# 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）

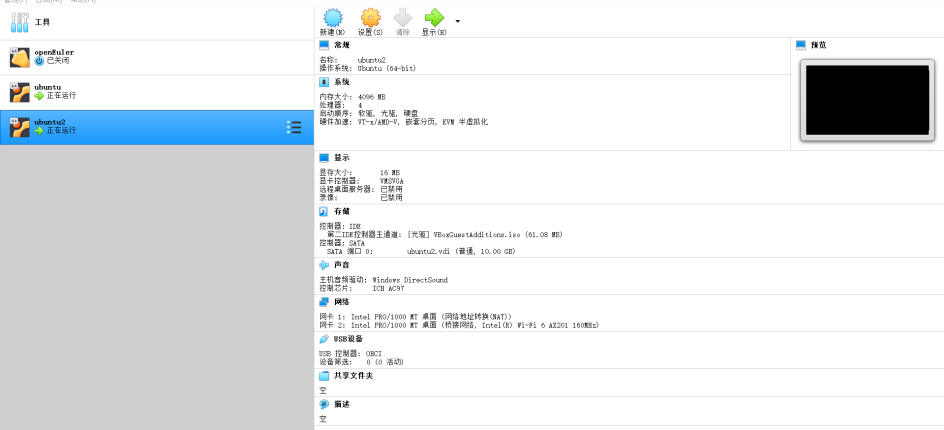
### 三、实验过程及结果分析

**步骤一、安装虚拟机平台**

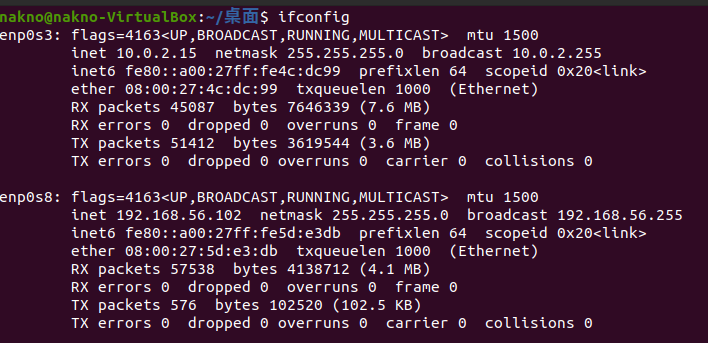
在实验一中已成功安装VirtualBox。

**步骤二、安装虚拟机**

本次实验的环境为安装Apache2的ubuntu虚拟机。



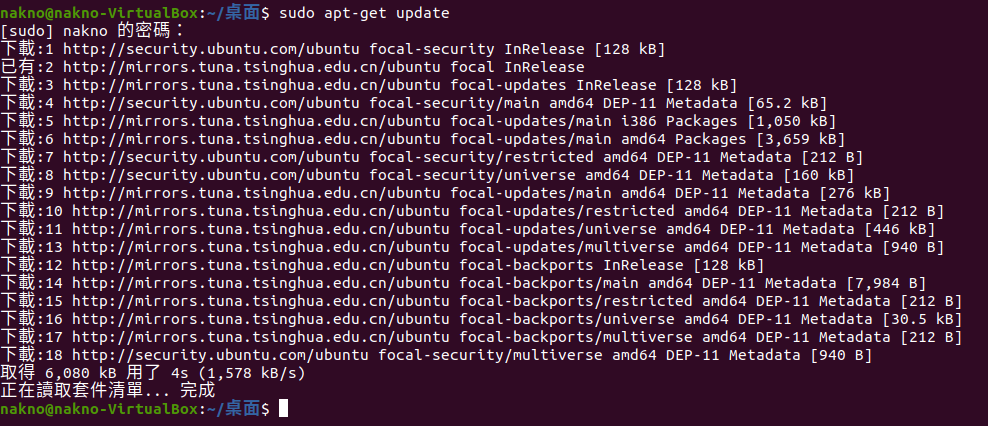
本次实验中，需要安装两个虚拟机，一个作为被攻击者，另一个作为攻击者，它们需要处于同一局域网。因此，需要将两个虚拟机都调整为交接模式。通过ifconfig命令即可查看网络连接信息。在下图出结果中可以发现服务器ip为192.168.56.102。

****

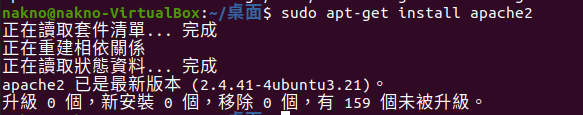
**步骤三、在server虚拟机上安装apache服务器**

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。

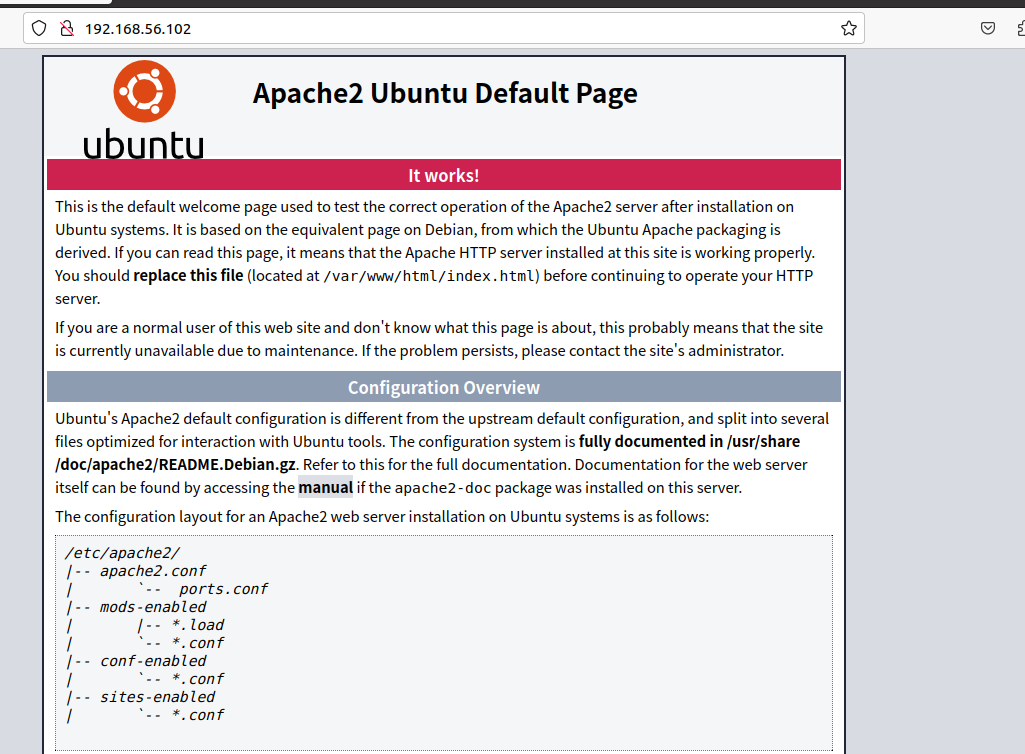
在终端输入命令：apt-get update成功进行更新。



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

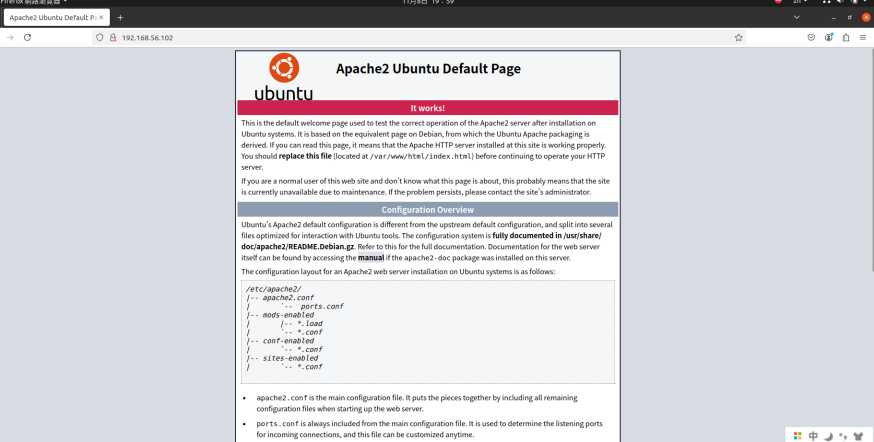


在浏览器中输入127.0.0.1访问，出现安装成功界面，下面开始攻击实验过程。

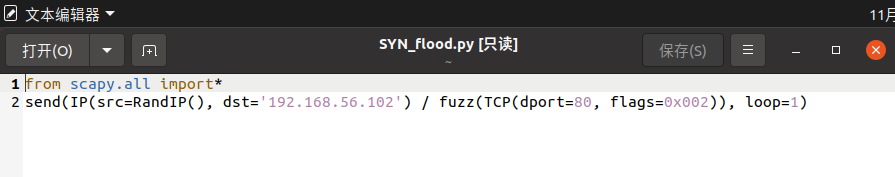


**步骤四、SYN-Flood 攻击**

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。



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。

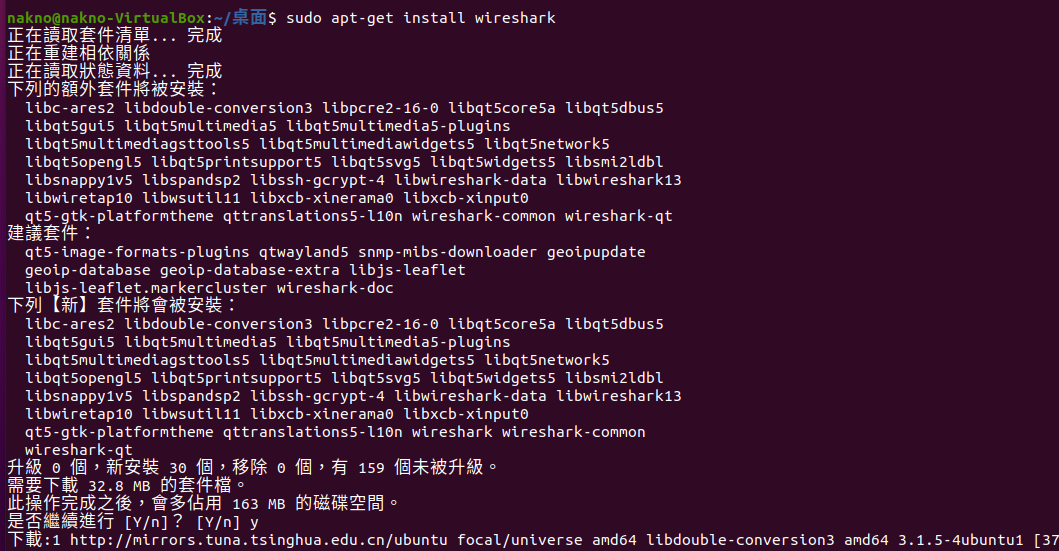
IP(src=RandIP(), dst='192.168.56.102')用于生成一个IP层数据包，src=RandIP()中的RandIP()生成一个随机IP地址，作为数据包的源IP地址。每次发送的数据包的源IP会有所不同，这模拟了来自不同IP的请求，用于伪造数据包或进行拒绝服务攻击。dst='192.168.56.102’是目标IP地址。这里设定的目标是192.168.56.102，代表该IP的设备是攻击目标。

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。

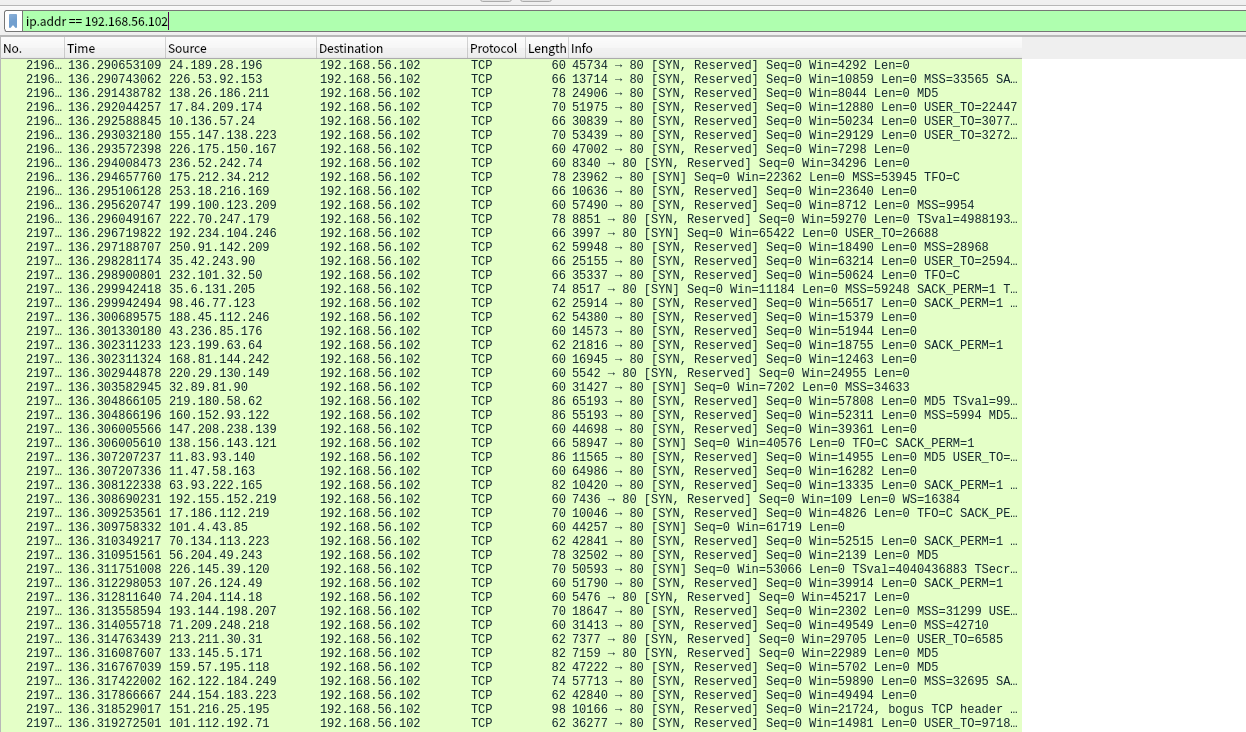
fuzz(TCP(...))则对数据包进行模糊测试，fuzz()函数会随机化TCP头中的字段值，除了已经指定的dport和flags。

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。

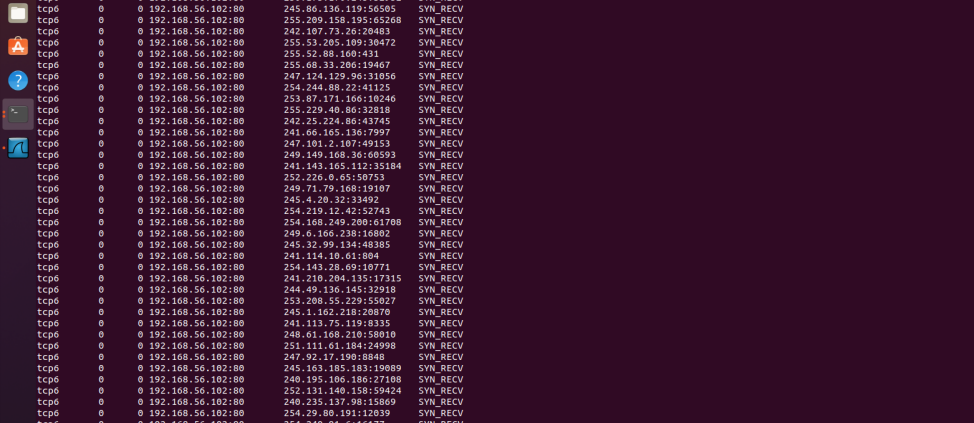
在 sever 端的终端输入 sudo apt-get install wireshark 安装 wireshark



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。

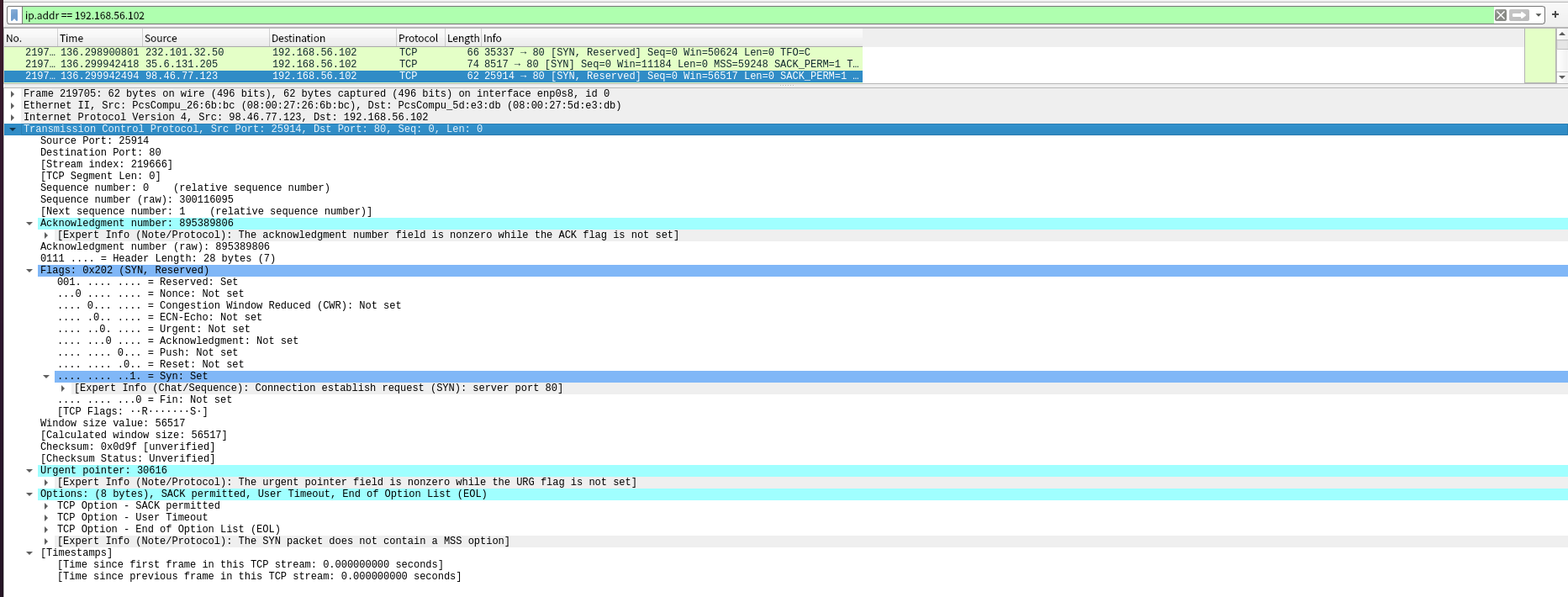


在sever 端内输入 netstat-atn，查看网络信息，可以看到，出现了很多的 SYN 请求。



**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。



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、遇到的问题和解决方法。

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/VBoxGuestAdditions.iso-CSDN博客](https://blog.csdn.net/quantum7/article/details/106443540)，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。

1. 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](https://blog.csdn.net/u012809062/article/details/118102545)中找到解决方法。

1. 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。

### 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。