Lab7

57117112 吴泽辉

1.Network Setup

```
[09/22/20]seed@VM:~$ ifconfig
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:ee:90:2f
          inet addr:10.0.2.11  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::cb91:149b:97d1:9a7d/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:3 errors:0 dropped:0 overruns:0 frame:0
          TX packets:95 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1770 (1.7 KB)  TX bytes:11455 (11.4 KB)

enp0s8    Link encap:Ethernet  HWaddr 08:00:27:39:ef:88
          inet addr:192.168.70.1  Bcast:192.168.70.255  Mask:255.255.255.0
          inet6 addr: fe80::3251:2cf9:2258:7a8f/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:7 errors:0 dropped:0 overruns:0 frame:0
          TX packets:65 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:688 (688.0 B)  TX bytes:7052 (7.0 KB)
```

```
[09/22/20]seed@VM:~$ ifconfig
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:13:2a:b5
          inet addr:192.168.70.101  Bcast:192.168.70.255  Mask:255.255.255.0
          inet6 addr: fe80::3c35:8d8b:6558:4f02/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:64 errors:0 dropped:0 overruns:0 frame:0
          TX packets:79 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:6864 (6.8 KB)  TX bytes:7935 (7.9 KB)
```

在主机 U 上可以 ping 通 VPN 服务器。

```
[09/22/20]seed@VM:~$ ping 10.0.2.11
PING 10.0.2.11 (10.0.2.11) 56(84) bytes of data.
64 bytes from 10.0.2.11: icmp_seq=1 ttl=64 time=0.558 ms
```

```
[09/22/20]seed@VM:~$ ping 10.0.2.8
PING 10.0.2.8 (10.0.2.8) 56(84) bytes of data.
64 bytes from 10.0.2.8: icmp_seq=1 ttl=64 time=0.277 ms
```

VPN 服务器可以 ping 通主机 V。

```
[09/22/20]seed@VM:~$ ping 192.168.70.101
PING 192.168.70.101 (192.168.70.101) 56(84) bytes of data.
64 bytes from 192.168.70.101: icmp_seq=1 ttl=64 time=0.589 ms
```

```
[09/22/20]seed@VM:~$ ping 192.168.70.1
PING 192.168.70.1 (192.168.70.1) 56(84) bytes of data.
64 bytes from 192.168.70.1: icmp_seq=1 ttl=64 time=0.379 ms
```

主机 U 不能 ping 通主机 V。

```
[09/22/20]seed@VM:~$ ping 192.168.70.101
PING 192.168.70.101 (192.168.70.101) 56(84) bytes of data.
```

```
[09/22/20]seed@VM:~$ ping 10.0.2.8
PING 10.0.2.8 (10.0.2.8) 56(84) bytes of data.
```

综上，网络配置成功。

2.Create and Configure TUN Interface

2.a Name of the Interface

```
[09/22/20]seed@VM:~$ ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t 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: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
group default qlen 1000
    link/ether 08:00:27:a9:a3:fd brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.8/24 brd 10.0.2.255 scope global dynamic enp0s3
       valid_lft 559sec preferred_lft 559sec
    inet6 fe80::6245:239e:6f5b:cc67/64 scope link
       valid_lft forever preferred_lft forever
4: yujie0: <POINTOPOINT,MULTICAST,NOARP> mtu 1500 qdisc noop state DOWN group de
fault qlen 500
    link/none
```

2.b Set up the TUN Interface

```
#!/usr/bin/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'chen%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))
while True:
        time.sleep(10)
```

运行程序 tun.py，输出了接口的名称 chen0。在另一个终端使用 ip address 指令

```
5: yujie0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast s
tate UNKNOWN group default qlen 500
    link/none
    inet 192.168.53.99/24 scope global yujie0
       valid_lft forever preferred_lft forever
    inet6 fe80::5a1e:121f:bf2d:1b78/64 scope link flags 800
       valid_lft forever preferred_lft forever
```

2.c Read from the TUN Interface

修改 tun.py，利用 scapy 输出从 chen0 接口发出的 IP 数据包的数据信息。

```
#!/usr/bin/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'chen%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")
os.system("ip addr add 192.168.53.99/24 dev {}".format(ifname))
os.system("ip link set dev {} up".format(ifname))
print("Interface Name: {}".format(ifname))
'''
while True:
        time.sleep(10)
'''
while True:
# Get a packet from the tun interface
        packet = os.read(tun, 2048)
        if True:
                ip = IP(packet)
                ip.show()
```

2.d Write to the TUN Interface

```
[09/22/20]seed@VM:~$ ping 192.168.53.1
PING 192.168.53.1 (192.168.53.1) 56(84) bytes of data.
^C
```

```
###[ IP ]###
  version   = 4
  ihl       = 5
  tos       = 0x0
  len       = 84
  id        = 20199
  flags     = DF
  frag      = 0
  ttl       = 64
  proto     = icmp
  chksum    = 0xd
  src       = 192.168.53.99
  dst       = 192.168.53.1
  \options   \
###[ ICMP ]###
     type      = echo-request
     code      = 0
     chksum    = 0x25d8
     id        = 0x1d3a
     seq       = 0xa
###[ Raw ]###
        load      = '\x9a\xb8i_\xb9\xc8\x0c\x00\x08\t\n\x0b\x0c\r\x0e\x0f\x10\x1
1\x12\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x1f !"#$%&\'()*+,-./012345
67'
```

```
Traceback (most recent call last):
  File "./tun_client.py", line 37, in <module>
    os.write(tun, bytes(newpkt))
OSError: [Errno 22] Invalid argument
```

3.Send the IP Packet to VPN Server Through a Tunnel

```
[09/22/20]seed@VM:~$ chmod a+x tun_server.py
[09/22/20]seed@VM:~$ sudo ./tun_server.py
```

```
[09/22/20]seed@VM:~$ chmod a+x tun_server.py
[09/22/20]seed@VM:~$ sudo ./tun_server.py
10.0.2.8:35613 --> 0.0.0.0:9090
 Inside: 0.0.0.0 --> 238.147.237.222
10.0.2.8:35613 --> 0.0.0.0:9090
 Inside: 192.168.53.99 --> 192.168.53.1
10.0.2.8:35613 --> 0.0.0.0:9090
 Inside: 192.168.53.99 --> 192.168.53.1
10.0.2.8:35613 --> 0.0.0.0:9090
 Inside: 192.168.53.99 --> 192.168.53.1
10.0.2.8:35613 --> 0.0.0.0:9090
 Inside: 0.0.0.0 --> 238.147.237.222
```

```
10.0.2.8:35887 --> 0.0.0.0:9090
 Inside: 192.168.53.99 --> 192.168.70.101
10.0.2.8:35887 --> 0.0.0.0:9090
 Inside: 192.168.53.99 --> 192.168.70.101
10.0.2.8:35887 --> 0.0.0.0:9090
 Inside: 192.168.53.99 --> 192.168.70.101
10.0.2.8:35887 --> 0.0.0.0:9090
 Inside: 192.168.53.99 --> 192.168.70.101
10.0.2.8:35887 --> 0.0.0.0:9090
 Inside: 192.168.53.99 --> 192.168.70.101
```

```
10.0.2.8        192.168.70.101      ICMP     100 Echo (ping) request  id=0x0a7d, seq=2/512,
10.0.2.8        192.168.70.101      ICMP     100 Echo (ping) request  id=0x0a7d, seq=3/768,
10.0.2.8        192.168.70.101      ICMP     100 Echo (ping) request  id=0x0a7d, seq=4/1024,
10.0.2.8        192.168.70.101      ICMP     100 Echo (ping) request  id=0x0a7d, seq=5/1280,
10.0.2.8        192.168.70.101      ICMP     100 Echo (ping) request  id=0x0a7d, seq=6/1536,
```

至此，隧道搭建成功。

4. Set Up the VPN Server

```
#create UDP socket
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))
```

```
    inet6 fe80::769e:baab:1ddb:e269/64 scope link
       valid_lft forever preferred_lft forever
5: chen0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast s
tate UNKNOWN group default qlen 500
    link/none
    inet 192.168.53.5/24 scope global chen0
       valid_lft forever preferred_lft forever
    inet6 fe80::2326:3e0c:2513:c1d0/64 scope link flags 800
       valid_lft forever preferred_lft forever
[09/22/20]seed@VM:~$
```

```
[09/22/20]seed@VM:~$ sudo sysctl net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
```

```
192.168.70.1        192.168.70.101      UDP        92 9090 → 9090 Len=48
192.168.70.101      192.168.70.1        ICMP      120 Destination unreachable
```

5. Handling Traffic in Both Directions

```python
#!/usr/bin/python3
import fcntl
import struct
import os
import time
import socket,select
from scapy.all import *

SERVER_IP="10.0.2.6"
SERVER_PORT=9090

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'chen%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))
#set up the tun interface and routing
os.system("ip addr add 192.168.53.99/24 dev {}".format(ifname))
os.system("ip link set dev {} up".format(ifname))
#set up routing
os.system("sudo route add -net 192.168.70.0/24 {}".format(ifname))
```

```python
#!/usr/bin/python3
import fcntl
import struct
import os
import time
import socket,select
from scapy.all import *

IP_A = "0.0.0.0"
PORT = 9090

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'chen%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))
#configure the tun interface
os.system("ip addr add 192.168.53.5/24 dev {}".format(ifname))
os.system("ip link set dev {} up".format(ifname))
#os.system("sudo route add -net 192.168.53.0/24 {}".format(ifname))
```

```
[09/23/20]seed@VM:~$ telnet 192.168.70.101
Trying 192.168.70.101...
Connected to 192.168.70.101.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
VM login: seed
Password:
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)
```

```
[09/23/20]seed@VM:~$ ping 192.168.70.101
PING 192.168.70.101 (192.168.70.101) 56(84) bytes of data.
64 bytes from 192.168.70.101: icmp_seq=1 ttl=63 time=6.09 ms
64 bytes from 192.168.70.101: icmp_seq=2 ttl=63 time=7.79 ms
64 bytes from 192.168.70.101: icmp_seq=3 ttl=63 time=6.37 ms
64 bytes from 192.168.70.101: icmp_seq=4 ttl=63 time=6.96 ms
64 bytes from 192.168.70.101: icmp_seq=5 ttl=63 time=8.18 ms
64 bytes from 192.168.70.101: icmp_seq=6 ttl=63 time=8.18 ms
```

```
192.168.53.99    192.168.70.101   ICMP    100 Echo (ping) request  id=0x0fd2, seq=8/2048,
192.168.70.101   192.168.53.99    ICMP    100 Echo (ping) reply    id=0x0fd2, seq=8/2048,
192.168.53.99    192.168.70.101   ICMP    100 Echo (ping) request  id=0x0fd2, seq=9/2304,
192.168.70.101   192.168.53.99    ICMP    100 Echo (ping) reply    id=0x0fd2, seq=9/2304,
192.168.53.99    192.168.70.101   ICMP    100 Echo (ping) request  id=0x0fd2, seq=10/2560
192.168.70.101   192.168.53.99    ICMP    100 Echo (ping) reply    id=0x0fd2, seq=10/2560
192.168.53.99    192.168.70.101   ICMP    100 Echo (ping) request  id=0x0fd2, seq=11/2816
```

6. Tunnel-Breaking Experiment

```
[09/23/20]seed@VM:~$ ifconfig
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:13:2a:b5
          inet addr:192.168.70.101  Bcast:192.168.70.255  Mask:255.255.255.0
          inet6 addr: fe80::3c35:8d8b:6558:4f02/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2230 errors:0 dropped:0 overruns:0 frame:0
          TX packets:305 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:156825 (156.8 KB)  TX bytes:29107 (29.1 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:1786 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1786 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:95582 (95.5 KB)  TX bytes:95582 (95.5 KB)
```

7.Routing Experiment on Host V

```
[09/23/20]seed@VM:~$ sudo ip route del 0.0.0.0/0
[09/23/20]seed@VM:~$ sudo ip route add 192.168.53.0/24 dev enp0s3 via 192.168.70
.1
[09/23/20]seed@VM:~$ sudo ip route add 10.0.2.0/24 dev enp0s3 via 192.168.70.1
[09/23/20]seed@VM:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.0.2.0        192.168.70.1    255.255.255.0   UG    0      0        0 enp0s3
169.254.0.0     0.0.0.0         255.255.0.0     U     1000   0        0 enp0s3
192.168.53.0    192.168.70.1    255.255.255.0   UG    0      0        0 enp0s3
192.168.70.0    0.0.0.0         255.255.255.0   U     100    0        0 enp0s3
```

主机 U telnet 主机 V:

```
[09/23/20]seed@VM:~$ telnet 192.168.70.101
Trying 192.168.70.101...
Connected to 192.168.70.101.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
VM login: seed
Password:
Last login: Wed Sep 23 06:41:45 EDT 2020 on pts/19
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)
```

可以成功通信。

8.Experiment with the TUN IP Address

```
#set up the tun interface and routing
os.system("ip addr add 192.168.30.99/24 dev {}".format(ifname))
os.system("ip link set dev {} up".format(ifname))
#set up routing
os.system("sudo route add -net 192.168.70.0/24 {}".format(ifname))
```

```
10.0.2.8          10.0.2.11         UDP      128 9000 → 9000 Len=84
192.168.30.99     192.168.70.101    ICMP     100 Echo (ping) request  id=0x1b85, seq=4/1024,
10.0.2.8          10.0.2.11         UDP      128 9000 → 9000 Len=84
192.168.30.99     192.168.70.101    ICMP     100 Echo (ping) request  id=0x1b85, seq=5/1280,
10.0.2.8          10.0.2.11         UDP      128 9000 → 9000 Len=84
::1               ::1               UDP       64 51066 → 46070 Len=0
192.168.30.99     192.168.70.101    ICMP     100 Echo (ping) request  id=0x1b85, seq=6/1536,
```

在 VPN 服务器上添加与 chen0 接口关联的路由表项，让服务器反向查询路由时能找到正确的接口，从而转发报文。

```
[09/23/20]seed@VM:~$ sudo ip route add 192.168.30.0/24 dev yujie0
[09/23/20]seed@VM:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.70.1    0.0.0.0         UG    100    0        0 enp0s8
0.0.0.0         10.0.2.1        0.0.0.0         UG    101    0        0 enp0s3
10.0.2.0        0.0.0.0         255.255.255.0   U     100    0        0 enp0s3
169.254.0.0     0.0.0.0         255.255.0.0     U     1000   0        0 enp0s8
192.168.30.0    0.0.0.0         255.255.255.0   U     0      0        0 yujie0
192.168.53.0    0.0.0.0         255.255.255.0   U     0      0        0 yujie0
192.168.70.0    0.0.0.0         255.255.255.0   U     100    0        0 enp0s8
```

此时再开启 VPN 隧道进行通信，可以看到此时能够成功 ping 通了

```
[09/23/20]seed@VM:~$ ping 192.168.70.101
PING 192.168.70.101 (192.168.70.101) 56(84) bytes of data.
64 bytes from 192.168.70.101: icmp_seq=1 ttl=63 time=6.09 ms
64 bytes from 192.168.70.101: icmp_seq=2 ttl=63 time=7.79 ms
64 bytes from 192.168.70.101: icmp_seq=3 ttl=63 time=6.37 ms
64 bytes from 192.168.70.101: icmp_seq=4 ttl=63 time=6.96 ms
64 bytes from 192.168.70.101: icmp_seq=5 ttl=63 time=8.18 ms
64 bytes from 192.168.70.101: icmp_seq=6 ttl=63 time=8.18 ms
```

9.Experiment with the TAP Interface

```
# Create the tun interface
tap = os.open("/dev/net/tun", os.O_RDWR)
ifr = struct.pack('16sH', b'tap%d', IFF_TUN | IFF_NO_PI)
ifname_bytes = fcntl.ioctl(tap, TUNSETIFF, ifr)
# Get the interface name
ifname = ifname_bytes.decode('UTF-8')[:16].strip("\x00")
print("Interface Name: {}".format(ifname))
#set up the tun interface and routing
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()
```

在客户端上 ping 192.168.53.0/24 网段的 192.168.53.21。

```
[09/23/20]seed@VM:~$ ping 192.168.53.21
PING 192.168.53.21 (192.168.53.21) 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
From 192.168.53.99 icmp_seq=4 Destination Host Unreachable
From 192.168.53.99 icmp_seq=5 Destination Host Unreachable
```

同时，在客户端上用 wireshark 抓包，tap0 接口结果如下：

```
1 2020-09… 9e:a0:d1:85:0… Broadcast    ARP         42 Who has 192.168.53.21?
2 2020-09… 9e:a0:d1:85:0… Broadcast    ARP         42 Who has 192.168.53.21?
3 2020-09… 9e:a0:d1:85:0… Broadcast    ARP         42 Who has 192.168.53.21?
4 2020-09… 9e:a0:d1:85:0… Broadcast    ARP         42 Who has 192.168.53.21?
5 2020-09… 9e:a0:d1:85:0… Broadcast    ARP         42 Who has 192.168.53.21?
```