ARP 欺骗与 IP 假冒

1. 实验目的

通过实验了解并掌握同一局域网中两台计算机的通信过程及期间使用的相关协议内容。学会在同一 SSID 下，嗅探获取指定网段的 IP 和 MAC 地址，并通过 ARP 欺骗和 IP 假冒来对客户机进行攻击。

2. 实验分工

在实验一中我负责搭建基于 VMware 的实验环境，石英昊对 wireshark 获取的包进行分析。实验二中，我们两个人用各自的电脑进行实验。我的 ubuntu 作为攻击机，而石英昊的 MAC 则正常登录 TUNET。石英昊负责对同一 SSID 下的 IP 及其对应的 Mac 地址进行嗅探，而我则负责用获取到的 IP 和 Mac 进行 IP 假冒。

3. 实验过程

在虚拟机中使用 ifconfig 查看相应的 ip 地址及子网掩码等信息。最开始，两个虚拟机的 ip 地址分别为 192.168.158.128 和 192.168.158.129，子网掩码均为 255.255.255.0。

1> 使用 ifconfig ens33 192.168.158.23 netmask 255.255.255.224 命令将原始 ip 为 192.168.158.129 的虚拟机进行修改。两者网关均为 192.168.158.2，保持不变。

修改之后的 ip 信息和网关如下图。

```
yu@ubuntu: ~
yu@ubuntu:~$ ifconfig
ens33     Link encap:Ethernet  HWaddr 00:0c:29:64:45:42
          inet addr:192.168.158.23  Bcast:192.168.158.31  Mask:255.255.255.224
          inet6 addr: fe80::d162:60d0:c0f2:8075/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:606 errors:0 dropped:0 overruns:0 frame:0
          TX packets:677 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:77146 (77.1 KB)  TX bytes:85051 (85.0 KB)
```

网卡 ens33 对应的 ip 信息

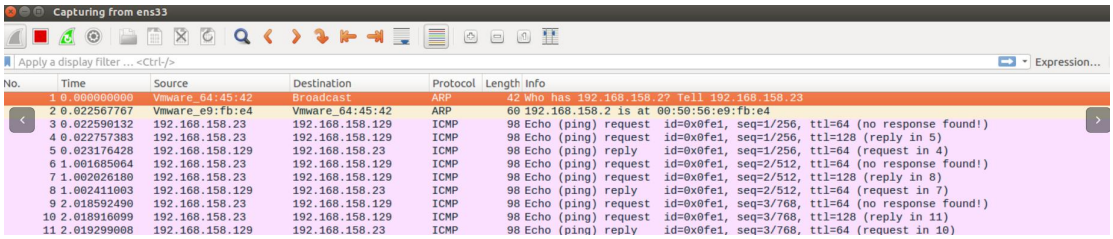
```
yu@ubuntu:~$ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         192.168.158.2   0.0.0.0         UG    100    0        0 ens33
link-local      *               255.255.0.0     U     1000   0        0 ens33
192.168.158.0   *               255.255.255.224 U     100    0        0 ens33
```

第一个为默认网关

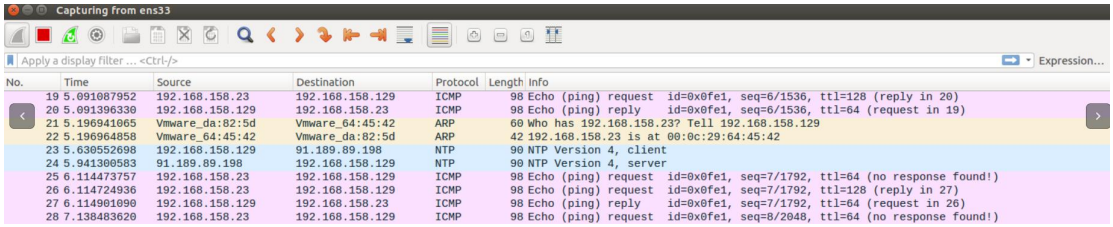
假设修改的虚拟机为 B，未修改的为 A。修改完后的一段时间内，B 无法 ping 通 A，B 也无法 ping 通外网。我们小组和其他小组讨论认为，这个阶段虚拟机应该在重启网络服务，相关网络配置尚未完成，所以造成了 B 短暂地无法 ping 通 A。

2> 当 B 能正常 ping 通 A，我们使用 wireshark 对数据进行了抓包分析。

wireshark 抓包分析如下图：



第一次 arp 广播询问网关 mac 地址



第二次 arp 广播询问目标 ip 对应的 mac 地址

从 arp 的抓包分析及子网掩码的设置我们可知，B 可以 ping 通 A 的原因。由于 B 修改了子网掩码的大小，导致 B 第一次检查时，认为 A 和其不属于同一子网。所以给 A 的数据包需要借由网关进行转发。由于刚开始 arp 表项为空（arp -d ip 命令可删除对应 ip 的表项），所以进行第一次 arp 广播，获取网关的 mac 地址。

数据包到达网关之后，网关首先询问自己的子网，判断对应的 ip 是否在子网之中。所以，有了第二次 arp 广播。这次网关获取了 A 的 mac 地址，则 B 通过 ICMP 协议 ping 通 A。

抓包分析前，我们使用 arp -a 和 arp -d ip 的命令将 A、B 两者的 arp 表项全部清空。

IP 假冒实验，我们使用一个 linux 主机和 MAC 主机进行实验，linux 主机不登录 TUNET，而 MAC 主机连接同一个 Tsinghua WiFi，进行正常的 TUNET 登录。

1> Ubuntu 中使用 ifconfig 获取当前主机的相关信息

```
wlp2s0    Link encap:以太网  硬件地址 b4:6d:83:0e:99:e4
          inet 地址:183.172.216.78  广播:183.172.223.255  掩码:255.255.248.0
          inet6 地址: 2402:f000:2:d801:7c77:378a:c5a4:d978/64 Scope:Global
          inet6 地址: 2402:f000:2:d801:56b0:68f8:1466:7484/64 Scope:Global
          inet6 地址: fe80::8c86:612a:7ab0:42f6/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  跃点数:1
          接收数据包:17426 错误:0 丢弃:0 过载:0 帧数:0
          发送数据包:15364 错误:0 丢弃:0 过载:0 载波:0
          碰撞:0 发送队列长度:1000
          接收字节:13151715 (13.1 MB)  发送字节:2377101 (2.3 MB)
```

最初 wlp2s0 网卡的相关 ip 信息

2> 使用 nmap -sP 进行指定网段下的 ip 嗅探

nmap -sP 183.172.216.78/20 对指定网段内的 ip 进行嗅探。

```
yu (master_nightmode *) ~ $ nmap -sP 183.172.216.78/20

Starting Nmap 7.01 ( https://nmap.org ) at 2018-10-06 15:43 CST
Nmap scan report for 183.172.208.1
Host is up (0.0061s latency).
Nmap scan report for 183.172.208.12
Host is up (0.0064s latency).
Nmap scan report for 183.172.208.13
Host is up (0.0036s latency).
Nmap scan report for 183.172.208.14
Host is up (0.039s latency).
Nmap scan report for 183.172.208.16
Host is up (0.0063s latency).
Nmap scan report for 183.172.208.19
Host is up (0.0062s latency).
Nmap scan report for 183.172.208.22
Host is up (0.0035s latency).
```

nmap 嗅探结果

3> 指定 ip 的 mac 地址的获取

实验要求是在同一个 wifi 路由器下，所以可以直接在命令行中 ping ip，ping 通之后使用 arp -a 即可获取目标 ip 对应的 mac 地址。

至此，我们获取到了 MAC 主机的 ip 和 mac 地址。

4> 修改 linux 主机的 ip 及 mac 地址，进行 ip 假冒

修改 ip 及 mac 地址命令如下：

ifconfig wlp2s0 183.172.240.86 netmask 255.255.248.0　修改 ip 地址和子网掩码

ifconfig wlp2s0 down　关闭 wlp2s0 的服务

Ifconfig wlp2s0 hw ether 60:f8:1d:c0:41:bc　修改 linux 主机的 mac 地址

ifconfig wlp2s0 up　重启 wlp2s0 的服务

```
root@chaihj15:/home/yu# ifconfig wlp2s0 down
root@chaihj15:/home/yu# ifconfig wlp2s0 183.172.240.86 netmask 255.255.248.0
root@chaihj15:/home/yu# ifconfig wlp2s0 hw ether 60:f8:1d:c0:41:bc
SIOCSIFHWADDR: 设备或资源忙 - you may need to down the interface
root@chaihj15:/home/yu# ifconfig wlp2s0 down
root@chaihj15:/home/yu# ifconfig wlp2s0 hw ether 60:f8:1d:c0:41:bc
root@chaihj15:/home/yu# ifconfig wlp2s0 up
```

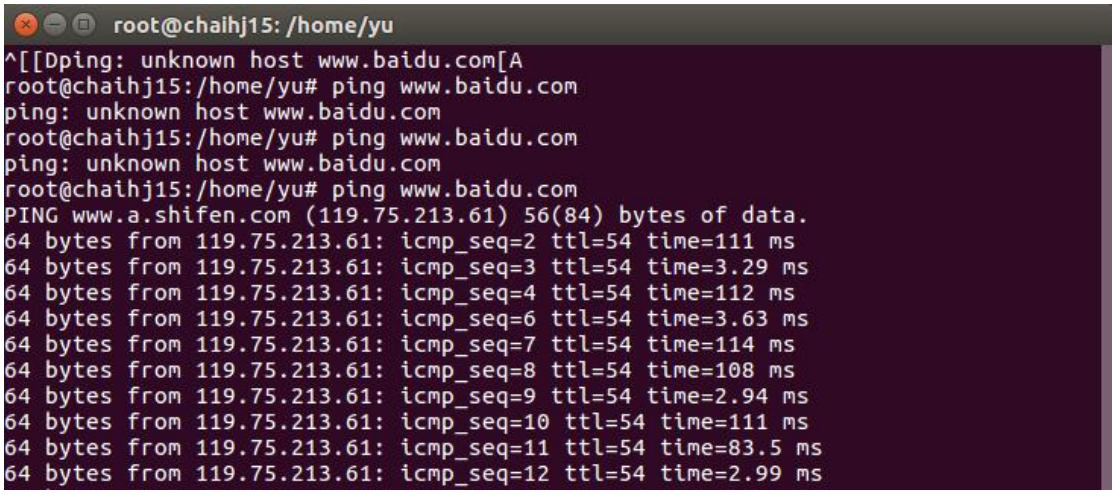
修改 ip 及 mac 操作

```
root@chaihj15: /home/yu
wlp2s0    Link encap:以太网  硬件地址 60:f8:1d:c0:41:bc
          inet 地址:183.172.240.86  广播:183.172.255.255  掩码:255.255.0.0
          UP BROADCAST MULTICAST  MTU:1500  跃点数:1
          接收数据包:66074 错误:0 丢弃:0 过载:0 帧数:0
          发送数据包:133316 错误:0 丢弃:0 过载:0 载波:0
          碰撞:0 发送队列长度:1000
          接收字节:51993302 (51.9 MB)  发送字节:15866723 (15.8 MB)

root@chaihj15:/home/yu# ping www.baidu.com
ping: unknown host www.baidu.com
```

linux 主机修改后的 ip 信息

修改完成之后，同实验一，linux 主机进行网络配置较慢，在一段时间内无法 ping 通外网。但之后就可以正常 ping 外网。



ip 假冒后进行 ping 操作

5> 假冒 ip 后，进行流量劫持

如上图所示，linux 主机不断用假冒的 ip 和 mac 地址进行 ping 操作，在实际过程中，MAC 主机的同学联网受限，在我不断 ping 的过程中，MAC 主机的同学甚至无法登录网页版微信。当我停止 ping 操作时，其又可以正常联网，只不过网络速度较慢。

与其他小组交流后得知网络交换机通过逆向学习的算法实时更新 mac 地址对应的端口，ip 地址和 mac 地址均相同的 linux 主机和 MAC 主机处于竞争状态，数据会转发给上一次发送数据的主机。

4. 实验总结及体会

通过两个实验，我们学会了同一局域网内两台主机是如何相互访问传输数据。对于 arp 协议也有了更深刻的印象。在实验中，我们也知道了如何去手动修改主机的 ip 地址和子网掩码。第二个实验中，我们学会了如何利用工具去进行 ip 嗅探和 mac 地址的获取，对于假冒 ip 的流量劫持有了更清晰的认知。

通过本次实验，让我们对于网络协议有了更深的了解，同时让我们对网络安全研究的兴趣更加浓厚。