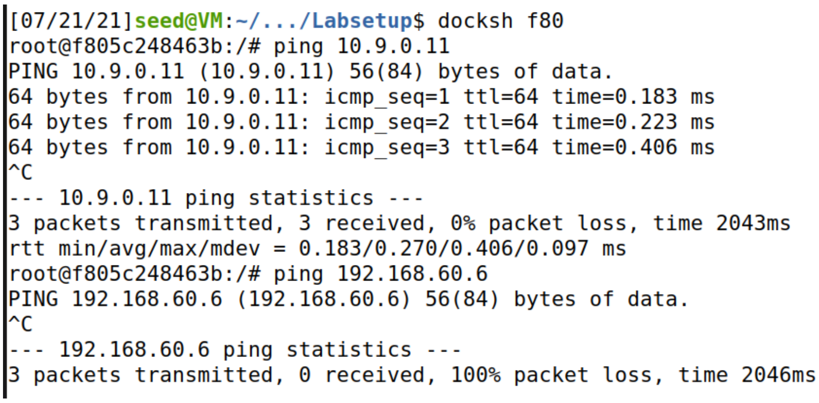
Chapter7 实验报告

57118238 刘欣宇

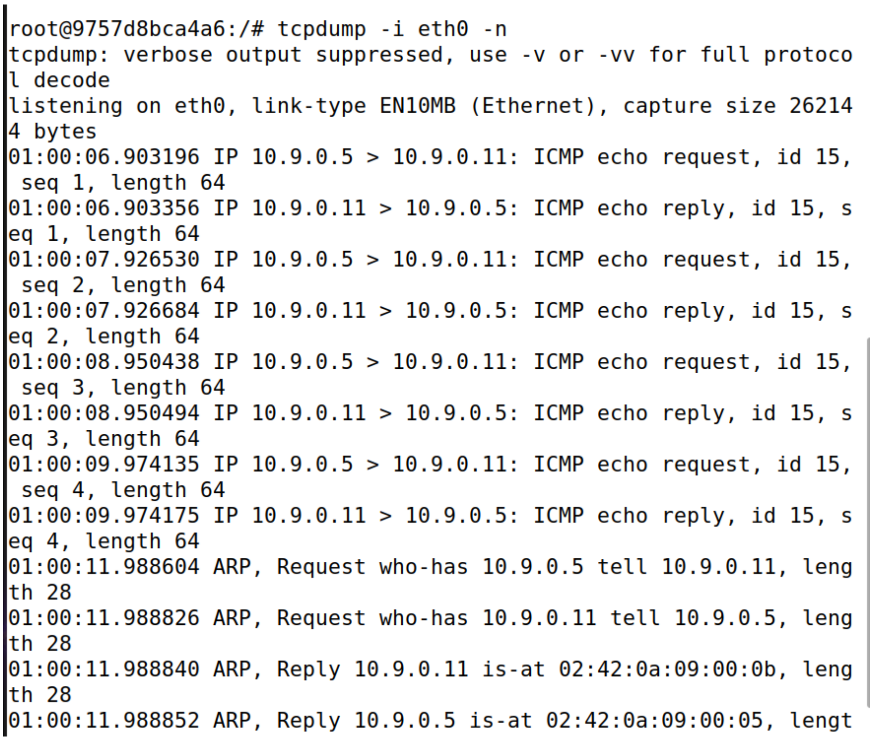
Task1 检测环境配置

Host U 能 ping 通 VPN Server，不能 ping 通 Host V：



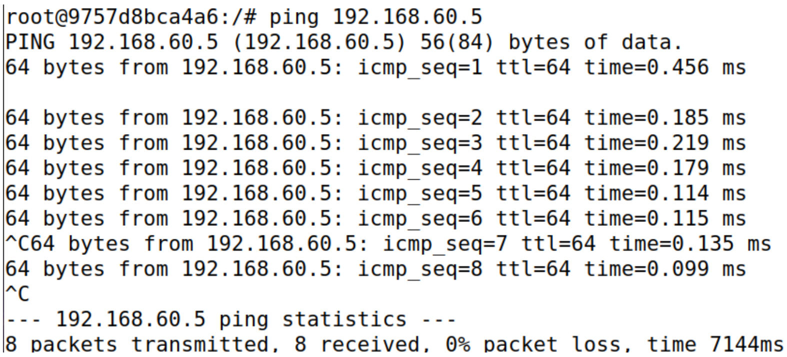
```
[07/21/21]seed@VM:~/.../Labsetup$ docksh f80
root@f805c248463b:/# ping 10.9.0.11
PING 10.9.0.11 (10.9.0.11) 56(84) bytes of data.
64 bytes from 10.9.0.11: icmp_seq=1 ttl=64 time=0.183 ms
64 bytes from 10.9.0.11: icmp_seq=2 ttl=64 time=0.223 ms
64 bytes from 10.9.0.11: icmp_seq=3 ttl=64 time=0.406 ms
^C
--- 10.9.0.11 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2043ms
rtt min/avg/max/mdev = 0.183/0.270/0.406/0.097 ms
root@f805c248463b:/# ping 192.168.60.6
PING 192.168.60.6 (192.168.60.6) 56(84) bytes of data.
^C
--- 192.168.60.6 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2046ms
```

Router 可以利用 tcpdump 抓包：



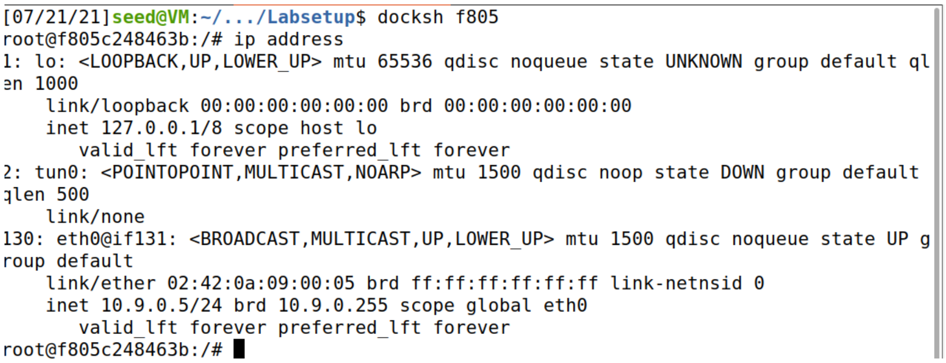
```
root@9757d8bca4a6:/# tcpdump -i eth0 -n
tcpdump: verbose output suppressed, use -v or -vv for full protoco
l decode
listening on eth0, link-type EN10MB (Ethernet), capture size 26214
4 bytes
01:00:06.903196 IP 10.9.0.5 > 10.9.0.11: ICMP echo request, id 15,
 seq 1, length 64
01:00:06.903356 IP 10.9.0.11 > 10.9.0.5: ICMP echo reply, id 15, s
eq 1, length 64
01:00:07.926530 IP 10.9.0.5 > 10.9.0.11: ICMP echo request, id 15,
 seq 2, length 64
01:00:07.926684 IP 10.9.0.11 > 10.9.0.5: ICMP echo reply, id 15, s
eq 2, length 64
01:00:08.950438 IP 10.9.0.5 > 10.9.0.11: ICMP echo request, id 15,
 seq 3, length 64
01:00:08.950494 IP 10.9.0.11 > 10.9.0.5: ICMP echo reply, id 15, s
eq 3, length 64
01:00:09.974135 IP 10.9.0.5 > 10.9.0.11: ICMP echo request, id 15,
 seq 4, length 64
01:00:09.974175 IP 10.9.0.11 > 10.9.0.5: ICMP echo reply, id 15, s
eq 4, length 64
01:00:11.988604 ARP, Request who-has 10.9.0.5 tell 10.9.0.11, leng
th 28
01:00:11.988826 ARP, Request who-has 10.9.0.11 tell 10.9.0.5, leng
th 28
01:00:11.988840 ARP, Reply 10.9.0.11 is-at 02:42:0a:09:00:0b, leng
th 28
01:00:11.988852 ARP, Reply 10.9.0.5 is-at 02:42:0a:09:00:05, lengt
```

Router 可以 ping 通 Host V：



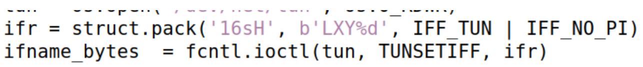
```
root@9757d8bca4a6:/# ping 192.168.60.5
PING 192.168.60.5 (192.168.60.5) 56(84) bytes of data.
64 bytes from 192.168.60.5: icmp_seq=1 ttl=64 time=0.456 ms

64 bytes from 192.168.60.5: icmp_seq=2 ttl=64 time=0.185 ms
64 bytes from 192.168.60.5: icmp_seq=3 ttl=64 time=0.219 ms
64 bytes from 192.168.60.5: icmp_seq=4 ttl=64 time=0.179 ms
64 bytes from 192.168.60.5: icmp_seq=5 ttl=64 time=0.114 ms
64 bytes from 192.168.60.5: icmp_seq=6 ttl=64 time=0.115 ms
^C64 bytes from 192.168.60.5: icmp_seq=7 ttl=64 time=0.135 ms
64 bytes from 192.168.60.5: icmp_seq=8 ttl=64 time=0.099 ms
^C
--- 192.168.60.5 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7144ms
```

Task2.a：

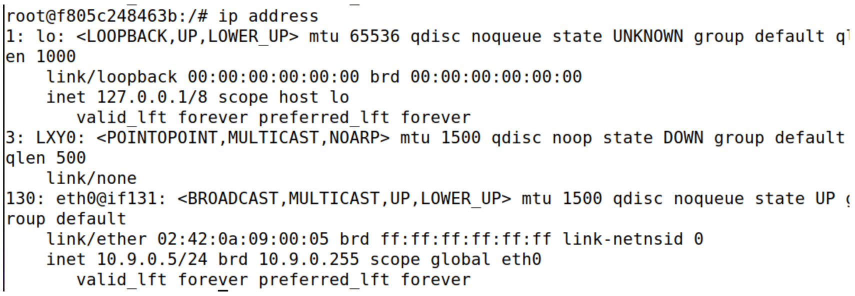


```
[07/21/21]seed@VM:~/.../Labsetup$ docksh f805
root@f805c248463b:/# ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default ql
en 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
2: tun0: <POINTOPOINT,MULTICAST,NOARP> mtu 1500 qdisc noop state DOWN group default
qlen 500
    link/none
130: eth0@if131: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP g
roup default
    link/ether 02:42:0a:09:00:05 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.9.0.5/24 brd 10.9.0.255 scope global eth0
       valid_lft forever preferred_lft forever
root@f805c248463b:/#
```

更改接口名为 LXY0：



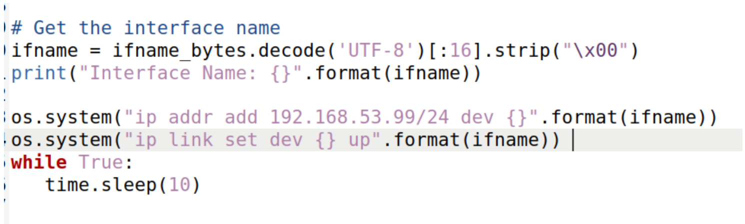
```
ifr = struct.pack('16sH', b'LXY%d', IFF_TUN | IFF_NO_PI)
ifname_bytes  = fcntl.ioctl(tun, TUNSETIFF, ifr)
```



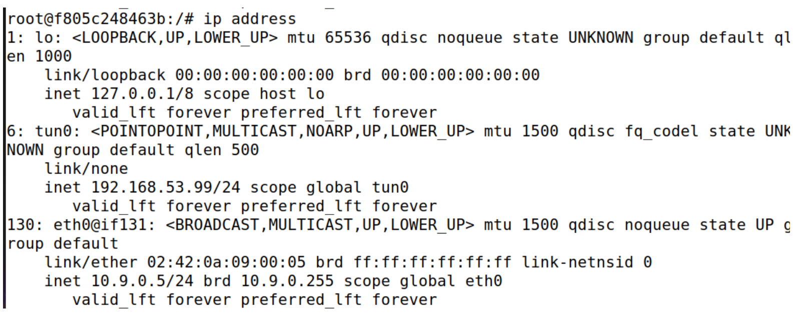
```
root@f805c248463b:/# ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default q
en 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
3: LXY0: <POINTOPOINT,MULTICAST,NOARP> mtu 1500 qdisc noop state DOWN group default
qlen 500
    link/none
130: eth0@if131: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP g
roup default
    link/ether 02:42:0a:09:00:05 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.9.0.5/24 brd 10.9.0.255 scope global eth0
       valid_lft forever preferred_lft forever
```

Task2.b：

具有 ip，接口被开启



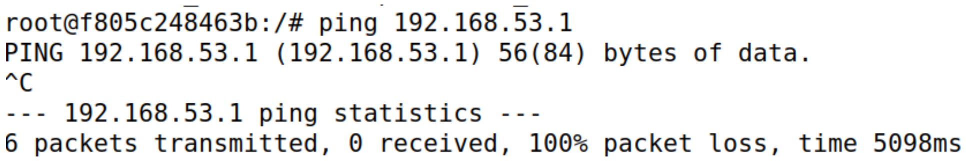
```python
# Get the interface name
ifname = ifname_bytes.decode('UTF-8')[:16].strip("\x00")
print("Interface Name: {}".format(ifname))

os.system("ip addr add 192.168.53.99/24 dev {}".format(ifname))
os.system("ip link set dev {} up".format(ifname))
while True:
   time.sleep(10)
```



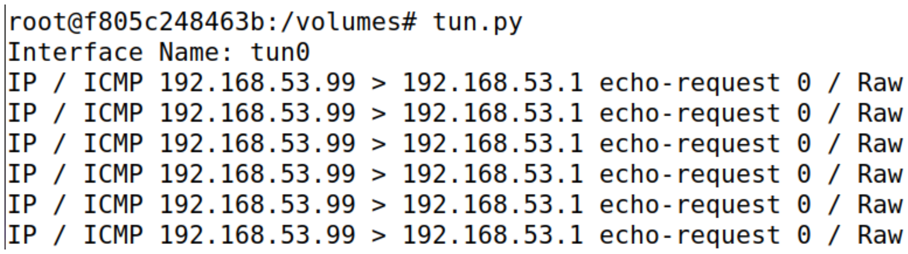
```
root@f805c248463b:/# ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default ql
en 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
6: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNK
NOWN group default qlen 500
    link/none
    inet 192.168.53.99/24 scope global tun0
       valid_lft forever preferred_lft forever
130: eth0@if131: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP g
roup default
    link/ether 02:42:0a:09:00:05 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.9.0.5/24 brd 10.9.0.255 scope global eth0
       valid_lft forever preferred_lft forever
```

Task2.c：

ping 192.168.53.1，发现有输出，因为 tun0 的 ip 网段设置为 192.168.52.0/24，ping 本网段的主机会被网卡端口转发：

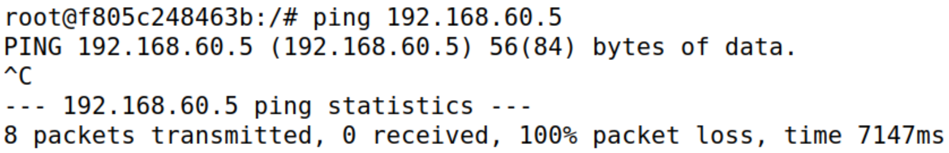


```
root@f805c248463b:/# ping 192.168.53.1
PING 192.168.53.1 (192.168.53.1) 56(84) bytes of data.
^C
--- 192.168.53.1 ping statistics ---
6 packets transmitted, 0 received, 100% packet loss, time 5098ms
```

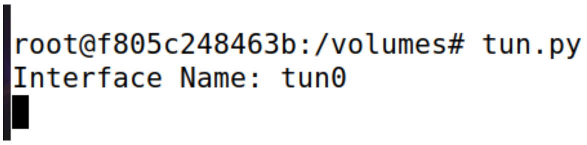


```
root@f805c248463b:/volumes# tun.py
Interface Name: tun0
IP / ICMP 192.168.53.99 > 192.168.53.1 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.1 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.1 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.1 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.1 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.1 echo-request 0 / Raw
```

ping 192.168.60.5，此时没有任何输出，因为 192.168.60.0/24 网段未被 tun0 端口转发，没有报文流过：



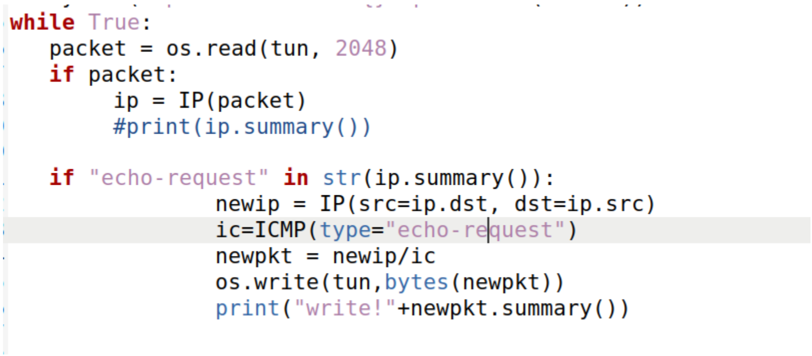
```
root@f805c248463b:/# ping 192.168.60.5
PING 192.168.60.5 (192.168.60.5) 56(84) bytes of data.
^C
--- 192.168.60.5 ping statistics ---
8 packets transmitted, 0 received, 100% packet loss, time 7147ms
```



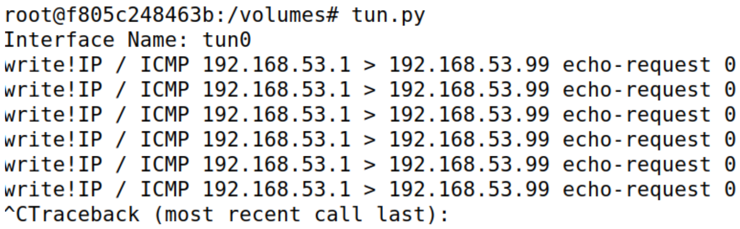
```
root@f805c248463b:/volumes# tun.py
Interface Name: tun0
```

Task2.d：

写入报文：

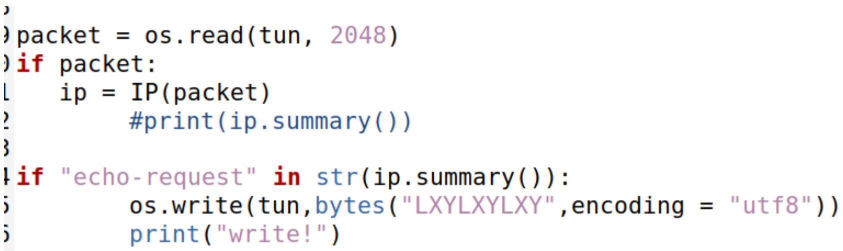


```
while True:
   packet = os.read(tun, 2048)
   if packet:
        ip = IP(packet)
        #print(ip.summary())

   if "echo-request" in str(ip.summary()):
                ic=ICMP(type="echo-request")
                newip = IP(src=ip.dst, dst=ip.src)
                newpkt = newip/ic
                os.write(tun,bytes(newpkt))
                print("write!"+newpkt.summary())
```

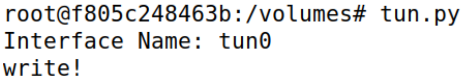


```
root@f805c248463b:/volumes# tun.py
Interface Name: tun0
write!IP / ICMP 192.168.53.1 > 192.168.53.99 echo-request 0
write!IP / ICMP 192.168.53.1 > 192.168.53.99 echo-request 0
write!IP / ICMP 192.168.53.1 > 192.168.53.99 echo-request 0
write!IP / ICMP 192.168.53.1 > 192.168.53.99 echo-request 0
write!IP / ICMP 192.168.53.1 > 192.168.53.99 echo-request 0
write!IP / ICMP 192.168.53.1 > 192.168.53.99 echo-request 0
^CTraceback (most recent call last):
```

写入字符，此时发现写入需要设置 bytes 和 encoding，才不会报错：



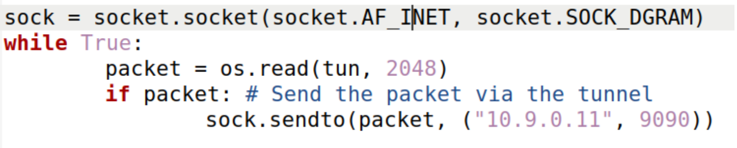
```
packet = os.read(tun, 2048)
if packet:
   ip = IP(packet)
        #print(ip.summary())

if "echo-request" in str(ip.summary()):
        os.write(tun,bytes("LXYLXYLXY",encoding = "utf8"))
        print("write!")
```



```
root@f805c248463b:/volumes# tun.py
Interface Name: tun0
write!
```

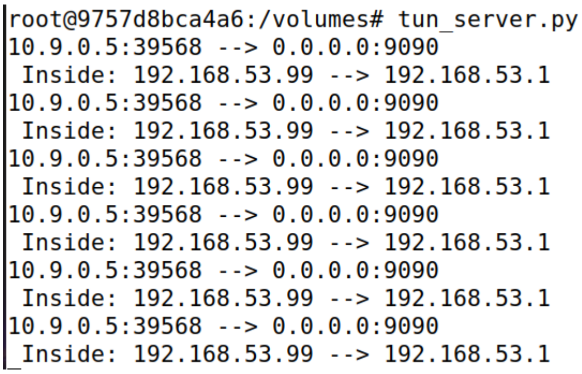
Task3：

Tun_client.py：



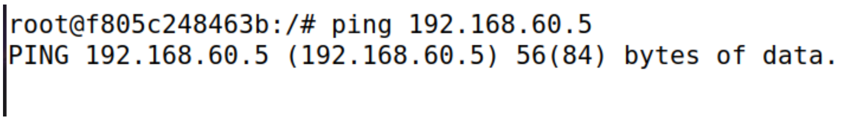
```
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
while True:
        packet = os.read(tun, 2048)
        if packet: # Send the packet via the tunnel
                sock.sendto(packet, ("10.9.0.11", 9090))
```

在 Host U 运行 tun_client.py，router 运行 tun_server.py，Host U ping 192.168.53.1：

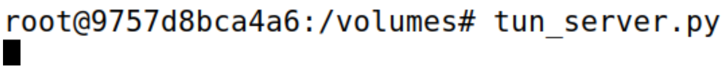


```
root@9757d8bca4a6:/volumes# tun_server.py
10.9.0.5:39568 --> 0.0.0.0:9090
 Inside: 192.168.53.99 --> 192.168.53.1
10.9.0.5:39568 --> 0.0.0.0:9090
 Inside: 192.168.53.99 --> 192.168.53.1
10.9.0.5:39568 --> 0.0.0.0:9090
 Inside: 192.168.53.99 --> 192.168.53.1
10.9.0.5:39568 --> 0.0.0.0:9090
 Inside: 192.168.53.99 --> 192.168.53.1
10.9.0.5:39568 --> 0.0.0.0:9090
 Inside: 192.168.53.99 --> 192.168.53.1
10.9.0.5:39568 --> 0.0.0.0:9090
 Inside: 192.168.53.99 --> 192.168.53.1
```

Ping 192.168.60.5，发现并没有转发：

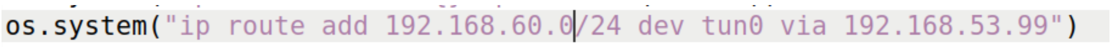


```
root@f805c248463b:/# ping 192.168.60.5
PING 192.168.60.5 (192.168.60.5) 56(84) bytes of data.
```



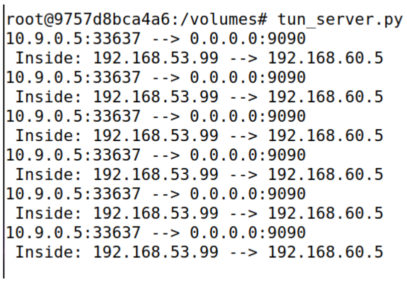
```
root@9757d8bca4a6:/volumes# tun_server.py
```

设置 ip 路由(此处尝试在另外的终端中添加发现添加失效，因为 tun0 动态开启，只能在程序中添加静态路由)：



```
os.system("ip route add 192.168.60.0/24 dev tun0 via 192.168.53.99")
```

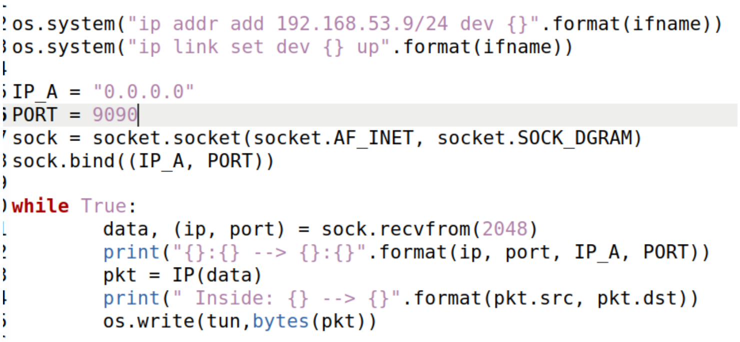
再 Ping 192.168.60.5：



```
root@9757d8bca4a6:/volumes# tun_server.py
10.9.0.5:33637 --> 0.0.0.0:9090
 Inside: 192.168.53.99 --> 192.168.60.5
10.9.0.5:33637 --> 0.0.0.0:9090
 Inside: 192.168.53.99 --> 192.168.60.5
10.9.0.5:33637 --> 0.0.0.0:9090
 Inside: 192.168.53.99 --> 192.168.60.5
10.9.0.5:33637 --> 0.0.0.0:9090
 Inside: 192.168.53.99 --> 192.168.60.5
10.9.0.5:33637 --> 0.0.0.0:9090
 Inside: 192.168.53.99 --> 192.168.60.5
10.9.0.5:33637 --> 0.0.0.0:9090
 Inside: 192.168.53.99 --> 192.168.60.5
```

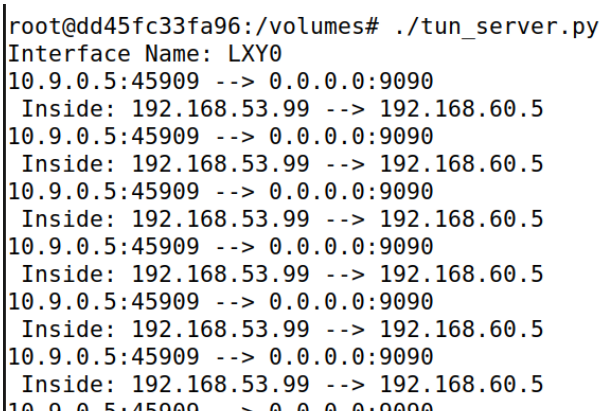
Task4

修改 tun_server.py 的代码如图，并在 tun_client.py 中加入到达 192.168.60.0/24 的路由，此任务中接口名为 LXY0：



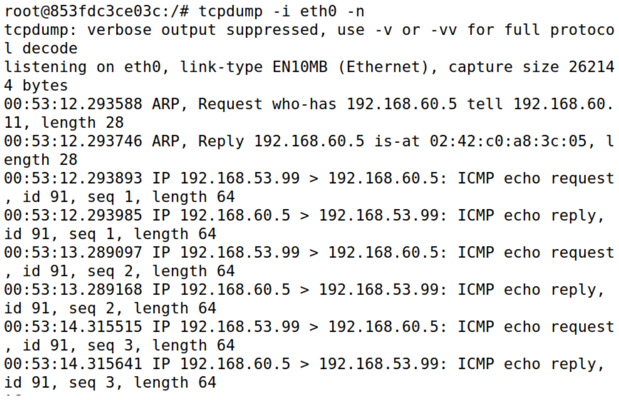
```
os.system("ip addr add 192.168.53.9/24 dev {}".format(ifname))
os.system("ip link set dev {} up".format(ifname))

IP_A = "0.0.0.0"
PORT = 9090
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock.bind((IP_A, PORT))

while True:
        data, (ip, port) = sock.recvfrom(2048)
        print("{}:{} --> {}:{}".format(ip, port, IP_A, PORT))
        pkt = IP(data)
        print(" Inside: {} --> {}".format(pkt.src, pkt.dst))
        os.write(tun,bytes(pkt))
```

Ping 时:



```
root@dd45fc33fa96:/volumes# ./tun_server.py
Interface Name: LXY0
10.9.0.5:45909 --> 0.0.0.0:9090
 Inside: 192.168.53.99 --> 192.168.60.5
10.9.0.5:45909 --> 0.0.0.0:9090
 Inside: 192.168.53.99 --> 192.168.60.5
10.9.0.5:45909 --> 0.0.0.0:9090
 Inside: 192.168.53.99 --> 192.168.60.5
10.9.0.5:45909 --> 0.0.0.0:9090
 Inside: 192.168.53.99 --> 192.168.60.5
10.9.0.5:45909 --> 0.0.0.0:9090
 Inside: 192.168.53.99 --> 192.168.60.5
10.9.0.5:45909 --> 0.0.0.0:9090
 Inside: 192.168.53.99 --> 192.168.60.5
10.9.0.5:45909 --> 0.0.0.0:9090
```

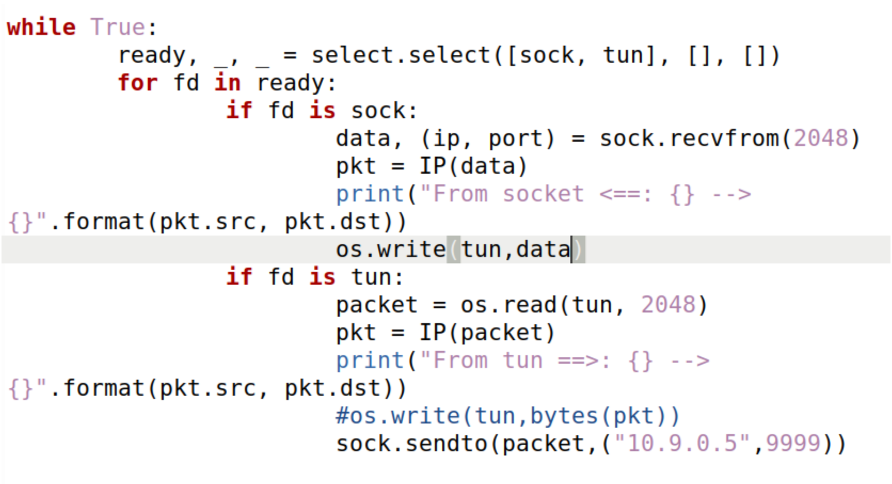
在 192.168.60.5 主机上运行 tcpdump 抓到 ping 报文:



```
root@853fdc3ce03c:/# tcpdump -i eth0 -n
tcpdump: verbose output suppressed, use -v or -vv for full protoco
l decode
listening on eth0, link-type EN10MB (Ethernet), capture size 26214
4 bytes
00:53:12.293588 ARP, Request who-has 192.168.60.5 tell 192.168.60.
11, length 28
00:53:12.293746 ARP, Reply 192.168.60.5 is-at 02:42:c0:a8:3c:05, l
ength 28
00:53:12.293893 IP 192.168.53.99 > 192.168.60.5: ICMP echo request
, id 91, seq 1, length 64
00:53:12.293985 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply,
id 91, seq 1, length 64
00:53:13.289097 IP 192.168.53.99 > 192.168.60.5: ICMP echo request
, id 91, seq 2, length 64
00:53:13.289168 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply,
id 91, seq 2, length 64
00:53:14.315515 IP 192.168.53.99 > 192.168.60.5: ICMP echo request
, id 91, seq 3, length 64
00:53:14.315641 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply,
id 91, seq 3, length 64
```

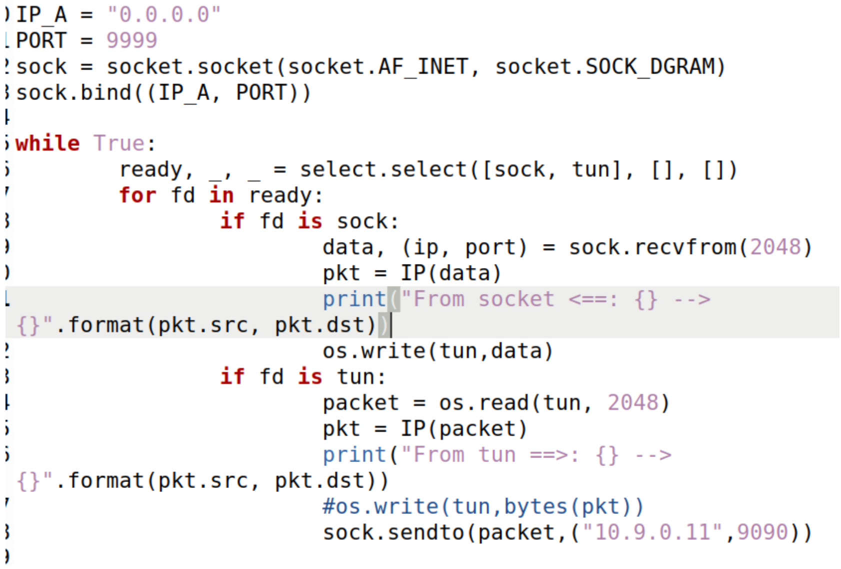
Task5：

Server 端的代码



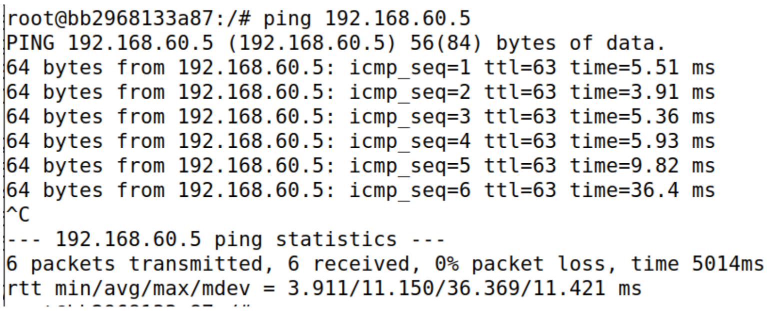
```
while True:
        ready, _, _ = select.select([sock, tun], [], [])
        for fd in ready:
                if fd is sock:
                        data, (ip, port) = sock.recvfrom(2048)
                        pkt = IP(data)
                        print("From socket <==: {} -->
{}".format(pkt.src, pkt.dst))
                        os.write(tun,data)
                if fd is tun:
                        packet = os.read(tun, 2048)
                        pkt = IP(packet)
                        print("From tun ==>: {} -->
{}".format(pkt.src, pkt.dst))
                        #os.write(tun,bytes(pkt))
                        sock.sendto(packet,("10.9.0.5",9999))
```

Client 端的代码：

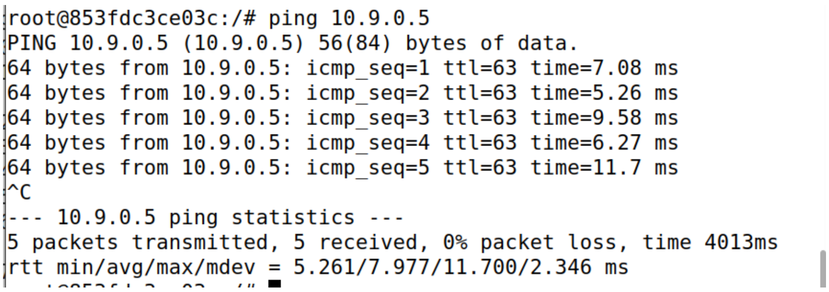


```
IP_A = "0.0.0.0"
PORT = 9999
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock.bind((IP_A, PORT))

while True:
        ready, _, _ = select.select([sock, tun], [], [])
        for fd in ready:
                if fd is sock:
                        data, (ip, port) = sock.recvfrom(2048)
                        pkt = IP(data)
                        print("From socket <==: {} -->
{}".format(pkt.src, pkt.dst))
                        os.write(tun,data)
                if fd is tun:
                        packet = os.read(tun, 2048)
                        pkt = IP(packet)
                        print("From tun ==>: {} -->
{}".format(pkt.src, pkt.dst))
                        #os.write(tun,bytes(pkt))
                        sock.sendto(packet,("10.9.0.11",9090))
```

互相 ping 能够 ping 通：

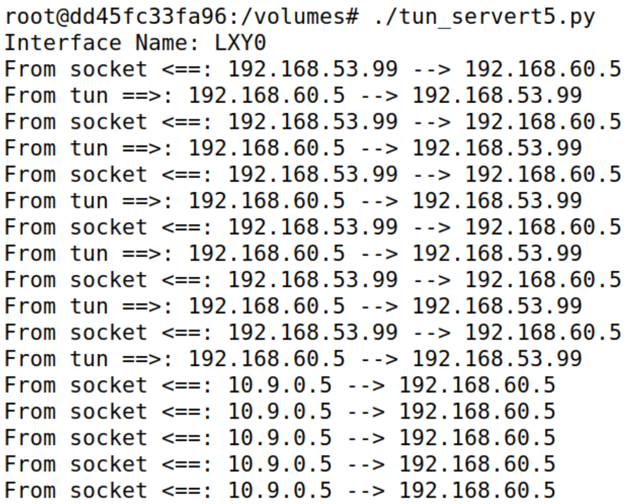


```
root@bb2968133a87:/# ping 192.168.60.5
PING 192.168.60.5 (192.168.60.5) 56(84) bytes of data.
64 bytes from 192.168.60.5: icmp_seq=1 ttl=63 time=5.51 ms
64 bytes from 192.168.60.5: icmp_seq=2 ttl=63 time=3.91 ms
64 bytes from 192.168.60.5: icmp_seq=3 ttl=63 time=5.36 ms
64 bytes from 192.168.60.5: icmp_seq=4 ttl=63 time=5.93 ms
64 bytes from 192.168.60.5: icmp_seq=5 ttl=63 time=9.82 ms
64 bytes from 192.168.60.5: icmp_seq=6 ttl=63 time=36.4 ms
^C
--- 192.168.60.5 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5014ms
rtt min/avg/max/mdev = 3.911/11.150/36.369/11.421 ms
```



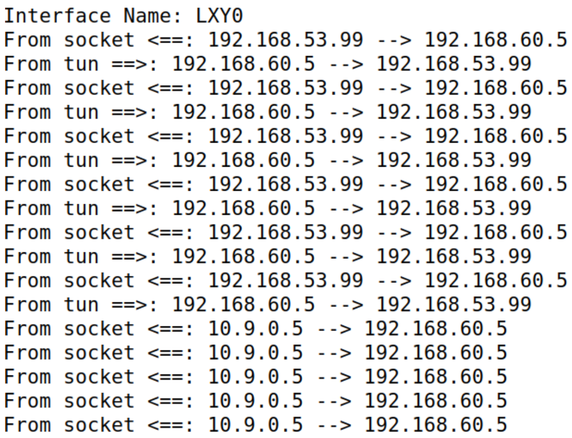
```
root@853fdc3ce03c:/# ping 10.9.0.5
PING 10.9.0.5 (10.9.0.5) 56(84) bytes of data.
64 bytes from 10.9.0.5: icmp_seq=1 ttl=63 time=7.08 ms
64 bytes from 10.9.0.5: icmp_seq=2 ttl=63 time=5.26 ms
64 bytes from 10.9.0.5: icmp_seq=3 ttl=63 time=9.58 ms
64 bytes from 10.9.0.5: icmp_seq=4 ttl=63 time=6.27 ms
64 bytes from 10.9.0.5: icmp_seq=5 ttl=63 time=11.7 ms
^C
--- 10.9.0.5 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4013ms
rtt min/avg/max/mdev = 5.261/7.977/11.700/2.346 ms
```

Server 端:



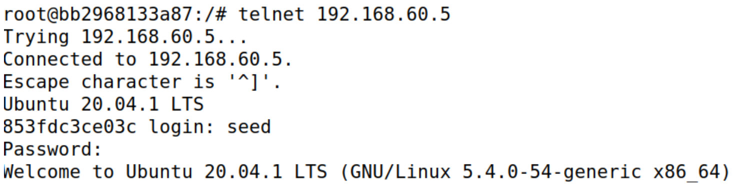
```
root@dd45fc33fa96:/volumes# ./tun_servert5.py
Interface Name: LXY0
From socket <==: 192.168.53.99 --> 192.168.60.5
From tun ==>: 192.168.60.5 --> 192.168.53.99
From socket <==: 192.168.53.99 --> 192.168.60.5
From tun ==>: 192.168.60.5 --> 192.168.53.99
From socket <==: 192.168.53.99 --> 192.168.60.5
From tun ==>: 192.168.60.5 --> 192.168.53.99
From socket <==: 192.168.53.99 --> 192.168.60.5
From tun ==>: 192.168.60.5 --> 192.168.53.99
From socket <==: 192.168.53.99 --> 192.168.60.5
From tun ==>: 192.168.60.5 --> 192.168.53.99
From socket <==: 192.168.53.99 --> 192.168.60.5
From tun ==>: 192.168.60.5 --> 192.168.53.99
From socket <==: 10.9.0.5 --> 192.168.60.5
From socket <==: 10.9.0.5 --> 192.168.60.5
From socket <==: 10.9.0.5 --> 192.168.60.5
From socket <==: 10.9.0.5 --> 192.168.60.5
From socket <==: 10.9.0.5 --> 192.168.60.5
```

Client 端:



```
Interface Name: LXY0
From socket <==: 192.168.53.99 --> 192.168.60.5
From tun ==>: 192.168.60.5 --> 192.168.53.99
From socket <==: 192.168.53.99 --> 192.168.60.5
From tun ==>: 192.168.60.5 --> 192.168.53.99
From socket <==: 192.168.53.99 --> 192.168.60.5
From tun ==>: 192.168.60.5 --> 192.168.53.99
From socket <==: 192.168.53.99 --> 192.168.60.5
From tun ==>: 192.168.60.5 --> 192.168.53.99
From socket <==: 192.168.53.99 --> 192.168.60.5
From tun ==>: 192.168.60.5 --> 192.168.53.99
From socket <==: 192.168.53.99 --> 192.168.60.5
From tun ==>: 192.168.60.5 --> 192.168.53.99
From socket <==: 10.9.0.5 --> 192.168.60.5
From socket <==: 10.9.0.5 --> 192.168.60.5
From socket <==: 10.9.0.5 --> 192.168.60.5
From socket <==: 10.9.0.5 --> 192.168.60.5
From socket <==: 10.9.0.5 --> 192.168.60.5
```

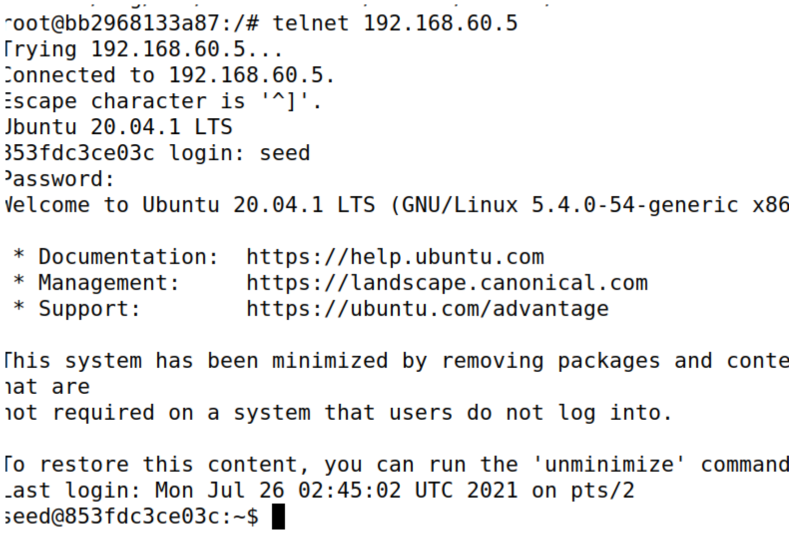
10.9.0.5 能够 telnet 成功内网主机:



```
root@bb2968133a87:/# telnet 192.168.60.5
Trying 192.168.60.5...
Connected to 192.168.60.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
853fdc3ce03c login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)
```

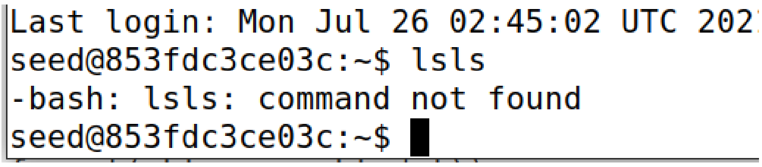
Task6:

当 telnet 连接建立，再断开 tunnel 时，发现输入没有响应，说明连接已经断开了：



```
·oot@bb2968133a87:/# telnet 192.168.60.5
Trying 192.168.60.5...
Connected to 192.168.60.5.
Escape character is '^]'.
Jbuntu 20.04.1 LTS
353fdc3ce03c login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

This system has been minimized by removing packages and conte
hat are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command
Last login: Mon Jul 26 02:45:02 UTC 2021 on pts/2
seed@853fdc3ce03c:~$ 
```

短时间内重新建立连接后，发现 telnet 重新被接通：此前的输入也被显示：



```
Last login: Mon Jul 26 02:45:02 UTC 202
seed@853fdc3ce03c:~$ lsls
-bash: lsls: command not found
seed@853fdc3ce03c:~$ 
```