Firewall Evasion Lab: Bypassing Firewalls using VPN

> *18307130089 吴嘉琪*

VM A:172.16.133.129 client

VM B:172.16.133.130 server

Task 2: Set up Firewall

在A上设置防火墙：

```
sudo ufw deny out on ens33 from 172.16.133.129 to 202.120.224.81
```

访问fudan主页，加载失败

Task 3：Bypassing Firewall using VPN

> *When a user on VM1 tries to access a blocked site, the traffic will not directly go through its network adapter, because it will be blocked. Instead, the packets to the blocked site from VM1 will be routed to the VPN tunnel and arrive at VM2. Once they arrive there, VM2 will route them to the final destination. When the reply packets come back, it will come back to VM2, which will then redirect the packets to the VPN tunnel, and eventually get the packet back to VM1. That is how the VPN helps VM1 to bypass firewalls.*

Step 1: Run VPN Server.

```
[12/01/20]seed@VM:~/.../vpn$ sudo ./vpnserver
```

Terminal

```
[12/01/20]seed@VM:~$ sudo ifconfig tun0 192.168.53.1/24 up
[12/01/20]seed@VM:~$
[12/01/20]seed@VM:~$ sudo sysctl net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
[12/01/20]seed@VM:~$
```

给tunnel的server端分配192.168.53.1的IP地址；

Step 2: Run VPN Client.

```
[12/01/20]seed@VM:~/.../vpn$ sudo ./vpnclient
```

Terminal

```
[12/01/20]seed@VM:~$ sudo ifconfig tun0 192.168.53.5/24 up
[12/01/20]seed@VM:~$
```

给tunnel的client端分配192.168.53.5的IP地址；

Step 3: Set Up Routing on Client and Server VMs.

Client设置：

```
[12/01/20]seed@VM:~$ sudo route add -net 202.120.224.0/24 tun0
```

```
[12/01/20]seed@VM:~$ sudo route add -net 192.168.53.0/24 tun0
```

增加路由，使得所有目标地址是202.120.224.0/24的流量都被导向tun0接口。

server设置：

```
[12/01/20]seed@VM:~$ sudo route add -net 192.168.53.0/24 tun0
[12/01/20]seed@VM:~$
```

增加一条路由，使得所有目标地址是192.68.53.0/24的流量都被导向tun0接口。

Step 4: Set Up NAT on Server VM

When the final destination sends packets back to users, the packet will be sent to the VPN Server first (think about why and write down your answer in the report).

Answer:因为用户的packet是VPNserver的route进行转发的，并且“the source IPs of all the outgoing packets from the Server VM are changed to the NAT’s external IP address”，所以返回的通信数据会先返回到VPN server，然后被server通过tunnel转发回用户。

> *The actual recipient should be the VPN Server VM, even though it does not own the IP address 192.168.53.5.If we can configure the NAT as a gateway, we can ask the NAT to route the packets for 192.168.53.5 to the VPN Server, which will eventually deliver the packets through the tunnel to the VPN Client.*

One idea is to “fool” the NAT to believe that the MAC address of 192.168.53.5 is the VPN Server VM’s MAC address,We can achieve this using an ARP cache poisoning on the NAT, basically telling the NAT before hand about the MAC address of 192.168.53.5.

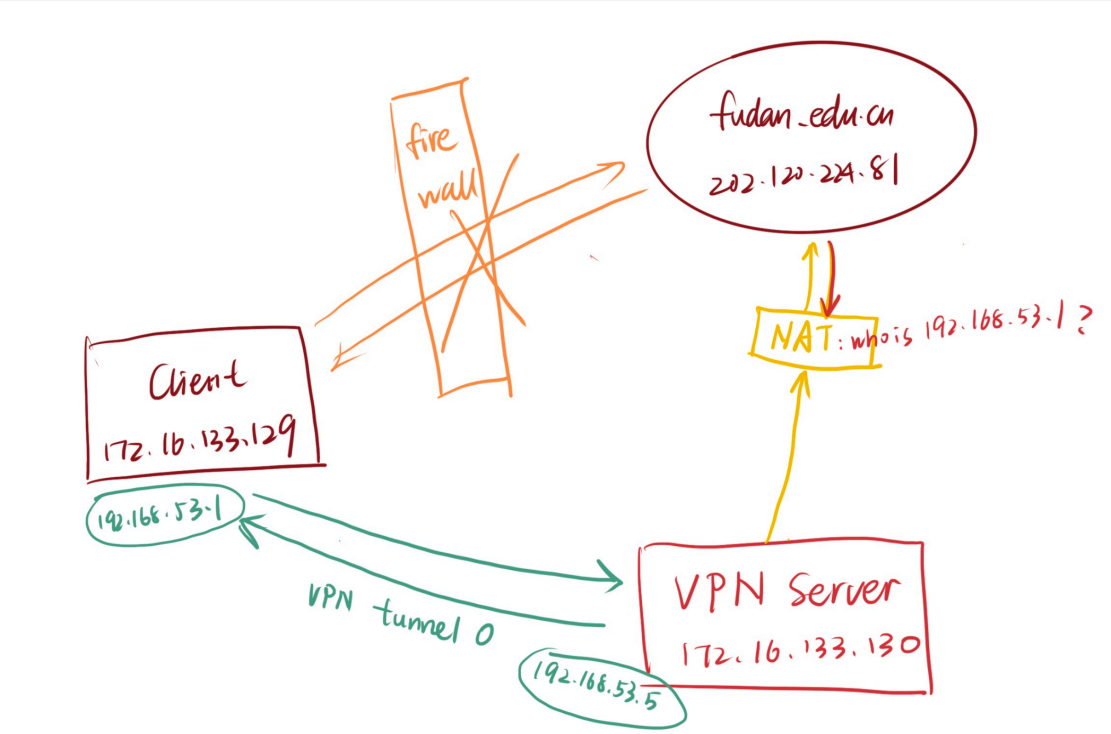
A better solution to get round the limitation of the NAT is to create another NAT right on the Server VM, so all packets coming out of the Server VM will have this VM's IP address as their source IP. To reach the Internet, these packets will go through another NAT, which is provided by VirtualBox, but since the source IP is the Server VM, this second NAT will have no problem relaying back the returned packets from the Internet to the Server VM.

配置VPN server：

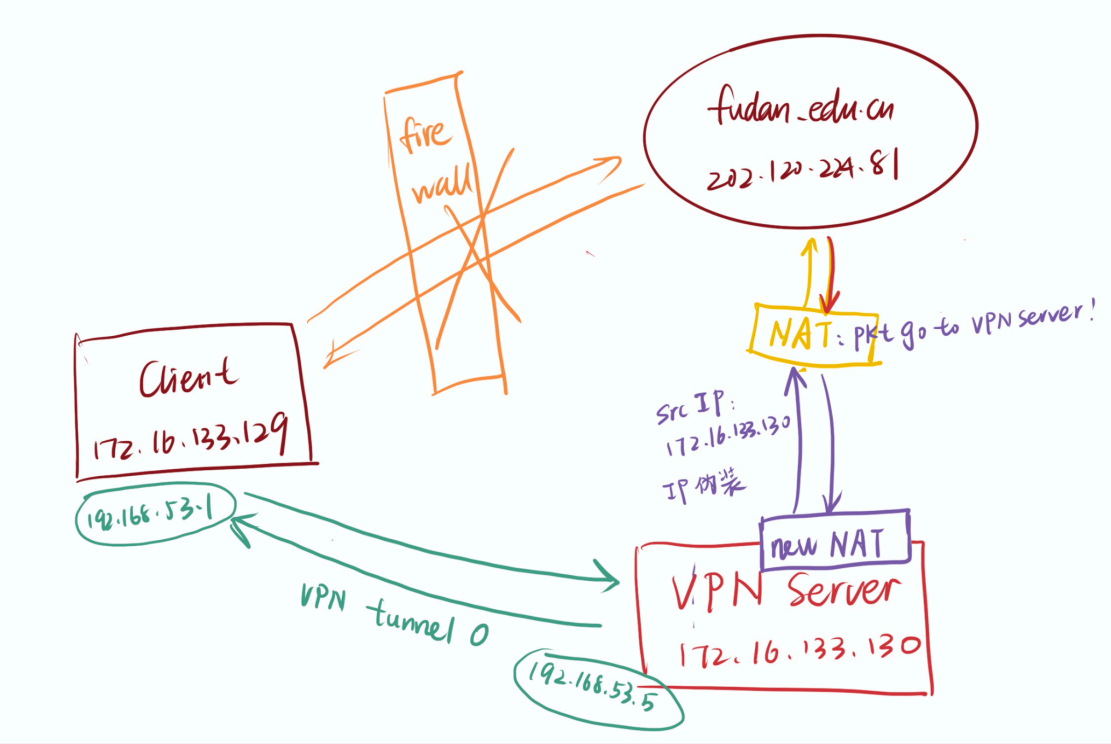
```
Clean all iptables rules: $ sudo iptables -F
$ sudo iptables -t nat -F
Add a rule on postrouting position to the NatNetwork adapter
(eth33) connected to VPN server.
$ sudo iptables -t nat -A POSTROUTING -j MASQUERADE -o ens33
```

> `sudo iptables -t nat -A POSTROUTING -j MASQUERADE -o ens33`:
>
> *Enable the NAT on the Server VM，配置这个新NAT的iptable，使得对于NAT链上发自ens33网络的流量在查询过iptable之后进行IP伪装，再发向VM自带的NAT；*
>
> *当外网数据反回时NAT网络将在en33网络中寻找tun0分配的192.168.53.0/24的MAC地址，而这不会被ARP地址解析命令来找到，所以流量没有办法被正确转发回；这样配置以后NAT会寻找ens33分配的172.16.133.0/24的IP地址，这使得返回流量能够正确返回client。*

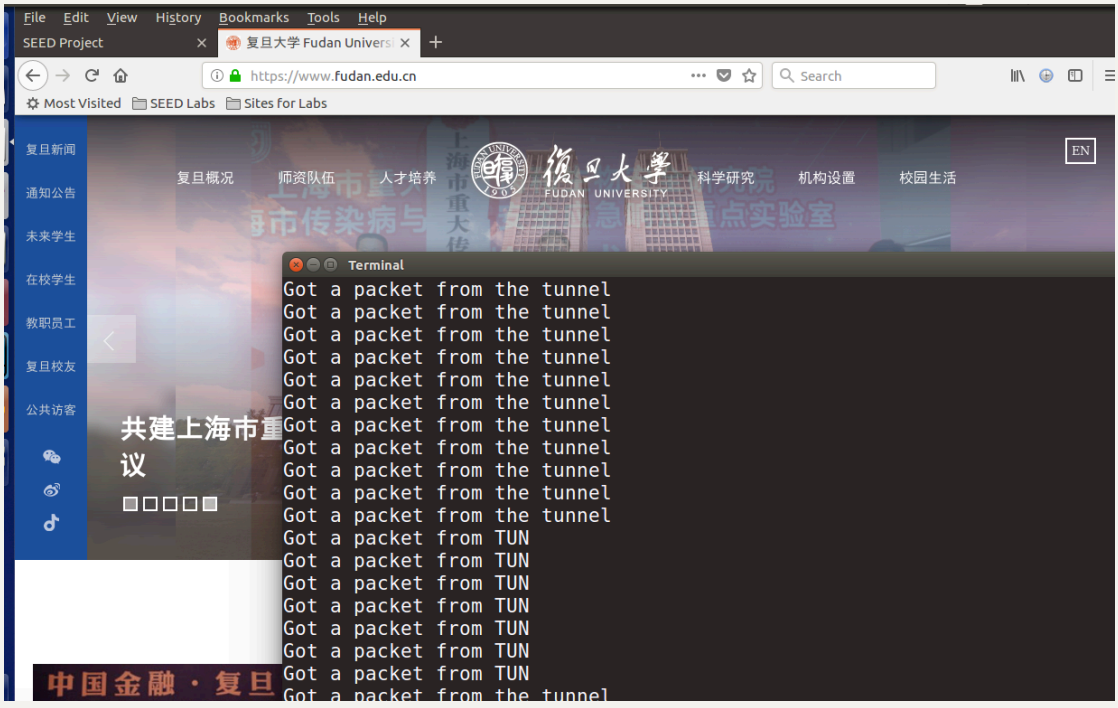
配置前：

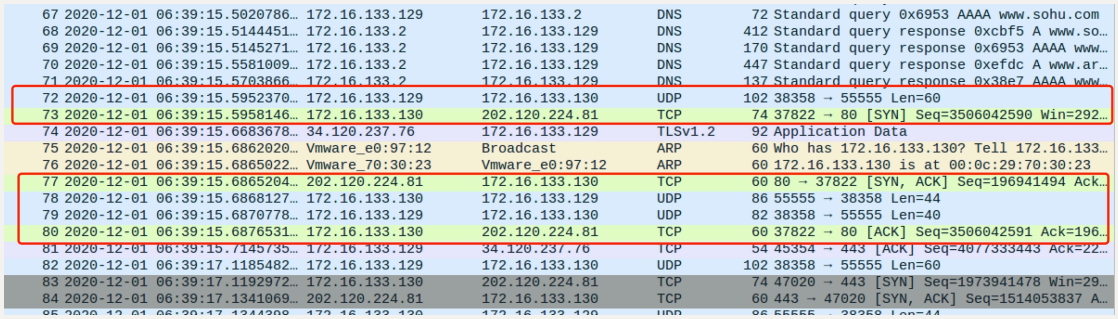


配置后：



配置完成，成功访问fudan.com





通过wireshark查看，client（172.16.133.129）与VPN server之间通过UDP建立了VPN tunnel，从而实现server与fudan.com(202.120.224.81)的通信；最后返回内容通过VPN tunnel又被传到client上，实现了client绕过防火墙对fudan.com的访问