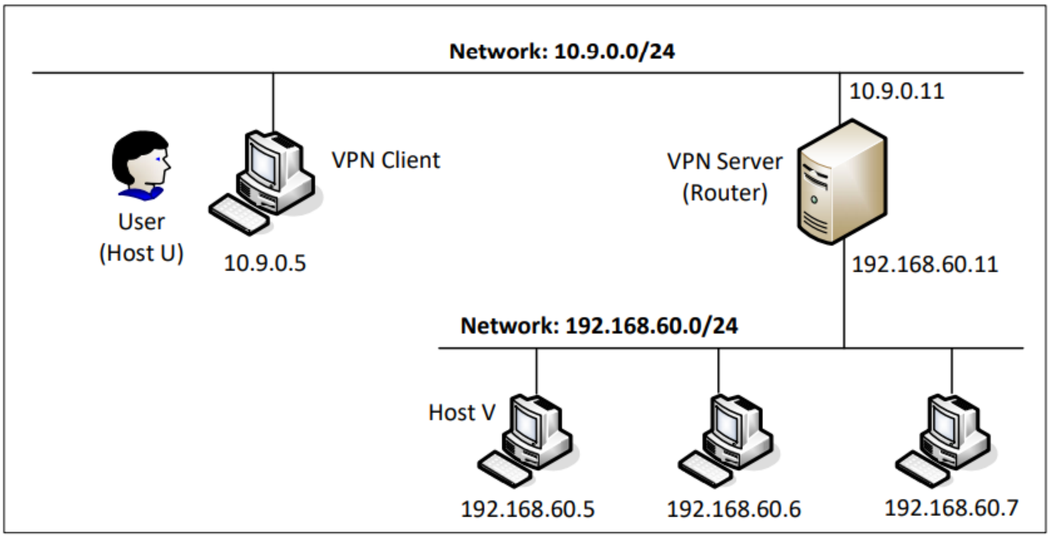
VPN Lab: The Container Version

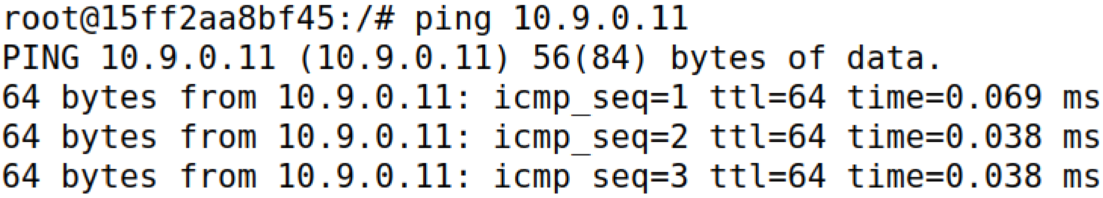
57118103 郭欣然

实验环境



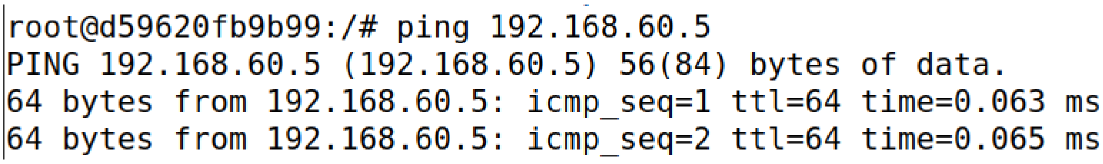
Task1

在主机 U ping 服务器，可以通过



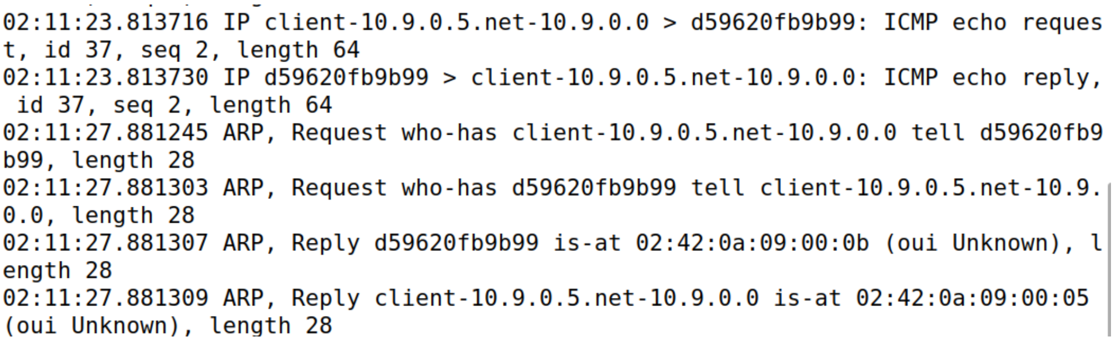
```
root@15ff2aa8bf45:/# ping 10.9.0.11
PING 10.9.0.11 (10.9.0.11) 56(84) bytes of data.
64 bytes from 10.9.0.11: icmp_seq=1 ttl=64 time=0.069 ms
64 bytes from 10.9.0.11: icmp_seq=2 ttl=64 time=0.038 ms
64 bytes from 10.9.0.11: icmp_seq=3 ttl=64 time=0.038 ms
```

在 VPN 上 ping 主机 V，能够连接



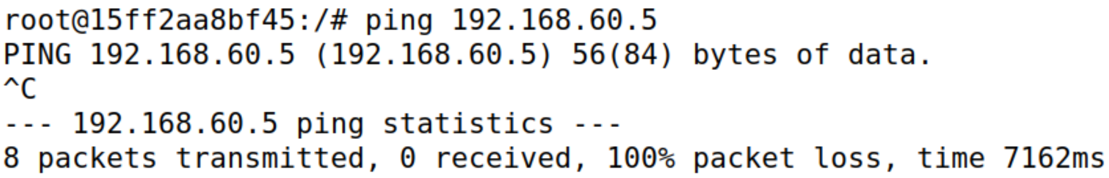
```
root@d59620fb9b99:/# ping 192.168.60.5
PING 192.168.60.5 (192.168.60.5) 56(84) bytes of data.
64 bytes from 192.168.60.5: icmp_seq=1 ttl=64 time=0.063 ms
64 bytes from 192.168.60.5: icmp_seq=2 ttl=64 time=0.065 ms
```

VPN 上利用 tcpdump 抓包



```
02:11:23.813716 IP client-10.9.0.5.net-10.9.0.0 > d59620fb9b99: ICMP echo reques
t, id 37, seq 2, length 64
02:11:23.813730 IP d59620fb9b99 > client-10.9.0.5.net-10.9.0.0: ICMP echo reply,
 id 37, seq 2, length 64
02:11:27.881245 ARP, Request who-has client-10.9.0.5.net-10.9.0.0 tell d59620fb9
b99, length 28
02:11:27.881303 ARP, Request who-has d59620fb9b99 tell client-10.9.0.5.net-10.9.
0.0, length 28
02:11:27.881307 ARP, Reply d59620fb9b99 is-at 02:42:0a:09:00:0b (oui Unknown), l
ength 28
02:11:27.881309 ARP, Reply client-10.9.0.5.net-10.9.0.0 is-at 02:42:0a:09:00:05
(oui Unknown), length 28
```

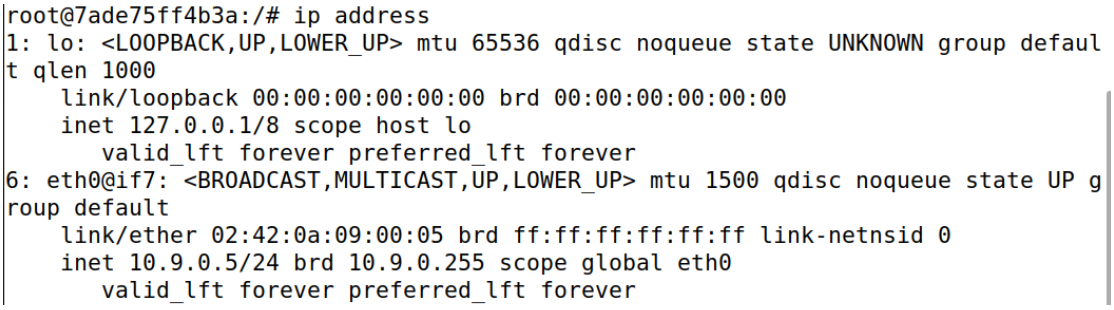
在主机 U 上 ping 主机 V，无法连接



```
root@15ff2aa8bf45:/# 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 7162ms
```

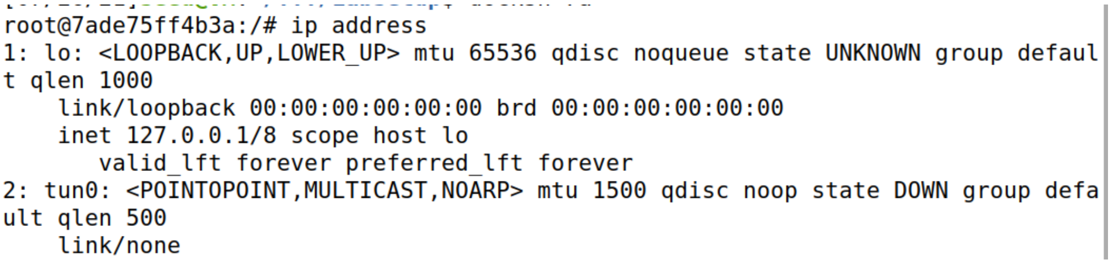
Task2 A

未开启端口前



```
root@7ade75ff4b3a:/# ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 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: eth0@if7: <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
```

开启端口后



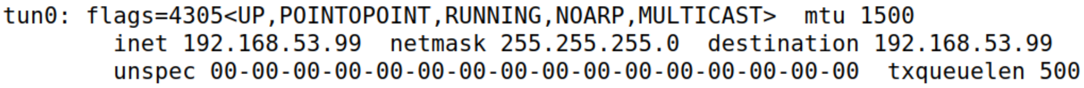
```
root@7ade75ff4b3a:/# ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
2: tun0: <POINTOPOINT,MULTICAST,NOARP> mtu 1500 qdisc noop state DOWN group defa
ult qlen 500
    link/none
```

Task2 B

在程序中添加以下内容.

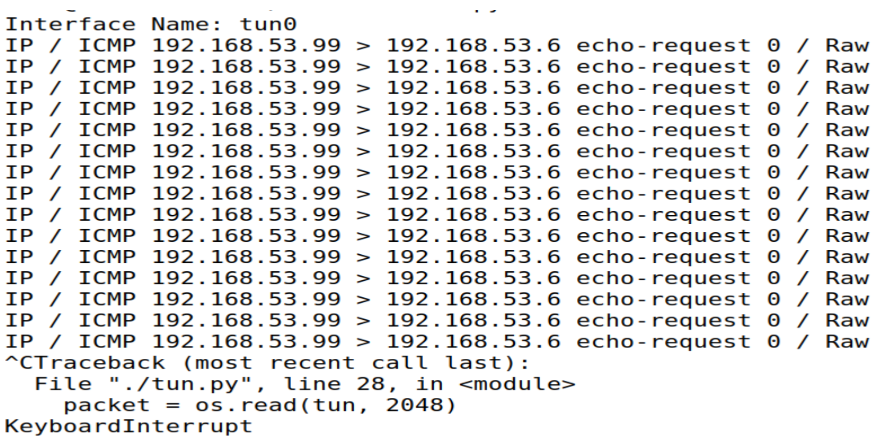
```
os.system("ip addr add 192.168.53.99/24 dev {}".format(ifname))
os.system("ip link set dev {} up".format(ifname))
```

运行 tun.py 后，使用 ifconfig 查看信息。



```
tun0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST>  mtu 1500
        inet 192.168.53.99  netmask 255.255.255.0  destination 192.168.53.99
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 500
```

Task2 C



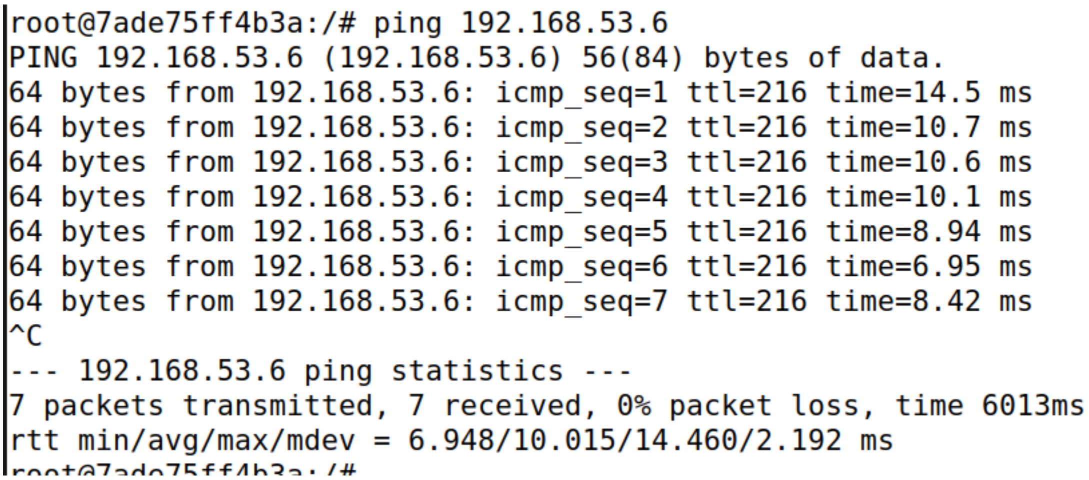
```
Interface Name: tun0
IP / ICMP 192.168.53.99 > 192.168.53.6 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.6 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.6 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.6 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.6 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.6 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.6 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.6 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.6 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.6 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.6 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.6 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.6 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.6 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.6 echo-request 0 / Raw
^CTraceback (most recent call last):
  File "./tun.py", line 28, in <module>
    packet = os.read(tun, 2048)
KeyboardInterrupt
```

client 上 ping 192.168.53.0/24 网段内的主机，程序输出 ICMP 请求信息，因为在循环中输出报文信息

ping 192.168.60.0/24 网段内的主机，程序不输出，因为该子网无法连接

其中代码修改如下：

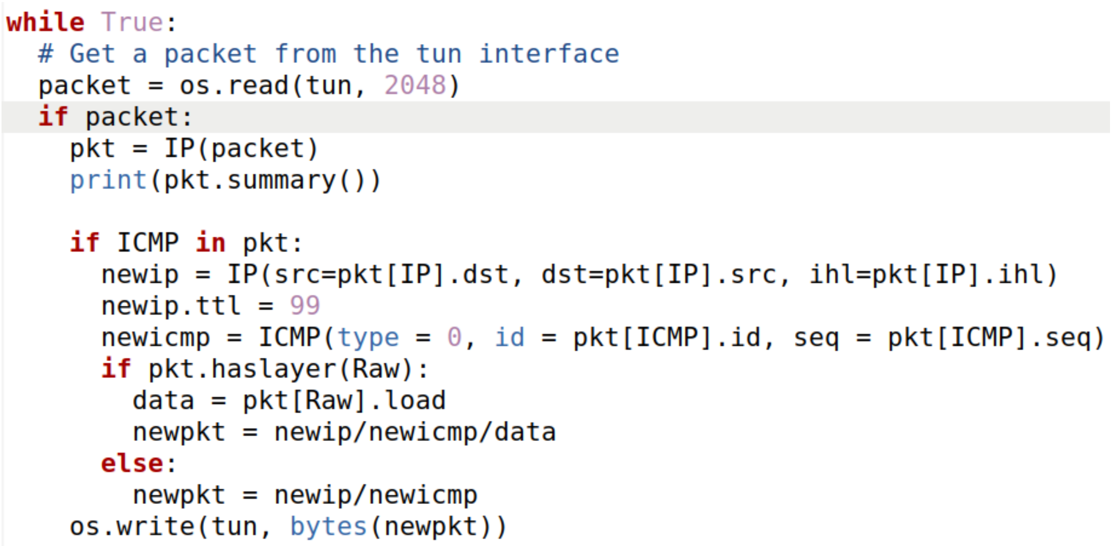
```
#!/usr/bin/env python3

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

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

# Create the tun interface
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)

# 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:
# Get a packet from the tun interface
  packet = os.read(tun, 2048)
  if packet:
    ip = IP(packet)
print(ip.summary())



d
#!/usr/bin/env python3

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

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

# Create the tun interface
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)

# 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:
# Get a packet from the tun interface
  packet = os.read(tun, 2048)
  if True:
    pkt = IP(packet)
    print(pkt.summary())

    if ICMP in pkt:
        newip = IP(src=pkt[IP].dst, dst=pkt[IP].src, ihl=pkt[IP].ihl)
        newip.ttl = 216
        newicmp =ICMP(type=0, id=pkt[ICMP].id,seq=pkt[ICMP].seq)
        if pkt.haslayer(Raw):
            data = pkt[Raw].load
            newpkt = newip/newicmp/data
        else:
            newpkt = newip/newicmp
    os.write(tun,bytes(newpkt))
```



```
root@7ade75ff4b3a:/# ping 192.168.53.6
PING 192.168.53.6 (192.168.53.6) 56(84) bytes of data.
64 bytes from 192.168.53.6: icmp_seq=1 ttl=216 time=14.5 ms
64 bytes from 192.168.53.6: icmp_seq=2 ttl=216 time=10.7 ms
64 bytes from 192.168.53.6: icmp_seq=3 ttl=216 time=10.6 ms
64 bytes from 192.168.53.6: icmp_seq=4 ttl=216 time=10.1 ms
64 bytes from 192.168.53.6: icmp_seq=5 ttl=216 time=8.94 ms
64 bytes from 192.168.53.6: icmp_seq=6 ttl=216 time=6.95 ms
64 bytes from 192.168.53.6: icmp_seq=7 ttl=216 time=8.42 ms
^C
--- 192.168.53.6 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6013ms
rtt min/avg/max/mdev = 6.948/10.015/14.460/2.192 ms
root@7ade75ff4b3a:/#
```

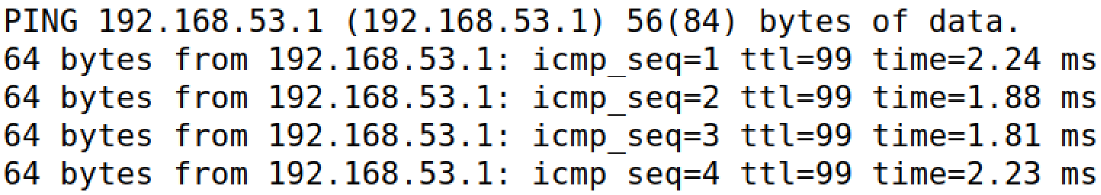
Task2 D

代码修改如下：



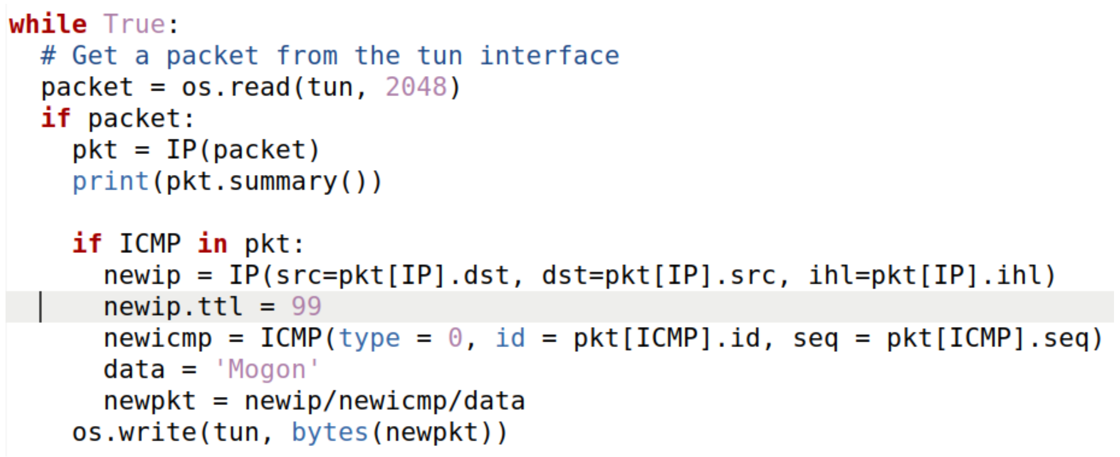
```
while True:
  # Get a packet from the tun interface
  packet = os.read(tun, 2048)
  if packet:
    pkt = IP(packet)
    print(pkt.summary())

    if ICMP in pkt:
      newip = IP(src=pkt[IP].dst, dst=pkt[IP].src, ihl=pkt[IP].ihl)
      newip.ttl = 99
      newicmp = ICMP(type = 0, id = pkt[ICMP].id, seq = pkt[ICMP].seq)
      if pkt.haslayer(Raw):
        data = pkt[Raw].load
        newpkt = newip/newicmp/data
      else:
        newpkt = newip/newicmp
    os.write(tun, bytes(newpkt))
```

可以 ping 通 53 网段



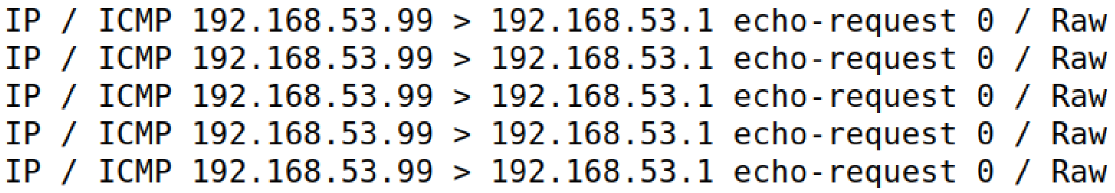
```
PING 192.168.53.1 (192.168.53.1) 56(84) bytes of data.
64 bytes from 192.168.53.1: icmp_seq=1 ttl=99 time=2.24 ms
64 bytes from 192.168.53.1: icmp_seq=2 ttl=99 time=1.88 ms
64 bytes from 192.168.53.1: icmp_seq=3 ttl=99 time=1.81 ms
64 bytes from 192.168.53.1: icmp_seq=4 ttl=99 time=2.23 ms
```

随意字符串的代码如下

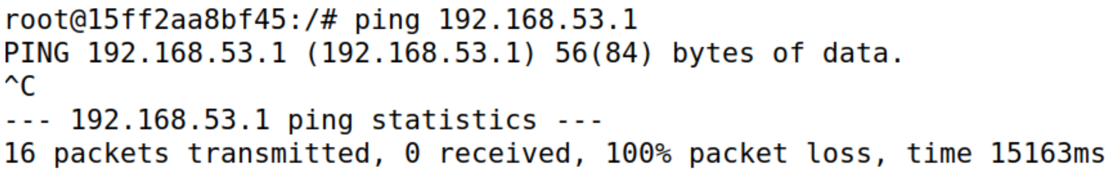


```
while True:
  # Get a packet from the tun interface
  packet = os.read(tun, 2048)
  if packet:
    pkt = IP(packet)
    print(pkt.summary())

    if ICMP in pkt:
      newip = IP(src=pkt[IP].dst, dst=pkt[IP].src, ihl=pkt[IP].ihl)
      newip.ttl = 99
      newicmp = ICMP(type = 0, id = pkt[ICMP].id, seq = pkt[ICMP].seq)
      data = 'Mogon'
      newpkt = newip/newicmp/data
    os.write(tun, bytes(newpkt))
```

tun.py 接收到，但 ping 不通，说明随意字符串不能完成 ping 的过程



```
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
```



```
root@15ff2aa8bf45:/# 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 ---
16 packets transmitted, 0 received, 100% packet loss, time 15163ms
```

Task3

代码修改如下：

服务器

```
#!/usr/bin/env python3

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

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

# Create the tun interface
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)
```

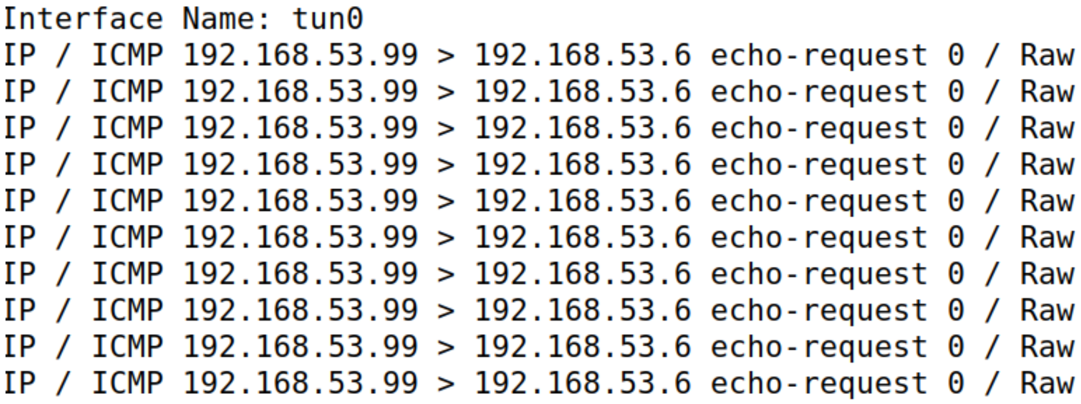
```
# Get the interface name
ifname = ifname_bytes.decode('UTF-8')[:16].strip("\x00")
print("Interface Name: {}".format(ifname))
os.system("ip addr add 192.168.53.11/24 dev {}".format(ifname))
os.system("ip link set dev {} up".format(ifname))

server = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
SERVER_IP="0.0.0.0"
SERVER_PORT=9090
server.bind((SERVER_IP,SERVER_PORT))

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

用户

```
#!/usr/bin/env python3

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

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

# Create the tun interface
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)

# 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))
os.system("ip addr add 192.168.60.0/24 dev {}".format(ifname))
```

```
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
SERVER_IP="10.9.0.11"
SERVER_PORT=9090

while True:
  # Get a packet from the tun interface
  packet = os.read(tun, 2048)
  if packet:
    pkt = IP(packet)
    print(pkt.summary())
    sock.sendto(packet, (SERVER_IP, SERVER_PORT))
```

隧道发送成功



```
Interface Name: tun0
IP / ICMP 192.168.53.99 > 192.168.53.6 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.6 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.6 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.6 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.6 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.6 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.6 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.6 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.6 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.6 echo-request 0 / Raw
```

Task4

客户端代码修改如下

```
#!/usr/bin/env python3

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

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

# Create the tun interface
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)

# Get the interface name
```

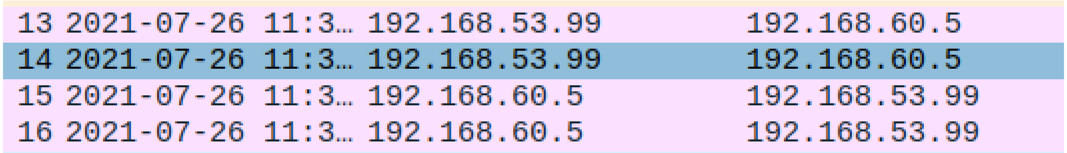
```
ifname = ifname_bytes.decode('UTF-8')[:16].strip("\x00")
print("Interface Name: {}".format(ifname))

os.system("ip addr add 192.168.53.99/24 dev {}".format(ifname))
os.system("ip link set dev {} up".format(ifname))
os.system("ip addr add 192.168.60.0/24 dev {}".format(ifname))

sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
SERVER_IP="10.9.0.11"
SERVER_PORT=9090
fds=[sock,tun]
while True:
  ready,_,_=select.select(fds,[],[])
  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)
        if packet:
          pkt=IP(packet)
          print(pkt.summary())
          sock.sendto(packet,(SERVER_IP,SERVER_PORT))
```

服务器代码修改如下

```
#!/usr/bin/env python3

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

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

# Create the tun interface
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)
```

```
# Get the interface name
ifname = ifname_bytes.decode('UTF-8')[:16].strip("\x00")
print("Interface Name: {}".format(ifname))
os.system("ip addr add 192.168.53.11/24 dev {}".format(ifname))
os.system("ip link set dev {} up".format(ifname))

sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
SERVER_IP="0.0.0.0"
SERVER_PORT=9090
ip = "10.9.0.5"
port = 10000
sock.bind((SERVER_IP,SERVER_PORT))
fds=[sock,tun]
while True:
  ready,_,_=select.select(fds,[],[])
  for fd in ready:
     if fd is sock:
       print("sock...")
       data,(ip,port)=sock.recvfrom(2048)
       print("{}:{}-->{}:{}".format(ip,port,SERVER_IP,SERVER_PORT))
       pkt=IP(data)
       print("Inside : {}:{}".format(pkt.src,pkt.dst))
       os.write(tun,data)
     if fd is tun:
       print("tun...")
       packet = os.read(tun,2048)
       pkt=IP(packet)
       print("Return : {}:{}".format(pkt.src,pkt.dst))
       sock.sendto(packet,(ip,port))
```

在 wireshark 中看到 60.5 已经做出了回应，但因为没有返回的代码，reply 传不回 53.99



```
13 2021-07-26 11:3… 192.168.53.99        192.168.60.5
14 2021-07-26 11:3… 192.168.53.99        192.168.60.5
15 2021-07-26 11:3… 192.168.60.5         192.168.53.99
16 2021-07-26 11:3… 192.168.60.5         192.168.53.99
```

Task5

客户端代码修改如下

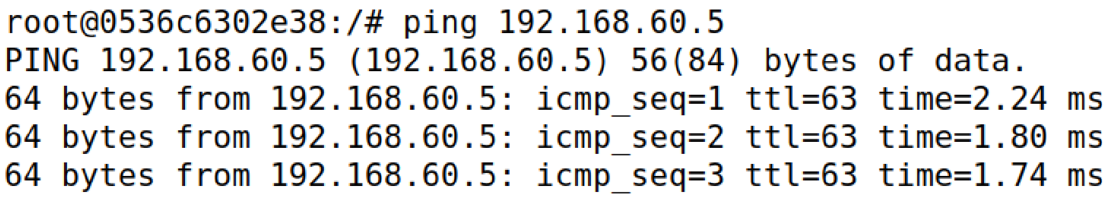
```
#!/usr/bin/env python3

import fcntl
import struct
import os
import time
```

```
from scapy.all import *

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

# Create the tun interface
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)

# 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))
os.system("ip addr add 192.168.60.0/24 dev {}".format(ifname))

sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
SERVER_IP="10.9.0.11"
SERVER_PORT=9090
fds=[sock,tun]
while True:
  ready,_,_=select.select(fds,[],[])
  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)
       if packet:
          pkt=IP(packet)
          print(pkt.summary())
          sock.sendto(packet,(SERVER_IP,SERVER_PORT))
```

服务器代码修改如下

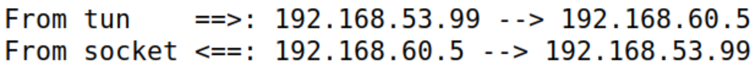
```
#!/usr/bin/env python3

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

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

# Create the tun interface
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)

# Get the interface name
ifname = ifname_bytes.decode('UTF-8')[:16].strip("\x00")
print("Interface Name: {}".format(ifname))
os.system("ip addr add 192.168.53.11/24 dev {}".format(ifname))
os.system("ip link set dev {} up".format(ifname))

sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
SERVER_IP="0.0.0.0"
SERVER_PORT=9090
ip = "10.9.0.5"
port = 10000
sock.bind((SERVER_IP,SERVER_PORT))
fds=[sock,tun]
while True:
  ready,_,_=select.select(fds,[],[])
  for fd in ready:
     if fd is sock:
       print("sock...")
       data,(ip,port)=sock.recvfrom(2048)
       print("{}:{}-->{}:{}".format(ip,port,SERVER_IP,SERVER_PORT))
       pkt=IP(data)
       print("Inside : {}:{}".format(pkt.src,pkt.dst))
       os.write(tun,data)
     if fd is tun:
       print("tun...")
       packet = os.read(tun,2048)
       pkt=IP(packet)
       print("Return : {}:{}".format(pkt.src,pkt.dst))
       sock.sendto(packet,(ip,port))
```

ping 通 192.168.60.5



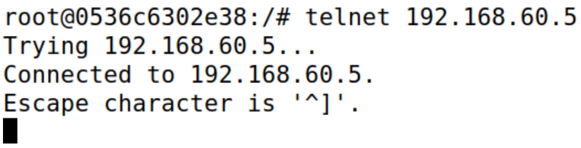
```
root@0536c6302e38:/# ping 192.168.60.5
PING 192.168.60.5 (192.168.60.5) 56(84) bytes of data.
64 bytes from 192.168.60.5: icmp_seq=1 ttl=63 time=2.24 ms
64 bytes from 192.168.60.5: icmp_seq=2 ttl=63 time=1.80 ms
64 bytes from 192.168.60.5: icmp_seq=3 ttl=63 time=1.74 ms
```

服务器端



```
From tun    ==>: 192.168.53.99 --> 192.168.60.5
From socket <==: 192.168.60.5 --> 192.168.53.99
```

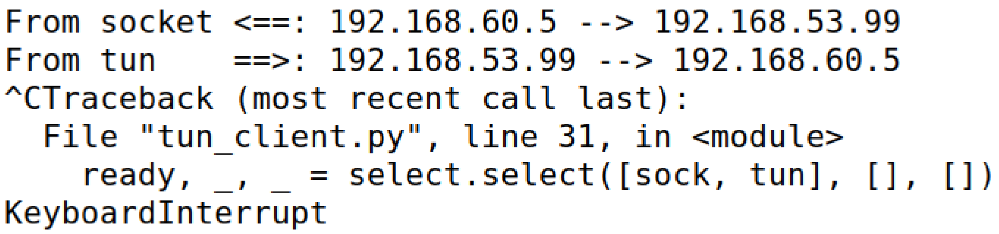
Telnet 连接成功



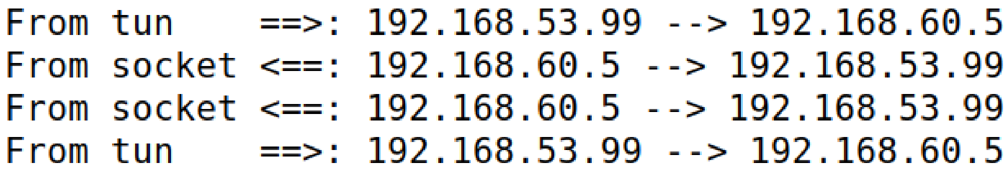
```
root@0536c6302e38:/# telnet 192.168.60.5
Trying 192.168.60.5...
Connected to 192.168.60.5.
Escape character is '^]'.
```

Task6

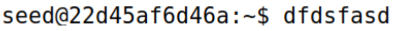
telnet 连接后，断连再重连



```
From socket <==: 192.168.60.5 --> 192.168.53.99
From tun    ==>: 192.168.53.99 --> 192.168.60.5
^CTraceback (most recent call last):
  File "tun_client.py", line 31, in <module>
    ready, _, _ = select.select([sock, tun], [], [])
KeyboardInterrupt
```



```
From tun    ==>: 192.168.53.99 --> 192.168.60.5
From socket <==: 192.168.60.5 --> 192.168.53.99
From socket <==: 192.168.60.5 --> 192.168.53.99
From tun    ==>: 192.168.53.99 --> 192.168.60.5
```

断连时输入字符不显示，重连后一下全部显示，因为 telnet 需要将输入内容传去传回后才会显示。 `seed@22d45af6d46a:~$ dfdsfasd`