# Linux Firewall Exploration Lab

> *18307130089 吴嘉琪*

VM A:172.16.133.129

VM B:172.16.133.130

## Task 1: Using Firewall

配置主机A:

```
# Set the default input policy to ACCEPT, DROP, or REJECT. Please note that if
# you change this you will most likely want to adjust your rules.
DEFAULT_INPUT_POLICY="ACCEPT"
```

允许自定义修改规定

• Prevent A from doing telnet to Machine B.

```
[11/30/20]seed@VM:~$ sudo ufw deny out from 172.16.133.129 to 172.16.133.130
Rule added
```

```
[11/30/20]seed@VM:~$ telnet 172.16.133.130
Trying 172.16.133.130...
```

成功阻止telnet

• Prevent B from doing telnet to Machine A.

```
[11/30/20]seed@VM:~$ sudo ufw deny in from 172.16.133.130 to 172.16.133.129 port
 23
Rule added
```

```
[11/30/20]seed@VM:~/.../WEB$ telnet 172.16.133.129
Trying 172.16.133.129...
```

成功阻止telnet

• Prevent A from visiting an external web site. You can choose any web site that you like to block, but keep in mind, some web servers have multiple IP addresses.

阻止A访问 fudan.edu.cn(202.120.224.81):

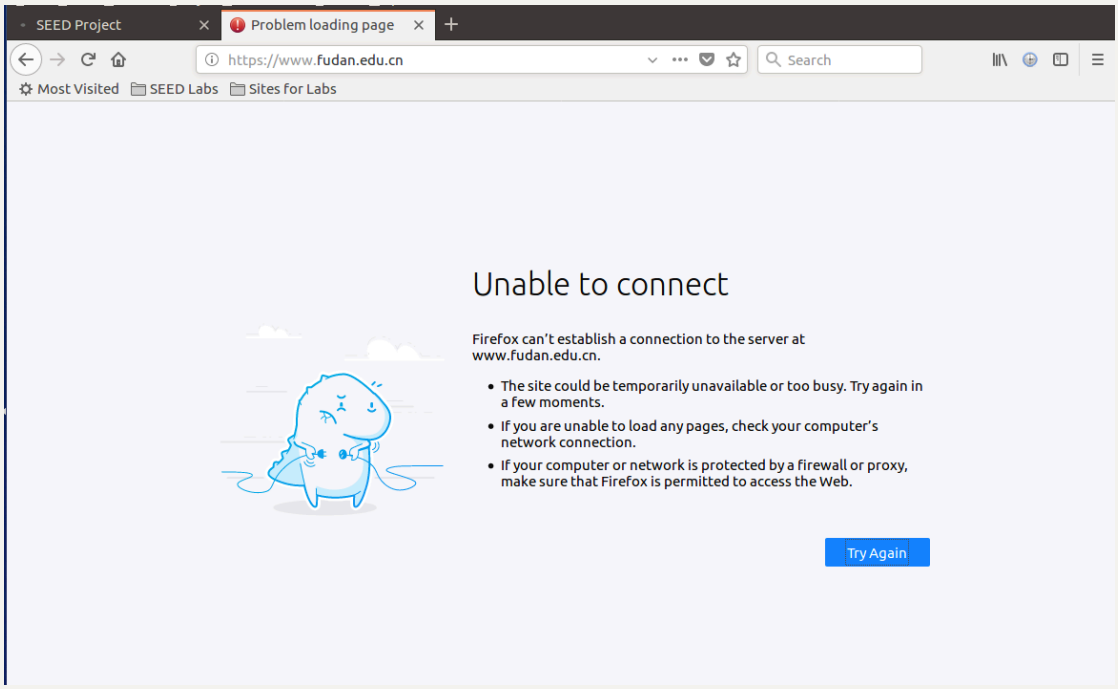
对HTTP服务

```
[11/30/20]seed@VM:~$ sudo ufw deny out from 172.16.133.129 to 202.120.224.81 por
t 80
Rule added
```

对HTTPS服务

```
[11/30/20]seed@VM:~$ sudo ufw deny out from 172.16.133.129 to 202.120.224.81 por
t 443
Rule added
```

加载失败




## Task 2: Implementing a Simple Firewall

```c
#include <linux/kernel.h>
#include <linux/module.h>
#include <linux/netfilter.h>
#include <linux/netfilter_ipv4.h>
#include <linux/ip.h>
#include <linux/tcp.h>
#include <linux/string.h>
#define MAX_RULE_NUM 10

static struct nf_hook_ops FilterHookRule[MAX_RULE_NUM];
static int regist_num = 0;
```

```
int eq_daddr(const struct iphdr *iph, const char *ip_addr)
{
    //check if the dst ip address equals the given address
    char source[16];
    snprintf(source, 16, "%pI4", &iph->daddr);
    if (strcmp(source, ip_addr) == 0)
        return 1;
    return 0;
}

int eq_saddr(const struct iphdr *iph, const char *ip_addr)
{
    //check if the src ip address equals the given address
    char source[16];
    snprintf(source, 16, "%pI4", &iph->saddr);
    if (strcmp(source, ip_addr) == 0)
        return 1;
    return 0;
}

unsigned int telnetFilter_1(void *priv, struct sk_buff *skb,
                            const struct nf_hook_state *state)
// rule for task 1.1: Prevent A from doing `telnet` to Machine B
{
    struct iphdr *iph;
    struct tcphdr *tcph;

    iph = ip_hdr(skb);
    tcph = (void *)iph + iph->ihl * 4;

    if (iph->protocol == IPPROTO_TCP && tcph->dest == htons(23)
&& eq_daddr(iph, "172.16.133.130") && eq_saddr(iph,
"172.16.133.129"))
    {
        printk(KERN_INFO "kill telnet:Dropping telnet from %pI4
packet to %pI4\n", &iph->saddr, &iph->daddr);
        return NF_DROP;
    }
    else
    {
        return NF_ACCEPT;
    }
}

unsigned int telnetFilter_2(void *priv, struct sk_buff *skb,
```

```
                                const struct nf_hook_state *state)
// rule for task 1.2: Prevent B from doing `telnet` to Machine A
{
    struct iphdr *iph;
    struct tcphdr *tcph;

    iph = ip_hdr(skb);
    tcph = (void *)iph + iph->ihl * 4;

    if (iph->protocol == IPPROTO_TCP && tcph->dest == htons(23)
&& eq_daddr(iph, "172.16.133.129") && eq_saddr(iph,
"172.16.133.130"))
    {
        printk(KERN_INFO "kill telnet:Dropping telnet from %pI4
packet to %pI4\n", &iph->saddr, &iph->daddr);
        return NF_DROP;
    }
    else
    {
        return NF_ACCEPT;
    }
}

unsigned int block_fudan(void *priv, struct sk_buff *skb,
                            const struct nf_hook_state *state)
// rule for task 1.3: Prevent B from doing `telnet` to Machine A
// assume the host ip of "noteboke.xyli.me" is: `104.18.21.226`
and `103.235.46.191` (obtained by `wireshark`)
// please NOTE that `netfliter` cannot block a domain for that
there are usually multiple ip addresses for a domain.
{
    struct iphdr *iph;
    struct tcphdr *tcph;
    iph = ip_hdr(skb);
    tcph = (void *)iph + iph->ihl * 4;
    if (tcph->dest == htons(80) || (tcph->dest == htons(443)) &&
(eq_daddr(iph, "202.120.224.81"))&& (eq_saddr(iph,
"172.16.133.129")))
    {
        printk(KERN_INFO "Dropping http/https from %pI4 packet to
%pI4\n", &iph->saddr, &iph->daddr);
        return NF_DROP;
    }
    else
    {
        return NF_ACCEPT;
```

```
    }
}

int setUpFilter(void)
{
    int i;
    printk(KERN_INFO "Registering filters.\n");
    FilterHookRule[0] = (struct nf_hook_ops){.hook =
telnetFilter_1, .hooknum = NF_INET_LOCAL_OUT, .pf = PF_INET,
.priority = NF_IP_PRI_FIRST};
    FilterHookRule[1] = (struct nf_hook_ops){.hook =
telnetFilter_2, .hooknum = NF_INET_LOCAL_IN, .pf = PF_INET,
.priority = NF_IP_PRI_FIRST};
    FilterHookRule[2] = (struct nf_hook_ops){.hook = block_fudan,
.hooknum = NF_INET_LOCAL_OUT, .pf = PF_INET, .priority =
NF_IP_PRI_FIRST};

    // set the amount of filter rules
    regist_num = 3;

    for (i = 0; i < regist_num; i++)
        nf_register_hook(&FilterHookRule[i]);
    return 0;
}

void removeFilter(void)
{
    int i;
    printk(KERN_INFO "Filters are being removed.\n");
    //unregist hooks one by one
    for (i = 0; i < regist_num; i++)
        nf_unregister_hook(&FilterHookRule[i]);
    regist_num = 0;
}

module_init(setUpFilter);
module_exit(removeFilter);

MODULE_LICENSE("GPL");
```

makefile :

```
obj-m += sfw.o
all:
  make -C /lib/modules/$(shell uname -r)/build M=$(PWD) modules

clean:
  make -C /lib/modules/$(shell uname -r)/build M=$(PWD) clean
```

make之后运行：

```
sudo insmod sfw.ko
```

A telnet B：

```
[12/01/20]seed@VM:~/.../minifw-2$ telnet 172.16.133.130
Trying 172.16.133.130...
```

输入dmesg

```
[ 2861.515525] Dropping http/https from 172.16.133.129 packet to 34.107.221.82
[ 2861.900359] kill telnet:Dropping telnet from 172.16.133.129 packet to 172.16.
133.130
[ 2863.915103] kill telnet:Dropping telnet from 172.16.133.129 packet to 172.16.
133.130
[ 2868.172424] kill telnet:Dropping telnet from 172.16.133.129 packet to 172.16.
133.130
[ 2876.364717] kill telnet:Dropping telnet from 172.16.133.129 packet to 172.16.
133.130
```

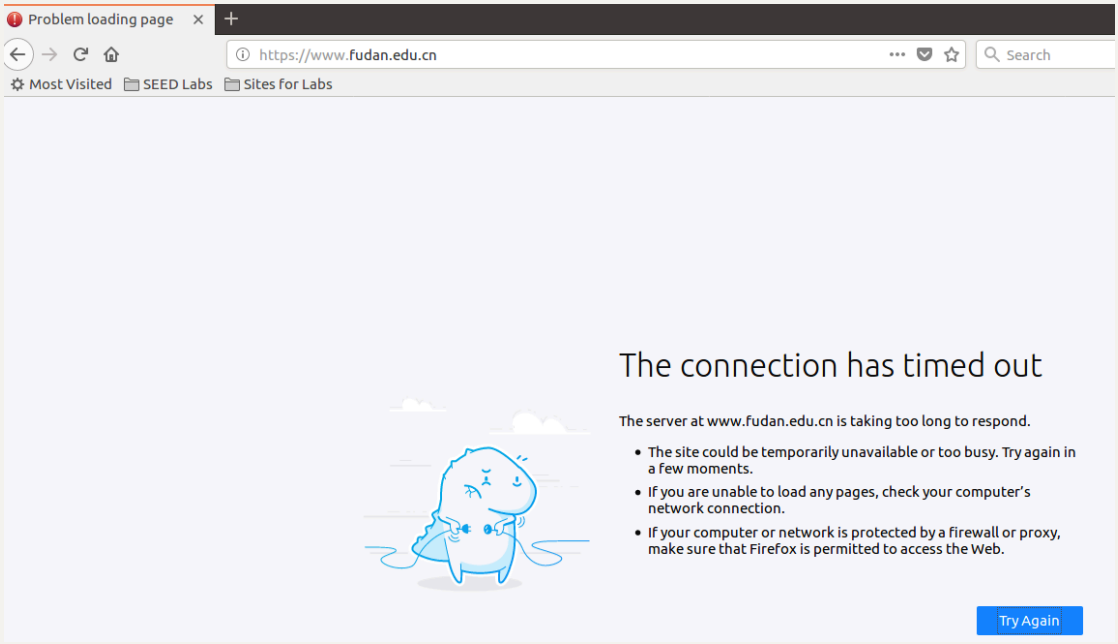
B telnet A

```
[12/01/20]seed@VM:~/.../minifw-2$ telnet 172.16.133.129
Trying 172.16.133.129...
```

输入dmesg

```
[ 2911.675963] kill telnet:Dropping telnet from 172.16.133.130 packet to 172.16.133.129
[ 2911.724216] Dropping http/https from 172.16.133.129 packet to 34.107.221.82
[ 2911.947070] Dropping http/https from 172.16.133.129 packet to 34.107.221.82
[ 2912.685833] kill telnet:Dropping telnet from 172.16.133.130 packet to 172.16.133.129
[ 2913.740967] Dropping http/https from 172.16.133.129 packet to 34.107.221.82
[ 2913.963190] Dropping http/https from 172.16.133.129 packet to 34.107.221.82
[ 2914.702372] kill telnet:Dropping telnet from 172.16.133.130 packet to 172.16.133.129
```

A 访问fudan.edu.cn(202.120.224.81):




输入dmesg命令查看：

```
[ 2555.728917] Dropping http/https from 172.16.133.129 packet to 34.107.221.82
[ 2555.998881] Dropping http/https from 172.16.133.129 packet to 34.107.221.82
[ 2556.731168] Dropping http/https from 172.16.133.129 packet to 34.107.221.82
[ 2557.017459] Dropping http/https from 172.16.133.129 packet to 34.107.221.82
[ 2558.746248] Dropping http/https from 172.16.133.129 packet to 34.107.221.82
[ 2559.034195] Dropping http/https from 172.16.133.129 packet to 34.107.221.82
[ 2560.707087] Dropping http/https from 172.16.133.129 packet to 34.107.221.82
[ 2560.971296] Dropping http/https from 172.16.133.129 packet to 34.107.221.82
[ 2561.721887] Dropping http/https from 172.16.133.129 packet to 34.107.221.82
[ 2561.977699] Dropping http/https from 172.16.133.129 packet to 34.107.221.82
[ 2563.738896] Dropping http/https from 172.16.133.129 packet to 34.107.221.82
[ 2563.993995] Dropping http/https from 172.16.133.129 packet to 34.107.221.82
[ 2595.733182] Dropping http/https from 172.16.133.129 packet to 34.107.221.82
[ 2596.007565] Dropping http/https from 172.16.133.129 packet to 34.107.221.82
[ 2596.762379] Dropping http/https from 172.16.133.129 packet to 34.107.221.82
[ 2597.017849] Dropping http/https from 172.16.133.129 packet to 34.107.221.82
[ 2598.779427] Dropping http/https from 172.16.133.129 packet to 34.107.221.82
[ 2599.034230] Dropping http/https from 172.16.133.129 packet to 34.107.221.82
```

移除模块：

```
sudo rmmod sfw
```

# Task 3: Evading Egress Filtering

## Task 3.a: Telnet to Machine B through the firewall

在A上用防火墙阻止向B的telnet；

在A上运行：`ssh -L 8000:172.16.133.130:23 seed@172.16.133.131`

先建立与C（172.16.133.131）的ssh链接，然后再间接telnet B；

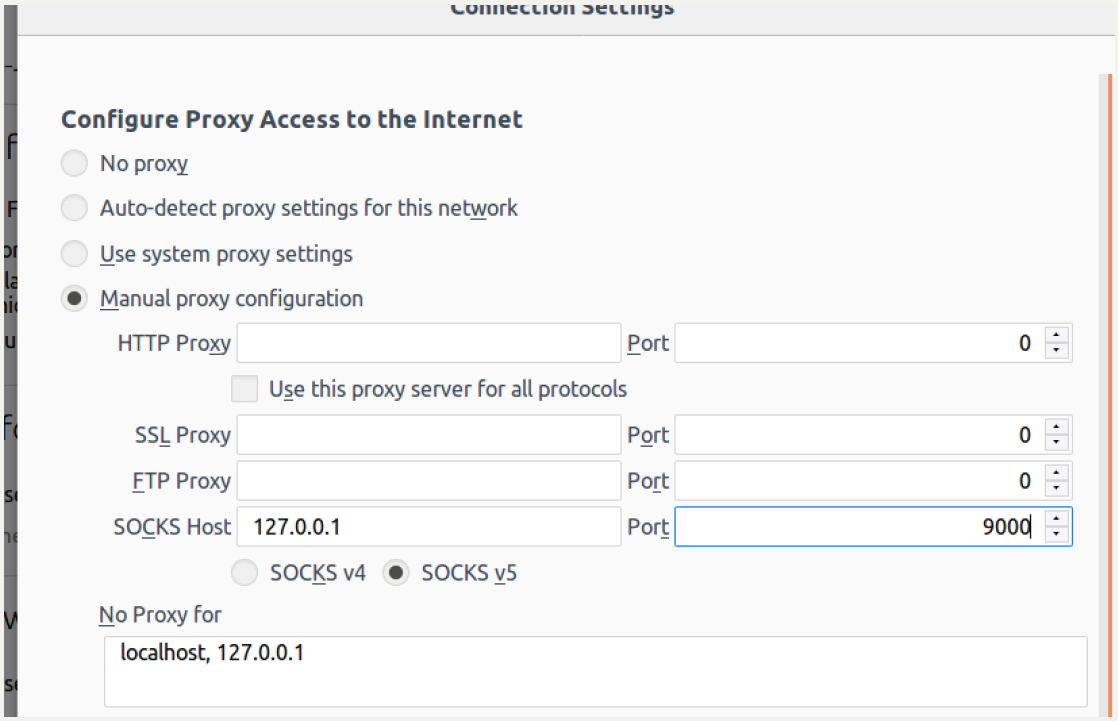
```
[12/01/20]seed@VM:~/.../minifw-2$ ssh -L 8000:172.16.133.130 seed@172.16.133.129
Bad local forwarding specification '8000:172.16.133.130'
[12/01/20]seed@VM:~/.../minifw-2$ ssh -L 8000:172.16.133.130:23 seed@172.16.133.
131
The authenticity of host '172.16.133.131 (172.16.133.131)' can't be established.
ECDSA key fingerprint is SHA256:p1zAio6c1bI+8HDp5xa+eKRi561aFDaPE1/xq1eYzCI.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '172.16.133.131' (ECDSA) to the list of known hosts.
seed@172.16.133.131's password:
Permission denied, please try again.
seed@172.16.133.131's password:
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.

Last login: Mon Oct  5 06:07:28 2020 from 172.16.133.128
[12/01/20]seed@VM:~$ telnet 172.16.133.130
Trying 172.16.133.130...
Connected to 172.16.133.130.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
VM login:
```

**Task 3.b: Connect to Fudan using SSH Tunnel**

A上用防火墙阻止向fudan.edu.cn 的访问

> *we use a more generic approach, called dynamic port forwarding, instead of a static one like that in Task 3.a. To do that, we only specify the local port number, not the final destination. When Machine B receives a packet from the tunnel, it will dynamically decide where it should forward the packet to based on the destination information of the packet.*

设置浏览器



然后运行：

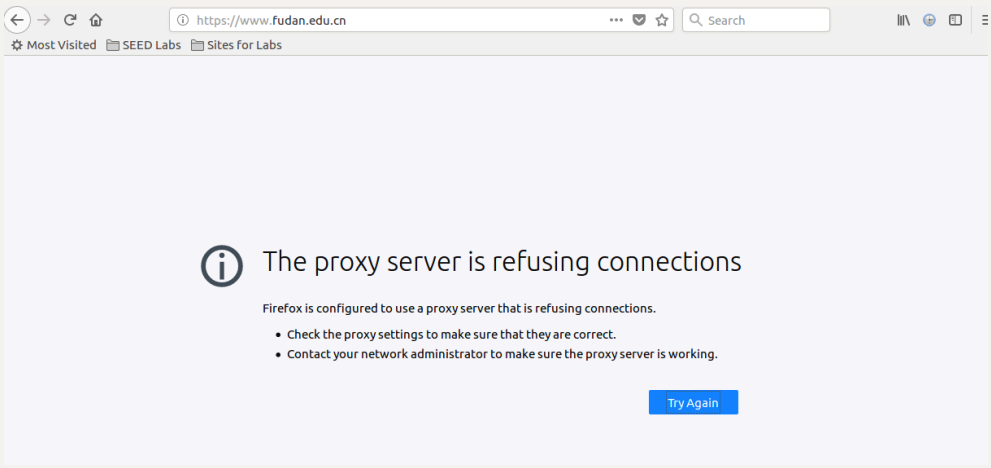
```
ssh -D 9000 -C seed@172.16.133.130
```

1. Run Firefox and go visit the Facebook page. Can you see the Facebook page? Please describe your observation.


正常访问

2. Afteryougetthefacebookpage,breaktheSSHtunnel,cleartheFirefoxcache,andtryt heconnection again. Please describe your observation.

   终端ssh链接后无法再访问



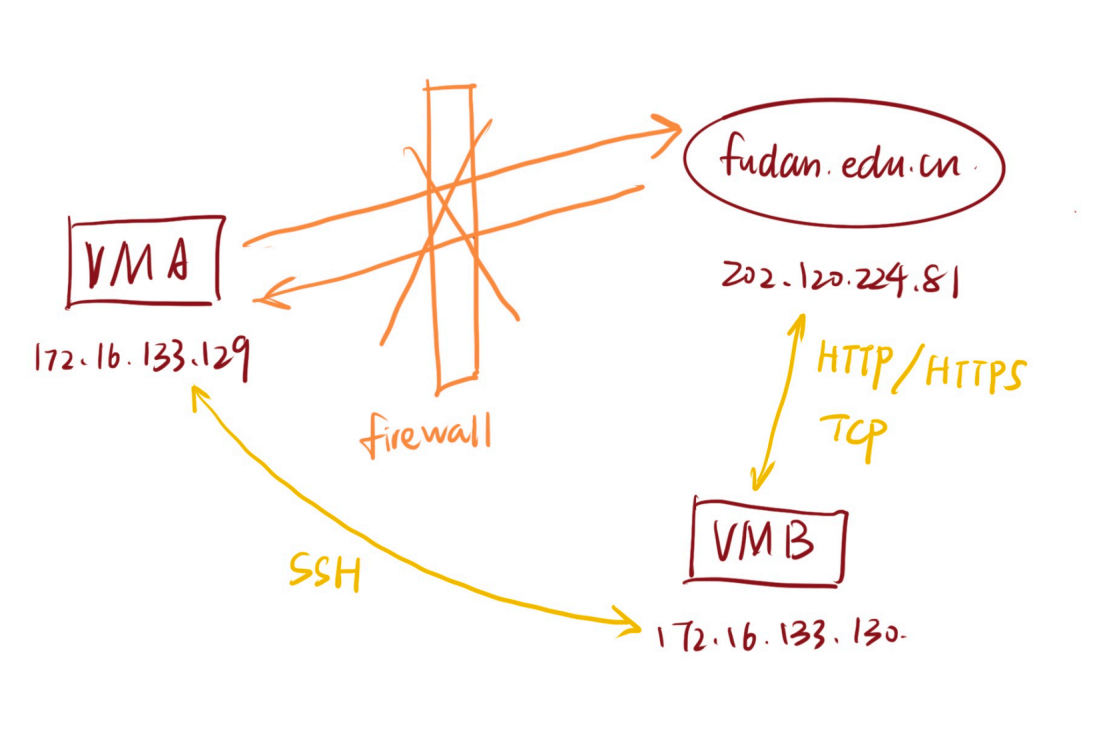

3. Establish the SSH tunnel again and connect to Facebook. Describe your observation.

   可以再次被打开

4. Please explain what you have observed, especially on why the SSH tunnel can help bypass the egress filtering. You should use Wireshark to see what exactly is happening on the wire. Please describe your observations and explain them using the packets that you have captured.

| | | | | | | |
|---|---|---|---|---|---|---|
| 8 | 2020-12-01 07:38:23.5507773… | 172.16.133.129 | 172.16.133.130 | SSH | 150 | Client: Encrypted packet (len=84) |
| 9 | 2020-12-01 07:38:23.5516618… | 172.16.133.130 | 202.120.224.81 | TCP | 74 | 45446 → 443 [SYN] Seq=1131662301 Win=29… |
| 10 | 2020-12-01 07:38:23.5654438… | 202.120.224.81 | 172.16.133.130 | TCP | 60 | 443 → 45446 [SYN, ACK] Seq=1619776060 A… |
| 11 | 2020-12-01 07:38:23.5659214… | 172.16.133.130 | 202.120.224.81 | TCP | 60 | 45446 → 443 [ACK] Seq=1131662302 Ack=16… |
| 12 | 2020-12-01 07:38:23.5659258… | 172.16.133.130 | 172.16.133.129 | SSH | 110 | Server: Encrypted packet (len=44) |
| 13 | 2020-12-01 07:38:23.5659588… | 172.16.133.129 | 172.16.133.130 | TCP | 66 | 57444 → 22 [ACK] Seq=818115804 Ack=8872… |
| 14 | 2020-12-01 07:38:23.5696812… | 172.16.133.129 | 172.16.133.130 | SSH | 454 | Client: Encrypted packet (len=388) |
| 15 | 2020-12-01 07:38:23.5703868… | 172.16.133.130 | 202.120.224.81 | TLSv1.2 | 571 | Client Hello |
| 16 | 2020-12-01 07:38:23.5703899… | 202.120.224.81 | 172.16.133.130 | TCP | 60 | 443 → 45446 [ACK] Seq=1619776061 Ack=11… |
| 17 | 2020-12-01 07:38:23.5818289… | 202.120.224.81 | 172.16.133.130 | TLSv1.2 | 195 | Server Hello, Change Cipher Spec, Encry… |
| 18 | 2020-12-01 07:38:23.5820902… | 172.16.133.130 | 202.120.224.81 | TCP | 60 | 45446 → 443 [ACK] Seq=1131662819 Ack=16… |
| 19 | 2020-12-01 07:38:23.5822212… | 172.16.133.130 | 172.16.133.129 | SSH | 246 | Server: Encrypted packet (len=180) |
| 20 | 2020-12-01 07:38:23.5827655… | 172.16.133.129 | 172.16.133.130 | SSH | 150 | Client: Encrypted packet (len=84) |
| 21 | 2020-12-01 07:38:23.5831972… | 172.16.133.130 | 202.120.224.81 | TLSv1.2 | 105 | Change Cipher Spec, Hello Request, Hell… |
| 22 | 2020-12-01 07:38:23.5831995… | 202.120.224.81 | 172.16.133.130 | TCP | 60 | 443 → 45446 [ACK] Seq=1619776202 Ack=11… |
| 23 | 2020-12-01 07:38:23.5834907… | 172.16.133.129 | 172.16.133.130 | SSH | 286 | Client: Encrypted packet (len=220) |
| 24 | 2020-12-01 07:38:23.5837646… | 172.16.133.129 | 172.16.133.130 | SSH | 470 | Client: Encrypted packet (len=404) |
| 25 | 2020-12-01 07:38:23.5838859… | 172.16.133.130 | 172.16.133.129 | TCP | 66 | 22 → 57444 [ACK] Seq=887257262 Ack=8181… |
| 26 | 2020-12-01 07:38:23.5840152… | 172.16.133.130 | 202.120.224.81 | TLSv1.2 | 231 | Application Data |
| 27 | 2020-12-01 07:38:23.5840170… | 202.120.224.81 | 172.16.133.130 | TCP | 60 | 443 → 45446 [ACK] Seq=1619776202 Ack=11… |

查看抓包结果，A首先通过ssh链接与B进行通信，之后与fudan.edu.cn的TCP/HTTP/TLS协议交互全都是B（172.16.133.130）进行，得到的数据包再由B传回A；通过B这个中介，A成功绕过了与fudan.edu.cn之间的防火墙。




## Task 4: Evading Ingress Filtering

> *https://www.howtogeek.com/428413/what-is-reverse-ssh-tunneling-and-how-to-use-it/*
>
> *https://unix.stackexchange.com/questions/46235/how-does-reverse-ssh-tunneling-work*

首先在A上设置防火墙，阻止B的http和ssh请求：

```
[12/01/20]seed@VM:~$ sudo ufw deny in from 172.16.133.130 to 172.16.133.129 port
 22
Rules updated
[12/01/20]seed@VM:~$ sudo ufw deny in from 172.16.133.130 to 172.16.133.129 port
 80
Rules updated
```

阻止B的incoming链接

create a reverse `ssh` tunnel on machine A:

```
[12/01/20]seed@VM:~$ ssh -R 10022:localhost:22 seed@172.16.133.130
```

A已经成功建立了到B的ssh链接，并且会监听B的10022端口；

现在在B上运行：

```
[12/01/20]seed@VM:~$ ssh -p 10022 seed@localhost
```

B的请求会被传递到A的22端口上，实现了B向A的链接；

```
[12/01/20]seed@VM:~$ ssh -p 10022 seed@localhost
The authenticity of host '[localhost]:10022 ([127.0.0.1]:10022)' can't be established.
ECDSA key fingerprint is SHA256:p1zAio6c1bI+8HDp5xa+eKRi561aFDaPE1/xq1eYzCI.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[localhost]:10022' (ECDSA) to the list of known hosts.
seed@localhost's password:
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.

Last login: Tue Oct 20 07:58:38 2020 from 172.16.133.128
[12/01/20]seed@VM:~$ cd Desktop
[12/01/20]seed@VM:~/Desktop$ ls
attacker_vm  this is 129          WEB
lab2         vmware-tools-distrib
```