

Firewall Evasion Lab Report

Task 1: VM Setup

VM1 VPN Client 10.0.2.4

VM2 VPN Server 10.0.2.7

Task 2: Set up Firewall

```
$ sudo ufw deny out on enp0s3 from 10.0.2.4 to 202.120.224.81
```

```
[12/03/20]seed@VM:~$ sudo ufw deny out on enp0s3 from 10.0.2.4 to 202.120.224.81
Rule added
[12/03/20]seed@VM:~$ sudo ufw status
Status: active

To Action From
--
202.120.224.81 DENY OUT 10.0.2.4 on enp0s3

[12/03/20]seed@VM:~$ ping www.fudan.edu.cn
PING www.fudan.edu.cn (202.120.224.81) 56(84) bytes of data.
ping: sendmsg: Operation not permitted
^C
--- www.fudan.edu.cn ping statistics ---
1 packets transmitted, 0 received, 100% packet loss, time 0ms
```

Task 3: Bypassing Firewall using VPN

Step 1: Run VPN Server

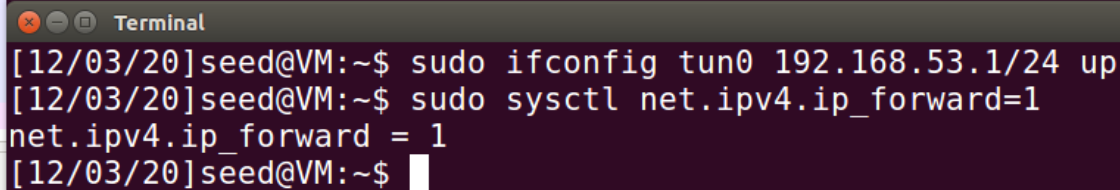
```
$ make
$ sudo ./vpnservice
```

以太网适配器 VirtualBox Host-Only Network:

```
连接特定的 DNS 后缀 . . . . . :
IPv4 地址 . . . . . : 192.168.56.1
子网掩码 . . . . . : 255.255.255.0
默认网关. . . . . :
```

```
$ sudo ifconfig tun0 192.168.56.1/24 up
$ sudo sysctl net.ipv4.ip_forward=1
```

```
[12/03/20]seed@VM:~/.../vpn$ make
gcc -o vpnserver vpnserver.c
gcc -o vpnclient vpnclient.c
[12/03/20]seed@VM:~/.../vpn$ sudo ./vpnserver
```



```
[12/03/20]seed@VM:~$ sudo ifconfig tun0 192.168.53.1/24 up
[12/03/20]seed@VM:~$ sudo sysctl net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
[12/03/20]seed@VM:~$
```

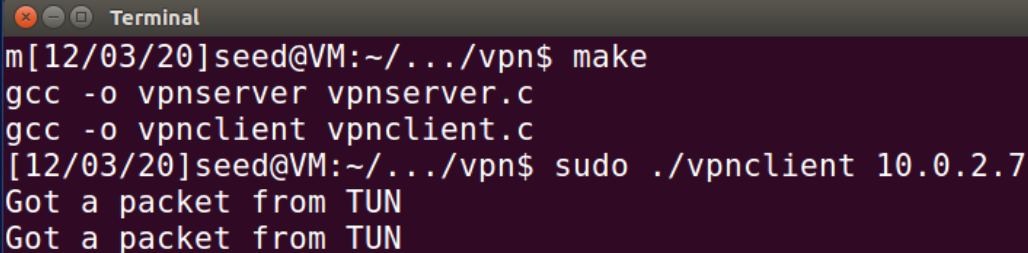
Step 2: Run VPN Client

```
// vpnclient.c line 12
#define SERVER_IP "127.0.0.1" --> #define SERVER_IP "10.0.2.7"
```

```
$ make
$ sudo ./vpnclient
```

```
$ sudo ifconfig tun0 192.168.56.5/24 up
```

```
[12/03/20]seed@VM:~$ sudo ifconfig tun0 192.168.53.5/24 up
[12/03/20]seed@VM:~$
```



```
m[12/03/20]seed@VM:~/.../vpn$ make
gcc -o vpnserver vpnserver.c
gcc -o vpnclient vpnclient.c
[12/03/20]seed@VM:~/.../vpn$ sudo ./vpnclient 10.0.2.7
Got a packet from TUN
Got a packet from TUN
```

Step 3: Set Up Routing on Client and Server VMs

```
# Client VM
$ sudo route add -net 202.120.224.0/24 tun0
$ sudo route add -net 192.168.56.0/24 tun0

# Server VM
$ sudo route add -net 192.168.56.0/24 tun0
```

Step 4: Set Up NAT on Server VM

```
# Server VM
$ sudo iptables -F
$ sudo iptables -t nat -F
$ sudo iptables -t nat -A POSTROUTING -j MASQUERADE -o enp0s3
# -F 清空规则链
# -t 表名(raw、mangle、nat、filter)
# -A 追加规则
# -j 指定如何处理
# -o 匹配出口网卡流出的数据
```

```
# Demonstration on Client VM
$ ping www.fudan.edu.cn -c 1
```

```
[12/03/20]seed@VM:~/Desktop$ ping www.fudan.edu.cn -c 1
PING www.fudan.edu.cn (202.120.224.81) 56(84) bytes of data.
64 bytes from 224.fudan.edu.cn (202.120.224.81): icmp_seq=1 ttl=250
time=12.6 ms

--- www.fudan.edu.cn ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 12.674/12.674/12.674/0.000 ms
```

Wireshark of different interfaces on client VM

Interface tun0

从tun0接口的数据包来看，ping的数据流是在给client VM上的tun0分配的IP地址和目的域名的IP地址之间传输的。

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	2020-12-03 06:43:13.9978689...	192.168.56.5	202.120.224.81	ICMP	84	Echo (ping) request id=0xb84, seq=1/256, ttl=64 (no response found!)
2	2020-12-03 06:43:14.0096757...	202.120.224.81	192.168.56.5	ICMP	84	Echo (ping) reply id=0xb84, seq=1/256, ttl=250 (request in 1)

Interface enp0s3

从enp0s3接口可以看到，实际上是10.0.2.7作为10.0.2.4和目标域名之间的中介完成了这次ping的请求和响应。10.0.2.4发出的ping请求和收到的ping响应实际上是发给10.0.2.7的，然后由10.0.2.7向目标域名发送真正的ping请求并将得到的ping响应发回给10.0.2.4完成这一次ping命令。

4	2020-12-03 06:43:56.1871824...	10.0.2.4	10.0.2.7	UDP	126	58747 → 55555 Len=84
5	2020-12-03 06:43:56.1877063...	10.0.2.7	202.120.224.81	ICMP	98	Echo (ping) request id=0xb8c, seq=1/256, ttl=63 (reply in 6)
6	2020-12-03 06:43:56.2583614...	202.120.224.81	10.0.2.7	ICMP	98	Echo (ping) reply id=0xb8c, seq=1/256, ttl=251 (request in 5)
7	2020-12-03 06:43:56.2584883...	10.0.2.7	10.0.2.4	UDP	126	55555 → 58747 Len=84

Interface any

从any端口可以看到tun0接口的以为的ping命令和实际上10.0.2.4和10.0.2.7的交互的先后顺序。

6	2020-12-03 06:44:43.7729207...	127.0.1.1	127.0.0.1	DNS	94	Standard query response 0xf95 A www.fudan.edu.cn A 202.120.224.81
7	2020-12-03 06:44:43.7730432...	192.168.56.5	202.120.224.81	ICMP	100	Echo (ping) request id=0xb9c, seq=1/256, ttl=64 (reply in 10)
8	2020-12-03 06:44:43.7730717...	10.0.2.4	10.0.2.7	UDP	128	58747 → 55555 Len=84
9	2020-12-03 06:44:43.8374515...	10.0.2.7	10.0.2.4	UDP	128	55555 → 58747 Len=84
10	2020-12-03 06:44:43.8375215...	202.120.224.81	192.168.56.5	ICMP	100	Echo (ping) reply id=0xb9c, seq=1/256, ttl=250 (request in 7)
11	2020-12-03 06:44:43.8377923...	127.0.0.1	127.0.1.1	DNS	89	Standard query 0xf646 PTR 81.224.120.202.in-addr.arpa
12	2020-12-03 06:44:43.8379076...	10.0.2.4	1.0.0.1	DNS	89	Standard query 0xe79 PTR 81.224.120.202.in-addr.arpa
13	2020-12-03 06:44:44.0495873...	1.0.0.1	10.0.2.4	DNS	119	Standard query response 0xe79 PTR 81.224.120.202.in-addr.arpa PTR 224.fudan.edu.cn
14	2020-12-03 06:44:44.0496856...	127.0.1.1	127.0.0.1	DNS	119	Standard query response 0xf646 PTR 81.224.120.202.in-addr.arpa PTR 224.fudan.edu.cn