

James Oswald  
ICSI 424 Computer Security  
Lab 10

## Task 1: Using Firewall

For this task I use machine A as being 10.0.2.4, while machine B is 10.0.2.5

**This was my first attempt, I then realized that I should be setting it to block the outgoing connection on the host machine rather than the incoming connection on the destination machine, Please scroll down to “Task 1 Fixed” For the right way the lab specifies.**

I begin by setting up to prevent A from doing telnet to Machine B. To do this I set machine B to drop incoming packets on port 23 from machine As IP.

To test that this works, first I try Telneting to machine B from Machine A.

```
[11/20/20 J0481765]seed@VM:~$ telnet 10.0.2.5
Trying 10.0.2.5...
Connected to 10.0.2.5.
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-gener
ic i686)
```

Now that we know telnet works by default, I then set machine B to drop incoming packets on port 23 from machine As IP.

```
[11/20/20 J0481765]seed@VM:~$ sudo iptables -A INPUT -p
tcp --destination-port 23 -s 10.0.2.4 -j DROP
[11/20/20 J0481765]seed@VM:~$
```

Now trying to telnet again on machine A

```
[11/20/20 J0481765]seed@VM:~$ telnet 10.0.2.5
Trying 10.0.2.5...
█
```

It is stuck on the attempting to connect line.

I verify that I can connect normally from my host machine as well to ensure that i blocked the correct IP and not just all connections on port 23

```
C:\Users\James>telnet 10.0.2.5
Trying 10.0.2.5...
Connected to 10.0.2.5.
```

Next I perform the same process to block incoming telnet connections from machine B to machine A.

```
[11/20/20 J0481765]seed@VM:~$ sudo iptables -A INPUT -p
tcp --destination-port 23 -s 10.0.2.5 -j DROP
[11/20/20 J0481765]seed@VM:~$
```

Now trying to connect to machine A from machine B using telnet

```
[11/20/20 J0481765]seed@VM:~$ telnet 10.0.2.4
Trying 10.0.2.4...

```

I get the same result, It's stuck on trying to connect.

**Task 1 Fixed:** While doing this I read the first part about how we should be setting the firewall on the host machine, So i readjusted my methods to block outgoing telnet connections to a server from the host machine.

I clear all IP tables on both machines using "sudo iptables -F"

```
[11/20/20 J0481765]seed@VM:~$ sudo iptables -F
```

```
[11/20/20 J0481765]seed@VM:~$ sudo iptables -F
```

On machine A I then block outgoing connections to machine B on port 23, then test it by running telnet to machine B.

```
[11/20/20 J0481765]seed@VM:~$ sudo iptables -A OUTPUT -p
tcp --dport telnet -d 10.0.2.5 -j REJECT
[11/20/20 J0481765]seed@VM:~$ telnet 10.0.2.5
Trying 10.0.2.5...
telnet: Unable to connect to remote host: Connection re
fused
[11/20/20 J0481765]seed@VM:~$
```

I then do the exact same thing for machine B to machine A and test it.

```
[11/20/20 J0481765]seed@VM:~$ sudo iptables -A OUTPUT -  
p tcp --dport telnet -d 10.0.2.4 -j REJECT  
[11/20/20 J0481765]seed@VM:~$ telnet 10.0.2.4  
Trying 10.0.2.4...  
telnet: Unable to connect to remote host: Connection re  
fused  
[11/20/20 J0481765]seed@VM:~$ █
```

Finally I set machine A to block a specific website. Of course this could be an issue since many sites have multiple IP addresses. While trying to figure out how to do this I've come across two approaches, 1) block on every IP you can find for the website or 2) use string pattern matching to block requests that contain the domain name of the website.

I will just be using the first approach by blocking the outgoing requests for a certain IP address or a site I know has only one IP address, namely, my personal site <http://joswald.net/>.

I begin by getting the IP of my site using ping

```
[11/20/20 J0481765]seed@VM:~$ ping joswald.net  
PING joswald.net (192.185.13.60) 56(84) bytes of data.
```

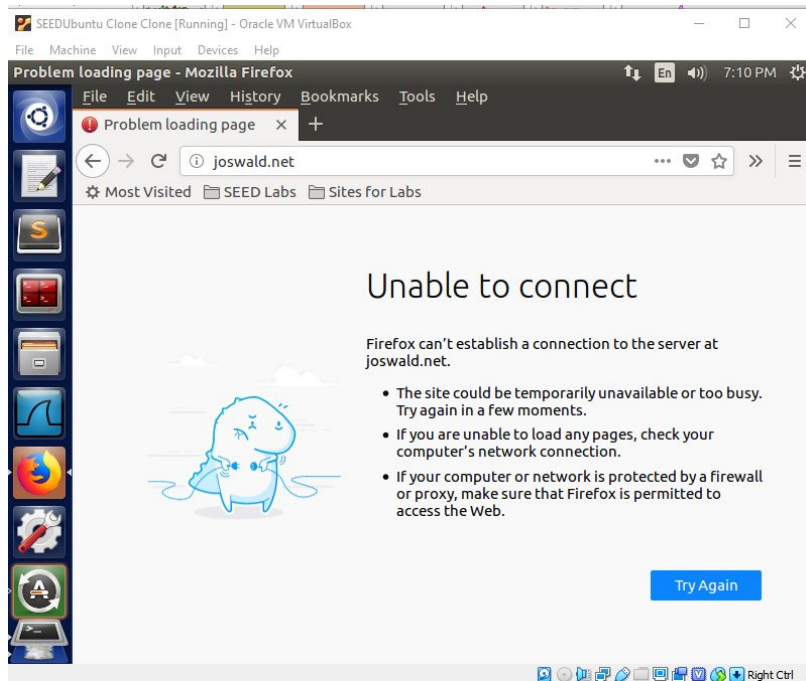
I then use iptables to block outgoing tcp connections on port 80 (this covers both HTTP and HTTPS) with a destination of 192.185.13.60

```
[11/20/20 J0481765]seed@VM:~$ sudo iptables -A OUTPUT -  
p tcp --dport 80 -d 192.185.13.60 -j REJECT
```

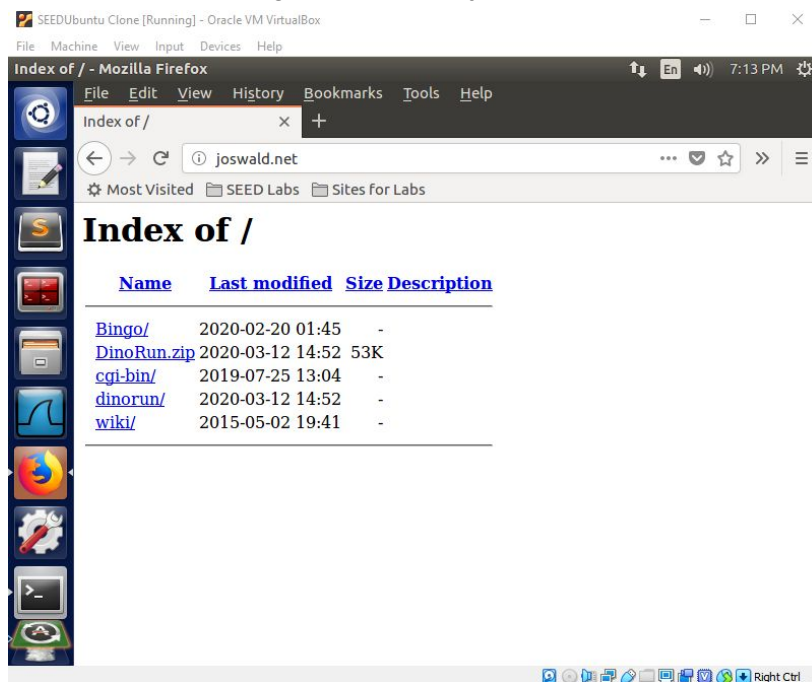
See next page for results

I use firefox to attempt to connect to the site I just blocked on machine A, I also run it on machine B to prove it can connect properly without the firewall rule I just added.

Machine A attempting to connect to joswald.net:



Machine B attempting to connect to joswald.net



Thus I have successfully blocked machine A from accessing a website.



## Task 2: Implementing a Simple Firewall

I begin by flushing all iptable rules so they don't interfere with the rules I set in my LKM

```
[11/20/20 J0481765]seed@VM:~/lab10$ sudo iptables -F
[11/20/20 J0481765]seed@VM:~/lab10$
```

Next I write my filter program is use the raw integer IPV4 addresses computed elsewhere since the code to call `inet_pton` is bulky. I implement 5 rules to block various outgoing services. I pick 5 arbitrary rules, block FTP, SSH, and Telnet to host B as well as 2 websites, `joswald.com` and `example.com`

```
/* This is the hook function itself */
unsigned int hook_func(void *priv, struct sk_buff *skb, const struct nf_hook_ops *nfho)
{
    struct iphdr* ip_header = ip_hdr(skb);
    if(ip_header->protocol != 6) //our filters will only use TCP
        return NF_ACCEPT;

    struct tcphdr* tcp_header = (void*)ip_header+ip_header->ihl*4;
    printk(KERN_INFO "TCP packet to %d to port %d from %d", ip_header->dst, tcp_header->dest, ip_header->src);

    //rule 1, block telnet from A to B
    //inet_pton("10.2.0.5")
    if(ip_header->daddr == 84017162 && tcp_header->dest == htons(23))
        return NF_DROP;
    //rule 2 block website joswald.net
    //inet_pton("192.185.13.60")
    if(ip_header->daddr == 1007532480 && tcp_header->dest == htons(80))
        return NF_DROP;
    //rule 3 block SSH from A to B
    //inet_pton("10.2.0.5")
    if(ip_header->daddr == 84017162 && tcp_header->dest == htons(22))
        return NF_DROP;
    //rule 4 block FTP from A to B
    //inet_pton("10.2.0.5")
    if(ip_header->daddr == 84017162 && tcp_header->dest == htons(21))
        return NF_DROP;
    //rule 5 block website example.com
    //inet_pton("93.184.216.34")
    if(ip_header->daddr == 584628317 && tcp_header->dest == htons(80))
        return NF_DROP;
    return NF_ACCEPT;
}
```

```
/* Initialization routine */
int init_module(){
    nfho.hook = hook_func; /* Fill in our hook structure */
    nfho.hooknum = NF_INET_POST_ROUTING; /* Handler function */
    nfho.pf = PF_INET; /* First hook for IPv4 */
    nfho.priority = NF_IP_PRI_FIRST; /* Make our function first */
    nf_register_hook(&nfho);
    return 0;
}

/* Cleanup routine */
void cleanup_module()
{
    nf_unregister_hook(&nfho);
}
```

I then write the Makefile to compile my LKM

```
obj-m += task2.o

all:
    make -C /lib/modules/$(shell uname -r)/build M=/home/seed/lab10/Task2
clean:
    make -C /lib/modules/$(shell uname -r)/build M=/home/seed/lab10/Task2
```

I then run make to build my LKM

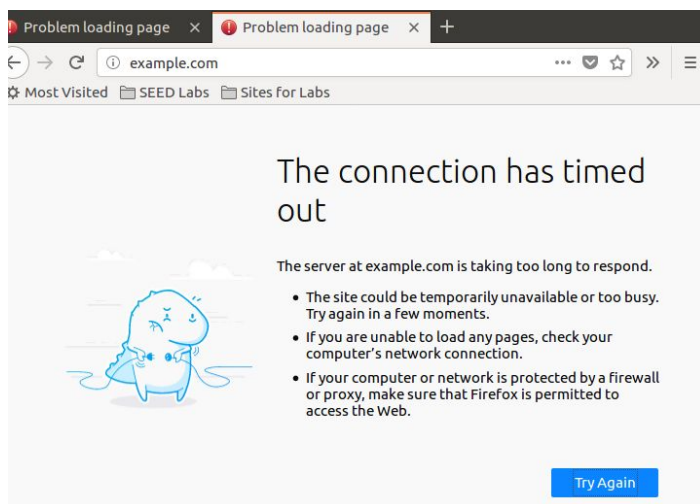
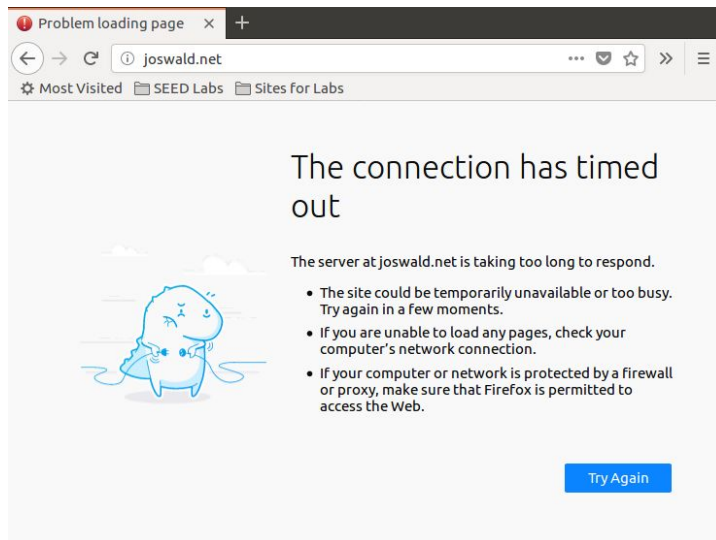
```
[11/20/20 J0481765]seed@VM:~/.../Task2$ make
make -C /lib/modules/4.8.0-36-generic/build M=/home/seed/lab10/Task2 modules
make[1]: Entering directory '/usr/src/linux-headers-4.8.0-36-generic'
  CC [M] /home/seed/lab10/Task2/task2.o
/home/seed/lab10/Task2/task2.c: In function 'hook_func':
/home/seed/lab10/Task2/task2.c:19:2: warning: ISO C90 forbids mixed declarations and code [-Wdeclaration-after-statement]
    struct tcphdr* tcp_header = (void*)ip_header+ip_header->tcp_header_offset;
    ^
Building modules, stage 2.
MODPOST 1 modules
  CC      /home/seed/lab10/Task2/task2.mod.o
  LD [M]  /home/seed/lab10/Task2/task2.ko
make[1]: Leaving directory '/usr/src/linux-headers-4.8.0-36-generic'
[11/20/20 J0481765]seed@VM:~/.../Task2$
```

Finally I enable my LKM by inserting it.

```
[11/20/20 J0481765]seed@VM:~/.../Task2$ sudo insmod task2.ko
[11/20/20 J0481765]seed@VM:~/.../Task2$
```

Since I am using NF\_DROP, we see that requests time out rather than get instantly rejected like before due to the fact that the application is waiting for a response from its packet but we deleted the packet before it went out.

We see that it successfully blocks joswald.net and example.com by dropping the outgoing requests until a time out.



We also see that the telnet connection times out failing to get past the trying screen to 10.0.2.5:

```
[11/20/20 J0481765]seed@VM:~/.../Task2$ telnet 10.0.2.5
Trying 10.0.2.5...
```

The same goes for SSH, and FTP attempts.



### Task 3: Evading Egress Filtering

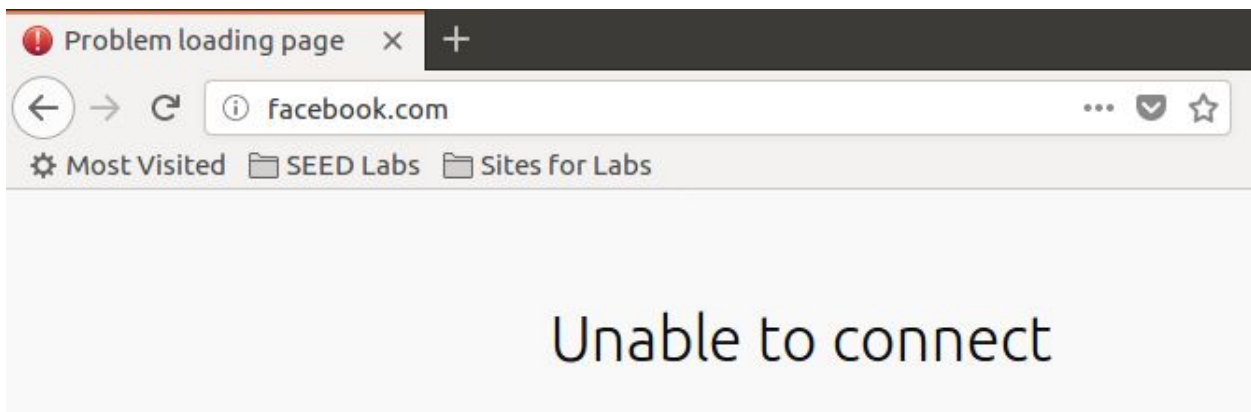
I begin by setting up the iptables rules to block all outgoing telnet traffic:

```
[11/20/20 J0481765]seed@VM:~/.../Task2$ sudo iptables -  
A OUTPUT -p tcp --dport telnet -j REJECT
```

I use dig to find facebook's IP address then block it using iptables.

```
;; ANSWER SECTION:  
facebook.com.      30      IN      A      31.13.6  
7.35
```

```
[11/20/20 J0481765]seed@VM:~/.../Task2$ sudo iptables -  
A OUTPUT -p tcp --dport 80 -d 31.13.67.35 -j REJECT  
[11/20/20 J0481765]seed@VM:~/.../Task2$
```



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

I use an SSH tunnel from machine A to bypass the telnet firewall on machine A. I begin by setting up the tunnel using ssh with the -L flag. I also use -N so we don't get logged in and the -f flag so it will run in the background. I then try telnetting through localhost

```
[11/20/20 J0481765]seed@VM:~$ ssh -f -N seed@10.0.2.5 -  
L 8000:10.0.2.5:23  
seed@10.0.2.5's password:  
[11/20/20 J0481765]seed@VM:~$ telnet localhost 8000  
Trying 127.0.0.1...  
Connected to localhost.  
Escape character is '^]'.  
Ubuntu 16.04.2 LTS  
VM login: █
```



Just to make sure I run IF config and see we are indeed telneted in on machine B.

```
[11/20/20 J0481765]seed@VM:~$ ifconfig
enp0s3      Link encap:Ethernet  HWaddr 08:00:27:e9:ef:03

              inet addr:10.0.2.5  Bcast:10.0.2.255  Mask:255.255.0
```

Looking at wireshare it appears all traffic both to and from machine B is using different ports

No.	Time	Source	Destination	Protocol	Length
338	2020-11-20 23:03:41.4840333...	10.0.2.4	10.0.2.5	TCP	66
339	2020-11-20 23:03:41.4841706...	10.0.2.5	10.0.2.4	SSHv2	126
340	2020-11-20 23:03:41.4841746...	10.0.2.4	10.0.2.5	TCP	66
341	2020-11-20 23:03:41.4843113...	10.0.2.5	10.0.2.4	SSHv2	102
342	2020-11-20 23:03:41.4843152...	10.0.2.4	10.0.2.5	TCP	66
343	2020-11-20 23:03:41.4955625...	10.0.2.5	10.0.2.4	SSHv2	134
344	2020-11-20 23:03:41.4955759...	10.0.2.4	10.0.2.5	TCP	66

▶ Frame 342: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface enp0s3
▶ Ethernet II, Src: PcsCompu_e9:ef:03 (08:00:27:e9:ef:03), Dst: PcsCompu_d0:43:87
▶ Internet Protocol Version 4, Src: 10.0.2.4, Dst: 10.0.2.5
▼ Transmission Control Protocol, Src Port: 53260, Dst Port: 22, Seq: 2182289828, Win: 0, Len: 0
Source Port: 53260
Destination Port: 22

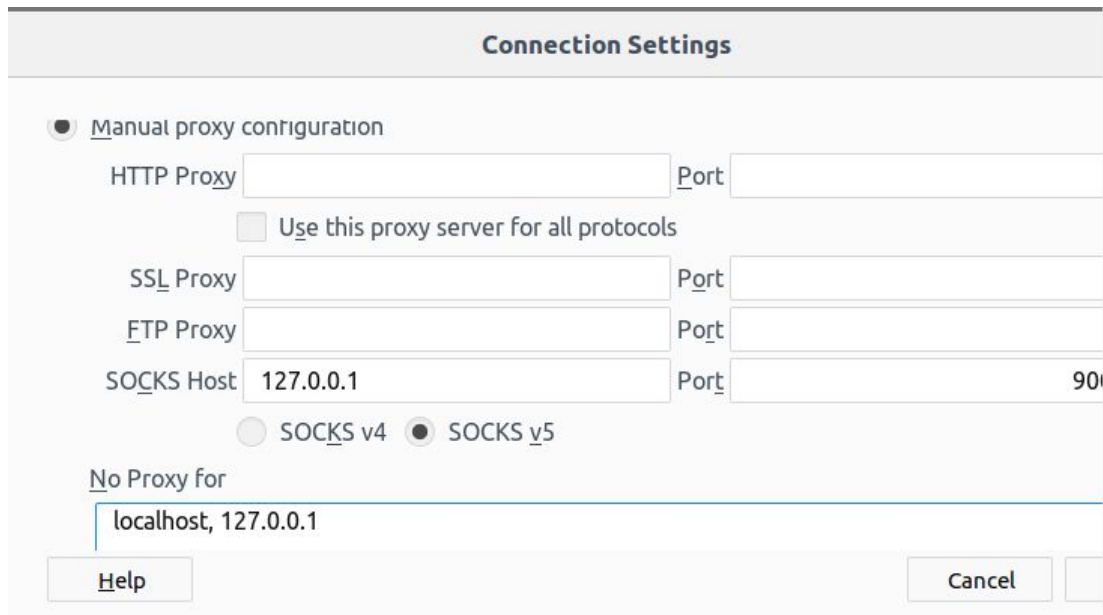
For some reason, the machine A port isn't 8000 but 53260 for both the TCP and SSHv2 packets, meaning there is some internal SSH mechanism changing it client side from 8000 to 53260. Regardless of this. Because port 23 is bypassed, the telnet firewall is completely bypassed as well.

### Task 3.b: Connect to Facebook using SSH Tunnel

I begin setting up my facebook proxy via the SSH tunnel command this time using dynamic port forwarding.

```
[11/20/20 J0481765]seed@VM:~$ ssh -f -N seed@10.0.2.5 -D 9000
seed@10.0.2.5's password:
[11/20/20 J0481765]seed@VM:~$
```

Then I set up firefox to use it as a proxy.



The image shows the 'Connection Settings' dialog box in Firefox. The 'Manual proxy configuration' tab is selected. The 'SOCKS v5' option is chosen under the 'SOCKS Host' section. The 'SOCKS Host' is set to '127.0.0.1' and the 'Port' is set to '9001'. The 'No Proxy for' list contains 'localhost, 127.0.0.1'. There are 'Help', 'Cancel', and 'OK' buttons at the bottom.

Connection Settings

☒ Manual proxy configuration

HTTP Proxy  Port

☐ Use this proxy server for all protocols

SSL Proxy  Port

FTP Proxy  Port

SOCKS Host  Port

☐ SOCKS v4 ☒ SOCKS v5

No Proxy for

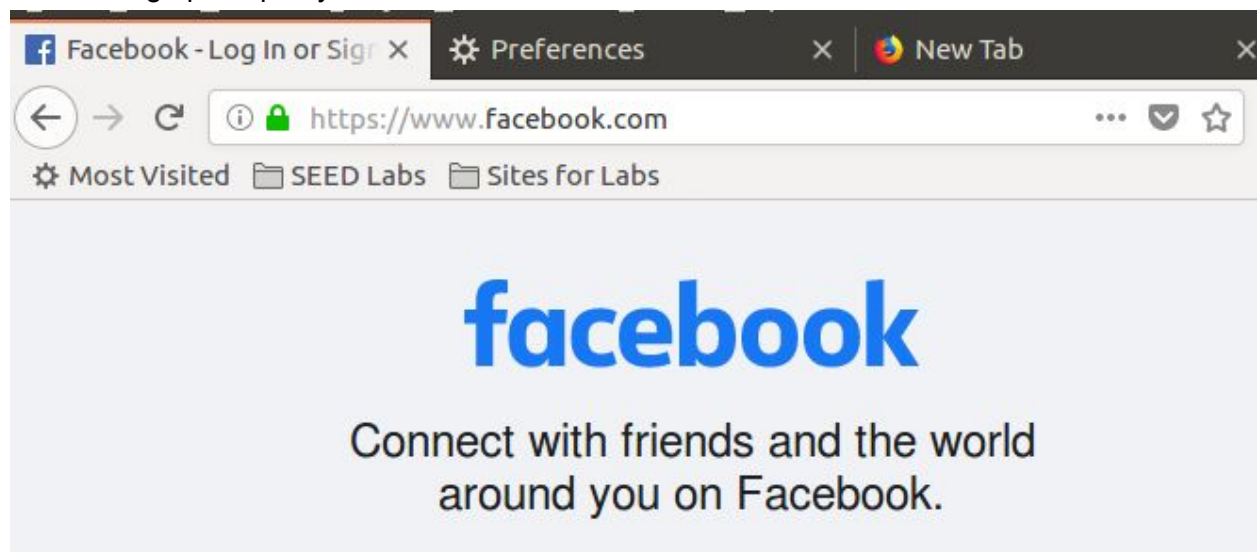
Help Cancel OK

Note: Facebook was unblocked because the IP changed. I reblocked it before doing this using its new IP

```
[11/20/20 J0481765]seed@VM:~$ sudo iptables -A OUTPUT -p tcp --dport 80 -d 157.240.14.35 -j REJECT
[11/20/20 J0481765]seed@VM:~$
```

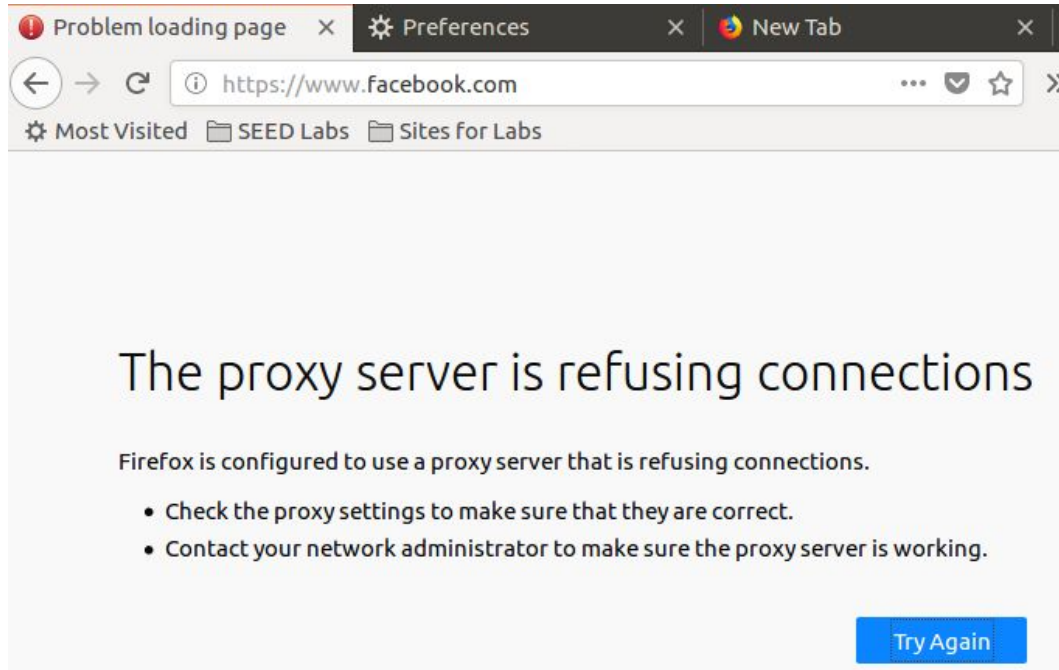
### Question 1:

After setting up the proxy, I see that I can access facebook fine.



### Question 2:

While answering this question I realized that I ran ssh with the -f option but without the -S option, meaning the only way to break the tunnel was to reboot the machine. After I did this I cleared the cache then run try accessing facebook and am told that firefox is using a proxy that is refusing connections. This makes sense since we told all firefox traffic to use the proxy.



### Question 3:

This time when connecting the tunnel, I make sure to use the -S option to name it.

```
[11/20/20 J0481765]seed@VM:~$ ssh -f -N seed@10.0.2.5 -  
D 9000 -S firefoxTunnel  
seed@10.0.2.5's password:
```

I can once again access facebook:

#### Question 4:

I use Wireshark to observe what's happening on the wire. Much like task 3a, I see it's not even using the port 9000, but all traffic is indeed passing through the proxy using SSH and landing at the proxy on port 22 which is to be expected.

No.	Time	Source	Destination	Protocol
197	2020-11-20 23:30:49.4429426...	10.0.2.5	10.0.2.4	SSH
198	2020-11-20 23:30:49.4429576...	10.0.2.4	10.0.2.5	TCP
199	2020-11-20 23:30:49.4435724...	10.0.2.5	10.0.2.4	SSH
200	2020-11-20 23:30:49.4435776...	10.0.2.4	10.0.2.5	TCP
201	2020-11-20 23:30:50.4619397...	10.0.2.4	10.0.2.5	SSH
202	2020-11-20 23:30:50.4689879...	10.0.2.5	10.0.2.4	SSH
203	2020-11-20 23:30:50.4690020...	10.0.2.4	10.0.2.5	TCP

▶ Frame 144: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface eth0

▶ Ethernet II, Src: PcsCompu\_e9:ef:03 (08:00:27:e9:ef:03), Dst: PcsCompu\_d0:43:87:7b:11:00

▶ Internet Protocol Version 4, Src: 10.0.2.4, Dst: 10.0.2.5

▼ Transmission Control Protocol, Src Port: 55460, Dst Port: 22, Seq: 3288020426, Len: 0

Source Port: 55460

Destination Port: 22

[Stream index: 0]

[TCP Segment Len: 0]

Sequence number: 3288020426

#### Task 4: Evading Ingress Filtering

I begin by flushing the iptables

```
[11/20/20 J0481765]seed@VM:~$ sudo iptables -F
```

I set up new rules to block everything on port 80 and incoming SSH on port 22.

```
^C[11/20/20 J0481765]seed@VM:~$ sudo iptables -A OUTPUT -p tcp --dport 80 -j REJECT
[11/20/20 J0481765]seed@VM:~$ sudo iptables -A INPUT -p tcp --dport 80 -j REJECT
[11/20/20 J0481765]seed@VM:~$ sudo iptables -A INPUT -p tcp --dport 22 -j REJECT
[11/20/20 J0481765]seed@VM:~$
```



My thought for solving this seems to have an issue with the fact that a port 23 is already in use and I cant kill the process.

I'd like to set it up so that I develop an SSH tunnel to host B from which I can telnet into to get access to the web server. That way

```
[11/20/20 J0481765]seed@VM:~$ sudo ssh -N seed@10.0.2.5  
-L 23:10.0.2.5:8000 -S telnetTunnel  
seed@10.0.2.5's password:  
bind: Address already in use
```

My second attempt involved doing much the same by trying to get around the firewall by landing on port 80.

```
[11/20/20 J0481765]seed@VM:~$ sudo ssh -N seed@10.0.2.5  
-L 80:10.2.5:8000 -S webTunnel  
seed@10.0.2.5's password:  
listen: Address already in use  
listen: Address already in use  
channel_setup_fwd_listener_tcpip: cannot listen to port  
: 80  
Could not request local forwarding.
```

My final attempt is to use 2 arbitrary ports and wire the webserver to use 4000 instead by catching and changing packets. Then having machine B forward localhost:4000 to this tunnel.

```
[11/20/20 J0481765]seed@VM:~$ sudo ssh -N seed@10.0.2.5  
-L 4000:10.0.2.5:4000 -S finalTunnel  
seed@10.0.2.5's password:
```