Task 1:

VPN Server:

```
[09/26/20]seed@VM:~$ ifconfig
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:67:74:9f
          inet addr:10.0.2.7  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::1f18:5bf6:2184:623c/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:7 errors:0 dropped:0 overruns:0 frame:0
          TX packets:73 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1039 (1.0 KB)  TX bytes:7436 (7.4 KB)

enp0s8    Link encap:Ethernet  HWaddr 08:00:27:c7:b3:3e
          inet addr:192.168.60.1  Bcast:192.168.60.255  Mask:255.255.255.0
          inet6 addr: fe80::8c47:9cde:7c92:2994/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:57 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:6211 (6.2 KB)
```

Host V:

```
[09/26/20]seed@VM:~$ ifconfig
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:32:f9:e8
          inet addr:192.168.60.101  Bcast:192.168.60.255  Mask:255.255.255.0
          inet6 addr: fe80::8d9:8a08:853a:373/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1 errors:0 dropped:0 overruns:0 frame:0
          TX packets:58 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:106 (106.0 B)  TX bytes:6389 (6.3 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:45 errors:0 dropped:0 overruns:0 frame:0
          TX packets:45 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:11928 (11.9 KB)  TX bytes:11928 (11.9 KB)
```

Host V Ping VPN Server:

```
[09/26/20]seed@VM:~$ ping 192.168.60.1
PING 192.168.60.1 (192.168.60.1) 56(84) bytes of data.
64 bytes from 192.168.60.1: icmp_seq=1 ttl=64 time=0.886 ms
64 bytes from 192.168.60.1: icmp_seq=2 ttl=64 time=0.364 ms
64 bytes from 192.168.60.1: icmp_seq=3 ttl=64 time=0.288 ms
64 bytes from 192.168.60.1: icmp_seq=4 ttl=64 time=0.306 ms
64 bytes from 192.168.60.1: icmp_seq=5 ttl=64 time=0.380 ms
64 bytes from 192.168.60.1: icmp_seq=6 ttl=64 time=0.274 ms
64 bytes from 192.168.60.1: icmp_seq=7 ttl=64 time=0.322 ms
```

Task 2.a:

```
[09/26/20]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 08:00:27:2e:d0:60 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.5/24 brd 10.0.2.255 scope global dynamic enp0s3
       valid_lft 719sec preferred_lft 719sec
    inet6 fe80::58b1:b122:5294:a985/64 scope link
       valid_lft forever preferred_lft forever
3: tun0: <POINTOPOINT,MULTICAST,NOARP> mtu 1500 qdisc noop state DOWN group defa
ult qlen 500
    link/none
[09/26/20]seed@VM:~$
```

Task 2.b:

```
[09/26/20]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 08:00:27:2e:d0:60 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.5/24 brd 10.0.2.255 scope global dynamic enp0s3
       valid_lft 1063sec preferred_lft 1063sec
    inet6 fe80::58b1:b122:5294:a985/64 scope link
       valid_lft forever preferred_lft forever
3: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast sta
te UNKNOWN group default qlen 500
    link/none
    inet 192.168.53.99/24 scope global tun0
       valid_lft forever preferred_lft forever
    inet6 fe80::9fd1:f1dd:acc8:5bca/64 scope link flags 800
       valid_lft forever preferred_lft forever
```

Task 2.c:

Ping 192.168.53.1 –c 5

```
###[ IP ]###
  version   = 4
  ihl       = 5
  tos       = 0x0
  len       = 84
  id        = 46134
  flags     = DF
  frag      = 0
  ttl       = 64
  proto     = icmp
  chksum    = 0x9abd
  src       = 192.168.53.99
  dst       = 192.168.53.1
  \options   \
###[ ICMP ]###
     type      = echo-request
     code      = 0
     chksum    = 0x44d5
     id        = 0x10f9
     seq       = 0x5
###[ Raw ]###
        load      = 'R\xe8o_\xef\xe1\x05\x00\x08\t\n\x0b\x0c\r\x0e\x0f\x10\x11\x
12\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x1f !"#$%&\'()*+,-./01234567'
```

因为默认在一个网段内，所以从 tun 口发出指向 192.168.53.1 的网卡

Ping 192.168.60.1 –c 5
没有任何反应，因为不在同一网段内

Task 2.d:

Ping 192.168.53.1 –c 5

1	2020-09-26 21:30:49.56960…	192.168.53.99	192.168.53.1	ICMP	84	Echo (ping) request
2	2020-09-26 21:30:49.57464…	1.2.3.4	192.168.53.99	ICMP	84	Echo (ping) request
3	2020-09-26 21:30:50.57741…	192.168.53.99	192.168.53.1	ICMP	84	Echo (ping) request
4	2020-09-26 21:30:50.58226…	1.2.3.4	192.168.53.99	ICMP	84	Echo (ping) request
5	2020-09-26 21:30:51.60080…	192.168.53.99	192.168.53.1	ICMP	84	Echo (ping) request
6	2020-09-26 21:30:51.60789…	1.2.3.4	192.168.53.99	ICMP	84	Echo (ping) request
7	2020-09-26 21:30:52.62497…	192.168.53.99	192.168.53.1	ICMP	84	Echo (ping) request
8	2020-09-26 21:30:52.62949…	1.2.3.4	192.168.53.99	ICMP	84	Echo (ping) request
9	2020-09-26 21:30:53.64892…	192.168.53.99	192.168.53.1	ICMP	84	Echo (ping) request
10	2020-09-26 21:30:53.65374…	1.2.3.4	192.168.53.99	ICMP	84	Echo (ping) request

将负载改为 123456789
如下图所示：

```
0000  45 00 00 1d 00 01 00 00  40 00 80 cf 01 02 03 04   E....... @.......
0010  c0 a8 35 63 31 32 33 34  35 36 37 38 39            ..5c1234 56789
```

Task 3:

Testing:

```
 Inside: 192.168.53.99 --> 192.168.53.66
10.0.2.5:43624 --> 0.0.0.0:9090
 Inside: 192.168.53.99 --> 192.168.53.66
10.0.2.5:43624 --> 0.0.0.0:9090
 Inside: 192.168.53.99 --> 192.168.53.66
10.0.2.5:43624 --> 0.0.0.0:9090
 Inside: 192.168.53.99 --> 192.168.53.66
10.0.2.5:43624 --> 0.0.0.0:9090
 Inside: 192.168.53.99 --> 192.168.53.66
```

VPN Server 成功收到了来自 VPN Client 的 ping 报文。

在 VPN Client 添加路由记录：

```
[09/27/20]seed@VM:~$ sudo ip route add 192.168.60.0/24 dev tun0
```

```
10.0.2.5:57108 --> 0.0.0.0:9090
 Inside: 192.168.53.99 --> 192.168.60.101
10.0.2.5:57108 --> 0.0.0.0:9090
 Inside: 192.168.53.99 --> 192.168.60.101
10.0.2.5:57108 --> 0.0.0.0:9090
 Inside: 192.168.53.99 --> 192.168.60.101
10.0.2.5:57108 --> 0.0.0.0:9090
 Inside: 192.168.53.99 --> 192.168.60.101
10.0.2.5:57108 --> 0.0.0.0:9090
 Inside: 192.168.53.99 --> 192.168.60.101
```

VPN Server 成功收到相关报文

Task 4:

修改后的 tun_server.py 如下图所示：

```python
#!/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'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 192.168.53.98/24 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))
```

Host V 接收到的报文

→	1	2020-09-27 08:53:51.9631525…	192.168.53.99	192.168.60.101	ICMP	98 Echo (ping…
←	2	2020-09-27 08:53:51.9631781…	192.168.60.101	192.168.53.99	ICMP	98 Echo (ping…
	3	2020-09-27 08:53:52.9744764…	192.168.53.99	192.168.60.101	ICMP	98 Echo (ping…
	4	2020-09-27 08:53:52.9745015…	192.168.60.101	192.168.53.99	ICMP	98 Echo (ping…
	5	2020-09-27 08:53:53.9993244…	192.168.53.99	192.168.60.101	ICMP	98 Echo (ping…
	6	2020-09-27 08:53:53.9993804…	192.168.60.101	192.168.53.99	ICMP	98 Echo (ping…
	7	2020-09-27 08:53:55.0235610…	192.168.53.99	192.168.60.101	ICMP	98 Echo (ping…
	8	2020-09-27 08:53:55.0236015…	192.168.60.101	192.168.53.99	ICMP	98 Echo (ping…
	9	2020-09-27 08:53:56.0466194…	192.168.53.99	192.168.60.101	ICMP	98 Echo (ping…
	10	2020-09-27 08:53:56.0466543…	192.168.60.101	192.168.53.99	ICMP	98 Echo (ping…

Task 5:

VPN Client 端的程序代码：

```
while True:
# this will block until at least one interface is ready
    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:
            # Get a packet from the tun interface
            packet = os.read(tun, 2048)
            if True:
                # Send the packet via the tunnel
                sock.sendto(packet, ("10.0.2.7", 9090))
```

VPN Server 端的程序代码：

```
IP_A = "0.0.0.0"
PORT = 9090
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock.bind((IP_A, PORT))
while True:
    # this will block until at least one interface is ready
    ready, _, _ = select.select([sock, tun], [], [])
    for fd in ready:
        if fd is sock:
            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))
        if fd is tun:
            packet = os.read(tun, 2048)
            pkt= IP(packet)
            print("From tun ==>: {} --> {}".format(pkt.src, pkt.dst))
            sock.sendto(packet, ("10.0.2.5", 9090))
```

VPN Client 端的 ping 结果：

```
[09/27/20]seed@VM:~$ ping -c 5 192.168.60.101
PING 192.168.60.101 (192.168.60.101) 56(84) bytes of data.
64 bytes from 192.168.60.101: icmp_seq=1 ttl=63 time=6.11 ms
64 bytes from 192.168.60.101: icmp_seq=2 ttl=63 time=3.81 ms
64 bytes from 192.168.60.101: icmp_seq=3 ttl=63 time=4.11 ms
64 bytes from 192.168.60.101: icmp_seq=4 ttl=63 time=3.92 ms
64 bytes from 192.168.60.101: icmp_seq=5 ttl=63 time=4.15 ms
```

Task 6:

```
[09/28/20]seed@VM:~$ telnet 192.168.60.101
Trying 192.168.60.101...
Connected to 192.168.60.101.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
VM login: seed
Password:
Last login: Fri Sep 18 04:57:50 EDT 2020 from 127.0.0.1 on pts/18
/usr/lib/update-notifier/update-motd-fsck-at-reboot:[:59: integer ex
ected:            0
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.

[09/28/20]seed@VM:~$
```

telnet 连接成功如上图所示。
关闭 telnet 服务器端的转发功能后，输入不进去。
重新打开 telnet 的服务器端后，又能成功输入。

```
[09/28/20]seed@VM:~$ telnet 192.168.60.101
Trying 192.168.60.101...
Connected to 192.168.60.101.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
VM login: seed
Password:
Last login: Mon Sep 28 04:50:53 EDT 2020 from 192.168.53.99 on pts/17
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.

[09/28/20]seed@VM:~$ ss
```

Task 7:

删除默认路由后，ping 不通，

```
[09/28/20]seed@VM:~$ ping -c 5 192.168.60.101
PING 192.168.60.101 (192.168.60.101) 56(84) bytes of data.

--- 192.168.60.101 ping statistics ---
5 packets transmitted, 0 received, 100% packet loss, time 4086ms
```

```
[09/28/20]seed@VM:~$ sudo ip route add 192.168.53.0/24 dev enp0s3 via 192.168.60
.1
```

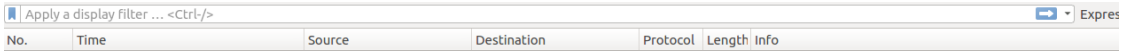
添加路由表相关项

```
[09/28/20]seed@VM:~$ ping -c 5 192.168.60.101
PING 192.168.60.101 (192.168.60.101) 56(84) bytes of data.
64 bytes from 192.168.60.101: icmp_seq=1 ttl=63 time=3.73 ms
64 bytes from 192.168.60.101: icmp_seq=2 ttl=63 time=4.47 ms
64 bytes from 192.168.60.101: icmp_seq=3 ttl=63 time=4.02 ms
64 bytes from 192.168.60.101: icmp_seq=4 ttl=63 time=3.58 ms
64 bytes from 192.168.60.101: icmp_seq=5 ttl=63 time=4.85 ms

--- 192.168.60.101 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4007ms
rtt min/avg/max/mdev = 3.586/4.135/4.858/0.473 ms
```

成功 ping 通

Task 8:



从 HostV 主机打开 wireshark，没有看到任何报文记录，但是能够从 VPN Server 端看到记录，说明 VPN Server 没有进行转发。

因为路由器启用了反向路径过滤措施，即检查以 192.168.30.0/24 为目的网段的报文是否从 tun0 传输出去，路由表中显然没有这一项，所以进行添加。

```
[09/28/20]seed@VM:~$ sudo ip route add  192.168.30.0/24 dev tun0
```

在 host V 中添加相关记录

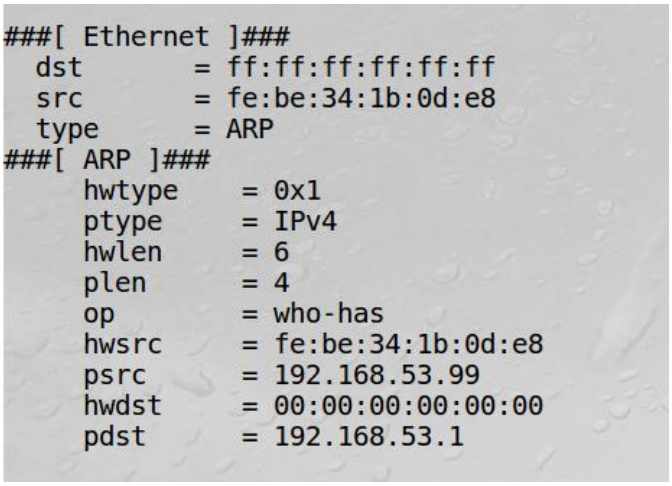
```
[09/28/20]seed@VM:~$ sudo ip route add 192.168.30.0/24 dev enp0s3 via 192.168.60
.1
```

成功 ping 通

```
[09/28/20]seed@VM:~$ ping -c 5 192.168.60.101
PING 192.168.60.101 (192.168.60.101) 56(84) bytes of data.
64 bytes from 192.168.60.101: icmp_seq=1 ttl=63 time=3.63 ms
64 bytes from 192.168.60.101: icmp_seq=2 ttl=63 time=4.52 ms
64 bytes from 192.168.60.101: icmp_seq=3 ttl=63 time=4.80 ms
64 bytes from 192.168.60.101: icmp_seq=4 ttl=63 time=4.14 ms
64 bytes from 192.168.60.101: icmp_seq=5 ttl=63 time=3.94 ms

--- 192.168.60.101 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4007ms
rtt min/avg/max/mdev = 3.631/4.210/4.807/0.418 ms
```

Task 9:



截取到了 ping 之前 ARP 的查询报文，如上图所示：
但是由于 ARP 没有回应，所以 host unreachable。

学号：57117127
姓名：贺博文