

Experiment No. 03 B

1. Aim: To study analysis of network packets by using open source sniffing tools like tcpdump and Wireshark in promiscuous and non-promiscuous mode.

2. Software Required : Ubuntu 14.04 OS, Wireshark 2.6.1

3. Theory :

tcpdump is a common packet analyzer that runs under the command line. It allows the user to display TCP/IP and other packets being transmitted or received over a network to which the computer is attached. It is available under most of the **Linux/Unix** based operating systems. tcpdump also gives us an option to save captured packets in a file for future analysis. It saves the file in a **pcap** format, that can be viewed by tcpdump command.

Installing tcpdump:

Many of Linux distributions already shipped with tcpdump tool, if in case you don't have it on systems, you can install it using following command.

sudo apt-get install tcpdump (on debian/ubuntu)

or

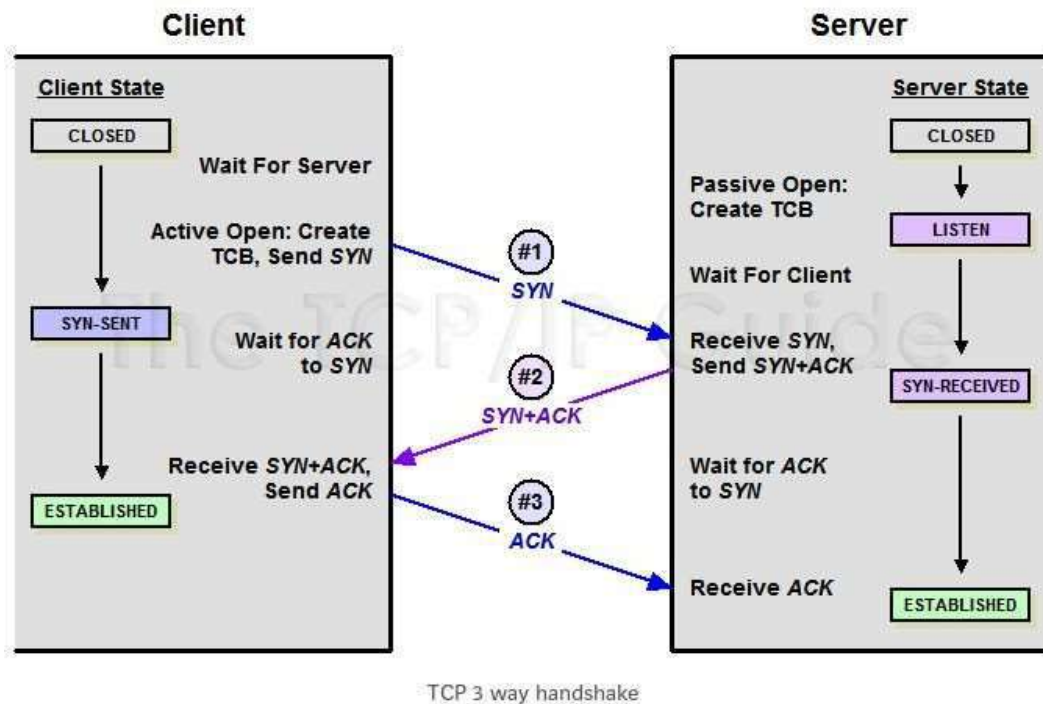
yum install tcpdump (on centos/fedora)

Once **tcpdump** tool is installed on systems, you can continue to browse following commands with their examples.

TCP message flow

1. Connection initialization

TCP connection initialization happens with 3 way handshake.



(1) Client will send a packet with SYN flag is set and random number(R1) included in the sequence number field.

(2) Server will send a packet with SYN flag and ACK flags are set. sequence number field will contain a new random number(R2) and acknowledgement number field will contain clients sequence number +1 ($R1+1$). (Which is the next sequence number server is expecting from the client)

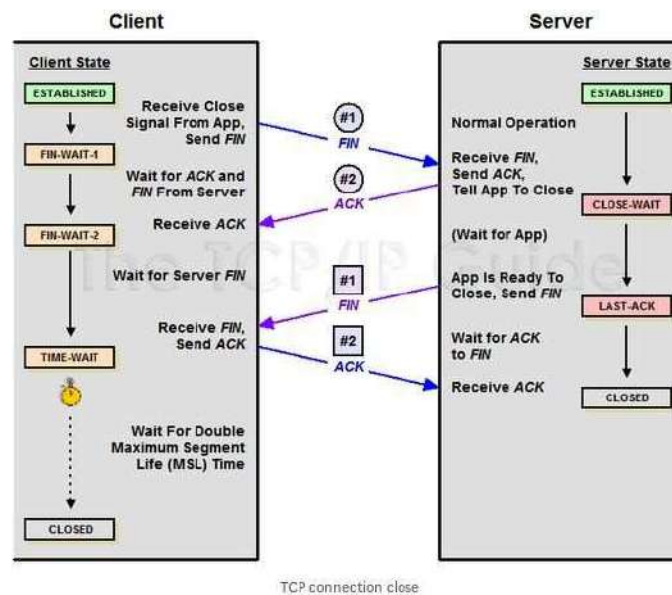
(3) Client will acknowledge servers SYN packet by sending a packet with ACK flag is set and acknowledge number field with $R2+1$. (Which is the next sequence number client is expecting from the server)

2. Load

Payloads will travel both the directions of the TCP connection after the connection initialization. All the packets will set the ACK flag, PSH, URG flags may or may not be set.

3. Termination

TCP connection is normally terminating using a special procedure where each side independently closes its end of the link. It normally begins with one of the application processes signaling to its TCP layer that the session is no longer needed. That device sends a message with FIN flag set to tell the other device that it wants to end the connection, which then get acknowledged. When the responding device is ready, it too sends a FIN, after waiting a period of time for the ACK to be received, the session is closed.



Running tcpdump :

Following are some of the commonly used commands with arguments that can be useful in generating TCP dumps with different level of information. We can use most of the arguments to specify the level of detail we need and to apply

filters. When you run `tcpdump` command it will capture all the packets for specified Interface, until you hit `ctrl+c` button. You might need root access to run following commands:

- **`tcpdump -D`** : display all available interfaces

```
apsit@apsit-HP-Notebook:/$ tcpdump -D
1.wlo1 [Up, Running]
2.any (Pseudo-device that captures on all interfaces) [Up, Running]
3.lo [Up, Running, Loopback]
4.enp1s0 [Up]
5.bluetooth0 (Bluetooth adapter number 0)
6.nflog (Linux netfilter log (NFLOG) interface)
7.nfqueue (Linux netfilter queue (NFQUEUE) interface)
8.usbmon1 (USB bus number 1)
9.usbmon2 (USB bus number 2)
apsit@apsit-HP-Notebook:/$
```

- **`tcpdump -i wlo1`** : capture traffic at the interface “wlo1”

```
apsit@apsit-HP-Notebook:/$ sudo tcpdump -i wlo1
[sudo] password for apsit:
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on wlo1, link-type EN10MB (Ethernet), capture size 262144 bytes
01:08:57.072974 IP 192.168.0.3.39146 > ec2-23-22-162-56.compute-1.amazonaws.com.https: F
lags [.], ack 3368797950, win 1444, options [nop,nop,TS val 142745535 ecr 2122488719], l
ength 0
01:08:57.162523 IP 192.168.0.3.38352 > domain.name.dlink.com.domain: 20947+ PTR? 56.162.
22.23.in-addr.arpa. (43)
01:08:57.230722 IP domain.name.dlink.com.domain > 192.168.0.3.38352: 20947 1/0/0 PTR ec2
-23-22-162-56.compute-1.amazonaws.com. (97)
01:08:57.231672 IP 192.168.0.3.38352 > domain.name.dlink.com.domain: 8090+ PTR? 3.0.168.
192.in-addr.arpa. (42)
01:08:57.236148 IP domain.name.dlink.com.domain > 192.168.0.3.38352: 8090 NXDomain 0/0/0
(42)
01:08:57.236893 IP 192.168.0.3.38352 > domain.name.dlink.com.domain: 32152+ PTR? 1.0.168
.192.in-addr.arpa. (42)
01:08:57.245049 IP domain.name.dlink.com.domain > 192.168.0.3.38352: 32152* 1/0/0 PTR do
main.name.dlink.com. (77)
01:08:57.322531 IP ec2-23-22-162-56.compute-1.amazonaws.com.https > 192.168.0.3.39146: F
lags [.], ack 1, win 422, options [nop,nop,TS val 2122491308 ecr 142745535], length 0
^C
8 packets captured
8 packets received by filter
0 packets dropped by kernel
apsit@apsit-HP-Notebook:/$
```

- **`tcpdump -i any`** : capture traffic at any interface
- **`tcpdump -i wlo1 port 80`** : capture traffic at the interface “wlo1” on port 80


```
apsit@apsit-HP-Notebook:/$ sudo tcpdump -i wlo1 port 80
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on wlo1, link-type EN10MB (Ethernet), capture size 262144 bytes
01:10:08.961873 IP 192.168.0.3.59832 > vz01-phx.stablehost.com.http: Flags [S], seq 9326
85527, win 29200, options [mss 1460,sackOK,TS val 1388467646 ecr 0,nop,wscale 7], length
0
01:10:09.215356 IP vz01-phx.stablehost.com.http > 192.168.0.3.59832: Flags [S.], seq 211
8994519, ack 932685528, win 14480, options [mss 1452,sackOK,TS val 620956985 ecr 1388467
646,nop,wscale 7], length 0
01:10:09.215393 IP 192.168.0.3.59832 > vz01-phx.stablehost.com.http: Flags [.], ack 1, w
in 229, options [nop,nop,TS val 1388467900 ecr 620956985], length 0
01:10:09.215841 IP 192.168.0.3.59832 > vz01-phx.stablehost.com.http: Flags [P.], seq 1:5
01, ack 1, win 229, options [nop,nop,TS val 1388467900 ecr 620956985], length 500: HTTP:
GET /capture-tcp-syn-ack-fin-packets-tcpdump.html HTTP/1.1
01:10:09.469501 IP vz01-phx.stablehost.com.http > 192.168.0.3.59832: Flags [.], ack 501,
win 122, options [nop,nop,TS val 620957239 ecr 1388467900], length 0
01:10:11.007879 IP vz01-phx.stablehost.com.http > 192.168.0.3.59832: Flags [.], seq 1441
:2881, ack 501, win 122, options [nop,nop,TS val 620958776 ecr 1388467900], length 1440:
HTTP
```

- **tcpdump -i wlo1 -c 5** : capture 5 packets at the interface “wlo1”

```
apsit@apsit-HP-Notebook:/$ sudo tcpdump -i wlo1 -c 5
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on wlo1, link-type EN10MB (Ethernet), capture size 262144 bytes
01:12:12.862633 IP 192.168.0.3.39666 > a23-35-33-226.deploy.static.akamaitechnologies.co
m.https: Flags [S], seq 1481548601, win 29200, options [mss 1460,sackOK,TS val 175521882
3 ecr 0,nop,wscale 7], length 0
01:12:12.863803 IP 192.168.0.3.38352 > domain.name.dlink.com.domain: 17570+ PTR? 226.33.
35.23.in-addr.arpa. (43)
01:12:12.891986 IP a23-35-33-226.deploy.static.akamaitechnologies.com.https > 192.168.0.
3.39666: Flags [S.], seq 2577026599, ack 1481548602, win 28960, options [mss 1452,sackOK
,TS val 137780409 ecr 1755218823,nop,wscale 7], length 0
01:12:12.892029 IP 192.168.0.3.39666 > a23-35-33-226.deploy.static.akamaitechnologies.co
m.https: Flags [.], ack 1, win 229, options [nop,nop,TS val 1755218852 ecr 137780409], l
ength 0
01:12:12.894756 IP 192.168.0.3.39666 > a23-35-33-226.deploy.static.akamaitechnologies.co
m.https: Flags [P.], seq 1:547, ack 1, win 229, options [nop,nop,TS val 1755218855 ecr 1
37780409], length 546
5 packets captured
17 packets received by filter
9 packets dropped by kernel
apsit@apsit-HP-Notebook:/$
```

- **tcpdump -i wlo1 tcp** : capture only tcp traffic at interface “wlo1”

```
apsit@apsit-HP-Notebook:/$ sudo tcpdump -i wlo1 tcp
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on wlo1, link-type EN10MB (Ethernet), capture size 262144 bytes
01:13:14.393309 IP 172.217.194.189.https > 192.168.0.3.51680: Flags [P.], seq 1402433658
:1402433718, ack 3397138618, win 255, options [nop,nop,TS val 855118485 ecr 1903280836],
length 60
01:13:14.393367 IP 192.168.0.3.51680 > 172.217.194.189.https: Flags [.], ack 60, win 254
, options [nop,nop,TS val 1903306808 ecr 855118485], length 0
01:13:14.608977 IP 192.168.0.3.32920 > ec2-184-72-237-155.compute-1.amazonaws.com.https:
Flags [.], ack 3310232932, win 319, options [nop,nop,TS val 1344348399 ecr 2589624678],
length 0
01:13:14.865798 IP ec2-184-72-237-155.compute-1.amazonaws.com.https > 192.168.0.3.32920:
Flags [.], ack 1, win 123, options [nop,nop,TS val 2589627302 ecr 1344306625], length 0
01:13:16.130666 IP 192.168.0.3.54928 > edge-star-z-mini-shv-01-bom1.facebook.com.https:
Flags [P.], seq 2423887626:2423887665, ack 502352641, win 515, options [nop,nop,TS val 1
49184123 ecr 768801575], length 39
01:13:16.131684 IP 192.168.0.3.44018 > ec2-13-112-136-133.ap-northeast-1.compute.amazona
ws.com.https: Flags [P.], seq 205128468:205128514, ack 3182194387, win 341, options [nop
,nop,TS val 182986181 ecr 100612615], length 46
```

- **tcpdump -i wlo1 src 192.168.43.169**: capture traffic at interface “wlo1” with source IP 192.168.43.169

```
apsit@apsit-HP-Notebook:/$ sudo tcpdump -i wlo1 src 192.168.43.169
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on wlo1, link-type EN10MB (Ethernet), capture size 262144 bytes
01:15:32.708950 IP 192.168.43.169.60668 > ec2-34-248-137-81.eu-west-1.compute.amazonaws.com.https: Flags [.), ack 3401443443, win 362, options [nop,nop,TS val 51721363 ecr 2506259663], length 0
01:15:32.710098 IP 192.168.43.169.43524 > 192.168.43.1.domain: 13796+ PTR? 81.137.248.34.in-addr.arpa. (44)
^C
2 packets captured
2 packets received by filter
0 packets dropped by kernel
apsit@apsit-HP-Notebook:/$
```

tcpdump -i wlo1 dst 192.168.43.169 : capture traffic at interface “wlo1” with destination IP 192.168.43.169

To capture only TCP SYN packets:

sudo tcpdump -i wlo1 "tcp[tcpflags] & (tcp-syn) != 0" >/home/apsit/Desktop/syn.txt

```
apsit@apsit-HP-Notebook:/$ sudo tcpdump -i wlo1 "tcp[tcpflags] & (tcp-syn) != 0" >/home/apsit/Desktop/syn.txt
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on wlo1, link-type EN10MB (Ethernet), capture size 262144 bytes
^C4 packets captured
8 packets received by filter
0 packets dropped by kernel
apsit@apsit-HP-Notebook:/$
```

Syn.txt :

```
1 01:30:31.099632 IP apsiti-HP-Notebook.49268 > ec2-52-71-204-3.compute-1.amazonaws.com.https:
  Flags [S], seq 3751027409, win 29200, options [mss 1460,sackOK,TS val 1706718246 ecr
  0,nop,wscale 7], length 0
2 01:30:31.471148 IP apsiti-HP-Notebook.42634 > ec2-54-69-151-54.us-
  west-2.compute.amazonaws.com.https: Flags [S], seq 2080293865, win 29200, options [mss
  1460,sackOK,TS val 2815575307 ecr 0,nop,wscale 7], length 0
3 01:30:31.487139 IP ec2-52-71-204-3.compute-1.amazonaws.com.https > apsiti-HP-Notebook.49268:
  Flags [S.], seq 268755605, ack 3751027410, win 26847, options [mss 1400,sackOK,TS val 109303526
  ecr 1706718246,nop,wscale 8], length 0
4 01:30:31.625455 IP apsiti-HP-Notebook.50954 > 104.219.111.135.https: Flags [S], seq 3201638441,
  win 29200, options [mss 1460,sackOK,TS val 2100860446 ecr 0,nop,wscale 7], length 0
5 |
```

To capture only TCP ACK packets:

```
sudo tcpdump -i wlo1 "tcp[tcpflags] & (tcp-ack) != 0" >/home/apsit/Desktop/ack.txt
```

```
1 01:34:00.950362 IP bom05s08-in-f14.1e100.net.https > apsit-HP-Notebook.45250: Flags [S.], seq
  2935813833, ack 3223070162, win 60192, options [mss 1380,sackOK,TS val 577951110 ecr
  4020449693,nop,wscale 8], length 0
2 01:34:00.950436 IP apsit-HP-Notebook.45250 > bom05s08-in-f14.1e100.net.https: Flags [.], ack 1,
  win 229, options [nop,nop,TS val 4020449795 ecr 577951110], length 0
3 01:34:00.956678 IP apsit-HP-Notebook.45250 > bom05s08-in-f14.1e100.net.https: Flags [P.], seq
  1:575, ack 1, win 229, options [nop,nop,TS val 4020449802 ecr 577951110], length 574
4 01:34:01.060352 IP bom05s08-in-f14.1e100.net.https > apsit-HP-Notebook.45250: Flags [.], ack
  575, win 240, options [nop,nop,TS val 577951219 ecr 4020449802], length 0
5 01:34:01.060399 IP bom05s08-in-f14.1e100.net.https > apsit-HP-Notebook.45250: Flags [P.], seq
  1:157, ack 575, win 240, options [nop,nop,TS val 577951219 ecr 4020449802], length 156
6 01:34:01.060432 IP apsit-HP-Notebook.45250 > bom05s08-in-f14.1e100.net.https: Flags [.], ack
  157, win 237, options [nop,nop,TS val 4020449905 ecr 577951219], length 0
```

To capture only TCP FIN packets:

```
sudo tcpdump -i wlo1 "tcp[tcpflags] & (tcp-fin) != 0" >/home/apsit/Desktop/fin.txt
```

```
1 01:35:57.791953 IP bom05s08-in-f10.1e100.net.https > 192.168.43.169.53626: Flags [F.], seq
  1046525628, ack 3550107812, win 244, options [nop,nop,TS val 4084820507 ecr 283862804], length 0
2 01:35:59.849334 IP 192.168.43.169.55630 > 117.18.232.12.https: Flags [F.], seq 2388221349, ack
  416919623, win 341, length 0
3 01:35:59.888280 IP 117.18.232.12.https > 192.168.43.169.55630: Flags [F.], seq 138, ack
  4294967265, win 290, length 0
4
```

To capture only TCP SYN or ACK packets:

```
sudo tcpdump -r <interface> "tcp[tcpflags] & (tcp-syn|tcp-ack) != 0"
```


To capture ssh packet:

```
sudo tcpdump -i wlo1 -x -X -A -nvvv port 22 > ssh.txt
```

```
apsit@apsit-HP-Notebook:~$ sudo tcpdump -i wlan0 -x -X -A -nvvv port 22 > ssh.txt
[sudo] password for apsit:
tcpdump: wlan0: SIOCETH00L(ETH00L_GET_TS_INFO) ioctl failed: No such device
apsit@apsit-HP-Notebook:~$ sudo tcpdump -i wlo1 -x -X -A -nvvv port 22 > ssh.txt

tcpdump: listening on wlo1, link-type EN10MB (Ethernet), capture size 262144 bytes
^C78 packets captured
78 packets received by filter
0 packets dropped by kernel
apsit@apsit-HP-Notebook:~$
```

```
apsit@apsit-HP-Notebook:/$ ssh apsit@192.168.43.32
apsit@192.168.43.32's password:
Welcome to Ubuntu 14.04.3 LTS (GNU/Linux 3.19.0-25-generic i686)

 * Documentation:  https://help.ubuntu.com/

Last login: Thu Aug 23 00:05:54 2018 from apsit-hp-notebook
apsit@apsit-Satellite-C660:~$ exit
logout
Connection to 192.168.43.32 closed.
apsit@apsit-HP-Notebook:/$ ssh apsit@192.168.43.32
apsit@192.168.43.32's password:
Welcome to Ubuntu 14.04.3 LTS (GNU/Linux 3.19.0-25-generic i686)

 * Documentation:  https://help.ubuntu.com/

Last login: Thu Aug 23 01:46:18 2018 from apsit-hp-notebook
```


ssh.txt :

```
1 01:45:38.324806 IP (tos 0x0, ttl 64, id 4461, offset 0, flags [DF], proto TCP (6), length 60)
2   192.168.43.169.52974 > 192.168.43.32.22: Flags [S], cksum 0xa548 (correct), seq 480432837, win 29208, options [mss 1460,sackOK,TS val
607548906 ecr 0,nop,wscale 7], length 0
3   0x0000: 4500 003c 116d 4000 4006 5135 c0a8 2ba9  E..<.nQ.@.Q5..+.
4   0x0010: c0a8 2b29 ceee 0016 1ca2 d2c5 0000 0000  ..+.....L.%
5   0x0020: 0002 7210 6548 0000 0204 05b4 0402 000a  ..f..H.....
6   0x0030: 2436 75ea 0000 0000 0103 0307  ..$0.....
7 01:45:38.328105 IP (tos 0x0, ttl 64, id 0, offset 0, flags [DF], proto TCP (6), length 60)
8   192.168.43.32.22 > 192.168.43.169.52974: Flags [S.], cksum 0xb0c0 (correct), seq 2957010960, ack 480432838, win 28960, options [mss
1460,sackOK,TS val 14326 ecr 607548906,nop,wscale 7], length 0
9   0x0000: 4500 003c 0000 4000 4006 62a2 c0a8 2b20  E..<.@.@.b...+.
10  0x0010: c0a8 2ba9 0016 ceee b04c b424 1ca2 d2c6  ..+.....L.%
11  0x0020: a012 7120 09c0 0000 0204 05b4 0402 000a  ..q.....
12  0x0030: 0000 37f6 2436 75ea 0103 0307  ..7.$0.....
13 01:45:38.328151 IP (tos 0x0, ttl 64, id 4462, offset 0, flags [DF], proto TCP (6), length 52)
14   192.168.43.169.52974 > 192.168.43.32.22: Flags [.], cksum 0xa8c4 (correct), seq 1, ack 1, win 229, options [nop,nop,TS val 607548909
ecr 14326], length 0
15  0x0000: 4500 003d 116e 4000 4006 513c c0a8 2ba9  E..4.nQ.@.Q<..+.
16  0x0010: c0a8 2b29 ceee 0016 1ca2 d2c6 b04c b425  ..+.....L.%
17  0x0020: 8018 00e5 a0c4 0000 0101 080a 2436 75ed  .....$0d.
18  0x0030: 0000 37f6  ..7.
19 01:45:38.329104 IP (tos 0x0, ttl 64, id 4463, offset 0, flags [DF], proto TCP (6), length 93)
20   192.168.43.169.52974 > 192.168.43.32.22: Flags [P.], cksum 0xb26a (correct), seq 1:42, ack 1, win 229, options [nop,nop,TS val
607548910 ecr 14326], length 41
21  0x0000: 4500 005d 116f 4000 4006 5112 c0a8 2ba9  E...@.@.Q...+.
22  0x0010: c0a8 2b29 ceee 0016 1ca2 d2c6 b04c b425  ..+.....L.%
23  0x0020: 8018 00e5 826a 0000 0101 080a 2436 75ee  ....j.....$0d.
24  0x0030: 0000 37f6 5353 482d 322e 302d 4f70 650e  ..7.5SH-2.0-Open
25  0x0040: 5353 485f 372e 3270 3220 5562 750e 7475  SSH.7.2pZ.Ubuntu
26  0x0050: 2d34 7562 750e 7475 322e 340d 0a  -Aubuntu2.4..
27 01:45:38.420770 IP (tos 0x0, ttl 64, id 33756, offset 0, flags [DF], proto TCP (6), length 52)
28   192.168.43.32.22 > 192.168.43.169.52974: Flags [.], cksum 0xa899 (correct), seq 1, ack 42, win 227, options [nop,nop,TS val 14329 ecr
607548910], length 0
```

To capture telnet packet:

sudo tcpdump -i wlo1 -x -X -A -nvvv port 23 > telnet.txt

```
apsit@apsit-HP-Notebook:~$ sudo tcpdump -i wlo1 -x -X -A -nvvv port 23 > telnet.
txt
tcpdump: listening on wlo1, link-type EN10MB (Ethernet), capture size 262144 byt
es
^C55 packets captured
55 packets received by filter
0 packets dropped by kernel
apsit@apsit-HP-Notebook:~$
```

```
apsit@apsit-HP-Notebook:/$ telnet 192.168.43.32
Trying 192.168.43.32...
Connected to 192.168.43.32.
Escape character is '^]'.
Ubuntu 14.04.3 LTS
apsit-Satellite-C660 login: apsit
Password:
Last login: Thu Aug 23 01:48:57 IST 2018 from apsit-hp-notebook on pts/4
Welcome to Ubuntu 14.04.3 LTS (GNU/Linux 3.19.0-25-generic i686)

 * Documentation:  https://help.ubuntu.com/

apsit@apsit-Satellite-C660:~$
```

telnet.txt:

```
1 01:49:45.562871 IP (tos 0x10, ttl 64, id 35235, offset 0, flags [DF], proto TCP (6), length 60)
2   192.168.43.169.48516 > 192.168.43.32.23: Flags [S], cksum 0x82cb (correct), seq 550714553, win 29200, options [mss 1460,sackOK,TS val
   607796136 ecr 0,nop,wscale 7], length 0
3   0x0000: 4510 003c 89a3 4000 4006 d0ee c0a8 2ba9  E...@.b....+
4   0x0010: c0a8 2b20 bd84 0017 20d3 3cb9 0000 0000  ..+.....<....
5   0x0020: a002 7210 82cb 0000 0204 05b4 0402 080a  ..F.....
6   0x0030: 243a 3ba8 0000 0000 0103 0307  ..$;.....
7 01:49:45.568377 IP (tos 0x10, ttl 64, id 0, offset 0, flags [DF], proto TCP (6), length 60)
8   192.168.43.32.23 > 192.168.43.169.48516: Flags [S.], cksum 0x76ea (correct), seq 2417906488, ack 550714554, win 28960, options [mss
   1460,sackOK,TS val 76136 ecr 607796136,nop,wscale 7], length 0
9   0x0000: 4500 003c 0000 4000 4006 62a2 c0a8 2b20  E...@.b....+
10  0x0010: c0a8 2ba9 0017 bd84 901e 5338 20d3 3cba  ..+.....S0...<
11  0x0020: a012 7120 76ea 0000 0204 05b4 0402 080a  ..G.V.....
12  0x0030: 0001 2908 243a 3ba8 0103 0307  ..)h$;.....
13 01:49:45.568450 IP (tos 0x10, ttl 64, id 35236, offset 0, flags [DF], proto TCP (6), length 52)
14   192.168.43.169.48516 > 192.168.43.32.23: Flags [.], cksum 0x15ed (correct), seq 1, ack 1, win 229, options [nop,nop,TS val 607796141
   ecr 76136], length 0
15  0x0000: 4510 0034 89a4 4000 4006 d0f5 c0a8 2ba9  E..4..@.b....+
16  0x0010: c0a8 2b20 bd84 0017 20d3 3cba 901e 5339  ..+.....<...59
17  0x0020: 8010 00e5 15ed 0000 0101 080a 243a 3bad  .....$;..
18  0x0030: 0001 2908  ..)h
19 01:49:45.569267 IP (tos 0x10, ttl 64, id 35237, offset 0, flags [DF], proto TCP (6), length 79)
20   192.168.43.169.48516 > 192.168.43.32.23: Flags [P.], cksum 0x967b (correct), seq 1:28, ack 1, win 229, options [nop,nop,TS val
   607796142 ecr 76136], length 27
21 Telnet:
22 0x0000: fffd 03          DO SUPPRESS GO AHEAD
23 0x0003: fffb 18          WILL TERMINAL TYPE
24 0x0006: fffb 1f          WILL NAMS
25 0x0009: fffb 20          WILL TSPEED
26 0x000c: fffb 21          WILL LFLOW
27 0x000f: fffb 22          WILL LINEMODE
28 0x0012: fffb 27          WILL NCH-ENVIRON
29 0x0015: fffd 05          DO STATUS
30 0x0018: fffb 23          WILL XD15PLOC [telnet]
```

Wireshark:

Wireshark is a free application that allows you to capture and view the data traveling back and forth on your network, providing the ability to drill down and read the contents of each packet – filtered to meet your specific needs. It is commonly utilized to troubleshoot network problems as well as to develop and test software. This open-source protocol analyzer is widely accepted as the industry standard, winning its fair share of awards over the years.

Wireshark has a rich feature set which includes the following:

- Deep inspection of hundreds of protocols, with more being added all the time
- Live capture and offline analysis
- Standard three-pane packet browser
- Multi-platform: Runs on Windows, Linux, OS X, Solaris, FreeBSD, NetBSD, and many others
- Captured network data can be browsed via a GUI, or via the TTY-mode TShark utility
- The most powerful display filters in the industry.

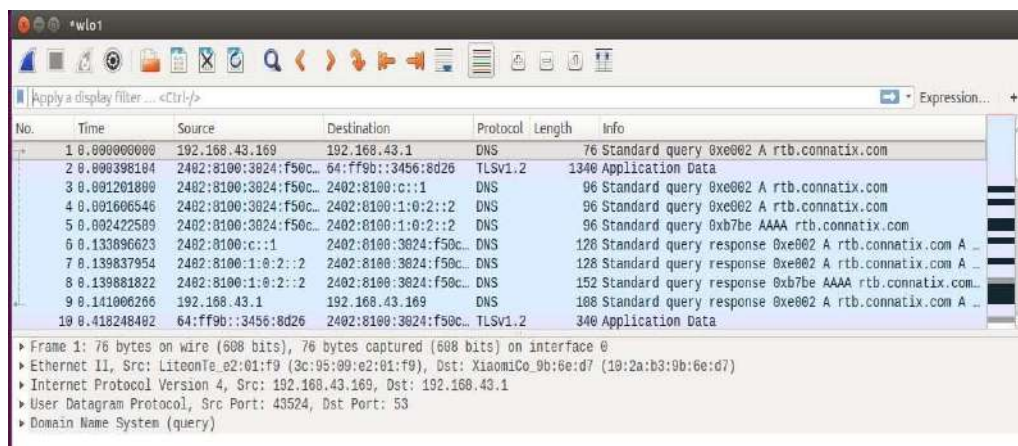
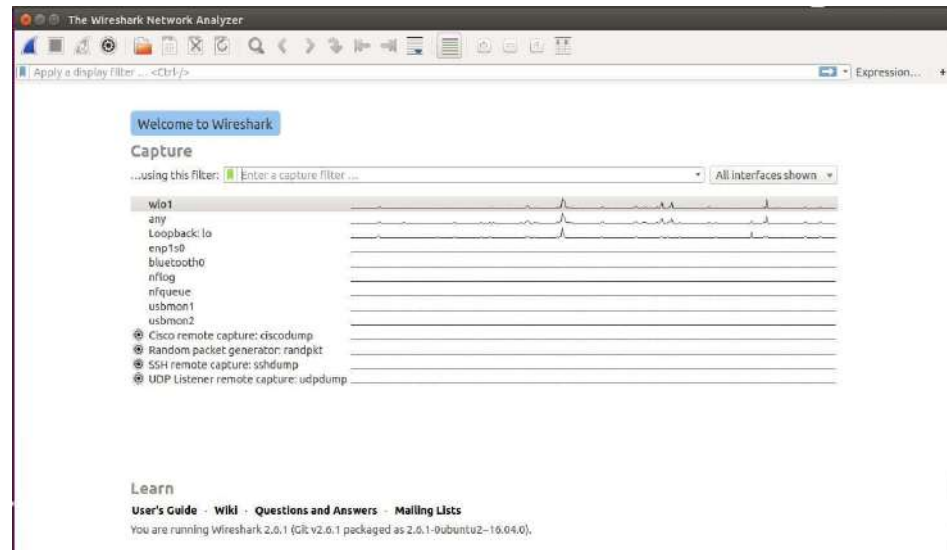
Installing wireshark :

Department of Information Technology | APSIT

sudo apt-get install wireshark

Capture Data Packets in Wireshark:

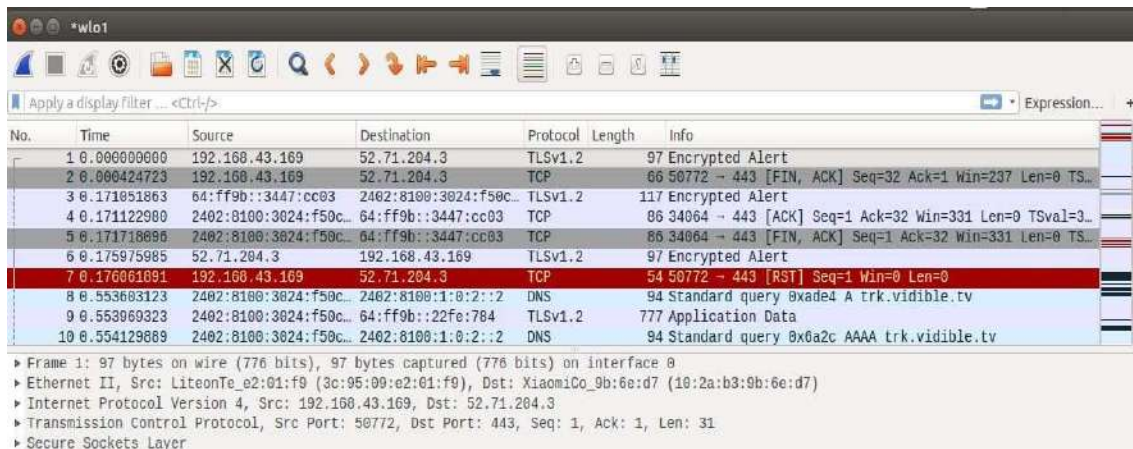
When you first launch Wireshark a welcome screen similar to the one shown above should be visible, containing a list of available network connections on your current device.



To begin capturing, select the interface and click on Capture button at the top.

Demonstration to capture telnet password using Wireshark:

1. Start capturing packets in Wireshark. While in process initiate a telnet connection.



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.43.169	52.71.204.3	TLSv1.2	97	Encrypted Alert
2	0.000424723	192.168.43.169	52.71.204.3	TCP	66	50772 → 443 [FIN, ACK] Seq=32 Ack=1 Win=237 Len=0 TS...
3	0.171051863	64:ff9b::3447:cc03	2402:8100:3024:f50c...	TLSv1.2	117	Encrypted Alert
4	0.171122900	2402:8100:3024:f50c...	64:ff9b::3447:cc03	TCP	86	34064 → 443 [ACK] Seq=1 Ack=32 Win=331 Len=0 TSval=3...
5	0.171716096	2402:8100:3024:f50c...	64:ff9b::3447:cc03	TCP	86	34064 → 443 [FIN, ACK] Seq=1 Ack=32 Win=331 Len=0 TS...
6	0.175975985	52.71.204.3	192.168.43.169	TLSv1.2	97	Encrypted Alert
7	0.176061091	192.168.43.169	52.71.204.3	TCP	54	50772 → 443 [RST] Seq=1 Win=0 Len=0
8	0.553603123	2402:8100:3024:f50c...	2402:8100:1:0:2::2	DNS	94	Standard query 0xade4 A trk.vidible.tv
9	0.553069323	2402:8100:3024:f50c...	64:ff9b::22fe:784	TLSv1.2	777	Application Data
10	0.554129069	2402:8100:3024:f50c...	2402:8100:1:0:2::2	DNS	94	Standard query 0x0a2c AAAA trk.vidible.tv

► Frame 1: 97 bytes on wire (776 bits), 97 bytes captured (776 bits) on interface 0
► Ethernet II, Src: LiteonTe_e2:01:f9 (3c:95:00:e2:01:f9), Dst: XiaomiCo_9b:6e:d7 (10:2a:b3:9b:6e:d7)
► Internet Protocol Version 4, Src: 192.168.43.169, Dst: 52.71.204.3
► Transmission Control Protocol, Src Port: 50772, Dst Port: 443, Seq: 1, Ack: 1, Len: 31
► Secure Sockets Layer

```
apsit@apsit-HP-Notebook:/$ telnet 192.168.43.32
Trying 192.168.43.32...
Connected to 192.168.43.32.
Escape character is '^]'.
Ubuntu 14.04.3 LTS
apsit-Satellite-C660 login: apsit
Password:
Last login: Thu Aug 23 01:52:59 IST 2018 from apsit-HP-Notebook on pts/5
Welcome to Ubuntu 14.04.3 LTS (GNU/Linux 3.19.0-25-generic i686)

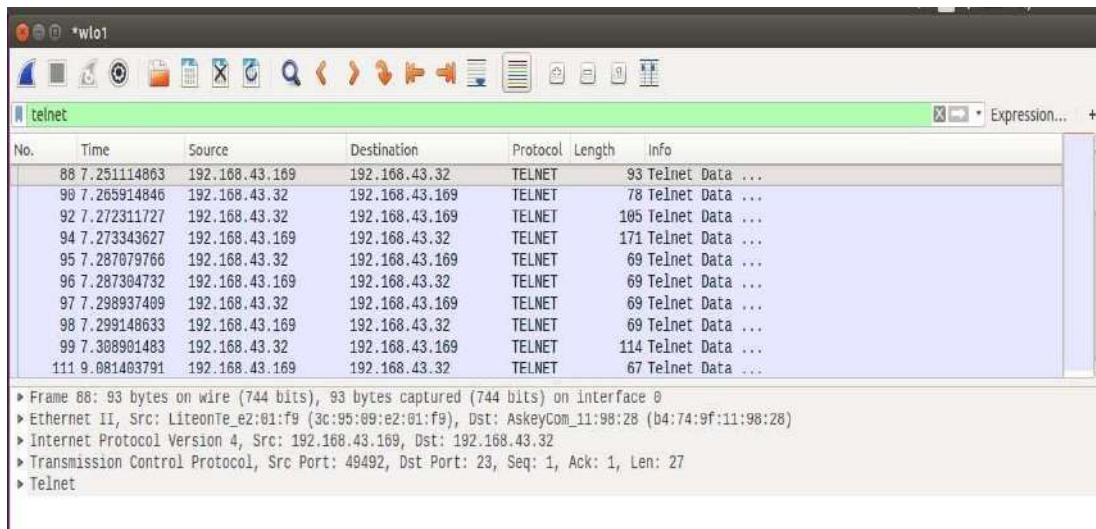
 * Documentation:  https://help.ubuntu.com/

609 packages can be updated.
428 updates are security updates.

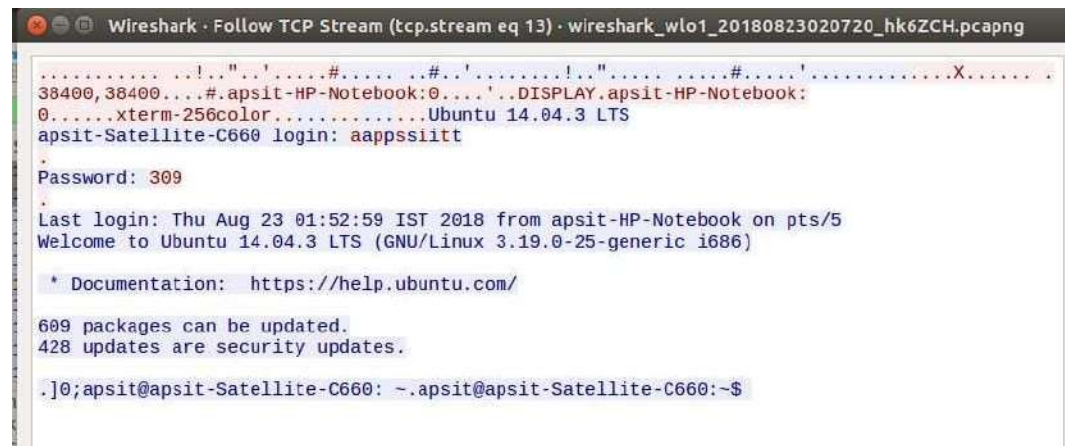
apsit@apsit-Satellite-C660:~$
```

Stop capturing by clicking the stop capturing button at the top in Wireshark.

2. Since we want to here analyze telnet packets, in wireshark in filters, type telnet and the telnet packets captured will be displayed.

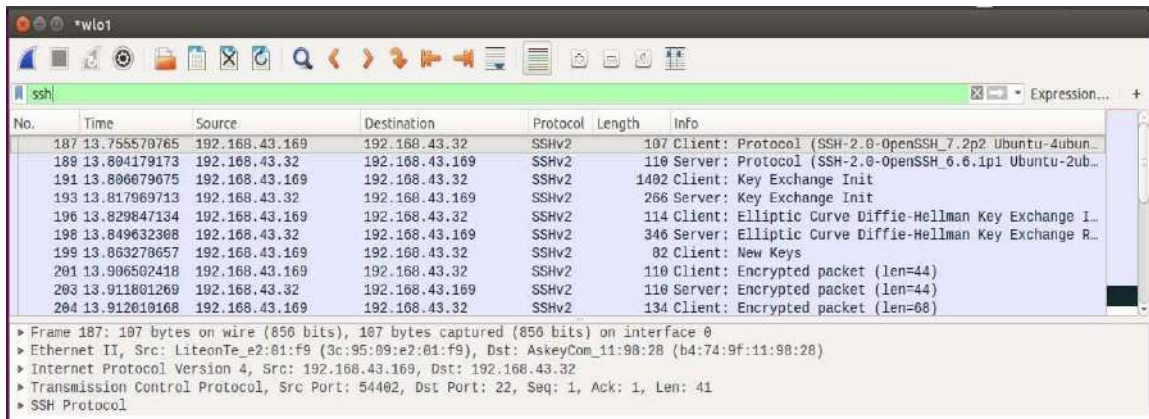


In the first line, we initiate the telnet connection to 192.168.43.32 from 192.168.43.169
In the second line, the connection requests for user login and password. We select this row and click on Analyze in top menu, select follow and then select TCP stream.



Note that the password is displayed along with the login information.

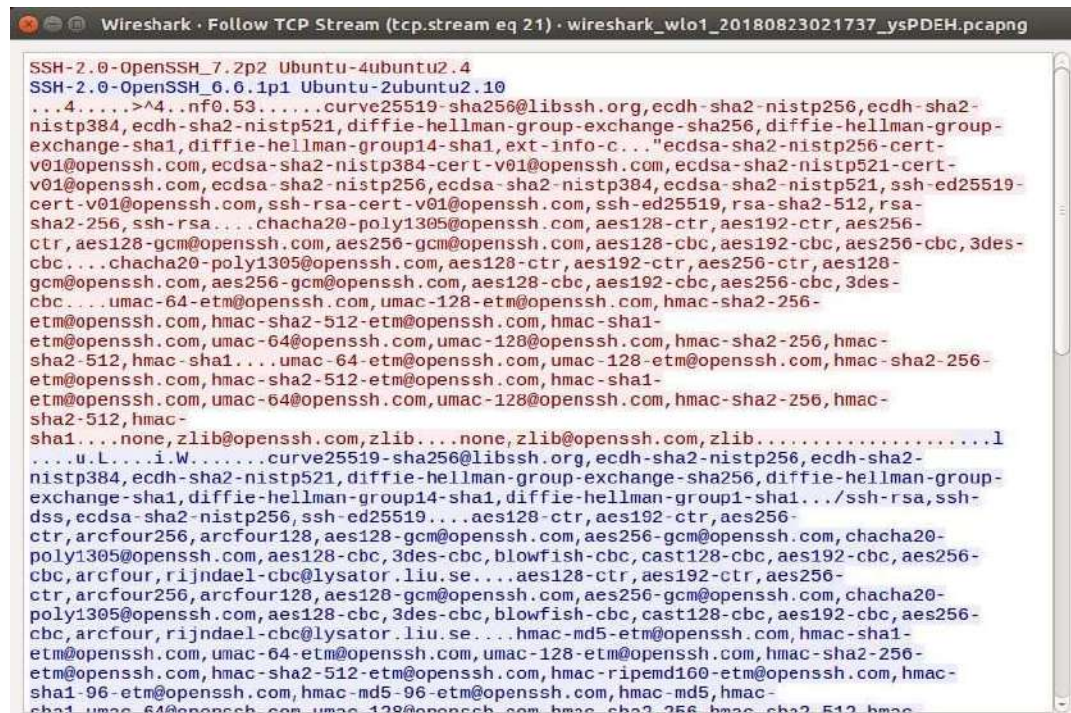
We can capture ssh packets in the same way. While packet capturing is in progress, initiate ssh connection and later monitor the ssh connection from Wireshark.



No.	Time	Source	Destination	Protocol	Length	Info
187	13.755570765	192.168.43.169	192.168.43.32	SSHv2	107	Client: Protocol (SSH-2.0-OpenSSH_7.2p2 Ubuntu-4ubun...
189	13.804179173	192.168.43.32	192.168.43.169	SSHv2	110	Server: Protocol (SSH-2.0-OpenSSH_6.6.1p1 Ubuntu-2ub...
191	13.806079675	192.168.43.169	192.168.43.32	SSHv2	1402	Client: Key Exchange Init
193	13.817969713	192.168.43.32	192.168.43.169	SSHv2	266	Server: Key Exchange Init
196	13.829847134	192.168.43.169	192.168.43.32	SSHv2	114	Client: Elliptic Curve Diffie-Hellman Key Exchange I...
198	13.849632308	192.168.43.32	192.168.43.169	SSHv2	346	Server: Elliptic Curve Diffie-Hellman Key Exchange R...
199	13.863270657	192.168.43.169	192.168.43.32	SSHv2	82	Client: New Keys
201	13.906502418	192.168.43.169	192.168.43.32	SSHv2	110	Client: Encrypted packet (len=44)
203	13.911801269	192.168.43.32	192.168.43.169	SSHv2	110	Server: Encrypted packet (len=44)
204	13.912610168	192.168.43.169	192.168.43.32	SSHv2	134	Client: Encrypted packet (len=68)

▶ Frame 187: 107 bytes on wire (856 bits), 107 bytes captured (856 bits) on interface 0
 ▶ Ethernet II, Src: LiteonTe_e2:81:f9 (3c:95:09:e2:81:f9), Dst: AskeyCom_11:98:28 (b4:74:9f:11:98:28)
 ▶ Internet Protocol Version 4, Src: 192.168.43.169, Dst: 192.168.43.32
 ▶ Transmission Control Protocol, Src Port: 54402, Dst Port: 22, Seq: 1, Ack: 1, Len: 41
 ▶ SSH Protocol

If we analyze ssh packets, we will get something like below:



```

SSH-2.0-OpenSSH_7.2p2 Ubuntu-4ubuntu2.4
SSH-2.0-OpenSSH_6.6.1p1 Ubuntu-2ubuntu2.10
...4....>4....nf0.53.....curve25519-sha256@libssh.org,ecdh-sha2-nistp256,ecdh-sha2-
nistp384,ecdh-sha2-nistp521,diffie-hellman-group-exchange-sha256,diffie-hellman-group-
exchange-sha1,diffie-hellman-group14-sha1,ext-info-c..."ecdsa-sha2-nistp256-cert-
v01@openssh.com,ecdsa-sha2-nistp384-cert-v01@openssh.com,ecdsa-sha2-nistp521-cert-
v01@openssh.com,ecdsa-sha2-nistp256,ecdsa-sha2-nistp384,ecdsa-sha2-nistp521,ssh-ed25519-
cert-v01@openssh.com,ssh-rsa-cert-v01@openssh.com,ssh-ed25519,rsa-sha2-512,rsa-
sha2-256,ssh-rsa....chacha20-poly1305@openssh.com,aes128-ctr,aes192-ctr,aes256-
ctr,aes128-gcm@openssh.com,aes256-gcm@openssh.com,aes128-cbc,aes192-cbc,aes256-
cbc....chacha20-poly1305@openssh.com,aes128-ctr,aes192-ctr,aes256-ctr,aes128-
gcm@openssh.com,aes256-gcm@openssh.com,aes128-cbc,aes192-cbc,aes256-cbc,3des-
cbc....umac-64-etm@openssh.com,umac-128-etm@openssh.com,hmac-sha2-256-
etm@openssh.com,hmac-sha2-512-etm@openssh.com,hmac-sha1-
etm@openssh.com,umac-64@openssh.com,umac-128@openssh.com,hmac-sha2-256,hmac-
sha2-512,hmac-sha1....umac-64-etm@openssh.com,umac-128-etm@openssh.com,hmac-sha2-256-
etm@openssh.com,hmac-sha2-512-etm@openssh.com,hmac-sha1-
etm@openssh.com,umac-64@openssh.com,umac-128@openssh.com,hmac-sha2-256,hmac-
sha2-512,hmac-
sha1....none,zlib@openssh.com,zlib....none,zlib@openssh.com,zlib.....1
....u.L....i.W.....curve25519-sha256@libssh.org,ecdh-sha2-nistp256,ecdh-sha2-
nistp384,ecdh-sha2-nistp521,diffie-hellman-group-exchange-sha256,diffie-hellman-group-
exchange-sha1,diffie-hellman-group14-sha1,diffie-hellman-group1-sha1.../ssh-rsa,ssh-
dss,ecdsa-sha2-nistp256,ssh-ed25519....aes128-ctr,aes192-ctr,aes256-
ctr,arcfour256,arcfour128,aes128-gcm@openssh.com,aes256-gcm@openssh.com,chacha20-
poly1305@openssh.com,aes128-cbc,3des-cbc,blowfish-cbc,cast128-cbc,aes192-cbc,aes256-
cbc,arcfour,rijndael-cbc@lysator.liu.se....aes128-ctr,aes192-ctr,aes256-
ctr,arcfour256,arcfour128,aes128-gcm@openssh.com,aes256-gcm@openssh.com,chacha20-
poly1305@openssh.com,aes128-cbc,3des-cbc,blowfish-cbc,cast128-cbc,aes192-cbc,aes256-
cbc,arcfour,rijndael-cbc@lysator.liu.se....hmac-md5-etm@openssh.com,hmac-sha1-
etm@openssh.com,umac-64-etm@openssh.com,umac-128-etm@openssh.com,hmac-sha2-256-
etm@openssh.com,hmac-sha2-512-etm@openssh.com,hmac-ripemd160-etm@openssh.com,hmac-
sha1-96-etm@openssh.com,hmac-md5-96-etm@openssh.com,hmac-md5,hmac-
sha1-umac-64@openssh.com,umac-128@openssh.com,hmac-sha2-256,hmac-sha2-512,hmac-
  
```

Thus, it can be stated that ssh is much more secure than telnet for remote connections.

Promiscuous and non promiscuous mode:

Promiscuous mode is often used to monitor network activity. Promiscuous mode is the opposite of non-promiscuous mode. When a data packet is transmitted in non-promiscuous mode, all the LAN devices "listen to" the data to determine if the network address included in the data packet is theirs.

- "Promiscuous mode" on both WiFi and Ethernet means having the card accept packets on the current network, even if they're sent to a different MAC address.
- "Non-Promiscuous mode" is WiFi-specific and means having the card accept packets for *any* network, without having to be associated to it.

Promiscuous mode can be enabled as below:

```
apsit@apsit-HP-Notebook:~$ sudo ip link set wlo1 promisc on
[sudo] password for apsit:
apsit@apsit-HP-Notebook:~$ netstat -i
Kernel Interface table
Iface MTU Met RX-OK RX-ERR RX-DRP RX-OVR TX-OK TX-ERR TX-DRP TX-OVR Flg
enp1s0 1500 0 78 0 0 0 121 0 0 0 0 BMU
lo 65536 0 23791 0 0 0 23791 0 0 0 0 LRU
wlo1 1500 0 120989 0 0 0 119907 0 0 0 0 BMPRU
apsit@apsit-HP-Notebook:~$
```

```
wlo1 Link encap:Ethernet HWaddr 3c:95:09:e2:01:f9
      inet addr:192.168.43.169 Bcast:192.168.43.255 Mask:255.255.255.0
      inet6 addr: 2402:8100:3024:f50c:d977:f24e:259:fdda/64 Scope:Global
      inet6 addr: 2402:8100:3024:f50c:ae45:4d3d:2b1f:d265/64 Scope:Global
      inet6 addr: fe80::594c:3e55:695d:8a23/64 Scope:Link
      UP BROADCAST RUNNING PROMISC MULTICAST MTU:1500 Metric:1
      RX packets:121102 errors:0 dropped:0 overruns:0 frame:0
      TX packets:120038 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:84604559 (84.6 MB) TX bytes:19875334 (19.8 MB)
```

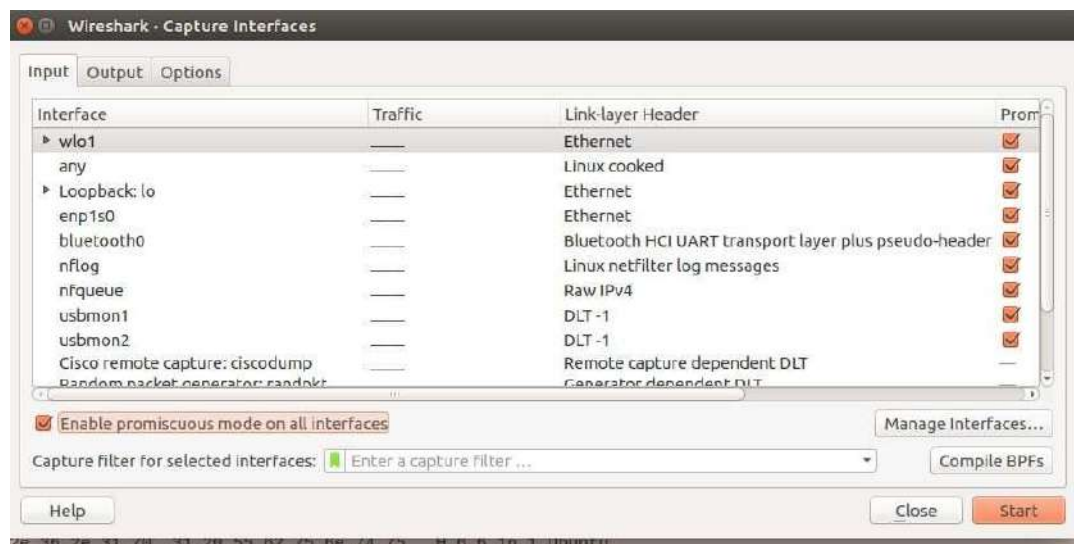



Can be also viewed in ifconfig output :

Promiscuous mode can be disabled as below:

```
apsit@apsit-HP-Notebook:~$ sudo ip link set wlo1 promisc off
apsit@apsit-HP-Notebook:~$ netstat -i
Kernel Interface table
Iface MTU Met RX-OK RX-ERR RX-DRP RX-OVR TX-OK TX-ERR TX-DRP TX-OVR Flg
enp1s0 1500 0 78 0 0 121 0 0 0 0 BMU
lo 65536 0 23905 0 0 0 23905 0 0 0 0 LRU
wlo1 1500 0 122756 0 0 0 121294 0 0 0 0 BMRU
apsit@apsit-HP-Notebook:~$
```

Promiscuous mode enable/disable in Wireshark:



4. Conclusion:

Sometimes a network service is just not behaving the way it should. And the log files do not help you either. Packet sniffing is useful to analyze the data during the transmission in the network. Sniffing tools like tcpdump and Wireshark are useful to implement it. It can be used for network traffic monitoring, traffic analysis, troubleshooting and other useful purposes. Packet sniffers can capture things like passwords and usernames or other sensitive information.