



《计算机网络与通信》课程

实验六：网络安全相关实验



2023年秋季学期

实验内容

任务1: IPTABLES

- 1.1 ICMP extension
- 1.2 访问控制
- 1.3 端口复用

任务2: nmap扫描

- 2.1 主机发现
- 2.2 端口扫描
- 2.3 nmap全面扫描

任务3: HTTPS协议分析 (TLS)

任务4: ARP spoofing 与 中间人攻击(MITM)

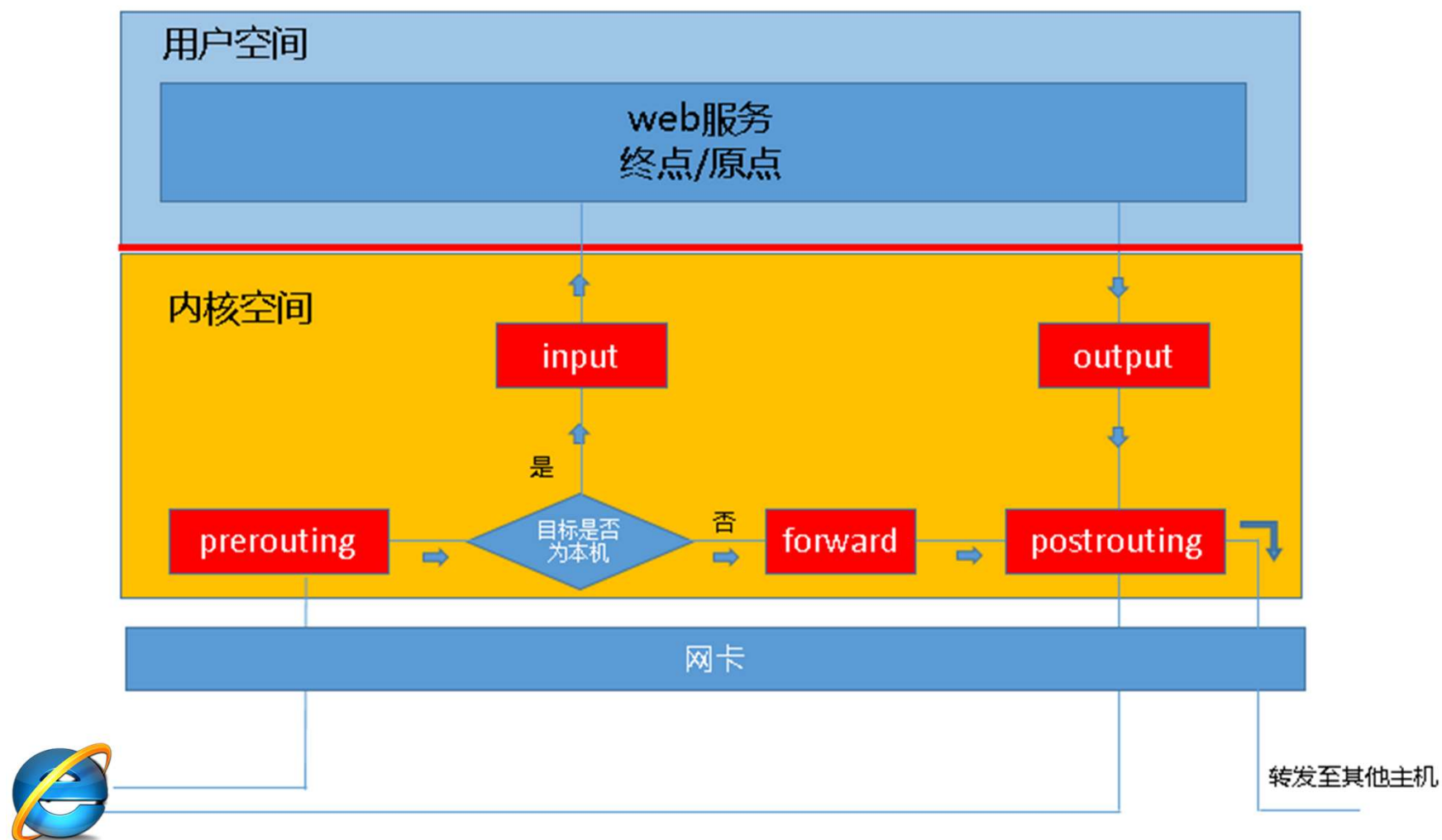
- 4.1 ARP spoofing
- 4.2 捕获网络服务内容和分析 (选作)

Linux防火墙-----iptables

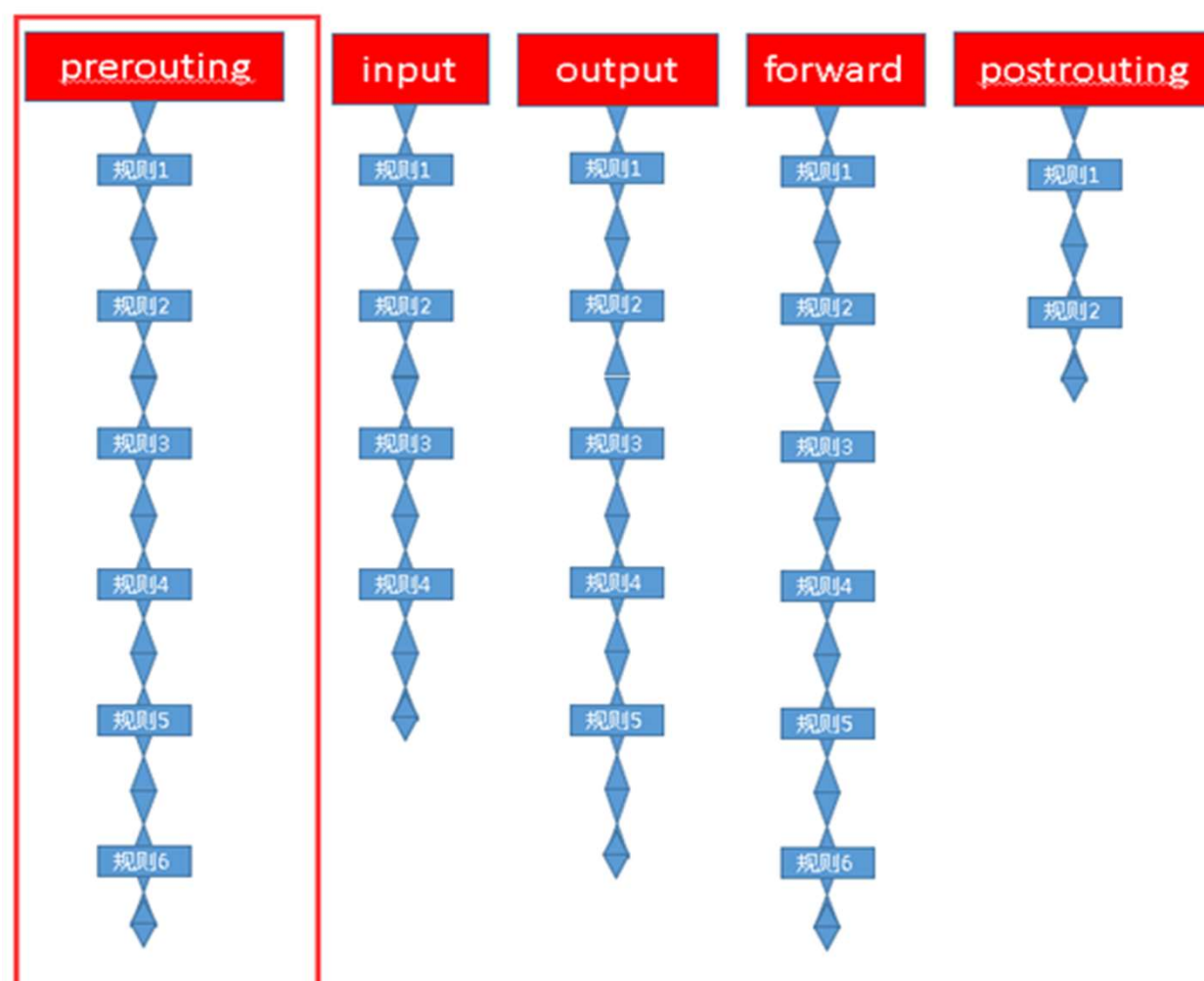
- ❑主机防火墙：针对于单个主机进行防护。
- ❑网络防火墙：往往处于网络入口或边缘，针对于网络入口进行防护，服务于防火墙背后的本地局域网。
- ❑netfilter/iptables 组成Linux平台下的包过滤防火墙
 - 网络地址转换(Network Address Translate)
 - 数据包内容修改
 - 数据包过滤的防火墙功能

官方手册 <https://linux.die.net/man/8/iptables>
论坛详解 <https://www.zsythink.net/archives/1199>

链的概念

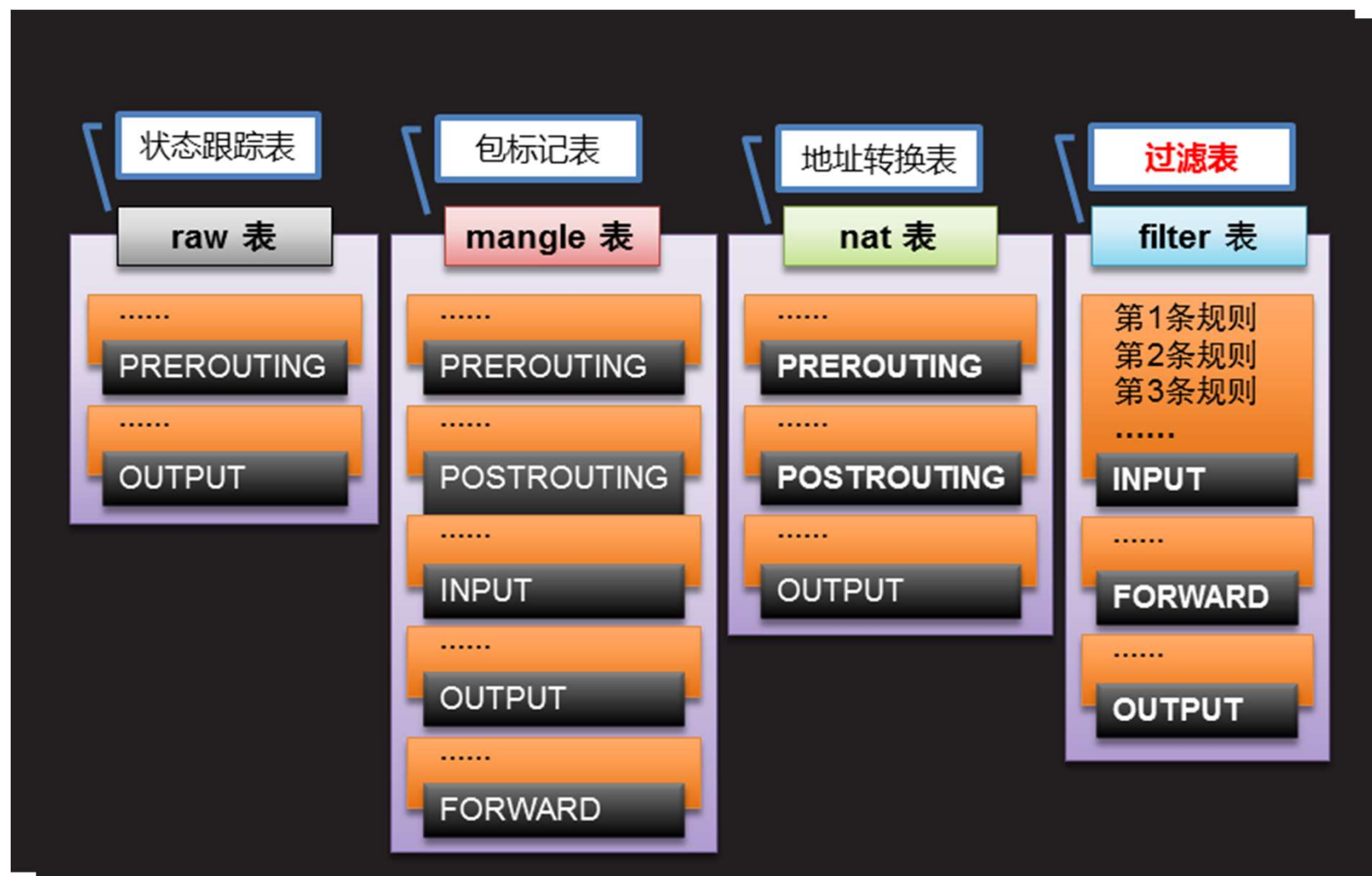


链的概念



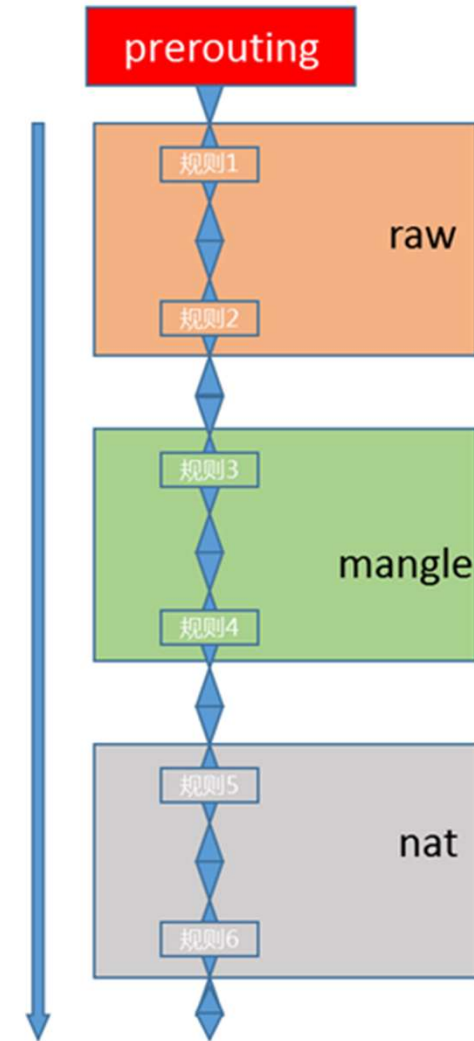
表的概念

- 具有相同功能的规则的集合叫做"表"
 - **filter**表：负责过滤功能，防火墙
 - **nat**表：网络地址转换功能
 - **mangle**表：拆解报文，做出修改，并重新封装
 - **raw**表：决定数据包是否被状态跟踪机制处理

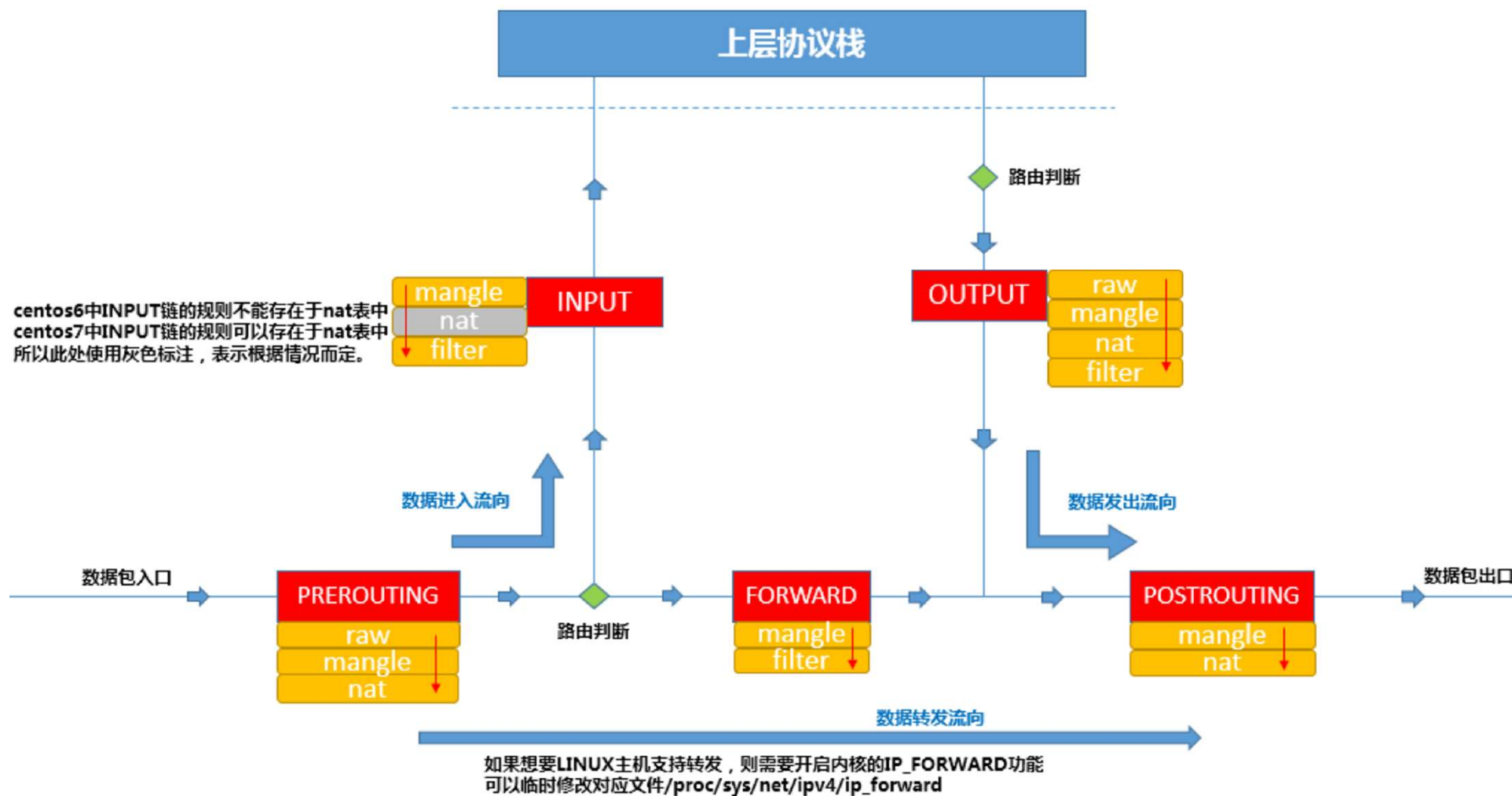


表链关系

- 不同链中包含不同类规则
- PREROUTING 的规则存在于：
 - raw、mangle、nat
- INPUT的规则存在于：
 - mangle、filter
- FORWARD的规则存在于：
 - mangle、filter
- OUTPUT的规则存在于：
 - raw、mangle、nat、filter
- POSTROUTING的规则存在于：
 - mangle、nat
- 优先级次序（由高而低）：
 - raw --> mangle --> nat --> filter



数据经过防火墙流程



规则的概念

❑规则：根据指定的匹配条件来尝试匹配每个流经此处的报文，一旦匹配成功，则由规则后面指定的处理动作进行处理；

❑匹配条件

- 基本匹配条件：
 - 源地址Source IP，目标地址 Destination IP
 - 报文类型tcp、udp、icmp…
 - 网卡
- 扩展匹配条件：以模块的形式存在

❑动作

- ACCEPT：允许数据包通过
- DROP：直接丢弃数据包，不给任何回应信息
- REJECT：拒绝数据包通过，会给发送端一个响应的信息。
- SNAT：源地址转换
- MASQUERADE：是SNAT的一种特殊形式，适用于动态的ip
- DNAT：目标地址转换

Iptables基本命令

✓查看

iptables -t 表名 -n -L 链名 --line-numbers

✓插入

iptables -t 表名 -I 链名 匹配条件 -j 动作

iptables -t 表名 -I 链名 规则序号 匹配条件 -j 动作

✓追加

iptables -t 表名 -A 链名 匹配条件 -j 动作

✓修改默认规则

iptables -t filter -P 链名 规则

✓删除

iptables -t 表名 -D 链名 规则序号

✓清空

iptables -t 表名 -F 链名

Iptables的匹配条件

- 基本匹配条件

- **-s**: 用于匹配报文的源地址
- **-d**: 用于匹配报文的目标地址
- **-p**: 用于匹配报文的协议类型

禁止响应ping: `iptables -t filter -I INPUT -s 192.168.59.0/24 -p icmp -j REJECT`

- 扩展匹配条件

- tcp模块 `-m tcp`

--sport: 用于匹配tcp协议报文的源端口

--dport: 目的端口

禁止访问80端口: `iptables -t filter -I INPUT -p tcp -m tcp --dport 80 -j REJECT`

- icmp模块 `-m icmp`

--icmp-type: 匹配icmp报文的具体类型

任务 1.1 iptables ICMP拓展

任务要求：禁止响应ping，并进行验证(自己能够ping别人)

实验环境：主机+邻居主机 (Ubuntu)

参考思路：

1. *Drop/REJECT ping-request on INPUT*

- (1)查看filter表下INPUT链规则
- (2)在INPUT链上添加一条规则，拒绝接收ICMP echo-request类型（ping请求）的数据包。
- (3)查看新增规则后，filter表下INPUT链规则
- (4)实验成功后，根据规则序号，删除filter表下INPUT链中新增的这条命令，复原环境

2. *Drop/REJECT ping-reply on OUTPUT*

- (1)查看filter表下OUTPUT链规则
- (2)在OUTPUT链上添加一个规则，拒绝回复ICMP echo-reply类型的ping请求。
- (3)查看新增规则后，filter表下OUTPUT链规则
- (4)实验成功后，根据规则序号，删除filter表下OUTPUT链中新增的这条命令，复原环境

3. *Drop/REJECT ICMP by source*

- (1)查看filter表下INPUT链规则
- (2)在INPUT链上添加一条规则，拒绝接收目的主机为自己的ICMP数据包。
- (3)查看新增规则后，filter表下INPUT链规则
- (4)实验成功后，根据规则序号，删除filter表下INPUT链中新增的这条命令，复原环境

ICMP扩展代码-1

类型TYPE	代码CODE	用途 描述 Description	查 询 类 Query	差 错 类 Error
0	0	Echo Reply——回显应答（Ping应答）	x	
3	0	Network Unreachable——网络不可达		x
	1	Host Unreachable——主机不可达		x
	2	Protocol Unreachable——协议不可达		x
	3	Port Unreachable——端口不可达		x
	4	Fragmentation needed but no frag. bit set——需要进行分片但设置不分片比特		x
	5	Source routing failed——源站选路失败		x
	6	Destination network unknown——目的网络未知		x
	7	Destination host unknown——目的主机未知		x
	8	Source host isolated (obsolete)——源主机被隔离（作废不用）		x
	9	Destination network administratively prohibited——目的网络被强制禁止		x
	10	Destination host administratively prohibited——目的主机被强制禁止		x
	11	Network unreachable for TOS——由于服务类型TOS，网络不可达		x
	12	Host unreachable for TOS——由于服务类型TOS，主机不可达		x

ICMP扩展代码-2

类型TYPE	代码CODE	用途 描述 Description	查 询 类 Query	差 错 类 Error
3	13	Communication administratively prohibited by filtering——由于过滤, 通信被强制禁止		x
	14	Host precedence violation——主机越权		x
	15	Precedence cutoff in effect——优先中止生效		x
4	0	Source quench——源端被关闭（基本流控制）		
5	0	Redirect for network——对网络重定向		
	1	Redirect for host——对主机重定向		
	2	Redirect for TOS and network——对服务类型和网络重定向		
	3	Redirect for TOS and host——对服务类型和主机重定向		
8	0	Echo request——回显请求（Ping请求）	x	
9	0	Router advertisement——路由器通告		
10	0	Route solicitation——路由器请求		
11	0	TTL equals 0 during transit——传输期间生存时间为0		x
	1	TTL equals 0 during reassembly——在数据报组装期间生存时间为0		x
12	0	IP header bad (catchall error)——坏的IP首部（包括各种差错）		x
	1	Required options missing——缺少必需的选项		x

ICMP扩展代码-3

类型TYPE	代码CODE	用途 描述 Description	查 询 类 Query	差 错 类 Error
13	0	Timestamp request (obsolete)——时间戳请求（作废不用）	x	
14		Timestamp reply (obsolete)——时间戳应答（作废不用）	x	
15	0	Information request (obsolete)——信息请求（作废不用）	x	
16	0	Information reply (obsolete)——信息应答（作废不用）	x	
17	0	Address mask request——地址掩码请求		
18	0	Address mask reply——地址掩码应答		

任务1.1检查要求

重要过程步骤，请截图，一并检查。

截图应当包括：

- 1.主机Ping邻居主机的记录
- 2.邻居主机Ping主机的记录
- 3.学号及姓名(echo "<id><name>")

注：删除添加的规则后再进行下一步。

任务1.2 访问控制

任务： Ubuntu服务器关闭所有端口之后仅开启80端口访问，客户端进行验证

第一步：

快速搭建HTTP服务： `$python -m SimpleHTTPServer`

同时在Ubuntu配置iptables关闭所有端口，并开启80端口访问(*How?*)

第二步：

Client验证： `$nc -v -w 5 -z IPofUbuntu 75-85`

任务1.2检查要求

重要过程步骤，请截图，一并检查。

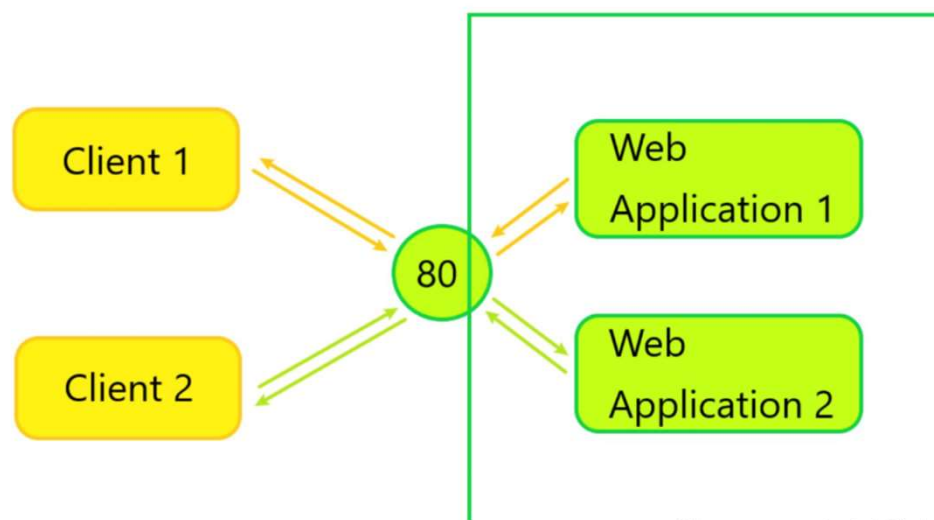
截图应当包括：

- 1.开启端口访问限制之前的扫描记录
- 2.开启端口访问限制之后的扫描记录
- 3.学号及姓名(echo "<id><name>")

任务1.3 端口复用

- **定义：** 端口复用是指不同的应用程序使用相同端口进行通讯。
- **示例场景：** 内网渗透中搭建隧道时，服务器仅允许指定的端口对外开放。利用端口复用可以将3389或22等端口转发到如80端口上，以便外部连接。

示意图：



任务1.3检查要求

实验环境：客户机+服务器（Ubuntu）

▣任务简化：

请将来自<Client IP>dport 2020端口的流量都重定向到2023端口

(1) Server端：

- 在nat表的PREROUTING链中添加一个规则，用于将来自特定客户端IP地址的TCP流量从2020端口重定向到2023端口。
- 监听2023端口

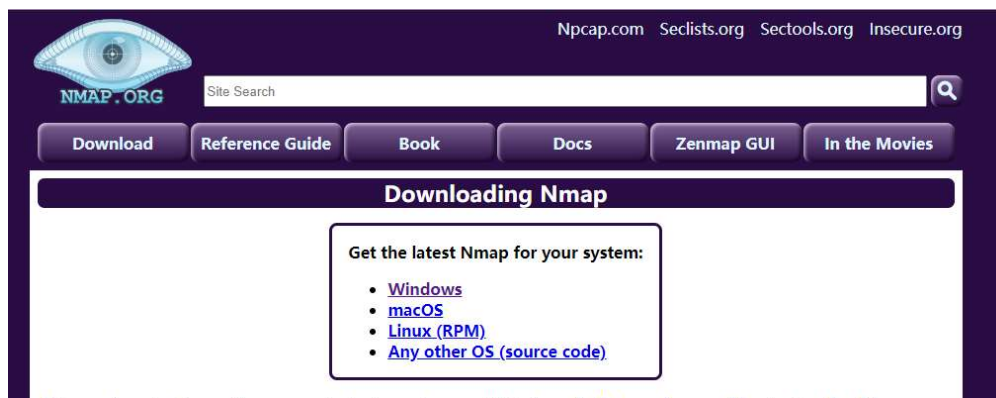
(2) Client端：

- 使用curl向server的2020、2022、2023端口发送消息，观察结果

▣请准备好所有截图，找助教或者老师检查。

任务2: nmap扫描工具

- 官方网址: <https://nmap.org/>
- Nmap是一款开放源代码的网络探测和安全审核的工具，基本包括了常用的扫描方式，并且提供了许多非常实用的辅助功能，可以发现远程服务器是否存活、对外开放的各种TCP端口的分配及提供的服务、所使用的软件版本，如操作系统或其他应用程序的版本，以及可能被利用的系统漏洞。



nmap基本功能

▣ Nmap 四项基本功能：

- 1 主机发现（Host Discovery）
- 2 端口扫描（Port Scanning）
- 3 版本侦测（Version Detection）
- 4 操作系统侦测（Operating System Detection）



nmap基本功能

- Nmap可任意指定主机、网段甚至是整个网络作为扫描目标，扫描方式亦可通过添加合适的选项按需组合。
- 本实验使用基于Windows的Nmap软件，其命令语法如下：
 - `nmap [扫描类型] [选项] <主机或网络 #1.....[#N]>`
 - 在Nmap的所有参数中，只在目标参数是必须给出的，其最简单的形式是在命令行直接输入一个主机名或者一个IP地址。
 - 如果希望扫描一个子网，可以在主机名或者IP地址的后面加上/掩码。



nmap参数

- 使用 **nmap -h** 可快速列出Nmap选项参数的说明
 - -sT 表示TCP全连接扫描（TCP connect()）。
 - -sS 表示TCP半开扫描。
 - -sP 表示ping扫描。
 - -sU 表示UDP扫描。
 - -sA 表示ACK扫描。
 - -sn 表示Nmap 不要执行端口扫描，只进行主机发现，也就是检测目标 IP 地址范围内响应 ICMP Echo 请求的主机。
 - --dns-servers <custom_DNS_server_IP>: 允许指定自定义的 DNS 服务器。
 - -v 表示冗余模式，会给出扫描过程中的详细信息。使用-d选项可以得到更加详细的信息。
 - -Pn表示Nmap 不要对目标主机进行主机发现（ping），而是直接对目标进行扫描，忽略主机的存活状态
 - -A: 表示进行“全面扫描”。这个参数告诉 Nmap 执行操作系统检测、服务版本检测、脚本扫描等，以获取尽可能多的信息。
 - -T0-6: 表示设置扫描速度。Nmap 的 -T 选项允许你调整扫描的速度和探测的侵入程度，例如，-T4 指定了一种相对较快的扫描模式。

其他参数见nmap官方文档手册: <https://nmap.org/man/zh/>

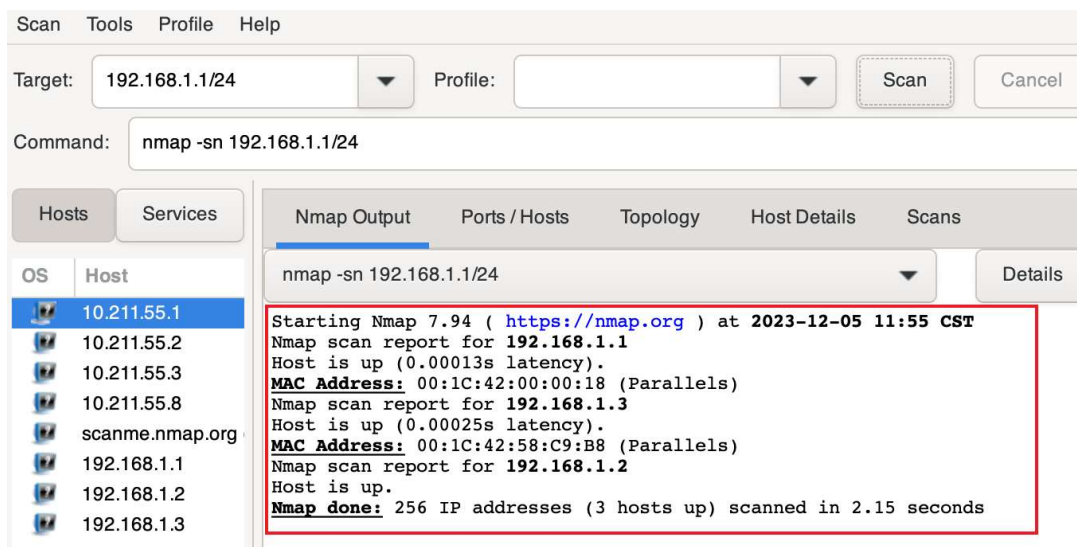
nmap扫描结果

- 主要包括：扫描主机端口的列表、Well-known端口的服务名（如果可能）、端口号、状态和协议
- 每个端口有三种状态：Open、Filtered和Unfiltered
 - Open状态表示，目标主机能够在这个端口使用Accept（）系统调用接受连接；
 - Filtered状态表示，防火墙、包过滤和其他的网络安全软件掩盖了这个端口，禁止Nmap探测其是否打开；
 - Unfiltered表示，这个端口关闭，并且没有防火墙/包过滤软件来隔离Nmap的探测企图。



任务2.1 nmap主机发现

Nmap在局域网范围扫描内存活的主机，`nmap -sn <CIDR地址>`，
提示：CIDR地址设置为自己的虚拟机所在的网络地址

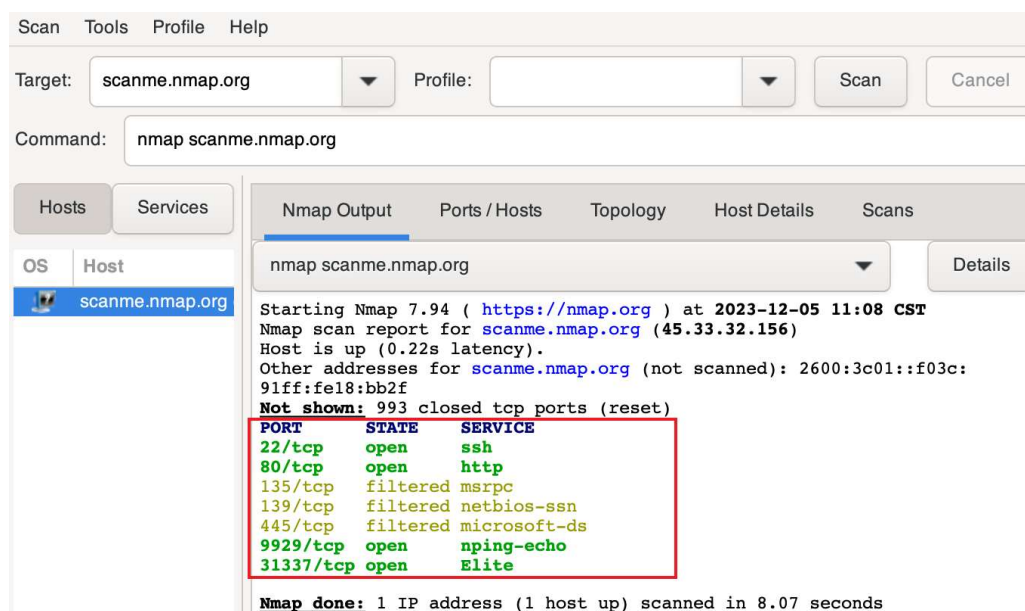


Notice: 请不要对任何非授权的主机/服务器进行扫描!!!

检查要求：主机发现的结果，并解释。

任务2.2 nmap端口扫描

- 扫描远程主机开放的端口，目标远程主机：scanme.nmap.org
- 扫描结果：



```
Scan Tools Profile Help
Target: scanme.nmap.org Profile: Scan Cancel
Command: nmap scanme.nmap.org
Hosts Services
OS Host
scanme.nmap.org
Nmap Output Ports / Hosts Topology Host Details Scans
nmap scanme.nmap.org Details
Starting Nmap 7.94 ( https://nmap.org ) at 2023-12-05 11:08 CST
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.22s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:
91ff:fe18:bb2f
Not shown: 993 closed tcp ports (reset)
PORT      STATE      SERVICE
22/tcp    open       ssh
80/tcp    open       http
135/tcp   filtered  msrpc
139/tcp   filtered  netbios-ssn
445/tcp   filtered  microsoft-ds
9929/tcp  open       nping-echo
31337/tcp open       Elite
Nmap done: 1 IP address (1 host up) scanned in 8.07 seconds
```

Notice: 请不要对任何非授权的主机/服务器进行扫描!!!

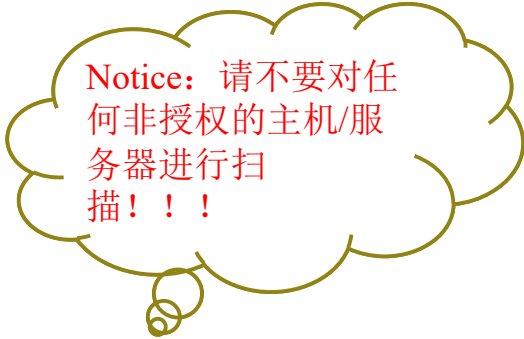
任务2.2 检查要求

- 1、扫描scanme.nmap.org端口的结果，增加参数指定使用google（8.8.8.8 / 8.8.4.4）或其他的DNS服务器进行扫描。
- 2、忽略主机的存活状态，禁止ICMP ping后重新扫描。
- 3、结合前面iptables的实验，开启多个端口（如22，80，443，8000，8080）扫描一次。然后关闭所有端口，开启80（或其他端口）再扫描一次。提示：target主机替换为你的虚拟机/主机 IP地址。



任务2.3 nmap全面扫描

首先使用一台虚拟机（例如地址为192.168.1.3，地址可自定义）作为服务器，在8000端口开启HTTP服务（SimpleHTTPServer或http.server）请使用nmap对其进行全面的扫描（操作系统检测、服务版本检测、脚本扫描等，以获取尽可能多的信息）、输出详细的信息、以较快的速度完成扫描。提示：相关参数请查阅资料。



Notice: 请不要对任何非授权的主机/服务器进行扫描!!!



任务2.3 全面扫描

```
Starting Nmap 7.94 ( https://nmap.org ) at 2023-12-05 12:18 CST
NSE: Loaded 156 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 12:18
Completed NSE at 12:18, 0.00s elapsed
Initiating NSE at 12:18
Completed NSE at 12:18, 0.00s elapsed
Initiating NSE at 12:18
Completed NSE at 12:18, 0.00s elapsed
Initiating ARP Ping Scan at 12:18
Scanning 192.168.1.3 [1 port]
Completed ARP Ping Scan at 12:18, 0.21s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 12:18
Completed Parallel DNS resolution of 1 host. at 12:18, 0.01s elapsed
Initiating SYN Stealth Scan at 12:18
Scanning 192.168.1.3 [1000 ports]
Discovered open port 80/tcp on 192.168.1.3
Discovered open port 8000/tcp on 192.168.1.3
Completed SYN Stealth Scan at 12:18, 0.03s elapsed (1000 total ports)
Initiating Service scan at 12:18
Scanning 2 services on 192.168.1.3
Completed Service scan at 12:19, 41.02s elapsed (2 services on 1 host)
Initiating OS detection (try #1) against 192.168.1.3
Retrying OS detection (try #2) against 192.168.1.3
Retrying OS detection (try #3) against 192.168.1.3
Retrying OS detection (try #4) against 192.168.1.3
Retrying OS detection (try #5) against 192.168.1.3
NSE: Script scanning 192.168.1.3.
Initiating NSE at 12:19
Completed NSE at 12:19, 20.96s elapsed
Initiating NSE at 12:19
Completed NSE at 12:19, 2.03s elapsed
Initiating NSE at 12:19
Completed NSE at 12:19, 0.00s elapsed
Nmap scan report for 192.168.1.3
Host is up (0.00085s latency).
Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
80/tcp    open  http?
8000/tcp  open  http    SimpleHTTPServer 0.6 (Python 2.7.18)
|_ http-title: Directory listing for /
|_ http-server-header: SimpleHTTP/0.6 Python/2.7.18
|_ http-methods:
```

```
PORT      STATE SERVICE VERSION
80/tcp    open  http?
8000/tcp  open  http    SimpleHTTPServer 0.6 (Python 2.7.18)
|_ http-title: Directory listing for /
|_ http-server-header: SimpleHTTP/0.6 Python/2.7.18
|_ http-methods:
|_ Supported Methods: GET HEAD
MAC Address: 00:1C:42:58:C9:B8 (Parallels)
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.94%E=4%D=12/5%OT=80%CT=1%CU=35776%PV=Y%DS=1%DC=D%G=Y%M=001C42%T
OS:M=656EA4DE%P=x86_64-apple-darwin21.6.0)SEQ(CI=Z%II=I)ECN(R=N)T1(R=N)T2(R
OS:=N)T3(R=N)T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T5(R=Y%DF=Y%T=40%W
OS:=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)
OS:T7(R=N)T7(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)U1(R=Y%DF=N%T=40%IP
OS:L=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=N%T=40%CD=S)

Network Distance: 1 hop

TRACEROUTE
HOP RTT      ADDRESS
1   0.85 ms  192.168.1.3

NSE: Script Post-scanning.
Initiating NSE at 12:19
Completed NSE at 12:19, 0.00s elapsed
Initiating NSE at 12:19
Completed NSE at 12:19, 0.00s elapsed
Initiating NSE at 12:19
Completed NSE at 12:19, 0.00s elapsed
Read data files from: /usr/local/bin/./share/nmap
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 82.87 seconds
Raw packets sent: 1278 (62.658KB) | Rcvd: 1032 (44.086KB)
```



任务2: HTTPS协议分析(TLS)

HTTPS vs. TLS/SSL 协议

SSL (Secure Socket Layer, 安全套接字层)

TLS (Transport Layer Security, 传输层安全协议)

TLS是在SSL的基础上标准化的产物

- ✓ 位于传输层和应用层之间，应用层数据不再直接传递给传输层，而是传递给TLS层
- ✓ TLS层对从应用层收到的数据进行加密，并增加TLS头。

HTTPS——“HTTP over TLS” 或 “HTTP over SSL”



HTTPS

HTTP为什么不安全？明文传输

HTTPS如何保证安全？引入加密和身份认证机制

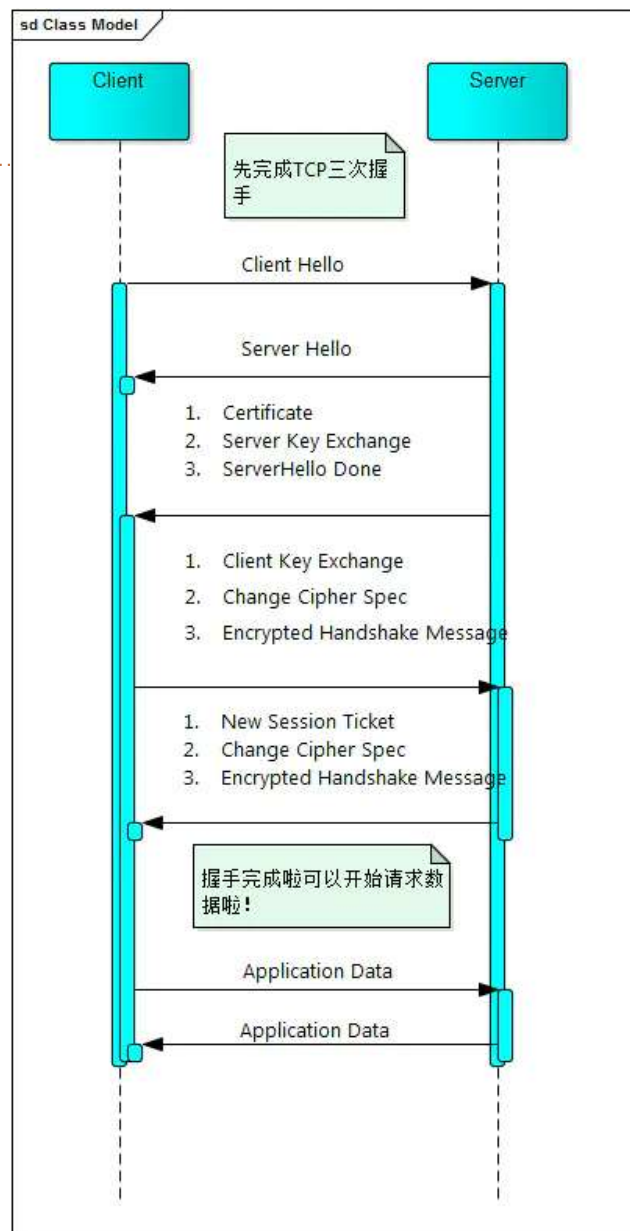
为什么需要证书？防止”中间人“攻击，同时可以为网站提供身份证明

HTTPS特点？对称加密/非对称加密、身份认证CA、数字证书和数据完整性验证
传输过程：

- ✓ 客户端发起 HTTPS 请求；
- ✓ 服务端返回证书；
- ✓ 客户端对证书进行验证，验证通过后本地生成用于改造对称加密算法的随机数，通过证书中的公钥对随机数进行加密传输到服务端；
- ✓ 服务端接收后通过私钥解密得到随机数；
- ✓ 之后的数据交互通过对称加密算法进行加解密。



完整的请求流程



任务3检查要求

开启Wireshark抓包

浏览器访问 xmu.edu.cn 的网站

在Wireshark中对浏览器和网站服务器互相发送的报文进行逐行分析

截图现场检查并提问

（主要检查TLS建立连接过程时的包信息，client Hello， server Hello等等）



任务4 ARP spoofing 与 中间人攻击(MITM)

地址解析协议（ARP，address resolution protocol）：

ARP协议的基本功能就是通过目标设备的IP地址，来查询目标设备的mac地址。

在局域网的任意一台主机中，都有一个ARP缓存表，里面保存本机已知的此局域网中各主机和路由器的IP地址和MAC地址的对照关系。

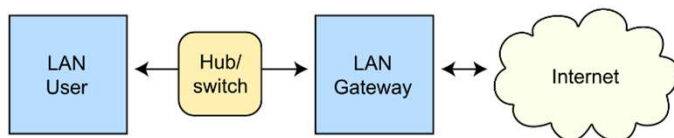
ARP缓存表的生命周期是有时限的（一般不超过20分钟）。

如果目标主机不在局域网内呢？

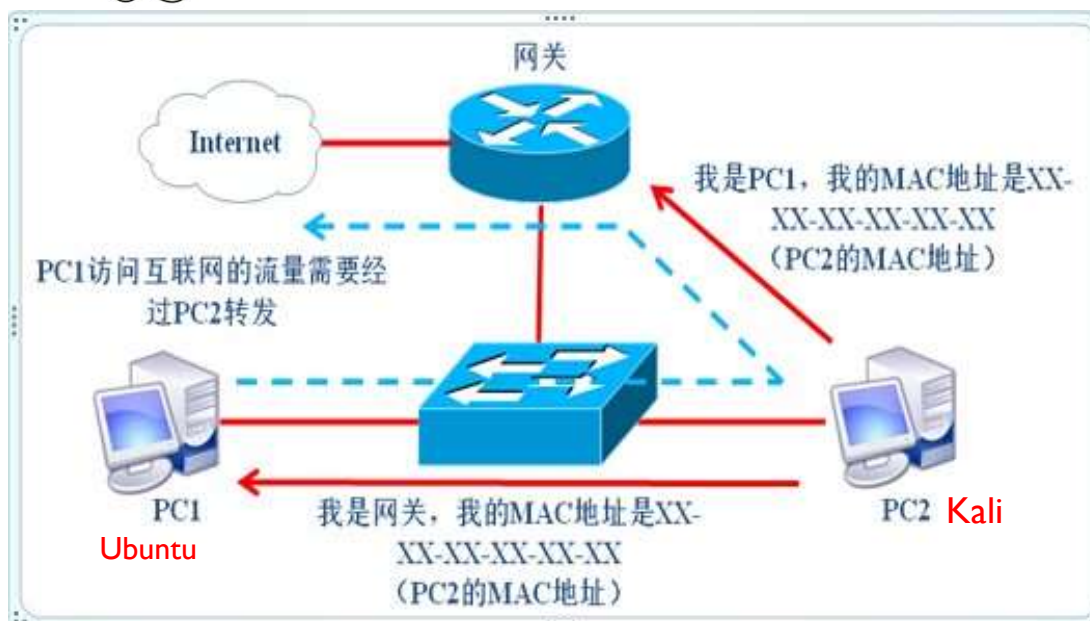
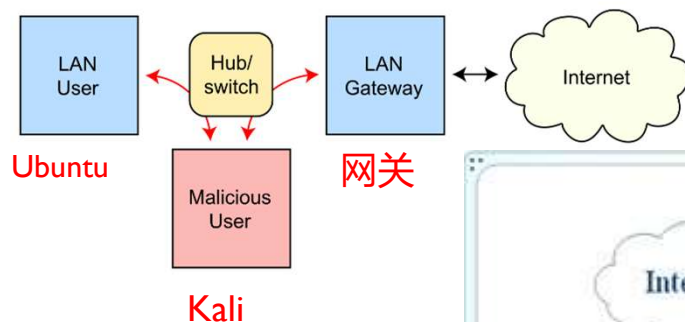


ARP 缓存投毒原理

Routing under normal operation



Routing subject to ARP cache poisoning



Kali Linux系统

镜像下载：【kali-linux-2023.3-vmware-amd64.7z】

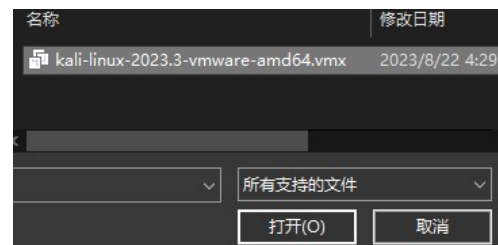
官方链接：

<http://old.kali.org/kali-images/kali-2023.3/kali-linux-2023.3-vmware-amd64.7z>

厦大云盘(速度快)：

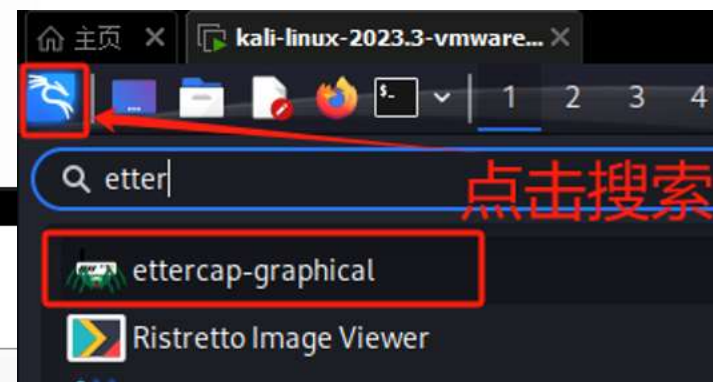
<https://box.xmu.edu.cn/share/7a0b1ea6f88e1426bd926bafbf>

解压后直接用Vmware打开，系统默认账号密码均为：**kali**



选用工具

Kali自带Ettercap, 无需下载



<https://www.ettercap-project.org>

任务4.1内容

- 使用Kali主机对Ubuntu主机发起中间人攻击
- 使用wireshark观察中间人攻击的过程
- 请分析ARP报文及通信方式等



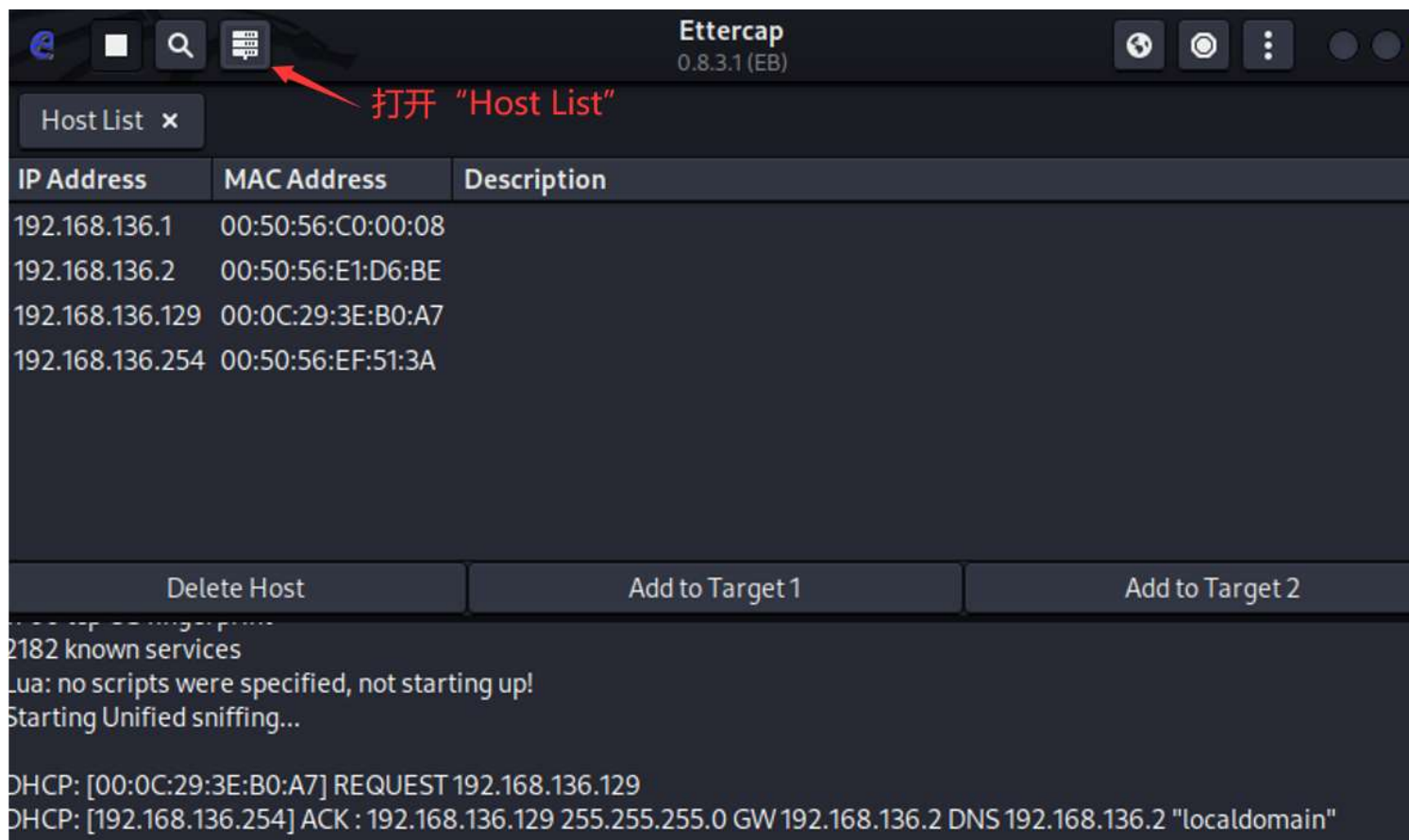
启动Ubuntu虚拟机，查看arp缓存信息

```
logan@ubuntu:~$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.136.129 netmask 255.255.255.0 broadcast 192.168.136.255
    inet6 fe80::d8ed:ff1a:c9cd:fae4 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:3e:b0:a7 txqueuelen 1000 (以太网)
    RX packets 7735 bytes 5657370 (5.6 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 4702 bytes 654397 (654.3 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (本地环回)
    RX packets 680 bytes 71143 (71.1 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 680 bytes 71143 (71.1 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

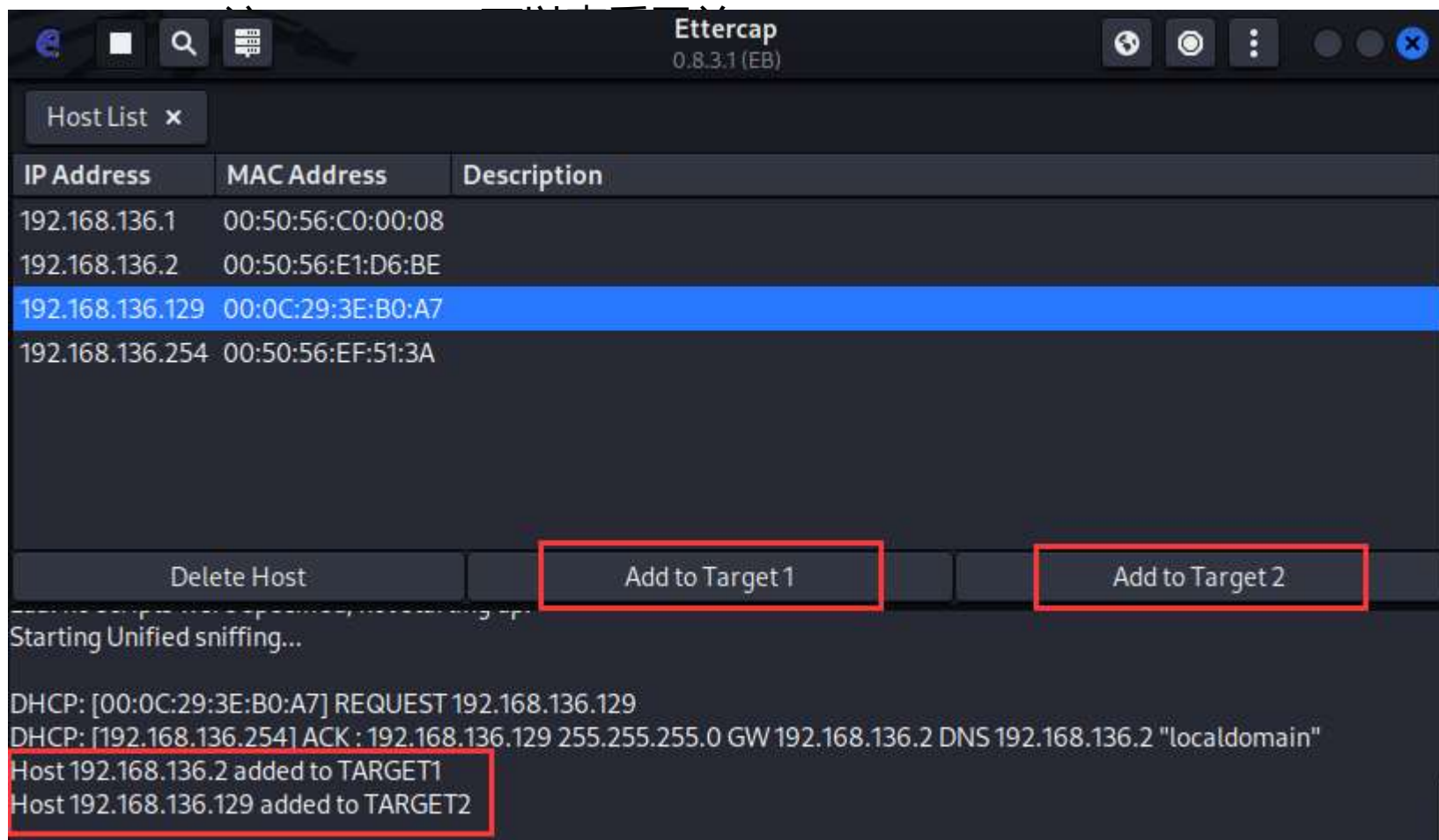
logan@ubuntu:~$ arp -a
? (192.168.136.130) 位于 00:0c:29:70:15:7d [ether] 在 ens33
? (192.168.136.254) 位于 00:50:56:ef:51:3a [ether] 在 ens33
_gateway (192.168.136.2) 位于 00:50:56:e1:d6:be [ether] 在 ens33
```

在Kali中启动Ettercap，并且扫描网络内主机

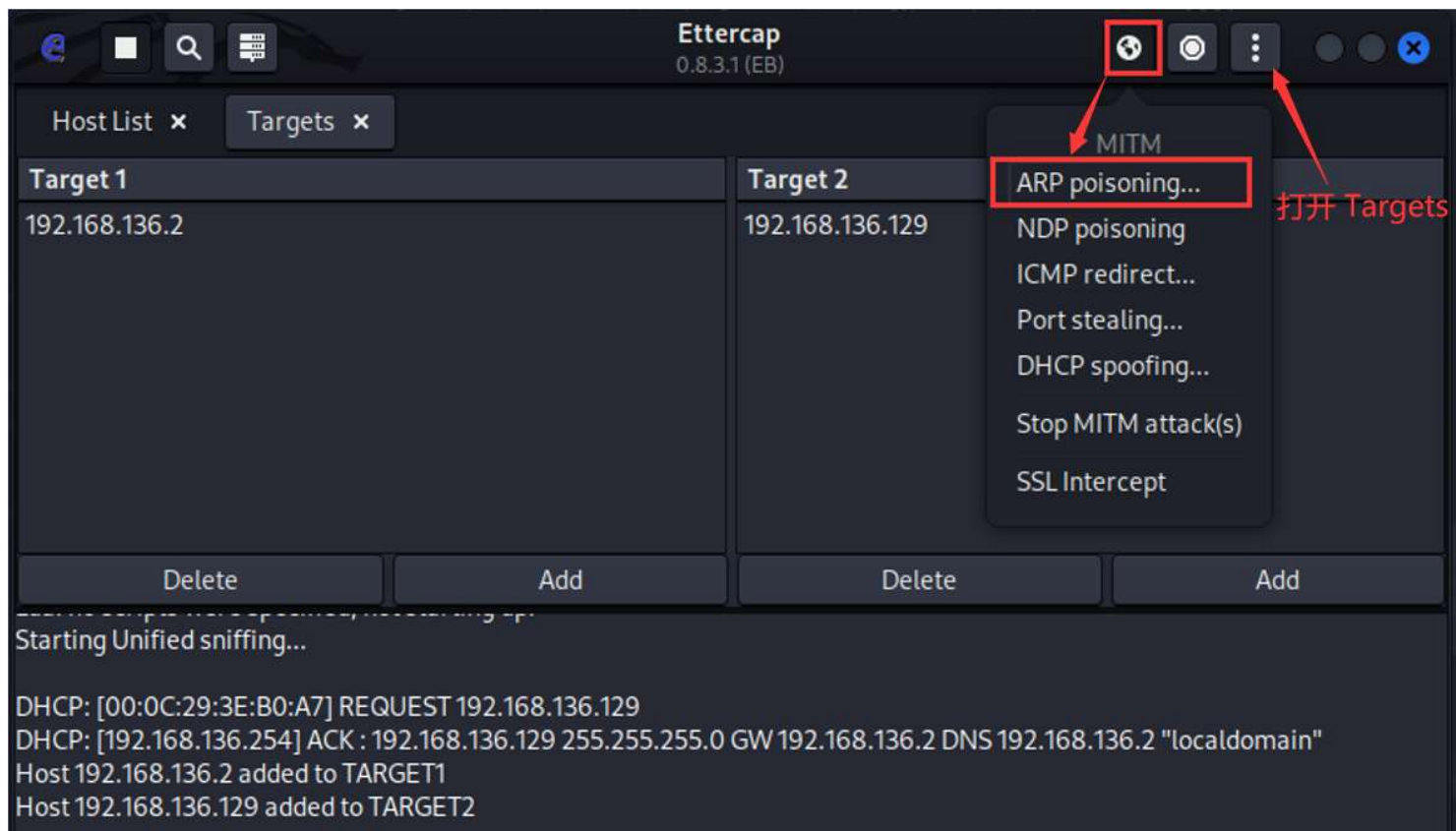


如果检测不到其他主机或者网关，可以在新的终端里ping一下他们的ip

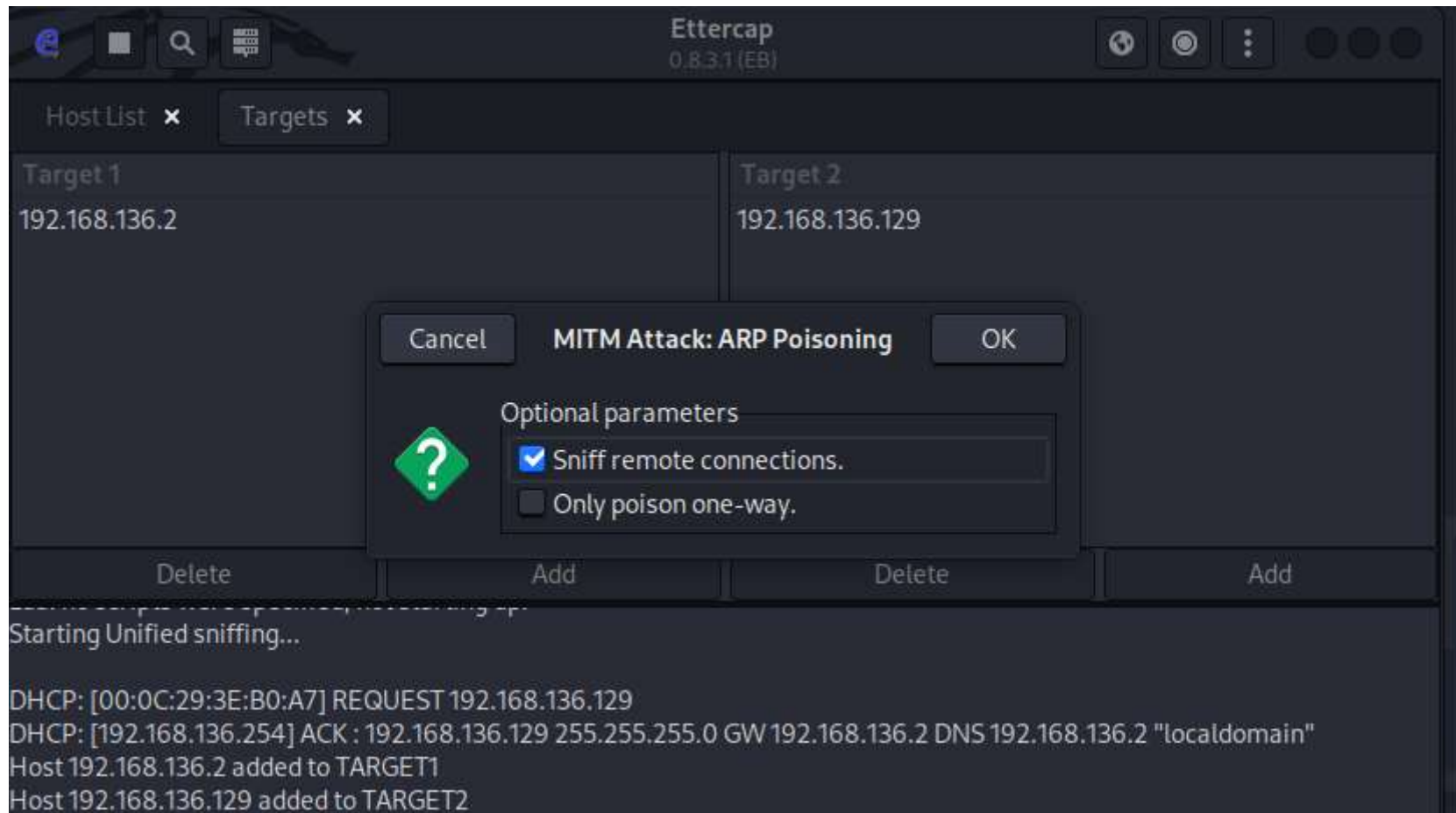
将目标主机和网关添加到targets中



选择ARP poisoning



选择Sniff remote connections



在Ubuntu中查看ARP缓存

发现网关的硬件地址出现变化

```
logan@ubuntu:~$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.136.129 netmask 255.255.255.0 broadcast 192.168.136.255
    inet6 fe80::d8ed:ff1a:c9cd:fae4 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:3e:b0:a7 txqueuelen 1000 (以太网)
    RX packets 7735 bytes 5657370 (5.6 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 4702 bytes 654397 (654.3 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (本地环回)
    RX packets 680 bytes 71143 (71.1 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 680 bytes 71143 (71.1 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

logan@ubuntu:~$ arp -a
? (192.168.136.130) 位于 00:0c:29:70:15:7d [ether] 在 ens33
? (192.168.136.254) 位于 00:50:56:ef:51:3a [ether] 在 ens33
gateway (192.168.136.2) 位于 00:50:56:e1:d6:be [ether] 在 ens33
logan@ubuntu:~$ arp -a
? (192.168.136.130) 位于 00:0c:29:70:15:7d [ether] 在 ens33
? (192.168.136.254) 位于 00:50:56:ef:51:3a [ether] 在 ens33
gateway (192.168.136.2) 位于 00:0c:29:70:15:7d [ether] 在 ens33
logan@ubuntu:~$
```

Hints

如果不成功请在Kali中开启转发后再重试、命令如下

\$echo "1" > /proc/sys/net/ipv4/ip_forward

如果出现权限问题，请尝试

echo 1 | sudo tee /proc/sys/net/ipv4/ip_forward

另外注意清理iptables中的所有规则：

sudo iptables -F



任务4.1 检查要求

- 成功发起中间人攻击
- 在Wireshark中观察分析攻击发起过程
- 请说明应当如何从Wireshark记录中分析出ARP攻击（选作）



任务4.2 捕获网络服务内容和分析（选做）

- ▣ 使用Ubuntu主机访问http目标网站，并分析该网络服务过程中输入/输出的敏感信息
- ▣ 任务要求：
 - 从Ettercap的Connections中分析Ubuntu使用网络服务过程中，输入的敏感信息（用户名、密码、搜索记录等）。
 - 请指出这信息的位置以及内容。
 - 推荐HTTP网址：<http://www.7k7k.com/>



在Ettercap的Connections中查看捕获的通信

