COMPUTER NETWORKS LABORATORY UE19CS255

4th Semester, Academic Year 2021-22

NAME: R Sharmila

SRN: PES2UG19CS309

SECTION: E

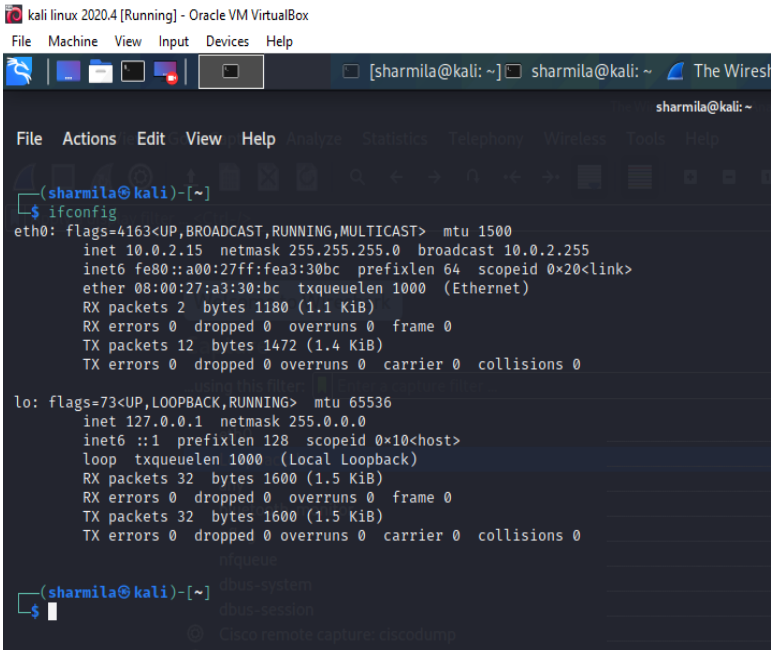
Date: 24-1-2021

Week: 1

<u>Objective Experiment:</u> Study and understand basic networking tools-Wireshark , Tcpdump, Ping, Traceroute and Netcat

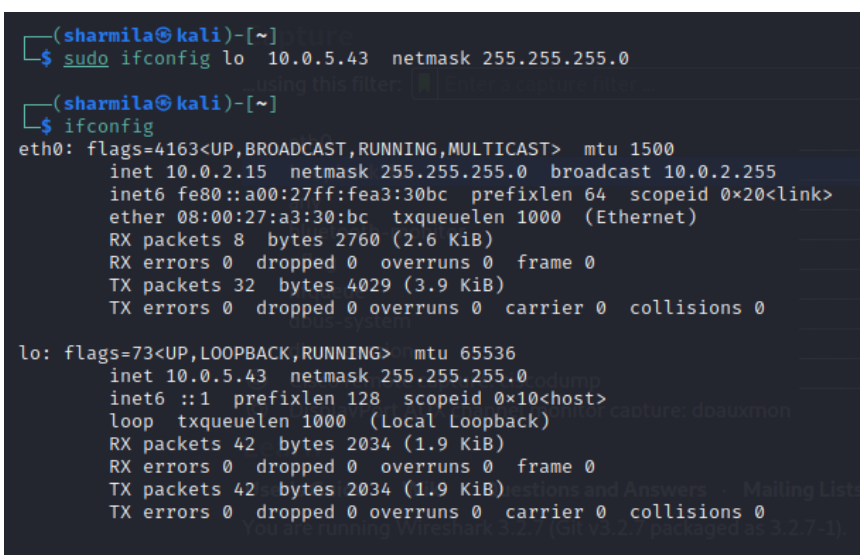
Task1: Linux Interface Configuration

Step 1: To display status of all active network interfaces.



Interface name	Ip address(IPv4/IPv6)	MAC address
eth0	10.0.2.15(IPv4) Fe80::a00:27ff:fea3:30bc(IPv6)	08:00:27:a3:30:bc
lo	127.0.0.1(IPv4) ::1(IPv6)	

Step 2: To assign an IP address to an interface



Step 3: To activate / deactivate a network interface, type.

```
┌──(sharmila㉿kali)-[~]
└─$ sudo ifconfig lo up

┌──(sharmila㉿kali)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::a00:27ff:fea3:30bc  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:a3:30:bc  txqueuelen 1000  (Ethernet)
        RX packets 13  bytes 3060 (2.9 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 38  bytes 4459 (4.3 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 42  bytes 2034 (1.9 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 42  bytes 2034 (1.9 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

```
┌──(sharmila㉿kali)-[~]
└─$ sudo ifconfig lo  down

┌──(sharmila㉿kali)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::a00:27ff:fea3:30bc  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:a3:30:bc  txqueuelen 1000  (Ethernet)
        RX packets 13  bytes 3060 (2.9 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 38  bytes 4459 (4.3 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

┌──(sharmila㉿kali)-[~]
└─$
```

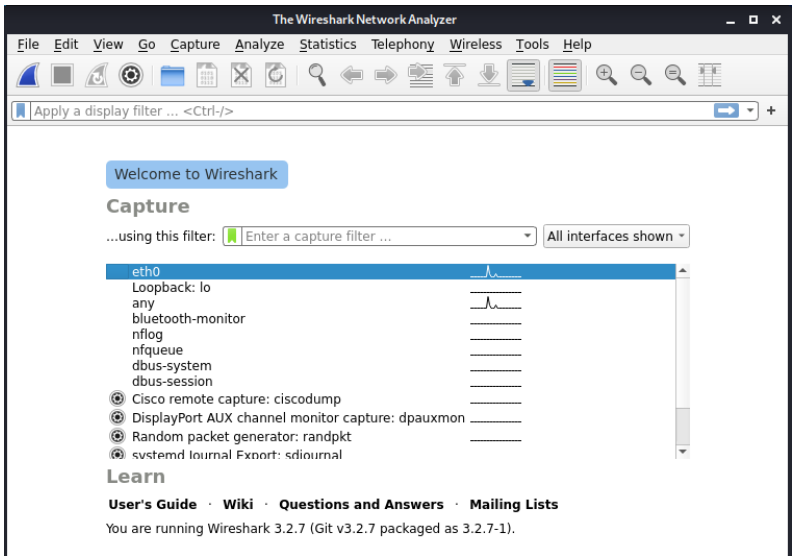
Step 4: To show the current neighbor table in kernel, type

```
┌──(sharmila㉿kali)-[~]
└─$ ip neigh
10.0.2.2 dev eth0 lladdr 52:54:00:12:35:02 STALE
```

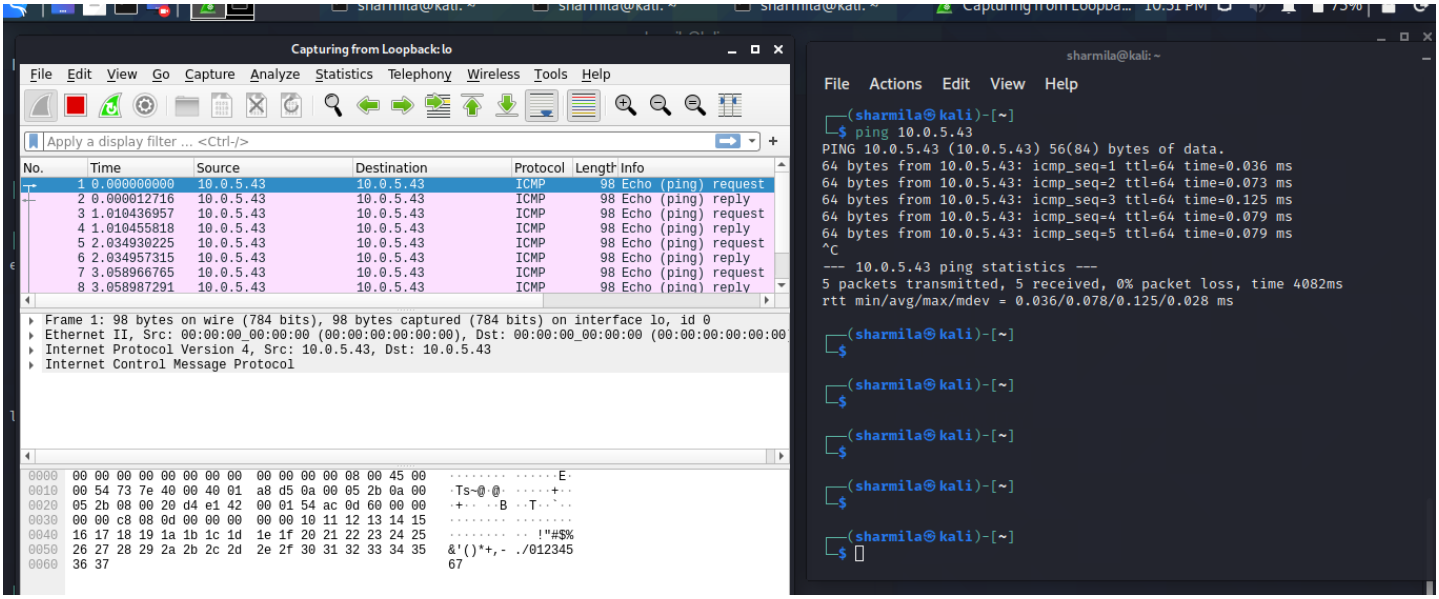
Task 2: Ping PDU (Packet Data Units or Packets) Capture

Step 1: Assign an IP address to the system (Host). Note: IP address of your system should be 10.0. your_section. your_sno.

Step 2: Launch Wireshark and select 'any' interface



Step 3: In terminal, type ping 10.0. your_section. your_sno



OBSERVATIONS TO BE MADE

Step 4: Analyse the following in Terminal

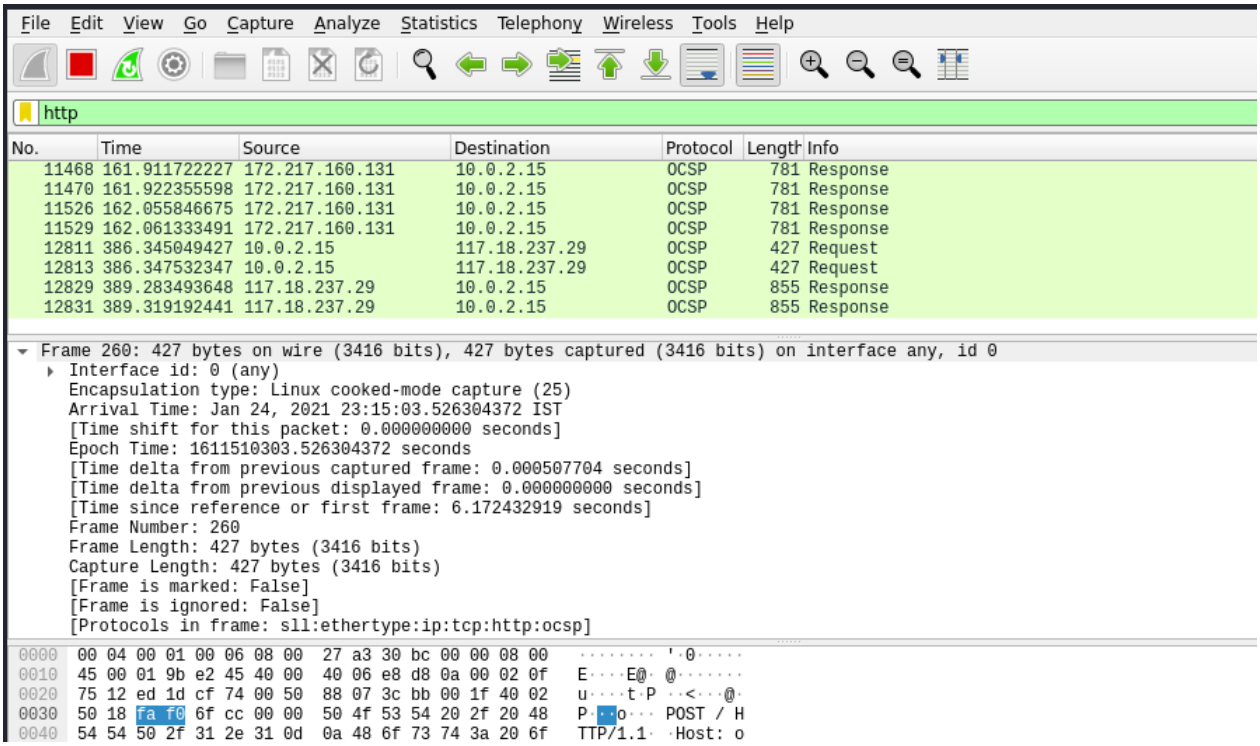
- TTL - 64
- PROTOCOL USED BY PING - ICMP
- TIME - 4081ms

Step 5: Analyse the following in Wireshark

DETAILS	FIRST ECHO REQUEST	FIRST ECHO REPLY
Frame Number	1	2
Source IP address	10.0.5.43	10.0.5.43
Destination IP address	10.0.5.43	10.0.5.43
ICMP Type Value	8	0
ICMP Code Value	0	0
Source Ethernet Address	Source: 00:00:00_00:00:00 (00:00:00:00:00:00)	Source: 00:00:00_00:00:00 (00:00:00:00:00:00)
Destination Ethernet Address	Destination: 00:00:00_00:00:00 (00:00:00:00:00:00)	Destination: 00:00:00_00:00:00 (00:00:00:00:00:00)
Internet Protocol Version	0100 .... = Version: 4	0100 .... = Version: 4
Time To Live (TTL) Value	ttl=64	ttl=64

TASK 3: HTTP PDU CAPTURE Using Wireshark's Filter feature

Step 1: Launch Wireshark and select 'any' interface. On the Filter toolbar, type-in 'http' and press enter
Step 2: Open Firefox browser, and browse www.flipkart.com Observations to be made
Step 3: Analyze the first (interaction of host to the web server) and second frame (response of server to the client). By analyzing the filtered frames, complete the table below:



DETAILS	FIRST ECHO REQUEST	FIRST ECHO REPLY
Frame Number	**260**	**293**
Source IP address	**10.0.2.15**	**117.18.237.29**
Destination IP address	**117.18.237.29**	**10.0.2.15**
Source Port	**53108**	**80**
Destination Port	**80**	**53110**
Source Ethernet Address	**PcsCompu_a3:30:bc (08:00:27:a3:30:bc)**	**RealtekU_12:35:02 (52:54:00:12:35:02)**
Destination Ethernet Address	**RealtekU_12:35:02 (52:54:00:12:35:02)**	**PcsCompu_a3:30:bc (08:00:27:a3:30:bc)**

HTTP Request		HTTP Response	
POST	**HTTP/1.1\r\n**	Server	**Server: ECS (tir/CDCC)\r\n**
HOST	**ocsp.digicert.com\r\n**	Content-Type	**Content-Type: application/ocsp-response\r\n**
User-Agent	**Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0\r\n**	Date	**Date: Sun, 24 Jan 2021 17:45:05 GMT\r\n**
Accept-Language	**en-US,en;q=0.5\r\n**	Location	
Accept-Encoding	**gzip, deflate\r\n**	Content-length	**Content-Length: 471\r\n**
Connection	**keep-alive\r\n**	Connection	**Connection: keep-alive\r\n**

Using Wireshark's Follow TCP Stream

Step 1: Make sure the filter is blank. Right-click any packet inside the Packet List Pane, then select 'Follow TCP Stream'. For demo purpose, a packet containing the HTTP GET request "GET / HTTP / 1.1" can be selected.

Step 2: Upon following a TCP stream, screenshot the whole window.

Wireshark · Follow TCP Stream (tcp.stream eq 10) · any

```
POST / HTTP/1.1
Host: ocsp.digicert.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/
78.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/ocsp-request
Content-Length: 83
Connection: keep-alive

0Q0O0M0K0I0..+.........)........N.      ......k......y........v......
{.l...D38._'..HTTP/1.1 200 OK
Accept-Ranges: bytes
Age: 5805
Cache-Control: max-age=131802
Content-Type: application/ocsp-response
Date: Sun, 24 Jan 2021 17:45:05 GMT
Etag: "600cfb4e-1d7"
Expires: Tue, 26 Jan 2021 06:21:47 GMT
Last-Modified: Sun, 24 Jan 2021 04:45:02 GMT
Server: ECS (tir/CDCC)
X-Cache: HIT
Content-Length: 471
Connection: keep-alive

0...
......0....+.....0......0...0.......k......y........v....20210124044502Z0s0q0I0
..+.........)........N.      ......k......y........v......
{.l...D38._'      20210124044502Z    20210131040002Z0
```

5 client pkts, 5 server pkts, 9 turns.

Entire conversation (5,970 bytes) | Show and save data as ASCII | Stream 10

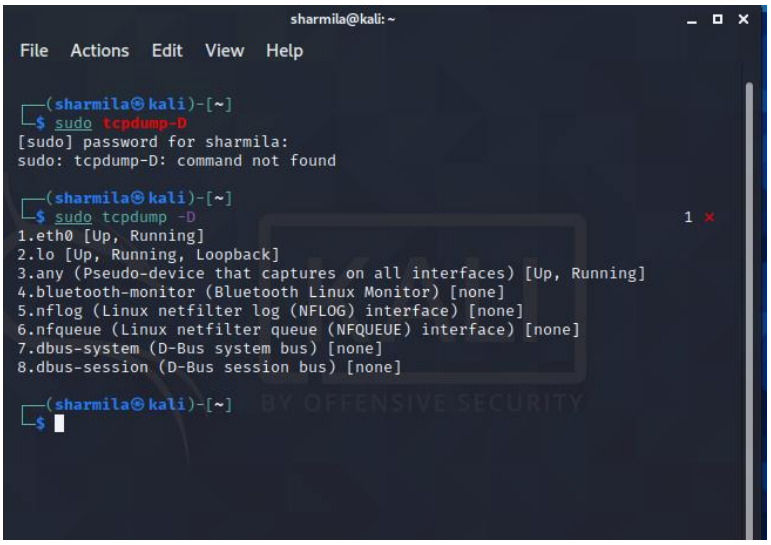
Find: | Find Next

Filter Out This Stream | Print | Save as... | Back | Close | Help

TASK 4: CAPTURING PACKETS WITH TCPDUMP

-

Step 1: Use the command tcpdump -D to see which interfaces are available for capture. sudo tcpdump -D

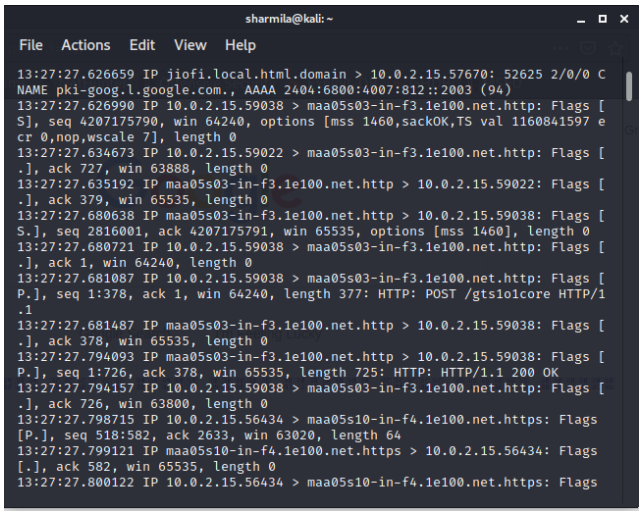


Step 2: Capture all packets in any interface by running this command: sudo tcpdump -i any

Note: Perform some pinging operation while giving above command. Also type www.google.com in browser.

OBSERVATION

Step 3: Understand the output format. Capture all packets in any interface by running this command:



Listen, report the list of link-layer types, report the list of time stamp types, or report the results of compiling a filter expression on interface.

Step 4: To filter packets based on protocol, specifying the protocol in the command line. For example, capture ICMP packets only by using this command: sudo tcpdump-i any -c5 icmp

```
┌──(sharmila㉿kali)-[~]
└─$ sudo tcpdump -i any -c5 icmp                                        1 ×
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on any, link-type LINUX_SLL (Linux cooked v1), capture size 26214
4 bytes
13:34:01.395325 IP 10.0.2.15 > 10.0.2.2: ICMP 10.0.2.15 udp port bootpc unr
eachable, length 556
```

Step 5: Check the packet content. For example, inspect the HTTP content of a web request like this: sudo tcpdump -i any -c10 -nn -A port 80

```
┌──(sharmila㉿kali)-[~]
└─$ sudo tcpdump -i any -c10 -nn -A port 80
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on any, link-type LINUX_SLL (Linux cooked v1), capture size 262144 bytes
16:59:35.985285 IP 10.0.2.15.41578 > 117.18.237.29.80: Flags [.], ack 39616801, win 63920, leng
th 0
E..(..@.@...
...u....j.P.vO{.\.!P...nY..
16:59:35.986164 IP 117.18.237.29.80 > 10.0.2.15.41578: Flags [.], ack 1, win 65535, length 0
E..(^X..@..9u...
....P.j.\.!.vO|P....j........
16:59:46.213080 IP 10.0.2.15.41578 > 117.18.237.29.80: Flags [.], ack 1, win 63920, length 0
E..(..@.@...
...u....j.P.vO{.\.!P...nY..
16:59:46.213624 IP 117.18.237.29.80 > 10.0.2.15.41578: Flags [.], ack 1, win 65535, length 0
E..(^k..@..&u...
....P.j.\.!.vO|P....j........
16:59:56.453562 IP 10.0.2.15.41578 > 117.18.237.29.80: Flags [.], ack 1, win 63920, length 0
E..(..@.@...
...u....j.P.vO{.\.!P...nY..
16:59:56.454233 IP 117.18.237.29.80 > 10.0.2.15.41578: Flags [.], ack 1, win 65535, length 0
E..(^x..@...u...
....P.j.\.!.vO|P....j........
17:00:06.683240 IP 10.0.2.15.41578 > 117.18.237.29.80: Flags [.], ack 1, win 63920, length 0
E..(..@.@...
...u....j.P.vO{.\.!P...nY..
17:00:06.686856 IP 117.18.237.29.80 > 10.0.2.15.41578: Flags [.], ack 1, win 65535, length 0
E..(^...@...u...
....P.j.\.!.vO|P....j........
17:00:16.932599 IP 10.0.2.15.41578 > 117.18.237.29.80: Flags [.], ack 1, win 63920, length 0
E..(..@.@...
17:00:16.932599 IP 10.0.2.15.41578 > 117.18.237.29.80: Flags [.], ack 1, win 63920, length 0
E..(..@.@...
...u....j.P.vO{.\.!P...nY..
17:00:16.932999 IP 117.18.237.29.80 > 10.0.2.15.41578: Flags [.], ack 1, win 65535, length 0
E..(^...@...u...
....P.j.\.!.vO|P....j........
10 packets captured
10 packets received by filter
0 packets dropped by kernel
```

Step 6: To save packets to a file instead of displaying them on screen, use the option -w: sudo tcpdump -i any -c10 -nn -w webserver.pcap port 80

```
┌──(sharmila㉿kali)-[~]
└─$ sudo tcpdump -i any -c10 -nn -w webserver.pcap port 80
tcpdump: listening on any, link-type LINUX_SLL (Linux cooked v1), capture size 262144 bytes
```

TASK 5: PERFORM TRACEROUTE CHECKS

Step 1: Run the traceroute using the following command.

sudo traceroute www.google.com

```
┌──(sharmila㉿kali)-[~]
└─$ sudo traceroute www.google.com
traceroute to www.google.com (142.250.67.196), 30 hops max, 60 byte packets
 1  10.0.2.2 (10.0.2.2)  0.794 ms  0.749 ms  0.736 ms
 2  * * *
 3  * * *
 4  * * *
 5  * * *
 6  * * *
 7  * * *
 8  * * *
 9  * * *
10  * * *
11  * * *
12  * * *
13  * * *
14  * * *
15  * * *
16  * * *
17  * * *
18  * * *
19  * * *
20  * * *
21  * * *
22  * * *
23  * * *
24  * * *
25  * * *
26  * * *
27  * * *
28  * * *
```

Step 2: Analyze destination address of google.com and no. of hops The destination address is 216.58.203.36 and there were 30 hops.

Step 3: To speed up the process, you can disable the mapping of IP addresses with hostnames by using the -n option sudo traceroute -n www.google.com

```
┌──(sharmila㉿kali)-[~]
└─$ sudo traceroute -n www.google.com
traceroute to www.google.com (142.250.67.68), 30 hops max, 60 byte packets
 1  10.0.2.2  0.716 ms  0.682 ms  0.668 ms
 2  * * *
 3  * * *
 4  * * *
 5  * * *
 6  * * *
 7  * * *
 8  * * *
 9  * * *
10  * * *
11  * * *
12  * * *
13  * * *
14  * * *
15  * * *
16  * * *
17  * * *
18  * * *
19  * * *
20  * * *
21  * * *
22  * * *
23  * * *
24  * * *
25  * * *
26  * * *
27  * * *
```

Step 4: The -I option is necessary so that the traceroute uses ICMP.

sudo traceroute -I www.google.com

```
┌──(sharmila㉿kali)-[~]
└─$ sudo traceroute -I www.google.com
traceroute to www.google.com (142.250.67.68), 30 hops max, 60 byte packets
 1  10.0.2.2 (10.0.2.2)  0.251 ms  0.224 ms  0.218 ms
 2  192.168.43.1 (192.168.43.1)  13.845 ms  16.677 ms  16.672 ms
 3  * * *
 4  10.50.123.137 (10.50.123.137)  60.951 ms  60.909 ms  60.876 ms
 5  10.50.123.201 (10.50.123.201)  68.290 ms  70.769 ms  84.627 ms
 6  125.17.138.169 (125.17.138.169)  51.164 ms  57.749 ms  47.270 ms
 7  182.79.198.20 (182.79.198.20)  71.615 ms  44.744 ms  45.920 ms
 8  72.14.208.234 (72.14.208.234)  45.974 ms  75.715 ms  57.342 ms
 9  108.170.234.1 (108.170.234.1)  52.747 ms  62.721 ms  62.848 ms
10  142.250.228.221 (142.250.228.221)  94.761 ms  56.107 ms  99.183 ms
11  maa05s13-in-f4.1e100.net (142.250.67.68)  87.799 ms  88.821 ms  80.270 ms
```

Step 5: By default, traceroute uses icmp (ping) packets. If you'd rather test a TCP connection to gather data more relevant to web server, you can use the -T flag. sudo traceroute -T www.google.com

```
┌──(sharmila㉿kali)-[~]
└─$ sudo traceroute -T www.google.com
traceroute to www.google.com (172.217.167.132), 30 hops max, 60 byte packets
 1  10.0.2.2 (10.0.2.2)  0.816 ms  0.699 ms  0.676 ms
 2  maa03s26-in-f4.1e100.net (172.217.167.132)  54.551 ms  59.000 ms  52.921 ms
```

TASK 6: EXPLORE AN ENTIRE NETWORK FOR INFORMATION (NMAP)

Step 1: You can scan a host using its host name or IP address, for instance. nmap www.pes.edu

```
┌──(sharmila㉿kali)-[~]
└─$ nmap www.pes.edu
Starting Nmap 7.91 ( https://nmap.org ) at 2021-01-25 16:52 IST
Nmap scan report for www.pes.edu (13.71.123.138)
Host is up (0.046s latency).
Not shown: 998 filtered ports
PORT    STATE SERVICE
80/tcp  open  http
443/tcp open  https

Nmap done: 1 IP address (1 host up) scanned in 22.35 seconds
```

Step 2: Alternatively, use an IP address to scan. nmap 163.53.78.128

```
┌──(sharmila㉿kali)-[~]
└─$ nmap 163.53.78.128
Starting Nmap 7.91 ( https://nmap.org ) at 2021-01-25 16:54 IST
Nmap scan report for 163.53.78.128
Host is up (0.047s latency).
Not shown: 998 filtered ports
PORT    STATE SERVICE
80/tcp  open  http
443/tcp open  https

Nmap done: 1 IP address (1 host up) scanned in 16.97 seconds
```

Step 3: Scan multiple IP address or subnet (IPv4) nmap 192.168.1.1 192.168.1.2 192.168.1.3

```
┌──(sharmila㉿kali)-[~]
└─$ nmap 192.168.1.1 192.168.1.2 192.168.1.3

Starting Nmap 7.91 ( https://nmap.org ) at 2021-01-25 16:56 IST
Nmap done: 3 IP addresses (0 hosts up) scanned in 3.09 seconds
```

TASK 7 A): NETCAT AS CHAT TOOL

a) Intra system communication (Using 2 terminals in the same system)

Step 1: Open a terminal (Ctrl+Alt+T). This will act as a Server.

Step 2: Type nc -l any_portnum (For eg., nc -l 1234)

Note: It will goto listening mode

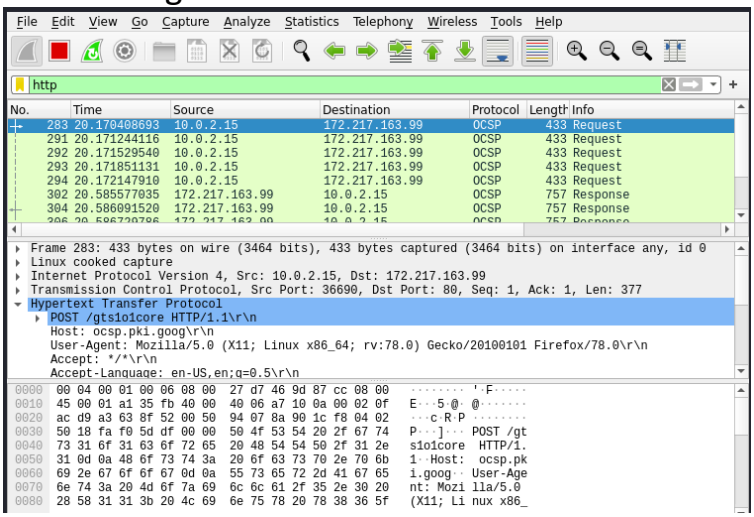
Step 3: Open another terminal and this will act as a client.

Step 4: Type nc portnum Note: portnum should be common in both the terminals (for eg., nc 10.0.2.8 1234)

Step 5: Type anything in client will appear in server

QUESTIONS:

1) Is your browser running HTTP version 1.0 or 1.1?
   It is running with HTTP 1.1 version.



2) How to tell ping to exit after a specified number of ECHO_REQUEST packets?

   Ping continuously sends the ICMP packets until it receives an interrupt .To specify the number of packets we use -c followed by number of packets.

3) How will you identify remote host apps and OS?

   We use nmap -0 -v