

Name: Khushei Meghana Meda

SRN: PES1201800416

Week number: 1

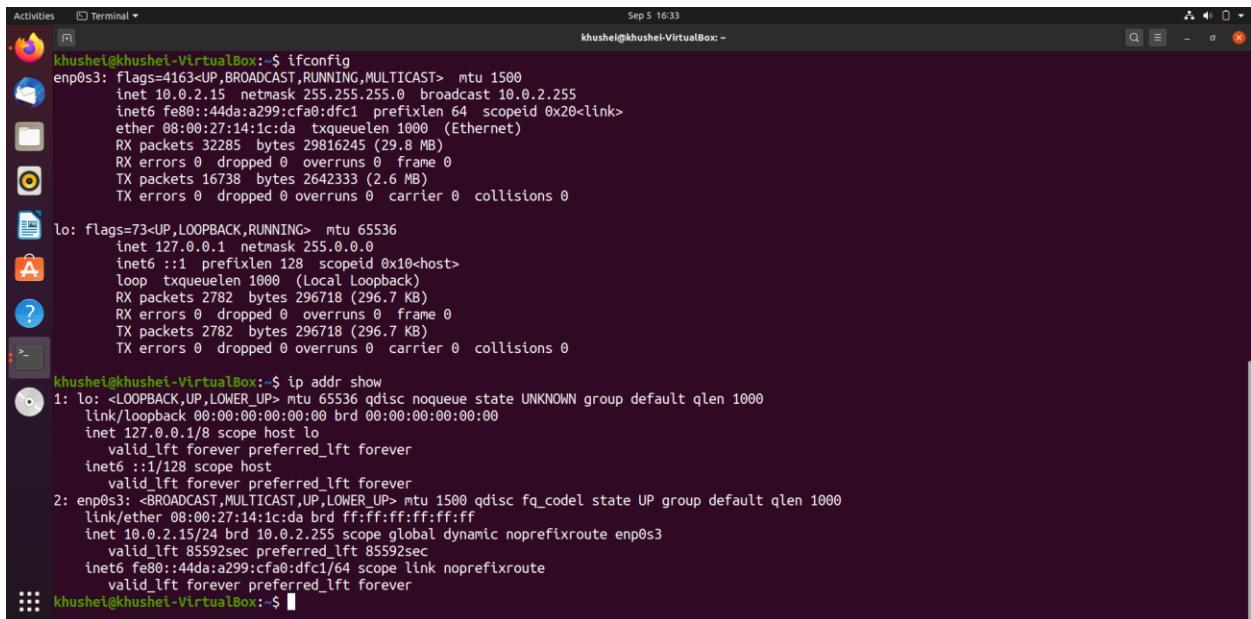
Name of experiment: Learn and Understand Network Tools

Date: 05-09-2020

Objectives of the experiment: Study and understand the basic networking tools - Wireshark, Tcpdump, Ping, Traceroute.

Task1: Linux Interface Configuration (ifconfig / IP command)

Step 1: To display status of all active network interfaces

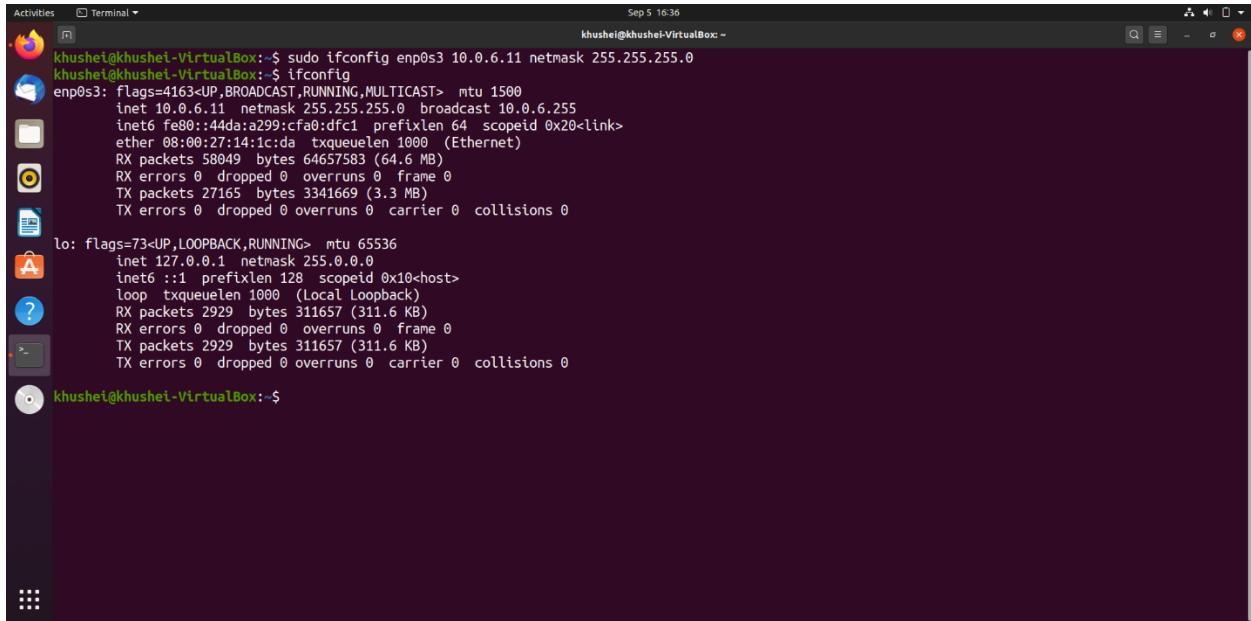


```
khushel@khushei-VirtualBox:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0 broadcast 10.0.2.255
                inet fe80::44da:a299:cfa0:dfc1  prefixlen 64  scopid 0x20<link>
                    ether 08:00:27:14:1c:da  txqueuelen 1000  (Ethernet)
                        RX packets 32285  bytes 29816245 (29.8 MB)
                        RX errors 0  dropped 0  overruns 0  frame 0
                        TX packets 16738  bytes 2642333 (2.6 MB)
                        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  scopid 0x10<host>
                    loop  txqueuelen 1000  (Local Loopback)
                        RX packets 2782  bytes 296718 (296.7 KB)
                        RX errors 0  dropped 0  overruns 0  frame 0
                        TX packets 2782  bytes 296718 (296.7 KB)
                        TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
khushel@khushei-VirtualBox:~$ ip addr show
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: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP>  mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:14:1c:da brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
        valid_lft 85592sec preferred_lft 85592sec
    inet6 fe80::44da:a299:cfa0:dfc1/64  scope link noprefixroute
        valid_lft forever preferred_lft forever
khushel@khushei-VirtualBox:~$
```

Interface Name	IP Address	MAC Address
lo	127.0.0.1	00:00:00:00:00:00
enp0s3	fe80::44da:a299:cfa0:dfc1	08:00:27:14:1c:da

Step 2: To assign an IP address to an interface, use the following command

Ifconfig is executed again to verify the changes have taken effect.



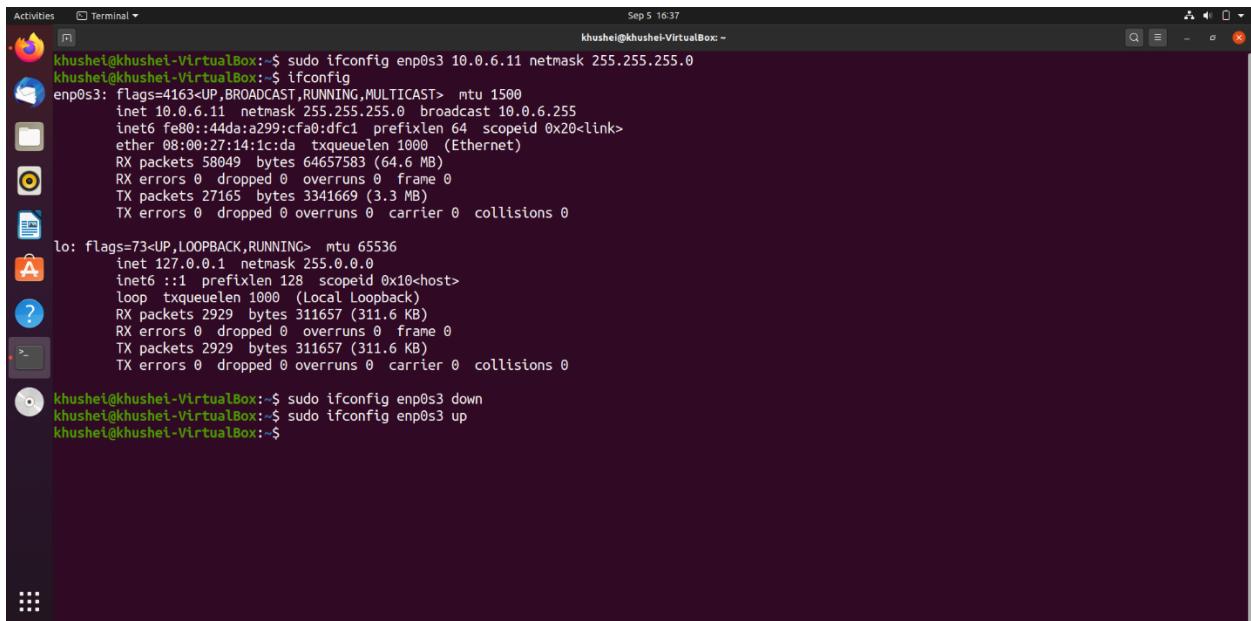
A screenshot of an Ubuntu desktop environment. On the left is a dock with icons for Dash, Home, Applications, and Help. A terminal window is open in the center, showing the command 'sudo ifconfig enp0s3 10.0.6.11 netmask 255.255.255.0' followed by the output of 'ifconfig'. The output shows two interfaces: 'enp0s3' and 'lo'. The 'enp0s3' interface has been configured with the new IP address and netmask. The 'lo' interface remains at 127.0.0.1.

```
khushel@khushel-VirtualBox:~$ sudo ifconfig enp0s3 10.0.6.11 netmask 255.255.255.0
khushel@khushel-VirtualBox:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.6.11  netmask 255.255.255.0 broadcast 10.0.6.255
                inet6 fe80::44da:a299:cfb0:fc1  prefixlen 64  scopeid 0x20<link>
                    ether 08:00:27:14:1c:da  txqueuelen 1000  (Ethernet)
                    RX packets 58049  bytes 64657583 (64.6 MB)
                    RX errors 0  dropped 0  overruns 0  frame 0
                    TX packets 27165  bytes 3341669 (3.3 MB)
                    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 2929  bytes 311657 (311.6 KB)
                    RX errors 0  dropped 0  overruns 0  frame 0
                    TX packets 2929  bytes 311657 (311.6 KB)
                    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

khushel@khushel-VirtualBox:~$
```

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



A screenshot of an Ubuntu desktop environment. The terminal window shows the same configuration as the previous screenshot. It then executes 'sudo ifconfig enp0s3 down' followed by 'sudo ifconfig enp0s3 up'. Finally, it runs 'ip neigh' which outputs a table of neighbors. The table includes entries for the local loopback interface and the external gateway at 10.0.6.254.

```
khushel@khushel-VirtualBox:~$ sudo ifconfig enp0s3 10.0.6.11 netmask 255.255.255.0
khushel@khushel-VirtualBox:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.6.11  netmask 255.255.255.0 broadcast 10.0.6.255
                inet6 fe80::44da:a299:cfb0:fc1  prefixlen 64  scopeid 0x20<link>
                    ether 08:00:27:14:1c:da  txqueuelen 1000  (Ethernet)
                    RX packets 58049  bytes 64657583 (64.6 MB)
                    RX errors 0  dropped 0  overruns 0  frame 0
                    TX packets 27165  bytes 3341669 (3.3 MB)
                    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 2929  bytes 311657 (311.6 KB)
                    RX errors 0  dropped 0  overruns 0  frame 0
                    TX packets 2929  bytes 311657 (311.6 KB)
                    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

khushel@khushel-VirtualBox:~$ sudo ifconfig enp0s3 down
khushel@khushel-VirtualBox:~$ sudo ifconfig enp0s3 up
khushel@khushel-VirtualBox:~$ ip neigh
  dev enp0s3 lladdr 08:00:27:14:1c:da brd ff:ff:ff:ff:ff:ff
    dst 00:00:00:00:00:00 via 10.0.6.254
    dst 00:00:00:00:00:00 via 10.0.6.254
  dev lo lladdr 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
    dst 00:00:00:00:00:00 via 127.0.0.1
```

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

Before executing ip neigh, we execute ifconfig and see that it the IP address has gone back to the original 10.0.2.15 from 10.0.6.11 for enp0s3 interface. I have left it as it is so that ip neigh shows me the current neighbor table while I am connected to the internet. If the IP address were set manually through GUI in step 3, it would have persisted as 10.0.6.11 and in that case we would not get any output for ip neigh as it would not be connected to the internet.

Activities Terminal Sep 5 16:38 khushel@khushel-VirtualBox: ~

```
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 2929 bytes 311657 (311.6 KB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 2929 bytes 311657 (311.6 KB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

khushel@khushel-VirtualBox:~$ sudo ifconfig enp0s3 down
khushel@khushel-VirtualBox:~$ sudo ifconfig enp0s3 up
khushel@khushel-VirtualBox:~$ ifconfig
enp0s3: 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::44da:a299:cf0:dfc1 prefixlen 64 scopeid 0x20<link>
            ether 08:00:27:14:1c:da txqueuelen 1000 (Ethernet)
            RX packets 58209 bytes 64682438 (64.6 MB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 27361 bytes 3362578 (3.3 MB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

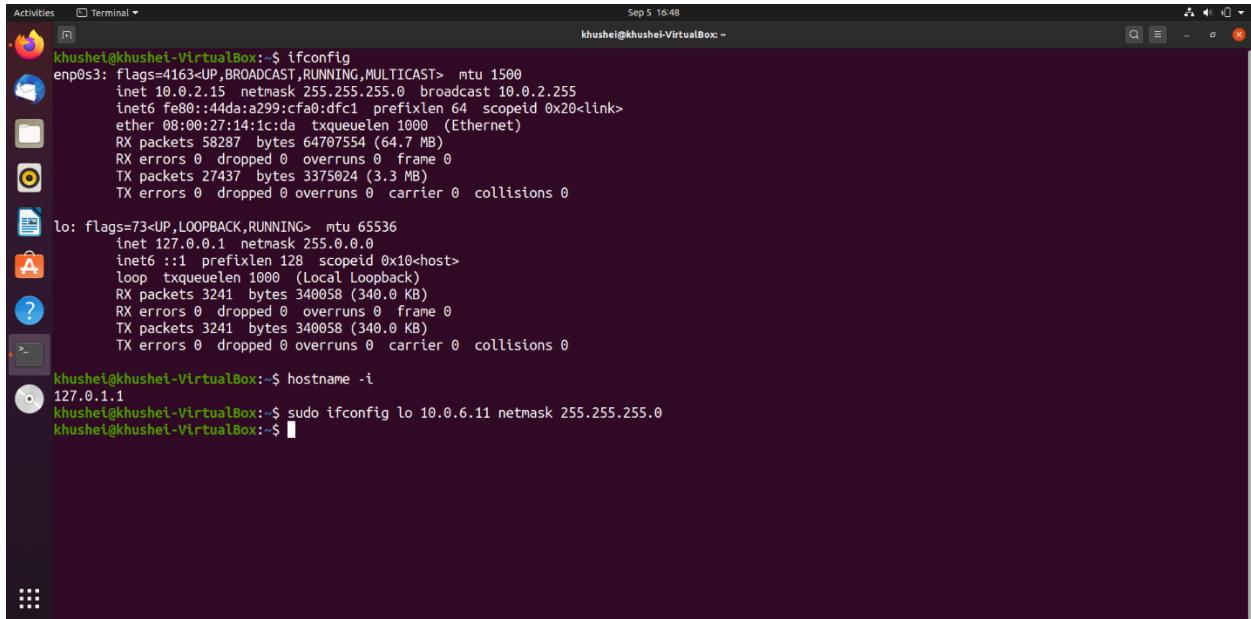
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
            loop txqueuelen 1000 (Local Loopback)
            RX packets 3217 bytes 337422 (337.4 KB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 3217 bytes 337422 (337.4 KB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

khushel@khushel-VirtualBox:~$ ip neigh
10.0.2.2 dev enp0s3 lladdr 52:54:00:12:35:02 REACHABLE
khushel@khushel-VirtualBox:~$ █
```

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

Step 1: Assign an IP address to the system (Host).

We start by checking the current ip address of the host by executing hostname -i command and find it to be 127.0.1.1. So I now change it to 10.0.6.11



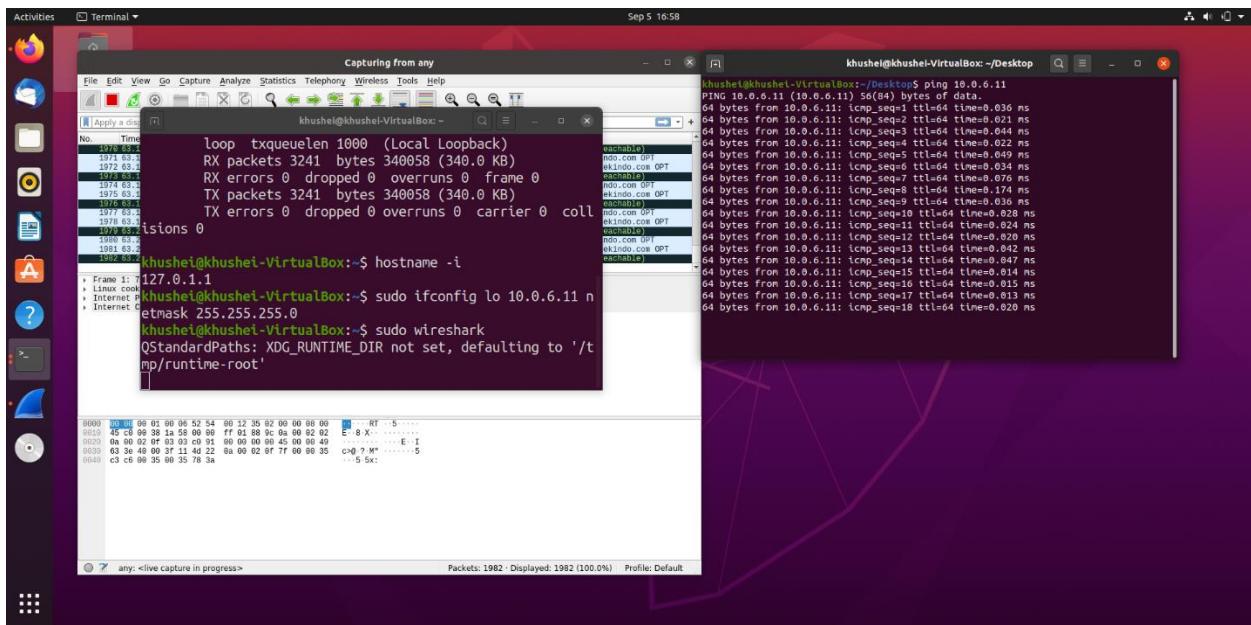
A screenshot of a Linux terminal window titled "Terminal". The window shows a command-line interface with the following text:

```
Activities Terminal Sep 5 16:48
khushet@khushet-VirtualBox:~$ ifconfig
enp0s3: 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::44da:a99::cfa0:fc1 prefixlen 64 scopeid 0x20<link>
            ether 08:00:27:14:1c:da txqueuelen 1000 (Ethernet)
            RX packets 58287 bytes 64707554 (64.7 MB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 27437 bytes 3375024 (3.3 MB)
            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 3241 bytes 340058 (340.0 KB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 3241 bytes 340058 (340.0 KB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
khushet@khushet-VirtualBox:~$ hostname -i
127.0.1.1
khushet@khushet-VirtualBox:~$ sudo ifconfig lo 10.0.6.11 netmask 255.255.255.0
khushet@khushet-VirtualBox:~$
```

Step 2: Launch Wireshark and select 'any' interface

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

Launch wireshark with superuser privileges to find the 'any' interface option



Step 4: Analyze the following in Terminal

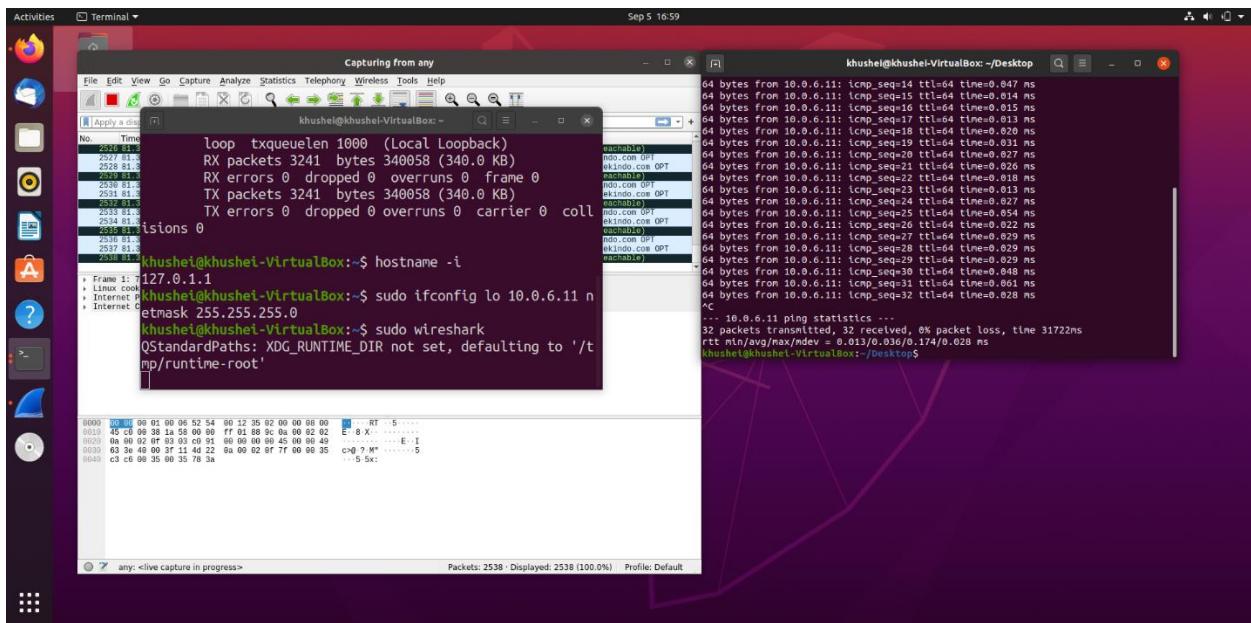
- TTL • Protocol used by ping • Time

Observations:

TTL: 64

Protocol: ICMP

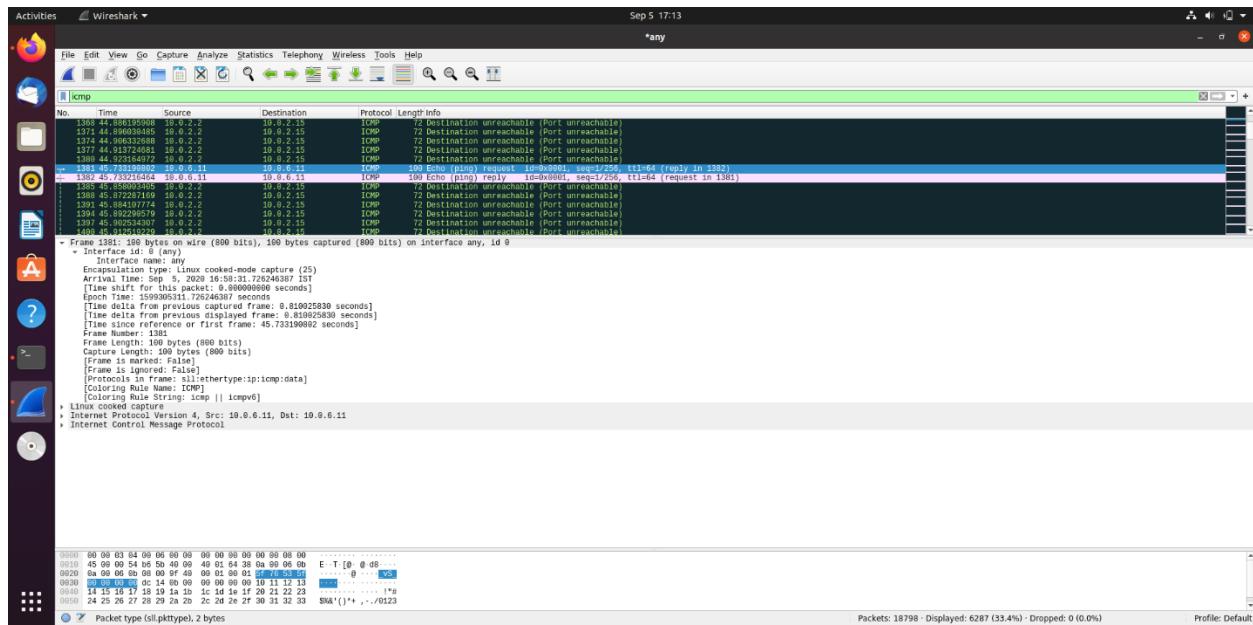
Time for transmission and reception of 32 packets: 31722 ms

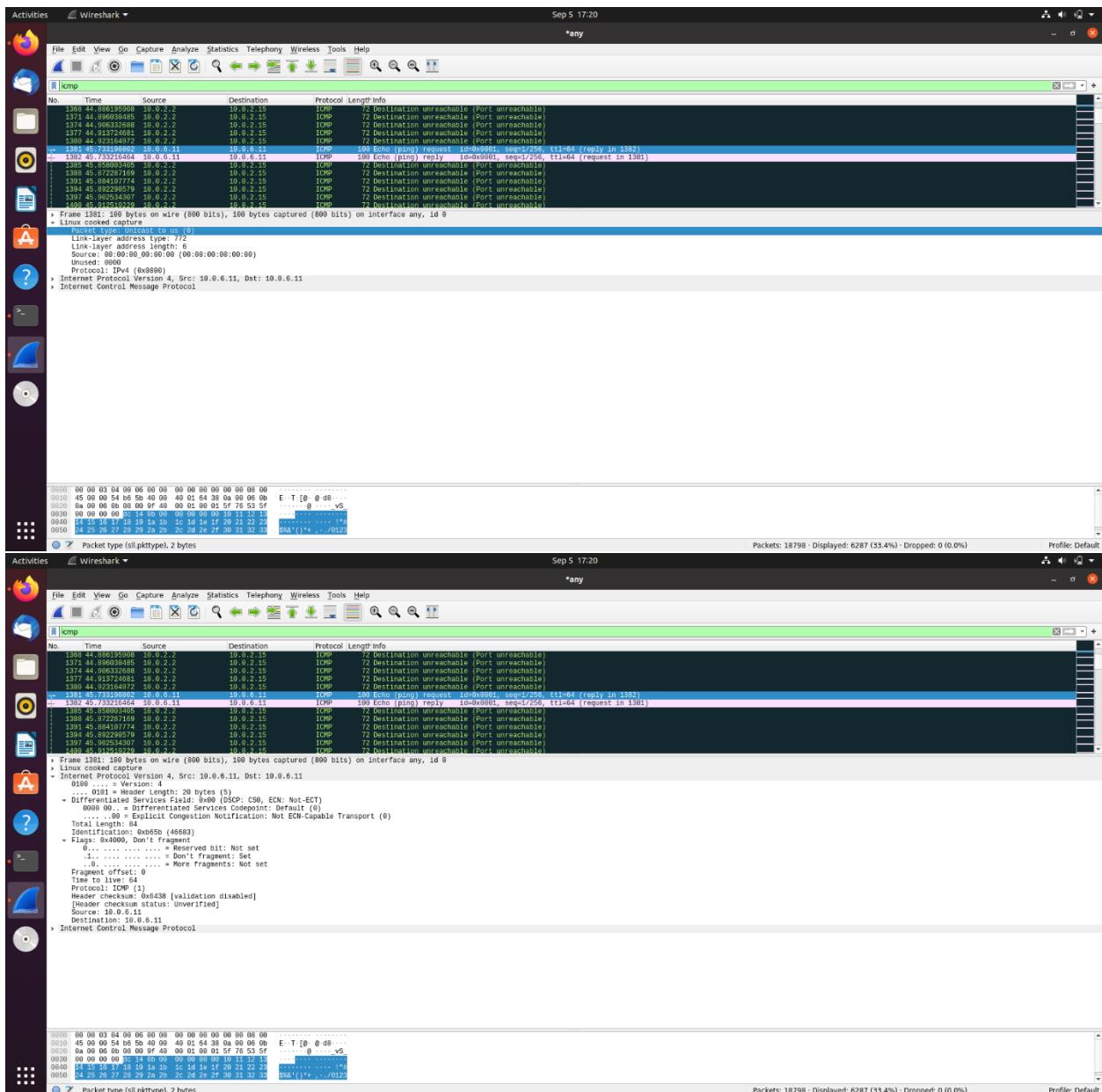


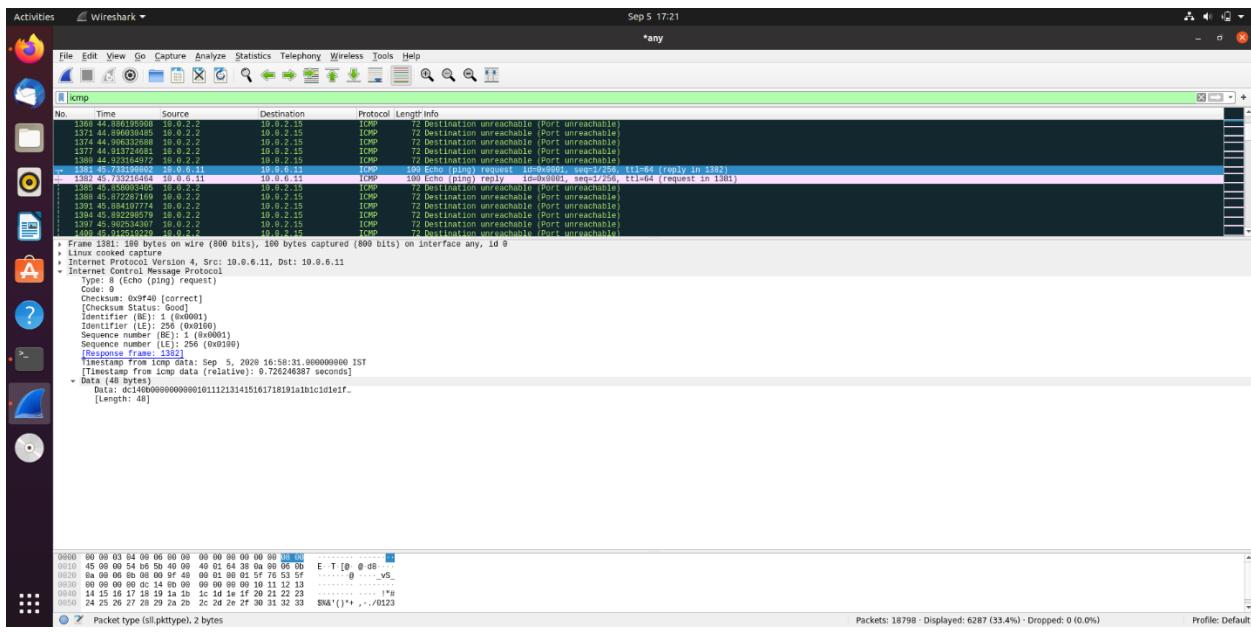
Step 5: Analyze the following in Wireshark

	First Echo Request	First Echo Reply
Frame Number	1381	1382
Source IP address	10.0.6.11	10.0.6.11
Destination IP address	10.0.6.11	10.0.6.11
ICMP Type Value	8	0
ICMP Code Value	0	0
Source Ethernet Address	00:00:00_00:00:00	00:00:00_00:00:00
Destination Ethernet Address	00:00:00_00:00:00	00:00:00_00:00:00
Internet Protocol Version	4	4
Time To Live (TTL) Value	64	64

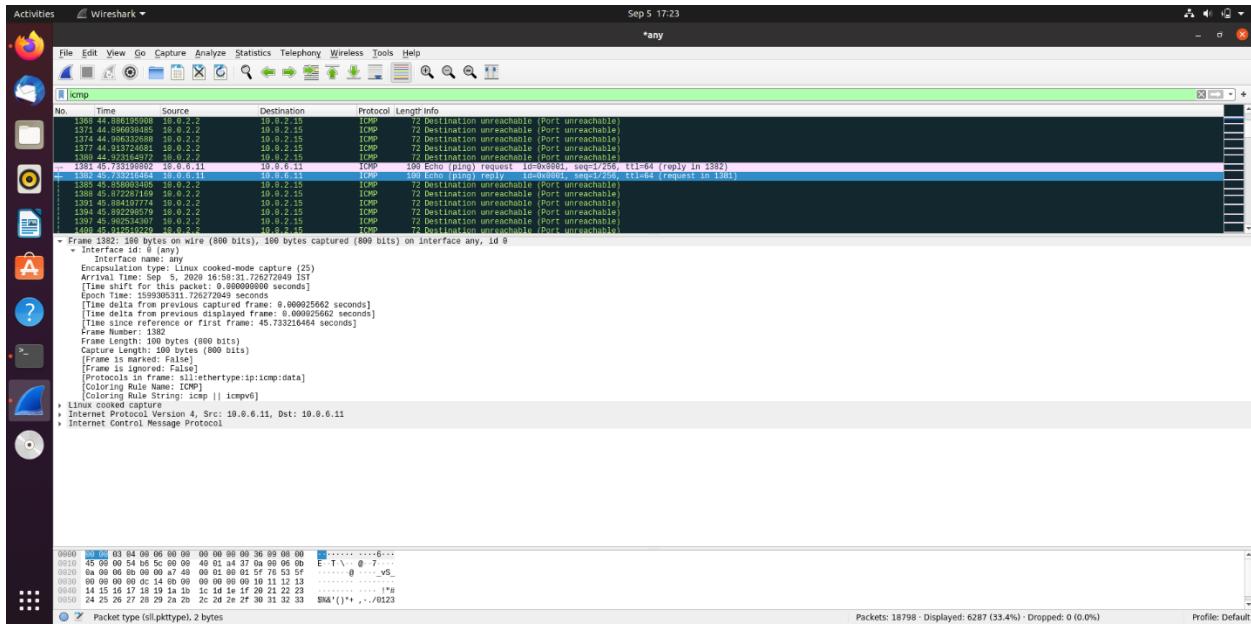
First echo request (all the information has been expanded in the following screenshots):

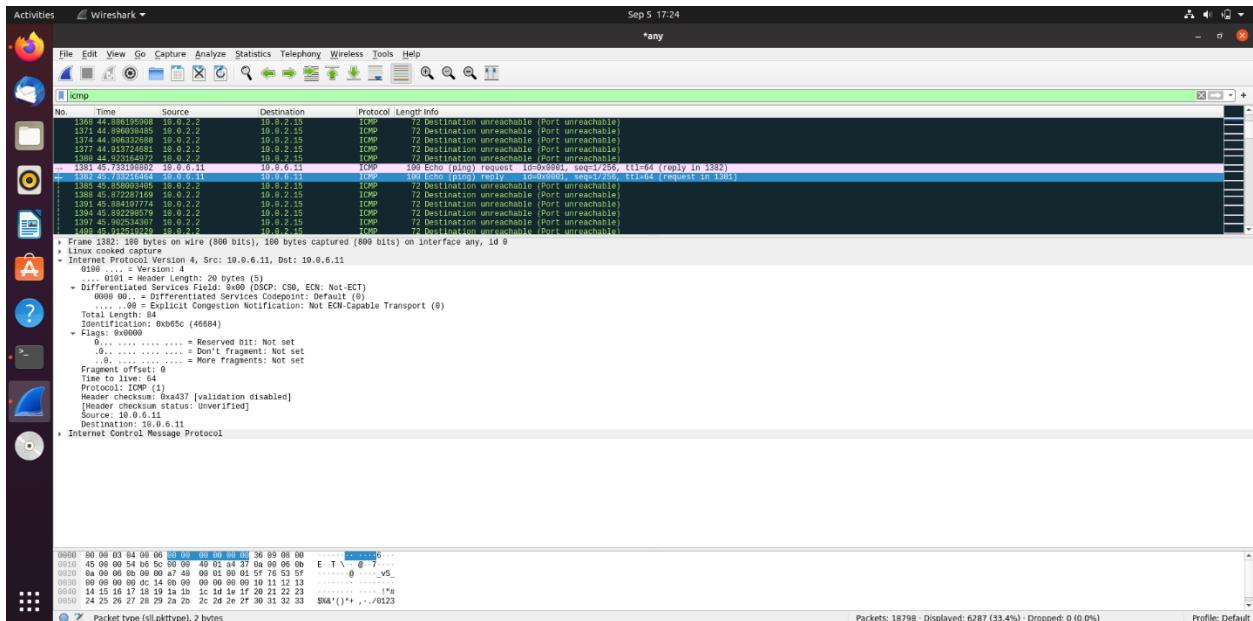
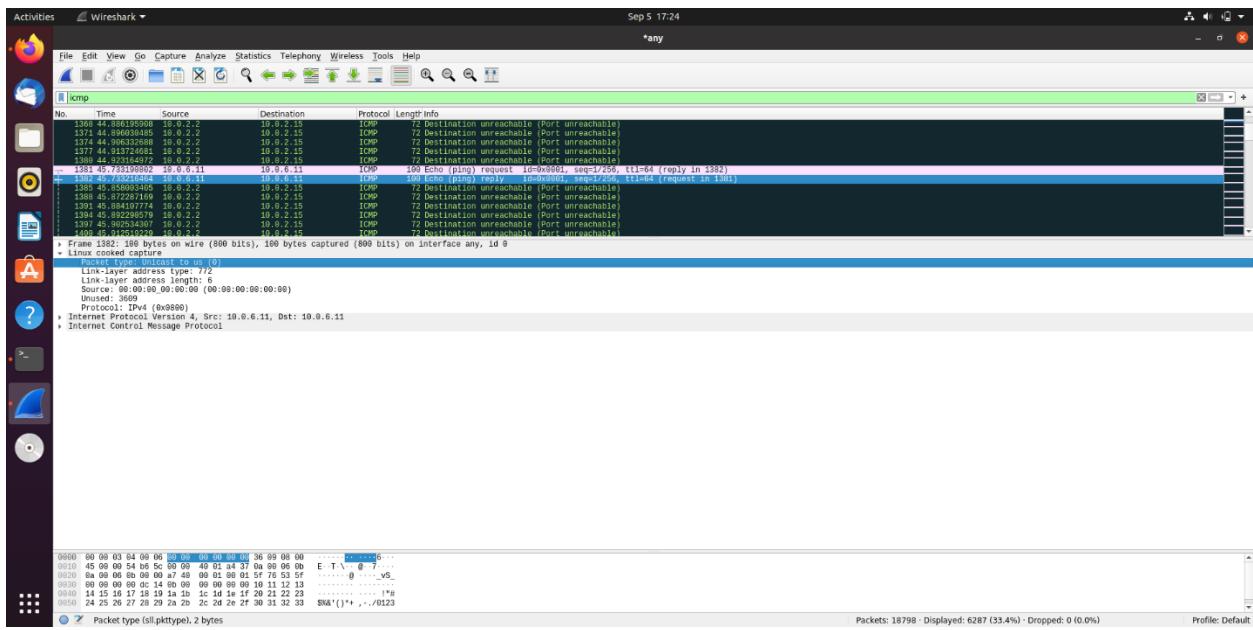


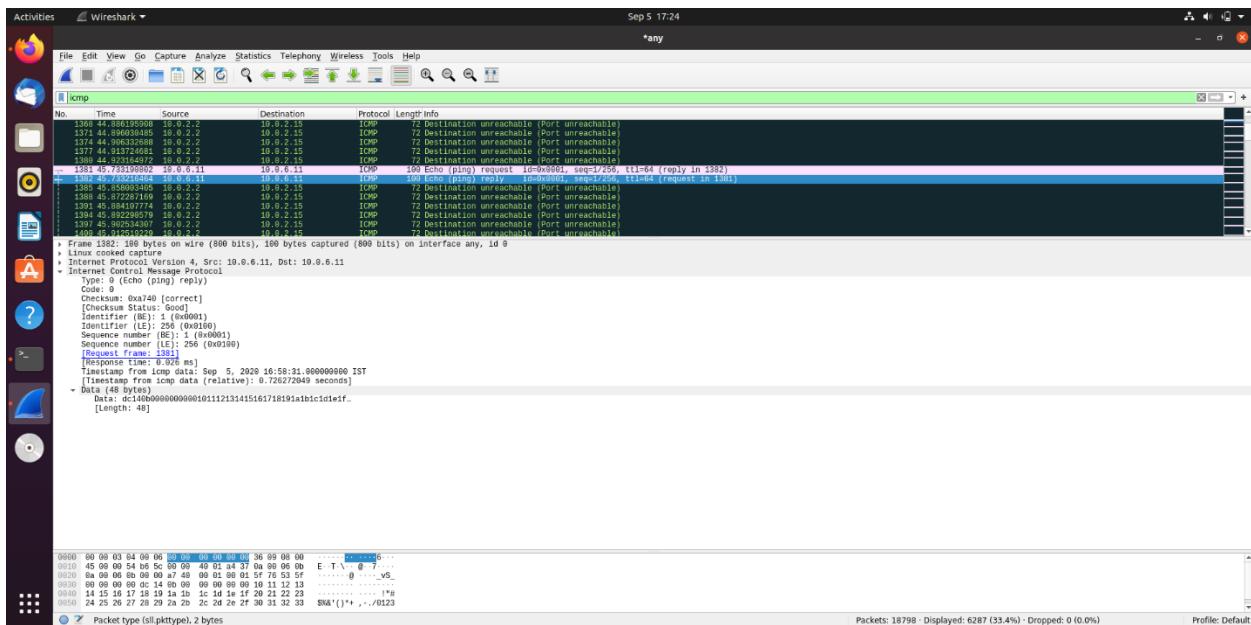




First echo response (all the information has been expanded in the following screenshots):



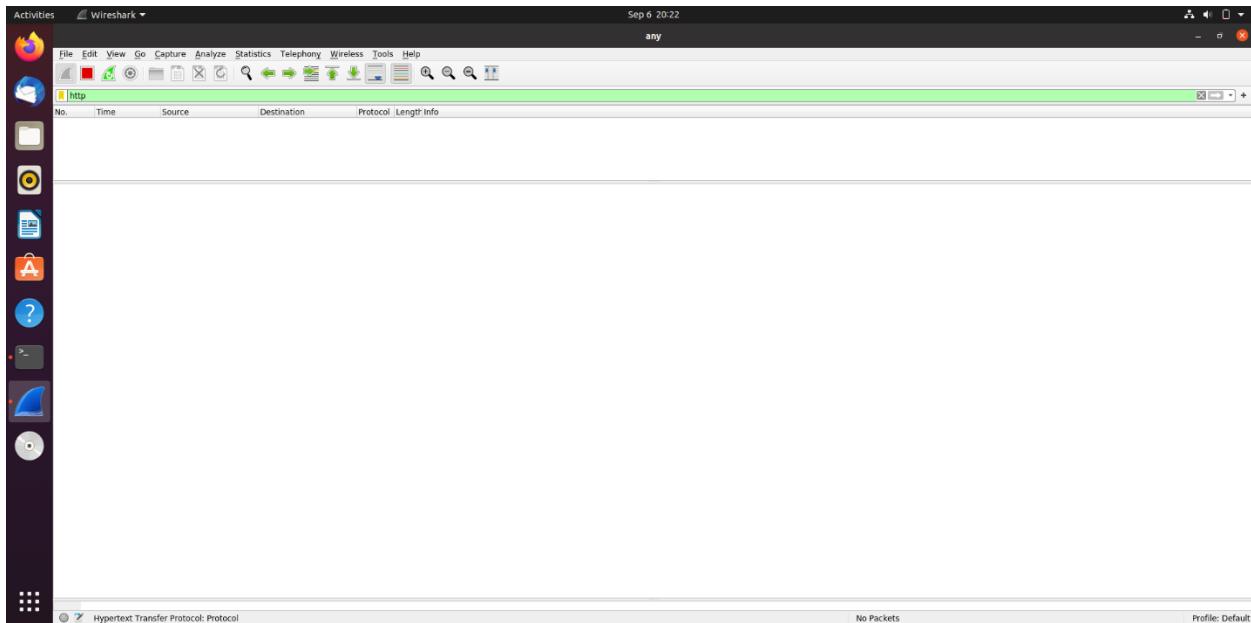




Task 3: HTTP PDU Capture

Using Wireshark's Filter feature

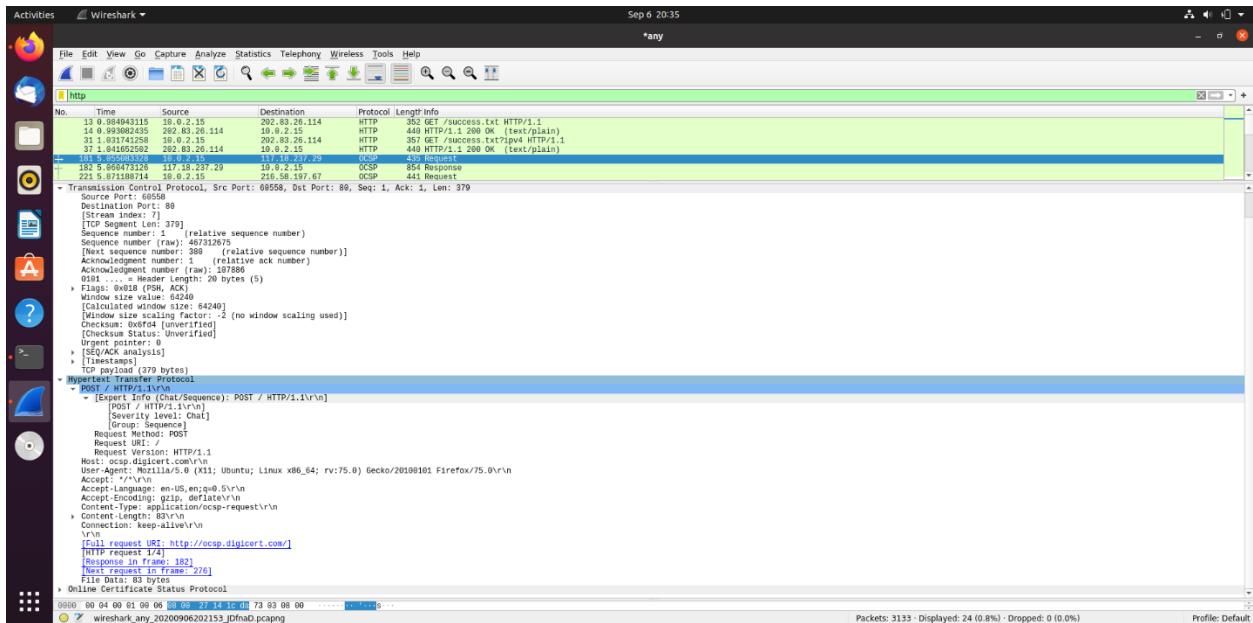
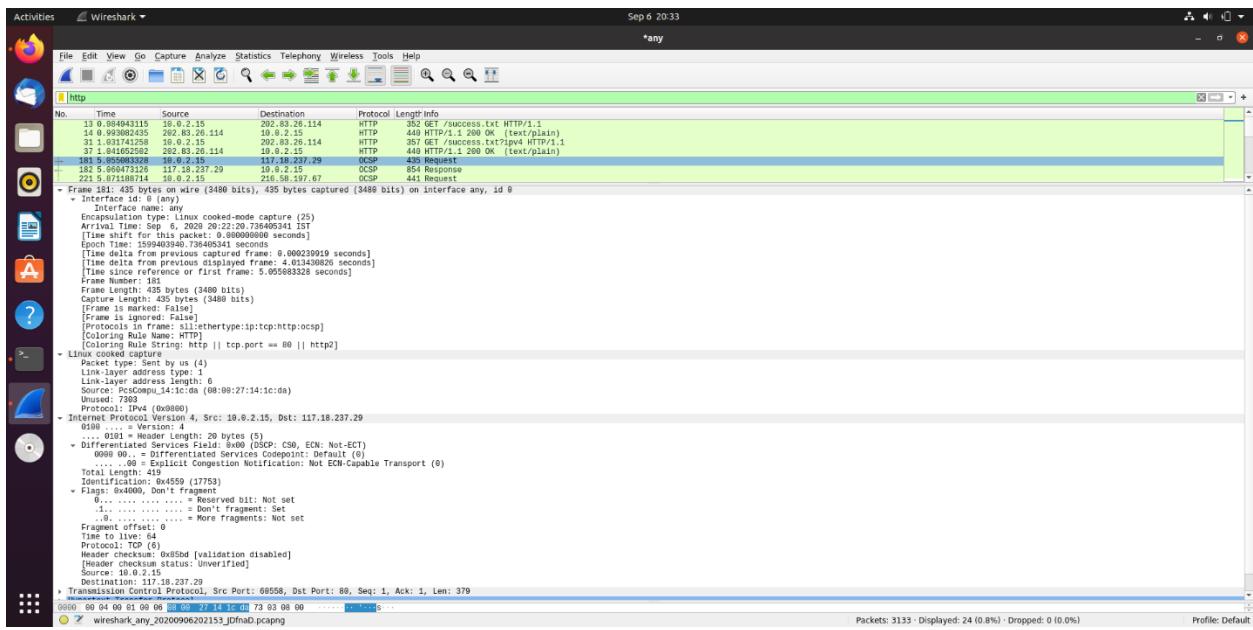
Step 1: Launch Wireshark and select ‘any’ interface. On the Filter toolbar, type-in ‘http’ and press enter



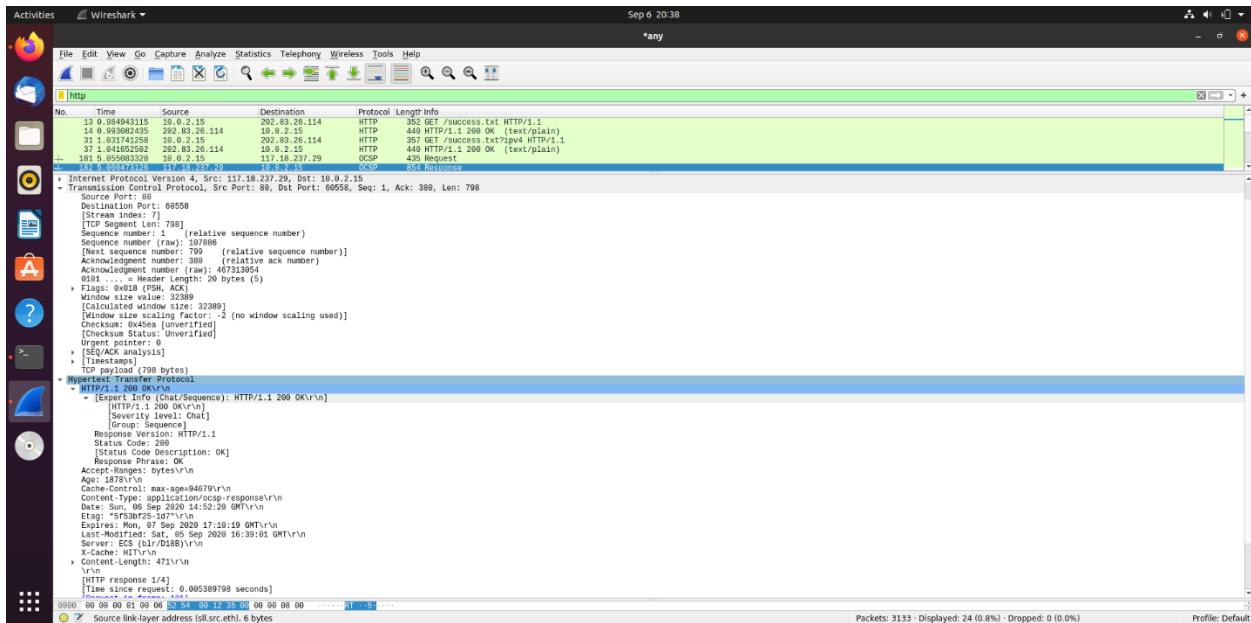
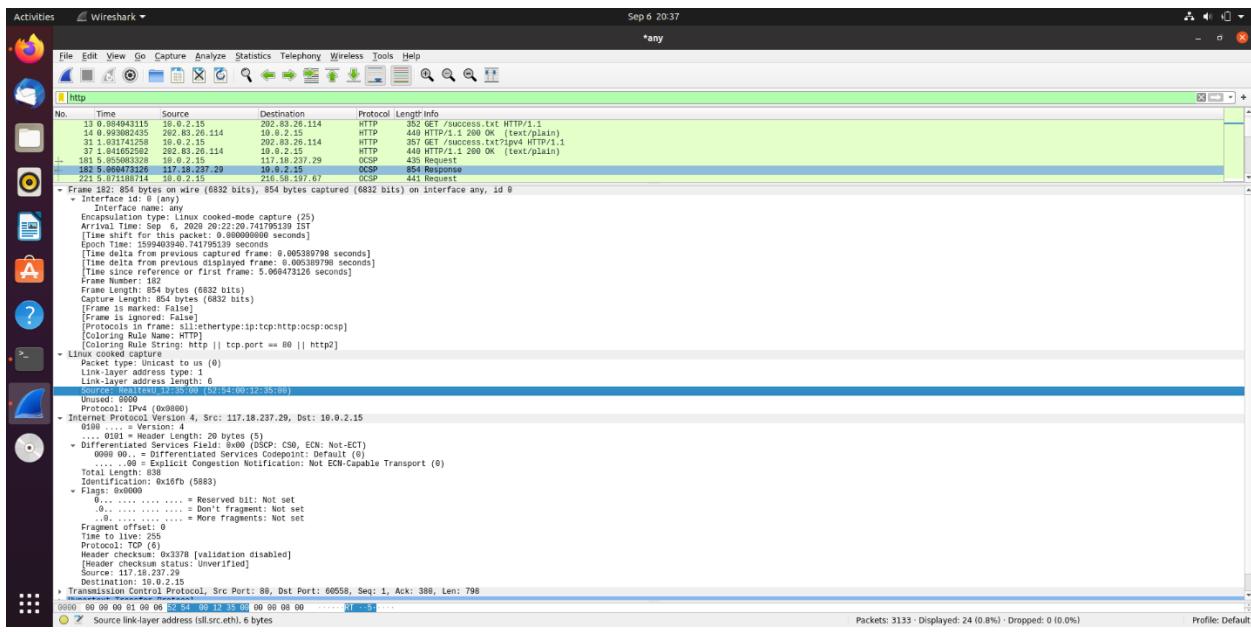
Step 2: Open Firefox browser, and browse www.nyu.edu

Details	First Echo Request	First Echo Reply
Frame Number	181	182
Source Port	60558	80
Destination Port	80	60558
Source IP address	10.0.2.15	117.18.237.29
Destination IP address	117.18.237.29	10.0.2.15
Source Ethernet Address	08:00:27:14:1c:da	52:54:00:12:35:00
Destination Ethernet Address	52:54:00:12:35:00	08:00:27:14:1c:da

First echo request (the required information has been expanded in the following screenshots):



First echo response (the required information has been expanded in the following screenshots):

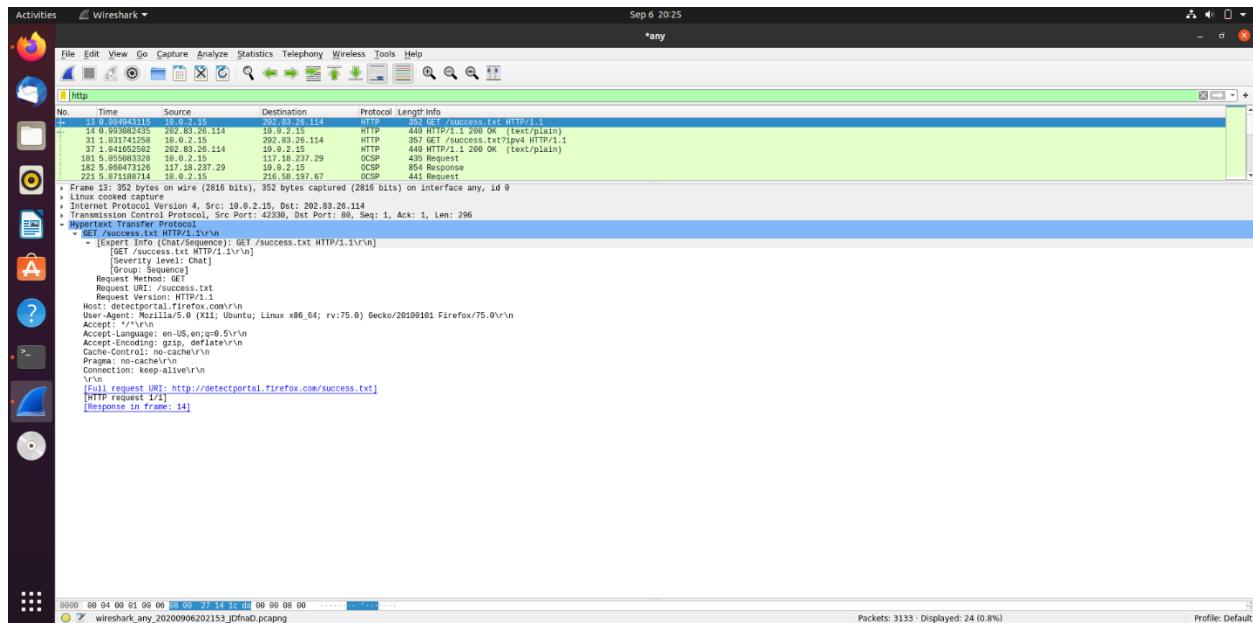


Step 4: Analyze the HTTP request and response and complete the table below.

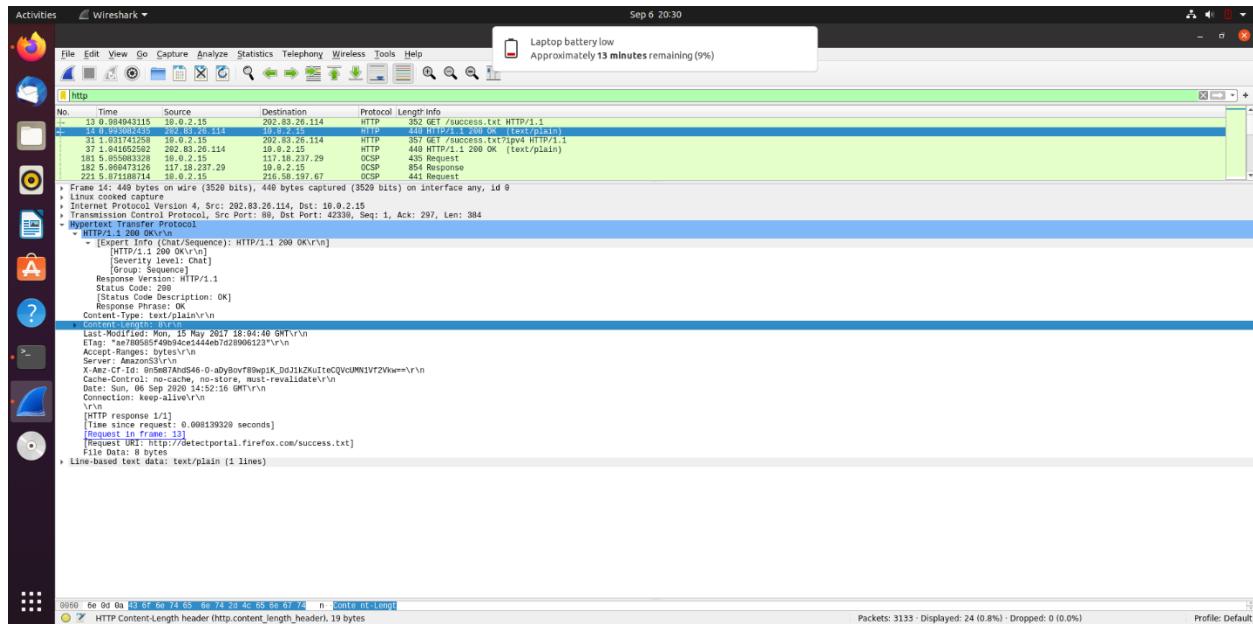
HTTP Request		HTTP Response	
Get	GET /success.txt HTTP/1.1\r\n	Server	AmazonS3
Host	detectportal.firefox.com\r\n	Content-Type	text/plain
User-Agent	Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:75.0) Gecko/20100101 Firefox/75.0	Date	Sun, 06 Sep 2020 14:52:16 GMT

Accept-Language	en-US,en;q=0.5	Location	N/A
Accept-Encoding	gzip, deflate	Content-Length	8\r\n
Connection	keep-alive\r\n	Connection	keep-alive\r\n

HTTP request:



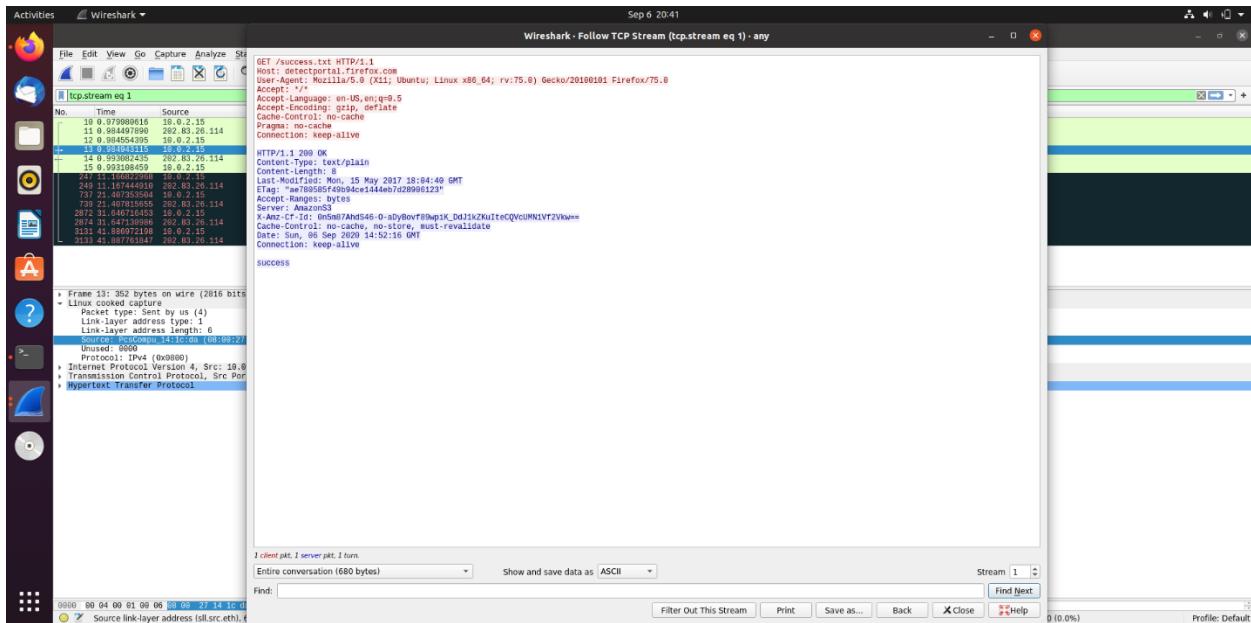
HTTP response (all the information has been expanded in the following screenshots):



Using Wireshark's Follow TCP Stream

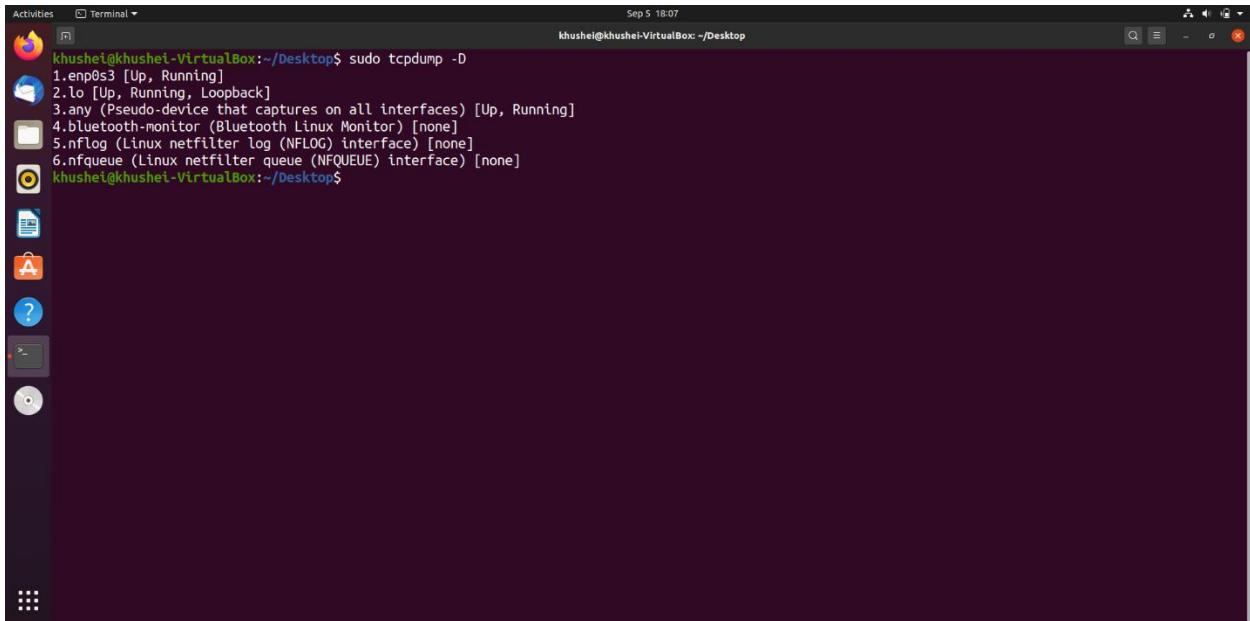
Step 1: Make sure the filter is blank. Right-click any packet inside the Packet List Pane, then select 'Follow TCP Stream'. For demo purpose, a packet containing the HTTP GET request "GET / HTTP / 1.1" can be selected.

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



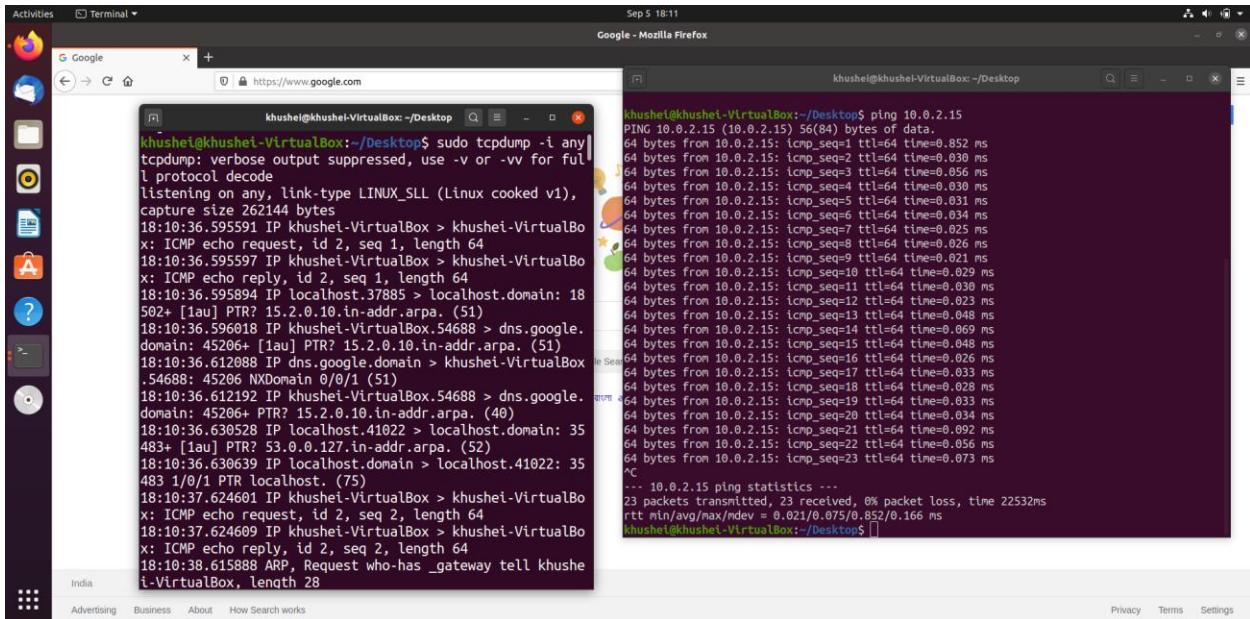
Task 4: Capturing packets with tcpdump

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



```
khushel@khushel-VirtualBox:~/Desktop$ sudo tcpdump -D
1.enp0s3 [Up, Running]
2.lo [Up, Running, Loopback]
3.any (Pseudo-device that captures on all interfaces) [Up, Running]
4.bluetooth-monitor (Bluetooth Linux Monitor) [none]
5.nflog (Linux netfilter log (NFLOG) interface) [none]
6.nfqueue (Linux netfilter queue (NFQUEUE) interface) [none]
```

Step 2: Capture all packets in any interface by running this command



```
khushel@khushel-VirtualBox:~/Desktop$ sudo tcpdump -i any
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on any, link-type LINUX_SLL (Linux cooked vi),
capture size 262144 bytes
18:10:36.595591 IP khushel-VirtualBox > khushel-VirtualBo
x: ICMP echo request, id 2, seq 1, length 64
18:10:36.595597 IP khushel-VirtualBox > khushel-VirtualBo
x: ICMP echo reply, id 2, seq 1, length 64
18:10:36.595894 IP localhost.37885 > localhost.domain: 18
502+ [iau] PTR? 15.2.0.10.in-addr.arpa. (51)
18:10:36.596018 IP khushel-VirtualBox.54688 > dns.google.
domain: 45206+ [iau] PTR? 15.2.0.10.in-addr.arpa. (51)
18:10:36.612088 IP dns.google.domain > khushel-VirtualBo
x.54688: 45206 NXDomain 0/0/1 (51)
18:10:36.612192 IP khushel-VirtualBox.54688 > dns.google.
domain: 45206+ PTR? 15.2.0.10.in-addr.arpa. (40)
18:10:36.630528 IP localhost.41022 > localhost.domain: 35
483+ [iau] PTR? 53.0.0.127.in-addr.arpa. (52)
18:10:36.630639 IP localhost.domain > localhost.41022: 35
483 1/0/1 PTR localhost. (75)
18:10:37.624601 IP khushel-VirtualBox > khushel-VirtualBo
x: ICMP echo request, id 2, seq 2, length 64
18:10:37.624609 IP khushel-VirtualBox > khushel-VirtualBo
x: ICMP echo reply, id 2, seq 2, length 64
18:10:38.615888 ARP, Request who-has _gateway tell khushe
i-VirtualBox, length 28
```

```
khushel@khushel-VirtualBox:~/Desktop$ ping 10.0.2.15
PING 10.0.2.15 (10.0.2.15) 56(84) bytes of data.
64 bytes from 10.0.2.15: icmp_seq=1 ttl=64 time=0.852 ms
64 bytes from 10.0.2.15: icmp_seq=2 ttl=64 time=0.838 ms
64 bytes from 10.0.2.15: icmp_seq=3 ttl=64 time=0.856 ms
64 bytes from 10.0.2.15: icmp_seq=4 ttl=64 time=0.838 ms
64 bytes from 10.0.2.15: icmp_seq=5 ttl=64 time=0.831 ms
64 bytes from 10.0.2.15: icmp_seq=6 ttl=64 time=0.834 ms
64 bytes from 10.0.2.15: icmp_seq=7 ttl=64 time=0.825 ms
64 bytes from 10.0.2.15: icmp_seq=8 ttl=64 time=0.826 ms
64 bytes from 10.0.2.15: icmp_seq=9 ttl=64 time=0.821 ms
64 bytes from 10.0.2.15: icmp_seq=10 ttl=64 time=0.829 ms
64 bytes from 10.0.2.15: icmp_seq=11 ttl=64 time=0.830 ms
64 bytes from 10.0.2.15: icmp_seq=12 ttl=64 time=0.823 ms
64 bytes from 10.0.2.15: icmp_seq=13 ttl=64 time=0.848 ms
64 bytes from 10.0.2.15: icmp_seq=14 ttl=64 time=0.869 ms
64 bytes from 10.0.2.15: icmp_seq=15 ttl=64 time=0.848 ms
64 bytes from 10.0.2.15: icmp_seq=16 ttl=64 time=0.826 ms
64 bytes from 10.0.2.15: icmp_seq=17 ttl=64 time=0.833 ms
64 bytes from 10.0.2.15: icmp_seq=18 ttl=64 time=0.828 ms
64 bytes from 10.0.2.15: icmp_seq=19 ttl=64 time=0.833 ms
64 bytes from 10.0.2.15: icmp_seq=20 ttl=64 time=0.834 ms
64 bytes from 10.0.2.15: icmp_seq=21 ttl=64 time=0.892 ms
64 bytes from 10.0.2.15: icmp_seq=22 ttl=64 time=0.856 ms
64 bytes from 10.0.2.15: icmp_seq=23 ttl=64 time=0.873 ms
^C
--- 10.0.2.15 ping statistics ---
23 packets transmitted, 23 received, 0% packet loss, time 22532ms
rtt min/avg/max/mdev = 0.021/0.075/0.852/0.166 ms
```

```

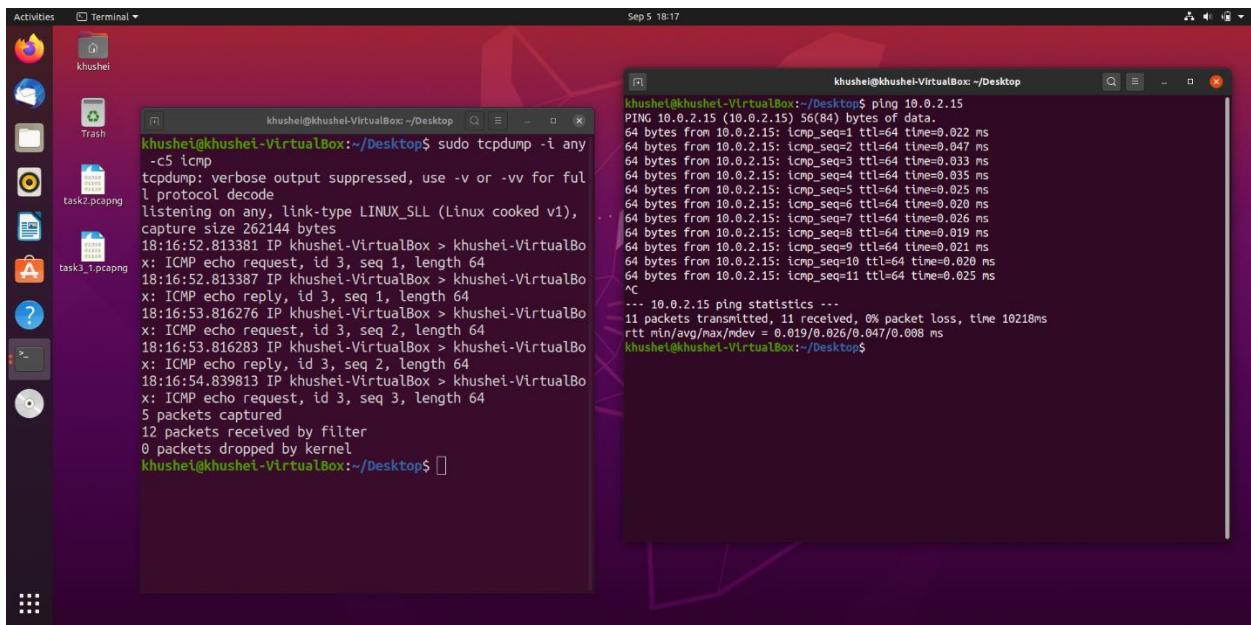
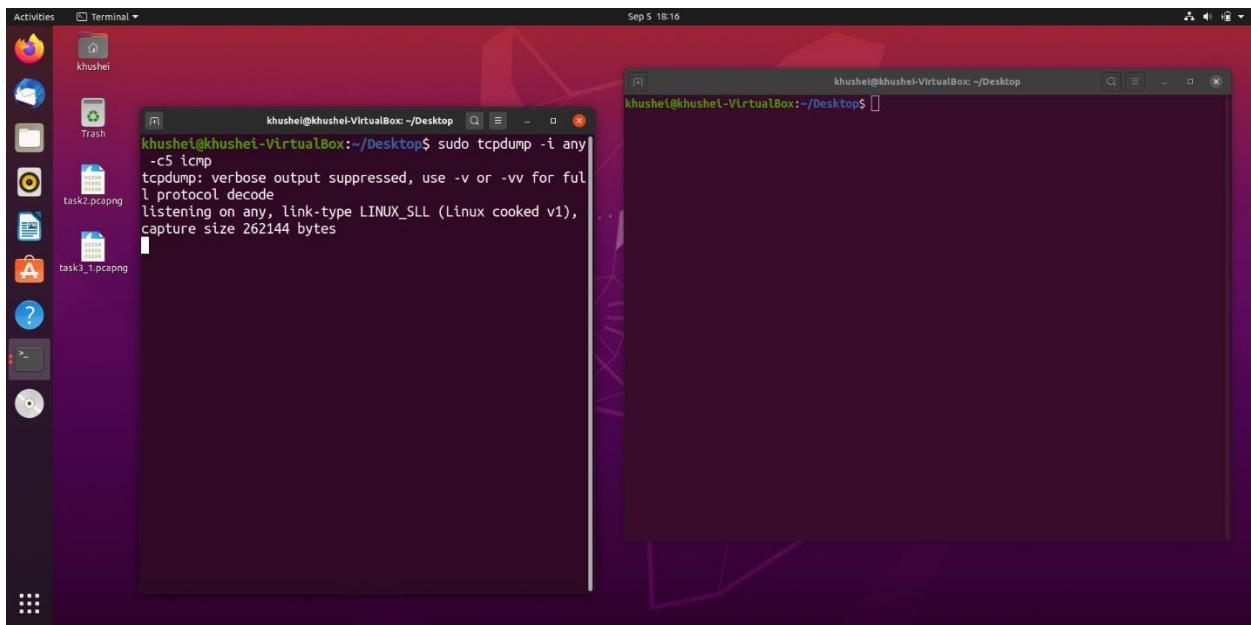
khushel@khushel-VirtualBox:~/Desktop$ ping 10.0.2.15
PING 10.0.2.15 (10.0.2.15) 56(84) bytes of data.
64 bytes From 10.0.2.15: icmp_seq=1 ttl=64 time=0.852 ms
64 bytes From 10.0.2.15: icmp_seq=2 ttl=64 time=0.838 ms
64 bytes From 10.0.2.15: icmp_seq=3 ttl=64 time=0.856 ms
64 bytes From 10.0.2.15: icmp_seq=4 ttl=64 time=0.830 ms
64 bytes From 10.0.2.15: icmp_seq=5 ttl=64 time=0.831 ms
64 bytes From 10.0.2.15: icmp_seq=6 ttl=64 time=0.834 ms
64 bytes From 10.0.2.15: icmp_seq=7 ttl=64 time=0.825 ms
64 bytes From 10.0.2.15: icmp_seq=8 ttl=64 time=0.826 ms
64 bytes From 10.0.2.15: icmp_seq=9 ttl=64 time=0.821 ms
64 bytes From 10.0.2.15: icmp_seq=10 ttl=64 time=0.829 ms
64 bytes From 10.0.2.15: icmp_seq=11 ttl=64 time=0.838 ms
64 bytes From 10.0.2.15: icmp_seq=12 ttl=64 time=0.823 ms
64 bytes From 10.0.2.15: icmp_seq=13 ttl=64 time=0.848 ms
64 bytes From 10.0.2.15: icmp_seq=14 ttl=64 time=0.869 ms
64 bytes From 10.0.2.15: icmp_seq=15 ttl=64 time=0.848 ms
64 bytes From 10.0.2.15: icmp_seq=16 ttl=64 time=0.826 ms
64 bytes From 10.0.2.15: icmp_seq=17 ttl=64 time=0.833 ms
64 bytes From 10.0.2.15: icmp_seq=18 ttl=64 time=0.828 ms
64 bytes From 10.0.2.15: icmp_seq=19 ttl=64 time=0.833 ms
64 bytes From 10.0.2.15: icmp_seq=20 ttl=64 time=0.834 ms
64 bytes From 10.0.2.15: icmp_seq=21 ttl=64 time=0.892 ms
64 bytes From 10.0.2.15: icmp_seq=22 ttl=64 time=0.856 ms
64 bytes From 10.0.2.15: icmp_seq=23 ttl=64 time=0.873 ms
^C
--- 10.0.2.15 ping statistics ---
23 packets transmitted, 23 received, 0% packet loss, time 22532ms
rtt min/avg/max/mdev = 0.821/0.875/0.852/0.166 ms
khushel@khushel-VirtualBox:~/Desktop$ 

```

Step 3: Understand the output format.

Each line begins with a timestamp. That is followed by the network layer protocol IP. Then we get the source IP and port, a ‘>’ symbol and then the destination IP and port. After this we get a flag indicating the type of interchange. Then there is an acknowledgement number representing the byte to be sent or received. Following this is win and a number that represents the size available in the receiving buffer. After this finally is the packet length.

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:



Step 5: Check the packet content. For example, inspect the HTTP content of a web request:

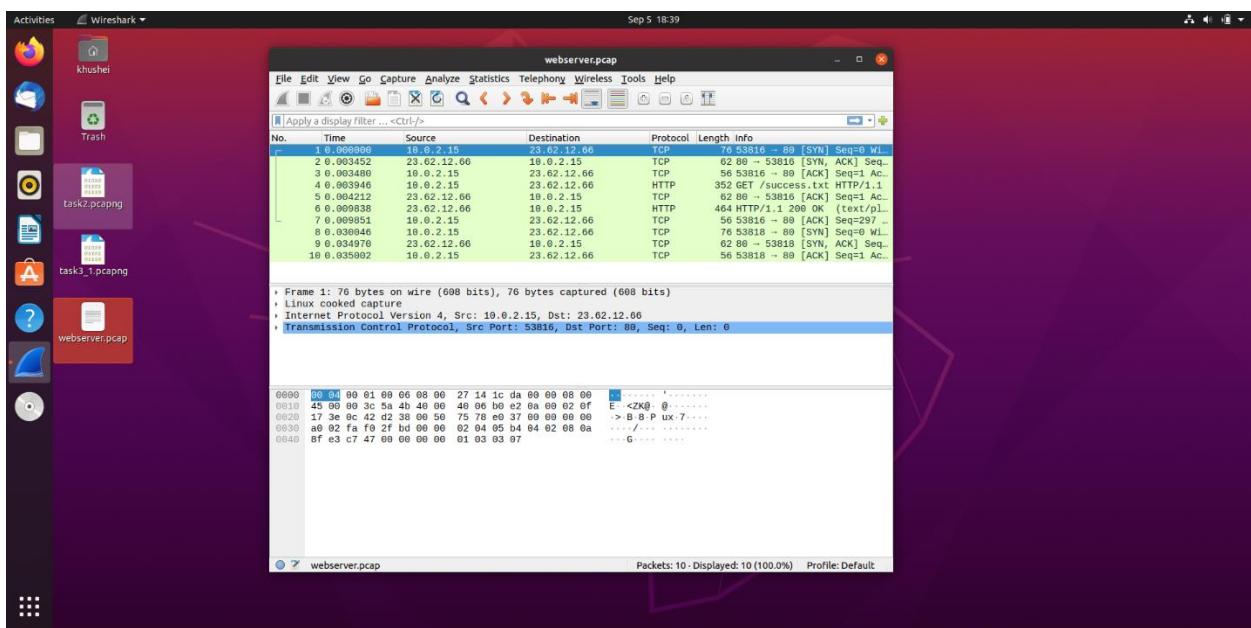
```
Activities Terminal Sep 5 18:25 khushel@khushel-VirtualBox: ~/Desktop$ sudo tcpdump -i any -c10 -nn -A port 80
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on any, link-type LINUX_SLL (Linux cooked v1), capture size 262144 bytes
18:24:29.847935 IP 10.0.2.15.43792 > 23.62.12.88.80: Flags [.], ack 514048410, win 63832, length 0
E..(q.@.0...
....>X...P.y].....P..X/...
18:24:29.847966 IP 10.0.2.15.43790 > 23.62.12.88.80: Flags [.], ack 513984410, win 63832, length 0
E..(.@.0.0...
....>X...P.....P..X/...
18:24:29.848424 IP 23.62.12.88.80 > 10.0.2.15.43792: Flags [.], ack 1, win 65535, length 0
E..(.r.@....>X
....P.....y].P....K.....
18:24:29.848436 IP 23.62.12.88.80 > 10.0.2.15.43790: Flags [.], ack 1, win 65535, length 0
E..(.s..@....>X
....P.....P.....
18:24:32.663702 IP 10.0.2.15.34170 > 117.18.237.29.80: Flags [.], ack 514625600, win 63920, length 0
E..(.t@.0.%p
....U....z.P3.....@P...nY...
18:24:32.664166 IP 117.18.237.29.80 > 10.0.2.15.34170: Flags [.], ack 1, win 65535, length 0
E..(.t@.h.u...
....P.z....@3....P....;
18:24:32.664166 IP 10.0.2.15.42810 > 216.58.200.131.80: Flags [.], ack 516419509, win 63791, length 0
E..(.t@.0.$
....P.....:I.....P...Q.....
18:24:36.311886 IP 216.58.200.131.80 > 10.0.2.15.42810: Flags [.], ack 1, win 65535, length 0
E..(.t@.B:...
....P.....~I.P....Q.....
18:24:40.087635 IP 10.0.2.15.43792 > 23.62.12.88.80: Flags [.], ack 1, win 63832, length 0
E..(q.@.0...
....>X...P.y].....P..X/...
18:24:40.087664 IP 10.0.2.15.43790 > 23.62.12.88.80: Flags [.], ack 1, win 63832, length 0
E..(.t@.0.Q...
....>X...P.....P..X/...
10 packets captured
```

```
Activities Terminal Sep 5 18:25 khushel@khushel-VirtualBox: ~/Desktop$ 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
18:24:29.847935 IP 10.0.2.15.43792 > 23.62.12.88.80: Flags [.], ack 514048410, win 63832, length 0
E..(q.@.0...
....>X...P.y].....P..X/...
18:24:29.847966 IP 10.0.2.15.43790 > 23.62.12.88.80: Flags [.], ack 513984410, win 63832, length 0
E..(.@.0.0...
....>X...P.....P..X/...
18:24:29.848424 IP 23.62.12.88.80 > 10.0.2.15.43792: Flags [.], ack 1, win 65535, length 0
E..(.r.@....>X
....P.....y].P....K.....
18:24:29.848436 IP 23.62.12.88.80 > 10.0.2.15.43790: Flags [.], ack 1, win 65535, length 0
E..(.s..@....>X
....P.....P.....
18:24:32.663702 IP 10.0.2.15.34170 > 117.18.237.29.80: Flags [.], ack 514625600, win 63920, length 0
E..(.t@.0.%p
....U....z.P3.....@P...nY...
18:24:32.664166 IP 117.18.237.29.80 > 10.0.2.15.34170: Flags [.], ack 1, win 65535, length 0
E..(.t@.h.u...
....P.z....@3....P....;
18:24:32.664166 IP 10.0.2.15.42810 > 216.58.200.131.80: Flags [.], ack 516419509, win 63791, length 0
E..(.t@.0.$
....P.....:I.....P...Q.....
18:24:36.311886 IP 216.58.200.131.80 > 10.0.2.15.42810: Flags [.], ack 1, win 65535, length 0
E..(.t@.B:...
....P.....~I.P....Q.....
18:24:40.087635 IP 10.0.2.15.43792 > 23.62.12.88.80: Flags [.], ack 1, win 63832, length 0
E..(q.@.0...
....>X...P.y].....P..X/...
18:24:40.087664 IP 10.0.2.15.43790 > 23.62.12.88.80: Flags [.], ack 1, win 63832, length 0
E..(.t@.0.Q...
....>X...P.....P..X/...
10 packets captured
10 packets received by filter
0 packets dropped by kernel
khushel@khushel-VirtualBox: ~/Desktop$
```

Step 6: To save packets to a file instead of displaying them on screen, use the option **-w**:

```
Activities Terminal Sep 5 18:28
khushel@khushel-VirtualBox:~/Desktop$ 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
10 packets captured
14 packets received by filter
0 packets dropped by kernel
khushel@khushel-VirtualBox:~/Desktop$
```

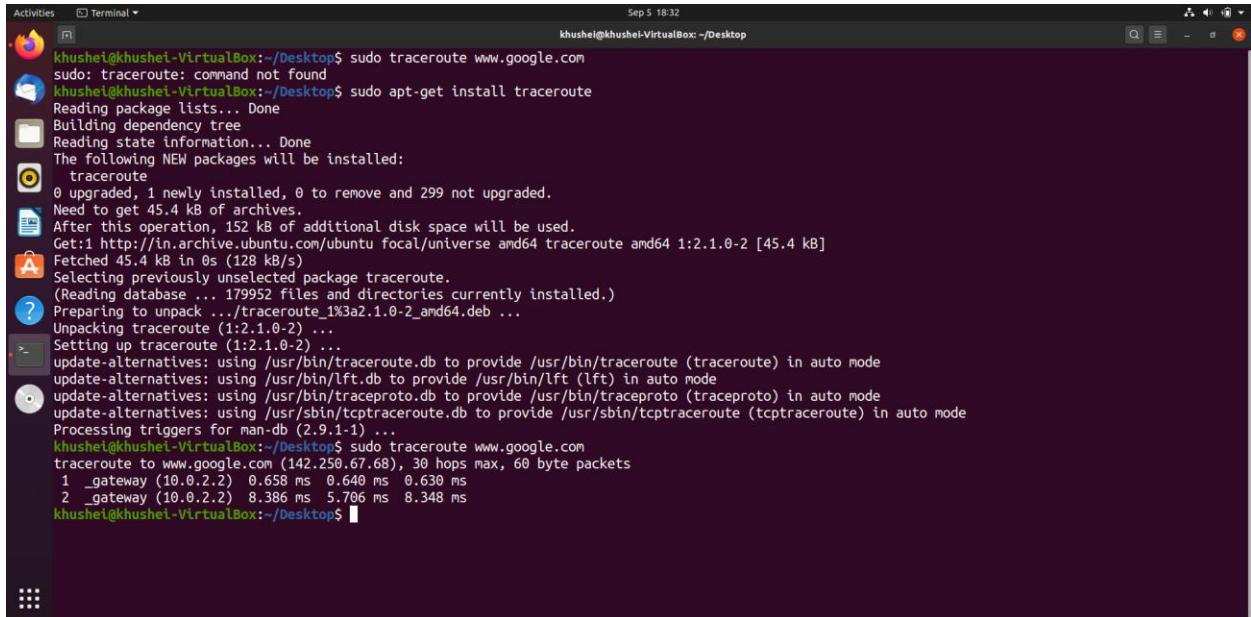
The wireshark file-



Task 5: Perform Traceroute checks

Step 1: Run the traceroute using the following command.

Traceroute was not installed in my Ubuntu VM, so I installed it with sudo apt-get install traceroute and then executed the given command.



The screenshot shows a terminal window titled "Terminal" with the command history and output. The user runs "sudo traceroute www.google.com" which fails because traceroute is not installed. The user then runs "sudo apt-get install traceroute" and installs it. After installation, the traceroute command is run again, successfully showing the traceroute to www.google.com with two hops.

```
khushel@khushei-VirtualBox:~/Desktop$ sudo traceroute www.google.com
sudo: traceroute: command not found
khushel@khushei-VirtualBox:~/Desktop$ sudo apt-get install traceroute
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  traceroute
0 upgraded, 1 newly installed, 0 to remove and 299 not upgraded.
Need to get 45.4 kB of archives.
After this operation, 152 kB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu focal/universe amd64 traceroute 1:2.1.0-2 [45.4 kB]
Fetched 45.4 kB in 0s (128 kB/s)
Selecting previously unselected package traceroute.
(Reading database ... 179952 files and directories currently installed.)
Preparing to unpack .../traceroute_1%3a2.1.0-2_amd64.deb ...
Unpacking traceroute (1:2.1.0-2) ...
Setting up traceroute (1:2.1.0-2) ...
update-alternatives: using /usr/bin/traceroute.db to provide /usr/bin/traceroute (traceroute) in auto mode
update-alternatives: using /usr/bin/lft.db to provide /usr/bin/lft (lft) in auto mode
update-alternatives: using /usr/bin/traceproto.db to provide /usr/bin/traceproto (traceproto) in auto mode
update-alternatives: using /usr/sbin/tcptraceroute.db to provide /usr/sbin/tcptraceroute (tcptraceroute) in auto mode
Processing triggers for man-db (2.9.1-1) ...
khushel@khushei-VirtualBox:~/Desktop$ sudo traceroute www.google.com
traceroute to www.google.com (142.250.67.68), 30 hops max, 60 byte packets
  1 _gateway (10.0.2.2)  0.658 ms  0.640 ms  0.630 ms
  2 _gateway (10.0.2.2)  8.386 ms  5.706 ms  8.348 ms
khushel@khushei-VirtualBox:~/Desktop$
```

Step 2: Analyze destination address of google.com and no. of hops

Observation: Destination IP address is 142.250.67.68 and number of hops are 2

Step 3: To speed up the process, you can disable the mapping of IP addresses with hostnames by using the *-n* option

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

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.

(Step 3,4,5 are all shown in the below screenshot)

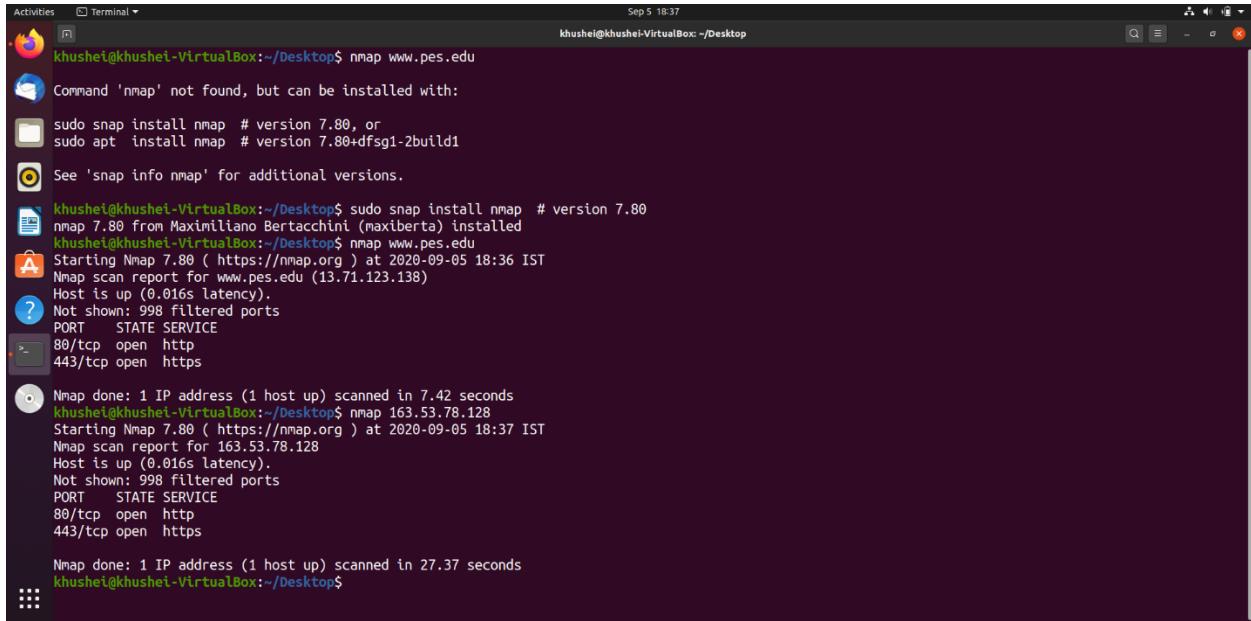
Activities Terminal Sep 5 18:35 khushei@khushel-VirtualBox: ~/Desktop

```
khushei@khushel-VirtualBox:~/Desktop$ sudo traceroute -n www.google.com
traceroute to www.google.com (172.217.163.164), 30 hops max, 60 byte packets
 1  10.0.2.2  0.308 ms  0.289 ms  0.279 ms
 2  10.0.2.2  2.590 ms  3.250 ms  5.911 ms
khushei@khushel-VirtualBox:~/Desktop$ sudo traceroute -I www.google.com
traceroute to www.google.com (172.217.163.164), 30 hops max, 60 byte packets
 1  _gateway (10.0.2.2)  0.246 ms  0.236 ms  0.234 ms
 2  192.168.0.1 (192.168.0.1)  4.234 ms  2.941 ms  4.067 ms
 3  10.232.0.1 (10.232.0.1)  4.932 ms  4.776 ms  5.077 ms
 4  broadband.actcorp.in (202.83.20.43)  19.295 ms  19.066 ms  19.303 ms
 5  14.141.145.5.static-Bangalore.vsnl.net.in (14.141.145.5)  5.936 ms  8.504 ms  6.733 ms
 6  172.31.167.58 (172.31.167.58)  14.681 ms  32.708 ms  32.945 ms
 7  14.140.100.6.static-vsnl.net.in (14.140.100.6)  10.379 ms  10.176 ms  10.474 ms
 8  115.112.71.65.STDILL-Chennai.vsnl.net.in (115.112.71.65)  11.595 ms  12.173 ms  16.283 ms
 9  121.240.1.50 (121.240.1.50)  11.037 ms  11.695 ms  11.994 ms
10  74.125.242.129 (74.125.242.129)  32.398 ms  33.072 ms  33.072 ms
11  209.85.248.181 (209.85.248.181)  15.946 ms  15.463 ms  10.777 ms
12  maa05s05-in-f4.1e100.net (172.217.163.164)  10.240 ms  9.416 ms  14.032 ms
khushei@khushel-VirtualBox:~/Desktop$ sudo traceroute -T www.google.com
traceroute to www.google.com (172.217.163.164), 30 hops max, 60 byte packets
 1  _gateway (10.0.2.2)  0.354 ms  0.335 ms  0.328 ms
 2  maa05s05-in-f4.1e100.net (172.217.163.164)  11.422 ms  12.256 ms  12.239 ms
khushei@khushel-VirtualBox:~/Desktop$
```

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.

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



The screenshot shows a terminal window in a Linux desktop environment. The title bar indicates it's a terminal window for the user 'khushet' on a 'VirtualBox' host. The date and time 'Sep 5 18:37' are also shown. The terminal output is as follows:

```
khushet@khushet-VirtualBox:~/Desktop$ nmap www.pes.edu
khushet@khushet-VirtualBox:~/Desktop$ Command 'nmap' not found, but can be installed with:
khushet@khushet-VirtualBox:~/Desktop$ sudo snap install nmap # version 7.80, or
khushet@khushet-VirtualBox:~/Desktop$ sudo apt install nmap # version 7.80+dfsg1-2build1
khushet@khushet-VirtualBox:~/Desktop$ See 'snap info nmap' for additional versions.

khushet@khushet-VirtualBox:~/Desktop$ sudo snap install nmap # version 7.80
nmap 7.80 from Maximiliano Bertacchini (maxiberta) installed
khushet@khushet-VirtualBox:~/Desktop$ nmap www.pes.edu
Starting Nmap 7.80 ( https://nmap.org ) at 2020-09-05 18:36 IST
Nmap scan report for www.pes.edu (13.71.123.138)
Host is up (0.016s latency).
Not shown: 998 filtered ports
PORT      STATE SERVICE
80/tcp    open  http
443/tcp   open  https

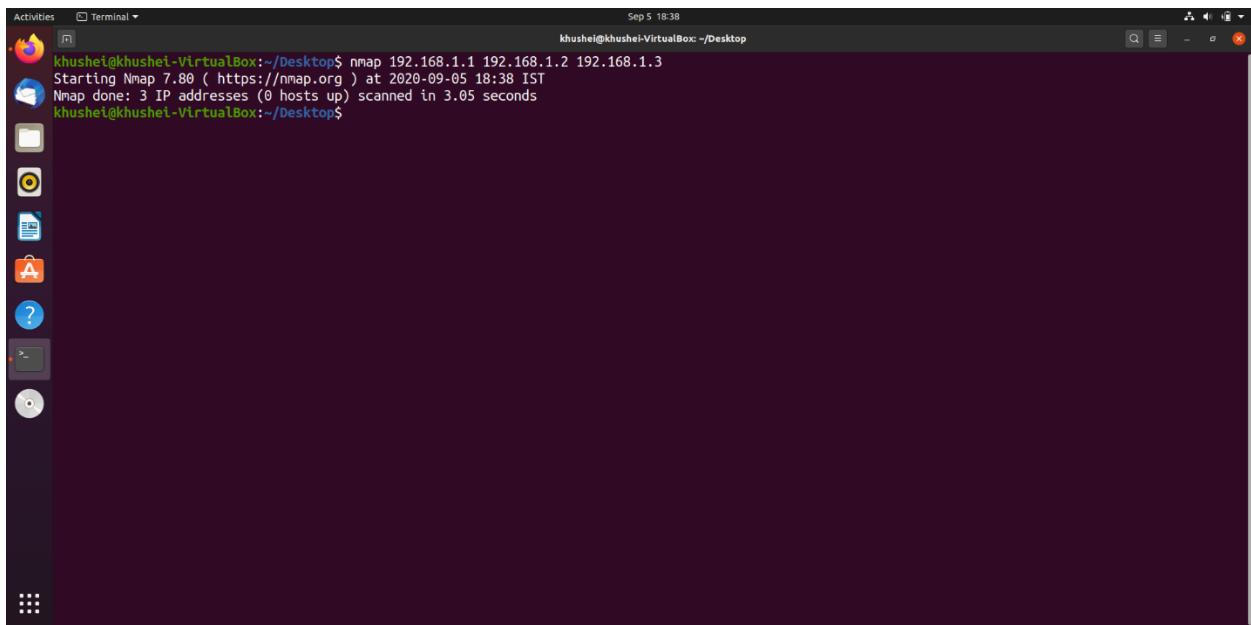
Nmap done: 1 IP address (1 host up) scanned in 7.42 seconds
khushet@khushet-VirtualBox:~/Desktop$ nmap 163.53.78.128
Starting Nmap 7.80 ( https://nmap.org ) at 2020-09-05 18:37 IST
Nmap scan report for 163.53.78.128
Host is up (0.016s latency).
Not shown: 998 filtered ports
PORT      STATE SERVICE
80/tcp    open  http
443/tcp   open  https

Nmap done: 1 IP address (1 host up) scanned in 27.37 seconds
khushet@khushet-VirtualBox:~/Desktop$
```

Step 3: Scan multiple IP address or subnet (IPv4)

The output shows 0 hosts up because neither of these three IP addresses are up and running in our network.

Activities Terminal Sep 5 18:38
khushel@khushei-VirtualBox:~/Desktop\$ nmap 192.168.1.1 192.168.1.2 192.168.1.3
Starting Nmap 7.80 (https://nmap.org) at 2020-09-05 18:38 IST
Nmap done: 3 IP addresses (0 hosts up) scanned in 3.05 seconds
khushel@khushei-VirtualBox:~/Desktop\$

A screenshot of an Ubuntu desktop environment. On the left is a dark vertical dock containing icons for Dash, Home, Applications, Files, Activities, Help, and a terminal. The main window is a terminal window titled 'Terminal'. The title bar shows the date and time: 'Sep 5 18:38'. The terminal window contains a command-line session. The user has run the 'nmap' command with three IP addresses as arguments: 192.168.1.1, 192.168.1.2, and 192.168.1.3. The output indicates that the scan started at 2020-09-05 18:38 IST and completed in 3.05 seconds. It found 3 IP addresses but none were up. The terminal window has a standard black background with white text and a light gray cursor.

Questions on above observations:

1) Is your browser running HTTP version 1.0 or 1.1? What version of HTTP is the server?

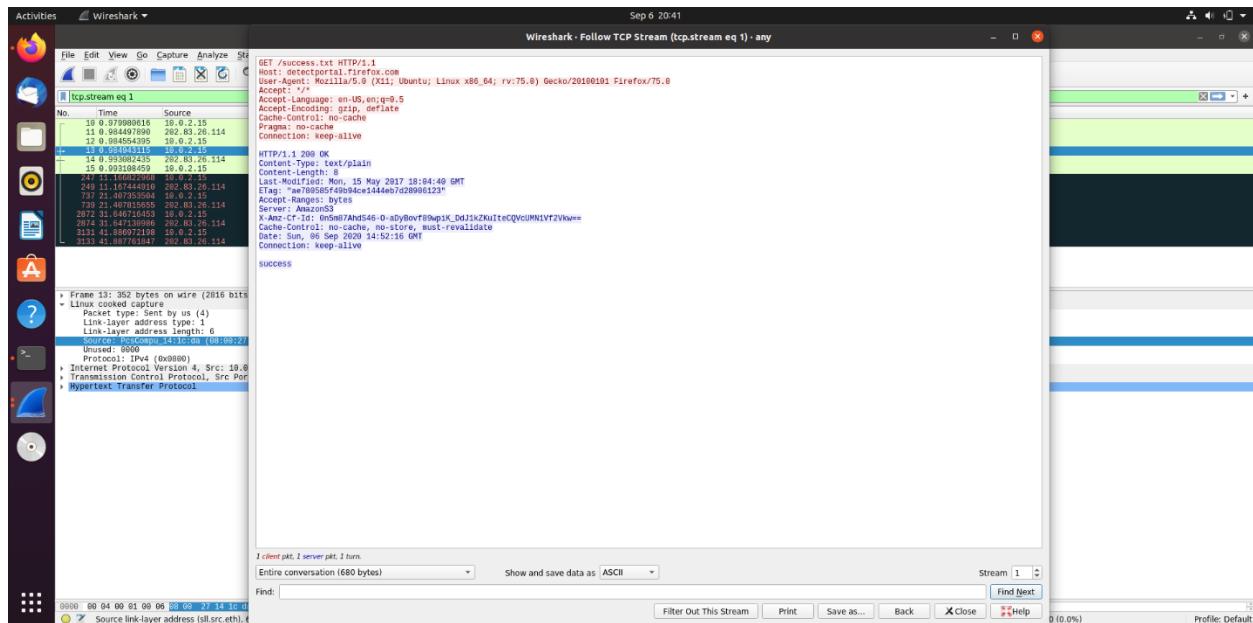
Browser: HTTP version 1.1

Server: HTTP version 1.1

2) When was the HTML file that you are retrieving last modified at the server?

In task 3, we had followed TCP stream for the packet containing "GET / HTTP / 1.1".

In that screenshot, we can see the Last-Modified field. It is Mon, 15 May 2017 18:04:40 GMT in my case.



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

By using the -c flag with ping command.

ping -c <number_of_packets> <hostname>

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

This can be done using nmap:

- 1) nmap -O -v localhost
- 2) nmap -O -v server.ip.address