Week #1
Computer Networks Lab
Ruchira R Vadiraj
PES2201800602

Date : 31-08-2020

Objective: Study and understand the basic networking tools - Wireshark, Tcpdump, Ping, Traceroute and Netcat.

Learn and Understand Network Tools

1. Wireshark

- Perform and analyze Ping PDU capture
- Examine HTTP packet capture
- Analyze HTTP packet capture using filter

2. Netcat

- Establish communication between client and server
- Transfer files

3. Tcpdump

- Capture packets

4. Ping

- Test the connectivity between 2 systems

5. Traceroute

- Perform traceroute checks

6. Nmap

- Explore an entire network

IMPORTANT INSTRUCTIONS:

- Perform **sudo apt-get update** before installing any tool or utility.
- Install any tool or utility using the command **sudo apt-get install name_of_the_tool**
- Take screenshots whenever necessary (paste in a .doc / .docx) and upload in Edmodo
- Write down the observations in your observation notebook.
- Instructors will give information, to define an IP address for your machine (e.g., Section – 'a' & Serial number is 1, then your IP address should be 10.0.1.1. Section – 'h' & & Serial number is 23, then your IP address should be 10.0.8.23)

Task 1: Linux Interface Configuration (ifconfig / IP command)

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

ifconfig (or) **ip addr show**

Analyze and fill the following table:

ip address table:

Interface name	IP address (IPv4 / IPv6)	MAC address	
Docker0	172.17.0.1	02:42:04:52:85:9d	
enp0s3	10.0.2.15	08:00:27:51:ee:52	
lo	127.0.0.1	-	

```
ruchira@ruchira-VirtualBox:~$ ifconfig
docker0   Link encap:Ethernet  HWaddr 02:42:04:52:85:9d
          inet addr:172.17.0.1  Bcast:172.17.255.255  Mask:255.255.0.0
          inet6 addr: fe80::42:4ff:fe52:859d/64 Scope:Link
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:90 (90.0 B)

enp0s3    Link encap:Ethernet  HWaddr 08:00:27:51:ee:52
          inet addr:10.0.2.15  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::74d1:663c:d238:e8da/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:237451 errors:0 dropped:0 overruns:0 frame:0
          TX packets:25338 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:319585385 (319.5 MB)  TX bytes:2036085 (2.0 MB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:852 errors:0 dropped:0 overruns:0 frame:0
          TX packets:852 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:126023 (126.0 KB)  TX bytes:126023 (126.0 KB)
```

The hardware address and the IP address is mentioned, when ifconfig is typed in the terminal.

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

sudo ifconfig interface_name 10.0.your_section.your_sno netmask 255.255.255.0 (or)

sudo ip addr add 10.0.your_section.your_sno /24 dev interface_name

```
ruchira@ruchira-VirtualBox:~$ sudo ifconfig enp0s3 10.0.4.47 netmask 255.255.255.0
```

10.0.4.47 is assigned as the IP address to the interface.

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

sudo ifconfig interface_name down

sudo ifconfig interface_name up

```
ruchira@ruchira-VirtualBox:~$ sudo ifconfig enp0s3 up
```

The configured interface is set to up and

running if it isn't.

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

ip neigh

```
ruchira@ruchira-VirtualBox:~$ ip neigh
10.0.2.2 dev enp0s3 lladdr 52:54:00:12:35:02 STALE
```

The neighbor table is shown in the output.

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

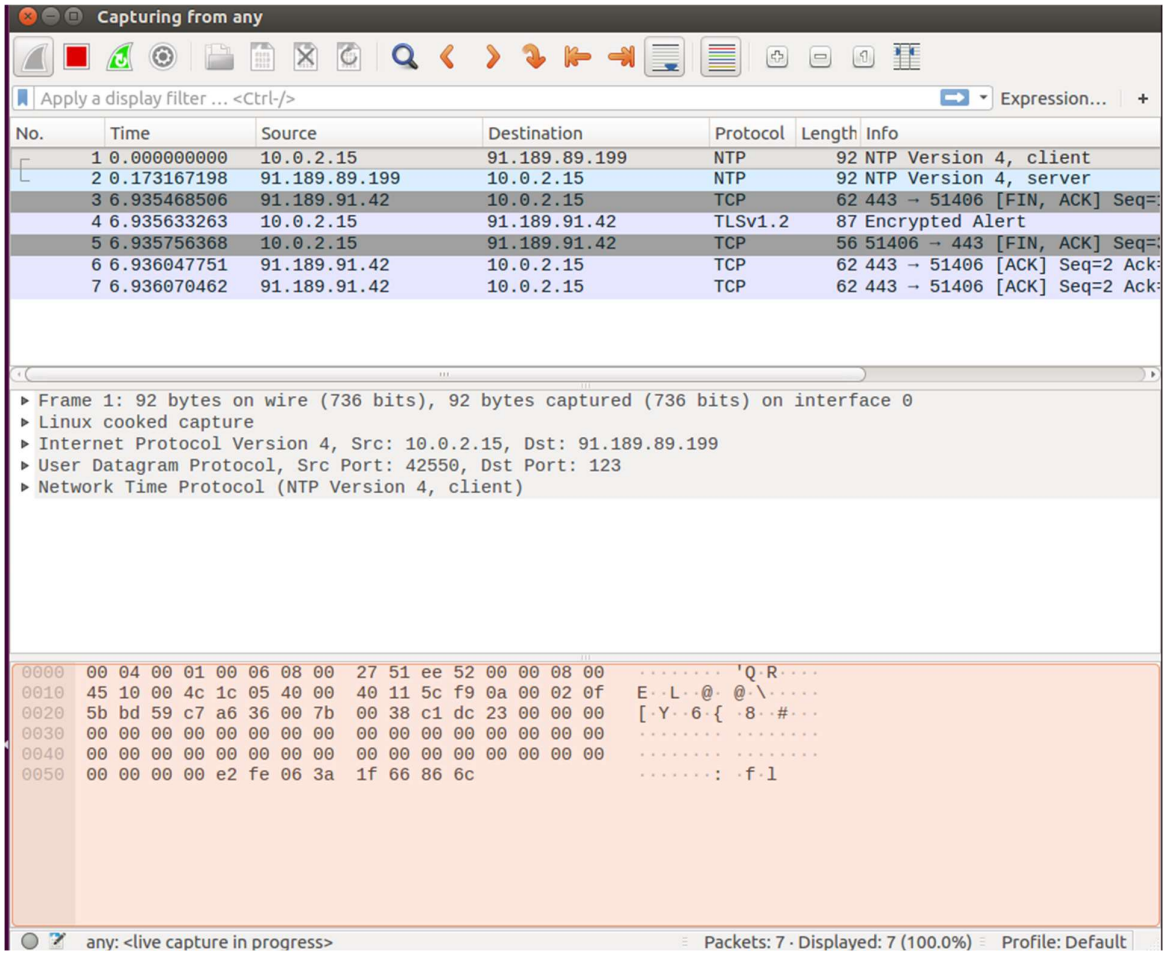
Step 1: Assign an IP address to the system (Host).

Note: IP address of your system should be 10.0.your_section.your_sno.

```
ruchira@ruchira-VirtualBox:~$ sudo ip addr add 10.0.4.47/27 dev enp0s3
```

The IP address is set to 10.0.4.47.

Step 2: Launch Wireshark and select ‘any’ interface



Wireshark on launch and opened into “any”.

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

```
ruchira@ruchira-VirtualBox:~$ ping 10.0.4.47
PING 10.0.4.47 (10.0.4.47) 56(84) bytes of data.
64 bytes from 10.0.4.47: icmp_seq=1 ttl=64 time=0.035 ms
64 bytes from 10.0.4.47: icmp_seq=2 ttl=64 time=0.036 ms
64 bytes from 10.0.4.47: icmp_seq=3 ttl=64 time=0.037 ms
64 bytes from 10.0.4.47: icmp_seq=4 ttl=64 time=0.036 ms
64 bytes from 10.0.4.47: icmp_seq=5 ttl=64 time=0.036 ms
64 bytes from 10.0.4.47: icmp_seq=6 ttl=64 time=0.036 ms
64 bytes from 10.0.4.47: icmp_seq=7 ttl=64 time=0.138 ms
64 bytes from 10.0.4.47: icmp_seq=8 ttl=64 time=0.030 ms
64 bytes from 10.0.4.47: icmp_seq=9 ttl=64 time=0.035 ms
64 bytes from 10.0.4.47: icmp_seq=10 ttl=64 time=0.030 ms
64 bytes from 10.0.4.47: icmp_seq=11 ttl=64 time=0.046 ms
64 bytes from 10.0.4.47: icmp_seq=12 ttl=64 time=0.038 ms
64 bytes from 10.0.4.47: icmp_seq=13 ttl=64 time=0.076 ms
64 bytes from 10.0.4.47: icmp_seq=14 ttl=64 time=0.046 ms
64 bytes from 10.0.4.47: icmp_seq=15 ttl=64 time=0.029 ms
64 bytes from 10.0.4.47: icmp_seq=16 ttl=64 time=0.030 ms
64 bytes from 10.0.4.47: icmp_seq=17 ttl=64 time=0.032 ms
64 bytes from 10.0.4.47: icmp_seq=18 ttl=64 time=0.036 ms
64 bytes from 10.0.4.47: icmp_seq=19 ttl=64 time=0.047 ms
64 bytes from 10.0.4.47: icmp_seq=20 ttl=64 time=0.035 ms
64 bytes from 10.0.4.47: icmp_seq=21 ttl=64 time=0.032 ms
64 bytes from 10.0.4.47: icmp_seq=22 ttl=64 time=0.033 ms
64 bytes from 10.0.4.47: icmp_seq=23 ttl=64 time=0.035 ms
64 bytes from 10.0.4.47: icmp_seq=24 ttl=64 time=0.028 ms
64 bytes from 10.0.4.47: icmp_seq=25 ttl=64 time=0.035 ms
64 bytes from 10.0.4.47: icmp_seq=26 ttl=64 time=0.029 ms
64 bytes from 10.0.4.47: icmp_seq=27 ttl=64 time=0.038 ms
64 bytes from 10.0.4.47: icmp_seq=28 ttl=64 time=0.037 ms
64 bytes from 10.0.4.47: icmp_seq=29 ttl=64 time=0.043 ms
64 bytes from 10.0.4.47: icmp_seq=30 ttl=64 time=0.032 ms
64 bytes from 10.0.4.47: icmp_seq=31 ttl=64 time=0.037 ms
^C
--- 10.0.4.47 ping statistics ---
31 packets transmitted, 31 received, 0% packet loss, time 30721ms
rtt min/avg/max/mdev = 0.028/0.040/0.138/0.020 ms
```

Observations to be made

Step 4: Analyze the following in Terminal

- TTL
- Protocol used by ping
- Time

The TTLis 64.

The protocol used by ping is ICMP.

The time taken is 0.037ms on average.

Step 5: Analyze the following in Wireshark

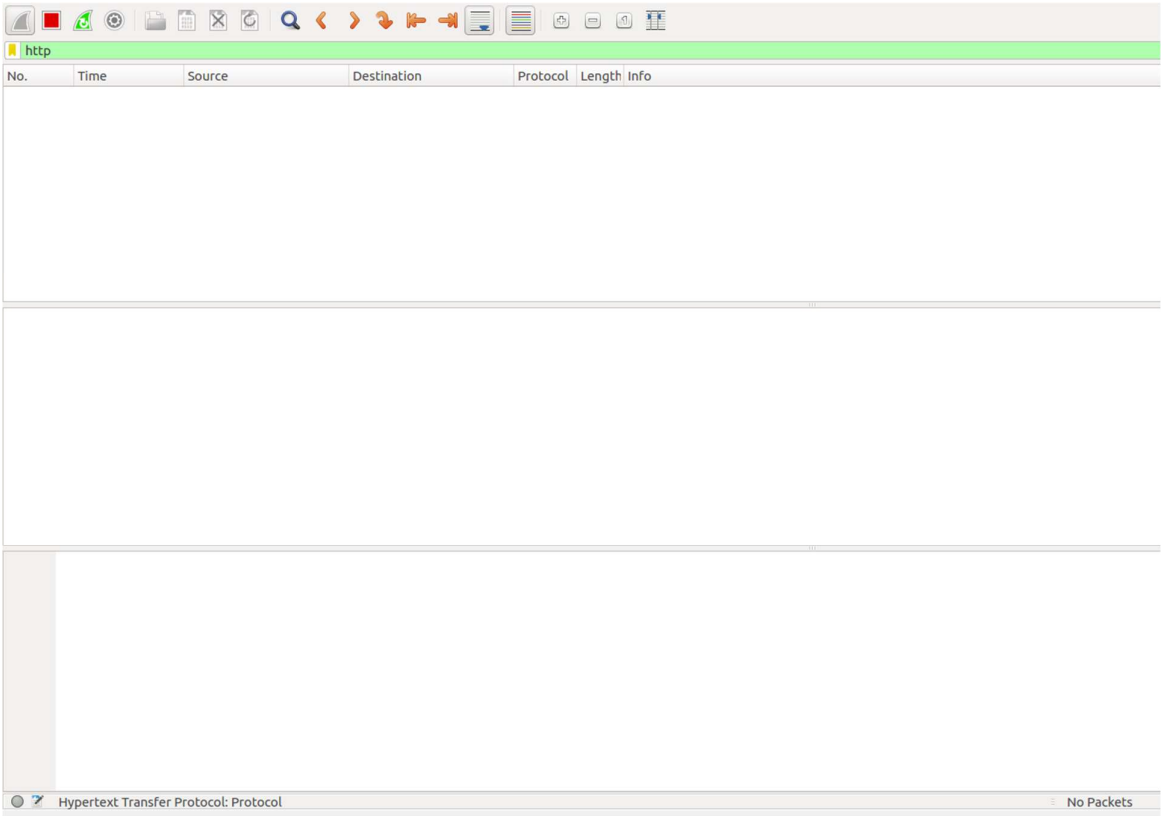
On Packet List Pane, select the first echo packet on the list. On Packet Details Pane, click on each of the four "+" to expand the information. Analyze the frames with the first echo request and echo reply and complete the table below.

Details	First Echo Request	First Echo Reply
Frame Number	1	2
Source IP address	10.0.4.47	10.0.4.47
Destination IP address	10.0.4.47	10.0.4.47
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 (reply in 2)	64 (reply in 1)

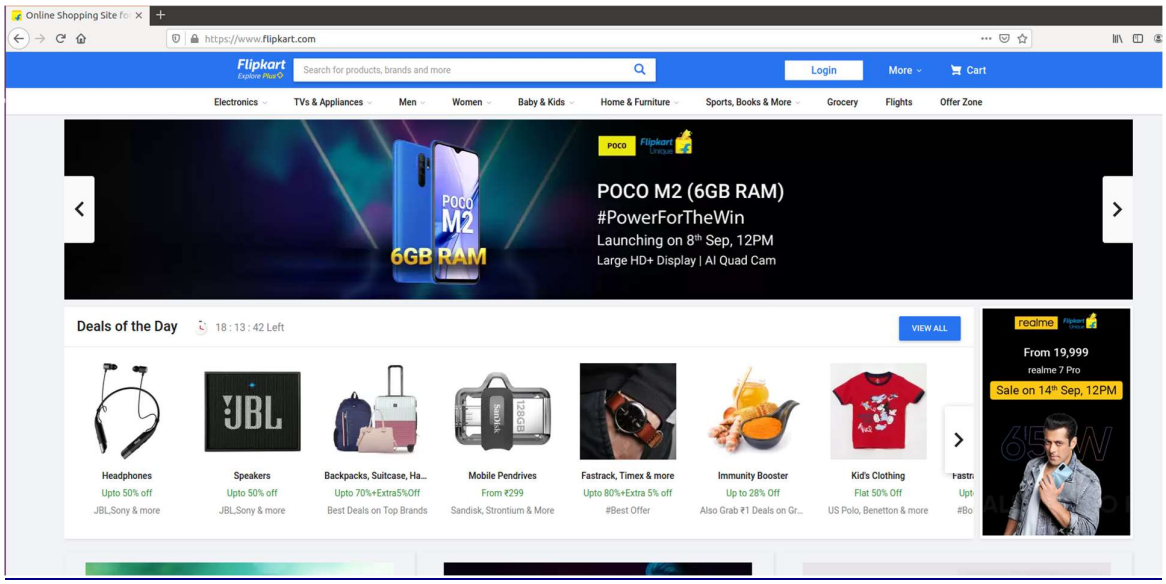
Task 3: HTTP PDU Capture

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	166	180

Source Port	58236	80
Destination Port	80	58236
Source IP address	10.0.4.47	216.58.196.163
Destination IP address	216.58.196.163	10.0.4.47
Source Ethernet Address	08:00:27:51:ee:52	52:54:00:12:35:02
Destination Ethernet Address	52:54:00:12:35:02	08:00:27:51:ee:52

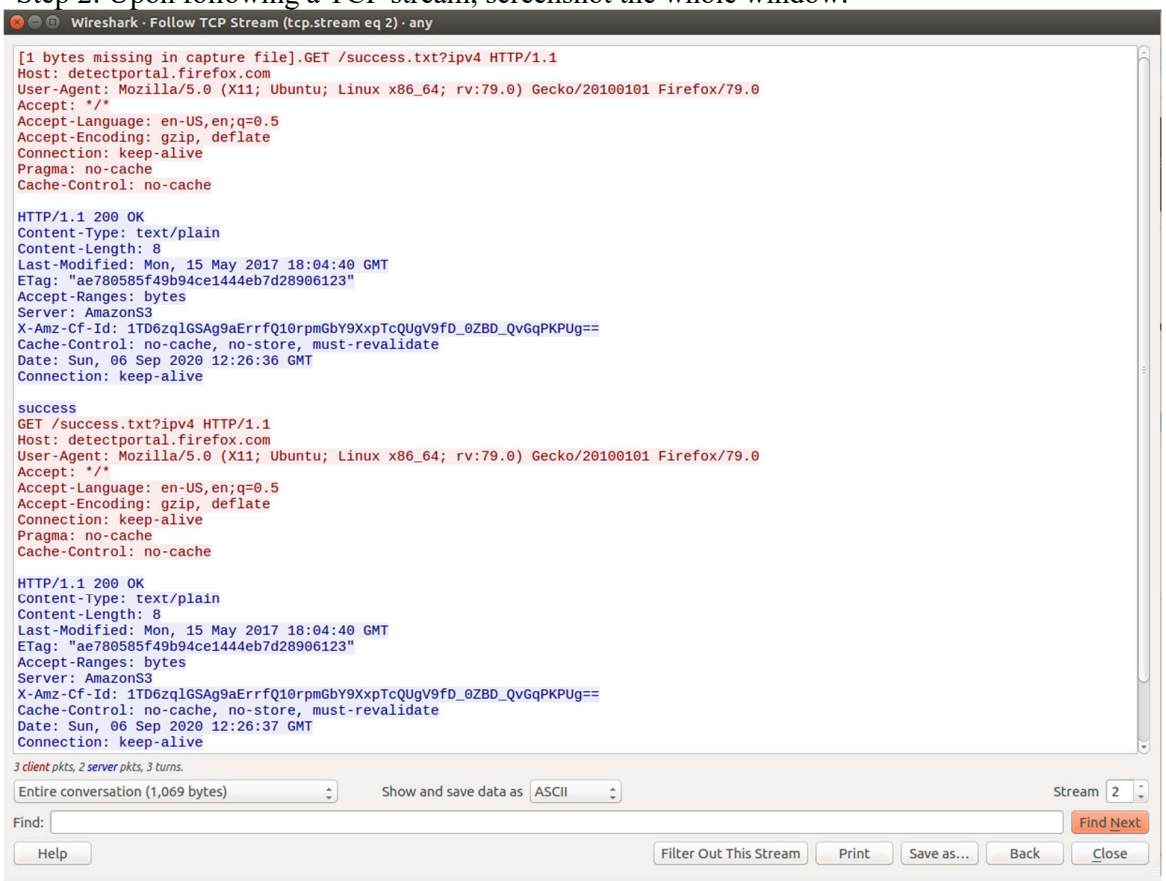
Step 4: Analyze the HTTP request and response and complete the table below.

HTTP Request		**HTTP Response**	
Get	/success.txt HTTP/1.1	Server	AmazonS3
Host	detectportal.firefox.com	Content-Type	text/plain
User-Agent	Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:79.0) Gecko 20100101 Firefox/79.0	Date	Sat, 05 Sep 2020 14:25:41 GMT
Accept-Language	en-US, en; q=0.5	Location	<NOT SPECIFIED>
Accept-Encoding	gzip, deflate	Content-Length	8
Connection	keep-alive	Connection	keep-alive

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.

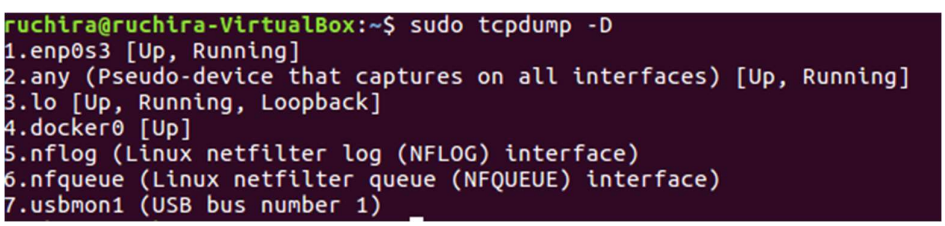
Step 2: Upon following a TCP stream, screenshot the whole window.



Task 4: Capturing packets with tcpdump

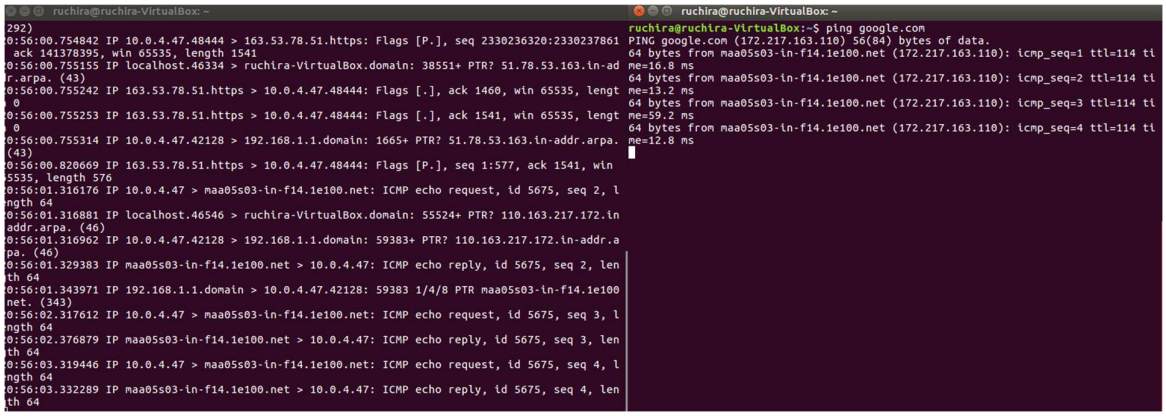
Step 1: Use the command **tcpdump -D** to see which interfaces are available for capture.

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.

The above command is used to capture all the packets from all the interfaces.

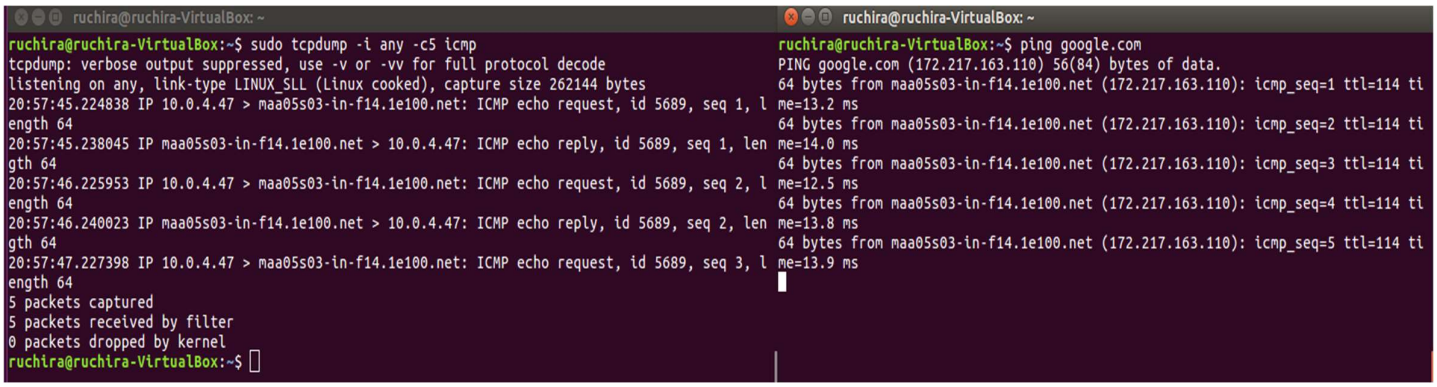
ICMP, UDP and TCP are the main packets that are visible in the above screenshot.

The timestamp followed by the link level headers, then by ARP/RARP packets if any,

Then by IPv4 packets if any, followed by TCP packets. The sequence numbers and the length finish defining the outputs.

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



```
ruchira@ruchira-VirtualBox: ~
ruchira@ruchira-VirtualBox:~$ sudo tcpdump -i any -c5 icmp
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on any, link-type LINUX_SLL (Linux cooked), capture size 262144 bytes
20:57:45.224838 IP 10.0.4.47 > maa05s03-in-f14.1e100.net: ICMP echo request, id 5689, seq 1, l
ength 64
20:57:45.238045 IP maa05s03-in-f14.1e100.net > 10.0.4.47: ICMP echo reply, id 5689, seq 1, len
gth 64
20:57:46.225953 IP 10.0.4.47 > maa05s03-in-f14.1e100.net: ICMP echo request, id 5689, seq 2, l
ength 64
20:57:46.240023 IP maa05s03-in-f14.1e100.net > 10.0.4.47: ICMP echo reply, id 5689, seq 2, len
gth 64
20:57:47.227398 IP 10.0.4.47 > maa05s03-in-f14.1e100.net: ICMP echo request, id 5689, seq 3, l
ength 64
5 packets captured
5 packets received by filter
0 packets dropped by kernel
ruchira@ruchira-VirtualBox:~$
```

```
ruchira@ruchira-VirtualBox: ~
ruchira@ruchira-VirtualBox:~$ ping google.com
PING google.com (172.217.163.110) 56(84) bytes of data.
64 bytes from maa05s03-in-f14.1e100.net (172.217.163.110): icmp_seq=1 ttl=114 ti
me=13.2 ms
64 bytes from maa05s03-in-f14.1e100.net (172.217.163.110): icmp_seq=2 ttl=114 ti
me=14.0 ms
64 bytes from maa05s03-in-f14.1e100.net (172.217.163.110): icmp_seq=3 ttl=114 ti
me=12.5 ms
64 bytes from maa05s03-in-f14.1e100.net (172.217.163.110): icmp_seq=4 ttl=114 ti
me=13.8 ms
64 bytes from maa05s03-in-f14.1e100.net (172.217.163.110): icmp_seq=5 ttl=114 ti
me=13.9 ms
```

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

On trying to access the Gmail account sign-in website.

```
ruchira@ruchira-VirtualBox:~$ 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), capture size 262144 bytes
21:01:43.413903 IP 10.0.4.47.57126 > 202.88.156.137.80: Flags [.], ack 568960915, win 63910, length 0
E..(R.@.@.s.
../.X...&.PQv..!...P...u+..
21:01:43.414215 IP 202.88.156.137.80 > 10.0.4.47.57126: Flags [.], ack 1, win 65535, length 0
E..(j<..@....X..
../.P.&!...Qv..P...;.........
21:01:49.045985 IP 10.0.4.47.58378 > 216.58.196.163.80: Flags [.], ack 569219509, win 63882, length 0
E..(..@.@...
../.:...
.P..xh!...P....'..
21:01:49.046215 IP 216.58.196.163.80 > 10.0.4.47.58378: Flags [.], ack 1, win 65535, length 0
E..(j>..@.e..:..
../.P.
!.....xiP...W.........
21:01:53.653930 IP 10.0.4.47.57126 > 202.88.156.137.80: Flags [.], ack 1, win 63910, length 0
E..(R.@.@.s.
../.X...&.PQv..!...P...u+..
21:01:53.654172 IP 202.88.156.137.80 > 10.0.4.47.57126: Flags [.], ack 1, win 65535, length 0
E..(j?..@....X..
../.P.&!...Qv..P...;.........
21:01:59.285844 IP 10.0.4.47.58378 > 216.58.196.163.80: Flags [.], ack 1, win 63882, length 0
E..(..@.@...
../.:...
.P..xh!...P....'..
21:01:59.286073 IP 216.58.196.163.80 > 10.0.4.47.58378: Flags [.], ack 1, win 65535, length 0
E..(jA..@.e..:..
../.P.
!.....xiP...W.........
21:02:03.894666 IP 10.0.4.47.57126 > 202.88.156.137.80: Flags [.], ack 1, win 63910, length 0
E..(R.@.@.s.
../.X...&.PQv..!...P...u+..
21:02:03.894956 IP 202.88.156.137.80 > 10.0.4.47.57126: Flags [.], ack 1, win 65535, length 0
E..(jI..@..v.X..
../.P.&!...Qv..P...;.........
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

```
ruchira@ruchira-VirtualBox:~$ sudo tcpdump -i any -c10 -nn -w webserver.pcap port 80
tcpdump: listening on any, link-type LINUX_SLL (Linux cooked), capture size 262144 bytes
10 packets captured
11 packets received by filter
0 packets dropped by kernel
```

Task 5: Perform Traceroute checks

Step 1: Run the traceroute using the following command.

sudo traceroute www.google.com

```
ruchira@ruchira-VirtualBox:~$ sudo traceroute www.google.com
traceroute to www.google.com (172.217.26.196), 30 hops max, 60 byte packets
 1  192.168.1.1 (192.168.1.1)  1.582 ms  1.809 ms  1.946 ms
 2  27.7.96.1 (27.7.96.1)  6.023 ms  5.989 ms  6.563 ms
 3  202.88.156.61 (202.88.156.61)  6.542 ms  6.377 ms  6.286 ms
 4  10.241.1.6 (10.241.1.6)  6.206 ms  7.445 ms  7.494 ms
 5  10.240.254.120 (10.240.254.120)  6.042 ms  5.749 ms  5.882 ms
 6  10.240.254.1 (10.240.254.1)  8.713 ms  6.902 ms  7.972 ms
 7  10.241.1.1 (10.241.1.1)  10.557 ms  6.321 ms  9.730 ms
 8  136.232.28.189.static.jio.com (136.232.28.189)  13.902 ms  17.402 ms  20.418 ms
 9  * * *
10  * * *
11  * * *
12  * * *
13  108.170.237.94 (108.170.237.94)  13.714 ms 108.170.236.196 (108.170.236.196)  10.471 ms 74.125.242.129 (74.125.242.129)  16.693 ms
14  * 74.125.242.146 (74.125.242.146)  11.794 ms *
15  108.170.253.97 (108.170.253.97)  12.053 ms * *
16  * * 72.14.237.165 (72.14.237.165)  54.976 ms
17  * * *
18  * * *
19  * * *
20  * * *
21  * * *
22  * * *
23  * * *
24  * * *
25  * * *
26  * * *
27  * * *
28  * * *
29  * * *
30  * * *
```

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

The destination address is 172.217.26.164. [FOUND OUT BY PINGING IN WINDOWS]

The total number of hops is 30, and most of pings have been timed out.

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

```
ruchira@ruchira-VirtualBox:~$ sudo traceroute -n www.google.com
traceroute to www.google.com (172.217.26.196), 30 hops max, 60 byte packets
 1  192.168.1.1  1.337 ms  1.424 ms  1.550 ms
 2  27.7.96.1  3.156 ms  17.030 ms  17.110 ms
 3  202.88.156.61  17.035 ms  17.020 ms  22.149 ms
 4  10.241.1.6  17.288 ms  16.941 ms  17.015 ms
 5  10.240.254.120  16.674 ms  21.804 ms  21.644 ms
 6  10.240.254.1  17.100 ms  16.803 ms  16.865 ms
 7  10.241.1.1  20.893 ms  26.940 ms  16.000 ms
 8  136.232.28.189  21.095 ms  22.745 ms  26.737 ms
 9  * * *
10  * * *
11  * * *
12  * 108.170.253.113  42.698 ms 108.170.253.97  35.963 ms
13  108.170.234.108  34.777 ms 216.239.59.230  28.523 ms 74.125.253.69  22.784 ms
14  * * 72.14.237.165  11.177 ms
15  * 108.170.253.113  13.359 ms  17.049 ms
16  * * 72.14.237.165  11.350 ms
17  * * *
18  * * *
19  * * *
20  * * *
21  * * *
22  * * *
23  * * *
24  * * *
25  * * *
26  * * *
27  * * *
28  * * *
29  * * *
30  * * *
```

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

sudo traceroute -I www.google.com

```
ruchira@ruchira-VirtualBox:~$ sudo traceroute -I www.google.com
traceroute to www.google.com (172.217.26.196), 30 hops max, 60 byte packets
 1  10.0.2.2 (10.0.2.2)  0.227 ms  0.087 ms  0.157 ms
 2  192.168.1.1 (192.168.1.1)  2.517 ms  2.441 ms  2.829 ms
 3  27.7.96.1 (27.7.96.1)  4.699 ms  4.795 ms  5.435 ms
 4  202.88.156.61 (202.88.156.61)  6.087 ms  8.207 ms  8.135 ms
 5  136.232.28.189.static.jio.com (136.232.28.189)  5.856 ms  7.179 ms  7.724 ms
 6  * * *
 7  * * *
 8  * * *
 9  74.125.252.219 (74.125.252.219)  9.826 ms  9.629 ms  9.537 ms
10  72.14.237.165 (72.14.237.165)  12.522 ms  12.447 ms  12.326 ms
11  maa03s23-in-f4.1e100.net (172.217.26.196)  9.591 ms  10.183 ms  11.781 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

```
ruchira@ruchira-VirtualBox:~$ sudo traceroute -T www.google.com
traceroute to www.google.com (172.217.26.196), 30 hops max, 60 byte packets
 1  192.168.1.1 (192.168.1.1)  1.502 ms  1.418 ms  1.904 ms
 2  27.7.96.1 (27.7.96.1)  5.257 ms  5.353 ms  5.902 ms
 3  202.88.156.61 (202.88.156.61)  5.891 ms  5.880 ms  5.800 ms
 4  10.241.1.6 (10.241.1.6)  7.662 ms  7.795 ms  8.194 ms
 5  10.240.254.50 (10.240.254.50)  5.848 ms  5.768 ms  5.758 ms
 6  10.240.254.1 (10.240.254.1)  8.147 ms  5.397 ms  5.633 ms
 7  10.241.1.1 (10.241.1.1)  3.554 ms  4.412 ms  4.158 ms
 8  136.232.28.189.static.jio.com (136.232.28.189)  4.009 ms  4.502 ms  4.366 ms
 9  * * *
10  * * *
11  * * *
12  108.170.226.93 (108.170.226.93)  11.160 ms 216.239.47.9 (216.239.47.9)  11.011 ms 108.170.226.93 (108.170.226.93)  9.994 ms
13  72.14.237.165 (72.14.237.165)  11.493 ms  11.052 ms 74.125.253.69 (74.125.253.69)  10.130 ms
14  maa03s23-in-f196.1e100.net (172.217.26.196)  12.229 ms  9.702 ms  10.015 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

```
ruchira@ruchira-VirtualBox:~$ nmap www.pes.edu

Starting Nmap 7.01 ( https://nmap.org ) at 2020-09-06 18:00 IST
Nmap scan report for www.pes.edu (13.71.123.138)
Host is up (0.014s 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 4.88 seconds
```

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

nmap 163.53.78.128

```
ruchira@ruchira-VirtualBox:~$ nmap 163.53.78.128

Starting Nmap 7.01 ( https://nmap.org ) at 2020-09-06 14:37 IST
Nmap scan report for 163.53.78.128
Host is up (0.011s 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 4.68 seconds
```

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

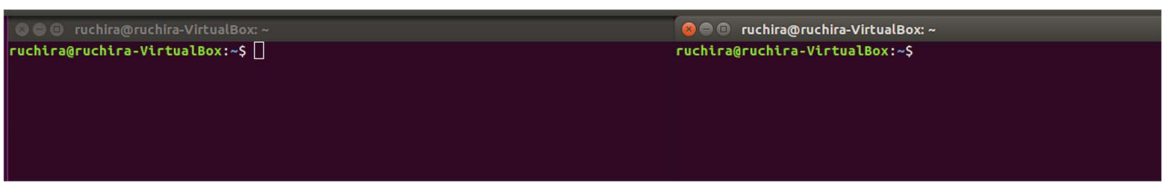
nmap 192.168.1.1 192.168.1.2 192.168.1.3

```
ruchira@ruchira-VirtualBox:~$ nmap 192.168.1.1 192.168.1.2 192.168.1.3

Starting Nmap 7.01 ( https://nmap.org ) at 2020-09-06 14:40 IST
Nmap scan report for 192.168.1.1
Host is up (0.011s latency).
Not shown: 996 closed ports
PORT      STATE SERVICE
53/tcp    open  domain
80/tcp    open  http
5555/tcp  open  freeciv
49152/tcp open  unknown

Nmap scan report for 192.168.1.3
Host is up (0.011s latency).
All 1000 scanned ports on 192.168.1.3 are closed

Nmap done: 3 IP addresses (2 hosts up) scanned in 1.76 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 <your-system-ip-address> 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

Note: 2 students can combine for the following tasks (switch and cables can be taken from Lab technicians)

DONE USING WINDOWS

It did not work with VM to Remote Linux.

b) Inter system communication

Setup a simple switched network of 2 PCs with one acting as Web server. Assign IP addresses for both PCs. Set the capture option as described above.

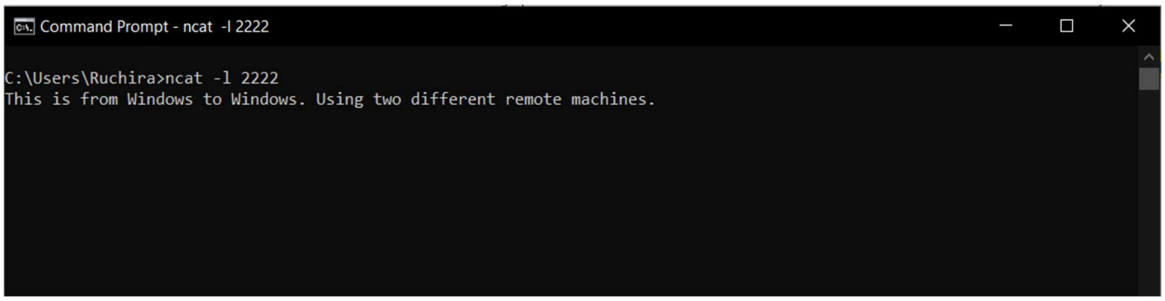
Step 1: Open terminal on Server machine (Machine 1).

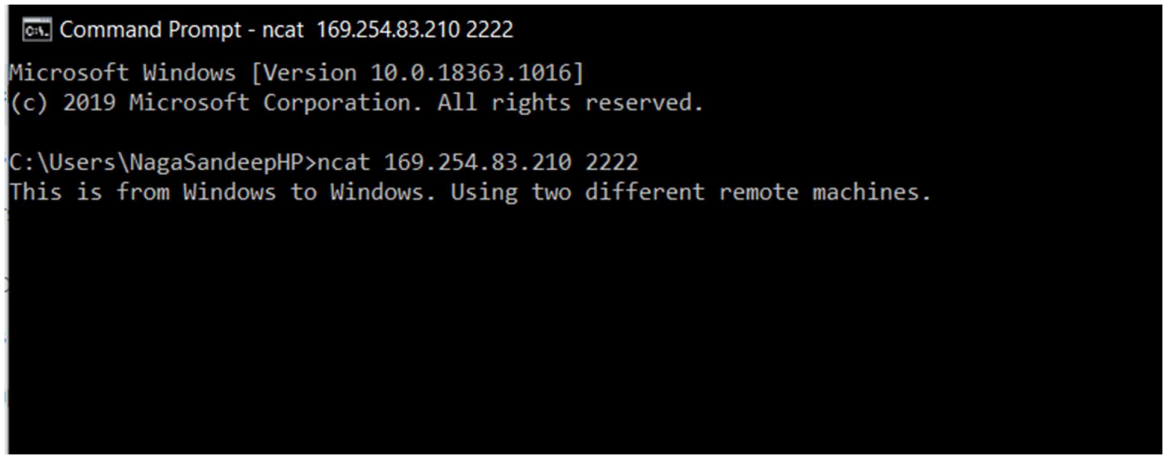
Step 2: Type **nc -l any_portnum**

Step 3: Open terminal on the Client machine (Machine 2)

Step 4: Type nc <server-ip-address> portnum

Step 5: Type anything in client will appear in the server terminal



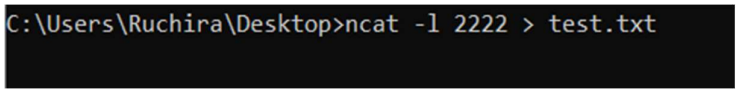


Task 7 b): Use Netcat to Transfer Files

The netcat utility can also be used to transfer files.

Step 1: At the server side, create an empty file named 'test.txt'

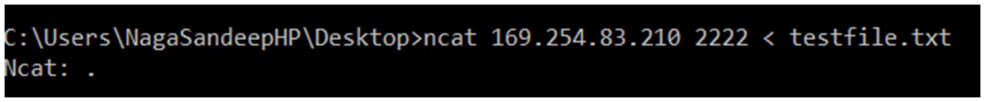
sudo nc -l 555 > test.txt



Step 2: At the client side, we have a file 'testfile.txt'. Add some contents to it.

Step 3: Run the client as:

sudo nc 10.0.2.8 555 < testfile.txt



Step 4: At server side, verify the file transfer using the command

cat test.txt

```
C:\Users\Ruchira\Desktop>ncat -l 2222
1234
```

Task 7 c): Other Commands

COULD NOT BE EXECUTED. PERMISSION DENIED.

1) To test if a particular TCP port of a remote host is open.

   nc -vn 10.0.2.8 555

COULD NOT BE EXECUTED. PERMISSION DENIED.

2) Run a web server with a static web page.
   Step 1: Run the command below on local host (e.g. 10.0.2.8) to start a web server that serves test.html on port 80.
   while true; do sudo nc -lp 80 < test.html; done

COULD NOT BE EXECUTED. PERMISSION DENIED.

Step 2: Now open **http://10.0.2.8/test.html** from another host to access it.

COULD NOT BE EXECUTED. PERMISSION DENIED.

Step 3: Observe the details on the terminal

COULD NOT BE EXECUTED. PERMISSION DENIED.

<u>Questions on above observations:</u>

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

Ans : 1.1. The version of the server is 1.1 as well.

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

Ans : Sun, 06 Sep 2020 01:03:00 GMT

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

Ans : $ ping -c <number of packets> <url>

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

Ans : Simply scan the entire subnet.

Eg:

$ nmap -sP 10.0.4.*

Exercises:

1) Capture and Analyze IPv4 / IPv6 packets

IPv4 / IPv6 packet header

GET	./success.txt HTTP/1.1
HOST	detectportal.firefox.com
USER-AGENT	Mozilla/5.0
ACCEPT-LANGUAGE	en-US, en; q=0.5
CACHE-CONTROL	no-cache
PRAGMA	no-cache
CONNECTION	keep-alive

2) Explore various other network configuration, troubleshooting and debugging tools such as Route, Netstat, etc.

```
ruchira@ruchira-VirtualBox:~$ ip route
default via 10.0.2.2 dev enp0s3  proto static  metric 100
10.0.2.0/24 dev enp0s3  proto kernel  scope link  src 10.0.2.15  metric 100
169.254.0.0/16 dev enp0s3  scope link  metric 1000
172.17.0.0/16 dev docker0  proto kernel  scope link  src 172.17.0.1 linkdown
ruchira@ruchira-VirtualBox:~$ ip route show table local
broadcast 10.0.2.0 dev enp0s3  proto kernel  scope link  src 10.0.2.15
local 10.0.2.15 dev enp0s3  proto kernel  scope host  src 10.0.2.15
broadcast 10.0.2.255 dev enp0s3  proto kernel  scope link  src 10.0.2.15
broadcast 127.0.0.0 dev lo  proto kernel  scope link  src 127.0.0.1
local 127.0.0.0/8 dev lo  proto kernel  scope host  src 127.0.0.1
local 127.0.0.1 dev lo  proto kernel  scope host  src 127.0.0.1
broadcast 127.255.255.255 dev lo  proto kernel  scope link  src 127.0.0.1
broadcast 172.17.0.0 dev docker0  proto kernel  scope link  src 172.17.0.1 linkdown
local 172.17.0.1 dev docker0  proto kernel  scope host  src 172.17.0.1
broadcast 172.17.255.255 dev docker0  proto kernel  scope link  src 172.17.0.1 linkdown
ruchira@ruchira-VirtualBox:~$ ip -4 route
default via 10.0.2.2 dev enp0s3  proto static  metric 100
10.0.2.0/24 dev enp0s3  proto kernel  scope link  src 10.0.2.15  metric 100
169.254.0.0/16 dev enp0s3  scope link  metric 1000
172.17.0.0/16 dev docker0  proto kernel  scope link  src 172.17.0.1 linkdown
ruchira@ruchira-VirtualBox:~$ ip -6 route
fe80::/64 dev enp0s3  proto kernel  metric 256  pref medium
```

```
ruchira@ruchira-VirtualBox:~$ netstat
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State

tcp        0      0 10.0.2.15:48324         43.255.166.254:http     ESTABLISHED
tcp        0      0 10.0.2.15:53498         actiontoad.canonic:http CLOSE_WAIT
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags       Type       State         I-Node   Path
unix  2      [ ]         DGRAM                    12534    /run/systemd/cgroups-agent
unix  2      [ ]         DGRAM                    22703    /run/user/1000/systemd/notify
unix  2      [ ]         DGRAM                    12541    /run/systemd/journal/syslog
unix  16     [ ]         DGRAM                    12547    /run/systemd/journal/dev-log
unix  8      [ ]         DGRAM                    12553    /run/systemd/journal/socket
unix  3      [ ]         DGRAM                    12533    /run/systemd/notify
unix  3      [ ]         STREAM     CONNECTED     26345    @/tmp/dbus-cfuYfJHnRI
unix  3      [ ]         STREAM     CONNECTED     25989
unix  3      [ ]         STREAM     CONNECTED     25299
unix  3      [ ]         STREAM     CONNECTED     14134
unix  3      [ ]         STREAM     CONNECTED     26858    @/tmp/ibus/dbus-mTfqElx2
unix  3      [ ]         STREAM     CONNECTED     26564    /run/systemd/journal/stdout
unix  3      [ ]         STREAM     CONNECTED     27149
unix  3      [ ]         STREAM     CONNECTED     23999
unix  3      [ ]         STREAM     CONNECTED     26996
unix  3      [ ]         STREAM     CONNECTED     26583
unix  3      [ ]         STREAM     CONNECTED     25498    @/tmp/.X11-unix/X0
unix  3      [ ]         STREAM     CONNECTED     14135    /run/systemd/journal/stdout
unix  3      [ ]         STREAM     CONNECTED     26857
unix  3      [ ]         STREAM     CONNECTED     26562
unix  3      [ ]         STREAM     CONNECTED     25211
unix  3      [ ]         STREAM     CONNECTED     27158    /run/systemd/journal/stdout
unix  3      [ ]         STREAM     CONNECTED     20629
unix  3      [ ]         STREAM     CONNECTED     24429    /run/systemd/journal/stdout
unix  3      [ ]         DGRAM                    20307
unix  2      [ ]         DGRAM                    27088
unix  3      [ ]         STREAM     CONNECTED     26574    @/tmp/dbus-fEDneI7UbW
unix  3      [ ]         STREAM     CONNECTED     25596    /var/run/dbus/system_bus_socket
unix  3      [ ]         STREAM     CONNECTED     25300    @/tmp/dbus-fEDneI7UbW
unix  3      [ ]         STREAM     CONNECTED     27068
unix  3      [ ]         STREAM     CONNECTED     26588    @/tmp/dbus-fEDneI7UbW
unix  3      [ ]         STREAM     CONNECTED     24634    @/tmp/ibus/dbus-mTfqElx2
unix  3      [ ]         STREAM     CONNECTED     27156
unix  3      [ ]         STREAM     CONNECTED     23998
```

```
ruchira@ruchira-VirtualBox:~$ netstat -r
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
default         10.0.2.2        0.0.0.0         UG        0 0          0 enp0s3
10.0.2.0        *               255.255.255.0   U         0 0          0 enp0s3
link-local      *               255.255.0.0     U         0 0          0 enp0s3
172.17.0.0      *               255.255.0.0     U         0 0          0 docker0
```

```
ruchira@ruchira-VirtualBox:~$ netstat -s
Ip:
    15482 total packets received
    1 with invalid addresses
    0 forwarded
    0 incoming packets discarded
    15481 incoming packets delivered
    14209 requests sent out
    40 outgoing packets dropped
Icmp:
    80 ICMP messages received
    0 input ICMP message failed.
    ICMP input histogram:
        destination unreachable: 80
    80 ICMP messages sent
    0 ICMP messages failed
    ICMP output histogram:
        destination unreachable: 80
IcmpMsg:
        InType3: 80
        OutType3: 80
Tcp:
    69 active connections openings
    0 passive connection openings
    2 failed connection attempts
    0 connection resets received
    0 connections established
    15145 segments received
    13867 segments send out
    0 segments retransmited
    0 bad segments received.
    4 resets sent
Udp:
    172 packets received
    80 packets to unknown port received.
    0 packet receive errors
    260 packets sent
    IgnoredMulti: 6
UdpLite:
TcpExt:
    2 TCP sockets finished time wait in fast timer
    11 delayed acks sent
    13878 packet headers predicted
```