Lab7

57118202 付孜

Task 1：Network Setup

验证主机 U 可以与 VPN Server 通信以及在路由器上 tcpdump 捕获的报文。

```
[08/01/21]seed@VM:~/Desktop$ docksh 37
root@3715c8f3b893:/# 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=0.171 ms
64 bytes from 10.9.0.11: icmp_seq=2 ttl=64 time=0.098 ms
64 bytes from 10.9.0.11: icmp_seq=3 ttl=64 time=0.092 ms
64 bytes from 10.9.0.11: icmp_seq=4 ttl=64 time=0.150 ms
64 bytes from 10.9.0.11: icmp_seq=5 ttl=64 time=0.113 ms
64 bytes from 10.9.0.11: icmp_seq=6 ttl=64 time=0.088 ms
64 bytes from 10.9.0.11: icmp_seq=7 ttl=64 time=0.117 ms
64 bytes from 10.9.0.11: icmp_seq=8 ttl=64 time=0.146 ms
64 bytes from 10.9.0.11: icmp_seq=9 ttl=64 time=0.065 ms
64 bytes from 10.9.0.11: icmp_seq=10 ttl=64 time=0.184 ms
64 bytes from 10.9.0.11: icmp_seq=11 ttl=64 time=0.060 ms
64 bytes from 10.9.0.11: icmp_seq=12 ttl=64 time=0.079 ms
64 bytes from 10.9.0.11: icmp_seq=13 ttl=64 time=0.068 ms
64 bytes from 10.9.0.11: icmp_seq=14 ttl=64 time=0.070 ms
^C
--- 10.9.0.11 ping statistics ---
14 packets transmitted, 14 received, 0% packet loss, time 13312ms
rtt min/avg/max/mdev = 0.060/0.107/0.184/0.039 ms
root@3715c8f3b893:/#
```

```
root@93e357d2a5f3:/# tcpdump -i eth0 -n
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
17:48:16.929889 ARP, Request who-has 10.9.0.11 tell 10.9.0.5, length 28
17:48:16.929947 ARP, Reply 10.9.0.11 is-at 02:42:0a:09:00:0b, length 28
17:48:16.929968 IP 10.9.0.5 > 10.9.0.11: ICMP echo request, id 14, seq 1, length
 64
17:48:16.929983 IP 10.9.0.11 > 10.9.0.5: ICMP echo reply, id 14, seq 1, length 6
4
17:48:17.954309 IP 10.9.0.5 > 10.9.0.11: ICMP echo request, id 14, seq 2, length
 64
17:48:17.954355 IP 10.9.0.11 > 10.9.0.5: ICMP echo reply, id 14, seq 2, length 6
4
17:48:18.978247 IP 10.9.0.5 > 10.9.0.11: ICMP echo request, id 14, seq 3, length
 64
17:48:18.978287 IP 10.9.0.11 > 10.9.0.5: ICMP echo reply, id 14, seq 3, length 6
4
17:48:20.008766 IP 10.9.0.5 > 10.9.0.11: ICMP echo request, id 14, seq 4, length
 64
17:48:20.008838 IP 10.9.0.11 > 10.9.0.5: ICMP echo reply, id 14, seq 4, length 6
4
```

验证主机 V 可以与 VPN Server 通信以及在路由器上 tcpdump 捕获的报文。

```
[00/01/21]seed@vm: /Desktop$ docksh c5
root@c3ade50c482f:/# ping 192.168.60.11
PING 192.168.60.11 (192.168.60.11) 56(84) bytes of data.
64 bytes from 192.168.60.11: icmp_seq=1 ttl=64 time=0.439 ms
64 bytes from 192.168.60.11: icmp_seq=2 ttl=64 time=0.074 ms
64 bytes from 192.168.60.11: icmp_seq=3 ttl=64 time=0.080 ms
64 bytes from 192.168.60.11: icmp_seq=4 ttl=64 time=0.059 ms
64 bytes from 192.168.60.11: icmp_seq=5 ttl=64 time=0.074 ms
64 bytes from 192.168.60.11: icmp_seq=6 ttl=64 time=0.060 ms
64 bytes from 192.168.60.11: icmp_seq=7 ttl=64 time=0.066 ms
64 bytes from 192.168.60.11: icmp_seq=8 ttl=64 time=0.073 ms
64 bytes from 192.168.60.11: icmp_seq=9 ttl=64 time=0.082 ms
64 bytes from 192.168.60.11: icmp_seq=10 ttl=64 time=0.093 ms
64 bytes from 192.168.60.11: icmp_seq=11 ttl=64 time=0.058 ms
64 bytes from 192.168.60.11: icmp_seq=12 ttl=64 time=0.094 ms
^C
--- 192.168.60.11 ping statistics ---
12 packets transmitted, 12 received, 0% packet loss, time 11246ms
rtt min/avg/max/mdev = 0.058/0.104/0.439/0.101 ms
root@c3ade50c482f:/#
```

```
root@93e357d2a5f3:/# tcpdump -i eth1 -n
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth1, link-type EN10MB (Ethernet), capture size 262144 bytes
17:51:18.933098 ARP, Request who-has 192.168.60.11 tell 192.168.60.5, length 28
17:51:18.933110 ARP, Reply 192.168.60.11 is-at 02:42:c0:a8:3c:0b, length 28
17:51:18.933128 IP 192.168.60.5 > 192.168.60.11: ICMP echo request, id 30, seq 1
, length 64
17:51:18.933139 IP 192.168.60.11 > 192.168.60.5: ICMP echo reply, id 30, seq 1,
length 64
17:51:19.940532 IP 192.168.60.5 > 192.168.60.11: ICMP echo request, id 30, seq 2
, length 64
17:51:19.940555 IP 192.168.60.11 > 192.168.60.5: ICMP echo reply, id 30, seq 2,
length 64
17:51:20.964774 IP 192.168.60.5 > 192.168.60.11: ICMP echo request, id 30, seq 3
, length 64
17:51:20.964795 IP 192.168.60.11 > 192.168.60.5: ICMP echo reply, id 30, seq 3,
length 64
17:51:21.989229 IP 192.168.60.5 > 192.168.60.11: ICMP echo request, id 30, seq 4
, length 64
17:51:21.989246 IP 192.168.60.11 > 192.168.60.5: ICMP echo reply, id 30, seq 4,
```

验证主机 U 不可与主机 V 通信。

```
root@3715c8f3b893:/# 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 ---
3 packets transmitted, 0 received, 100% packet loss, time 2041ms

root@3715c8f3b893:/#
```

Task 2：Create and Configure TUN Interface

Task 2.A：Name of the Interface

在代码此处将 `tun` 修改成自己名字简拼 `fz` 。

```
ifr = struct.pack('16sH', b'fz%d', IFF_TUN | IFF_NO_PI)
```

在主机 U(10.9.0.5) 上运行 `chmod a+x tun.py` 和 `tun.py` 可以观察到修改接口成功。

```
root@3715c8f3b893:/volumes# chmod a+x tun.py
root@3715c8f3b893:/volumes# tun.py
Interface Name: fz0
```

然后在主机 U(10.9.0.5) 上运行 `ip address` 查看所有接口，可发现我们修改的 `tun` 接口，命名为 `fz0` 。

```
root@3715c8f3b893:/# 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
3: fz0: <POINTOPOINT,MULTICAST,NOARP> mtu 1500 qdisc noop state DOWN group defau
lt qlen 500
    link/none
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
root@3715c8f3b893:/#
```

Task 2.B：Set up the TUN Interface

在 `tun.py` 文件中添加以下两行代码，编译运行后主机 `U(10.9.0.5)` 上运行 `ifconfig` 查看所有接口，可观察到绑定 `IP` 地址。

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

```
fz0: 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
(UNSPEC)
```

Task 2.C：Read from the TUN Interface

`ping 192.168.53.5`，可以看到程序有输出，但是请求无响应，因为实际主机不存在。

```
root@3715c8f3b893:/# ping 192.168.53.5
PING 192.168.53.5 (192.168.53.5) 56(84) bytes of data.
^C
--- 192.168.53.5 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3073ms

root@3715c8f3b893:/volumes# tun.py
Interface Name: fz0
IP / ICMP 192.168.53.99 > 192.168.53.5 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.5 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.5 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.5 echo-request 0 / Raw
^CTraceback (most recent call last):
  File "./tun.py", line 28, in <module>
    packet = os.read(tun, 2048)
KeyboardInterrupt
```

在 `ping 192.168.60.5` 时，由于未添加路由，程序并无输出。

```
root@3715c8f3b893:/# 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 ---
5 packets transmitted, 0 received, 100% packet loss, time 4094ms

root@3715c8f3b893:/#
```

```
root@3715c8f3b893:/volumes# tun.py
Interface Name: fz0
```

Task 2.D: Write to the TUN Interface

代码如下:

```python
#!/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'fz%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 = 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))
```

此时我们 `ping 192.168.53.5` 可以观察到返回的是我们构造的报文 (`ttl=99`)，在接口处我们可以看到完整的 `IP/ICMP/Raw` 三层报文。

```
root@3715c8f3b893:/# ping 192.168.53.5
PING 192.168.53.5 (192.168.53.5) 56(84) bytes of data.
64 bytes from 192.168.53.5: icmp_seq=1 ttl=99 time=1.94 ms
64 bytes from 192.168.53.5: icmp_seq=2 ttl=99 time=1.68 ms
64 bytes from 192.168.53.5: icmp_seq=3 ttl=99 time=1.67 ms
64 bytes from 192.168.53.5: icmp_seq=4 ttl=99 time=1.54 ms
64 bytes from 192.168.53.5: icmp_seq=5 ttl=99 time=1.44 ms
64 bytes from 192.168.53.5: icmp_seq=6 ttl=99 time=1.56 ms
64 bytes from 192.168.53.5: icmp_seq=7 ttl=99 time=1.58 ms
64 bytes from 192.168.53.5: icmp_seq=8 ttl=99 time=1.52 ms
64 bytes from 192.168.53.5: icmp_seq=9 ttl=99 time=1.36 ms
64 bytes from 192.168.53.5: icmp_seq=10 ttl=99 time=1.45 ms
^C
--- 192.168.53.5 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9025ms
rtt min/avg/max/mdev = 1.362/1.573/1.941/0.154 ms
root@3715c8f3b893:/#
```

```
root@3715c8f3b893:/volumes# tun1.py
Interface Name: fz0
IP / ICMP 192.168.53.99 > 192.168.53.5 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.5 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.5 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.5 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.5 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.5 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.5 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.5 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.5 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.5 echo-request 0 / Raw
```

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

代码如下:

`tun-task3-client.py`

```python
#!/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'fz%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 route add 192.168.60.0/24 dev {}".format(ifname))
```

```python
# Create UDP socket
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))
```

`tun-task3-server.py`

```python
#!/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'fz%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 route add 192.168.60.0/24 dev {}".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))
```

在服务器端我们可以看到管道外部是 `10.9.0.5-->0.0.0.0`，内部是 `192.168.53.99--`
`>192.168.60.5`。

```
root@93e357d2a5f3:/volumes# tun-task3-server.py
Interface Name: fz0
RTNETLINK answers: File exists
10.9.0.5:38643 --> 0.0.0.0:9090
Inside: 192.168.53.99 --> 192.168.60.5
10.9.0.5:38643 --> 0.0.0.0:9090
Inside: 192.168.53.99 --> 192.168.60.5
10.9.0.5:38643 --> 0.0.0.0:9090
Inside: 192.168.53.99 --> 192.168.60.5
10.9.0.5:38643 --> 0.0.0.0:9090
Inside: 192.168.53.99 --> 192.168.60.5
10.9.0.5:38643 --> 0.0.0.0:9090
Inside: 192.168.53.99 --> 192.168.60.5
10.9.0.5:38643 --> 0.0.0.0:9090
Inside: 192.168.53.99 --> 192.168.60.5
10.9.0.5:38643 --> 0.0.0.0:9090
Inside: 192.168.53.99 --> 192.168.60.5
10.9.0.5:38643 --> 0.0.0.0:9090
Inside: 192.168.53.99 --> 192.168.60.5
10.9.0.5:38643 --> 0.0.0.0:9090
Inside: 192.168.53.99 --> 192.168.60.5
10.9.0.5:38643 --> 0.0.0.0:9090
Inside: 192.168.53.99 --> 192.168.60.5
10.9.0.5:38643 --> 0.0.0.0:9090
Inside: 192.168.53.99 --> 192.168.60.5
10.9.0.5:38643 --> 0.0.0.0:9090
```

```
root@3715c8f3b893:/volumes# tun-task3-client.py
Interface Name: fz0
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
```

Task 4: Set Up the VPN Server

`tun-task4-server.py`

```python
#!/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)
```

```python
ifr = struct.pack('16sH', b'fz%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))
os.system("ip route add 192.168.60.0/24 dev {}".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)
    print("write")
```

在 `server` 上的 `eth1` 的接口可以看到收到了返回。

```
root@93e357d2a5f3:/# tcpdump -nni eth1
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth1, link-type EN10MB (Ethernet), capture size 262144 bytes
19:02:35.459666 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 89, seq 20, length 6
4
19:02:35.459725 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 89, seq 20, length 64
19:02:36.484063 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 89, seq 21, length 6
4
19:02:36.484090 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 89, seq 21, length 64
19:02:37.509813 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 89, seq 22, length 6
4
19:02:37.509860 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 89, seq 22, length 64
19:02:38.534814 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 89, seq 23, length 6
4
19:02:38.534844 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 89, seq 23, length 64
19:02:39.556174 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 89, seq 24, length 6
4
19:02:39.556205 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 89, seq 24, length 64
19:02:40.485441 ARP, Request who-has 192.168.60.5 tell 192.168.60.11, length 28
19:02:40.485551 ARP, Request who-has 192.168.60.11 tell 192.168.60.5, length 28
19:02:40.485561 ARP, Reply 192.168.60.11 is-at 02:42:c0:a8:3c:0b, length 28
19:02:40.485566 ARP, Reply 192.168.60.5 is-at 02:42:c0:a8:3c:05, length 28
19:02:40.580711 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 89, seq 25, length 6
```

Task 5: Handling Traffic in Both Directions

`tun-task5-client.py`

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

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

TUNSETIFF = 0x400454ca
IFF_TUN   = 0x0001
IFF_TAP   = 0x0002
```

```python
IFF_NO_PI = 0x1000

# Create the tun interface
tun = os.open("/dev/net/tun", os.O_RDWR)
ifr = struct.pack('16sH', b'fz%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 route 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))
```

`tun-task5-server.py`

```python
#!/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'fz%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))
```

```python
os.system("ip addr add 192.168.53.11/24 dev {}".format(ifname))
os.system("ip link set dev {} up".format(ifname))
os.system("ip route add 192.168.60.0/24 dev {}".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))
```

此时 `ping 192.168.60.5` 可以 `ping` 通，并且能看到返回报文。

```
root@3715c8f3b893:/# 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=2.67 ms
64 bytes from 192.168.60.5: icmp_seq=2 ttl=63 time=2.13 ms
64 bytes from 192.168.60.5: icmp_seq=3 ttl=63 time=1.98 ms
64 bytes from 192.168.60.5: icmp_seq=4 ttl=63 time=1.73 ms
64 bytes from 192.168.60.5: icmp_seq=5 ttl=63 time=2.09 ms
64 bytes from 192.168.60.5: icmp_seq=6 ttl=63 time=2.05 ms
64 bytes from 192.168.60.5: icmp_seq=7 ttl=63 time=1.97 ms
64 bytes from 192.168.60.5: icmp_seq=8 ttl=63 time=1.99 ms
64 bytes from 192.168.60.5: icmp_seq=9 ttl=63 time=2.05 ms
64 bytes from 192.168.60.5: icmp_seq=10 ttl=63 time=1.87 ms
64 bytes from 192.168.60.5: icmp_seq=11 ttl=63 time=1.88 ms
64 bytes from 192.168.60.5: icmp_seq=12 ttl=63 time=2.11 ms
64 bytes from 192.168.60.5: icmp_seq=13 ttl=63 time=1.92 ms
64 bytes from 192.168.60.5: icmp_seq=14 ttl=63 time=1.75 ms
64 bytes from 192.168.60.5: icmp_seq=15 ttl=63 time=1.90 ms
64 bytes from 192.168.60.5: icmp_seq=16 ttl=63 time=1.93 ms
64 bytes from 192.168.60.5: icmp_seq=17 ttl=63 time=2.10 ms
64 bytes from 192.168.60.5: icmp_seq=18 ttl=63 time=1.92 ms
64 bytes from 192.168.60.5: icmp_seq=19 ttl=63 time=2.00 ms
```

```
root@93e357d2a5f3:/volumes# chmod a+x tun-task5-server.py
root@93e357d2a5f3:/volumes# tun-task5-server.py
Interface Name: fz0
RTNETLINK answers: File exists
sock...
10.9.0.5:34588 --> 0.0.0.0:9090
Inside: 192.168.53.99 --> 192.168.60.5
tun...
Return: 192.168.60.5--192.168.53.99
sock...
10.9.0.5:34588 --> 0.0.0.0:9090
Inside: 192.168.53.99 --> 192.168.60.5
tun...
Return: 192.168.60.5--192.168.53.99
sock...
10.9.0.5:34588 --> 0.0.0.0:9090
Inside: 192.168.53.99 --> 192.168.60.5
tun...
Return: 192.168.60.5--192.168.53.99
sock...
10.9.0.5:34588 --> 0.0.0.0:9090
Inside: 192.168.53.99 --> 192.168.60.5
tun...
Return: 192.168.60.5--192.168.53.99

root@3715c8f3b893:/volumes# chmod a+x tun-task5-client.py
root@3715c8f3b893:/volumes# tun-task5-client.py
Interface Name: fz0
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
From socket: 192.168.60.5 --> 192.168.53.99
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
From socket: 192.168.60.5 --> 192.168.53.99
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
From socket: 192.168.60.5 --> 192.168.53.99
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
From socket: 192.168.60.5 --> 192.168.53.99
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
From socket: 192.168.60.5 --> 192.168.53.99
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
From socket: 192.168.60.5 --> 192.168.53.99
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
From socket: 192.168.60.5 --> 192.168.53.99
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
From socket: 192.168.60.5 --> 192.168.53.99
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
From socket: 192.168.60.5 --> 192.168.53.99
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
From socket: 192.168.60.5 --> 192.168.53.99
```

`telnet 192.168.60.5`，结果同理。

```
root@3715c8f3b893:/# telnet 192.168.60.5
Trying 192.168.60.5...
Connected to 192.168.60.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
c3ade50c482f login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)

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

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

To restore this content, you can run the 'unminimize' command.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
```

```
root@93e357d2a5f3:/volumes# tun-task5-server.py
Interface Name: fz0
RTNETLINK answers: File exists
sock...
10.9.0.5:59990 --> 0.0.0.0:9090
Inside: 192.168.53.99 --> 192.168.60.5
tun...
Return: 192.168.60.5--192.168.53.99
sock...
10.9.0.5:59990 --> 0.0.0.0:9090
Inside: 192.168.53.99 --> 192.168.60.5
sock...
10.9.0.5:59990 --> 0.0.0.0:9090
Inside: 192.168.53.99 --> 192.168.60.5
tun...
Return: 192.168.60.5--192.168.53.99
tun...
Return: 192.168.60.5--192.168.53.99
sock...
10.9.0.5:59990 --> 0.0.0.0:9090
Inside: 192.168.53.99 --> 192.168.60.5
tun...
```

```
root@3715c8f3b893:/volumes# tun-task5-client.py
Interface Name: fz0
IP / TCP 192.168.53.99:57752 > 192.168.60.5:telnet S
From socket: 192.168.60.5 --> 192.168.53.99
IP / TCP 192.168.53.99:57752 > 192.168.60.5:telnet A
IP / TCP 192.168.53.99:57752 > 192.168.60.5:telnet PA / Raw
From socket: 192.168.60.5 --> 192.168.53.99
From socket: 192.168.60.5 --> 192.168.53.99
IP / TCP 192.168.53.99:57752 > 192.168.60.5:telnet A
IP / TCP 192.168.53.99:57752 > 192.168.60.5:telnet PA / Raw
From socket: 192.168.60.5 --> 192.168.53.99
IP / TCP 192.168.53.99:57752 > 192.168.60.5:telnet A
From socket: 192.168.60.5 --> 192.168.53.99
IP / TCP 192.168.53.99:57752 > 192.168.60.5:telnet PA / Raw
From socket: 192.168.60.5 --> 192.168.53.99
From socket: 192.168.60.5 --> 192.168.53.99
IP / TCP 192.168.53.99:57752 > 192.168.60.5:telnet A
IP / TCP 192.168.53.99:57752 > 192.168.60.5:telnet PA / Raw
From socket: 192.168.60.5 --> 192.168.53.99
From socket: 192.168.60.5 --> 192.168.53.99
IP / TCP 192.168.53.99:57752 > 192.168.60.5:telnet A
IP / TCP 192.168.53.99:57752 > 192.168.60.5:telnet PA / Raw
From socket: 192.168.60.5 --> 192.168.53.99
```

Task 6：Tunnel-Breaking Experiment

一旦 `client` 或 `server` 程序中断，这时候敲击键盘没有任何反应，所有的敲击结果都在缓冲区不停地重发；当程序恢复运行，`VPN` 又建立起来，敲击结果就会显示在终端。

```
seed@c3ade50c482f:~$ lshusahuicbuohiohi
-bash: lshusahuicbuohiohi: command not found
seed@c3ade50c482f:~$
```