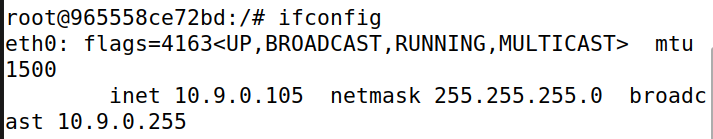
**ARP Cache Poisoning Attack Lab**

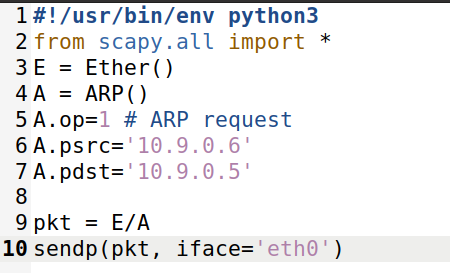
**Task 1: ARP Cache Poisoning**

1. **using ARP request**

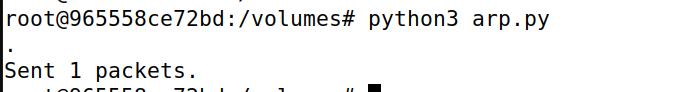
在主机 M 上，构造一个 ARP 请求包并发送给主机 A。查看A的ARP缓存，看M的MAC地址是否映射到B的IP地址。

编写代码

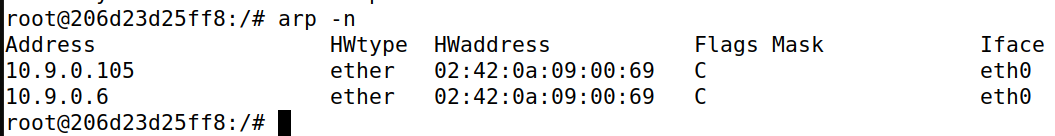




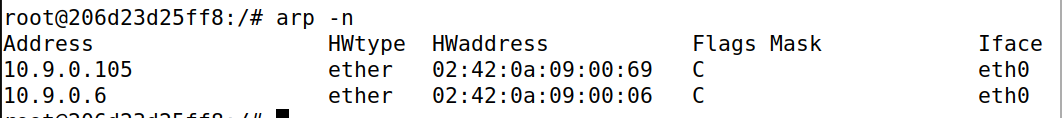
在M主机上运行脚本



在A主机上查看ARP缓存，发现M的MAC地址被映射到B的IP地址



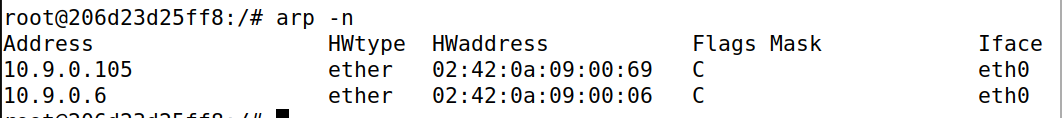
对比之前，可以发现B在A中的MAC地址被修改了



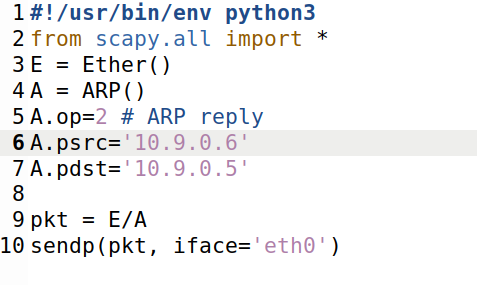
**B．using ARP reply**

在主机M上构造一个ARP回复包发送给主机A，查看A的ARP缓存，看M的MAC地址是否映射到B的IP地址。 尝试针对两种不同场景进行攻击：

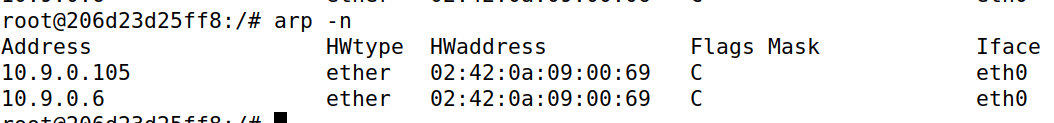
– 场景 1：B 的 IP 已经在 A 的缓存中。



编写arp reply脚本

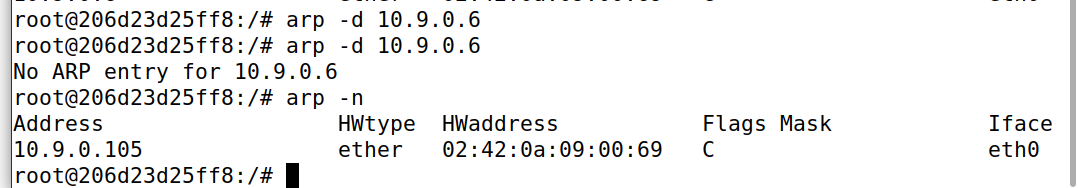


运行脚本之后，发现在A中B的ip地址已经被修改

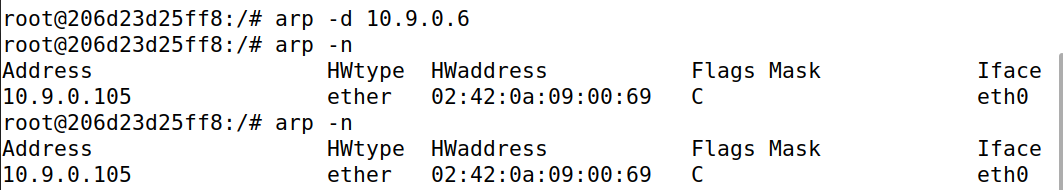


– 场景 2：B 的 IP 不在 A 的缓存中。

清除B在A中的ARP缓存



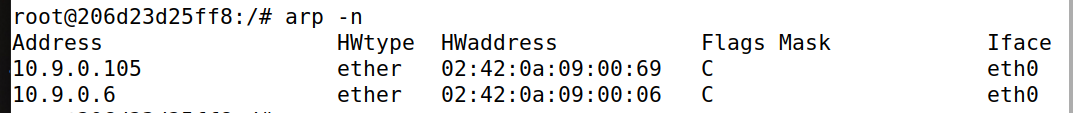
结果发现不能进行修改



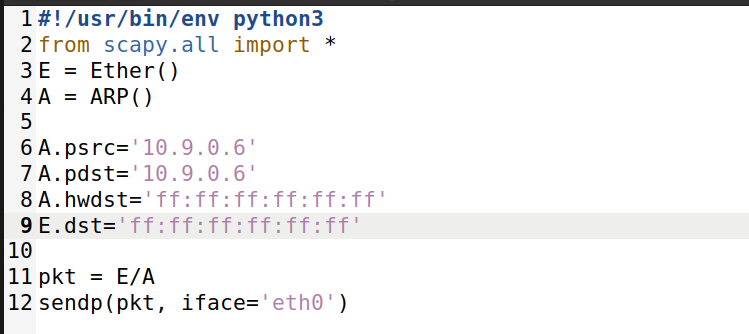
**C．using ARP gratuitous message**

在主机 M 上构造一个 ARP 免费报文，并将 M 的 MAC 地址映射到 B 的 IP 地址。 请在与任务 1.B 中描述的相同的两种情况下发起攻击。ARP免费报文是一种特殊的ARP请求报文。 当主机需要更新所有其他机器的 ARP 缓存上的过时信息时使用它。 免费 ARP 报文具有以下特点： – 源 IP 地址和目的 IP 地址相同，即发送免费 ARP 的主机的 IP 地址。– ARP 头和以太网头中的目的 MAC 地址都是广播 MAC 地址（ff:ff:ff:ff:ff:ff）。– 预计不会收到回复。

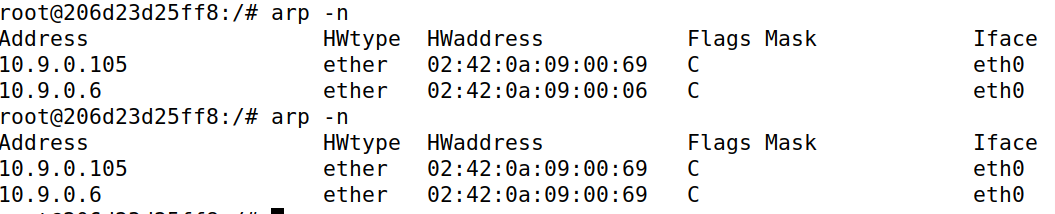
– 场景 1：B 的 IP 已经在 A 的缓存中。



修改脚本如下：

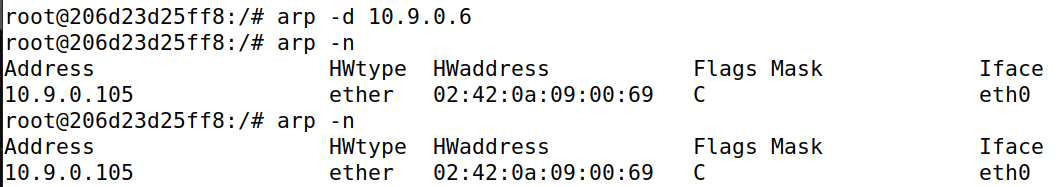


运行脚本，查看结果，发现也是成功修改了



– 场景 2：B 的 IP 不在 A 的缓存中。

同样，运行脚本，无法修改

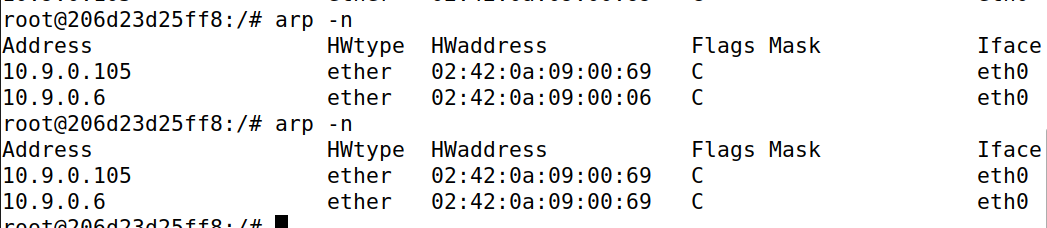


**Task 2: MITM Attack on Telnet using ARP Cache Poisoning**

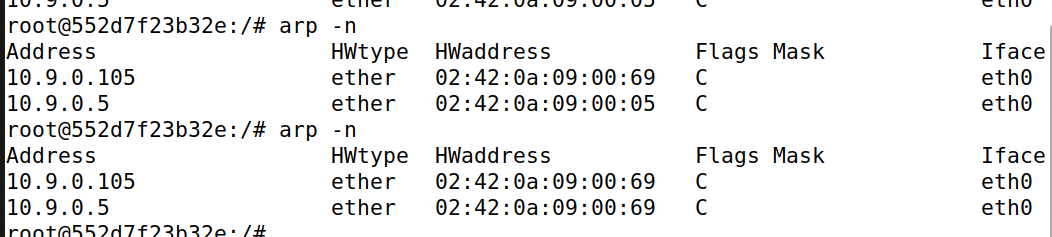
**Step 1 (Launch the ARP cache poisoning attack).**

**对A,B分别发动ARP cache poisoning attack**

**A中B的IP地址被定向到M的MAC地址**

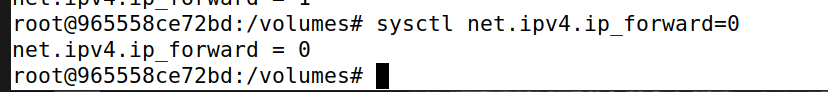


**B中A的IP地址被定向到M的MAC地址**

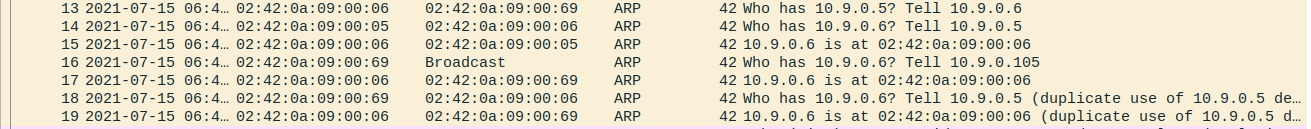


**Step 2(Testing)**

**关闭ip\_forward**

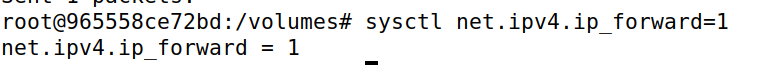


**使用wireshark抓取数据包，发现ARP报错，指示多个IP使用了同一个MAC**

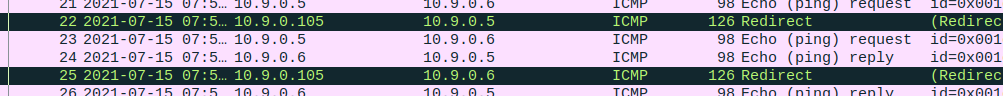


**Step 3 (Turn on IP forwarding).**

**重新打开ip forwarding**



**此时中间人会转发两台主机间的数据包，能够收到ping的回应了**

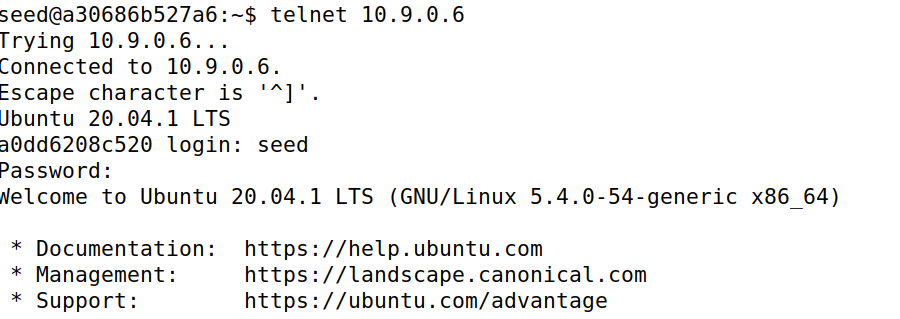


**Step 4 (Launch the MITM attack).**

**首先打开ip\_forward**



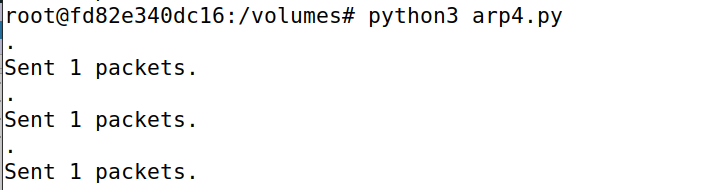
**建立telnet连接**



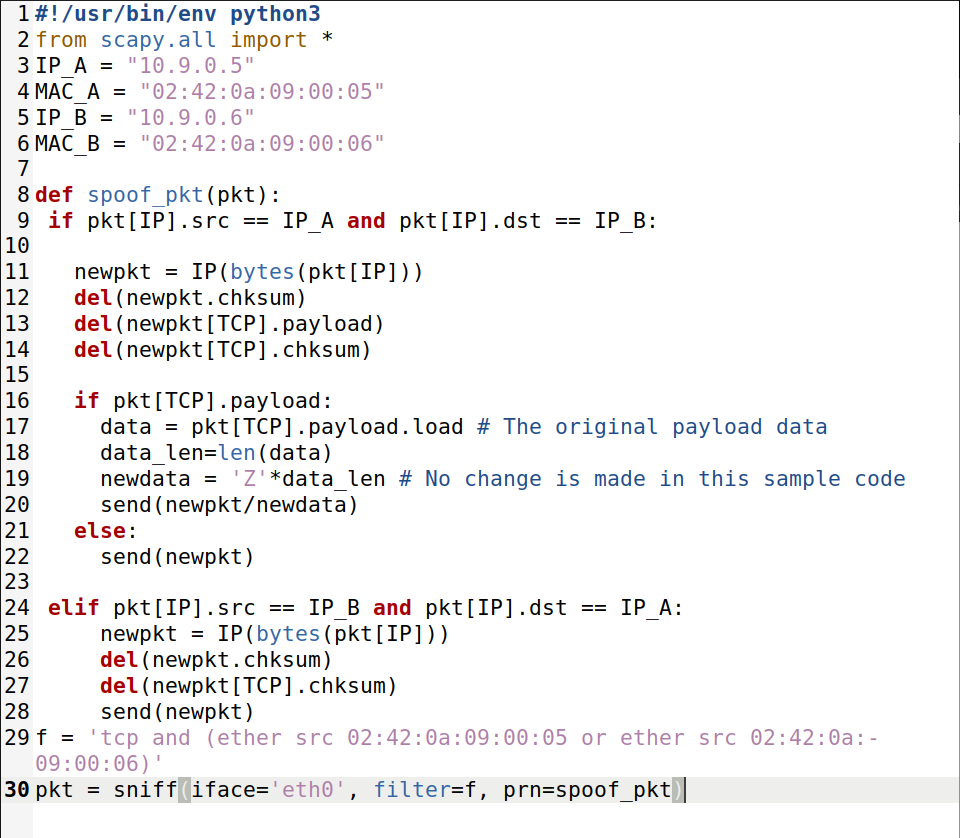
**返回，关闭ip forward**



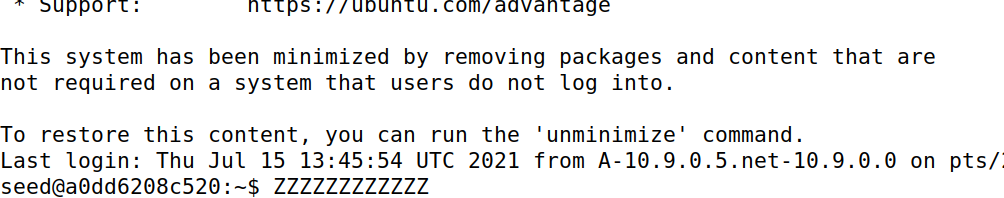
**此时，发动arp缓存中毒攻击**



**攻击成功后，在此基础上继续发动中间人攻击，代码如下：**

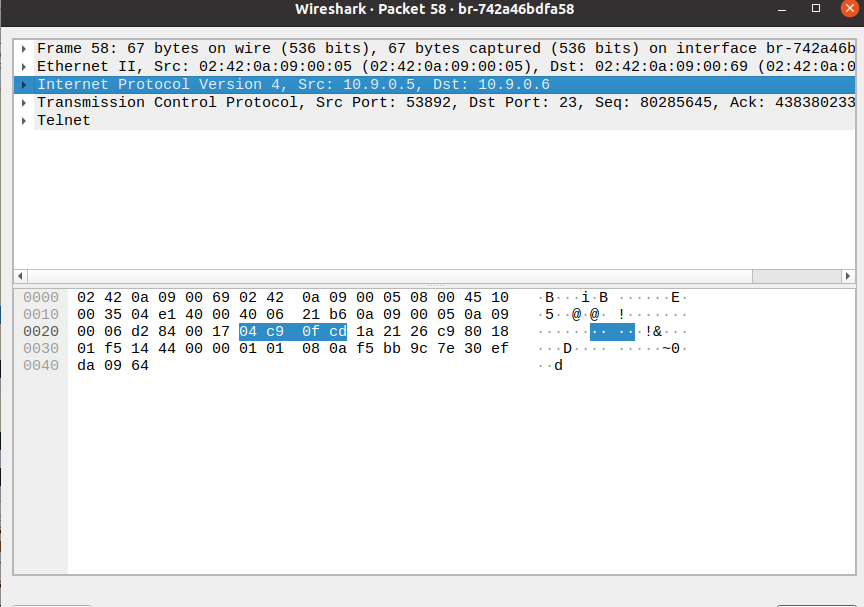


**此时再进入主机A 10.9.0.5效果就是，无论输入什么，都会出来ZZZZZ**

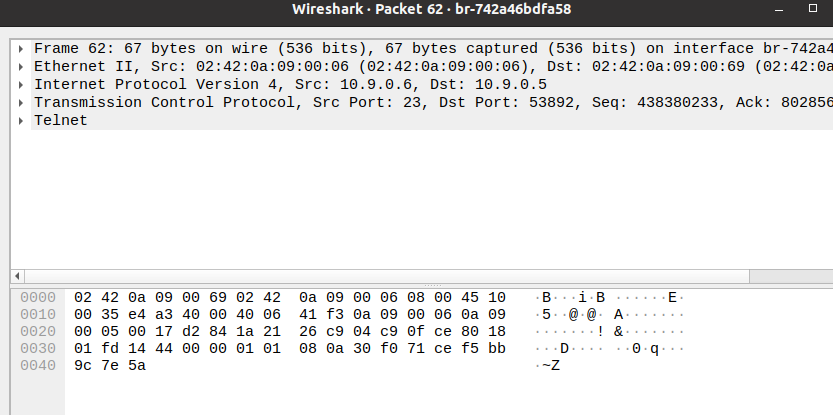


**从抓到的包来看**

**A发向M的是d**

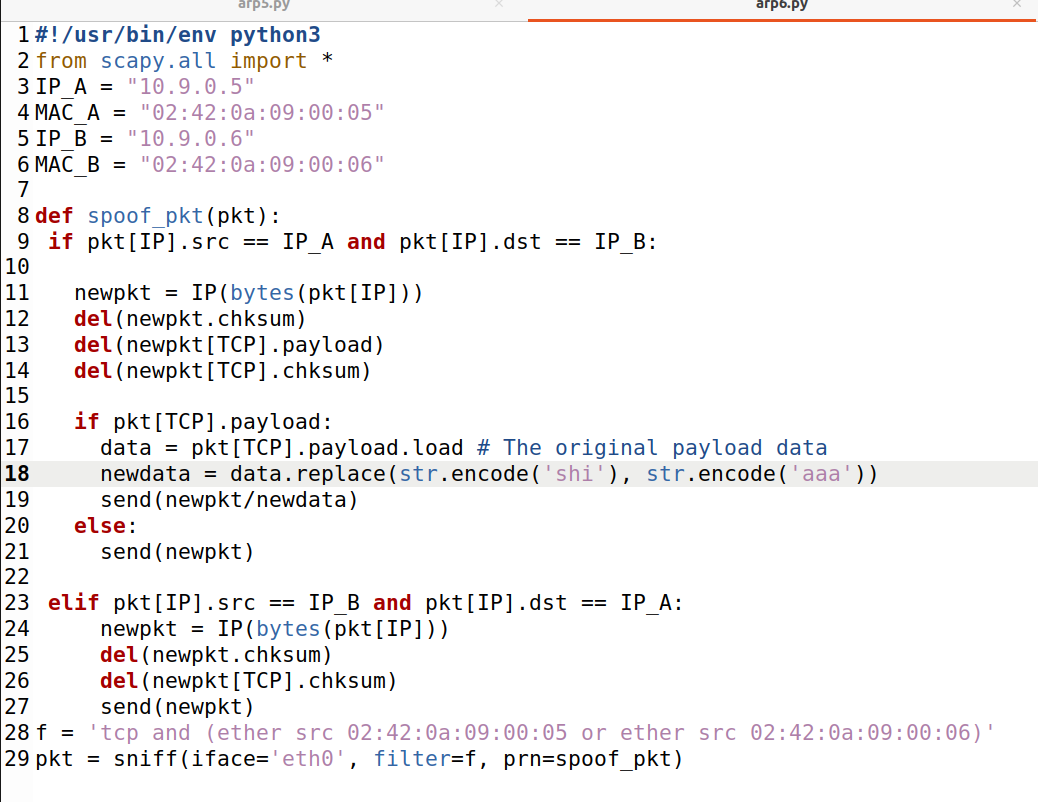


**B发向A的是Z**



**Task 3: MITM Attack on Netcat using ARP Cache Poisoning**

**与Task2类似，就是把脚本代码改一下，将shi改为aaa**



**与Task 2类似，只不过将telnet连接改为nc，运行结果如下，攻击成功。**

