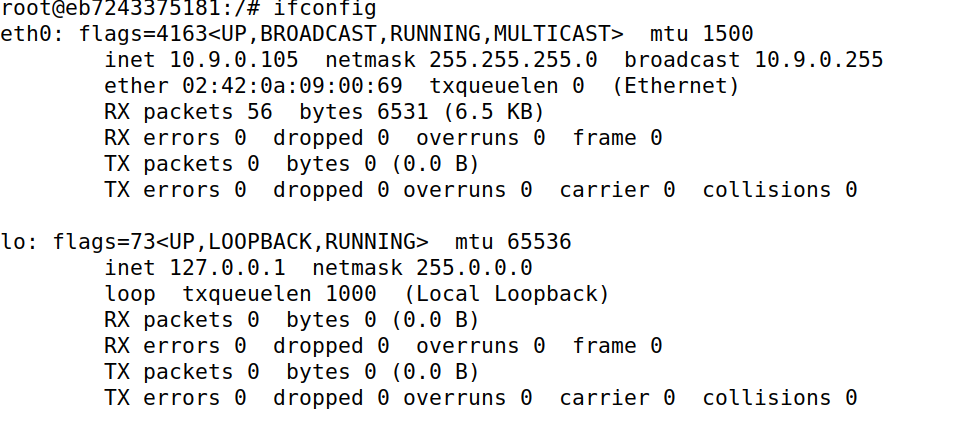
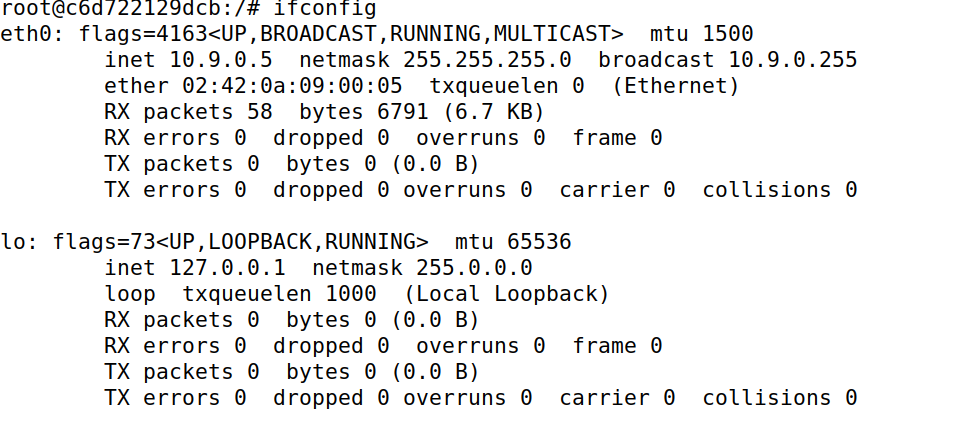
Task1：ARP Cache Poisoning

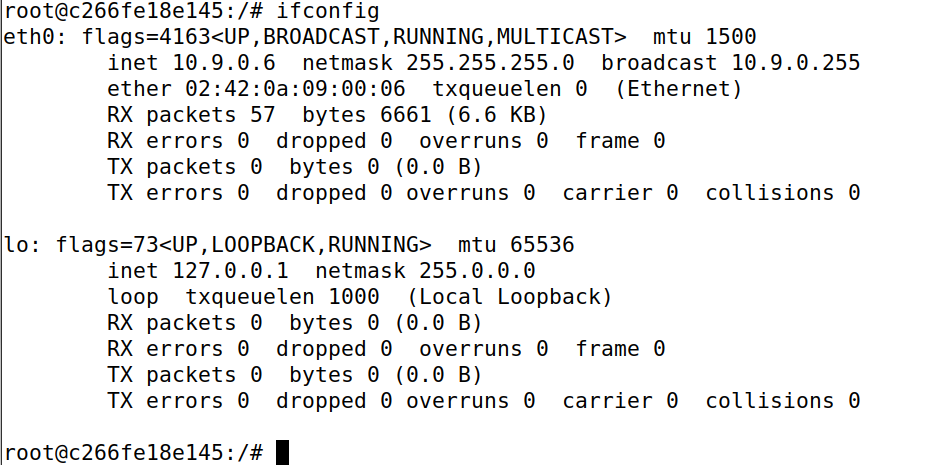
主机M上执行ifconfig命令查看端口名和mac地址：



查看主机A的ip地址和mac地址：



查看主机B的ip地址和mac地址：

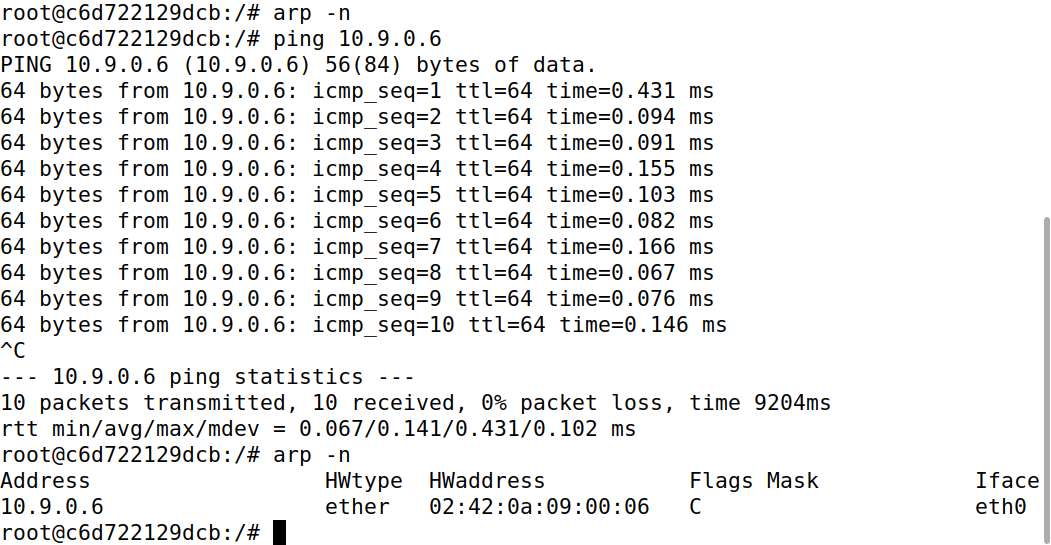


**Task1.A(using ARP request)**

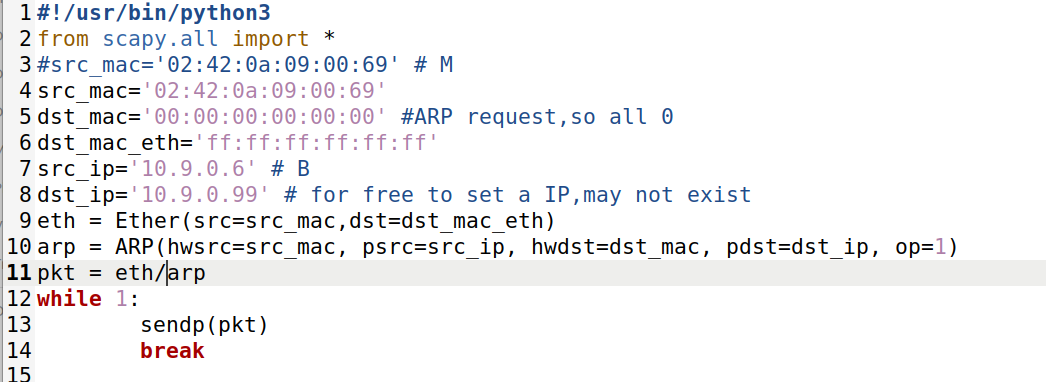
在主机A上查看arp缓存，发现未与其他主机建立连接前arp缓存为空：



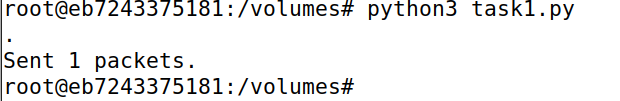
在主机A中ping主机B的ip地址，即ping 10.9.0.6，ping完后查看arp缓存，发现B的ip地址和mac地址的映射在A的arp缓存里。



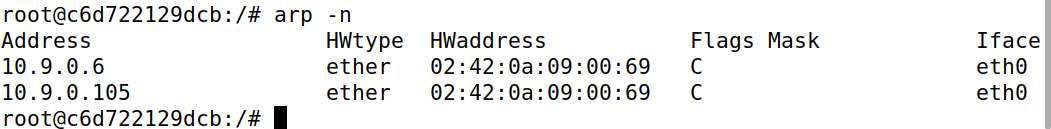
构造B发给A的arp请求包，但用的M的主机的mac地址，这样攻击成功后，就会将攻击主机的mac映射到B的ip地址上，代码如下：



在主机M上运行攻击程序（运行程序的时候需要保持主机A一直在ping主机B）：

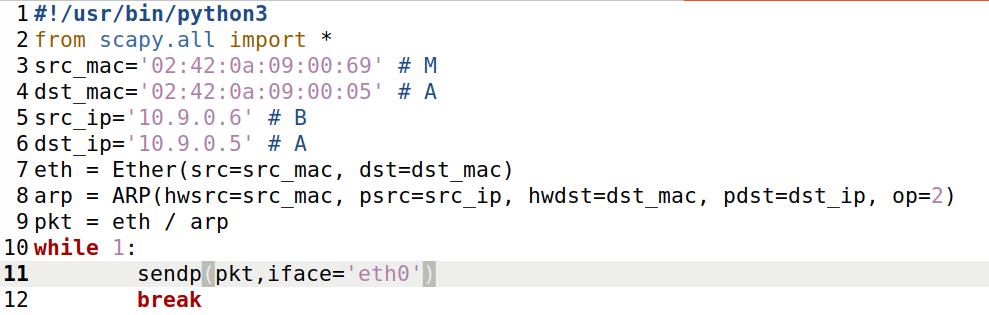


主机A查看arp缓存，发现成功的将M的mac地址映射到B的ip地址上：



**Task1.B(using ARP reply)**

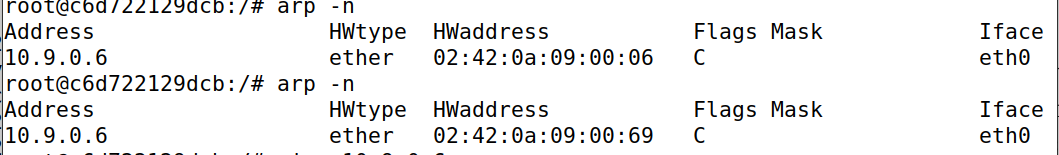
构造reply包的代码如下：



在运行时会遇到两种情况：

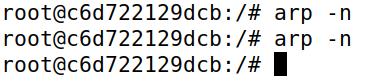
1. B的ip已经在A的缓存中

运行程序，再次查看缓存，发现主机B的ip地址成功映射到M主机的mac地址上，攻击成功



1. B的ip不在A的缓存中

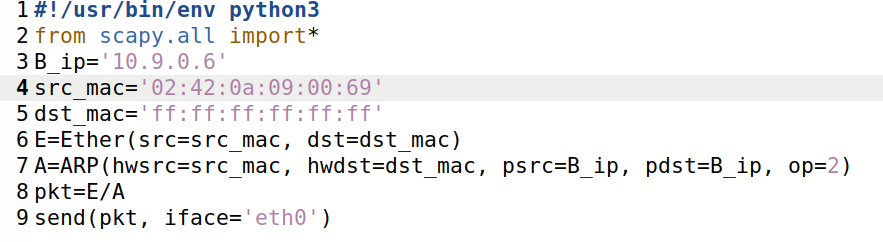
清空A的arp缓存，删除10.9.0.6的条目



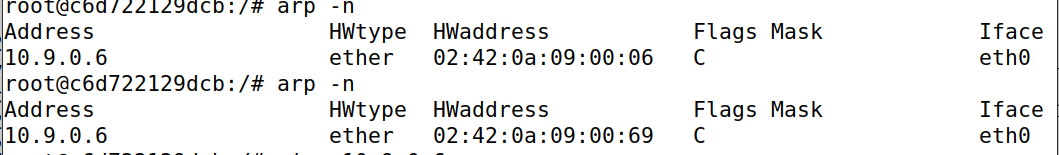
清空后B的ip不在A的缓存里，运行程序，再次查看缓存发现没有M的mac地址到B的ip地址间的映射，说明B的ip不在A的缓存中时，arp 缓存中毒攻击失败。这是因为reply 包只能更新而不能增加arp 缓存条目，因此当B 的ip 在A 的缓存中时可以成功，不在时则失败。

**Task1.C(using ARP gratuitous message)**

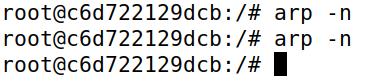
代码如下：



1. B的ip在A的缓存中，执行程序，结果如下：



1. B的ip不在A的缓存中，执行程序，结果如下：

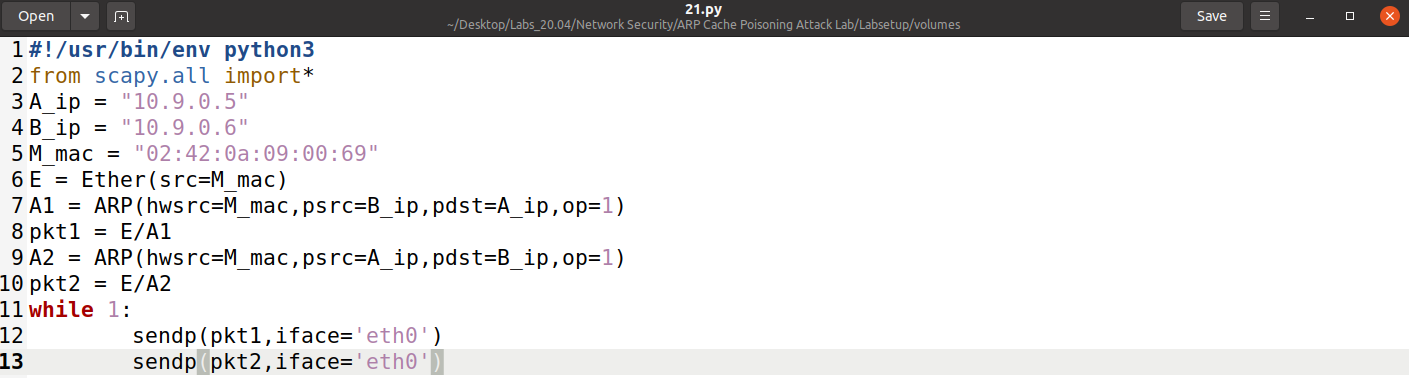


发现B的ip地址不在A的缓存时没有增加映射，攻击失败，可见免费arp报文应该和reply报文类似，只能更新不能增加缓存条目。

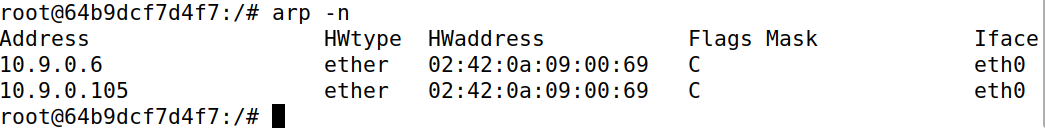
Task2：MITM Attack on Telnet using ARP Cache Poisoning

步骤1: (Launch the ARP cache poisoning attack)

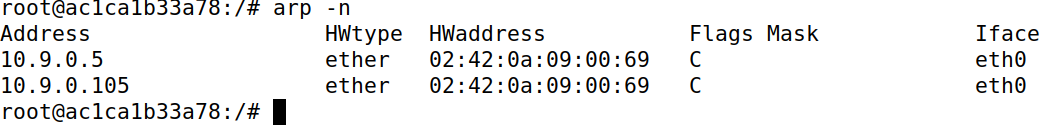
对主机A，B都进行arp缓存中毒攻击，代码中对A和B都发送请求包来进行攻击，同时设置循环不停发送报文，防止假条目被真条目替代，代码如下：



主机A的arp缓存：

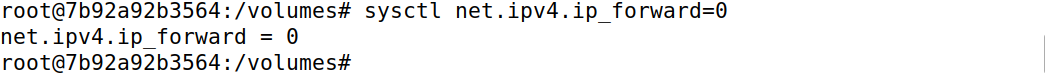


主机B的arp缓存：

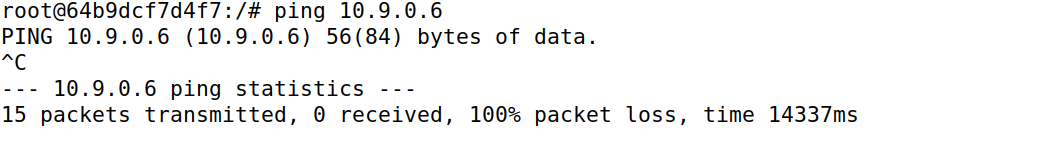


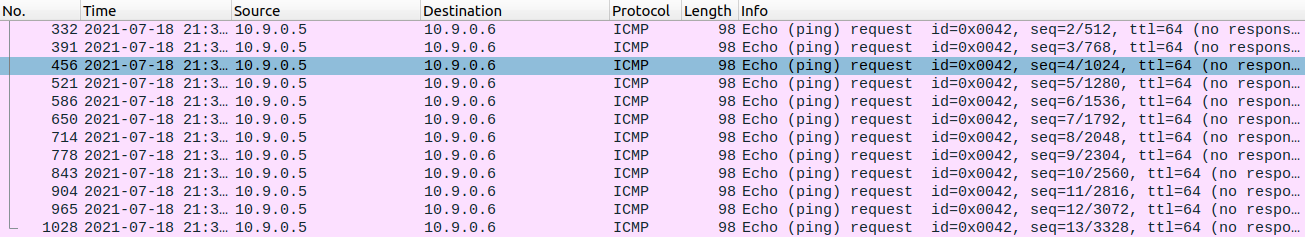
步骤2:(Testing)

关闭M上的ip转发：



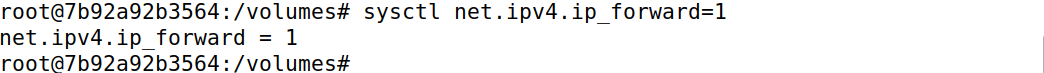
运行程序，然后在主机A上ping主机B的ip地址，发现ping不同，丢包率100%：



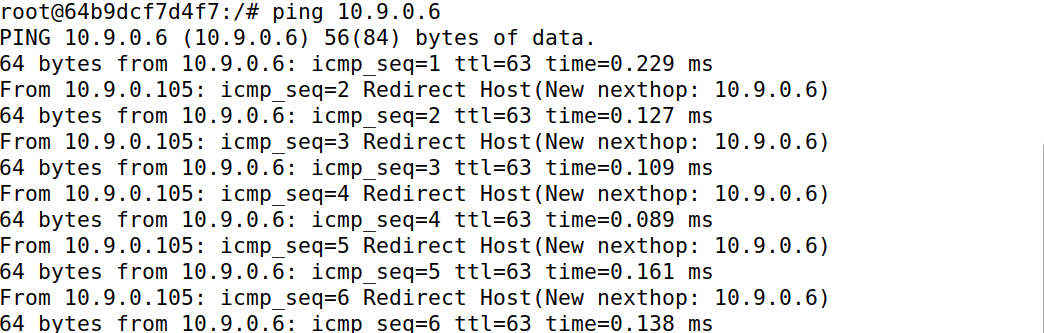


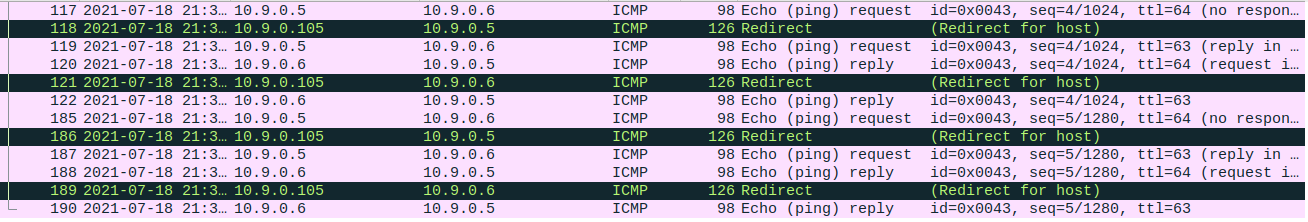
步骤3: (Turn on IP forwarding)

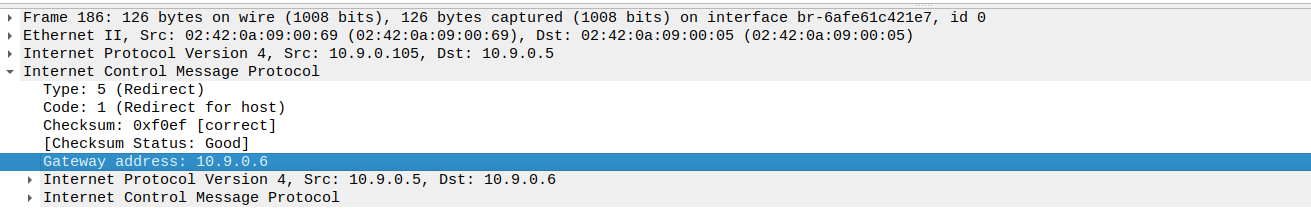
打开主机M的ip转发，重复步骤2的操作：



发现能够成功ping通：

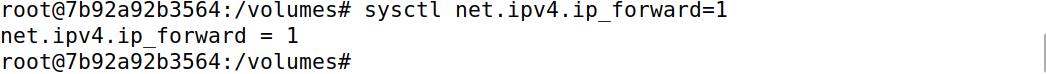




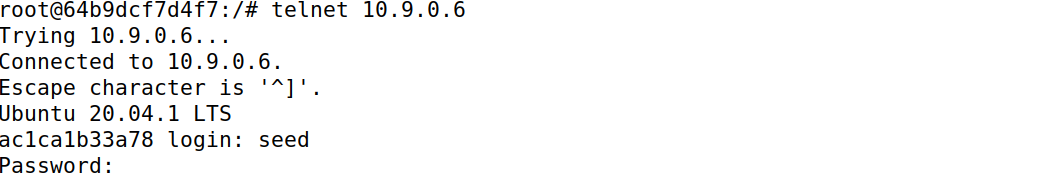


步骤4：(Launch the MITM attack)

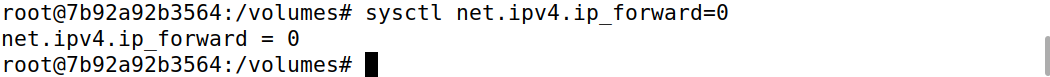
开启主机M的ip转发：



在主机A上对主机B进行telnet连接：



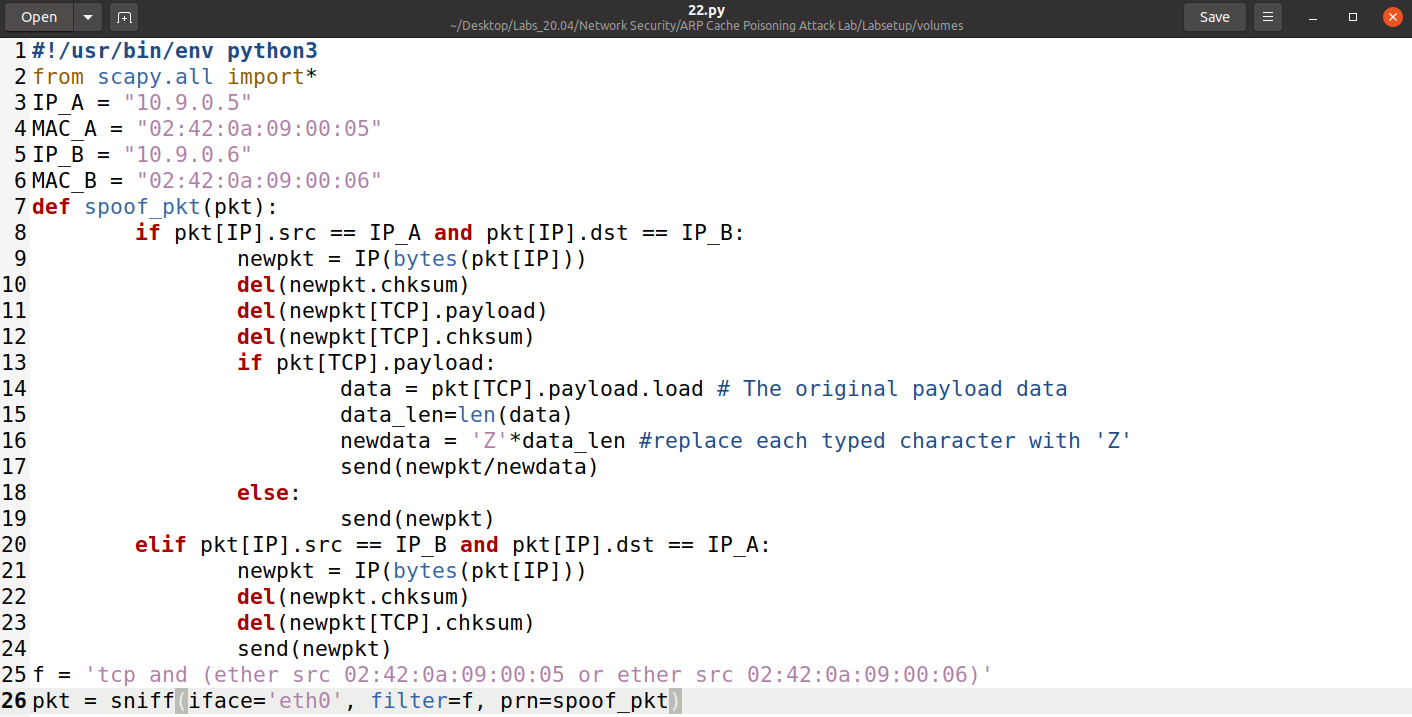
关闭主机M的ip转发：

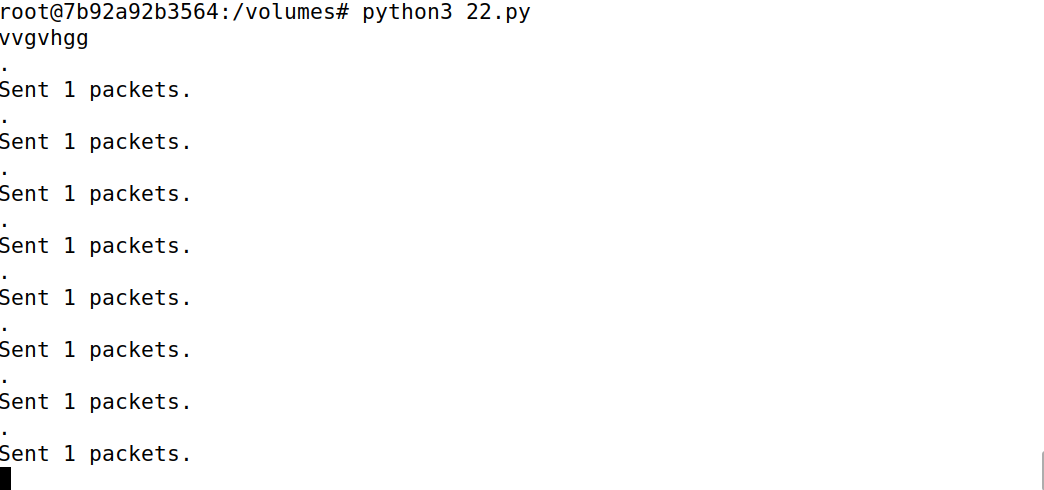


执行步骤1的代码，进行arp缓存中毒攻击：



此时在A的远程登录窗口无法输入任何字符。然后执行中间人攻击代码，截取A发往B的所有tcp报文，将输入字符改为‘Z’，B发往A的响应报文不做修改，代码如下：



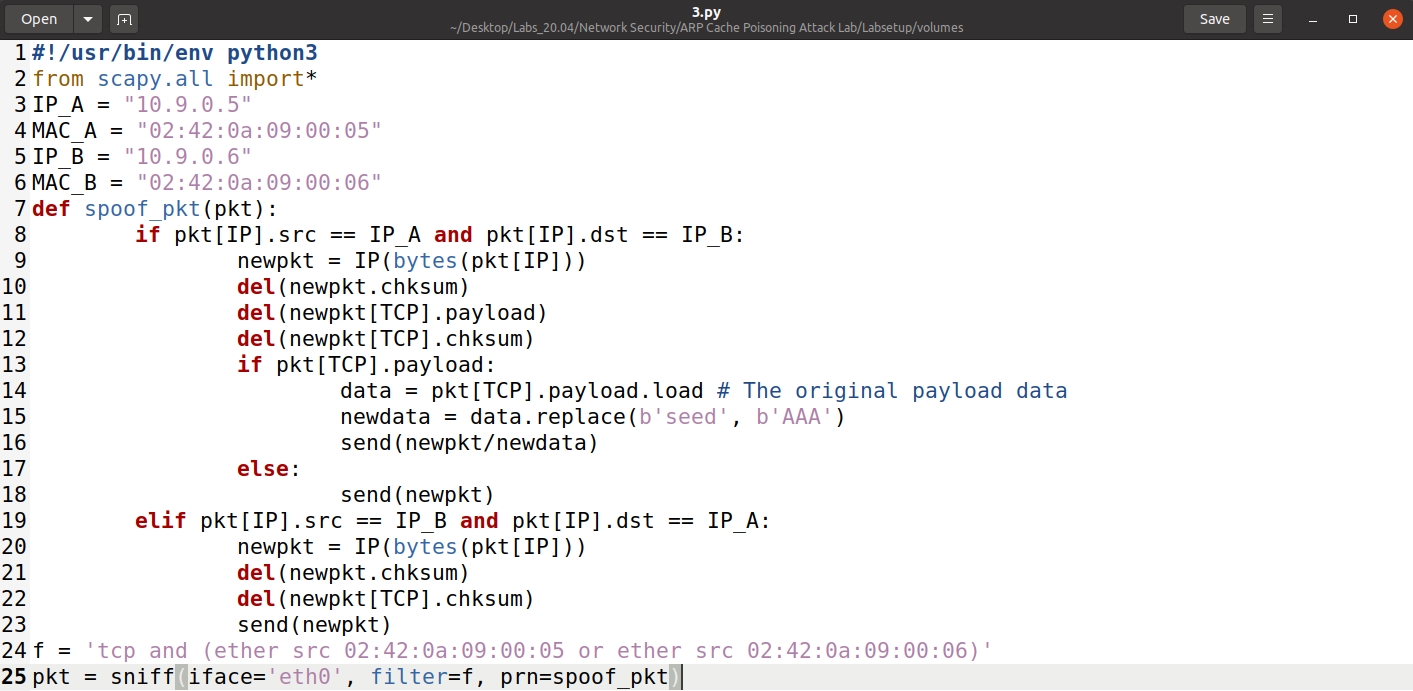


不管输入什么，最终显示的都为Z。攻击成功。



Task3：MITM Attack on Netcat using ARP Cache Poisoning

与task2 类似，只需将代码中修改数据部分变为把seed字符串改为AAA即可。步骤也类似，将Telnet 通信改为 netcat 通信。代码如下：



结果如下：





可见所有的字符串seed都改为了AAA,其他部分不变，攻击成功。