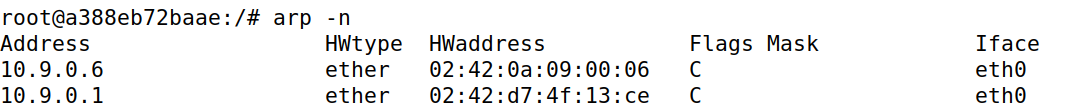
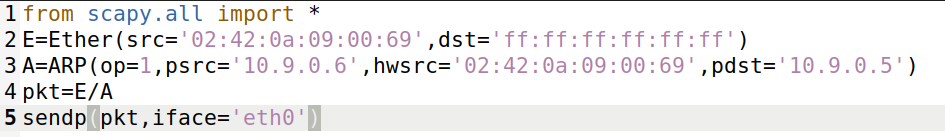
57118127孙博

**Task1.A**

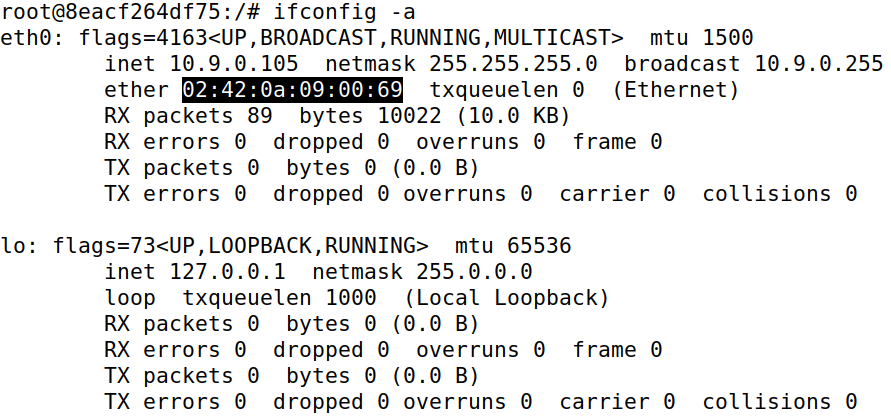
查看A中arp cache



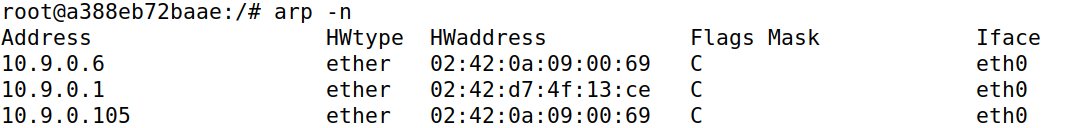
运行如下request代码



其中M的mac地址获取如下



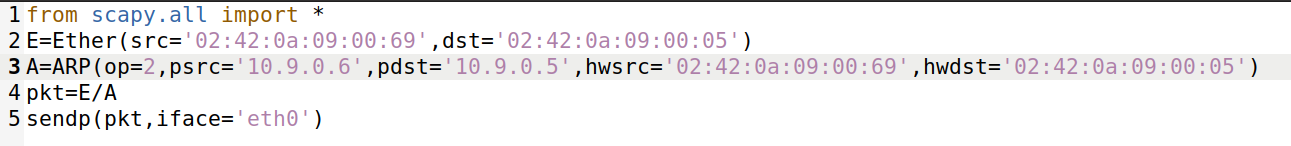
运行后再次在A中查看arp cache



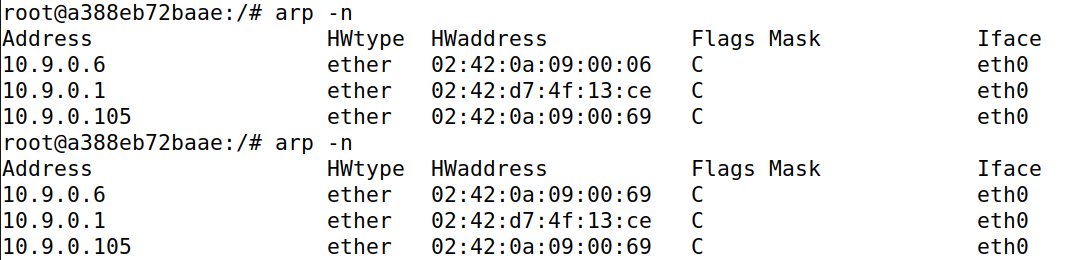
发现A中B的IP地址对应的mac地址已被修改为M的mac地址，攻击成功。

**Task1.B**

reply代码如下

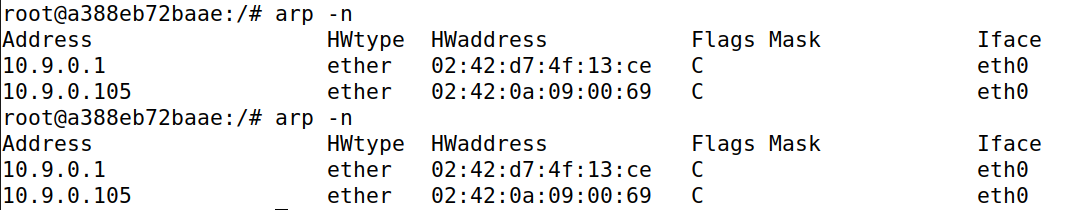


当10.9.0.6在A的arp cache中时



运行程序后发现修改成功

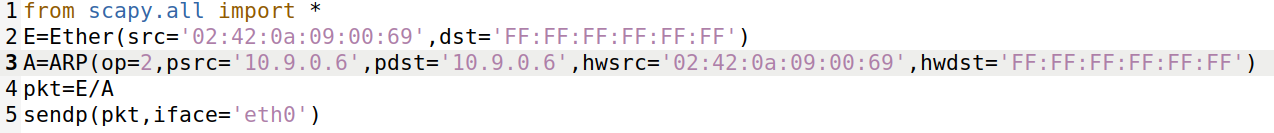
当10.9.0.6不在A的arp cache中，即用arp -d 10.0.9.6清除掉B时



运行发现修改失败

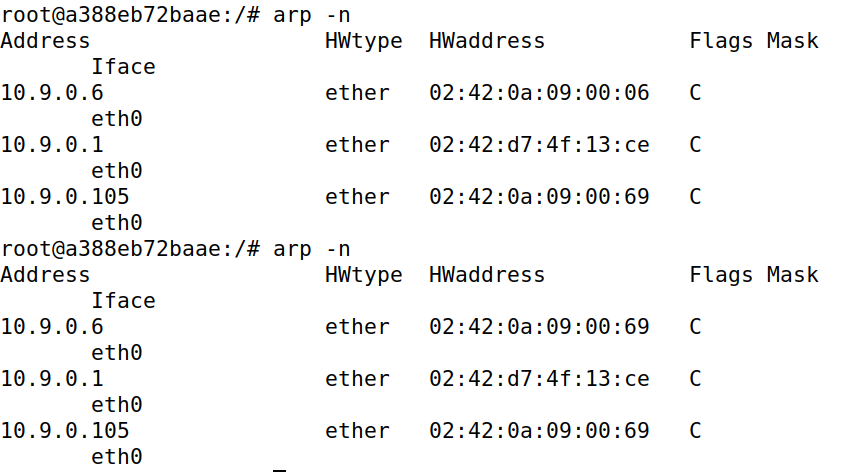
**Task1.C**

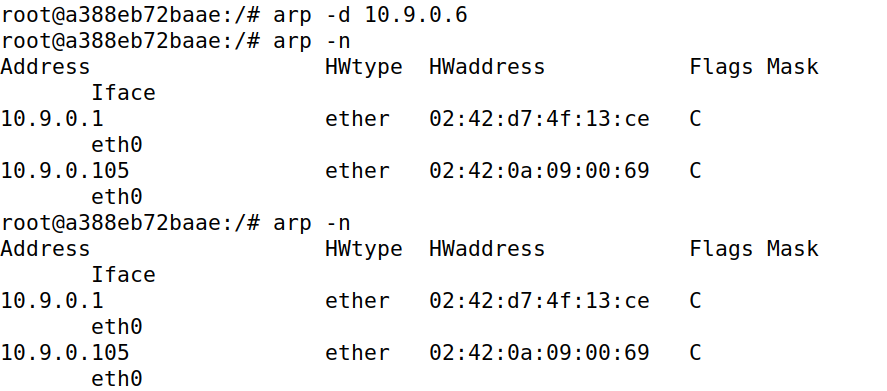
Arp免费报文代码构造如下



操作同1.B

当B的信息在A中时成功，不在时失败



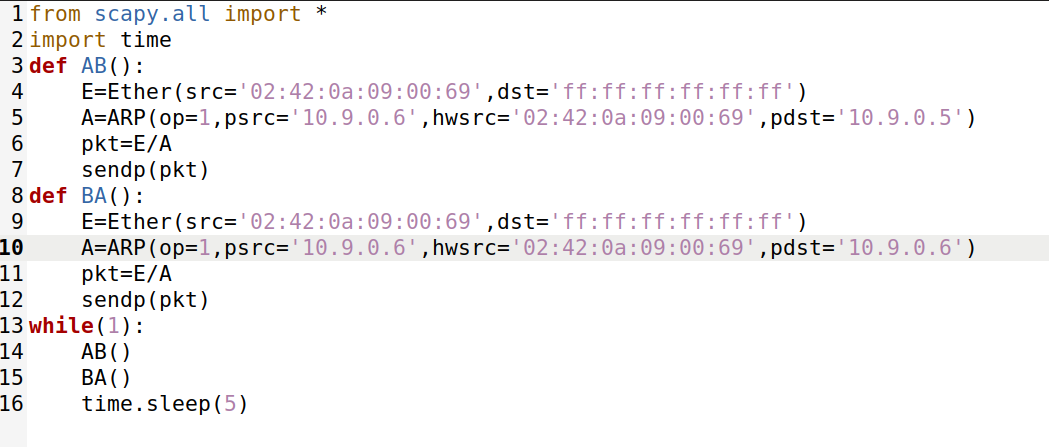


**Task2**

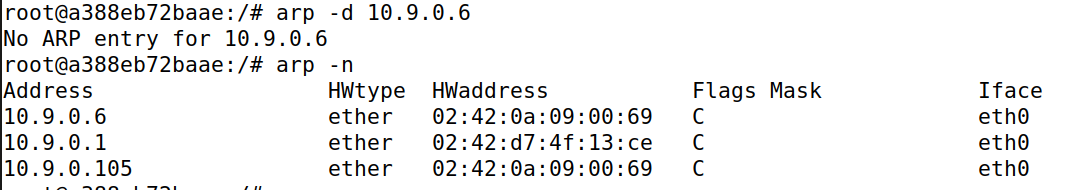
**Step1**

先在A和B上分别删掉彼此的arp缓存信息

采用1.A中方式，每隔5秒伪造一次报文。代码如下，并在M中运行



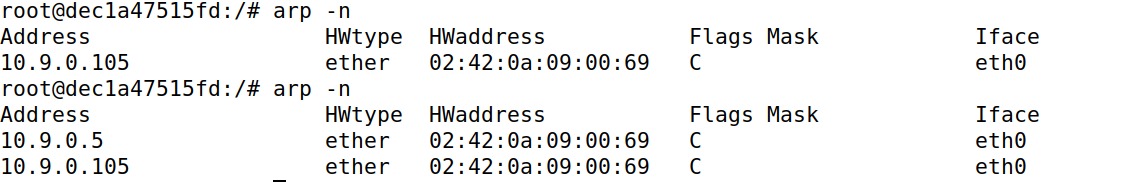
A中结果如下



出现到B的路由且MAC地址修改成功

B中结果如下

出现到A的路由且MAC地址修改成功



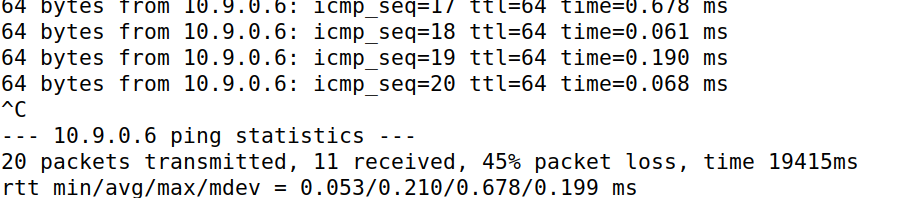
达成A和B间数据报经过攻击者的目标

**Step2**

关掉ipforward



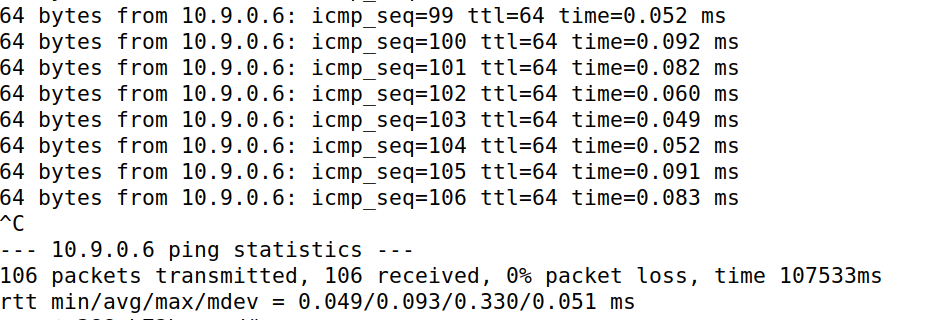
发现有丢包情况



**Step3**

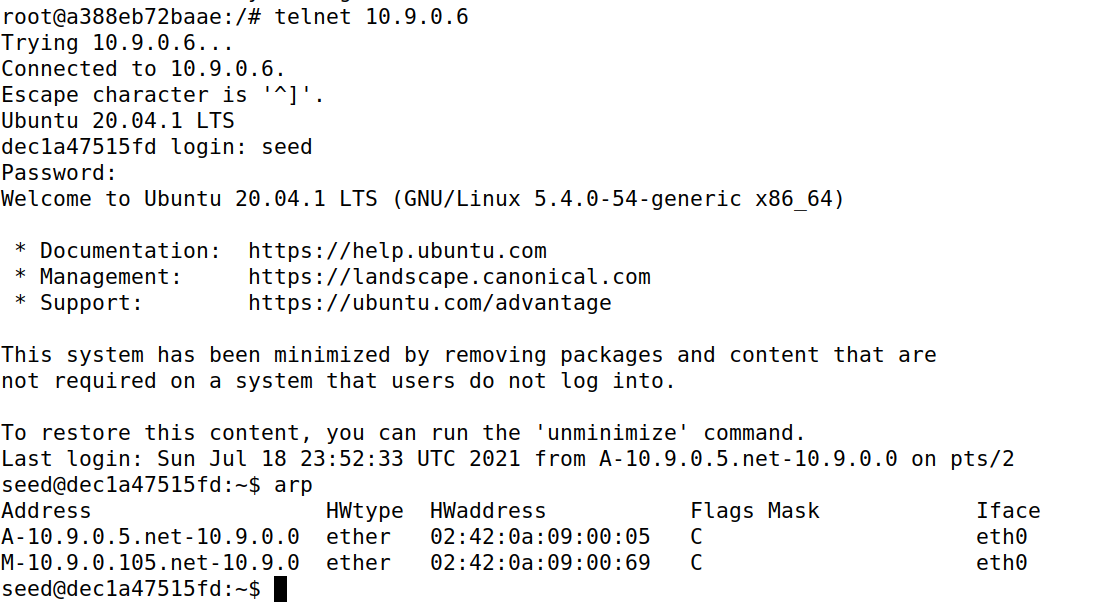
转发功能打开时并未受影响



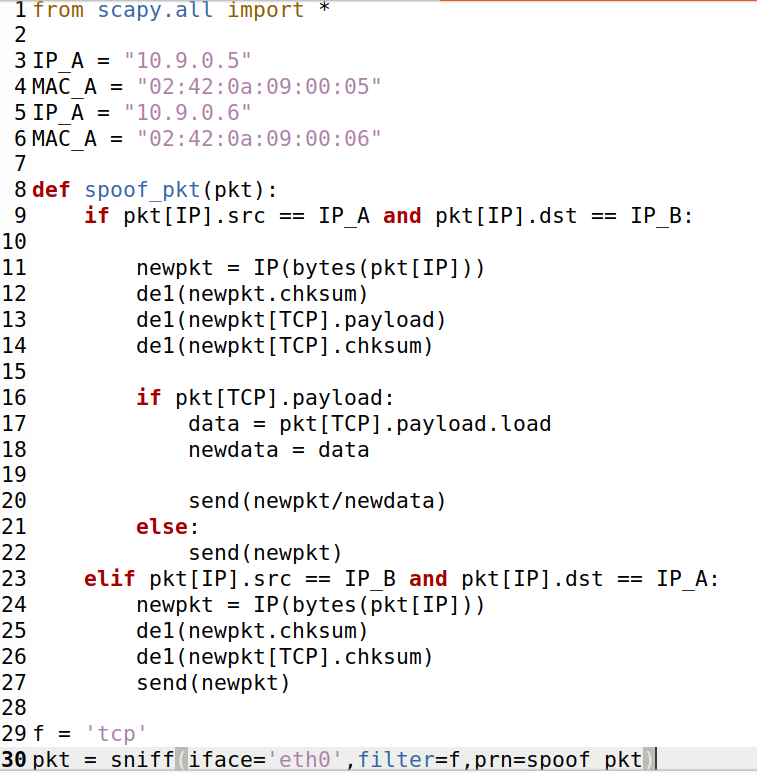


**Step4**

先在容器M中打开包转发功能，再在A容器中telnet容器B，发现可以连接上，且能正常输入命令并返回结果。此时关闭包转发功能，再输入命令，发现无法输入，因为关闭包转发后A和B无法通信。

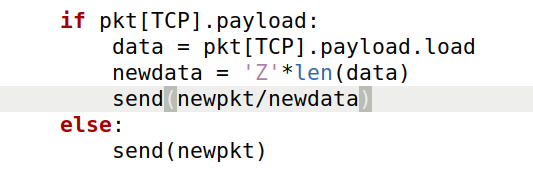


先用arp缓存中毒攻击攻击A和B，然后再M上打开包转发，此时AB间可以建立telnet通信，建立连接后关闭包转发，运行如下程序。此时A和B不能直接通信，但是发给B的包都发给了机器M，经过M发给B。代码如下：



攻击后A中telnet可以正常运行并输出结果，A中可以输出C中地址信息，说明攻击正常。

修改程序，目的是把A中键入的所有值都变成Z

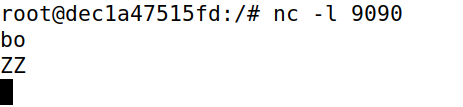


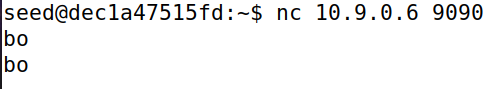
无论输入什么都会显示Z



**Task3**

在b中开netcat监听，并在A中输入名字，结果如下





输入字符被替换，攻击成功