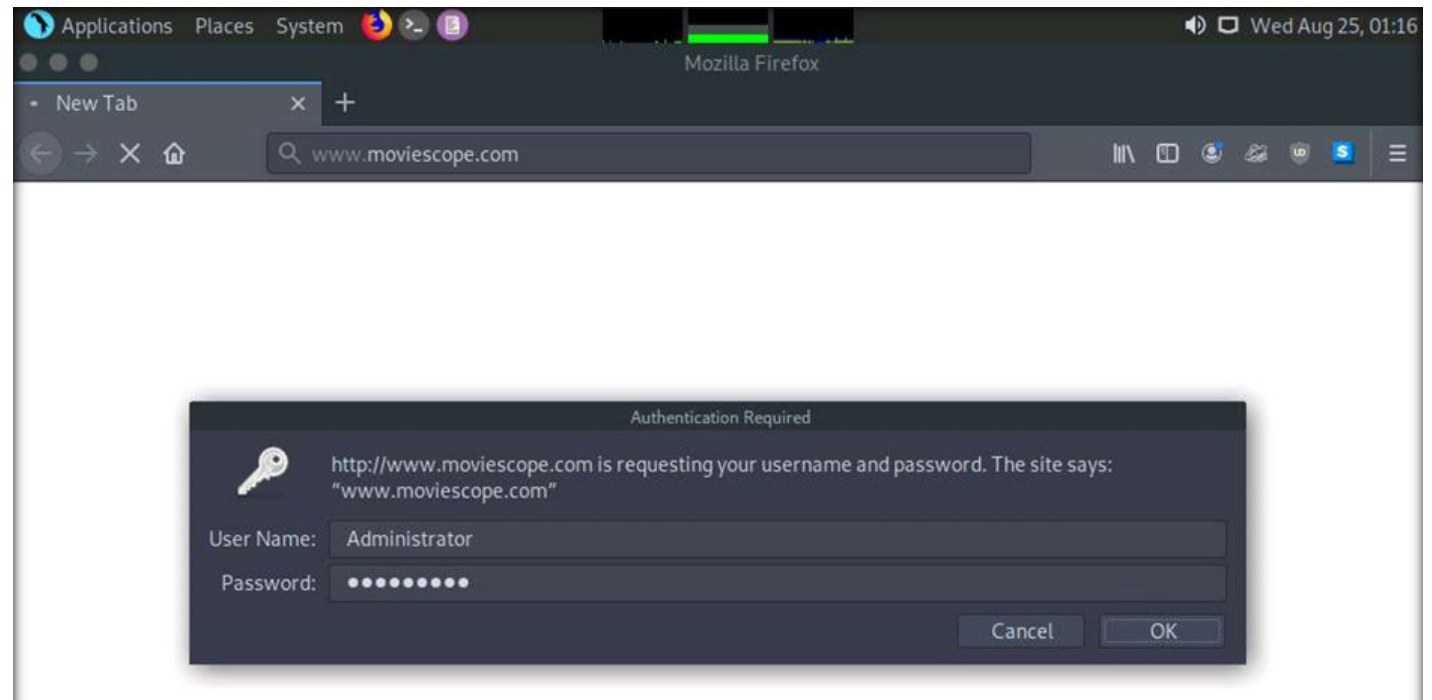
32. The `tcpdump` starts listening on `eth0` interface to capture HTTP packets, as shown in the screenshot below.



```
Parrot Terminal
File Edit View Search Terminal Help
[attacker@parrot]-[~]
└─$ sudo su
[sudo] password for attacker:
[root@parrot]-[/home/attacker]
└─# cd
[root@parrot]-[~]
└─# tcpdump -vv dst 10.10.1.16 and port www -w test.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
Got 0
```

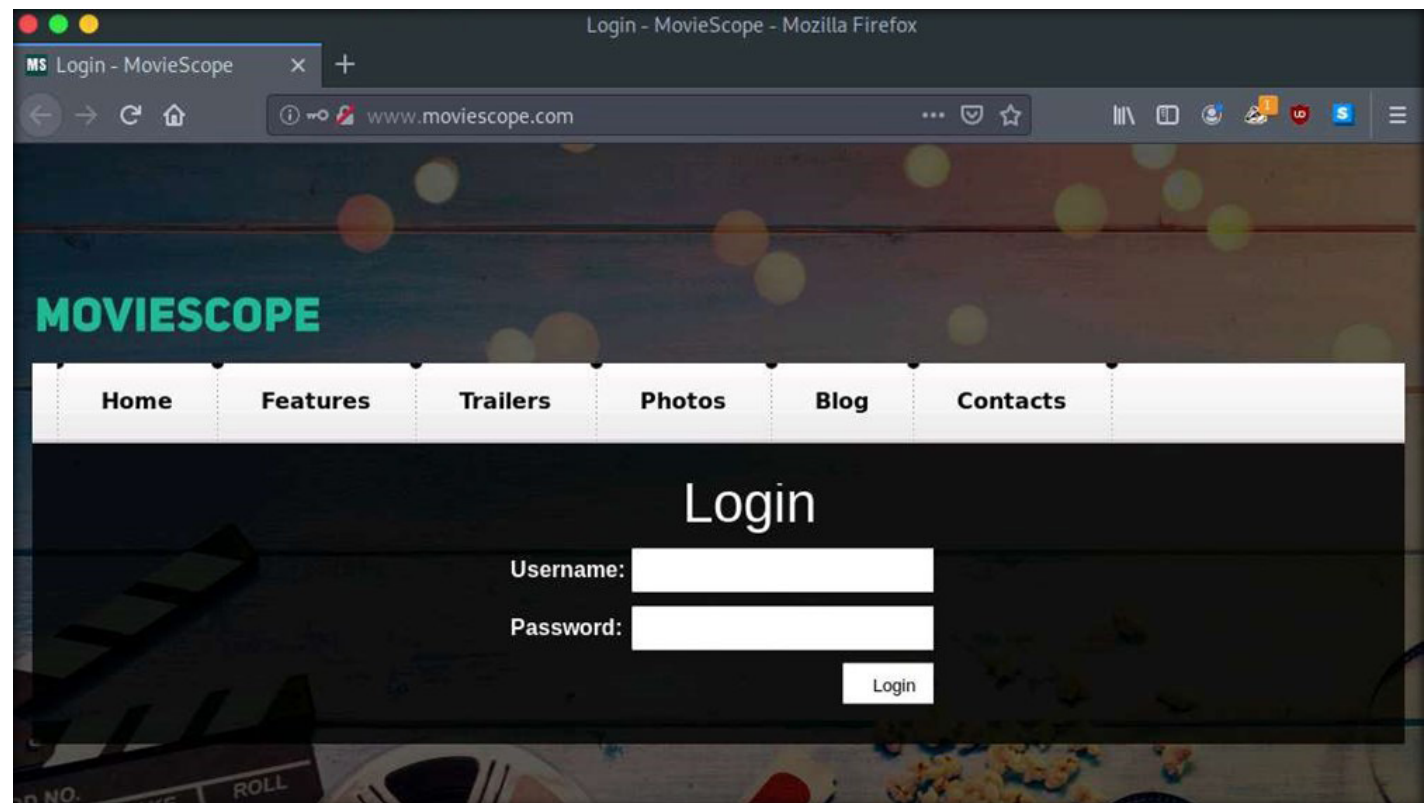

33. Click the Firefox icon from the top section of Desktop to launch the Mozilla Firefox browser.
 34. The Mozilla Firefox window appears; type `http://www.moviescope.com` into the address bar and press Enter.
 35. Authentication Required pop-up appears; type Administrator and `admin@123` as User Name and Password and click OK.
- Note: If Would you like Firefox to save this login for `moviescope.com`? pop-up appears, click Don't Save.

EXERCISE 1:
INTERCEPT
NETWORK TRAFFIC
USING WIRESHARK
AND TCPDUMP



36. You have now logged successfully to access the website, as shown in the screenshot below.

EXERCISE 1:
INTERCEPT
NETWORK TRAFFIC
USING WIRESHARK
AND TCPDUMP



37. Switch to terminal window and press Ctrl+C to terminate intercepting network traffic.

EXERCISE 1: INTERCEPT NETWORK TRAFFIC USING WIRESHARK AND TCPDUMP

```
Parrot Terminal
File Edit View Search Terminal Help
[attacker@parrot]-[~]
└─$ sudo su
[sudo] password for attacker:
[root@parrot]-[/home/attacker]
└─# cd
[root@parrot]-[~]
└─# tcpdump -vv dst 10.10.1.16 and port www -w test.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
^C73 packets captured
73 packets received by filter
0 packets dropped by kernel
[root@parrot]-[~]
└─#
```

- 38. Type ls and press Enter.
- 39. You can observe a file name test.pcap has been created in the /root directory.

EXERCISE 1:
INTERCEPT
NETWORK TRAFFIC
USING WIRESHARK
AND TCPDUMP

```
^C73 packets captured
73 packets received by filter
0 packets dropped by kernel
[root@parrot]-[~]
└─# ls
avml Desktop lazys3 social-engineer-toolkit test.pcap
buck-security-master DSSS shellphish Templates volatility-master
[root@parrot]-[~]
└─#
```

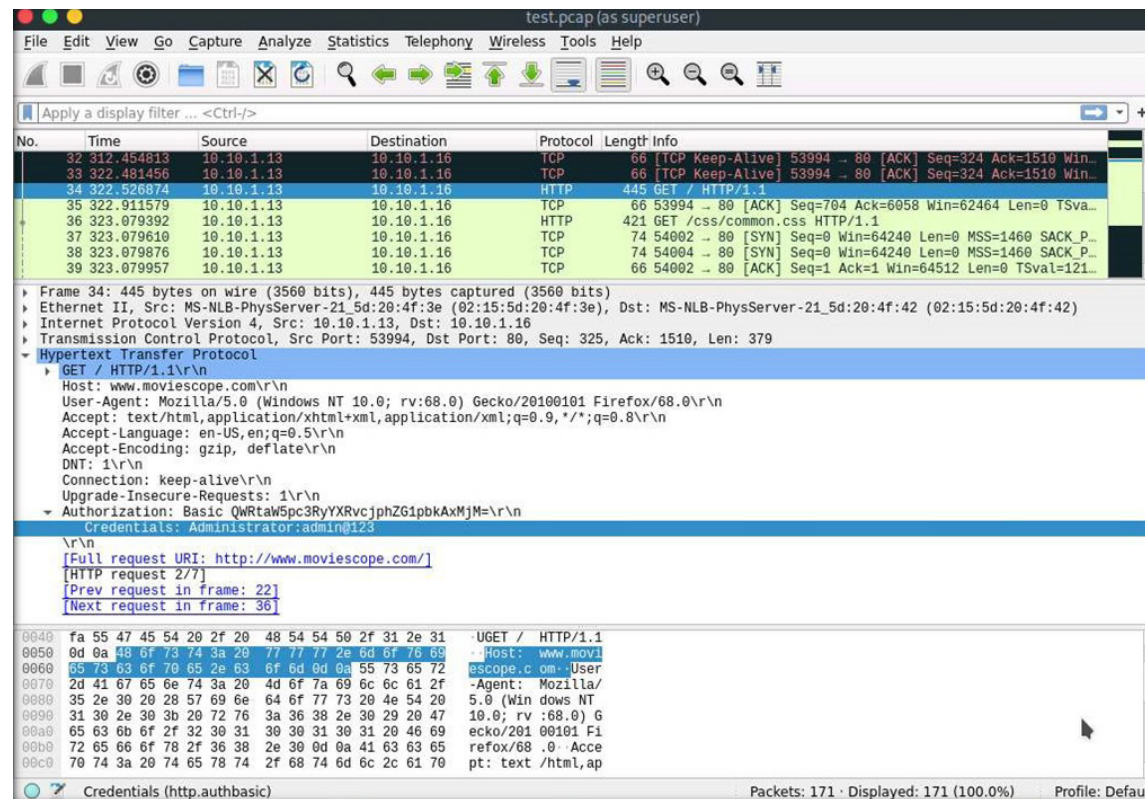
40. Now, type `wireshark test.pcap` and press Enter to open the file using Wireshark.

EXERCISE 1: INTERCEPT NETWORK TRAFFIC USING WIRESHARK AND TCPDUMP

```
[root@parrot]-[~]
└─# ls
avml          Desktop      lazys3       social-engineer-toolkit  test.pcap
buck-security-master  DSS         shellphish   Templates                volatility-master
└─# wireshark test.pcap
```

41. Click to select any HTTP message with GET request.
42. From the middle-pane, expand Hypertext Transfer Protocol node. Under Hypertext Transfer Protocol node, expand Authorization node. Note: If you do not see Authorization node in the first block of HTTP packets then select HTTP GET packet from a different block of HTTP packets.
43. You can observe that the credentials are displayed because the HTTP packets are unencrypted which makes them vulnerable to packet sniffing. Close the Wireshark window.

EXERCISE 1:
INTERCEPT
NETWORK TRAFFIC
USING WIRESHARK
AND TCPDUMP



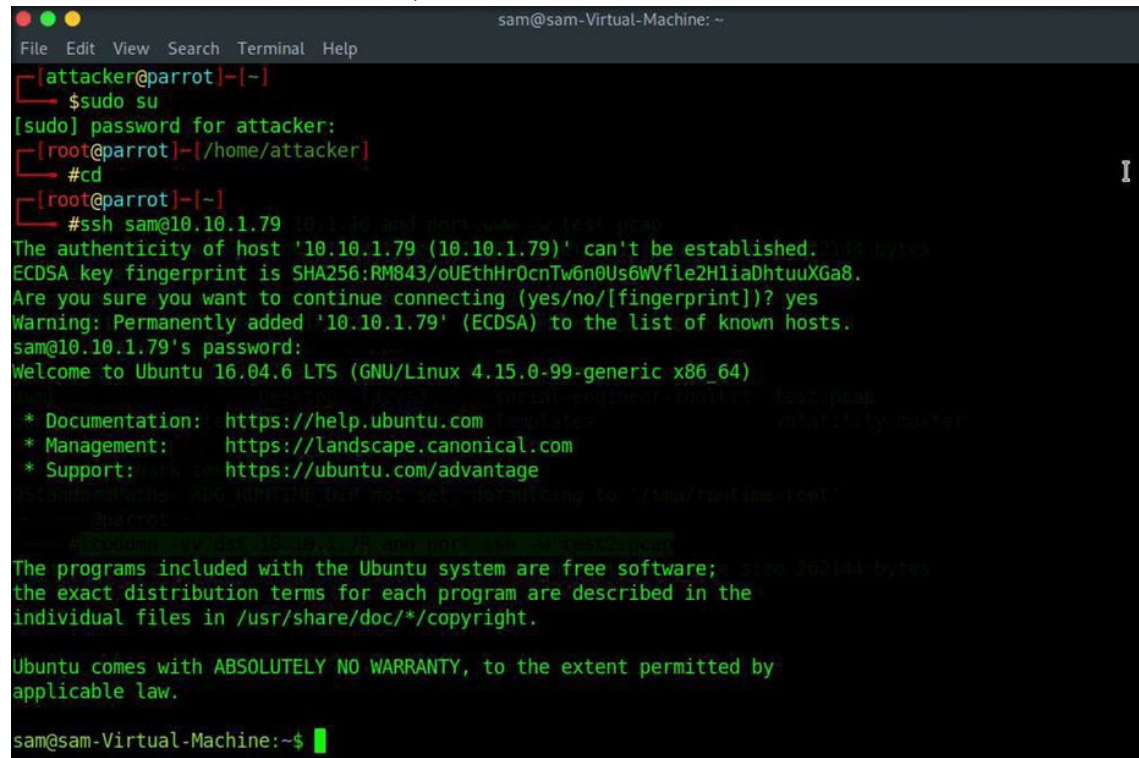
44. Now, in a similar way, we will intercept SSH traffic and observe the packet content.
45. In the terminal window, type `tcpdump -vv dst 10.10.1.79 and port ssh -w test2.pcap` and press Enter to capture SSH traffic to the target machine Admin Machine-2 (10.10.1.79).
46. The `tcpdump` starts listening on `eth0` interface to capture SSH traffic, as shown in the screenshot below.

```
[root@parrot]~# #wireshark test.pcap
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-root'
[root@parrot]~# #tcpdump -vv dst 10.10.1.79 and port ssh -w test2.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
Got 0
```

EXERCISE 1: INTERCEPT NETWORK TRAFFIC USING WIRESHARK AND TCPDUMP

47. Now, click the MATE Terminal icon at the top of the Desktop window to open another Terminal window.
 48. A Parrot Terminal window appears. In the terminal window, type `sudo su` and press Enter to run the programs as a root user.
 49. 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.
50. Now, type `cd` and press Enter to jump to the root directory.
 51. Type `ssh sam@10.10.1.79` and press Enter to establish SSH connection with Admin Machine-2.
- Note:** If connection attempt prompt appears, type `yes` and press Enter.
52. In the password field, type `admin@123` and press Enter.
 53. You can observe that a remote connection has been established, as shown in the screenshot below.

EXERCISE 1:
INTERCEPT
NETWORK TRAFFIC
USING WIRESHARK
AND TCPDUMP



```

sam@sam-Virtual-Machine: ~
File Edit View Search Terminal Help
[attacker@parrot]-[~]
└─$ sudo su
[sudo] password for attacker:
[root@parrot]-[/home/attacker]
└─# cd
[root@parrot]-[~]
└─# ssh sam@10.10.1.79
The authenticity of host '10.10.1.79 (10.10.1.79)' can't be established.
ECDSA key fingerprint is SHA256:RM843/oUEthHr0cnTw6n0Us6WVfle2HliaDhtuuXGa8.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.1.79' (ECDSA) to the list of known hosts.
sam@10.10.1.79's password:
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.15.0-99-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

sam@sam-Virtual-Machine:~$
    
```


54. Type exit and press Enter to terminate the connection.

EXERCISE 1: INTERCEPT NETWORK TRAFFIC USING WIRESHARK AND TCPDUMP

```
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.  
  
sam@sam-Virtual-Machine:~$ exit  
logout  
Connection to 10.10.1.79 closed.  
[root@parrot]-[~]  
#
```

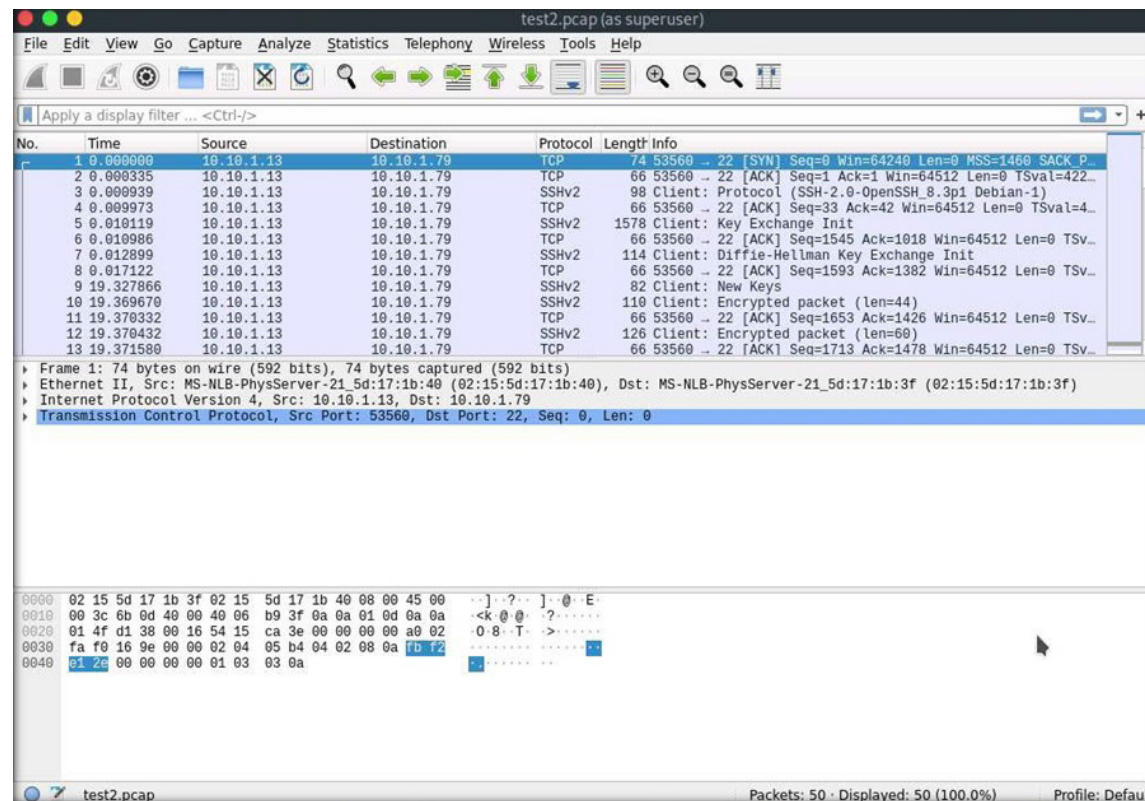
55. Now, switch back to the previous terminal window and press Ctrl+C to terminate packet capturing by tcpdump.

EXERCISE 1: INTERCEPT NETWORK TRAFFIC USING WIRESHARK AND TCPDUMP

```
Parrot Terminal
File Edit View Search Terminal Help
[root@parrot]~# #wireshark test.pcap
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-root'
[root@parrot]~# #tcpdump -vv dst 10.10.1.79 and port ssh -w test2.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
^C50 packets captured
50 packets received by filter (0 discarded), to the extent permitted by
0 packets dropped by kernel
[root@parrot]~# #
```

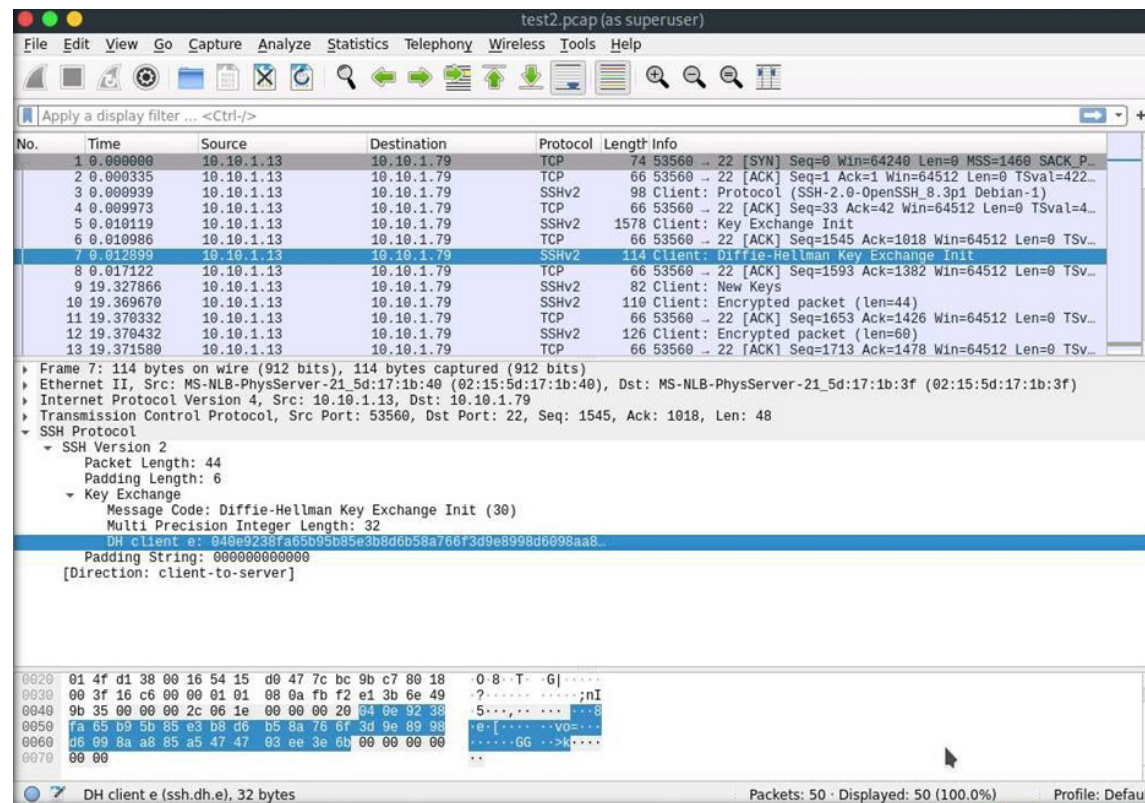
- 56. Type `wireshark test2.pcap` and press Enter to open the captured packet file using Wireshark.
- 57. The Wireshark window appears, displaying captured packets, as shown in the screenshot below.

EXERCISE 1:
INTERCEPT
NETWORK TRAFFIC
USING WIRESHARK
AND TCPDUMP



58. Click to select SSHv2 protocol packet with Info as Client: Diffie-Hellman Key Exchange Init.
59. In the middle-pane expand SSH Protocol node. Under SSH Protocol node, expand SSH Version 2 node and Key Exchange node.
60. You can observe that the captured password is in encrypted form, as shown in DH client e option.

EXERCISE 1:
INTERCEPT NETWORK TRAFFIC USING WIRESHARK AND TCPDUMP



61. This concludes the demonstration showing how to intercept network traffic using various packet sniffing tools.
62. Close all open windows.
63. Turn off the Attacker Machine-2 virtual machine.

EXERCISE 1: INTERCEPT NETWORK TRAFFIC USING WIRESHARK AND TCPDUMP

EXERCISE 2: APPLY VARIOUS FILTERS IN WIRESHARK

Wireshark provides numerous filters that can be applied to obtain only the required packets.

LAB SCENARIO

Wireshark filters traffic flowing through the entire network. This traffic contains various kinds of data packets associated with various protocols flowing between the source and destination. Therefore, searching for a specific packet, port, or an IP address manually is extremely difficult. In such cases, applying Wireshark filters helps a security professional track down a huge amount of traffic and discover the intended packets. A security professional must have a good knowledge of various Wireshark filters that help you narrow down the traffic and obtain the desired result.

OBJECTIVE

This lab will help you become familiar with various Wireshark filters.

OVERVIEW OF TROJAN

Wireshark has various filters that help you filter packets containing the following:

- Source IP address
- Destination IP address
- Internet Control Message Protocol (ICMP) traffic etc.

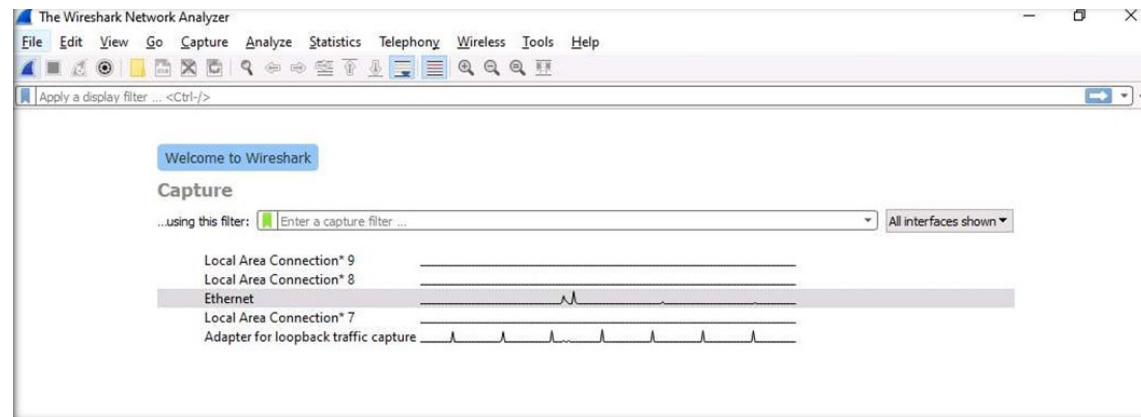
Note: Ensure that Admin Machine-1, Web Server and PfSense Firewall virtual machines are running.

1. In the Admin Machine-1 virtual machine, click Type here to Search field and type Wireshark. Select and open the Wireshark App.
2. The Wireshark main window appears.

Note: If Software Update Window appears, click on Skip this version.

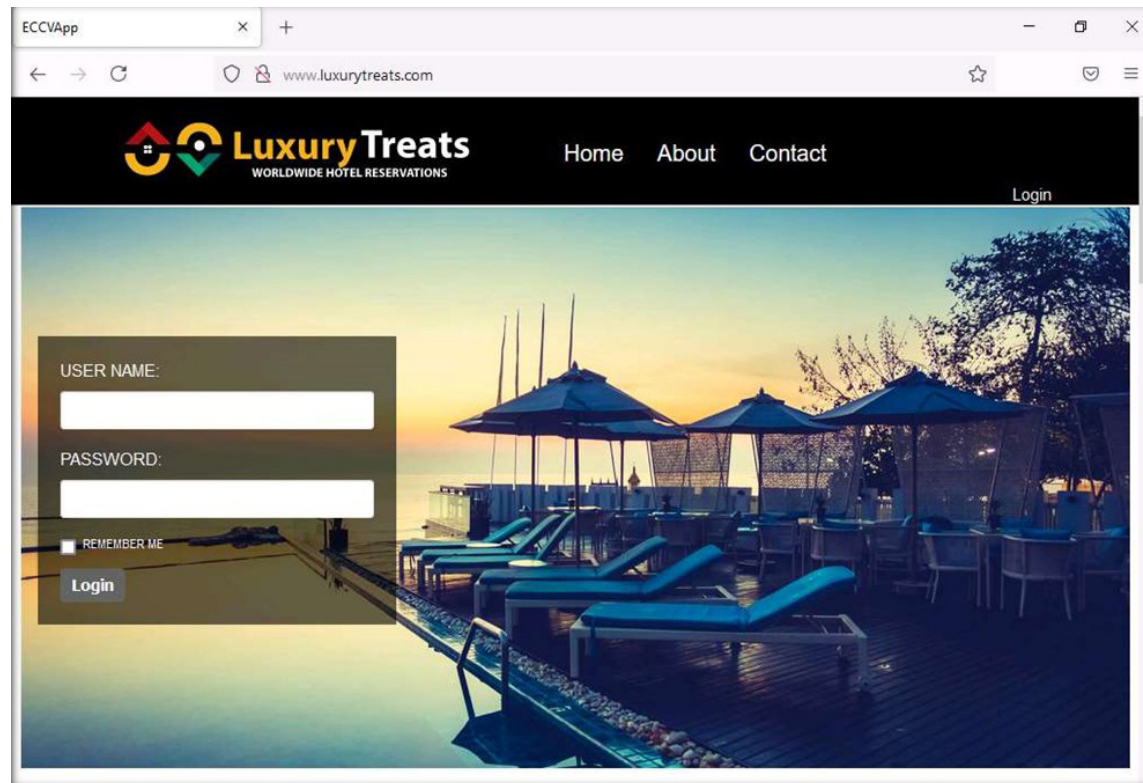
3. Select Ethernet as interface and click the Start capturing packets fin icon to start capturing the network traffic.

EXERCISE 2:
INTERCEPT
NETWORK TRAFFIC
USING WIRESHARK
AND TCPDUMP



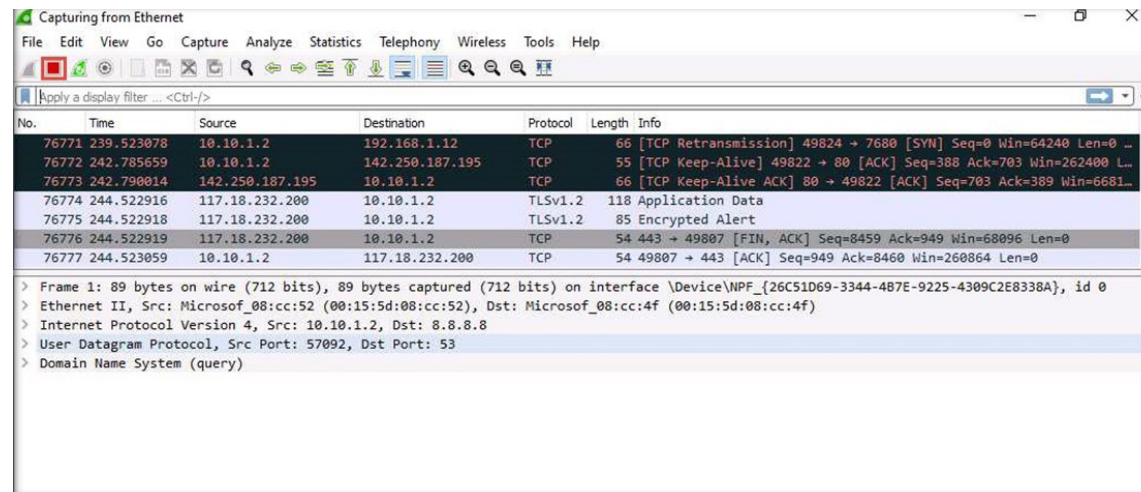
4. Next, in order to generate the network traffic between the local machine and Web Server machines, we will browse the website hosted on Web Server virtual machine.
 5. Minimize the Wireshark window.
 6. Open any web browser (here, Mozilla Firefox) and type `http://www.luxurytreats.com` in the url field and press Enter.
- Note:** If Default Browser pop-up appears, click Not now.
- Note:** Type the complete URL `www.luxurytreats.com` or `http://www.luxurytreats.com` as mentioned in the above instruction. Do not type an incomplete URL such as `luxurytreats.com`; otherwise, it will redirect you to some external website on the internet.

EXERCISE 2:
INTERCEPT
NETWORK TRAFFIC
USING WIRESHARK
AND TCPDUMP



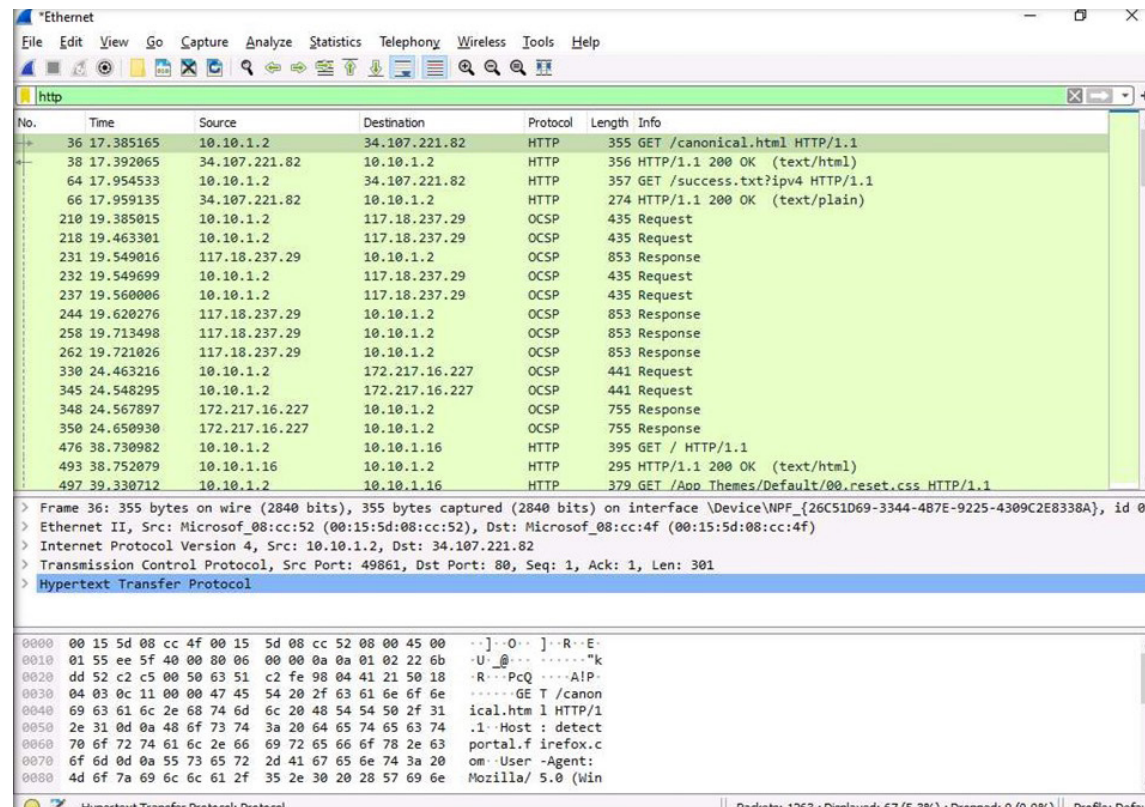
7. Now, navigate back to the Wireshark window and click Stop capturing packets icon (red color icon) in the tool bar (top-left corner) to stop the packet capturing.

EXERCISE 2:
INTERCEPT
NETWORK TRAFFIC
USING WIRESHARK
AND TCPDUMP



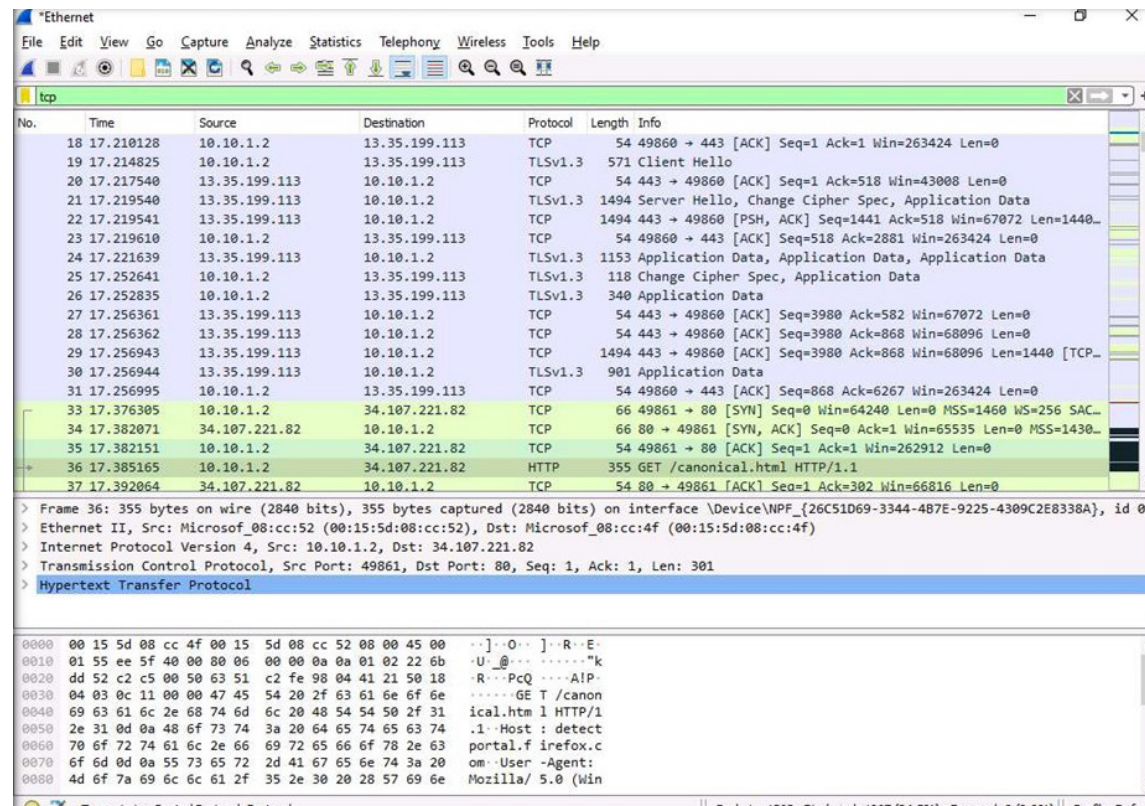
8. The Filter field at the top of the Wireshark main window allows you to apply various filters that help narrow down the traffic including filter traffic by protocol.
9. To view the HTTP-specific traffic flowing in your network, type http in the filter field and press Enter. By applying this filter, Wireshark filters and displays HTTP traffic flowing through the network.

EXERCISE 2:
 INTERCEPT
 NETWORK TRAFFIC
 USING WIRESHARK
 AND TCPDUMP



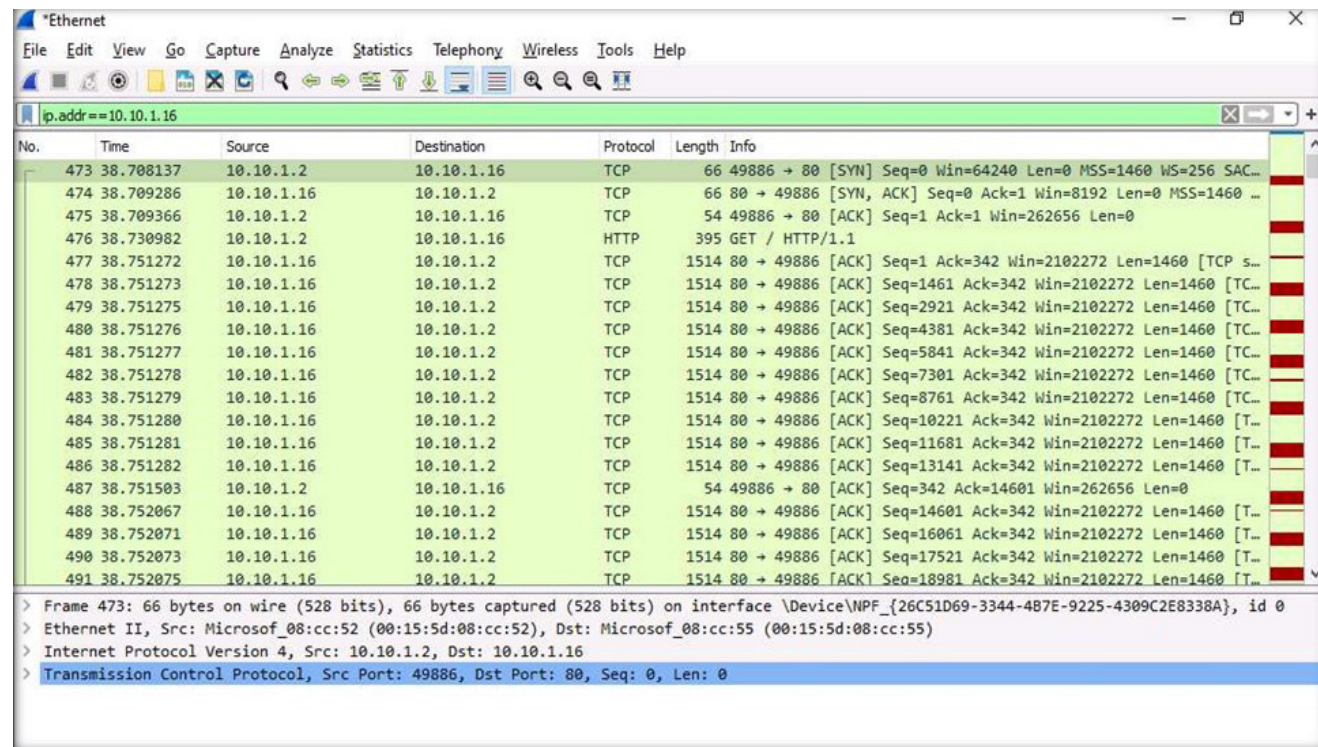
10. To view the TCP-specific traffic flowing in your network, type tcp in the filter field and press Enter. By applying this filter, Wireshark filters TCP traffic flowing through the network and displays.

EXERCISE 2:
INTERCEPT
NETWORK TRAFFIC
USING WIRESHARK
AND TCPDUMP



11. You can also filter traffic based on the source and destination IP addresses. To view traffic originating or destined to a specific IP address, apply the filter ip.addr==10.10.1.16 (the Web Server machine in which luxurytreat.com website is hosted (10.10.1.16)).

EXERCISE 2:
INTERCEPT
NETWORK TRAFFIC
USING WIRESHARK
AND TCPDUMP



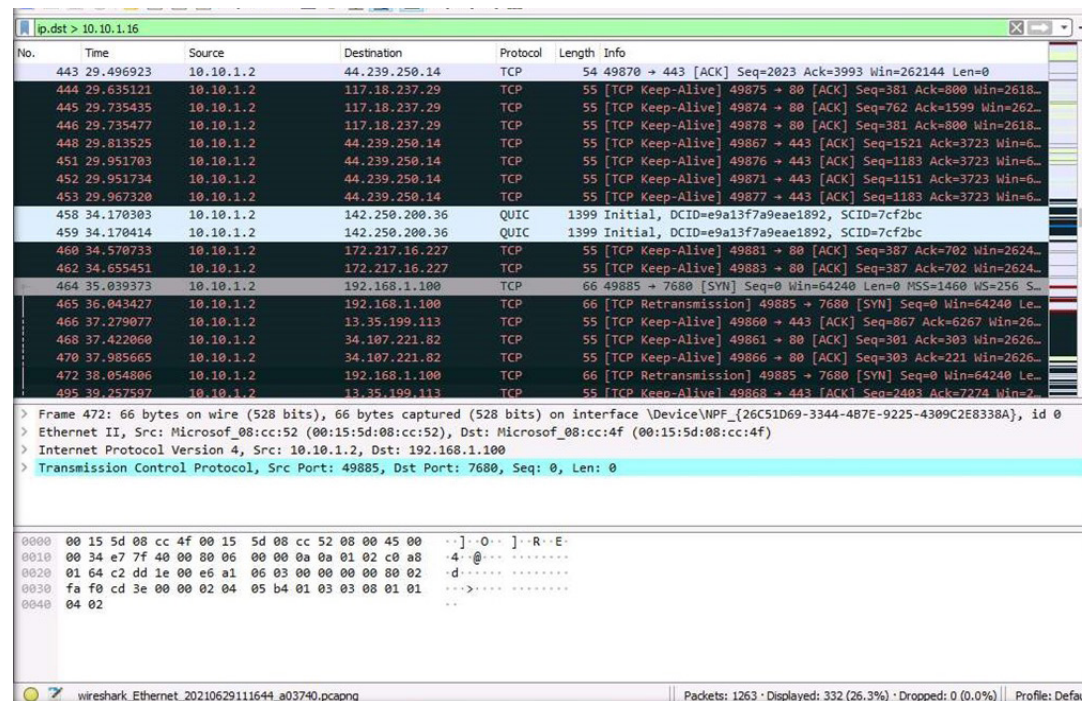
12. You can also use various conditional operators on IP address filtering to filter traffic based on your preference/requirement.

Symbol meaning:

- == Is equal to
- != Not equal to
- > Is greater than
- < Is lesser than
- >= Greater than or equal to
- <= Less than or equal to.

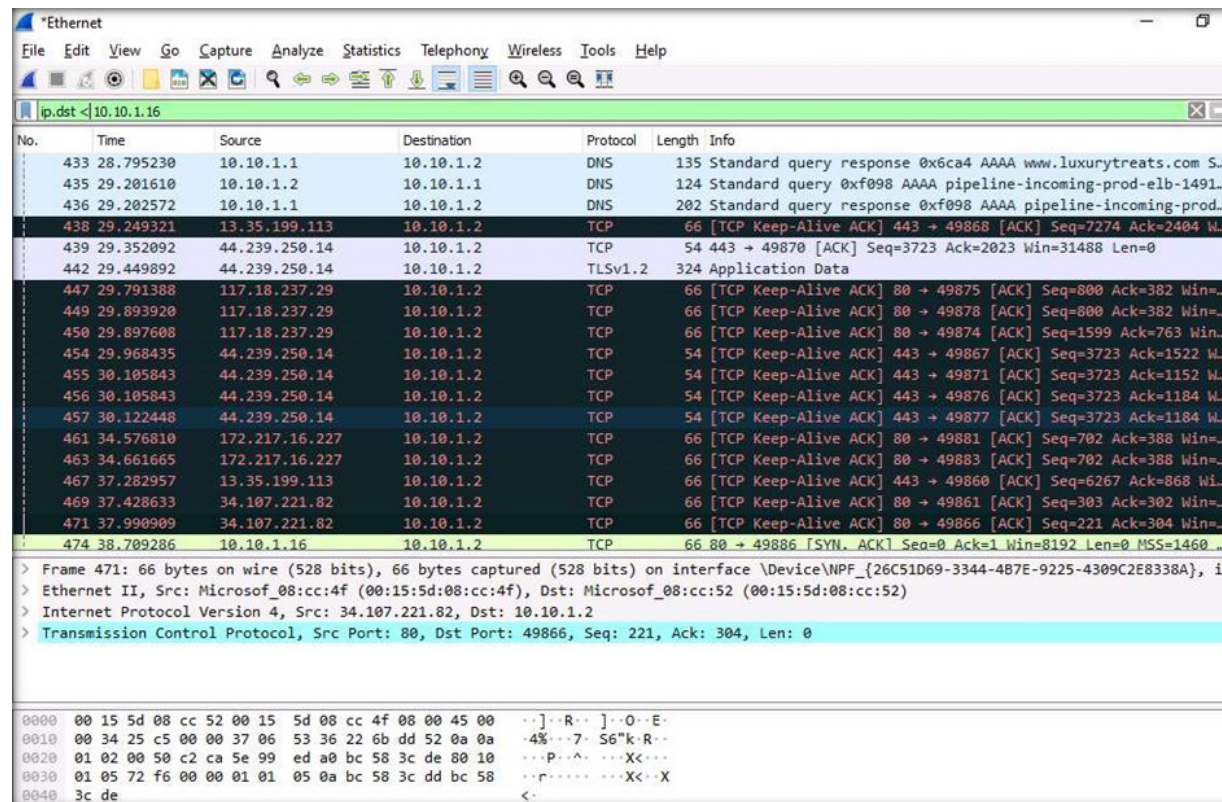
13. To view traffic higher than a specific IP address, use the > conditional operator in conjunction with IP address filtering. Apply the filter ip.dst > 10.10.1.16 to find the destination IP addresses greater than the specified IP address.

EXERCISE 2
INTERCEPT NETWORK TRAFFIC
USING WIRESHARK
AND TCPDUMP



14. To view traffic less than a specific IP address, use the < conditional operator in conjunction with IP address filtering. Apply the filter ip.dst < 10.10.1.16 to find the destination IP addresses less than the specified IP address.

EXERCISE 2:
INTERCEPT
NETWORK TRAFFIC
USING WIRESHARK
AND TCPDUMP



No.	Time	Source	Destination	Protocol	Length	Info
433	28.795230	10.10.1.1	10.10.1.2	DNS	135	Standard query response 0x6ca4 AAAA www.luxurytreats.com S...
435	29.201610	10.10.1.2	10.10.1.1	DNS	124	Standard query 0xf098 AAAA pipeline-incoming-prod-elb-1491...
436	29.202572	10.10.1.1	10.10.1.2	DNS	202	Standard query response 0xf098 AAAA pipeline-incoming-prod...
474	38.709286	10.10.1.16	10.10.1.2	TCP	66	80 → 49886 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 ...

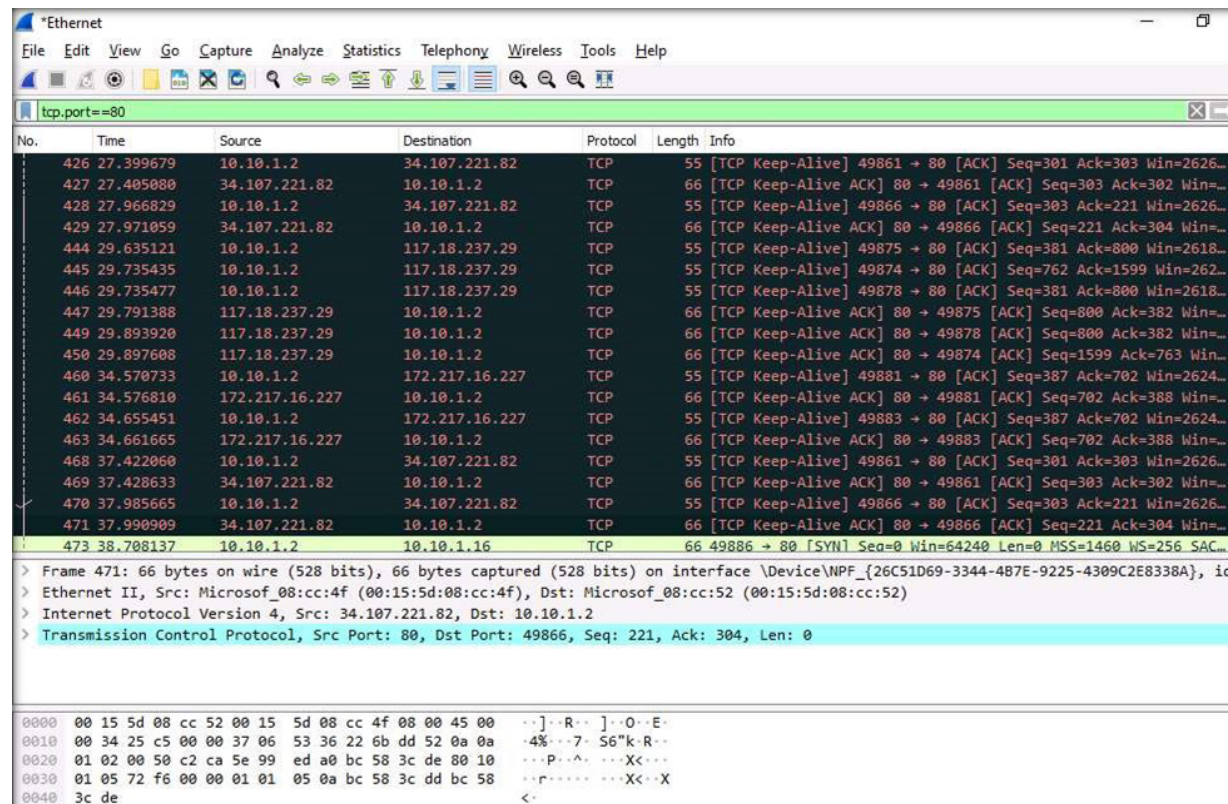
> Frame 471: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface \Device\NPF_{26C51D69-3344-4B7E-9225-4309C2E8338A}, id
 > Ethernet II, Src: Microsof_08:cc:4f (00:15:5d:08:cc:4f), Dst: Microsof_08:cc:52 (00:15:5d:08:cc:52)
 > Internet Protocol Version 4, Src: 34.107.221.82, Dst: 10.10.1.2
 > Transmission Control Protocol, Src Port: 80, Dst Port: 49866, Seq: 221, Ack: 304, Len: 0

```

0000  00 15 5d 08 cc 52 00 15 5d 08 cc 4f 08 00 45 00  ..]..R..]..O..E.
0010  00 34 25 c5 00 00 37 06 53 36 22 6b dd 52 0a 0a  ..4%...7. S6"K.R..
0020  01 02 00 50 c2 ca 5e 99 ed a0 bc 58 3c de 80 10  ...P...^...X<...
0030  01 05 72 f6 00 00 01 01 05 0a bc 58 3c dd bc 58  ...P...^...X<..X
0040  3c de                                     <
    
```

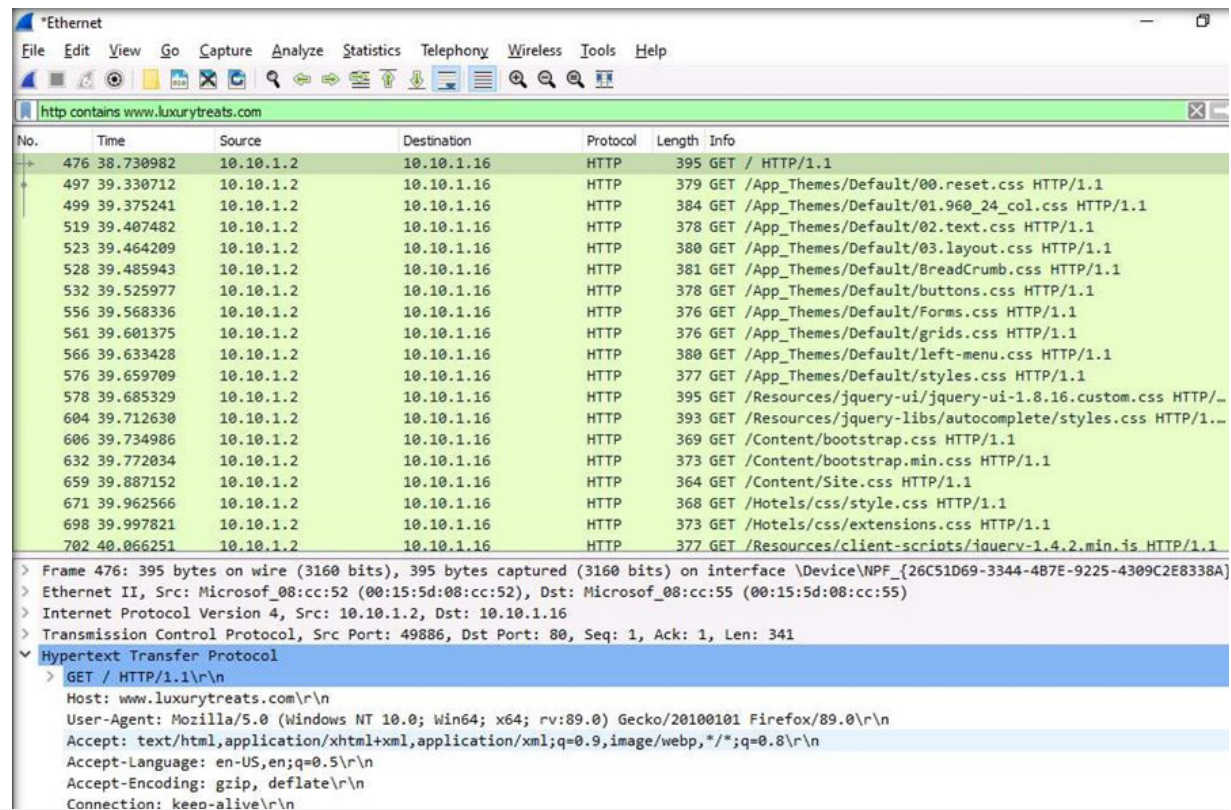
15. You can also filter traffic based on the source and destination ports. To view traffic originating or destined to the TCP port, apply the filter tcp.port==80.

EXERCISE 2:
INTERCEPT
NETWORK TRAFFIC
USING WIRESHARK
AND TCPDUMP



16. You can also filter traffic based on a specific string contained in the traffic. Apply the filter `http contains www.luxurytreats.com` to filter out the traffic that contains the mentioned string.

EXERCISE 2:
INTERCEPT
NETWORK TRAFFIC
USING WIRESHARK
AND TCPDUMP



The screenshot shows the Wireshark interface with a filter applied: `http contains www.luxurytreats.com`. The packet list pane displays a series of HTTP GET requests from 10.10.1.2 to 10.10.1.16. The selected packet (No. 476) is expanded to show the Hypertext Transfer Protocol details, including the Host, User-Agent, and Accept headers.

No.	Time	Source	Destination	Protocol	Length	Info
476	38.730982	10.10.1.2	10.10.1.16	HTTP	395	GET / HTTP/1.1
497	39.330712	10.10.1.2	10.10.1.16	HTTP	379	GET /App_Themes/Default/00.reset.css HTTP/1.1
499	39.375241	10.10.1.2	10.10.1.16	HTTP	384	GET /App_Themes/Default/01.960_24_col.css HTTP/1.1
519	39.407482	10.10.1.2	10.10.1.16	HTTP	378	GET /App_Themes/Default/02.text.css HTTP/1.1
523	39.464209	10.10.1.2	10.10.1.16	HTTP	380	GET /App_Themes/Default/03.layout.css HTTP/1.1
528	39.485943	10.10.1.2	10.10.1.16	HTTP	381	GET /App_Themes/Default/BreadCrumb.css HTTP/1.1
532	39.525977	10.10.1.2	10.10.1.16	HTTP	378	GET /App_Themes/Default/buttons.css HTTP/1.1
556	39.568336	10.10.1.2	10.10.1.16	HTTP	376	GET /App_Themes/Default/Forms.css HTTP/1.1
561	39.601375	10.10.1.2	10.10.1.16	HTTP	376	GET /App_Themes/Default/grids.css HTTP/1.1
566	39.633428	10.10.1.2	10.10.1.16	HTTP	380	GET /App_Themes/Default/left-menu.css HTTP/1.1
576	39.659709	10.10.1.2	10.10.1.16	HTTP	377	GET /App_Themes/Default/styles.css HTTP/1.1
578	39.685329	10.10.1.2	10.10.1.16	HTTP	395	GET /Resources/jquery-ui/jquery-ui-1.8.16.custom.css HTTP/1.1
604	39.712630	10.10.1.2	10.10.1.16	HTTP	393	GET /Resources/jquery-lib/autocomplete/styles.css HTTP/1.1
606	39.734986	10.10.1.2	10.10.1.16	HTTP	369	GET /Content/bootstrap.css HTTP/1.1
632	39.772034	10.10.1.2	10.10.1.16	HTTP	373	GET /Content/bootstrap.min.css HTTP/1.1
659	39.887152	10.10.1.2	10.10.1.16	HTTP	364	GET /Content/Site.css HTTP/1.1
671	39.962566	10.10.1.2	10.10.1.16	HTTP	368	GET /Hotels/css/style.css HTTP/1.1
698	39.997821	10.10.1.2	10.10.1.16	HTTP	373	GET /Hotels/css/extensions.css HTTP/1.1
702	40.066251	10.10.1.2	10.10.1.16	HTTP	377	GET /Resources/client-scripts/jquery-1.4.2.min.js HTTP/1.1

```

> Frame 476: 395 bytes on wire (3160 bits), 395 bytes captured (3160 bits) on interface \Device\NPF_{26C51D69-3344-487E-9225-4309C2E8338A}
> Ethernet II, Src: Microsof_08:cc:52 (00:15:5d:08:cc:52), Dst: Microsof_08:cc:55 (00:15:5d:08:cc:55)
> Internet Protocol Version 4, Src: 10.10.1.2, Dst: 10.10.1.16
> Transmission Control Protocol, Src Port: 49886, Dst Port: 80, Seq: 1, Ack: 1, Len: 341
* Hypertext Transfer Protocol
  > GET / HTTP/1.1\r\n
    Host: www.luxurytreats.com\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:89.0) Gecko/20100101 Firefox/89.0\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8\r\n
    Accept-Language: en-US,en;q=0.5\r\n
    Accept-Encoding: gzip, deflate\r\n
    Connection: keep-alive\r\n
  
```


Note: This can only be applied to characters and not numerals. It searches for a sequence of characters provided in the filter.

17. To view the HTTP traffic whose request header fields (host) contain a specific string, apply the http.host contains www.luxurytreats.com filter.

EXERCISE 2:
INTERCEPT
NETWORK TRAFFIC
USING WIRESHARK
AND TCPDUMP

The screenshot shows the Wireshark interface with a filter applied: `http.host contains www.luxurytreats.com`. The packet list pane displays several HTTP GET requests. The selected packet (No. 476) is expanded to show the following details:

```

    > GET / HTTP/1.1\r\n
    Host: www.luxurytreats.com\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:89.0) Gecko/20100101 Firefox/89.0\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8\r\n
    Accept-Language: en-US,en;q=0.5\r\n
    Accept-Encoding: gzip, deflate\r\n
    Connection: keep-alive\r\n
    Upgrade-Insecure-Requests: 1\r\n
    \r\n
    [Full request URI: http://www.luxurytreats.com/]
    [HTTP request 1/3]
    [Response in frame: 493]
    [Next request in frame: 497]
  
```

18. Similarly, you may use various other filters to filter the required traffic.
19. As described above, a security professional can specify one or more conditional and logical operators to find traffic based on their preference/requirement. Thus, Wireshark allows you to use a wide range of filters to filter traffic based on your preference/requirement.
20. This concludes the demonstration showing how to apply various filters using Wireshark.
21. Close all open windows.
22. Turn off the Admin Machine-1 virtual machine.

EXERCISE 2: INTERCEPT NETWORK TRAFFIC USING WIRESHARK AND TCPDUMP

EXERCISE 3: ANALYZE AND EXAMINE VARIOUS NETWORK PACKET HEADERS IN LINUX USING tcpdump

tcpdump is used to analyze TCP/IP and other packets on Linux host machine.

LAB SCENARIO

Each packet in a network contains control information and user data, known as the payload. The control information contains data for delivering the payload, which includes, for example, source and destination IP and MAC addresses and sequencing information. The header part of the packet stores this control information. Hence, the security professional needs to know how to examine the packet headers while examining the data packets.

OBJECTIVE

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

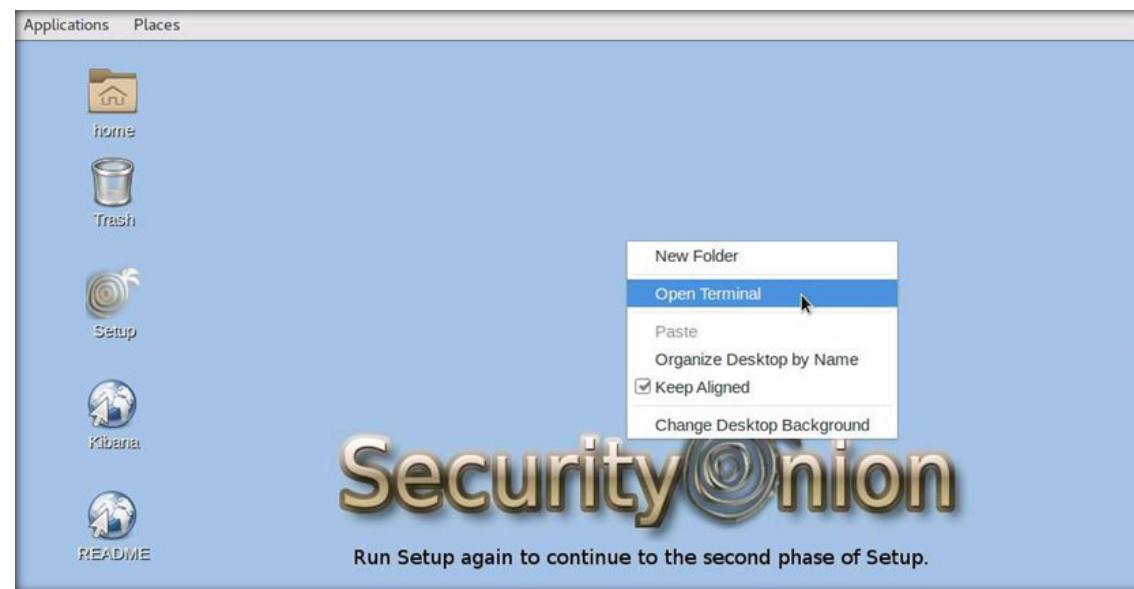
OVERVIEW OF TROJAN

Data packets traversing over a network can be intercepted using packet capture tools such as tcpdump. These captured packets are analyzed to determine whether proper network security policies are being followed.

Note: Ensure that Web Server and PfSense Firewall virtual machines are running.

1. Turn on the Admin Machine-2 virtual machine.
2. Type username as sam and password as admin@123 and click Log In.
3. Open a terminal by right-clicking on Desktop, and then click Open Terminal from the pop-up menu.

EXERCISE 3: ANALYZE AND EXAMINE VARIOUS NETWORK PACKET HEADERS IN LINUX USING TCPDUMP



4. The Terminal window will appear. Type `sudo tcpdump` in Command Prompt and press Enter to capture the network packets of the machine. If you encounter a password prompt, type `admin@123`.

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

EXERCISE 3: ANALYZE AND EXAMINE VARIOUS NETWORK PACKET HEADERS IN LINUX USING TCPDUMP



```
Applications  Places  Terminal  
  
sam@sam-Virtual-Machine: ~  
File Edit View Search Terminal Help  
sam@sam-Virtual-Machine:~$ sudo tcpdump
```

5. The tcpdump command shows the entire payloads captured packet.

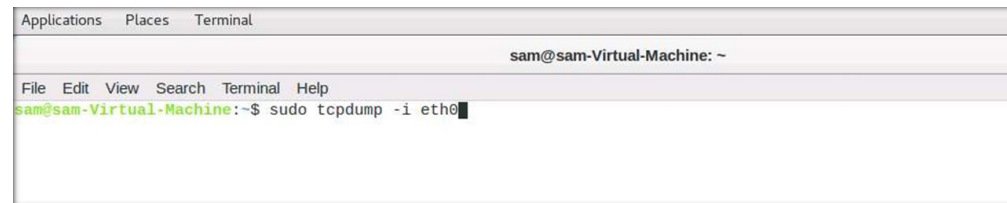
EXERCISE 3:
ANALYZE AND
EXAMINE VARIOUS
NETWORK PACKET
HEADERS IN LINUX
USING TCPDUMP

```

Applications  Places  Terminal  Thu 03:59
-----
sam@sam-Virtual-Machine: ~
File Edit View Search Terminal Help
03:59:02.800804 IP dns.google.domain > sam-Virtual-Machine.33089: 53577 1/0/0 PTR pugot.canonical.com. (75)
03:59:03.789457 IP sam-Virtual-Machine.ntp > ntp2.unix-solutions.be.ntp: NTPv4, Client, length 48
03:59:03.789475 IP sam-Virtual-Machine.ntp > ns0.luns.net.uk.ntp: NTPv4, Client, length 48
03:59:03.789481 IP sam-Virtual-Machine.ntp > ns2.vedur.is.ntp: NTPv4, Client, length 48
03:59:03.789622 IP sam-Virtual-Machine.59065 > dns.google.domain: 44142+ PTR? 220.204.111.185.in-addr.arpa. (46)
03:59:03.792466 IP dns.google.domain > sam-Virtual-Machine.59065: 44142 1/0/0 PTR ntp2.unix-solutions.be. (82)
03:59:03.792604 IP sam-Virtual-Machine.51694 > dns.google.domain: 46341+ PTR? 66.59.114.217.in-addr.arpa. (44)
03:59:03.795890 IP dns.google.domain > sam-Virtual-Machine.51694: 46341 1/0/0 PTR ns0.luns.net.uk. (73)
03:59:03.795985 IP sam-Virtual-Machine.56471 > dns.google.domain: 13966+ PTR? 152.87.208.130.in-addr.arpa. (45)
03:59:03.799427 IP ns0.luns.net.uk.ntp > sam-Virtual-Machine.ntp: NTPv4, Server, length 48
03:59:03.834145 IP ns2.vedur.is.ntp > sam-Virtual-Machine.ntp: NTPv4, Server, length 48
03:59:04.789458 IP sam-Virtual-Machine.ntp > ntp.grumpyminx.com.ntp: NTPv4, Client, length 48
03:59:04.789478 IP sam-Virtual-Machine.ntp > ntp3.ds.network.ntp: NTPv4, Client, length 48
03:59:04.789483 IP sam-Virtual-Machine.ntp > rilian.whisker.org.uk.ntp: NTPv4, Client, length 48
03:59:04.789622 IP sam-Virtual-Machine.41770 > dns.google.domain: 49042+ PTR? 66.138.70.82.in-addr.arpa. (43)
03:59:04.792363 IP dns.google.domain > sam-Virtual-Machine.41770: 49042 1/0/0 PTR ntp.grumpyminx.com. (75)
03:59:04.792526 IP sam-Virtual-Machine.34902 > dns.google.domain: 23725+ PTR? 252.125.124.27.in-addr.arpa. (45)
03:59:04.793583 IP rilian.whisker.org.uk.ntp > sam-Virtual-Machine.ntp: NTPv4, Server, length 48
03:59:04.793594 IP ntp3.ds.network.ntp > sam-Virtual-Machine.ntp: NTPv4, Server, length 48
03:59:04.806926 IP sam-Virtual-Machine.51386 > dns.google.domain: 58651+ PTR? 57.52.237.94.in-addr.arpa. (43)
03:59:05.789466 IP sam-Virtual-Machine.ntp > alphyn.canonical.com.ntp: NTPv4, Client, length 48
03:59:05.789619 IP sam-Virtual-Machine.33903 > dns.google.domain: 53912+ PTR? 157.91.189.91.in-addr.arpa. (44)
03:59:05.801907 IP dns.google.domain > sam-Virtual-Machine.33903: 53912 1/0/0 PTR alphyn.canonical.com. (78)
03:59:05.805139 IP alphyn.canonical.com.ntp > sam-Virtual-Machine.ntp: NTPv4, Server, length 48
03:59:06.789458 IP sam-Virtual-Machine.ntp > ntp1.doorhan.ru.ntp: NTPv4, Client, length 48
03:59:06.789477 IP sam-Virtual-Machine.ntp > host-212-159-133-83.static.as13285.net.ntp: NTPv4, Client, length 48
03:59:06.789483 IP sam-Virtual-Machine.ntp > wolke.ellsaesser.net.ntp: NTPv4, Client, length 48
03:59:06.789600 IP sam-Virtual-Machine.33464 > dns.google.domain: 32094+ PTR? 10.94.209.91.in-addr.arpa. (43)
03:59:06.798953 IP dns.google.domain > sam-Virtual-Machine.33464: 32094 1/0/0 PTR ntp1.doorhan.ru. (72)
03:59:06.799131 IP sam-Virtual-Machine.53001 > dns.google.domain: 11405+ PTR? 83.133.159.212.in-addr.arpa. (45)
03:59:06.801924 IP host-212-159-133-83.static.as13285.net.ntp > sam-Virtual-Machine.ntp: NTPv4, Server, length 48
03:59:06.808171 IP wolke.ellsaesser.net.ntp > sam-Virtual-Machine.ntp: NTPv4, Server, length 48
03:59:06.822471 IP sam-Virtual-Machine.42481 > dns.google.domain: 477+ PTR? 121.244.132.45.in-addr.arpa. (45)
03:59:07.789460 IP sam-Virtual-Machine.ntp > time.cloudflare.com.ntp: NTPv4, Client, length 48
03:59:07.790176 IP sam-Virtual-Machine.43225 > dns.google.domain: 37619+ PTR? 123.200.159.162.in-addr.arpa. (46)
03:59:07.795022 IP dns.google.domain > sam-Virtual-Machine.43225: 37619 1/0/0 PTR time.cloudflare.com. (79)
03:59:07.799123 IP time.cloudflare.com.ntp > sam-Virtual-Machine.ntp: NTPv4, Server, length 48
03:59:08.789456 IP sam-Virtual-Machine.ntp > cosima.470n.act.tsngl.co.ntp: NTPv4, Client, length 48
03:59:08.789500 IP sam-Virtual-Machine.56900 > dns.google.domain: 62363+ PTR? 166.166.48.144.in-addr.arpa. (45)
03:59:08.793567 IP dns.google.domain > sam-Virtual-Machine.56900: 62363 1/0/0 PTR cosima.470n.act.tsngl.co. (83)
03:59:09.643309 IP cosima.470n.act.tsngl.co.ntp > sam-Virtual-Machine.ntp: NTPv4, Server, length 48
03:59:09.789443 IP sam-Virtual-Machine.ntp > y.ns.gin.ntt.net.ntp: NTPv4, Client, length 48
03:59:09.790409 IP sam-Virtual-Machine.54094 > dns.google.domain: 32775+ PTR? 251.35.250.129.in-addr.arpa. (45)
03:59:09.793747 IP dns.google.domain > sam-Virtual-Machine.54094: 32775 1/0/0 PTR y.ns.gin.ntt.net. (75)
03:59:09.798684 IP y.ns.gin.ntt.net.ntp > sam-Virtual-Machine.ntp: NTPv4, Server, length 48
    
```

6. Press Ctrl + C to end the packet capture.
7. Type `sudo tcpdump -i eth0` in the terminal and press Enter to capture the network packets from the machine's specific interface.

EXERCISE 3: ANALYZE AND EXAMINE VARIOUS NETWORK PACKET HEADERS IN LINUX USING TCPDUMP



```
Applications  Places  Terminal
sam@sam-Virtual-Machine: ~
File Edit View Search Terminal Help
sam@sam-Virtual-Machine:~$ sudo tcpdump -i eth0
```

8. Press Ctrl + C to end the packet capture.

EXERCISE 3:
ANALYZE AND
EXAMINE VARIOUS
NETWORK PACKET
HEADERS IN LINUX
USING TCPDUMP

```

Applications  Places  Terminal  Thu 04:00
-----
sam@sam-Virtual-Machine: ~
File Edit View Search Terminal Help
04:00:05.916587 IP sam-Virtual-Machine.53471 > dns.google.domain: 22738+ PTR? 1.1.10.10.in-addr.arpa. (40)
04:00:05.923409 IP dns.google.domain > sam-Virtual-Machine.53471: 22738 NXDomain 0/0/0 (40)
04:00:05.924212 IP sam-Virtual-Machine.58221 > dns.google.domain: 36615+ PTR? 79.1.10.10.in-addr.arpa. (41)
04:00:05.928197 IP dns.google.domain > sam-Virtual-Machine.58221: 36615 NXDomain 0/0/0 (41)
04:00:05.931495 IP sam-Virtual-Machine.38306 > dns.google.domain: 10965+ PTR? 8.8.8.8.in-addr.arpa. (38)
04:00:05.936992 IP dns.google.domain > sam-Virtual-Machine.38306: 10965 1/0/0 PTR dns.google. (62)
04:00:07.789457 IP sam-Virtual-Machine.ntp > time.cloudflare.com.ntp: NTPv4, Client, length 48
04:00:07.789861 IP sam-Virtual-Machine.39489 > dns.google.domain: 26002+ PTR? 1.200.159.162.in-addr.arpa. (44)
04:00:07.793377 IP dns.google.domain > sam-Virtual-Machine.39489: 26002 1/0/0 PTR time.cloudflare.com. (77)
04:00:07.798516 IP time.cloudflare.com.ntp > sam-Virtual-Machine.ntp: NTPv4, Server, length 48
04:00:08.789455 IP sam-Virtual-Machine.ntp > pugot.canonical.com.ntp: NTPv4, Client, length 48
04:00:08.789588 IP sam-Virtual-Machine.41317 > dns.google.domain: 37896+ PTR? 4.94.189.91.in-addr.arpa. (42)
04:00:08.793516 IP pugot.canonical.com.ntp > sam-Virtual-Machine.ntp: NTPv4, Server, length 48
04:00:08.793537 IP dns.google.domain > sam-Virtual-Machine.41317: 37896 1/0/0 PTR pugot.canonical.com. (75)
04:00:09.789468 IP sam-Virtual-Machine.ntp > time.rdg.uk.as44574.net.ntp: NTPv4, Client, length 48
04:00:09.789612 IP sam-Virtual-Machine.34010 > dns.google.domain: 28146+ PTR? 2.34.159.193.in-addr.arpa. (43)
04:00:09.793968 IP dns.google.domain > sam-Virtual-Machine.34010: 28146 1/0/0 PTR time.rdg.uk.as44574.net. (80)
04:00:09.793937 IP time.rdg.uk.as44574.net.ntp > sam-Virtual-Machine.ntp: NTPv4, Server, length 48
04:00:10.789456 IP sam-Virtual-Machine.ntp > alphyn.canonical.com.ntp: NTPv4, Client, length 48
04:00:10.789476 IP sam-Virtual-Machine.ntp > ns3.turbodns.co.uk.ntp: NTPv4, Client, length 48
04:00:10.789483 IP sam-Virtual-Machine.ntp > thomas-avatar.bnr.la.ntp: NTPv4, Client, length 48
04:00:10.789489 IP sam-Virtual-Machine.ntp > ns2.vedur.is.ntp: NTPv4, Client, length 48
04:00:10.789494 IP sam-Virtual-Machine.ntp > 229.191.57.185.no-ptr.as201971.net.ntp: NTPv4, Client, length 48
04:00:10.789602 IP sam-Virtual-Machine.59787 > dns.google.domain: 13923+ PTR? 157.91.189.91.in-addr.arpa. (44)
04:00:10.798454 IP 229.191.57.185.no-ptr.as201971.net.ntp > sam-Virtual-Machine.ntp: NTPv4, Server, length 48
04:00:10.798713 IP sam-Virtual-Machine.42273 > dns.google.domain: 18328+ PTR? 168.65.21.81.in-addr.arpa. (43)
04:00:10.803995 IP sam-Virtual-Machine.50706 > dns.google.domain: 11959+ PTR? 20.34.213.112.in-addr.arpa. (44)
04:00:10.815974 IP sam-Virtual-Machine.33189 > dns.google.domain: 58364+ PTR? 152.87.208.130.in-addr.arpa. (45)
04:00:10.821490 IP sam-Virtual-Machine.43585 > dns.google.domain: 11808+ PTR? 229.191.57.185.in-addr.arpa. (45)
04:00:10.838351 IP ns2.vedur.is.ntp > sam-Virtual-Machine.ntp: NTPv4, Server, length 48
04:00:10.864211 IP alphyn.canonical.com.ntp > sam-Virtual-Machine.ntp: NTPv4, Server, length 48
04:00:11.639148 IP thomas-avatar.bnr.la.ntp > sam-Virtual-Machine.ntp: NTPv4, Server, length 48
04:00:11.789456 IP sam-Virtual-Machine.ntp > ntp2.unix-solutions.be.ntp: NTPv4, Client, length 48
04:00:11.789476 IP sam-Virtual-Machine.ntp > ntp3.ds.network.ntp: NTPv4, Client, length 48
04:00:11.789482 IP sam-Virtual-Machine.ntp > rillian.whisker.org.uk.ntp: NTPv4, Client, length 48
04:00:11.789606 IP sam-Virtual-Machine.54773 > dns.google.domain: 2936+ PTR? 220.204.111.185.in-addr.arpa. (46)
04:00:11.794015 IP dns.google.domain > sam-Virtual-Machine.54773: 2936 1/0/0 PTR ntp2.unix-solutions.be. (82)
04:00:11.794187 IP sam-Virtual-Machine.51234 > dns.google.domain: 14936+ PTR? 252.125.124.27.in-addr.arpa. (45)
04:00:11.794519 IP rillian.whisker.org.uk.ntp > sam-Virtual-Machine.ntp: NTPv4, Server, length 48
04:00:11.794532 IP ntp3.ds.network.ntp > sam-Virtual-Machine.ntp: NTPv4, Server, length 48
04:00:11.799167 IP sam-Virtual-Machine.37926 > dns.google.domain: 47427+ PTR? 57.52.237.94.in-addr.arpa. (43)
^C
43 packets captured
52 packets received by filter
9 packets dropped by kernel
sam@sam-Virtual-Machine:~$
    
```


9. Type `sudo tcpdump -i eth0 tcp` in the terminal and press Enter to capture only the TCP packets from the machine interface.




```
Applications  Places  Terminal  
sam@sam-Virtual-Machine: ~  
File Edit View Search Terminal Help  
sam@sam-Virtual-Machine:~$ sudo tcpdump -i eth0 tcp
```

EXERCISE 3: ANALYZE AND EXAMINE VARIOUS NETWORK PACKET HEADERS IN LINUX USING TCPDUMP

10. Open another terminal and type `dd if=/dev/urandom bs=1M count=1 | nc 10.10.1.50 9000` and press Enter; it generates the TCP packets of 1MB and sends them to destination 10.10.1.50.

EXERCISE 3: ANALYZE AND EXAMINE VARIOUS NETWORK PACKET HEADERS IN LINUX USING TCPDUMP



```
Applications  Places  Terminal  
sam@sam-Virtual-Machine: ~  
File Edit View Search Terminal Help  
sam@sam-Virtual-Machine:~$ dd if=/dev/urandom bs=1M count=1 | nc 10.10.1.50 9000  
sam@sam-Virtual-Machine:~$
```

11. Switch back to the first Terminal; here, you can see the traffic captured by the Tcpcdump.

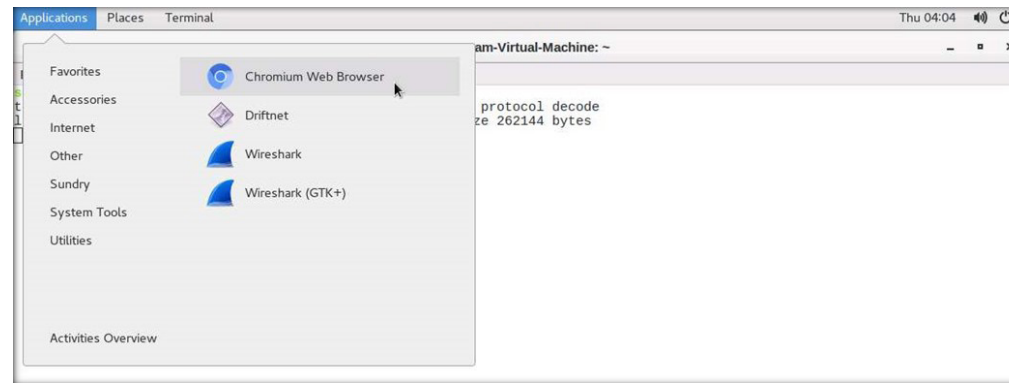
EXERCISE 3:
ANALYZE AND
EXAMINE VARIOUS
NETWORK PACKET
HEADERS IN LINUX
USING TCPDUMP

```

Applications Places Terminal Thu 04:02
sam@sam-Virtual-Machine: ~
File Edit View Search Terminal Help
sam@sam-Virtual-Machine:~$ sudo tcpdump -i eth0 tcp
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
04:02:30.961098 IP sam-Virtual-Machine.39258 > 10.10.1.50.9000: Flags [S], seq 2650857465, win 65535, options [mss 1460,sackOK
,TS val 3334729349 ecr 0,nop,wscale 11], length 0
04:02:30.961402 IP 10.10.1.50.9000 > sam-Virtual-Machine.39258: Flags [R.], seq 0, ack 2650857466, win 0, length 0
    
```

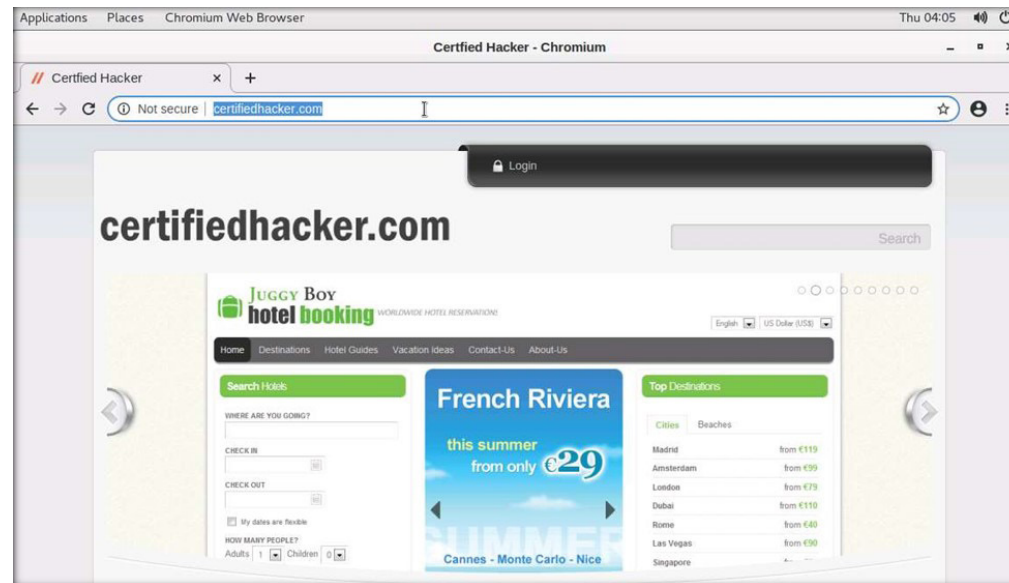
12. Press Ctrl + C to end the packet capture.
13. Type `sudo tcpdump -i eth0 port 80` in the terminal and press Enter to capture packets from the specific port on the machine interface.
14. Navigate to Applications → Internet and select Chromium Web Browser.

EXERCISE 3:
ANALYZE AND
EXAMINE VARIOUS
NETWORK PACKET
HEADERS IN LINUX
USING TCPDUMP



15. The Chromium Web Browser opens. Type `http://www.certifiedhacker.com` as url and press Enter.

EXERCISE 3:
ANALYZE AND
EXAMINE VARIOUS
NETWORK PACKET
HEADERS IN LINUX
USING TCPDUMP



16. Switch back to the opened Terminal; you can observe that the tcpdump is capturing port 80 http traffic.

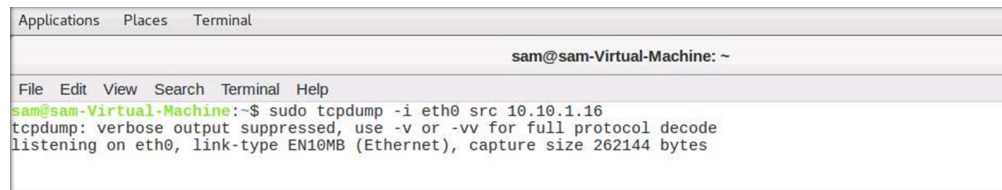
EXERCISE 3:
ANALYZE AND
EXAMINE VARIOUS
NETWORK PACKET
HEADERS IN LINUX
USING TCPDUMP

```

Applications  Places  Terminal  Thu 04:05  [Terminal Icon]
-----
sam@sam-Virtual-Machine: ~
File Edit View Search Terminal Help
options [nop,nop,TS val 4153001220 ecr 3599564789,nop,nop,sack 1 {5093541:6723278}], length 0
04:05:26.242350 IP 123.35.104.34.bc.googleusercontent.com.http > sam-Virtual-Machine.42660: Flags [.], seq 4795341:4798181, ac
k 387, win 261, options [nop,nop,TS val 3599564789 ecr 4153001211], length 2840: HTTP
04:05:26.248708 IP 123.35.104.34.bc.googleusercontent.com.http > sam-Virtual-Machine.42660: Flags [.], seq 4880541:4900421, ac
k 387, win 261, options [nop,nop,TS val 3599564795 ecr 4153001219], length 19880: HTTP
04:05:26.248742 IP sam-Virtual-Machine.42660 > 123.35.104.34.bc.googleusercontent.com.http: Flags [.], ack 4900421, win 1427,
options [nop,nop,TS val 4153001227 ecr 3599564795,nop,nop,sack 1 {5093541:6723278}], length 0
04:05:26.248749 IP 123.35.104.34.bc.googleusercontent.com.http > sam-Virtual-Machine.42660: Flags [.], seq 4900421:4907521, ac
k 387, win 261, options [nop,nop,TS val 3599564796 ecr 4153001219], length 7100: HTTP
04:05:26.248758 IP sam-Virtual-Machine.42660 > 123.35.104.34.bc.googleusercontent.com.http: Flags [.], ack 4907521, win 1424,
options [nop,nop,TS val 4153001227 ecr 3599564796,nop,nop,sack 1 {5093541:6723278}], length 0
04:05:26.249013 IP 123.35.104.34.bc.googleusercontent.com.http > sam-Virtual-Machine.42660: Flags [.], seq 4907521:4914621, ac
k 387, win 261, options [nop,nop,TS val 3599564796 ecr 4153001219], length 7100: HTTP
04:05:26.249027 IP sam-Virtual-Machine.42660 > 123.35.104.34.bc.googleusercontent.com.http: Flags [.], ack 4914621, win 1421,
options [nop,nop,TS val 4153001227 ecr 3599564796,nop,nop,sack 1 {5093541:6723278}], length 0
04:05:26.249082 IP 123.35.104.34.bc.googleusercontent.com.http > sam-Virtual-Machine.42660: Flags [.], seq 4914621:4921721, ac
k 387, win 261, options [nop,nop,TS val 3599564796 ecr 4153001219], length 7100: HTTP
04:05:26.250669 IP 123.35.104.34.bc.googleusercontent.com.http > sam-Virtual-Machine.42660: Flags [.], seq 4975681:4985621, ac
k 387, win 261, options [nop,nop,TS val 3599564797 ecr 4153001219], length 9940: HTTP
04:05:26.250881 IP 123.35.104.34.bc.googleusercontent.com.http > sam-Virtual-Machine.42660: Flags [.], seq 4985621:5004081, ac
k 387, win 261, options [nop,nop,TS val 3599564797 ecr 4153001219], length 18460: HTTP
04:05:26.250903 IP sam-Virtual-Machine.42660 > 123.35.104.34.bc.googleusercontent.com.http: Flags [.], ack 5004081, win 1300,
options [nop,nop,TS val 4153001229 ecr 3599564797,nop,nop,sack 1 {5093541:6723278}], length 0
04:05:26.251317 IP 123.35.104.34.bc.googleusercontent.com.http > sam-Virtual-Machine.42660: Flags [.], seq 5004081:5011181, ac
k 387, win 261, options [nop,nop,TS val 3599564798 ecr 4153001219], length 7100: HTTP
04:05:26.251336 IP sam-Virtual-Machine.42660 > 123.35.104.34.bc.googleusercontent.com.http: Flags [.], seq 5011181, win 1377,
options [nop,nop,TS val 4153001230 ecr 3599564798,nop,nop,sack 1 {5093541:6723278}], length 0
04:05:26.251340 IP 123.35.104.34.bc.googleusercontent.com.http > sam-Virtual-Machine.42660: Flags [.], seq 5011181:5012601, ac
k 387, win 261, options [nop,nop,TS val 3599564799 ecr 4153001220], length 1420: HTTP
04:05:26.251345 IP sam-Virtual-Machine.42660 > 123.35.104.34.bc.googleusercontent.com.http: Flags [.], ack 5012601, win 1377,
options [nop,nop,TS val 4153001230 ecr 3599564799,nop,nop,sack 1 {5093541:6723278}], length 0
04:05:26.251682 IP 123.35.104.34.bc.googleusercontent.com.http > sam-Virtual-Machine.42660: Flags [.], seq 5012601:5015441, ac
k 387, win 261, options [nop,nop,TS val 3599564799 ecr 4153001220], length 2840: HTTP
04:05:26.251694 IP sam-Virtual-Machine.42660 > 123.35.104.34.bc.googleusercontent.com.http: Flags [.], ack 5015441, win 1376,
options [nop,nop,TS val 4153001230 ecr 3599564799,nop,nop,sack 1 {5093541:6723278}], length 0
04:05:26.251740 IP 123.35.104.34.bc.googleusercontent.com.http > sam-Virtual-Machine.42660: Flags [.], seq 5015441:5029641, ac
k 387, win 261, options [nop,nop,TS val 3599564799 ecr 4153001220], length 14200: HTTP
04:05:26.252963 IP 123.35.104.34.bc.googleusercontent.com.http > sam-Virtual-Machine.42660: Flags [.], seq 5072241:5085021, ac
k 387, win 261, options [nop,nop,TS val 3599564800 ecr 4153001220], length 12780: HTTP
04:05:26.252977 IP sam-Virtual-Machine.42660 > 123.35.104.34.bc.googleusercontent.com.http: Flags [.], ack 5085021, win 1345,
options [nop,nop,TS val 4153001231 ecr 3599564800,nop,nop,sack 1 {5093541:6723278}], length 0
04:05:26.253265 IP 123.35.104.34.bc.googleusercontent.com.http > sam-Virtual-Machine.42660: Flags [.], seq 5085021:5093541, ac
k 387, win 261, options [nop,nop,TS val 3599564800 ecr 4153001221], length 8520: HTTP
04:05:26.253529 IP sam-Virtual-Machine.42660 > 123.35.104.34.bc.googleusercontent.com.http: Flags [.], ack 6723278, win 1213,
options [nop,nop,TS val 4153001232 ecr 3599564800], length 0

```

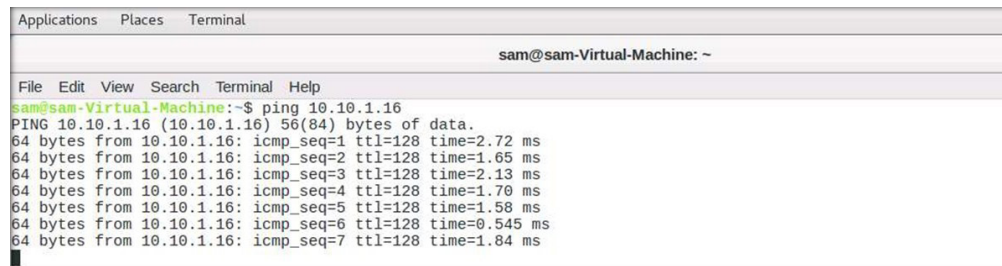
17. Press Ctrl + C to end the packet capture.
18. Now, capture packets from specific source and destination IP.
19. Type `sudo tcpdump -i eth0 src 10.10.1.16` in the terminal and press Enter to capture packets from the specific source on the machine interface.



```
Applications  Places  Terminal
sam@sam-Virtual-Machine: ~
File Edit View Search Terminal Help
sam@sam-Virtual-Machine:~$ sudo tcpdump -i eth0 src 10.10.1.16
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
```

EXERCISE 3: ANALYZE AND EXAMINE VARIOUS NETWORK PACKET HEADERS IN LINUX USING TCPDUMP

20. Open another Terminal and execute the command ping 10.10.1.16 to communicate with the machine. Leave the terminal open.



```
Applications  Places  Terminal
sam@sam-Virtual-Machine: ~
File Edit View Search Terminal Help
sam@sam-Virtual-Machine:~$ ping 10.10.1.16
PING 10.10.1.16 (10.10.1.16) 56(84) bytes of data:
64 bytes from 10.10.1.16: icmp_seq=1 ttl=128 time=2.72 ms
64 bytes from 10.10.1.16: icmp_seq=2 ttl=128 time=1.65 ms
64 bytes from 10.10.1.16: icmp_seq=3 ttl=128 time=2.13 ms
64 bytes from 10.10.1.16: icmp_seq=4 ttl=128 time=1.70 ms
64 bytes from 10.10.1.16: icmp_seq=5 ttl=128 time=1.58 ms
64 bytes from 10.10.1.16: icmp_seq=6 ttl=128 time=0.545 ms
64 bytes from 10.10.1.16: icmp_seq=7 ttl=128 time=1.84 ms
```

EXERCISE 3: ANALYZE AND EXAMINE VARIOUS NETWORK PACKET HEADERS IN LINUX USING TCPDUMP

21. Switch back to the first Terminal and observe the captured ICMP packets.

EXERCISE 3:
ANALYZE AND
EXAMINE VARIOUS
NETWORK PACKET
HEADERS IN LINUX
USING TCPDUMP

```

Applications  Places  Terminal  Thu 04:07  [Speaker] [Power]
sam@sam-Virtual-Machine: ~
File Edit View Search Terminal Help
sam@sam-Virtual-Machine:~$ sudo tcpdump -i eth0 src 10.10.1.16
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
04:07:21.215181 ARP, Reply 10.10.1.16 is-at 02:15:5d:12:99:6a (oui Unknown), length 28
04:07:21.216449 IP 10.10.1.16 > sam-Virtual-Machine: ICMP echo reply, id 11191, seq 1, length 64
04:07:22.216963 IP 10.10.1.16 > sam-Virtual-Machine: ICMP echo reply, id 11191, seq 2, length 64
04:07:23.218211 IP 10.10.1.16 > sam-Virtual-Machine: ICMP echo reply, id 11191, seq 3, length 64
04:07:24.218059 IP 10.10.1.16 > sam-Virtual-Machine: ICMP echo reply, id 11191, seq 4, length 64
04:07:25.218761 IP 10.10.1.16 > sam-Virtual-Machine: ICMP echo reply, id 11191, seq 5, length 64
04:07:26.078472 ARP, Request who-has sam-Virtual-Machine (02:15:5d:12:99:65 (oui Unknown)) tell 10.10.1.16, length 28
04:07:26.218434 IP 10.10.1.16 > sam-Virtual-Machine: ICMP echo reply, id 11191, seq 6, length 64
04:07:27.221162 IP 10.10.1.16 > sam-Virtual-Machine: ICMP echo reply, id 11191, seq 7, length 64
04:07:28.221050 IP 10.10.1.16 > sam-Virtual-Machine: ICMP echo reply, id 11191, seq 8, length 64
04:07:29.238304 IP 10.10.1.16 > sam-Virtual-Machine: ICMP echo reply, id 11191, seq 9, length 64
04:07:30.238875 IP 10.10.1.16 > sam-Virtual-Machine: ICMP echo reply, id 11191, seq 10, length 64
04:07:31.240410 IP 10.10.1.16 > sam-Virtual-Machine: ICMP echo reply, id 11191, seq 11, length 64
04:07:32.237805 IP 10.10.1.16 > sam-Virtual-Machine: ICMP echo reply, id 11191, seq 12, length 64
04:07:33.269350 IP 10.10.1.16 > sam-Virtual-Machine: ICMP echo reply, id 11191, seq 13, length 64
04:07:34.268998 IP 10.10.1.16 > sam-Virtual-Machine: ICMP echo reply, id 11191, seq 14, length 64
04:07:35.268524 IP 10.10.1.16 > sam-Virtual-Machine: ICMP echo reply, id 11191, seq 15, length 64
04:07:36.275815 IP 10.10.1.16 > sam-Virtual-Machine: ICMP echo reply, id 11191, seq 16, length 64
04:07:37.300438 IP 10.10.1.16 > sam-Virtual-Machine: ICMP echo reply, id 11191, seq 17, length 64
04:07:38.301407 IP 10.10.1.16 > sam-Virtual-Machine: ICMP echo reply, id 11191, seq 18, length 64
04:07:39.301207 IP 10.10.1.16 > sam-Virtual-Machine: ICMP echo reply, id 11191, seq 19, length 64
04:07:40.308849 IP 10.10.1.16 > sam-Virtual-Machine: ICMP echo reply, id 11191, seq 20, length 64
04:07:41.310118 IP 10.10.1.16 > sam-Virtual-Machine: ICMP echo reply, id 11191, seq 21, length 64
04:07:42.311446 IP 10.10.1.16 > sam-Virtual-Machine: ICMP echo reply, id 11191, seq 22, length 64
04:07:43.312613 IP 10.10.1.16 > sam-Virtual-Machine: ICMP echo reply, id 11191, seq 23, length 64
04:07:44.310903 IP 10.10.1.16 > sam-Virtual-Machine: ICMP echo reply, id 11191, seq 24, length 64
    
```

22. Close the second Terminal and press Ctrl + C in first Terminal to stop the packet capturing.
23. A security professional can use the tcpdump to capture the traffic.
24. This concludes the demonstration showing how to analyze and examine various network packet headers using tcpdump.
25. Close all open windows.

EXERCISE 3: ANALYZE AND EXAMINE VARIOUS NETWORK PACKET HEADERS IN LINUX USING TCPDUMP

EXERCISE 4: SCAN NETWORK TO IDENTIFY HOSTS IN THE LOCAL NETWORK

Network scanning is the process of gathering additional detailed information about the target by using highly complex and aggressive reconnaissance techniques.

LAB SCENARIO

A security professional must have the required knowledge to perform network scanning to identify active hosts in the entire network. Further, you must scan the machines for open ports and services running on them.

OBJECTIVE

This lab will demonstrate how to use Nmap to perform network scanning.

OVERVIEW NETWORK SCANNING

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

Note: Ensure that Admin Machine-2, Web Server and PfSense Firewall virtual machines are running.

1. Turn on, AD Domain Controller, Attacker Machine-1, Attacker Machine-2, Admin Machine-1, and Android Device virtual machines.
2. Switch to the Attacker Machine-2 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.

Note: If a Question pop-up window appears asking you to update the machine, click No to close the window.

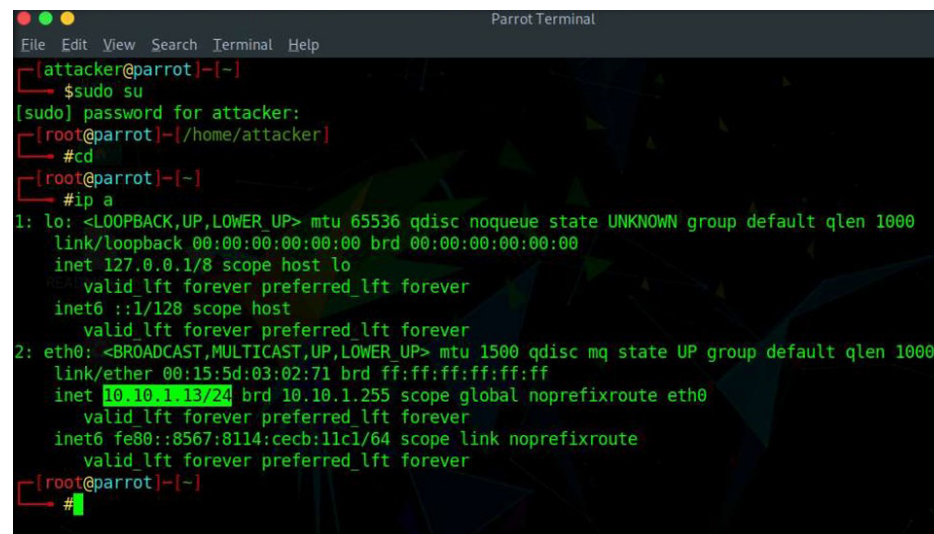
4. Click the MATE Terminal icon at the top of the Desktop window to open a Terminal window.
5. A Parrot Terminal window appears. In the terminal window, type `sudo su` and press Enter to run the programs as a root user.
6. In the [sudo] password for attacker field, type `toor` as a password and press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

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

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

8. In the Terminal window, type `ip a` and press Enter to display information related to network configuration.

Note: Note down the IP address of the machine, here, 10.10.1.13.



```
Parrot Terminal
File Edit View Search Terminal Help
[attacker@parrot]~$ sudo su
[sudo] password for attacker:
[root@parrot]~/home/attacker# cd
[root@parrot]~$ #ip a
#ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
   inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
   link/ether 00:15:5d:03:02:71 brd ff:ff:ff:ff:ff:ff
   inet 10.10.1.13/24 brd 10.10.1.255 scope global noprefixroute eth0
       valid_lft forever preferred_lft forever
   inet6 fe80::8567:8114:cecb:11c1/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
[root@parrot]~$ #
```

9. Type `ip route show` and press Enter to display the IP address of the default gateway.

Note: Note down the IP address of the default gateway, here, 10.10.1.1.

EXERCISE 4: SCAN NETWORK TO IDENTIFY HOSTS IN THE LOCAL NETWORK

```
[root@parrot]-[~]
#ip route show
default via 10.10.1.1 dev eth0 proto static metric 100
10.10.1.0/24 dev eth0 proto kernel scope link src 10.10.1.13 metric 100
[root@parrot]-[~]
#
```

10. Now, type `netdiscover -i eth0 -r 10.10.1.0/24` and press Enter to scan the local network and discover other hosts present in the network.
11. A total of 7 machines will be displayed with details such as MAC Address, Hostname, etc, as shown in the screenshot below.

EXERCISE 4:
SCAN NETWORK
TO IDENTIFY HOSTS
IN THE LOCAL
NETWORK

```

Parrot Terminal
File Edit View Search Terminal Help
Currently scanning: Finished! | Screen View: Unique Hosts
7 Captured ARP Req/Rep packets, from 7 hosts. Total size: 294
-----
IP           At MAC Address  Count  Len  MAC Vendor / Hostname
-----
10.10.1.1    00:15:5d:03:02:6e  1     42  Microsoft Corporation
10.10.1.2    00:15:5d:03:02:72  1     42  Microsoft Corporation
10.10.1.11   00:15:5d:03:02:76  1     42  Microsoft Corporation
10.10.1.16   00:15:5d:03:02:75  1     42  Microsoft Corporation
10.10.1.19   00:15:5d:03:02:73  1     42  Microsoft Corporation
10.10.1.50   00:15:5d:03:02:74  1     42  Microsoft Corporation
10.10.1.79   00:15:5d:03:02:70  1     42  Microsoft Corporation
    
```

12. Press Ctrl+C to terminate the scan.
 13. Switch to the AD Domain Controller virtual machine.
 14. Log in with the credentials CCT\Administrator and admin@123.
- Note:** The network screen appears, click Yes.
15. Click Type here to search icon, type cmd and select Command Prompt from the results.
 16. The Command Prompt window appears, type ipconfig and press Enter to display the details related to network configuration.

EXERCISE 4:
SCAN NETWORK
TO IDENTIFY HOSTS
IN THE LOCAL
NETWORK

```

CA Select Administrator: Command Prompt
Microsoft Windows [Version 10.0.17763.1397]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Administrator.DOMAINCONTROLL.000.001.002>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet 3:

    Connection-specific DNS Suffix . . . :
    Link-local IPv6 Address . . . . . : fe80::b00f:ba58:f665:3ac5%10
    IPv4 Address. . . . . : 10.10.1.19
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 10.10.1.1

C:\Users\Administrator.DOMAINCONTROLL.000.001.002>
    
```

17. Type pathping 10.10.1.13 and press Enter to check the connection between Attacker Machine-2 and AD Domain Controller machine.

Note: It takes a while for the scan to finish.

18. From the results, you can observe that Attacker Machine-2 machine is just 1 hop count away from the AD Domain Controller machine with packet lost count being 0 and success rate is 100, as shown in the screenshot below.

EXERCISE 4:
SCAN NETWORK
TO IDENTIFY HOSTS
IN THE LOCAL
NETWORK

```
C:\Users\Administrator.DOMAINCONTROLL.000.001.002>pathping 10.10.1.13
Tracing route to 10.10.1.13 over a maximum of 30 hops

 0  DomainControll.CCT.com [10.10.1.19]
 1  10.10.1.13

Computing statistics for 25 seconds...
Hop  RTT      Source to Here   This Node/Link   Address
 0                               0/ 100 = 0%      DomainControll.CCT.com [10.10.1.19]
 1    0ms     0/ 100 = 0%     0/ 100 = 0%      |
                               0/ 100 = 0%      10.10.1.13

Trace complete.

C:\Users\Administrator.DOMAINCONTROLL.000.001.002>
C:\Users\Administrator.DOMAINCONTROLL.000.001.002>
```


19. Now, we will use Nmap to discover hosts in the local network.
20. Switch back to the Attacker Machine-2 virtual machine.
21. In the terminal window, type `nmap 10.10.1.0/24` and press Enter to run a basic scan to discover the hosts in the local network.
22. A result appears displaying hosts in the network along with their open ports and service running services, as shown in the screenshot below.

EXERCISE 4:
SCAN NETWORK
TO IDENTIFY HOSTS
IN THE LOCAL
NETWORK

```

Parrot Terminal
File Edit View Search Terminal Help
[root@parrot]-[~]
#nmap 10.10.1.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2021-08-20 08:42 EDT
Nmap scan report for 10.10.1.1
Host is up (0.00065s latency).
Not shown: 997 filtered ports
PORT      STATE SERVICE
53/tcp    open  domain
80/tcp    open  http
443/tcp   open  https
MAC Address: 00:15:5D:03:02:6E (Microsoft)

Nmap scan report for 10.10.1.2
Host is up (0.00049s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
MAC Address: 00:15:5D:03:02:72 (Microsoft)

Nmap scan report for 10.10.1.11
Host is up (0.00034s latency).
Not shown: 999 closed ports
PORT      STATE SERVICE
5555/tcp  open  freeciv
MAC Address: 00:15:5D:03:02:76 (Microsoft)

Nmap scan report for www.moviescope.com (10.10.1.16)
Host is up (0.00082s latency).
    
```

EXERCISE 4:
SCAN NETWORK
TO IDENTIFY HOSTS
IN THE LOCAL
NETWORK

```

Parrot Terminal
File Edit View Search Terminal Help
Nmap scan report for www.moviescope.com (10.10.1.16)
Host is up (0.00082s latency).
Not shown: 990 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
1801/tcp  open  msmq
2103/tcp  open  zephyr-clt
2105/tcp  open  eklogin
2107/tcp  open  msmq-mgmt
3389/tcp  open  ms-wbt-server
MAC Address: 00:15:5D:03:02:75 (Microsoft)

Nmap scan report for 10.10.1.19
Host is up (0.00081s latency).
Not shown: 984 closed ports
PORT      STATE SERVICE
53/tcp    open  domain
88/tcp    open  kerberos-sec
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
389/tcp   open  ldap
445/tcp   open  microsoft-ds
464/tcp   open  kpasswd5
593/tcp   open  http-rpc-epmap
636/tcp   open  ldapssl
1801/tcp  open  msmq
    
```

EXERCISE 4:
SCAN NETWORK
TO IDENTIFY HOSTS
IN THE LOCAL
NETWORK

```

Parrot Terminal
File Edit View Search Terminal Help
593/tcp open  http-rpc-epmap
636/tcp open  ldapssl
1801/tcp open  msmq
2103/tcp open  zephyr-clt
2105/tcp open  eklogin
2107/tcp open  msmq-mgmt
3268/tcp open  globalcatLDAP
3269/tcp open  globalcatLDAPssl
3389/tcp open  ms-wbt-server
MAC Address: 00:15:5D:03:02:73 (Microsoft)

Nmap scan report for 10.10.1.50
Host is up (0.00032s latency).
All 1000 scanned ports on 10.10.1.50 are closed
MAC Address: 00:15:5D:03:02:74 (Microsoft)

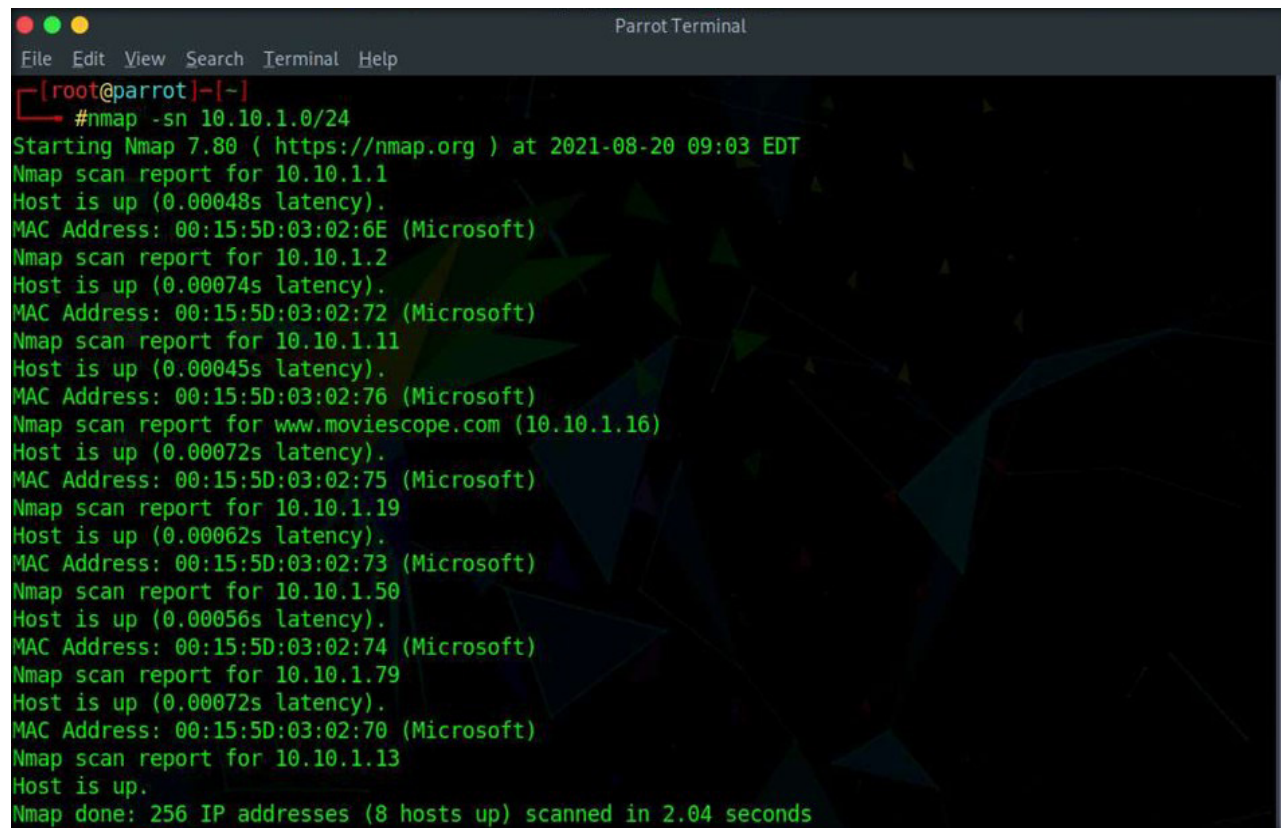
Nmap scan report for 10.10.1.79
Host is up (0.00060s latency).
Not shown: 999 filtered ports
PORT      STATE SERVICE
22/tcp   open  ssh
MAC Address: 00:15:5D:03:02:70 (Microsoft)

Nmap scan report for 10.10.1.13
Host is up (0.00013s latency).
All 1000 scanned ports on 10.10.1.13 are closed

Nmap done: 256 IP addresses (8 hosts up) scanned in 10.66 seconds
[root@parrot]-[~]
#
    
```

- 23. Type `nmap -sn 10.10.1.0/24` and press Enter to scan for active machines in the network.
 - 24. A result appears displaying active hosts in the entire network, as shown in the screenshot below.
- Note:** It takes a while for the results to display.

EXERCISE 4:
SCAN NETWORK
TO IDENTIFY HOSTS
IN THE LOCAL
NETWORK



```

Parrot Terminal
File Edit View Search Terminal Help
[root@parrot]~#
#nmap -sn 10.10.1.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2021-08-20 09:03 EDT
Nmap scan report for 10.10.1.1
Host is up (0.00048s latency).
MAC Address: 00:15:5D:03:02:6E (Microsoft)
Nmap scan report for 10.10.1.2
Host is up (0.00074s latency).
MAC Address: 00:15:5D:03:02:72 (Microsoft)
Nmap scan report for 10.10.1.11
Host is up (0.00045s latency).
MAC Address: 00:15:5D:03:02:76 (Microsoft)
Nmap scan report for www.moviescope.com (10.10.1.16)
Host is up (0.00072s latency).
MAC Address: 00:15:5D:03:02:75 (Microsoft)
Nmap scan report for 10.10.1.19
Host is up (0.00062s latency).
MAC Address: 00:15:5D:03:02:73 (Microsoft)
Nmap scan report for 10.10.1.50
Host is up (0.00056s latency).
MAC Address: 00:15:5D:03:02:74 (Microsoft)
Nmap scan report for 10.10.1.79
Host is up (0.00072s latency).
MAC Address: 00:15:5D:03:02:70 (Microsoft)
Nmap scan report for 10.10.1.13
Host is up.
Nmap done: 256 IP addresses (8 hosts up) scanned in 2.04 seconds
    
```

25. Type `nmap -p 10-300 10.10.1.0/24` and press Enter to scan the range of ports (10-300) in the entire network.
26. A result appears displaying different machines with open ports along with the services running on them.

EXERCISE 4:
SCAN NETWORK
TO IDENTIFY HOSTS
IN THE LOCAL
NETWORK

```

Parrot Terminal
File Edit View Search Terminal Help
[~] #nmap -p 10-300 10.10.1.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2021-08-20 08:59 EDT
Nmap scan report for 10.10.1.1
Host is up (0.00073s latency).
Not shown: 289 filtered ports
PORT      STATE SERVICE
53/tcp    open  domain
80/tcp    open  http
MAC Address: 00:15:5D:03:02:6E (Microsoft)

Nmap scan report for 10.10.1.2
Host is up (0.00049s latency).
Not shown: 289 closed ports
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
MAC Address: 00:15:5D:03:02:72 (Microsoft)

Nmap scan report for 10.10.1.11
Host is up (0.00044s latency).
All 291 scanned ports on 10.10.1.11 are closed
MAC Address: 00:15:5D:03:02:76 (Microsoft)

Nmap scan report for www.moviescope.com (10.10.1.16)
Host is up (0.00052s latency).
Not shown: 287 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
80/tcp    open  http
    
```

27. Type `nmap --top-port 20 10.10.1.0/24` and press Enter to scan for the twenty most common ports.
28. A result appears displaying different top 20 ports along with status as open/close/filtered, as shown in the screenshot below.

EXERCISE 4:
SCAN NETWORK
TO IDENTIFY HOSTS
IN THE LOCAL
NETWORK

```

Parrot Terminal
File Edit View Search Terminal Help
[~]
#nmap --top-port 20 10.10.1.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2021-08-20 09:01 EDT
Nmap scan report for 10.10.1.1
Host is up (0.0015s latency).

```

PORT	STATE	SERVICE
21/tcp	filtered	ftp
22/tcp	filtered	ssh
23/tcp	filtered	telnet
25/tcp	filtered	smtp
53/tcp	open	domain
80/tcp	open	http
110/tcp	filtered	pop3
111/tcp	filtered	rpcbind
135/tcp	filtered	msrpc
139/tcp	filtered	netbios-ssn
143/tcp	filtered	imap
443/tcp	open	https
445/tcp	filtered	microsoft-ds
993/tcp	filtered	imaps
995/tcp	filtered	pop3s
1723/tcp	filtered	pptp
3306/tcp	filtered	mysql
3389/tcp	filtered	ms-wbt-server
5900/tcp	filtered	vnc
8080/tcp	filtered	http-proxy

```

MAC Address: 00:15:5D:03:02:6E (Microsoft)
Nmap scan report for 10.10.1.2

```

EXERCISE 4:
SCAN NETWORK
TO IDENTIFY HOSTS
IN THE LOCAL
NETWORK

```

Parrot Terminal
File Edit View Search Terminal Help

Nmap scan report for 10.10.1.11
Host is up (0.00033s latency).

PORT      STATE SERVICE
21/tcp    closed ftp
22/tcp    closed ssh
23/tcp    closed telnet
25/tcp    closed smtp
53/tcp    closed domain
80/tcp    closed http
110/tcp   closed pop3
111/tcp   closed rpcbind
135/tcp   closed msrpc
139/tcp   closed netbios-ssn
143/tcp   closed imap
443/tcp   closed https
445/tcp   closed microsoft-ds
993/tcp   closed imaps
995/tcp   closed pop3s
1723/tcp  closed pptp
3306/tcp  closed mysql
3389/tcp  closed ms-wbt-server
5900/tcp  closed vnc
8080/tcp  closed http-proxy
MAC Address: 00:15:5D:03:02:76 (Microsoft)

Nmap scan report for www.moviescope.com (10.10.1.16)
Host is up (0.00036s latency).
    
```

29. Type `nmap -sn 10.10.1.0/24` and press Enter to scan for the active machines in the network.
 30. A result appears displaying active hosts in the entire network, as shown in the screenshot below.
- Note:** It takes a while for the results to display.

EXERCISE 4:
SCAN NETWORK
TO IDENTIFY HOSTS
IN THE LOCAL
NETWORK

```

Parrot Terminal
File Edit View Search Terminal Help
[root@parrot]-[~]
#nmap -sn 10.10.1.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2021-08-20 09:03 EDT
Nmap scan report for 10.10.1.1
Host is up (0.00048s latency).
MAC Address: 00:15:5D:03:02:6E (Microsoft)
Nmap scan report for 10.10.1.2
Host is up (0.00074s latency).
MAC Address: 00:15:5D:03:02:72 (Microsoft)
Nmap scan report for 10.10.1.11
Host is up (0.00045s latency).
MAC Address: 00:15:5D:03:02:76 (Microsoft)
Nmap scan report for www.moviescope.com (10.10.1.16)
Host is up (0.00072s latency).
MAC Address: 00:15:5D:03:02:75 (Microsoft)
Nmap scan report for 10.10.1.19
Host is up (0.00062s latency).
MAC Address: 00:15:5D:03:02:73 (Microsoft)
Nmap scan report for 10.10.1.50
Host is up (0.00056s latency).
MAC Address: 00:15:5D:03:02:74 (Microsoft)
Nmap scan report for 10.10.1.79
Host is up (0.00072s latency).
MAC Address: 00:15:5D:03:02:70 (Microsoft)
Nmap scan report for 10.10.1.13
Host is up.
Nmap done: 256 IP addresses (8 hosts up) scanned in 2.04 seconds
    
```


31. Now, we will perform a detailed scan on one host (here, AD Domain Controller machine (10.10.1.19)), to do so, type `nmap -A 10.10.1.19` and press Enter.
32. Nmap scans the target machine and displays information such as open ports and services, device type, details of OS, etc., as shown in the screenshot below.

EXERCISE 4:
SCAN NETWORK
TO IDENTIFY HOSTS
IN THE LOCAL
NETWORK

```

Parrot Terminal
File Edit View Search Terminal Help
[~] root@parrot [~]
#nmap -A 10.10.1.19
Starting Nmap 7.80 ( https://nmap.org ) at 2021-08-20 08:51 EDT
Nmap scan report for 10.10.1.19
Host is up (0.00064s latency).
Not shown: 984 closed ports
PORT      STATE SERVICE      VERSION
53/tcp    open  domain?
|
| fingerprint-strings:
|   DNSVersionBindReqTCP:
|     version
|     bind
|
88/tcp    open  kerberos-sec Microsoft Windows Kerberos (server time: 2021-08-20 12:51:40Z)
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn Microsoft Windows netbios-ssn
389/tcp   open  ldap         Microsoft Windows Active Directory LDAP (Domain: CCT.com0., Site: Default-First-Site-Name)
445/tcp   open  microsoft-ds?
464/tcp   open  kpasswd5?
593/tcp   open  ncacn_http   Microsoft Windows RPC over HTTP 1.0
636/tcp   open  tcpwrapped
1801/tcp  open  msmq?
2103/tcp  open  msrpc        Microsoft Windows RPC
2105/tcp  open  msrpc        Microsoft Windows RPC
2107/tcp  open  msrpc        Microsoft Windows RPC
3268/tcp  open  ldap         Microsoft Windows Active Directory LDAP (Domain: CCT.com0., Site: Default-First-Site-Name)
3269/tcp  open  tcpwrapped
3389/tcp  open  ms-wbt-server Microsoft Terminal Services
| rdp-ntlm-info:

```

33. In the terminal window, type `nmap -sT -v 10.10.1.19` and press Enter.

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

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

```
Parrot Terminal
File Edit View Search Terminal Help
[~]
[~] #nmap -sT -v 10.10.1.19
Starting Nmap 7.80 ( https://nmap.org ) at 2021-08-23 05:17 EDT
Initiating ARP Ping Scan at 05:17
Scanning 10.10.1.19 [1 port]
Completed ARP Ping Scan at 05:17, 0.04s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 05:17
Completed Parallel DNS resolution of 1 host. at 05:17, 0.00s elapsed
Initiating Connect Scan at 05:17
Scanning 10.10.1.19 [1000 ports]
Discovered open port 53/tcp on 10.10.1.19
Discovered open port 445/tcp on 10.10.1.19
Discovered open port 3389/tcp on 10.10.1.19
Discovered open port 139/tcp on 10.10.1.19
Discovered open port 135/tcp on 10.10.1.19
Discovered open port 3269/tcp on 10.10.1.19
Discovered open port 2107/tcp on 10.10.1.19
Discovered open port 2105/tcp on 10.10.1.19
Discovered open port 389/tcp on 10.10.1.19
Discovered open port 88/tcp on 10.10.1.19
Discovered open port 464/tcp on 10.10.1.19
Discovered open port 3268/tcp on 10.10.1.19
Discovered open port 593/tcp on 10.10.1.19
Discovered open port 636/tcp on 10.10.1.19
Discovered open port 2103/tcp on 10.10.1.19
Discovered open port 1801/tcp on 10.10.1.19
Completed Connect Scan at 05:17, 2.35s elapsed (1000 total ports)
Nmap scan report for 10.10.1.19
Host is up (0.00036s latency).
Not shown: 984 closed ports
```

EXERCISE 4: SCAN NETWORK TO IDENTIFY HOSTS IN THE LOCAL NETWORK

EXERCISE 4: SCAN NETWORK TO IDENTIFY HOSTS IN THE LOCAL NETWORK

```
Parrot Terminal
File Edit View Search Terminal Help
Discovered open port 2103/tcp on 10.10.1.19
Discovered open port 1801/tcp on 10.10.1.19
Completed Connect Scan at 05:17, 2.35s elapsed (1000 total ports)
Nmap scan report for 10.10.1.19
Host is up (0.00036s latency).
Not shown: 984 closed ports
PORT      STATE SERVICE
53/tcp    open  domain
88/tcp    open  kerberos-sec
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
389/tcp   open  ldap
445/tcp   open  microsoft-ds
464/tcp   open  kpasswd5
593/tcp   open  http-rpc-epmap
636/tcp   open  ldapssl
1801/tcp  open  msmq
2103/tcp  open  zephyr-clt
2105/tcp  open  eklogin
2107/tcp  open  msmq-mgmt
3268/tcp  open  globalcatLDAP
3269/tcp  open  globalcatLDAPssl
3389/tcp  open  ms-wbt-server
MAC Address: 02:15:5D:12:9F:99 (Unknown)

Read data files from: /usr/bin/./share/nmap
Nmap done: 1 IP address (1 host up) scanned in 2.53 seconds
Raw packets sent: 1 (28B) | Rcvd: 1 (28B)
```

35. This concludes the demonstration showing how to perform network scan using Nmap.
36. Close all open windows.
37. Turn off all the running virtual machines.

EXERCISE 4: SCAN NETWORK TO IDENTIFY HOSTS IN THE LOCAL NETWORK

EC-Council

