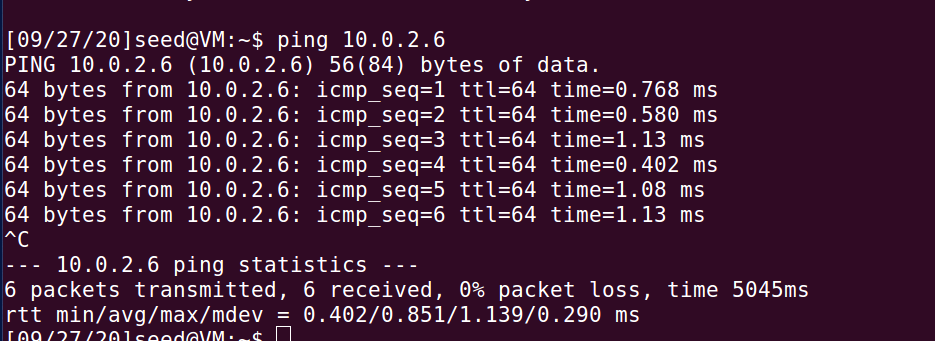
**Task1: Network Setup**

Host U - seed1 - **10.0.2.4**

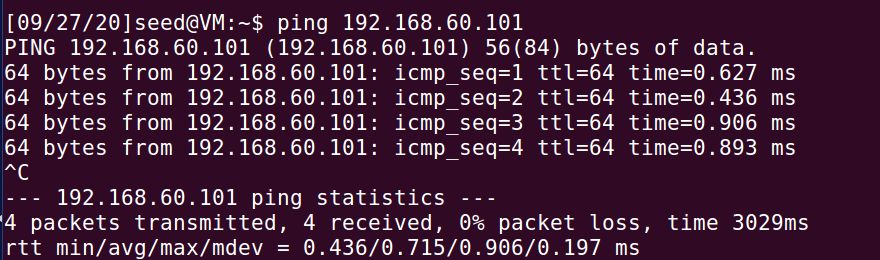
VPN server – seed2 -**10.0.2.6** - **192.168.60.1**

Host V - seed3 – **192.168.60.101**

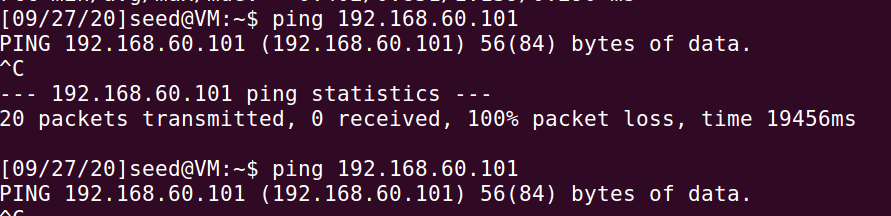
1.Host U can communicate with VPN Server.



2. VPN Server can communicate with Host V.



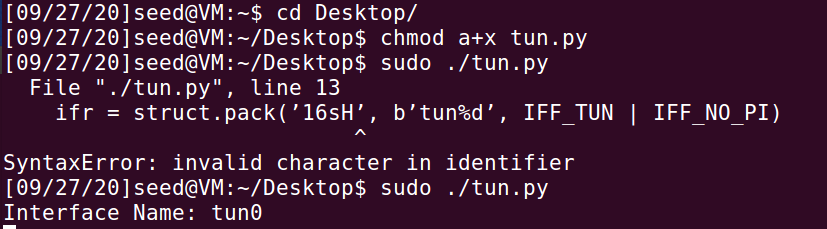
3. Host U should not be able to communicate with Host V.



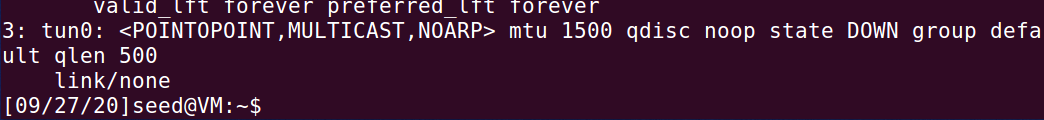
**Task2: Create and Configure TUN Interface**

**3.1 Task2.a: Name of the Interface**

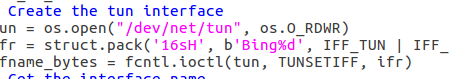
编译并运行tun.py



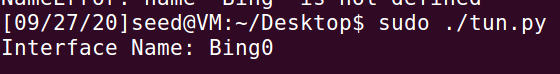
发现创建了一个tun0接口

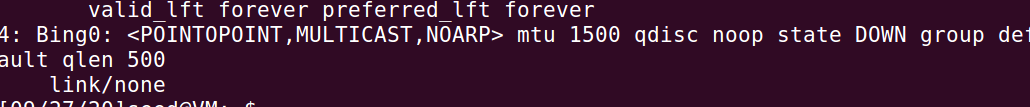


修改接口名字



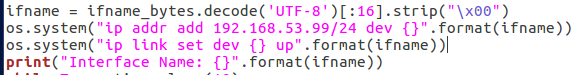
再次运行，结果如下



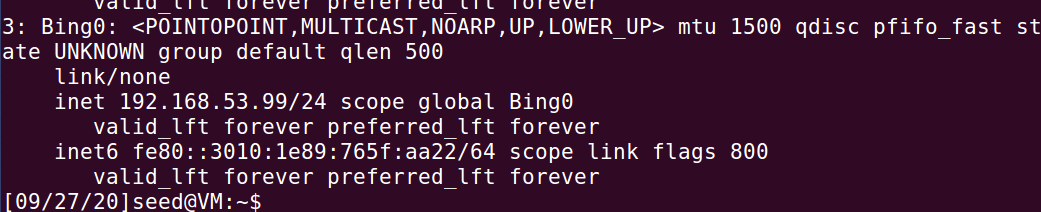


**Task2.b: Setup the TUN Interface**

按要求在代码中添加

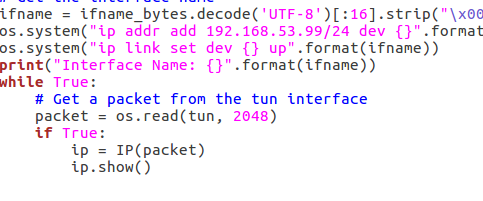


发现Bing0接口有了ip地址，并且被激活



**Task2.c: Read from the TUN Interface**

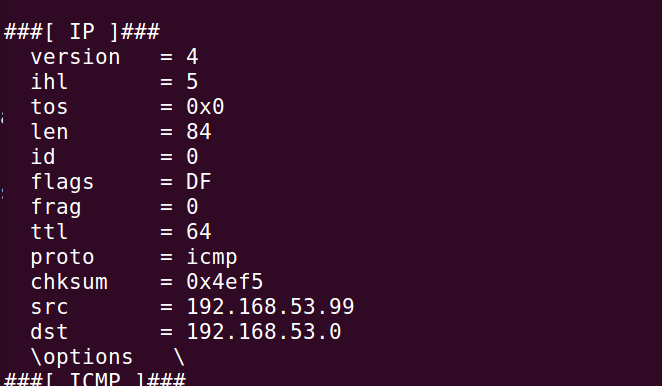
按要求替换while循环



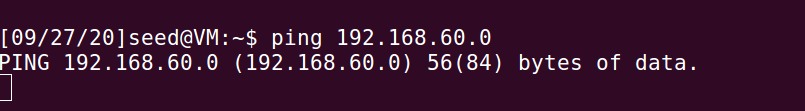
ping 192.168.53.0



打印出了dst为192.168.53.0的ip报文



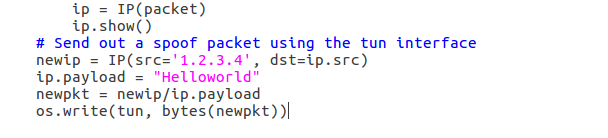
ping 192.168.60.0



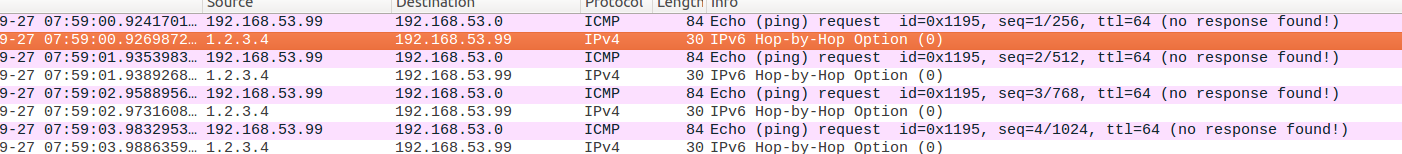
没有新的内容打印出来，因为ping不到内部网络



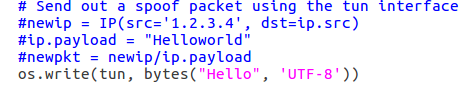
**Task2.d: Write to the TUN Interface**



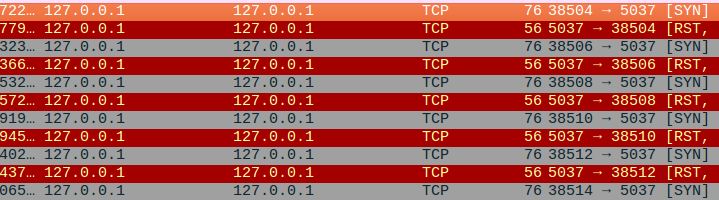
发出了IP报文



不是发送IP报文，而是将任意数据写入端口

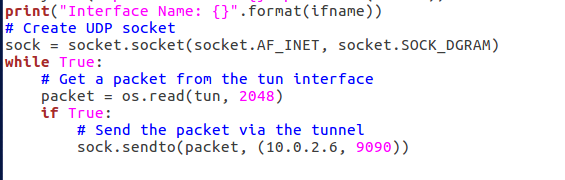


发现在本机间发送TCP请求报文和RST拒绝报文

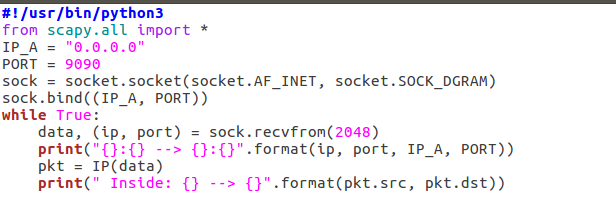


**Task3: Send the IP Packet to VPN Server Through a Tunnel**

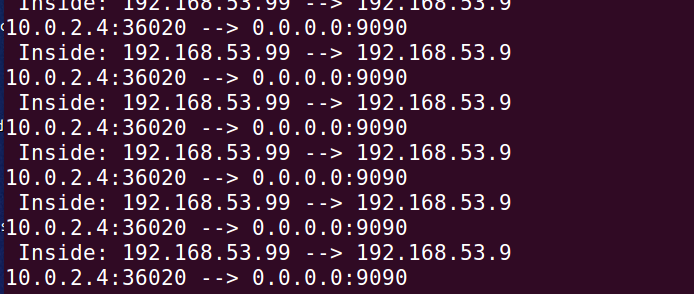
tun\_client.py



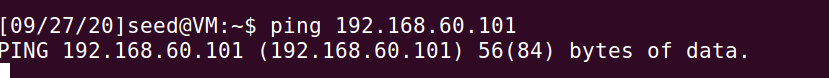
tun\_server.py



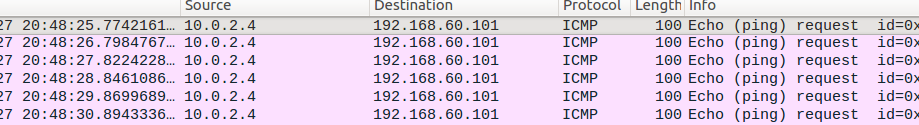
ping 192.168.53.9，打印结果如下：



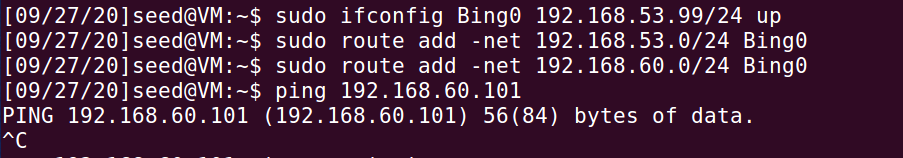
ping 192.168.60.101



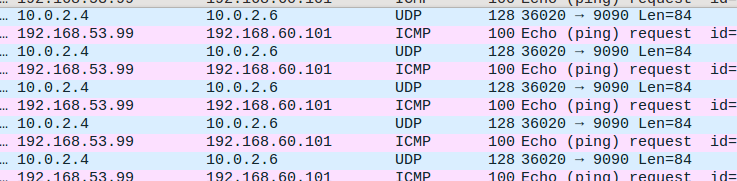
没有ICMP数据包发到VPN Server，而是直接发向了192.168.60.101



增加路由信息，再次ping 192.168.60.101

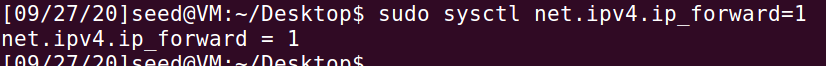


发现成功的把去往192.168.60.101的数据包路由到了Bing0接口，从而进入了VPN隧道

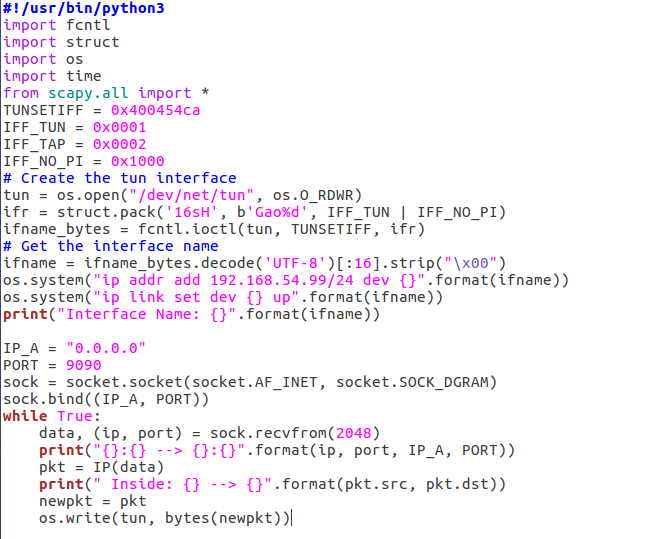


**Task4: Set Up the VPN Server**

启用IP转发



tun.server.py

结果如下

在hostV上看到了ping request的icmp报文

