57117216 殷广成

Task1：

```
[09/22/20]seed@VM:~$ ifconfig
ens33     Link encap:Ethernet  HWaddr 00:0c:29:99:36:68
          inet addr:192.168.220.129  Bcast:192.168.220.255  Mask:255.255.255.0
          inet6 addr: fe80::87c5:5446:9a64:9ea7/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2565 errors:0 dropped:0 overruns:0 frame:0
          TX packets:384 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:415089 (415.0 KB)  TX bytes:34450 (34.4 KB)
          Interrupt:19 Base address:0x2000
```

```
[09/22/20]seed@VM:~$ ifconfig
ens33     Link encap:Ethernet  HWaddr 00:0c:29:14:18:9a
          inet addr:192.168.220.133  Bcast:192.168.220.255  Mask:255.255.255.0
          inet6 addr: fe80::de98:8a3a:c686:99c6/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1166 errors:0 dropped:0 overruns:0 frame:0
          TX packets:362 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:104828 (104.8 KB)  TX bytes:38122 (38.1 KB)
          Interrupt:19 Base address:0x2000

ens38     Link encap:Ethernet  HWaddr 00:0c:29:14:18:a4
          inet addr:10.42.0.100  Bcast:10.42.0.255  Mask:255.255.255.0
          inet6 addr: fe80::f0e1:ab1a:8c68:74d0/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:490 errors:0 dropped:0 overruns:0 frame:0
          TX packets:396 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:75700 (75.7 KB)  TX bytes:64331 (64.3 KB)
          Interrupt:16 Base address:0x2080
```

```
[09/22/20]seed@VM:~$ ifconfig
ens33     Link encap:Ethernet  HWaddr 00:0c:29:73:43:db
          inet addr:10.42.0.101  Bcast:10.42.0.255  Mask:255.255.255.0
          inet6 addr: fe80::f517:6227:78a7:2efa/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:646 errors:0 dropped:0 overruns:0 frame:0
          TX packets:520 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:79142 (79.1 KB)  TX bytes:78924 (78.9 KB)
          Interrupt:19 Base address:0x2000
```

Vpn 主机、服务器以及内网主机的信息如上

```
[09/22/20]seed@VM:~$ ping 10.42.0.101
PING 10.42.0.101 (10.42.0.101) 56(84) bytes of data.
From 10.42.0.1 icmp_seq=3 Destination Host Unreachable
^C
--- 10.42.0.101 ping statistics ---
8 packets transmitted, 0 received, +1 errors, 100% packet loss
```

但是内外两个主机不能相互链接。

Task2.a

运行脚本，可以添加一个新的网卡 tun0，但是此时该网卡没有 IP 地址，因此

```
[09/22/20]seed@VM:~/code$ sudo python3 tun.py
Interface Name: tun0
```

```
[09/22/20]seed@VM:~/code$ ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
    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
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:0c:29:99:36:68 brd ff:ff:ff:ff:ff:ff
    inet 192.168.220.129/24 brd 192.168.220.255 scope global dynamic ens33
       valid_lft 1689sec preferred_lft 1689sec
    inet6 fe80::87c5:5446:9a64:9ea7/64 scope link
       valid_lft forever preferred_lft forever
3: tun0: <POINTOPOINT,MULTICAST,NOARP> mtu 1500 qdisc noop state DOWN group default qlen 500
    link/none
```

Task2.b

```
[09/22/20]seed@VM:~/code$ sudo python3 tun.py
Interface Name: tun0
```

```
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP grou
    link/ether 00:0c:29:99:36:68 brd ff:ff:ff:ff:ff:ff
    inet 192.168.220.129/24 brd 192.168.220.255 scope global dynamic ens33
       valid_lft 1217sec preferred_lft 1217sec
    inet6 fe80::87c5:5446:9a64:9ea7/64 scope link
       valid_lft forever preferred_lft forever
5: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state
    link/none
    inet 192.168.53.99/24 scope global tun0
       valid_lft forever preferred_lft forever
    inet6 fe80::ae8e:344d:ef42:2df5/64 scope link flags 800
       valid_lft forever preferred_lft forever
```

设置了新网卡的一系列参数

Task2.c

```
###[ IP ]###
  version   = 4
  ihl       = 5
  tos       = 0x0
  len       = 84
  id        = 33024
  flags     = DF
  frag      = 0
  ttl       = 64
  proto     = icmp
  chksum    = 0xcd90
  src       = 192.168.53.99
  dst       = 192.168.53.100
  \options   \
###[ ICMP ]###
     type      = echo-request
     code      = 0
     chksum    = 0x3517
     id        = 0x1eb9
     seq       = 0x2
###[ Raw ]###
        load      = '.fi_\x15e\x0c\x00\x08\t\n\x0b\x0c\r\x0e\x0f'
```

```
[09/22/20]seed@VM:~/code$ ping 192.168.53.100
PING 192.168.53.100 (192.168.53.100) 56(84) bytes of data.
^C
--- 192.168.53.100 ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1010ms
```

一条从 192.168.53.99 发往 192.168.53.100 的报文。因为 ping 主机属于 192.168.53.0/24 网段，因此报文从 tun0 网卡发出，而程序监听了 tun0 网卡上的所有报文，所以此 ICMP request 报文被打印出来。

```
^CTraceback (most recent call last):
  File "tun.py", line 23, in <module>
    packet = os.read(tun, 2048)
KeyboardInterrupt
[09/22/20]seed@VM:~/code$
```

```
[09/22/20]seed@VM:~/code$ ping 192.168.60.100
PING 192.168.60.100 (192.168.60.100) 56(84) bytes of data.
^C
--- 192.168.60.100 ping statistics ---
7 packets transmitted, 0 received, 100% packet loss, time 6111ms
```

```
^CTraceback (most recent call last):
  File "tun.py", line 23, in <module>
    packet = os.read(tun, 2048)
KeyboardInterrupt
[09/22/20]seed@VM:~/code$
```

```
[09/22/20]seed@VM:~/code$ ping 192.168.60.100
PING 192.168.60.100 (192.168.60.100) 56(84) bytes of data.
^C
--- 192.168.60.100 ping statistics ---
7 packets transmitted, 0 received, 100% packet loss, time 6111ms
```

而此时该报文不经过 192.168.53.0/24 网段，因此不通过 tun0 网卡发送。

Task2.d

利用 ping 程序发送请求 IP 包，当在 tun0 网卡上捕获到此请求包后，复制其中的负载以构成一个新的 IP 数据包发送出去。源地址为 1.2.3.4，目的地址是 192.168.53.99。

```
while True:
    packet = os.read(tun, 2048)
    if True:
        ip = IP(packet)
        ip.show()
        newip = IP(src='1.2.3.4', dst=ip.src)
        newpkt = newip/ip.payload
        os.write(tun, bytes(newpkt))
```

```
192.168.53.99    192.168.53.100   ICMP      84 Echo (ping) request  id=0x22f1,
1.2.3.4          192.168.53.99    ICMP      84 Echo (ping) request  id=0x22f1,
```

```
while True:
    packet = os.read(tun, 2048)
    if True:
        ip = IP(packet)
        ip.show()
        #newip = IP(src='1.2.3.4', dst=ip.src)
        #newpkt = newip/ip.payload
        #kxx = bytes(newpkt)
        test = b"what"
        os.write(tun,test)
```

改为写入任意负载

```
Traceback (most recent call last):
  File "tun.py", line 32, in <module>
    os.write(tun,test)
OSError: [Errno 22] Invalid argument
```

不能正常发送

Task3

服务器先解析报文的外层，解析完外层后会解析负载，而负载又是另一个 IP 报文，因此会解析另一个 IP 报文。

```
[09/22/20]seed@VM:~/code$ ls
tun_server.py
[09/22/20]seed@VM:~/code$ ./tun_server.py
192.168.220.129:59406-->0.0.0.0:9090
          Inside: 0.0.0.0-->232.83.169.5
192.168.220.129:59406-->0.0.0.0:9090
          Inside: 0.0.0.0-->232.83.169.5
192.168.220.129:59406-->0.0.0.0:9090
          Inside: 0.0.0.0-->232.83.169.5
192.168.220.129:59406-->0.0.0.0:9090
          Inside: 192.168.53.99-->192.168.53.200
192.168.220.129:59406-->0.0.0.0:9090
          Inside: 192.168.53.99-->192.168.53.200
192.168.220.129:59406-->0.0.0.0:9090
          Inside: 192.168.53.99-->192.168.53.200
192.168.220.129:59406-->0.0.0.0:9090
          Inside: 192.168.53.99-->192.168.53.200
```

修改路由设置，然后 ping 10.42.0.101，原本会因为没经过 192 那个网段而无法被 tun0 转发，现在可以成功处理了。

```
seed@VM:~/code$ sudo route add -net 10.42.0.0/24 tun0
```

```
[09/22/20]seed@VM:~/code$ ping 10.42.0.101
PING 10.42.0.101 (10.42.0.101) 56(84) bytes of data.
^C
--- 10.42.0.101 ping statistics ---
```

```
192.168.220.129:59406-->0.0.0.0:9090
          Inside: 192.168.53.99-->10.42.0.101
192.168.220.129:59406-->0.0.0.0:9090
          Inside: 192.168.53.99-->10.42.0.101
192.168.220.129:59406-->0.0.0.0:9090
          Inside: 192.168.53.99-->10.42.0.101
192.168.220.129:59406-->0.0.0.0:9090
          Inside: 192.168.53.99-->10.42.0.101
```

Task4

服务器代码：

```
import fcntl
import struct
import os
import time
from scapy.all import *
from os import write

TUNSETIFF = 0x400454ca
IFF_TUN = 0x0001
IFF_TAP = 0x0002
IFF_NO_PI = 0x1000

tun = os.open("/dev/net/tun", os.O_RDWR)
ifr = struct.pack("16sH", b"tun%d", IFF_TUN | IFF_NO_PI)
ifname_bytes = fcntl.ioctl(tun, TUNSETIFF, ifr)

ifname = ifname_bytes.decode("UTF-8")[:16].strip("\x00")
print("Interface Name: {}".format(ifname))

os.system("ip addr add 192.168.53.100/24 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(data))
```

```
.5262522… 192.168.53.99     10.42.0.101      ICMP    98 Echo (ping) request  id=0x42f3
.5262769… 10.42.0.101       192.168.53.99    ICMP    98 Echo (ping) reply    id=0x42f3
.5506890… 192.168.53.99     10.42.0.101      ICMP    98 Echo (ping) request  id=0x42f3
.5507143… 10.42.0.101       192.168.53.99    ICMP    98 Echo (ping) reply    id=0x42f3
```

```
/bin/bash
/bin/bash 80x23
[09/22/20]seed@VM:~$ ifconfig
ens33     Link encap:Ethernet  HWaddr 00:0c:29:73:43:db
          inet addr:10.42.0.101  Bcast:10.42.0.255  Mask:255.255.255.0
          inet6 addr: fe80::f517:6227:78a7:2efa/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1175 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1051 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:131212 (131.2 KB)  TX bytes:125983 (125.9 KB)
          Interrupt:19 Base address:0x2000
```

两个主机互相 ping。

Task5

```python
import fcntl
import struct
import os
import time
from scapy.all import *
from os import write

TUNSETIFF = 0x400454ca
IFF_TUN = 0x0001
IFF_TAP = 0x0002
IFF_NO_PI = 0x1000

tun = os.open("/dev/net/tun", os.O_RDWR)
ifr = struct.pack("16sH", b"tun%d", IFF_TUN | IFF_NO_PI)
ifname_bytes = fcntl.ioctl(tun, TUNSETIFF, ifr)

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))

sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

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, bytes(pkt))
        if fd is tun:
            packet = os.read(tun, 2048)
            pkt = IP(packet)
            print("From tun: {}-->{}".format(pkt.src, pkt.dst))
            sock.sendto(bytes(pkt), ("192.168.220.133", 9090))
```

```
From tun: 192.168.53.99-->10.42.0.101
From socket: 10.42.0.101-->192.168.53.99
From tun: 192.168.53.99-->10.42.0.101
From socket: 10.42.0.101-->192.168.53.99
From tun: 192.168.53.99-->10.42.0.101
From socket: 10.42.0.101-->192.168.53.99
^CTraceback (most recent call last):
  File "./tun_client.py", line 28, in <module>
    ready, _, _ = select.select([sock, tun], [], [])
KeyboardInterrupt
[09/22/20]seed@VM:~/code$
```

```
8 packets transmitted, 0 received, 100% packet loss, time 7122ms

[09/22/20]seed@VM:~/code$ sudo route add -net 10.42.0.0/24 tun
SIOCADDRT: No such device
[09/22/20]seed@VM:~/code$ sudo route add -net 10.42.0.0/24 tun0
[09/22/20]seed@VM:~/code$ ping 10.42.0.101
PING 10.42.0.101 (10.42.0.101) 56(84) bytes of data.
64 bytes from 10.42.0.101: icmp_seq=1 ttl=63 time=12.4 ms
64 bytes from 10.42.0.101: icmp_seq=2 ttl=63 time=3.45 ms
```

```
[09/23/20]seed@VM:~$ telnet 10.42.0.101
Trying 10.42.0.101...
Connected to 10.42.0.101.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
VM login: seed
Password:
Last login: Thu Sep 17 07:23:38 EDT 2020 from 192.168.220.133 on pts/4
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

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

1 package can be updated.
0 updates are security updates.

[09/23/20]seed@VM:~$ ifconfig
ens33     Link encap:Ethernet  HWaddr 00:0c:29:73:43:db
          inet addr:10.42.0.101  Bcast:10.42.0.255  Mask:255.255.255.0
          inet6 addr: fe80::f517:6227:78a7:2efa/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:115 errors:0 dropped:0 overruns:0 frame:0
          TX packets:241 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:8633 (8.6 KB)  TX bytes:20637 (20.6 KB)
          Interrupt:19 Base address:0x2000
```

主机二 telnet 主机一

按照 vpn 发送报文的顺序查看每个机器的报文：

Source	Destination	Protocol	Length	Info
192.168.53.99	10.42.0.101	ICMP	84	Echo (ping) request
10.42.0.101	192.168.53.99	ICMP	84	Echo (ping) reply

主机一 tun0

Source	Destination	Protocol	Length	Info
192.168.220.129	192.168.220.133	UDP	126	47418 → 9090 Len=84
192.168.220.133	192.168.220.129	UDP	126	9090 → 47418 Len=84
Vmware_99:36:68	Vmware_14:18:9a	ARP	42	Who has 192.168.220.1

主机一 ens33

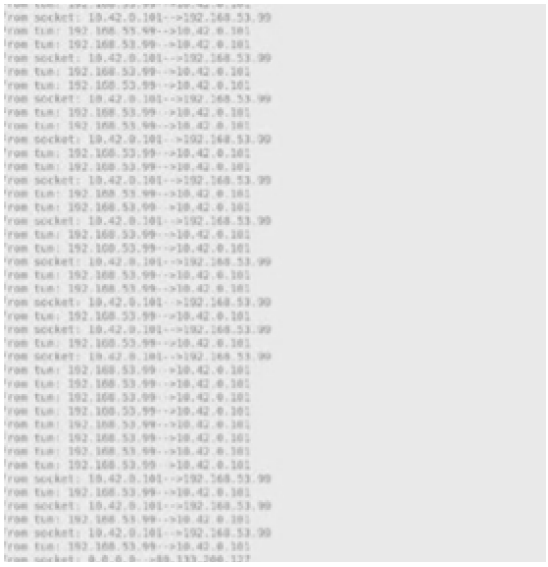
192.168.220.129	192.168.220.133	UDP	126	47418 → 9090 Len=84
192.168.220.133	192.168.220.129	UDP	126	9090 → 47418 Len=84
Vmware_c0:00:08	Broadcast	ARP	60	Who has 192.168.220.2?

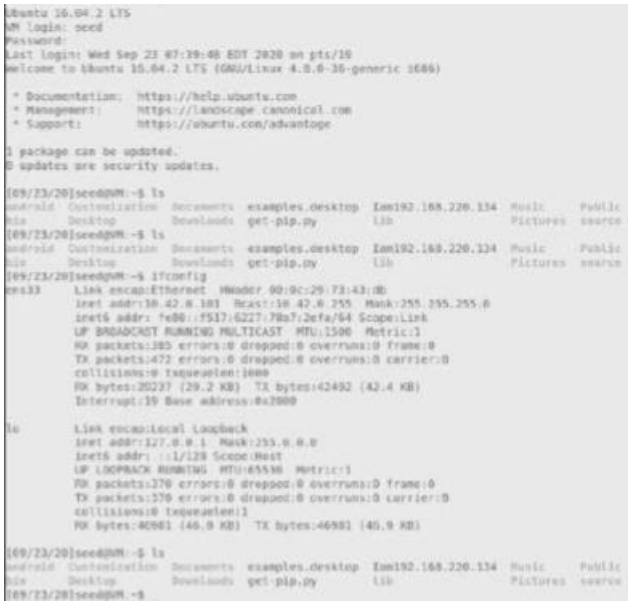
192.168.53.99	10.42.0.101	ICMP	84	Echo (ping) request
10.42.0.101	192.168.53.99	ICMP	84	Echo (ping) reply

192.168.53.99	10.42.0.101	ICMP	98	Echo (ping) request
10.42.0.101	192.168.53.99	ICMP	98	Echo (ping) reply

服务器 ens33，tun0，ens38

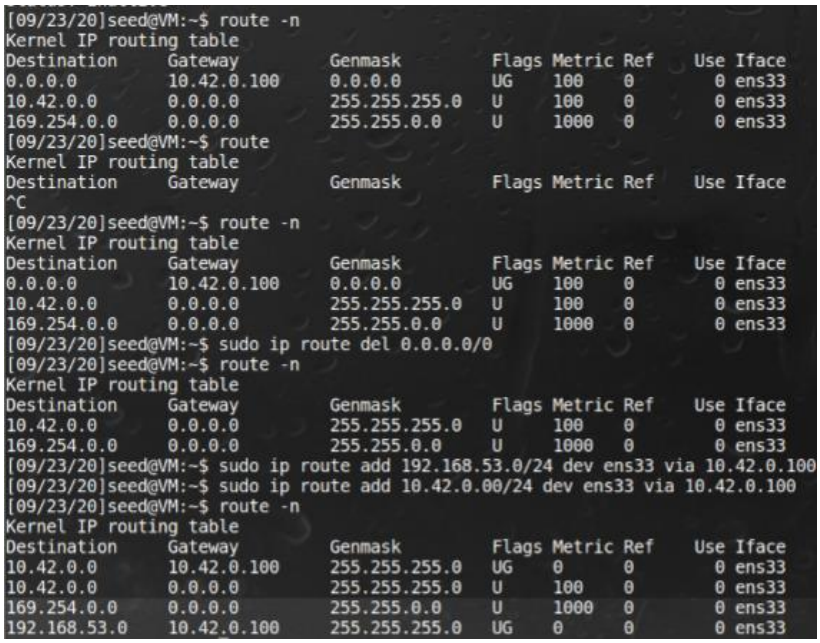
Task6





在保持连接时将 vpnserver 关闭，两方的报文会根据 tcp 协议不停重传，然后随着 vpnserver 再次启动而完成各自的功能。

Task7



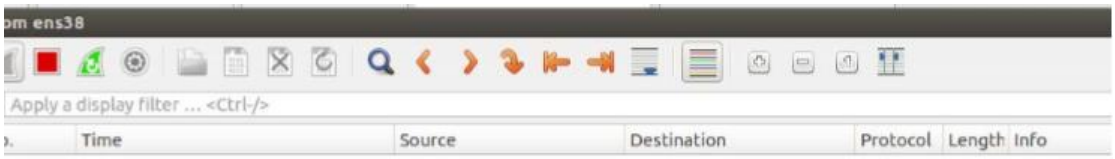
```
[09/23/20]seed@VM:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         10.42.0.100     0.0.0.0         UG    100    0        0 ens33
10.42.0.0       0.0.0.0         255.255.255.0   U     100    0        0 ens33
169.254.0.0     0.0.0.0         255.255.0.0     U     1000   0        0 ens33
[09/23/20]seed@VM:~$ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
^C
[09/23/20]seed@VM:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         10.42.0.100     0.0.0.0         UG    100    0        0 ens33
10.42.0.0       0.0.0.0         255.255.255.0   U     100    0        0 ens33
169.254.0.0     0.0.0.0         255.255.0.0     U     1000   0        0 ens33
[09/23/20]seed@VM:~$ sudo ip route del 0.0.0.0/0
[09/23/20]seed@VM:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.42.0.0       0.0.0.0         255.255.255.0   U     100    0        0 ens33
169.254.0.0     0.0.0.0         255.255.0.0     U     1000   0        0 ens33
[09/23/20]seed@VM:~$ sudo ip route add 192.168.53.0/24 dev ens33 via 10.42.0.100
[09/23/20]seed@VM:~$ sudo ip route add 10.42.0.00/24 dev ens33 via 10.42.0.100
[09/23/20]seed@VM:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.42.0.0       10.42.0.100     255.255.255.0   UG    0      0        0 ens33
10.42.0.0       0.0.0.0         255.255.255.0   U     100    0        0 ens33
169.254.0.0     0.0.0.0         255.255.0.0     U     1000   0        0 ens33
192.168.53.0    10.42.0.100     255.255.255.0   UG    0      0        0 ens33
```

Task8

```
os.system("ip addr add 192.168.60.99/24 dev {}".format(ifname))
os.system("ip link set dev {} up".format(ifname))
```

将发送内容修改，将网卡地址（主机 tun0 和服务器 tun0）各自改为不同网段。

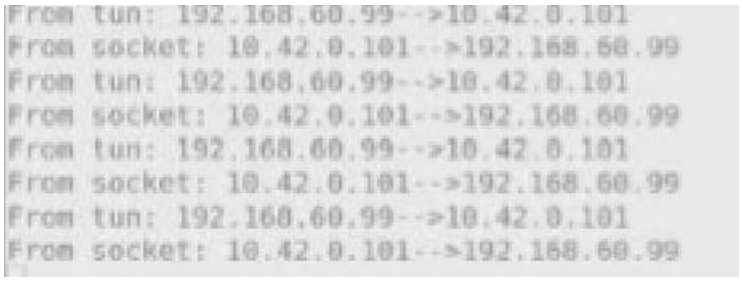
	Time	Source	Destination	Protocol	Length	Info
1	2020-09-23 08:04:38.4591347…	192.168.60.99	10.42.0.101	ICMP	84	Echo (ping) request
2	2020-09-23 08:04:39.4838189…	192.168.60.99	10.42.0.101	ICMP	84	Echo (ping) request
3	2020-09-23 08:04:40.5075394…	192.168.60.99	10.42.0.101	ICMP	84	Echo (ping) request
4	2020-09-23 08:04:41.5310473…	192.168.60.99	10.42.0.101	ICMP	84	Echo (ping) request
5	2020-09-23 08:04:42.5593835…	192.168.60.99	10.42.0.101	ICMP	84	Echo (ping) request
6	2020-09-23 08:04:43.5916741…	192.168.60.99	10.42.0.101	ICMP	84	Echo (ping) request

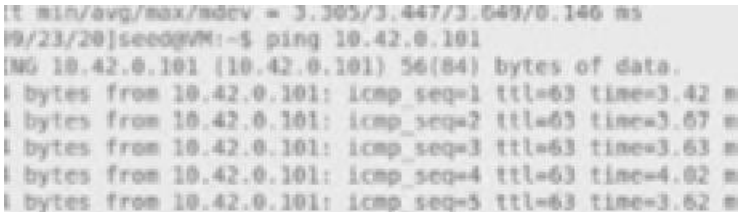


发现服务器上 tun0ping 不到 ens38，原因是没有相应的路由。

```
[09/23/20]seed@VM:~$ sudo sysctl -w net.ipv4.conf.all.rp_filter=0
net.ipv4.conf.all.rp_filter = 0
[09/23/20]seed@VM:~$ sudo sysctl -w net.ipv4.conf.default.rp_filter=0
net.ipv4.conf.default.rp_filter = 0
[09/23/20]seed@VM:~$ sudo route add -net 192.168.60.0/24 tun0
SIOCADDRT: File exists
```

修改路由，关闭反向检查





Ping 成功

Task9

```python
import fcntl
import struct
import os
import time
from scapy.all import *
from os import write

TUNSETIFF = 0x400454ca
IFF_TUN = 0x0001
IFF_TAP = 0x0002
IFF_NO_PI = 0x1000

tap = os.open("/dev/net/tun", os.O_RDWR)
ifr = struct.pack("16sH", b"tap%d", IFF_TAP | IFF_NO_PI)
ifname_bytes = fcntl.ioctl(tap, TUNSETIFF, ifr)
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:
    packet = os.read(tap, 2048)
    if True:
        ether = Ether(packet)
        ether.show()
```

```
tap0      Link encap:Ethernet  HWaddr 86:68:42:66:9e:46
          inet addr:192.168.53.99  Bcast:0.0.0.0  Mask:255.255.255.0
          inet6 addr: fe80::8468:42ff:fe66:9e46/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:39 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:4740 (4.7 KB)
```

创建网卡并修改网卡信息

```
[09/23/20]seed@VM:~$ ping 192.168.53.100
PING 192.168.53.100 (192.168.53.100) 56(84) bytes of data.
From 192.168.53.99 icmp_seq=1 Destination Host Unreachable
From 192.168.53.99 icmp_seq=2 Destination Host Unreachable
From 192.168.53.99 icmp_seq=3 Destination Host Unreachable
```

```
###[ Ethernet ]###
  dst       = ff:ff:ff:ff:ff:ff
  src       = 86:68:42:66:9e:46
  type      = ARP
###[ ARP ]###
     hwtype    = 0x1
     ptype     = IPv4
     hwlen     = 6
     plen      = 4
     op        = who-has
     hwsrc     = 86:68:42:66:9e:46
     psrc      = 192.168.53.99
     hwdst     = 00:00:00:00:00:00
     pdst      = 192.168.53.100
```

Ping，并产生报文。