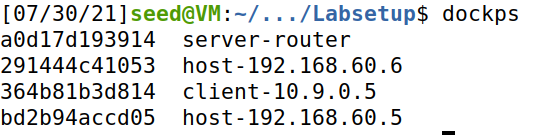
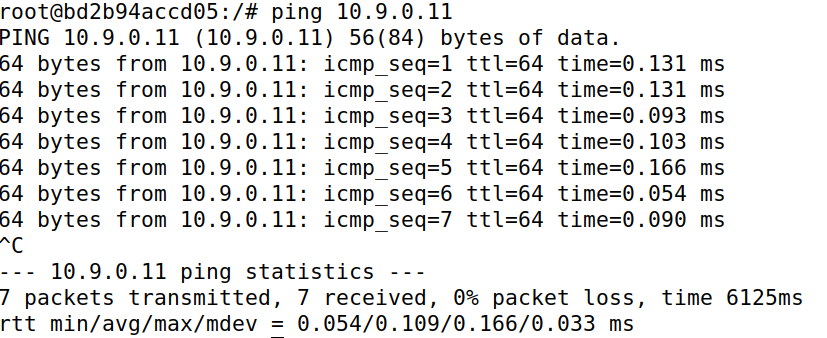
**VPN Lab: The Container Version**

57118112 王怡乐

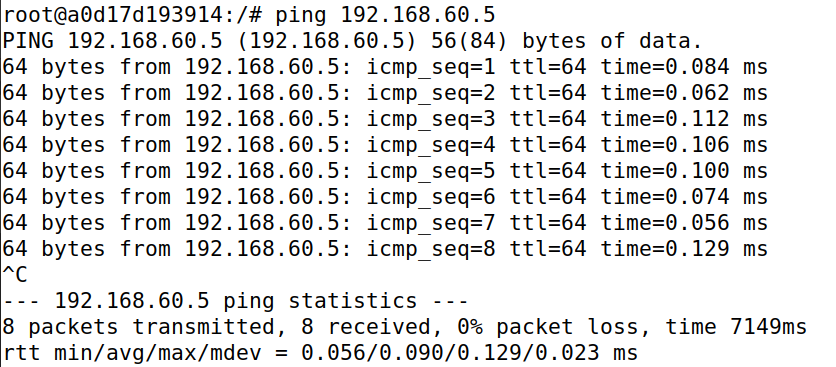


**Task 1: Network Setup**

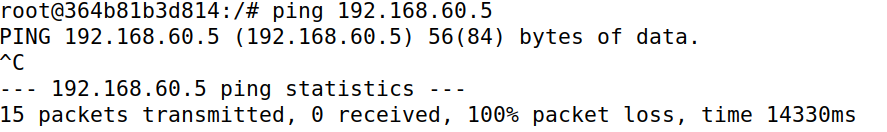
在主机U上ping VPN服务器，可以ping通



在VPN服务器上ping 主机V，可以ping通

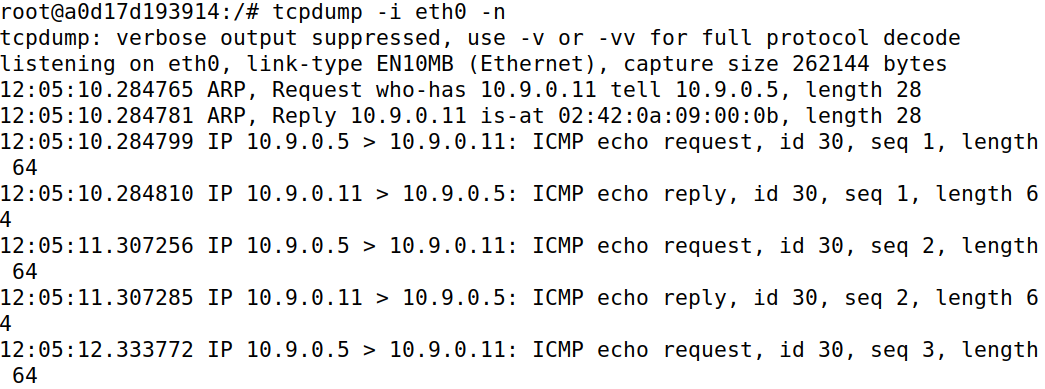


在主机U上ping主机V，不通

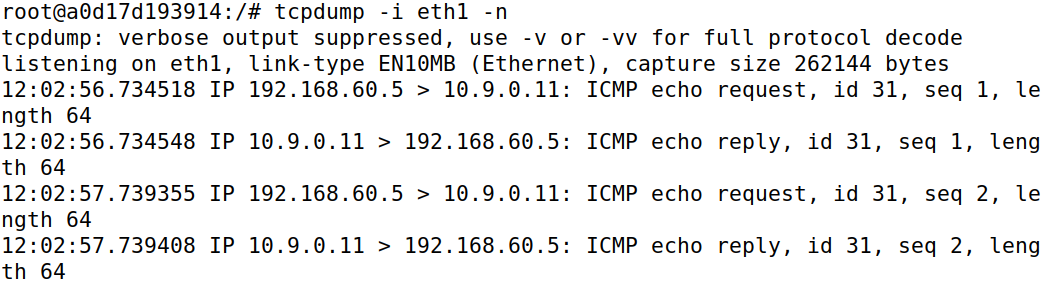


路由器上运行 tcpdump

监听端口eth0



监听端口eth1



**Task 2: Create and Configure TUN Interface**

**Task 2.a: Name of the Interface**

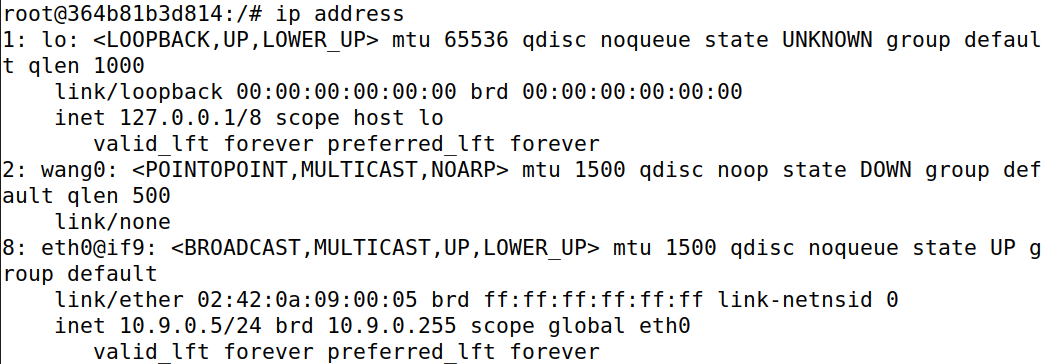
在代码中修改端口名为“wang”



在主机U上运行程序tun.py



打开另一个终端查看，端口名修改为“wang0”

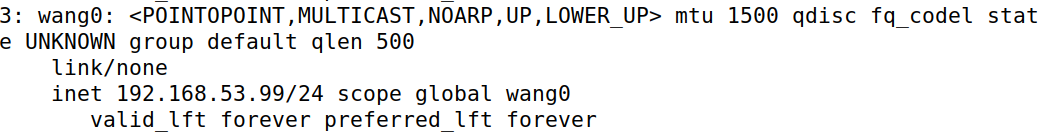


**Task 2.b: Set up the TUN Interface**

在tun.py中添加以下代码，给端口 wang0 自动分配 ip 地址

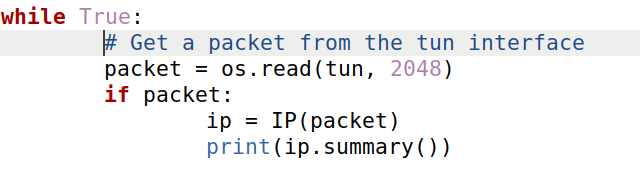


再次在主机U上运行程序tun.py，并另开一个终端输入ip address 命令，发现wang0端口已被分配ip地址

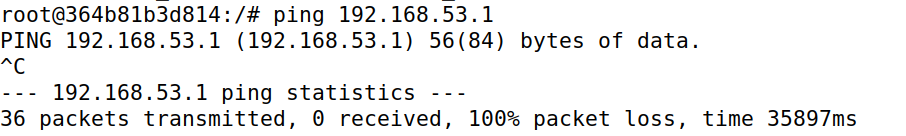


**Task 2.c: Read from the TUN Interface**

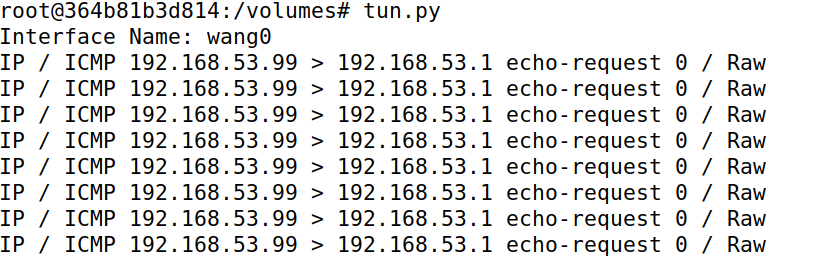
修改程序中的while循环如下



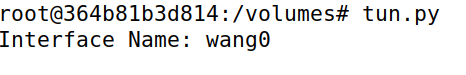
再次在主机U上运行程序tun.py，同时ping 192.168.53.1，显示ping不通

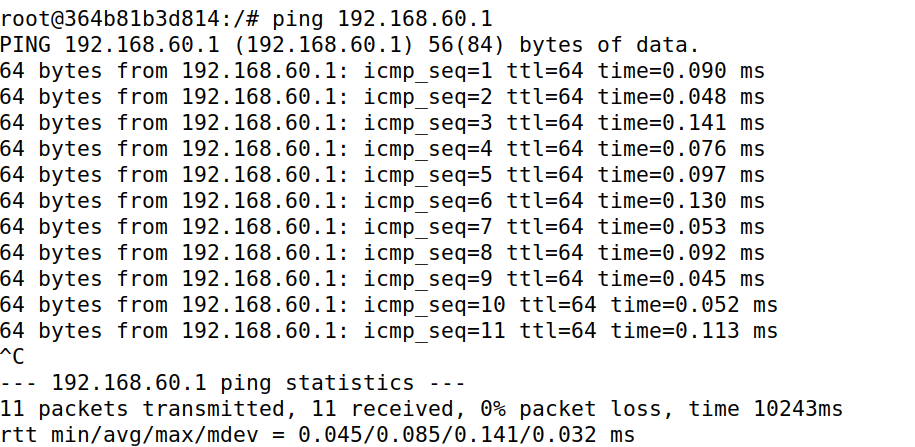


查看tun.py程序的输出，发现ICMP 请求报文都被端口捕获，这是因为发送给192.168.53.0/24 网段的数据包是从 wang0 端口发出



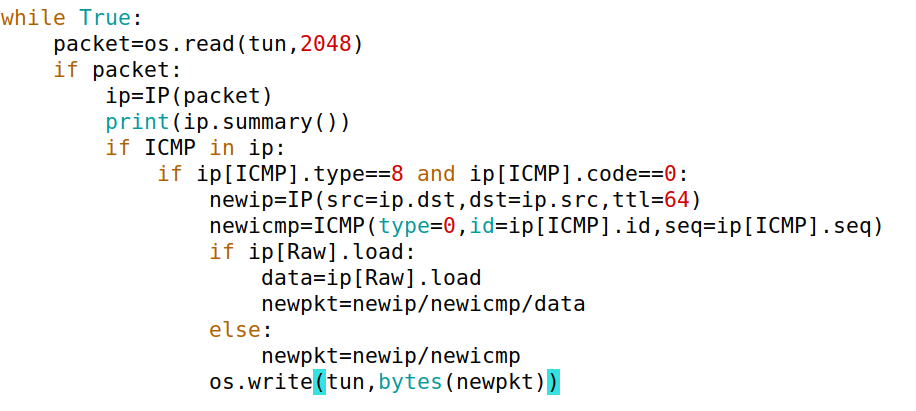
ping 192.168.60.1时，显示可以ping通，且程序无输出。这是因为发送给 192.168.60.1 的报文不经过 wang0 端口，所以没有捕获报文



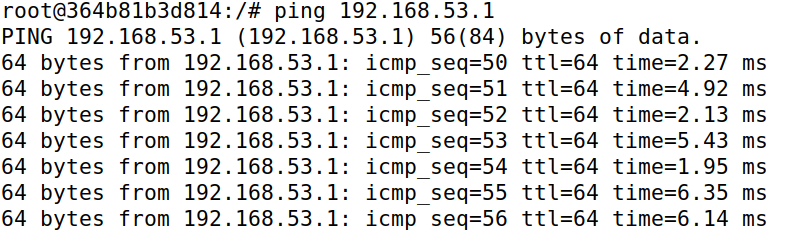


**Task 2.d: Write to the TUN Interface**

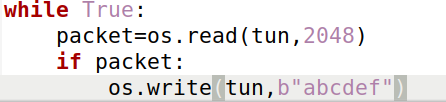
修改while循环如下



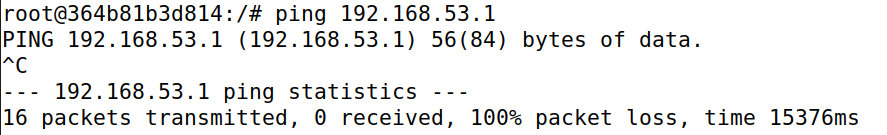
运行程序，然后再次 ping 192.168.53.1，显示可以ping通，说明伪造响应包成功



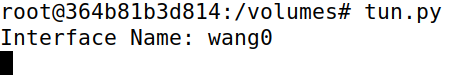
再次修改while循环



运行程序，然后ping 192.168.53.1，显示ping不通

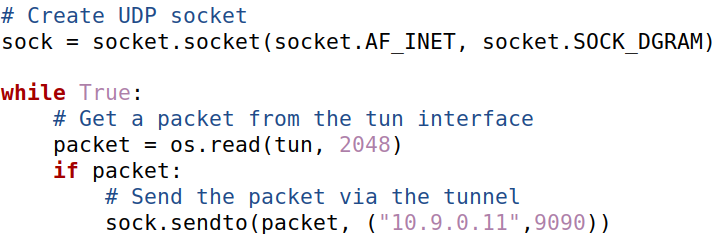


tun.py程序无输出

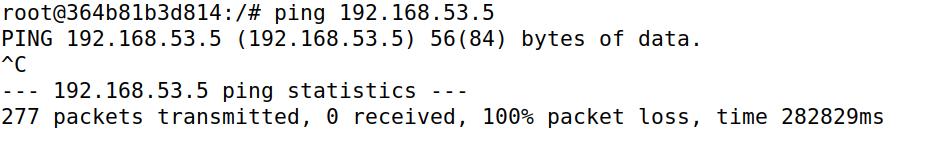


**Task 3: Send the IP Packet to VPN Server Through a Tunnel**

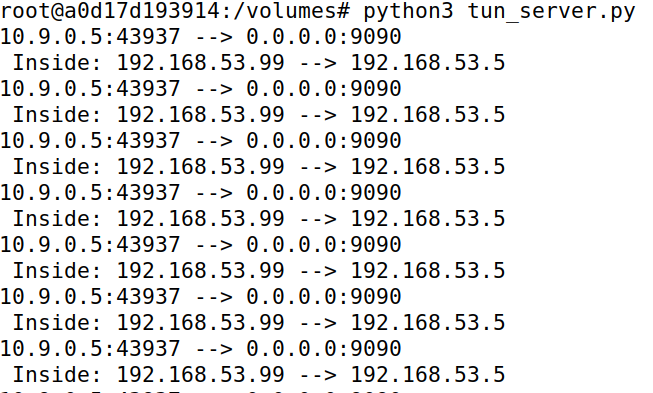
将原程序的while循环更改为如下代码，并重命名为tun\_server.py



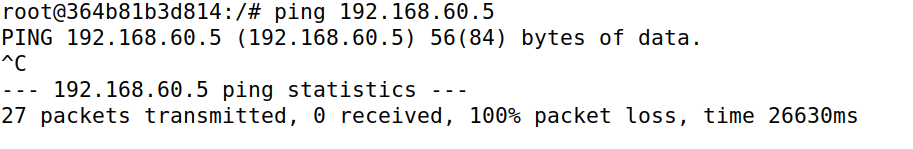
在主机U上运行tun\_client.py，在VPN服务器上运行tun\_server.py。然后在主机U上ping 192.168.50.5，显示ping 不通



VPN服务器输出如下。此时 VPN 服务器成功捕获到了报文。这是因为 tun\_client.py 程序将捕获的报文发给了 VPN 服务器的 9090 端口。



在主机U上ping主机V，显示ping不通

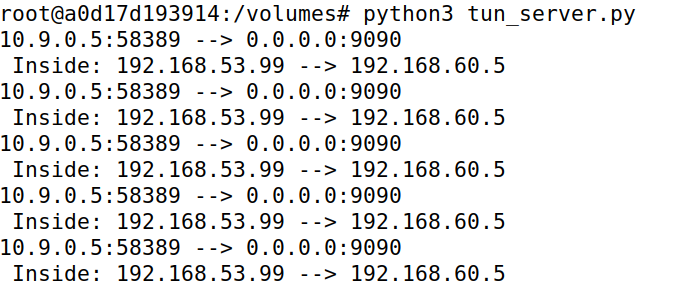


此时VPN服务器无输出，这是因为此时主机U上没有去往 192.168.60.0/24 的路由，报文不会从 tun 端口发出。

在tun\_client.py中添加以下代码自动配置路由

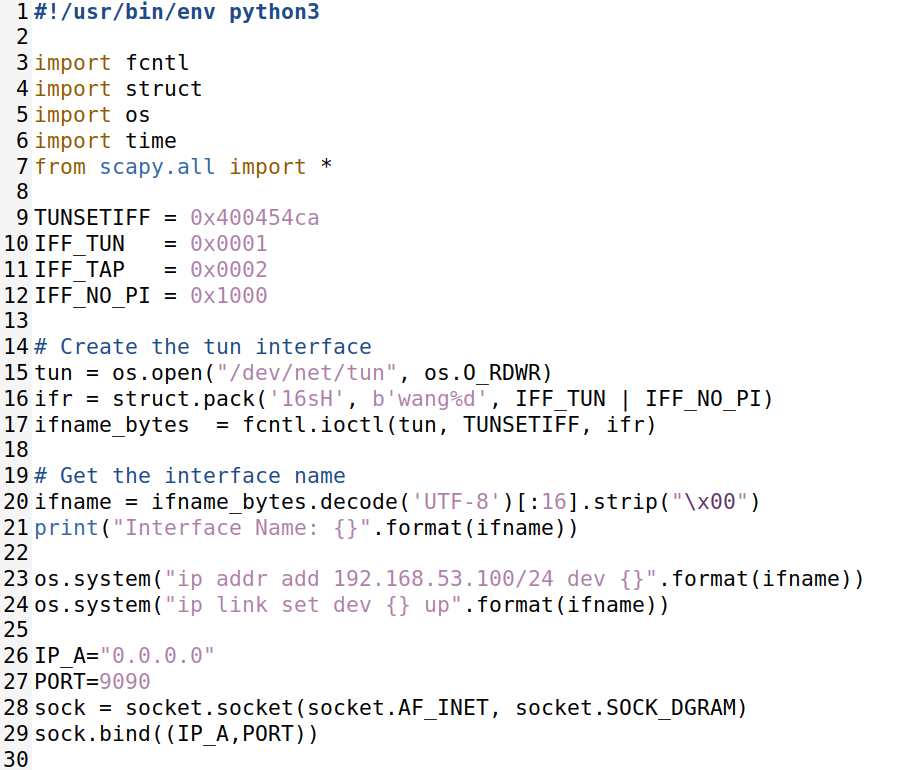


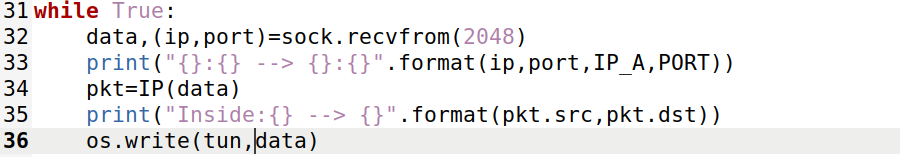
重复上述操作，VPV服务器有输出，说明 tun\_server.py 通过隧道接收到报文，实验成功



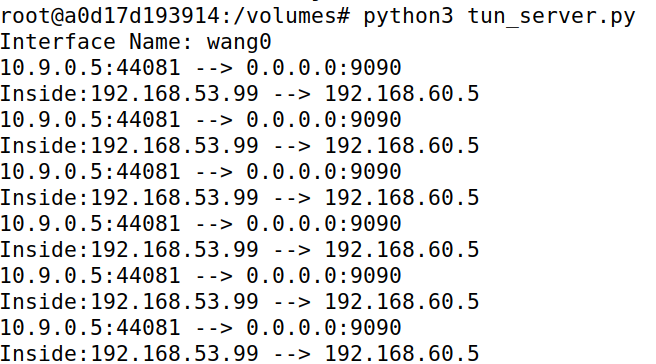
**Task 4: Set Up the VPN Server**

修改tun\_server.py的代码如下

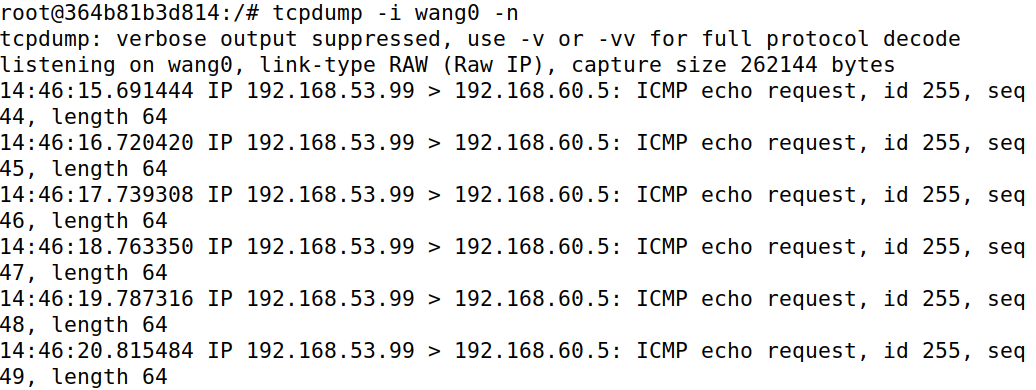




重复task3的操作，在主机U ping 192.168.60.5，VPN服务器输出如下

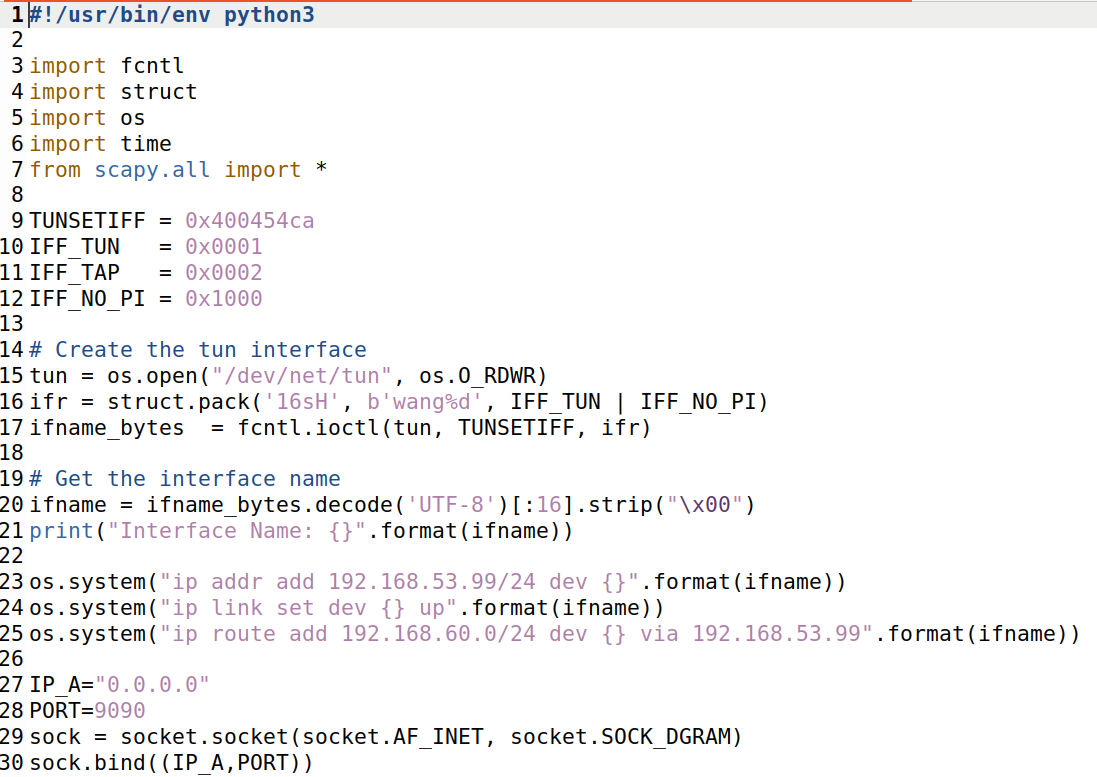


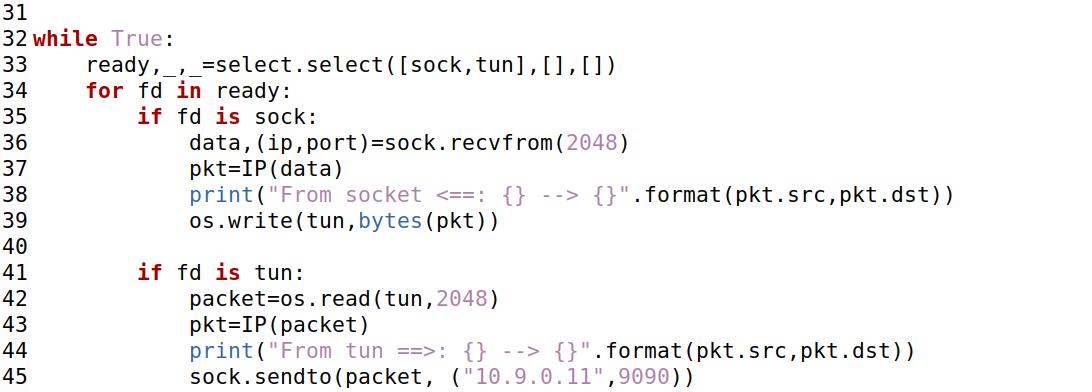
在 VPN 服务器上监听wang0端口，发现ICMP请求包成功通过隧道到达主机V，且收到了主机V的ICMP响应



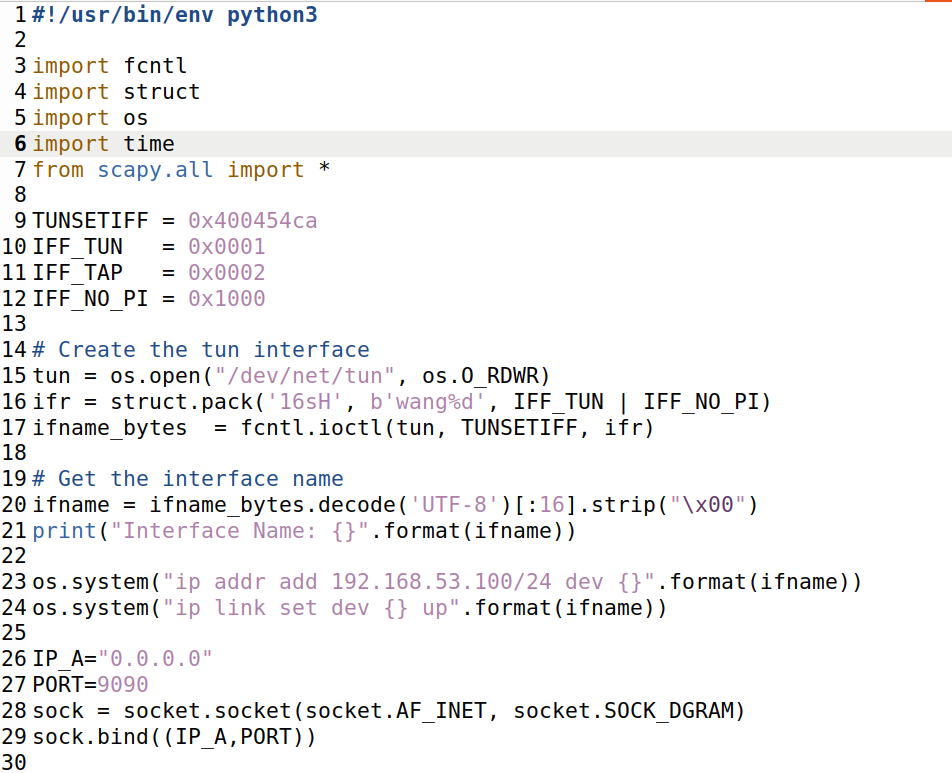
**Task 5: Handling Traffic in Both Directions**

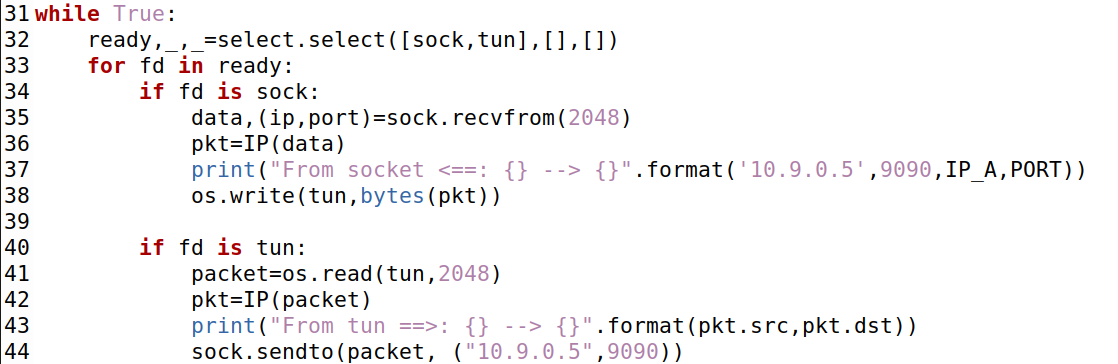
修改tun\_client.py的代码如下



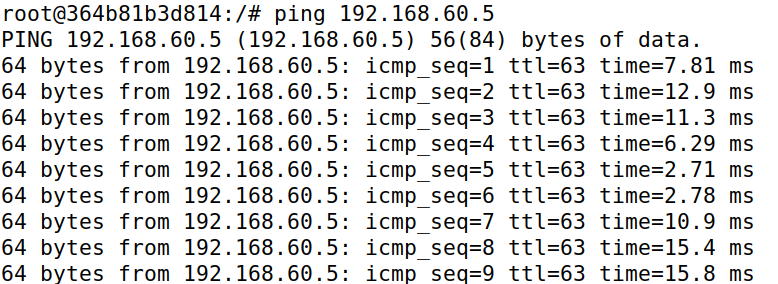


修改tun\_server.py的代码如下

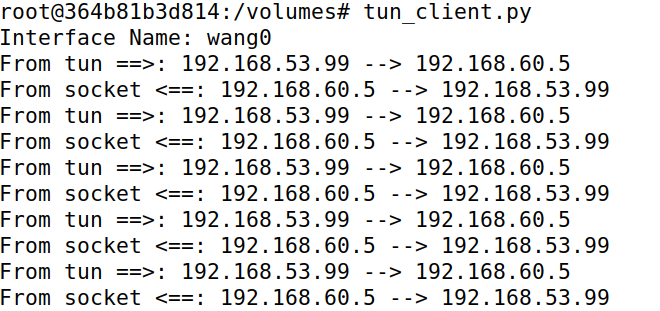




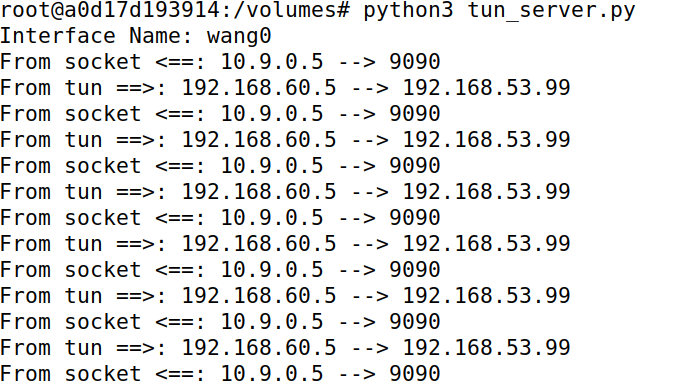
重复task3的操作，在在主机U ping 192.168.60.5，可以ping通



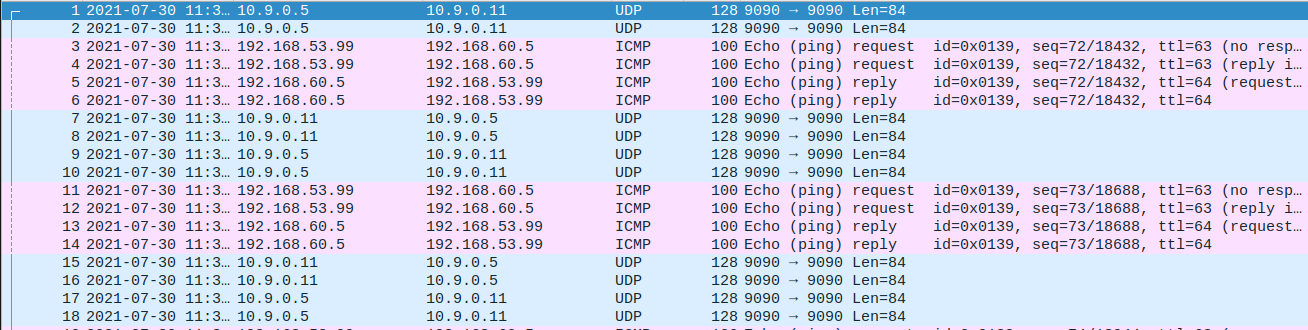
tun\_client.py输出如下



tun\_server.py输出如下

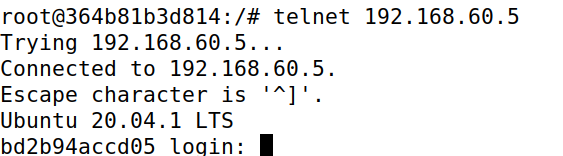


用wireshark捕获的数据包如下



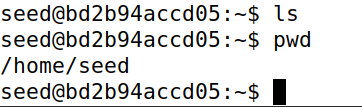
数据报文从主机 U 发向主机 V，报文先通过 tun 到达 VPN 服务器，然后 VPN 服务器通过 tun 发往主机 V 报文，然后主机 V 返回响应报文通过 tun 达到 VPN 服务器，VPN 服务器又通过 tun 将响应报文发给主机 U，从而完成主机 U 和主机 V 之间的通信。

在主机U上telnet主机V，成功



**Task 6: Tunnel-Breaking Experiment**

在主机U上telnet主机V，然后终止程序，发现无法输入任何字符。这是因为终止程序后隧道中断，数据包无法到达



短时间内重新启动运行程序，发现在终止程序时输入的字符出现。这是因为终止程序时的输入会在缓存区中一直发送报文，如果快速恢复连接，前面的输入仍然会显示

