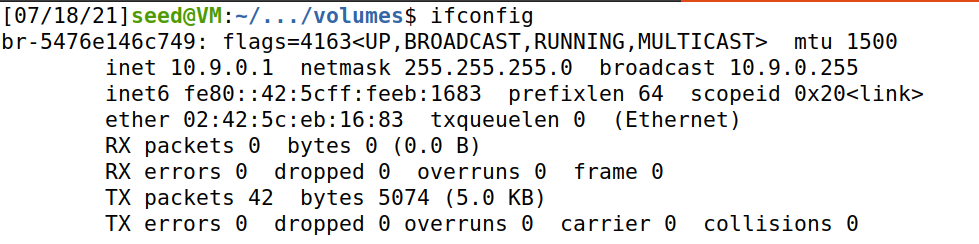
**ARP Cache Poisoning Attack Lab**

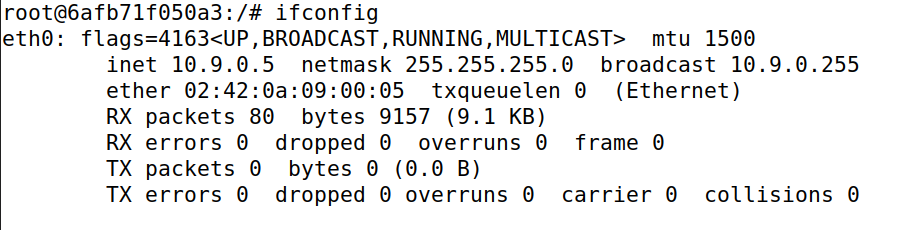
**57118113 蔡义涵**

**Task 1: ARP Cache Poisoning**

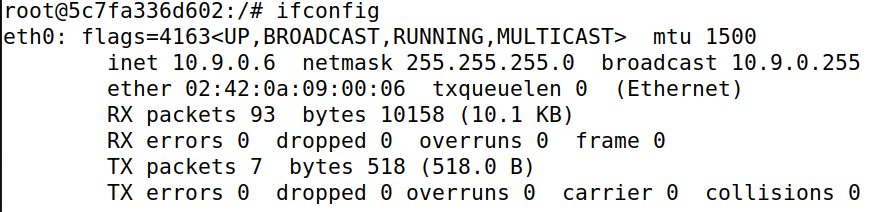
首先查看docker的网卡信息：



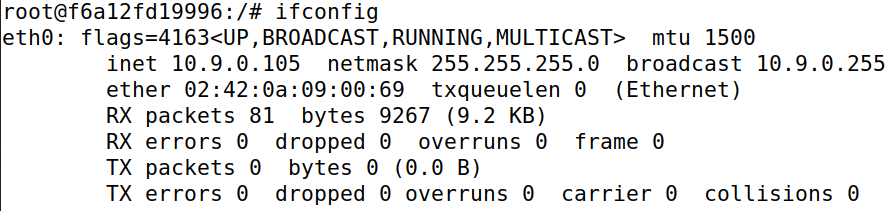
A-10.9.0.5



B-10.9.0.6



M-10.9.0.105

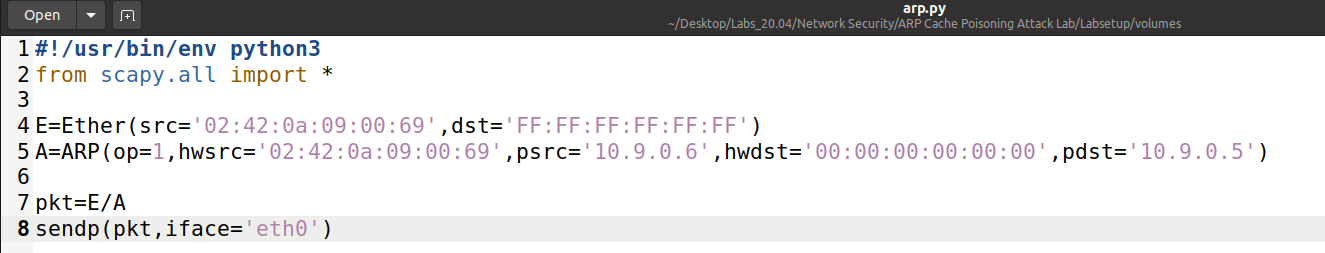


**Task 1.A (using ARP request)**

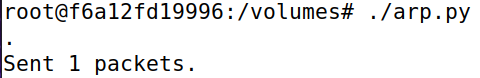
查看A-10.9.0.5上的ARP缓存



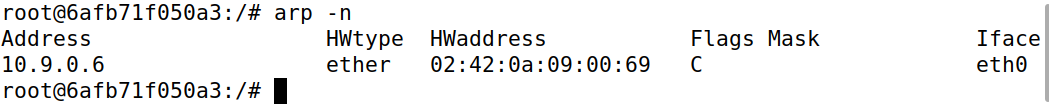
根据网卡信息完成arp.py程序



在M-10.9.0.105上执行上述程序

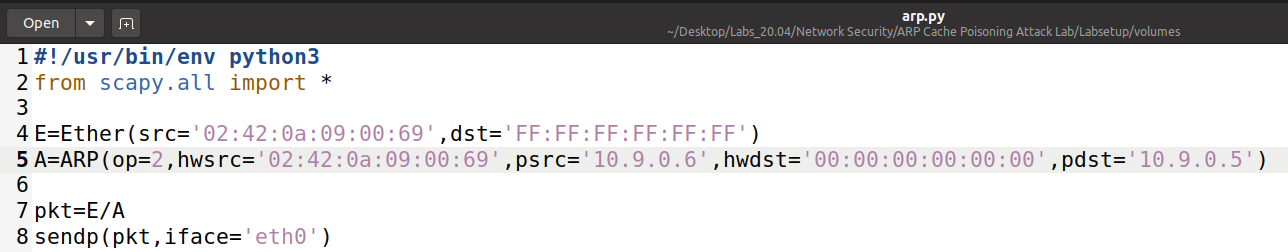


查看A-10.9.0.5上的ARP缓存，发现攻击成功。



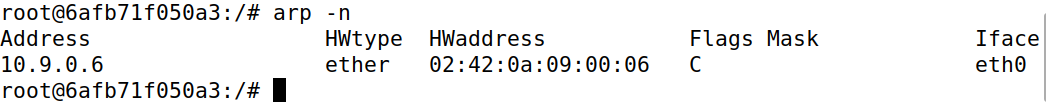
**Task 1.B (using ARP reply)**

修改arp.py如下：

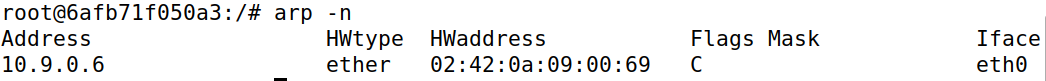


**Scenario 1: B’s IP is already in A’s cache**

查看A-10.9.0.5上的ARP缓存，此时host B对应的MAC地址为真实地址。

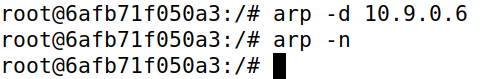


执行上述程序，结果如下：arp缓存被成功篡改。



**Scenario 2: B’s IP is not in A’s cache**

清空A上的ARP缓存表



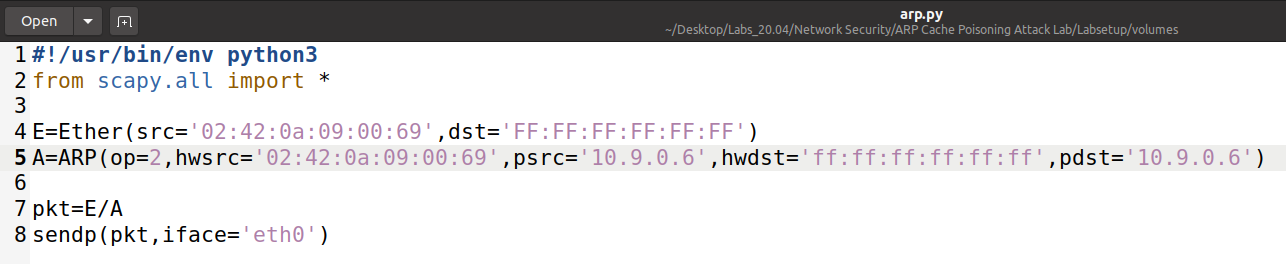
再次执行上述程序，结果如下：



发现攻击失败。

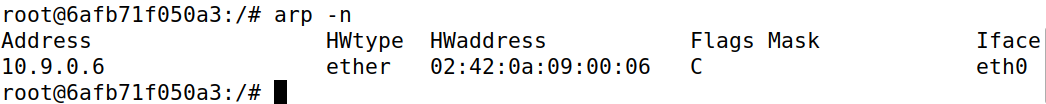
**Task 1.C (using ARP gratuitous message)**

修改arp.py程序为：

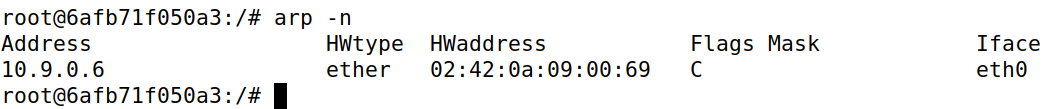


**Scenario 1: B’s IP is already in A’s cache**

查看A-10.9.0.5上的ARP缓存，此时host B对应的MAC地址为真实地址。



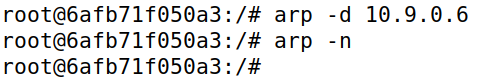
在M-10.9.0.105上执行上述程序，结果如下：



发现攻击成功。

**Scenario 2: B’s IP is not in A’s cache**

清空A-10.9.0.5上的ARP缓存



再次执行上述程序，打印结果：

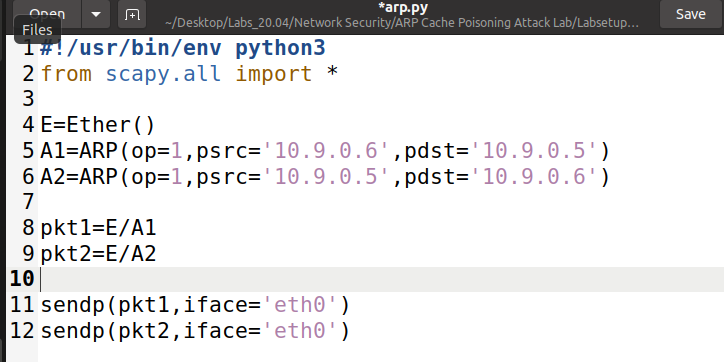


攻击失败。

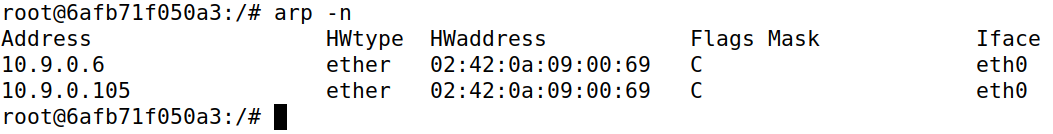
**Task 2: MITM Attack on Telnet using ARP Cache Poisoning**

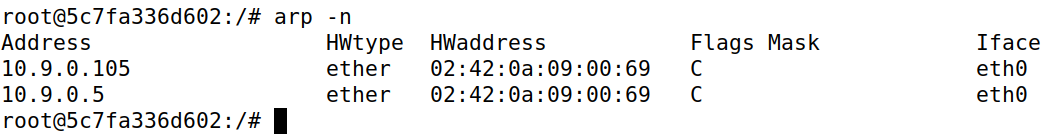
**Step 1 (Launch the ARP cache poisoning attack)**

首先构造ARP缓存攻击：



执行arp.py，查看A和B的arp缓存表

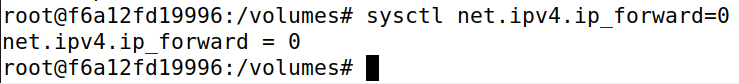




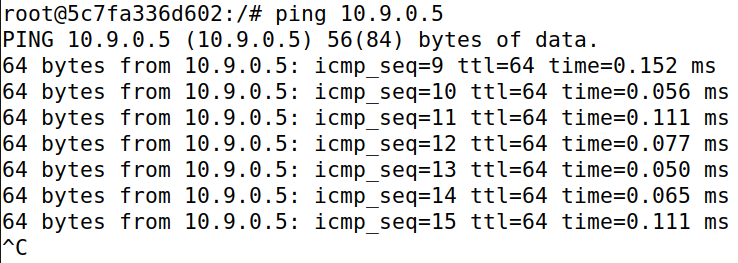
发现攻击成功。

**Step 2 (Testing)**

将Host-M上的ip\_forward设置为0：

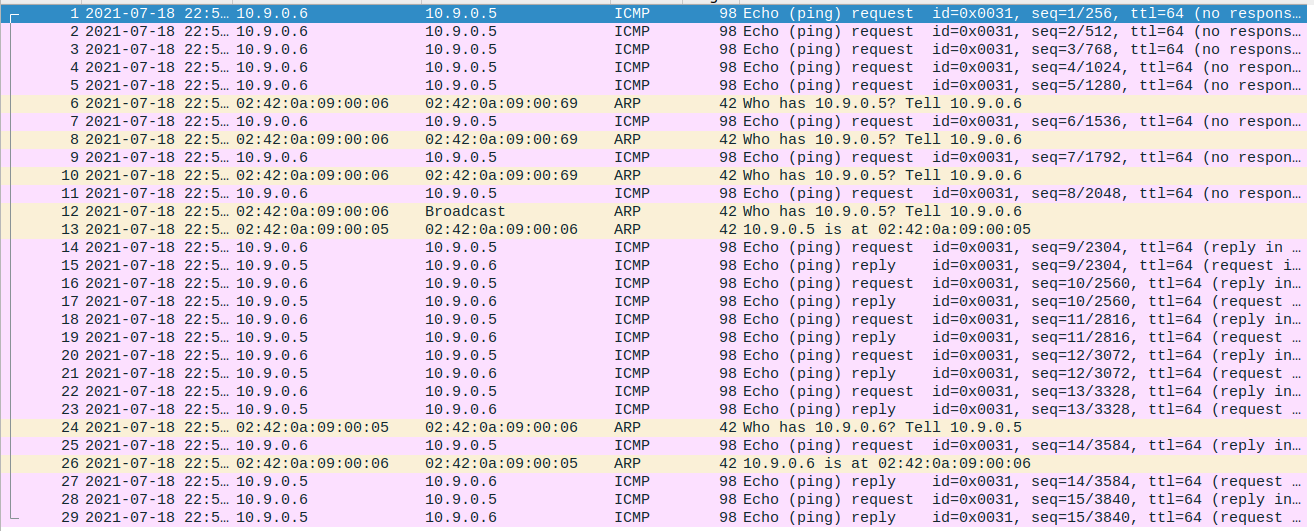


打开wireshark进行监听，同时在B上ping A



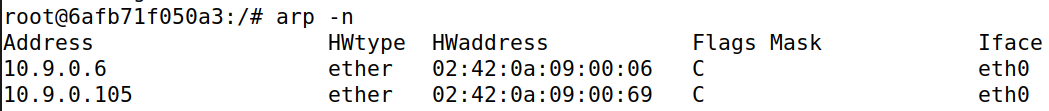
Ping时观察到，经过一段延时后，ping命令才成功执行。

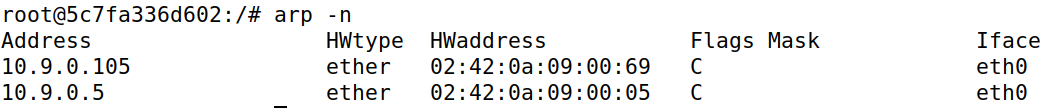
查看wireshark：



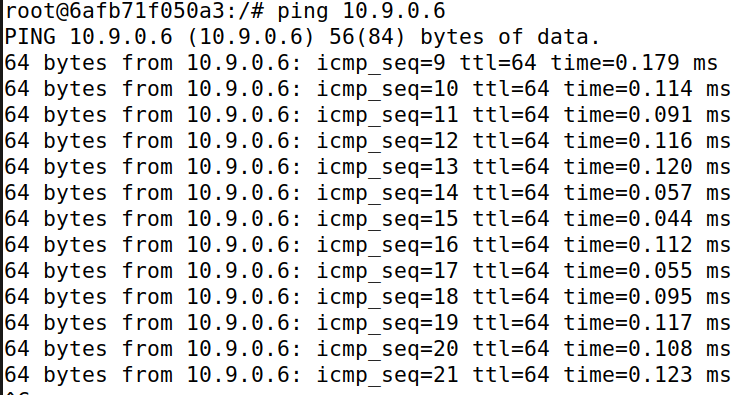
可以观察到，一开始B发出的报文，由于arp缓存表攻击的执行得不到回复，四次尝试后向其他MAC地址单播了ARP请求报文。三次尝试后广播发出了ARP请求报文，随后得到了A的arp应答报文。

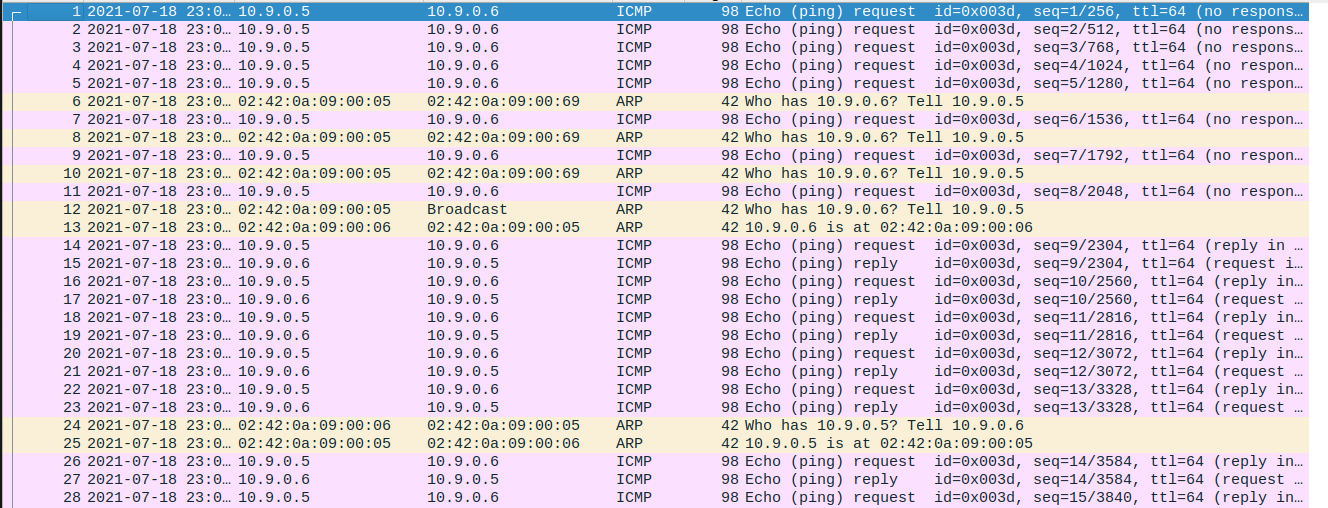
查看A和B的arp缓存表，结果发现被修改回正确的MAC地址。





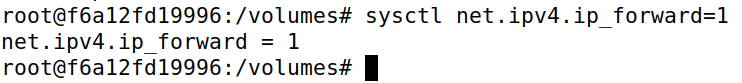
在A上ping B的结果类似：





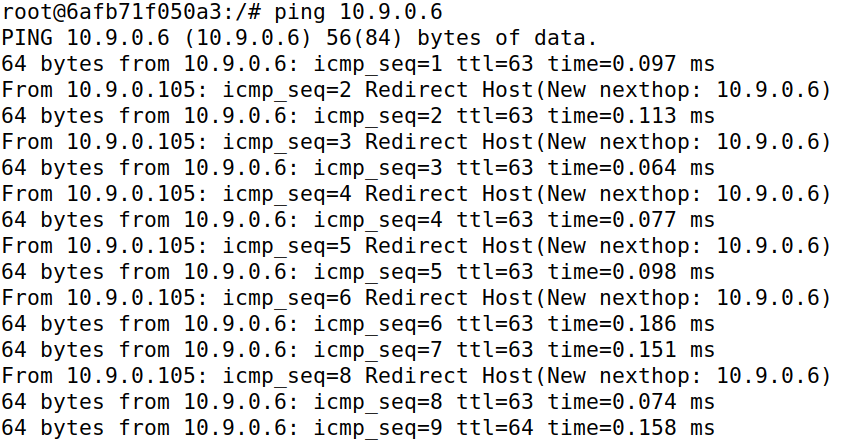
**Step 3 (Turn on IP forwarding)**

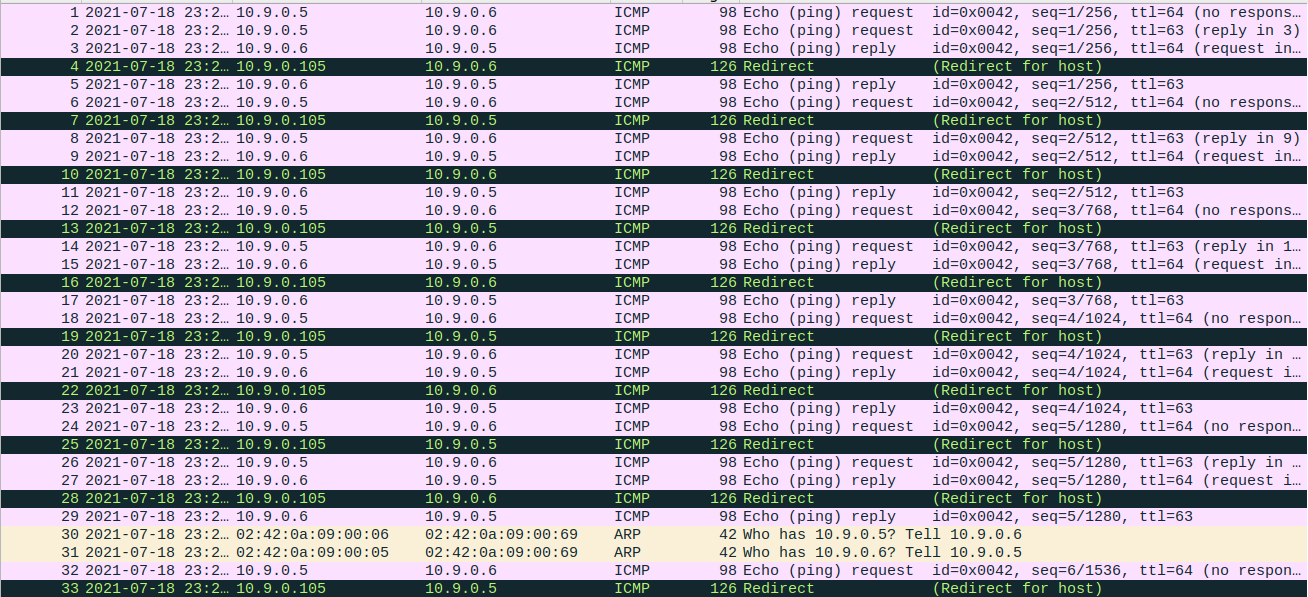
设置host M的ip\_forward为1：



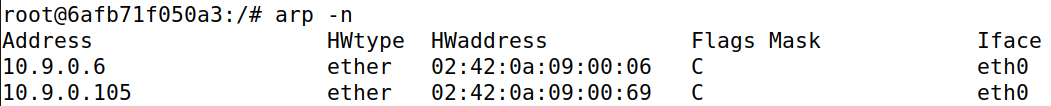
重新执行Step1、Step2过程

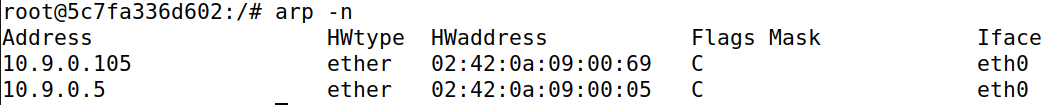
结果如下：





再次查看A和B的arp缓存表，发现已被修改

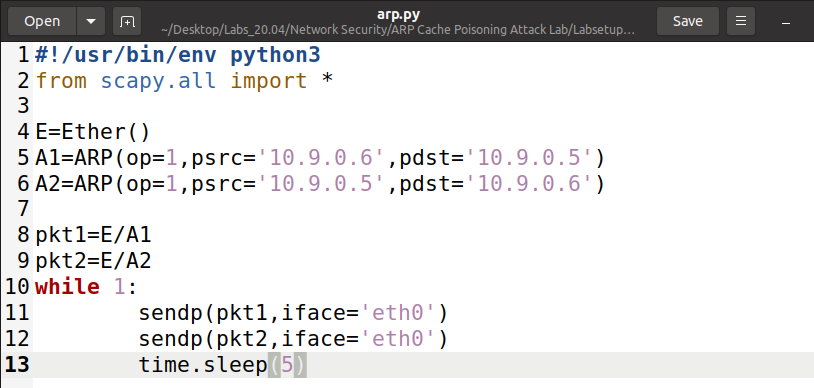




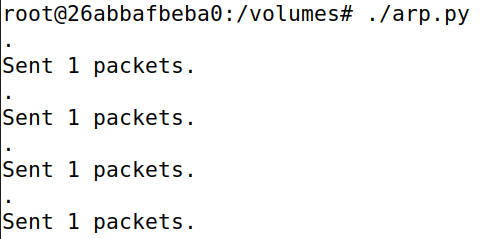
从结果来看，M确实起到了中间人的作用，A、B之间的通信都经过了M来转发。但是在每次M转发之后，都会对A/B发送一个ICMP重定向报文，告知A/B下次通信不必再发给自己，可以直接发给对方（因为在同一局域网内可以直接送达），这是因为计算机底层并不知道我们对A、B进行了ARP缓存攻击而企图纠正A、B的错误。但是ICMP作为网络层协议，给出的重定向自然是IP地址重定向，但中间人M的出现是由于A/B的IP地址映射到了M的MAC地址产生的，实际上是MAC地址错误导致，所以ICMP重定向报文不会起到作用。

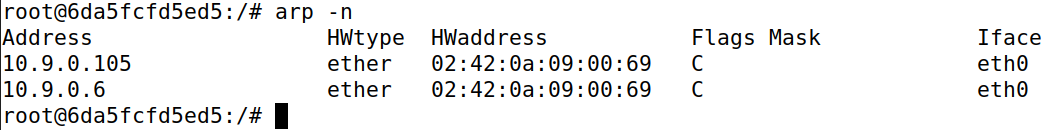
**Step 4 (Launch the MITM attack)**

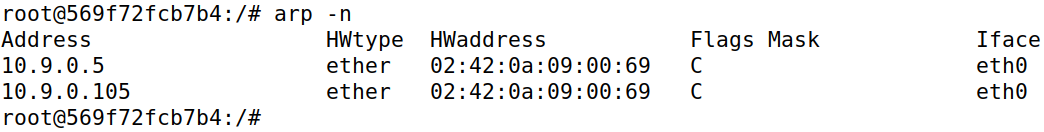
修改step1中的程序为每5秒执行一次：



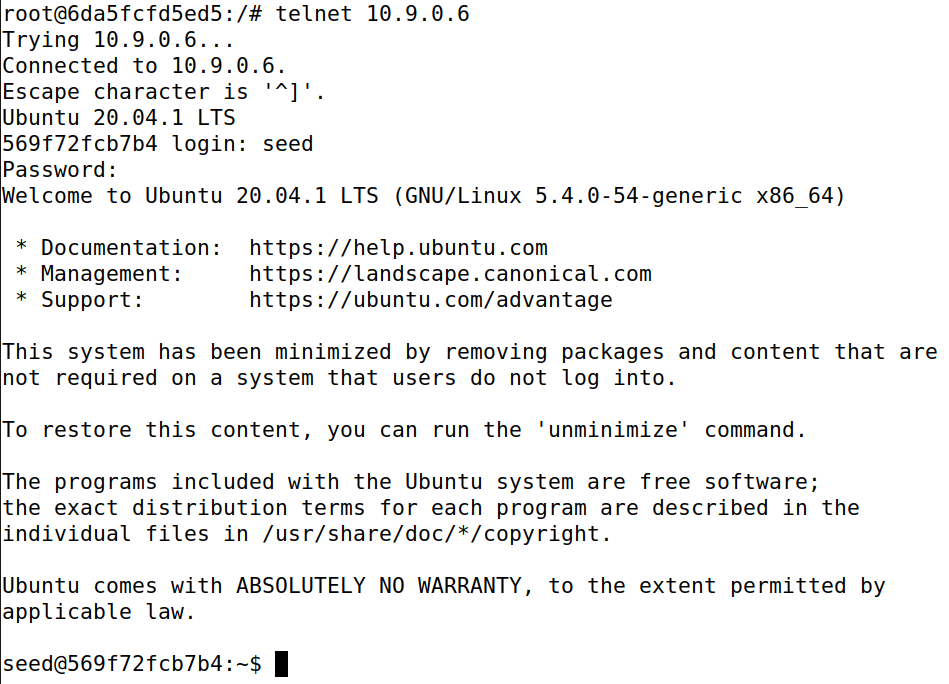
设置host M ip\_forward为1，随后执行上述攻击，检查A和B的arp缓存表是否被修改。



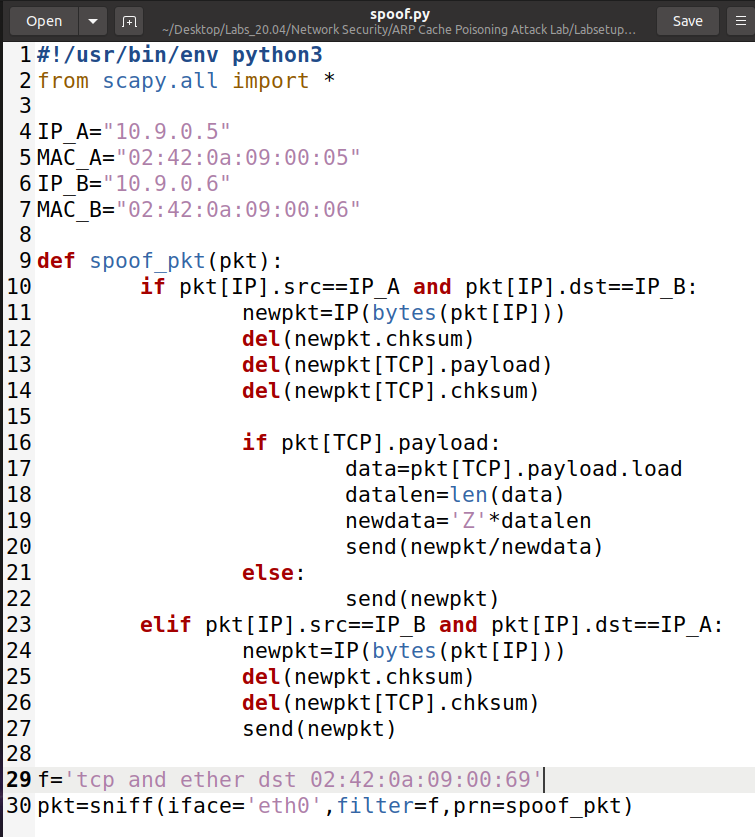




显示arp攻击成功。随后在A上telnet 10.9.0.6，



然后设置M的ip\_forward=0，运行以下程序：



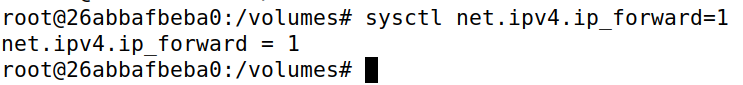
在A终端的telnet窗口输入几个变量，可以得到以下结果：



可以看到从A发往B的字符均被改成Z，攻击成功。

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

首先在M上设置sysctl net.ipv4.ip\_forward=1

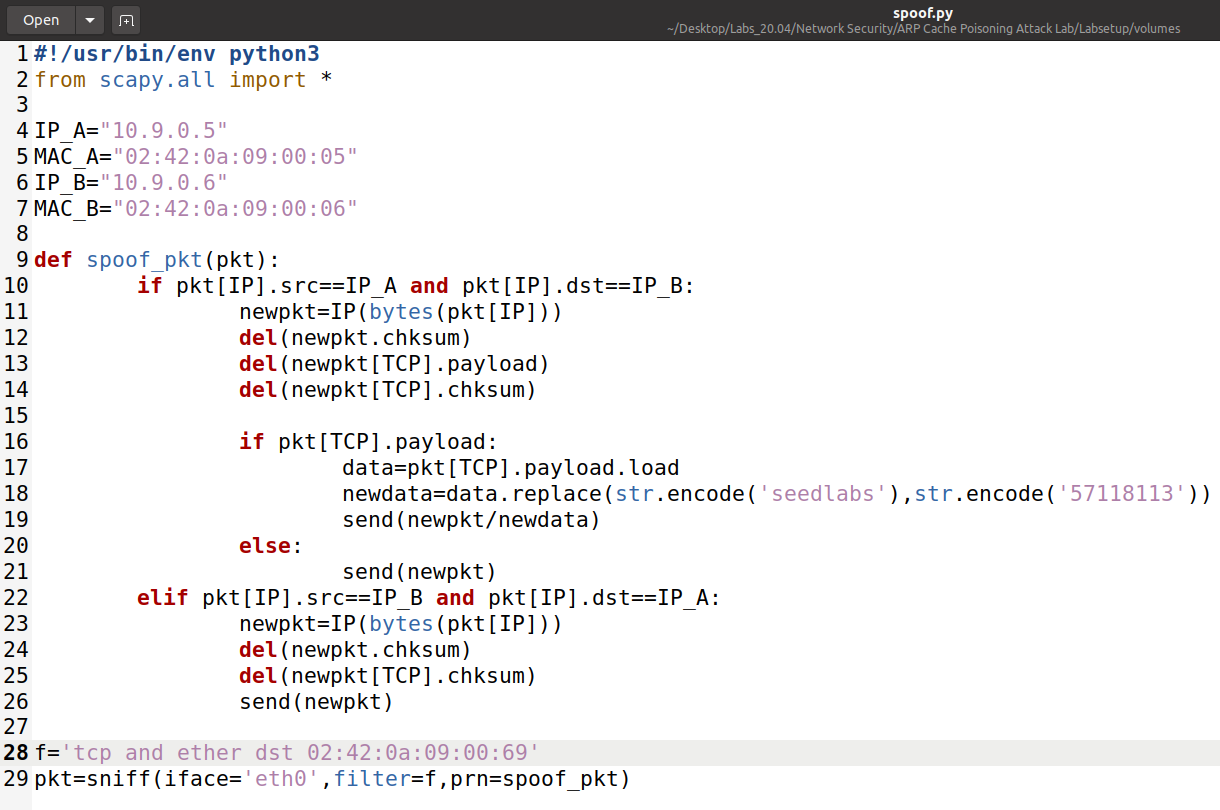


随后在M上执行arp.py，在host B上执行nc -lp 9090，在host A上执行nc -nv 10.9.0.6 9090。

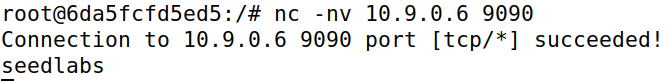


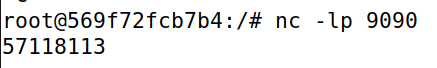


随后设置M ip\_forward为0，执行spoof.py。



在A终端中输入’seedlabs’，在B终端中查看结果：





攻击成功。