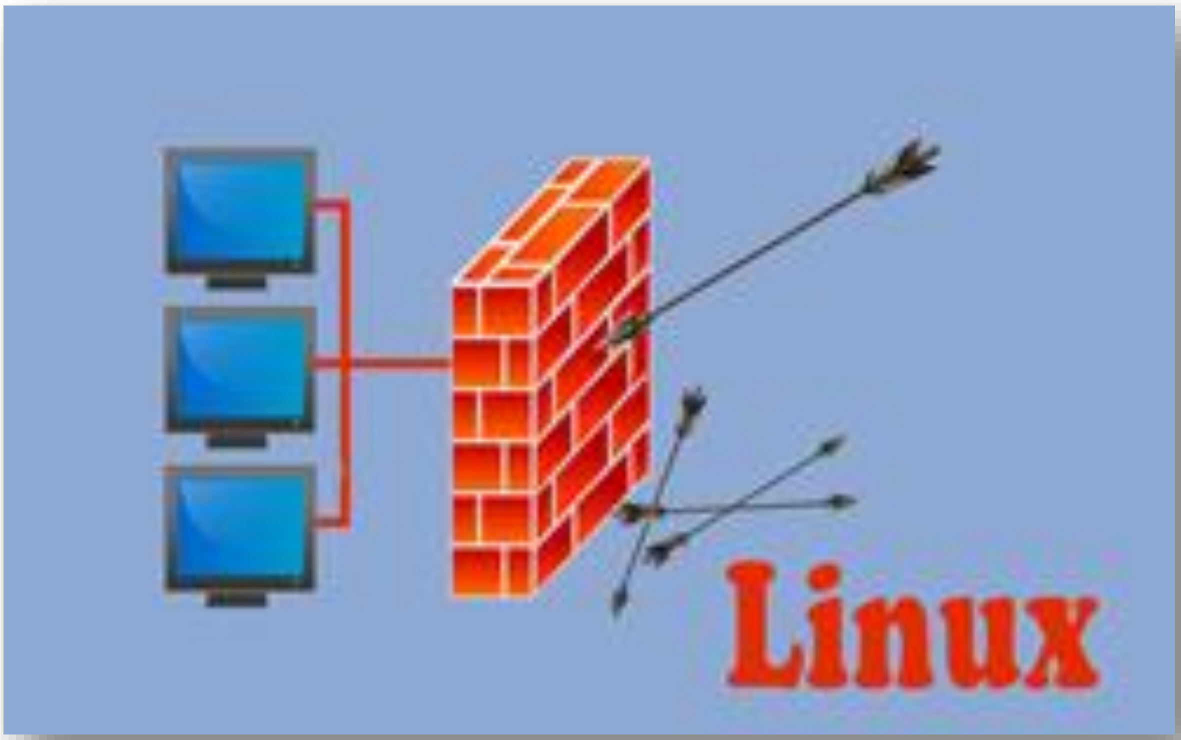
<u>Linux Firewall Exploration SEED Lab</u>



Linux Firewall Exploration SEED Lab

Main Introduction:

In this lab we will focus on packet filter. Packet filters inspect packets and decide whether to drop or forward.

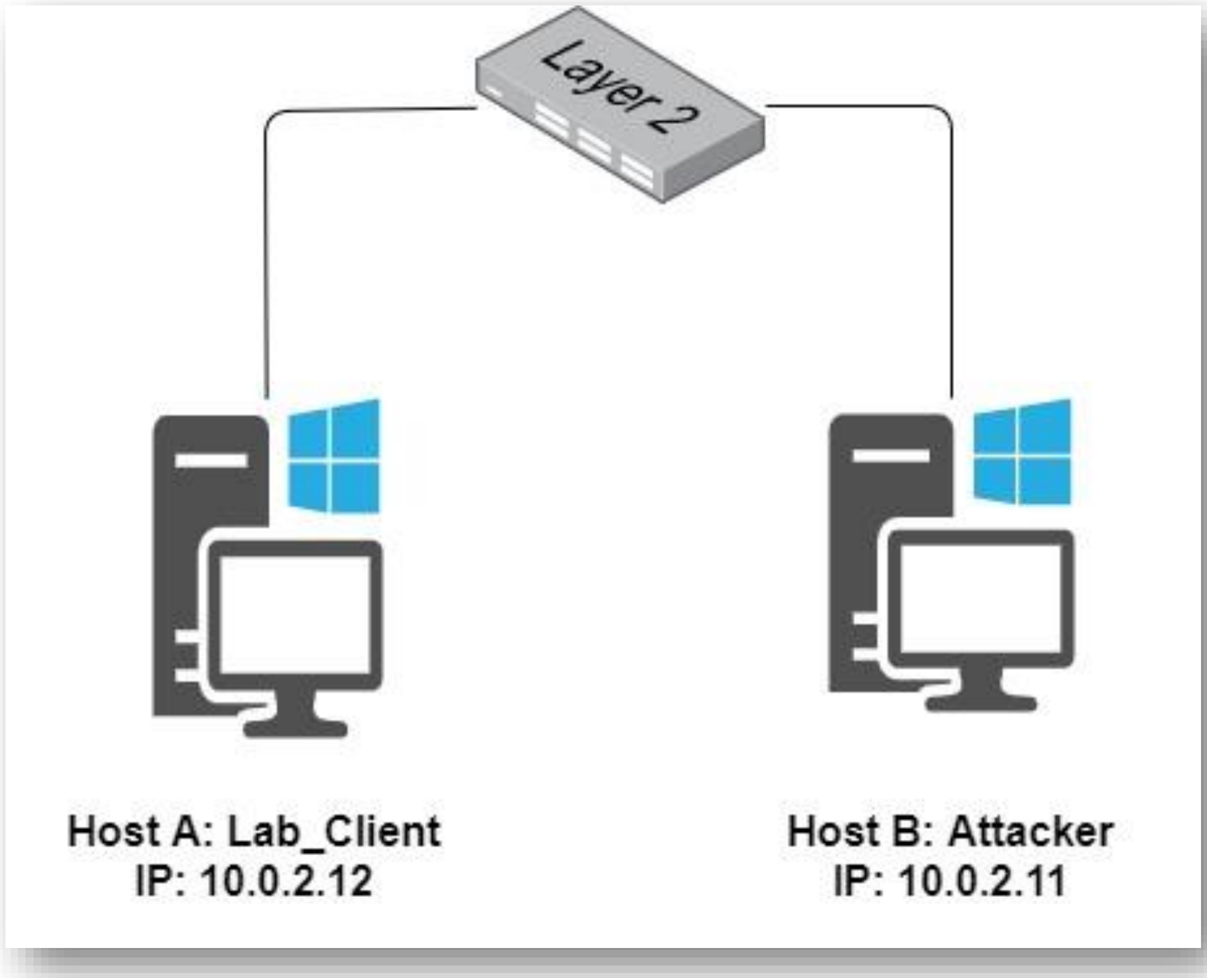
We will understand how firewalls work. We will learn how to use SSH tunnels to bypass firewalls.

I will perform and use the following:

- Set Firewall (iptables) Rules
- Netfilter
- Loadable kernel module
- Bypassing firewalls using SSH tunnel

Task 1: Using Firewall

Network physical topology:



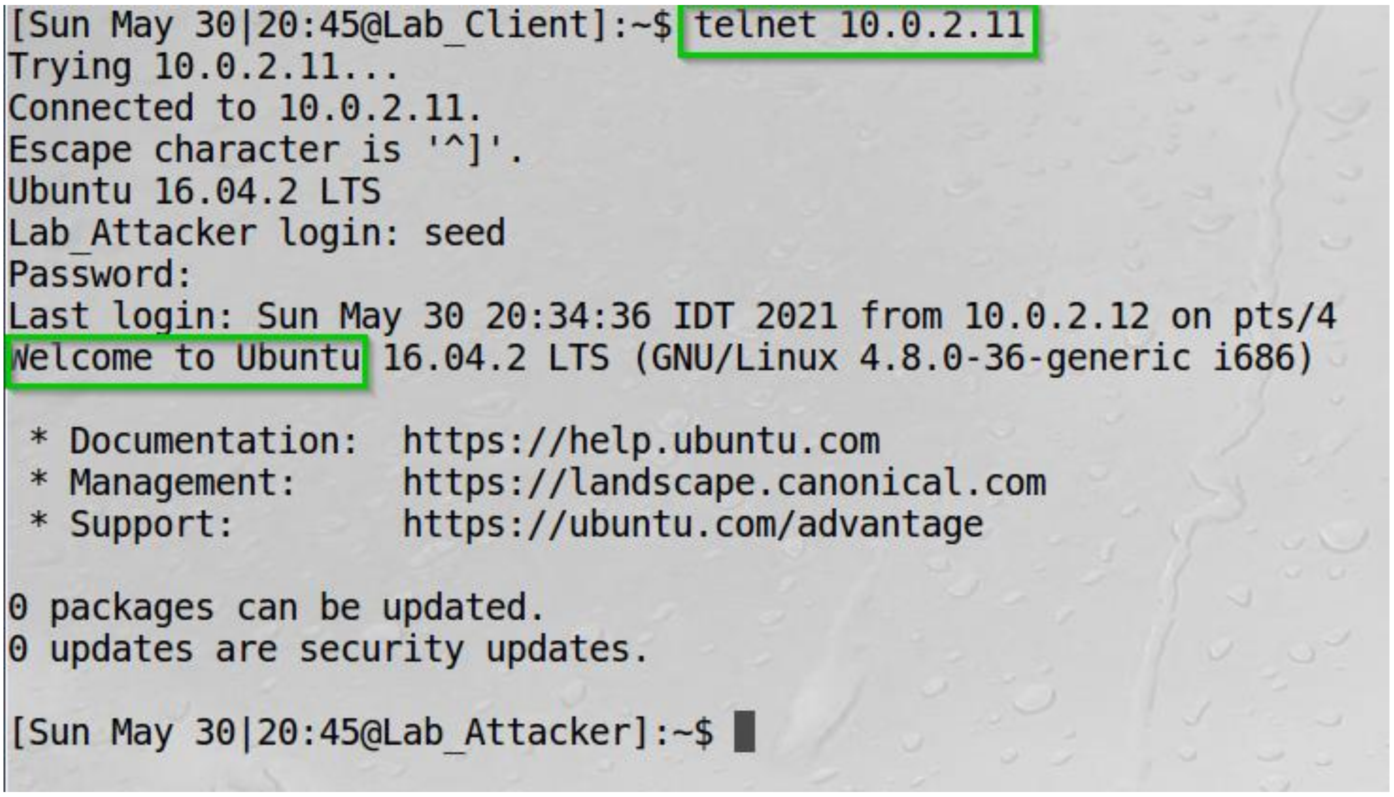
Task Description:

I will use iptables to prevent:

a. Host A (10.0.2.12) open telnet session to B (10.0.2.11)
b. Host A (10.0.2.11) open telnet session to B (10.0.2.12)
c. Host A (10.0.2.12) from surfing to a certain website

First, I will show successfully initiated telnet session between both.

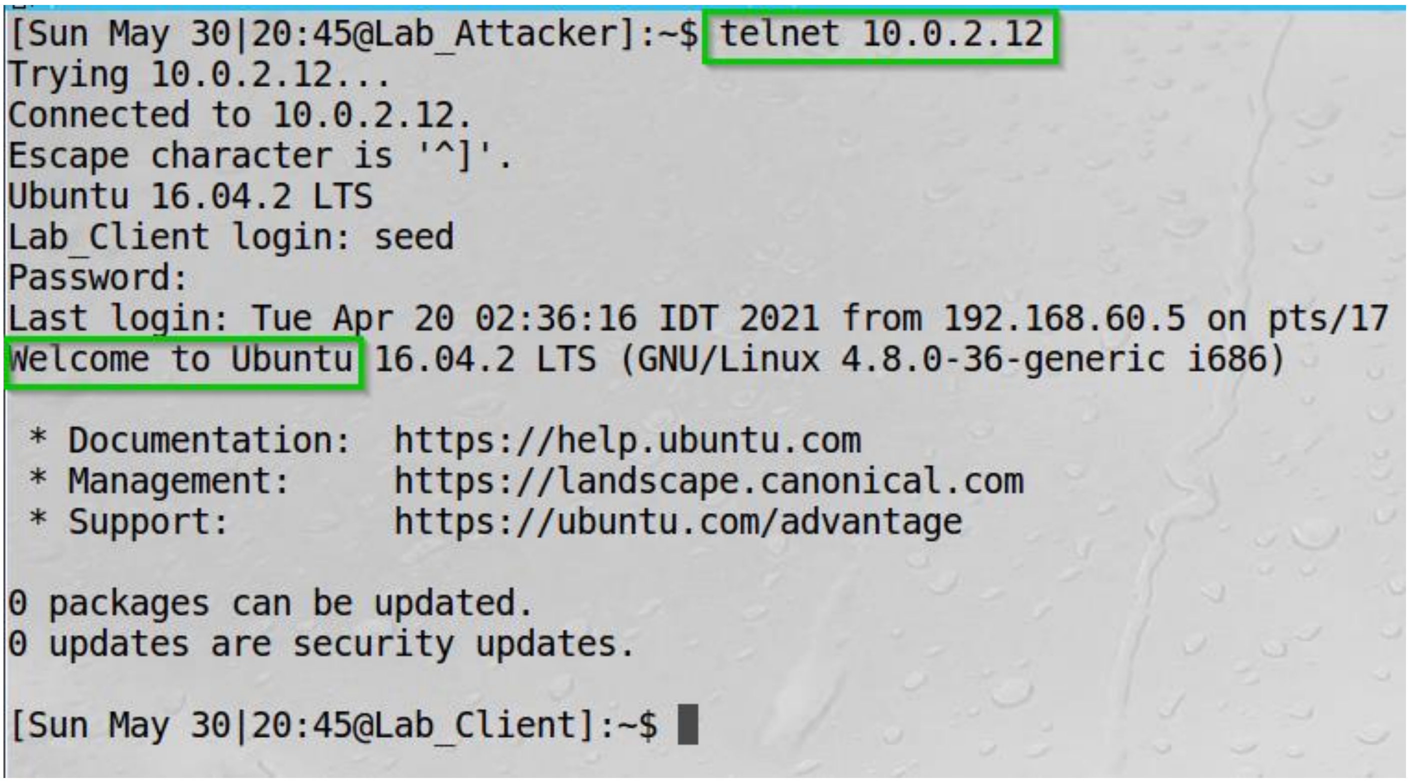
Telnet to 10.0.2.11



```
[Sun May 30|20:45@Lab_Client]:~$ telnet 10.0.2.11
Trying 10.0.2.11...
Connected to 10.0.2.11.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
Lab_Attacker login: seed
Password:
Last login: Sun May 30 20:34:36 IDT 2021 from 10.0.2.12 on pts/4
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

0 packages can be updated.
0 updates are security updates.

[Sun May 30|20:45@Lab_Attacker]:~$
```

We can see successful connection.

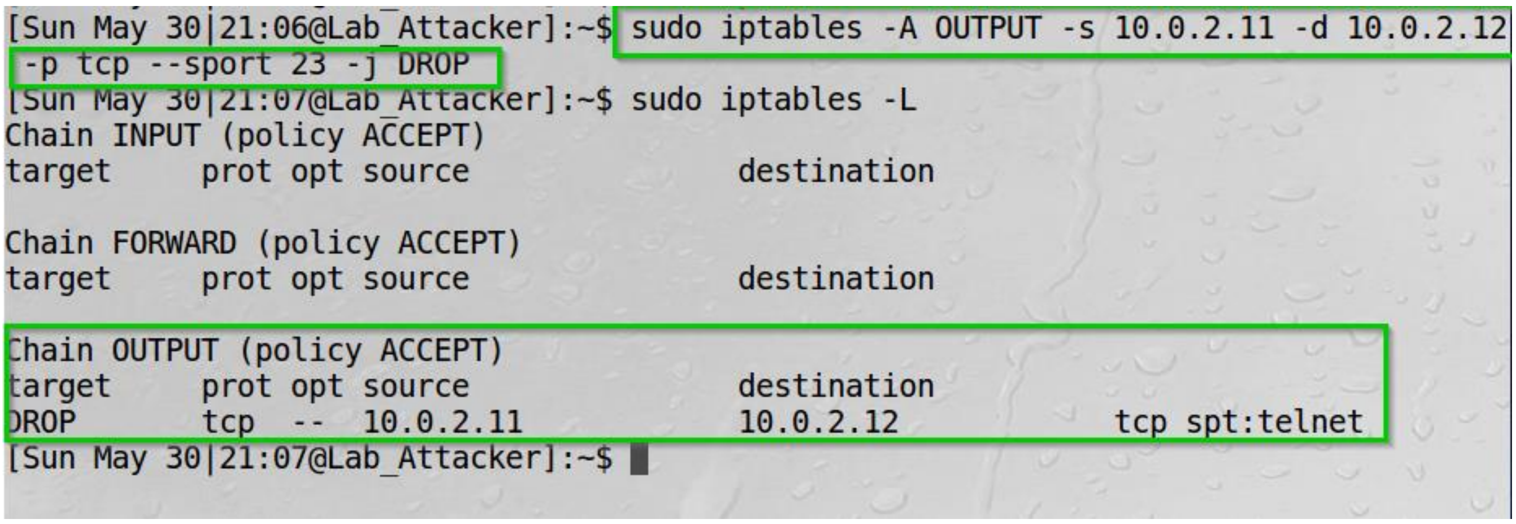
Telnet to 10.0.2.12



```
[Sun May 30|20:45@Lab_Attacker]:~$ telnet 10.0.2.12
Trying 10.0.2.12...
Connected to 10.0.2.12.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
Lab_Client login: seed
Password:
Last login: Tue Apr 20 02:36:16 IDT 2021 from 192.168.60.5 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

0 packages can be updated.
0 updates are security updates.

[Sun May 30|20:45@Lab_Client]:~$
```

We can see successful connection.

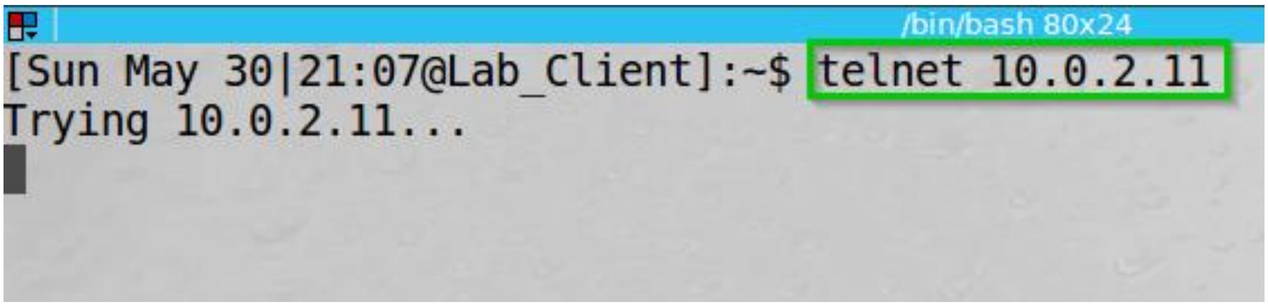
The following rule block telnet from A to B (executed in B)



```
[Sun May 30|21:06@Lab_Attacker]:~$ sudo iptables -A OUTPUT -s 10.0.2.11 -d 10.0.2.12
 -p tcp --sport 23 -j DROP
[Sun May 30|21:07@Lab_Attacker]:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
DROP       tcp  --  10.0.2.11            10.0.2.12            tcp spt:telnet
[Sun May 30|21:07@Lab_Attacker]:~$
```

We can see the rule in the iptables.

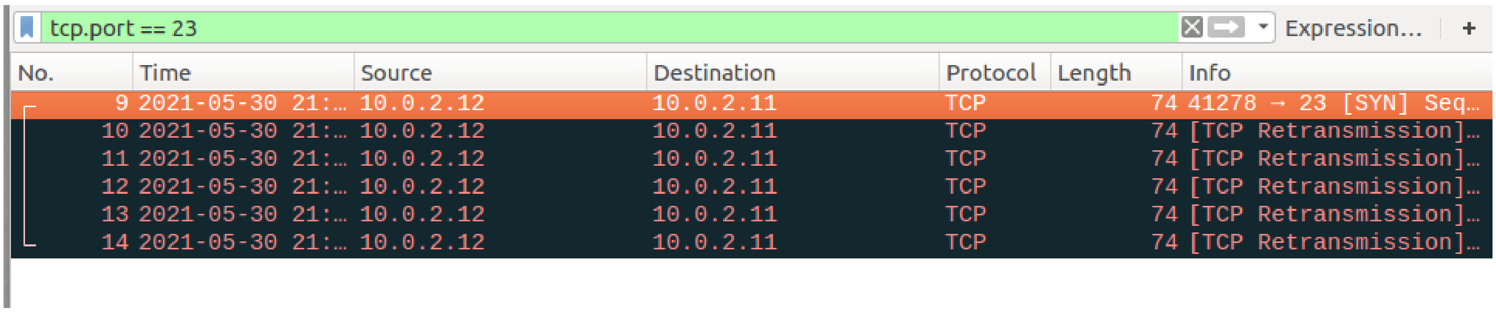
Try initiating telnet from A to B.



```
[Sun May 30|21:07@Lab_Client]:~$ telnet 10.0.2.11
Trying 10.0.2.11...
```

We can see the connection could not be set.

Wireshark view on 10.0.2.11.

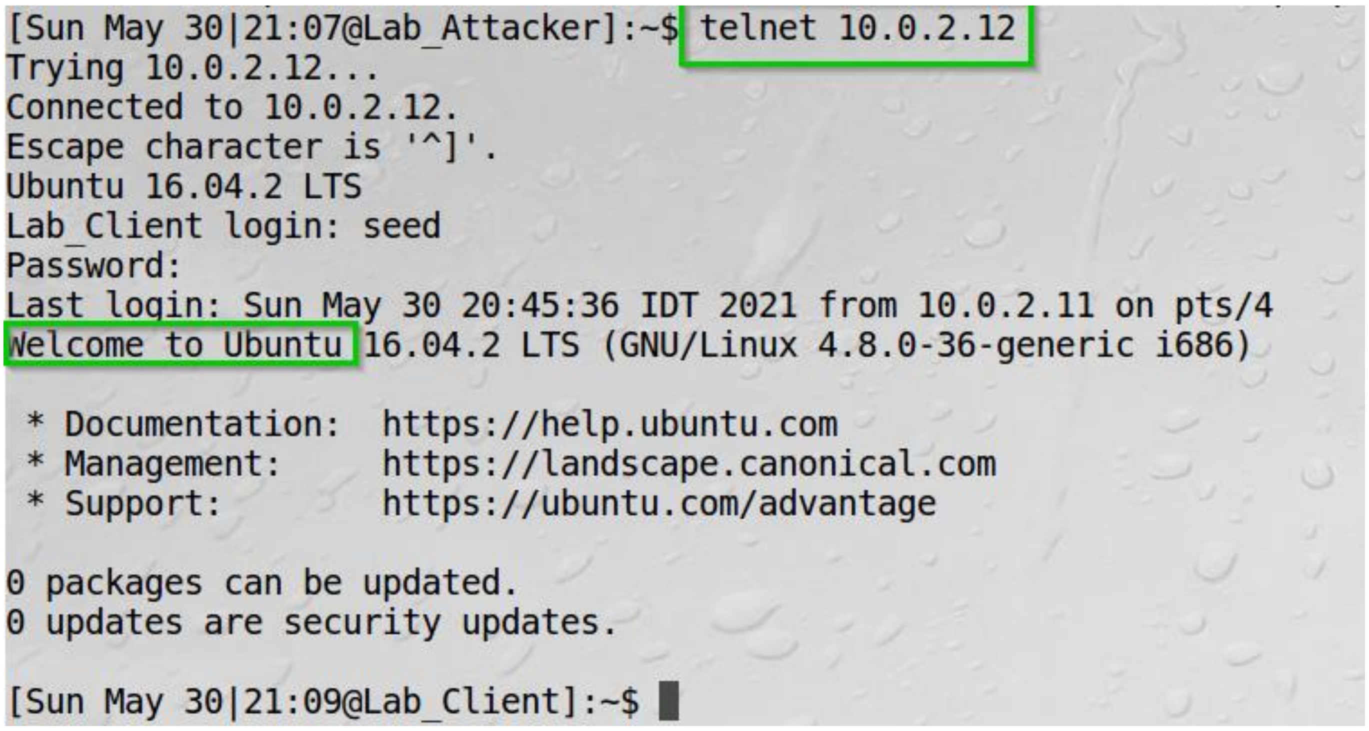


```
tcp.port == 23
```

No.	Time	Source	Destination	Protocol	Length	Info
9	2021-05-30 21:…	10.0.2.12	10.0.2.11	TCP	74	41278 → 23 [SYN] Seq…
10	2021-05-30 21:…	10.0.2.12	10.0.2.11	TCP	74	[TCP Retransmission]…
11	2021-05-30 21:…	10.0.2.12	10.0.2.11	TCP	74	[TCP Retransmission]…
12	2021-05-30 21:…	10.0.2.12	10.0.2.11	TCP	74	[TCP Retransmission]…
13	2021-05-30 21:…	10.0.2.12	10.0.2.11	TCP	74	[TCP Retransmission]…
14	2021-05-30 21:…	10.0.2.12	10.0.2.11	TCP	74	[TCP Retransmission]…

We can see 10.0.2.12 tries to connect to 10.0.2.11 but 10.0.2.11 does not reply since the packets are filtered.

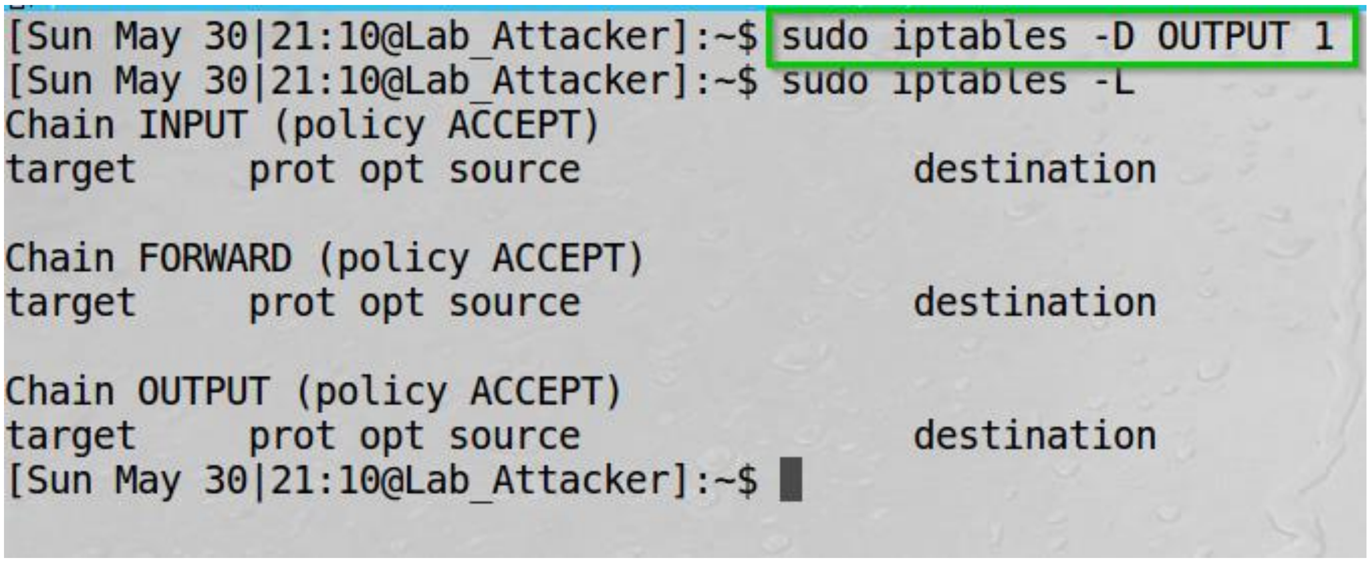
I now show that B (10.0.2.11) can still initiate telnet to A (10.0.2.12).



```
[Sun May 30|21:07@Lab_Attacker]:~$ telnet 10.0.2.12
Trying 10.0.2.12...
Connected to 10.0.2.12.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
Lab_Client login: seed
Password:
Last login: Sun May 30 20:45:36 IDT 2021 from 10.0.2.11 on pts/4
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

0 packages can be updated.
0 updates are security updates.

[Sun May 30|21:09@Lab_Client]:~$
```

We can see successful connection.

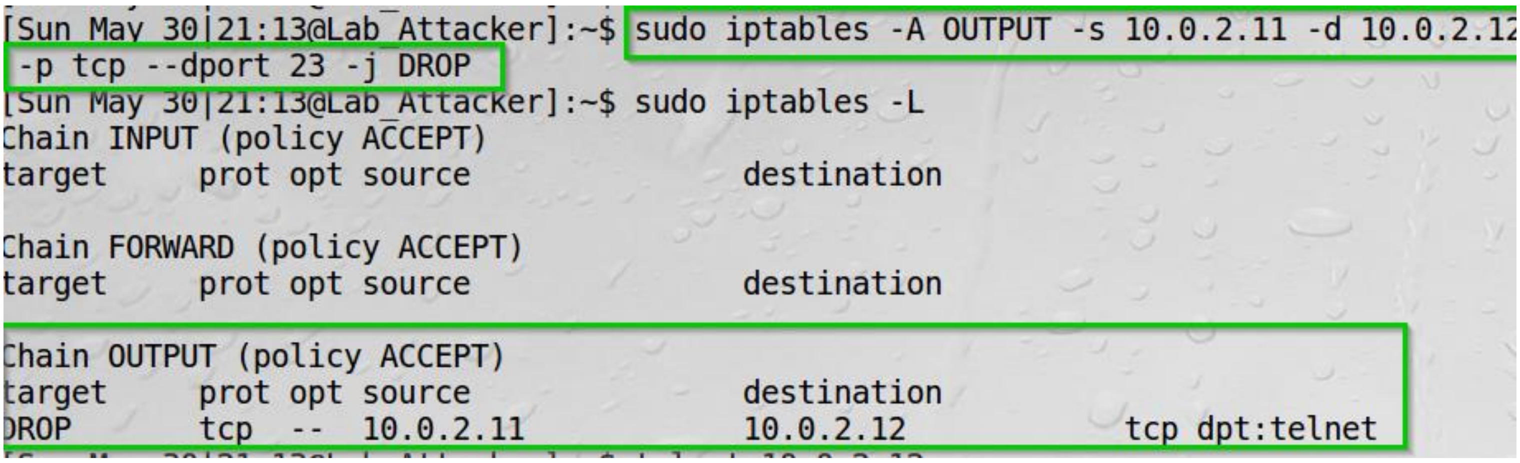
Deleting the iptables rule.



```
[Sun May 30|21:10@Lab_Attacker]:~$ sudo iptables -D OUTPUT 1
[Sun May 30|21:10@Lab_Attacker]:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
[Sun May 30|21:10@Lab_Attacker]:~$
```

We can see the rule is deleted from the iptables.

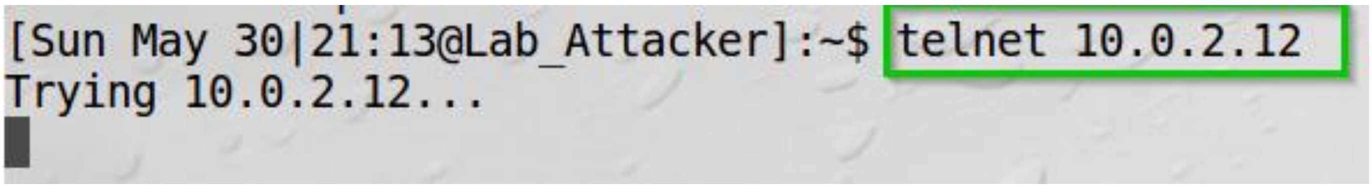
The following rule block telnet from B to A (executed in B).



```
[Sun May 30|21:13@Lab_Attacker]:~$ sudo iptables -A OUTPUT -s 10.0.2.11 -d 10.0.2.12
-p tcp --dport 23 -j DROP
[Sun May 30|21:13@Lab_Attacker]:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
DROP       tcp  --  10.0.2.11            10.0.2.12            tcp dpt:telnet
```

We can see the rule in the iptables.

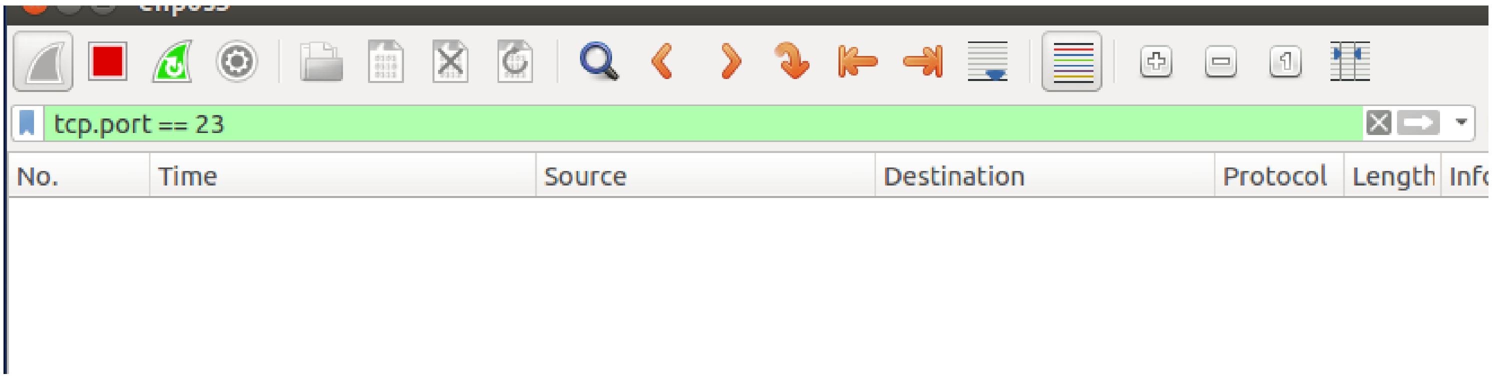
We try initiating telnet from B to A.



```
[Sun May 30|21:13@Lab_Attacker]:~$ telnet 10.0.2.12
Trying 10.0.2.12...
```

We can see the connection could not be set.

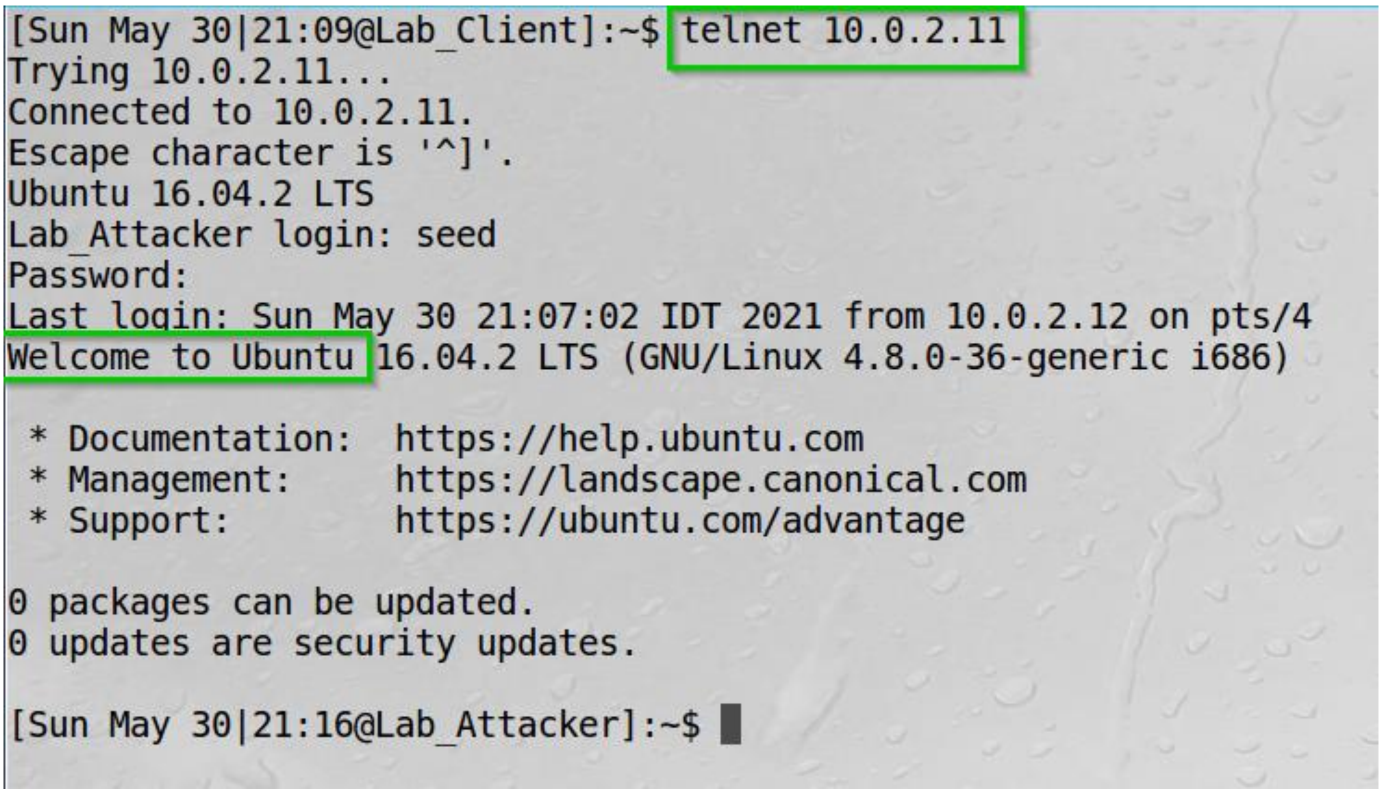
Wireshark view on 10.0.2.11.



```
tcp.port == 23
No.    Time    Source    Destination    Protocol    Length    Info
```

We can see there are no packets being sent on port 23.

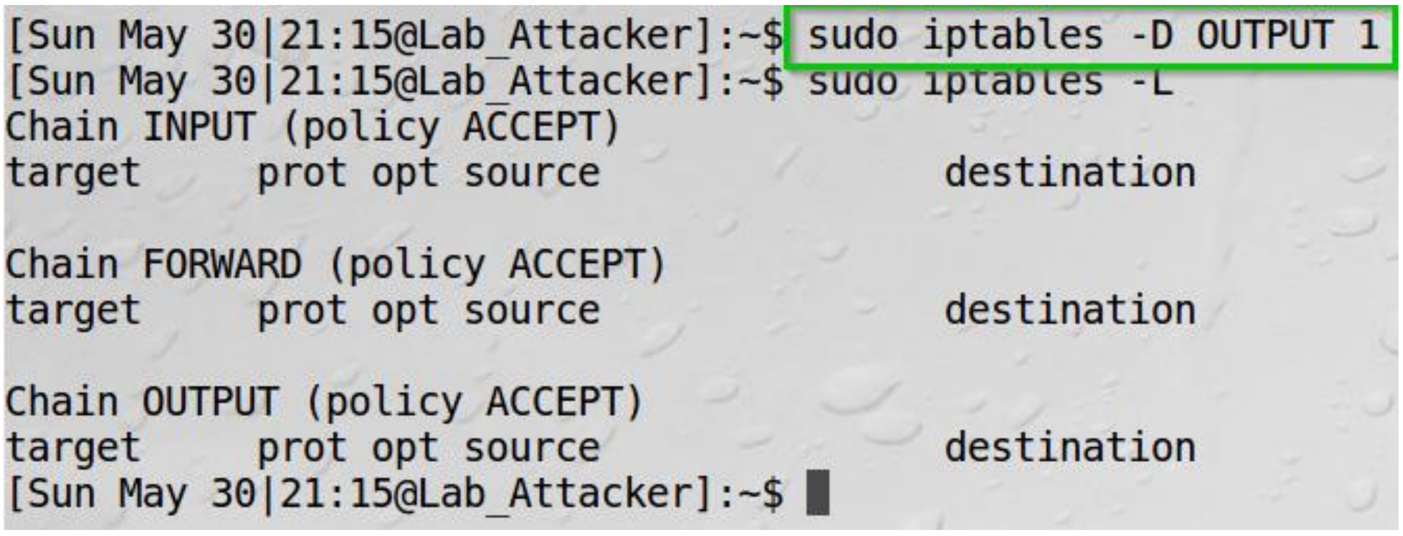
Now I show that A (10.0.2.12) can still initiate telnet to B (10.0.2.11).



```
[Sun May 30|21:09@Lab_Client]:~$ telnet 10.0.2.11
Trying 10.0.2.11...
Connected to 10.0.2.11.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
Lab_Attacker login: seed
Password:
Last login: Sun May 30 21:07:02 IDT 2021 from 10.0.2.12 on pts/4
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

0 packages can be updated.
0 updates are security updates.

[Sun May 30|21:16@Lab_Attacker]:~$ 
```

We can see successful connection.

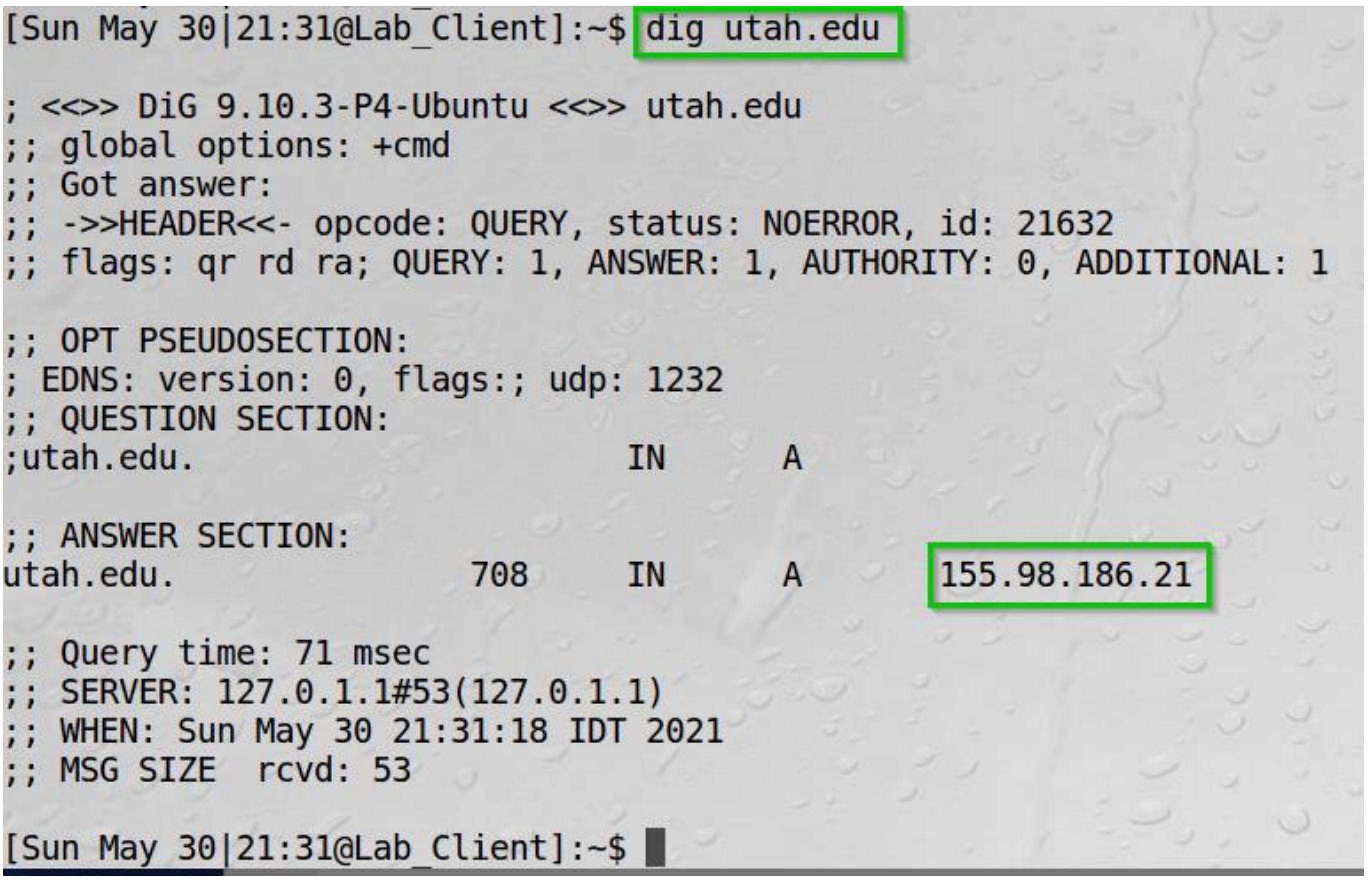
Deleting the iptables rule.



```
[Sun May 30|21:15@Lab_Attacker]:~$ sudo iptables -D OUTPUT 1
[Sun May 30|21:15@Lab_Attacker]:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
[Sun May 30|21:15@Lab_Attacker]:~$ 
```

We can see the rule is deleted from the iptables.

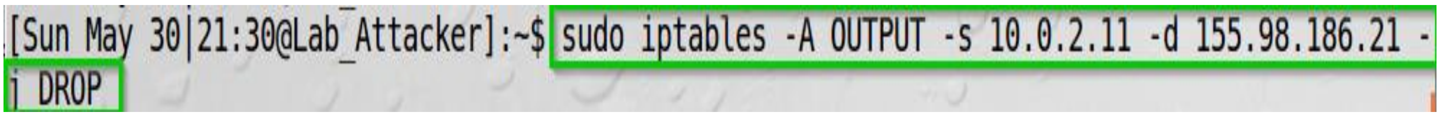
Using dig command, I got Utah University website IP.



```
[Sun May 30|21:31@Lab_Client]:~$ dig utah.edu

; <<>> DiG 9.10.3-P4-Ubuntu <<>> utah.edu
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 21632
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
;; QUESTION SECTION:
;utah.edu.                      IN      A

;; ANSWER SECTION:
utah.edu.               708     IN      A       155.98.186.21

;; Query time: 71 msec
;; SERVER: 127.0.1.1#53(127.0.1.1)
;; WHEN: Sun May 30 21:31:18 IDT 2021
;; MSG SIZE  rcvd: 53

[Sun May 30|21:31@Lab_Client]:~$
```

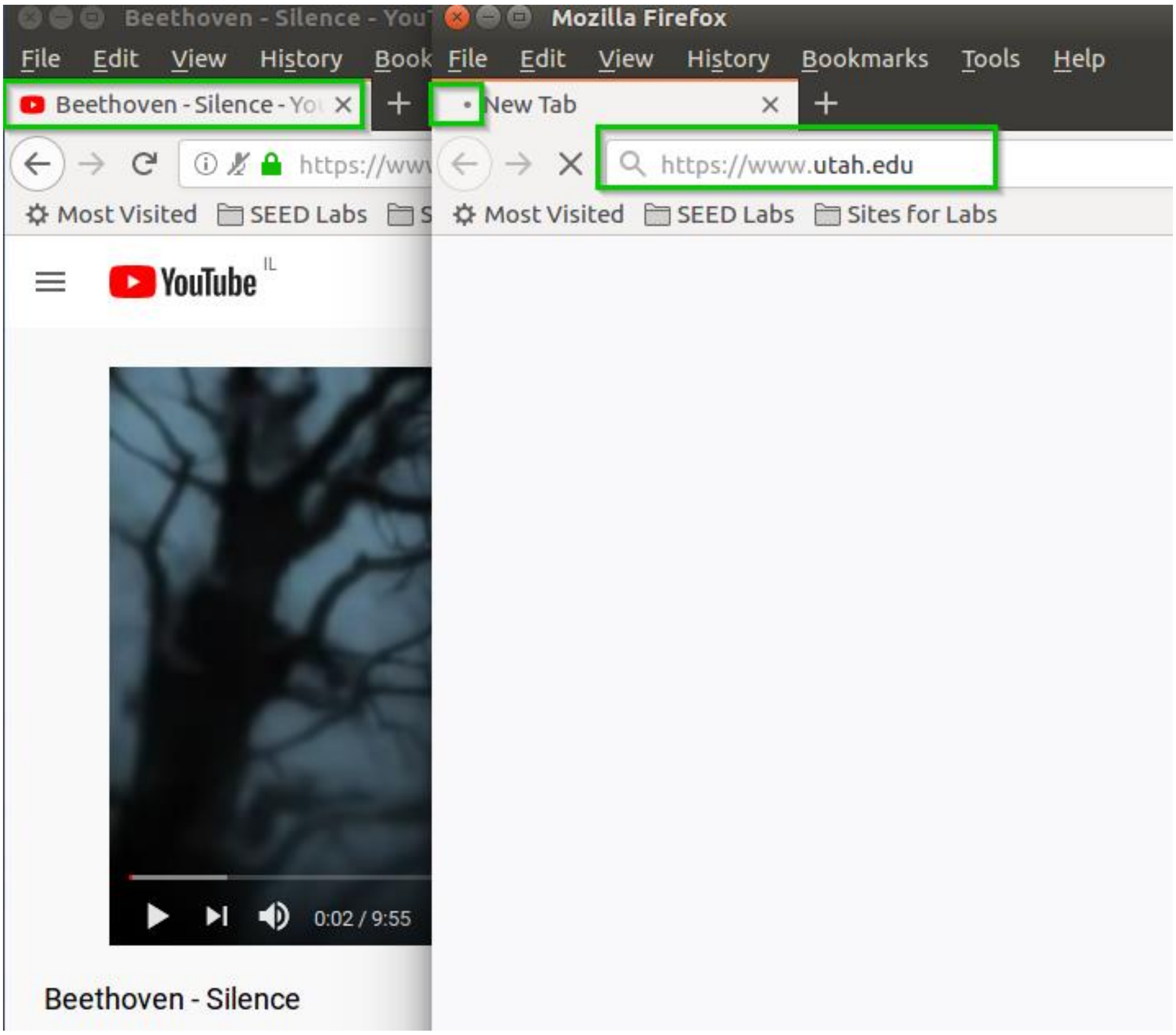
We can see the result 155.98.186.21.

The following rule block telnet from A (10.0.2.12) to Utah University website.



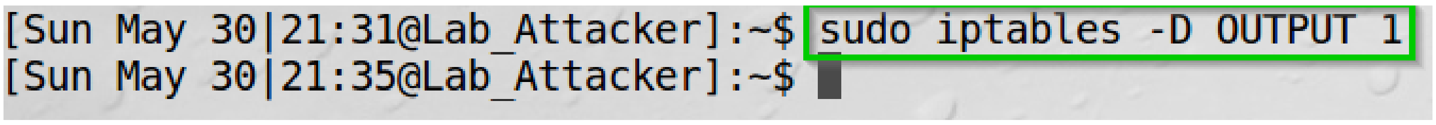
```
[Sun May 30|21:30@Lab_Attacker]:~$ sudo iptables -A OUTPUT -s 10.0.2.11 -d 155.98.186.21 -
j DROP
```

I now tried browsing to Utah University website.



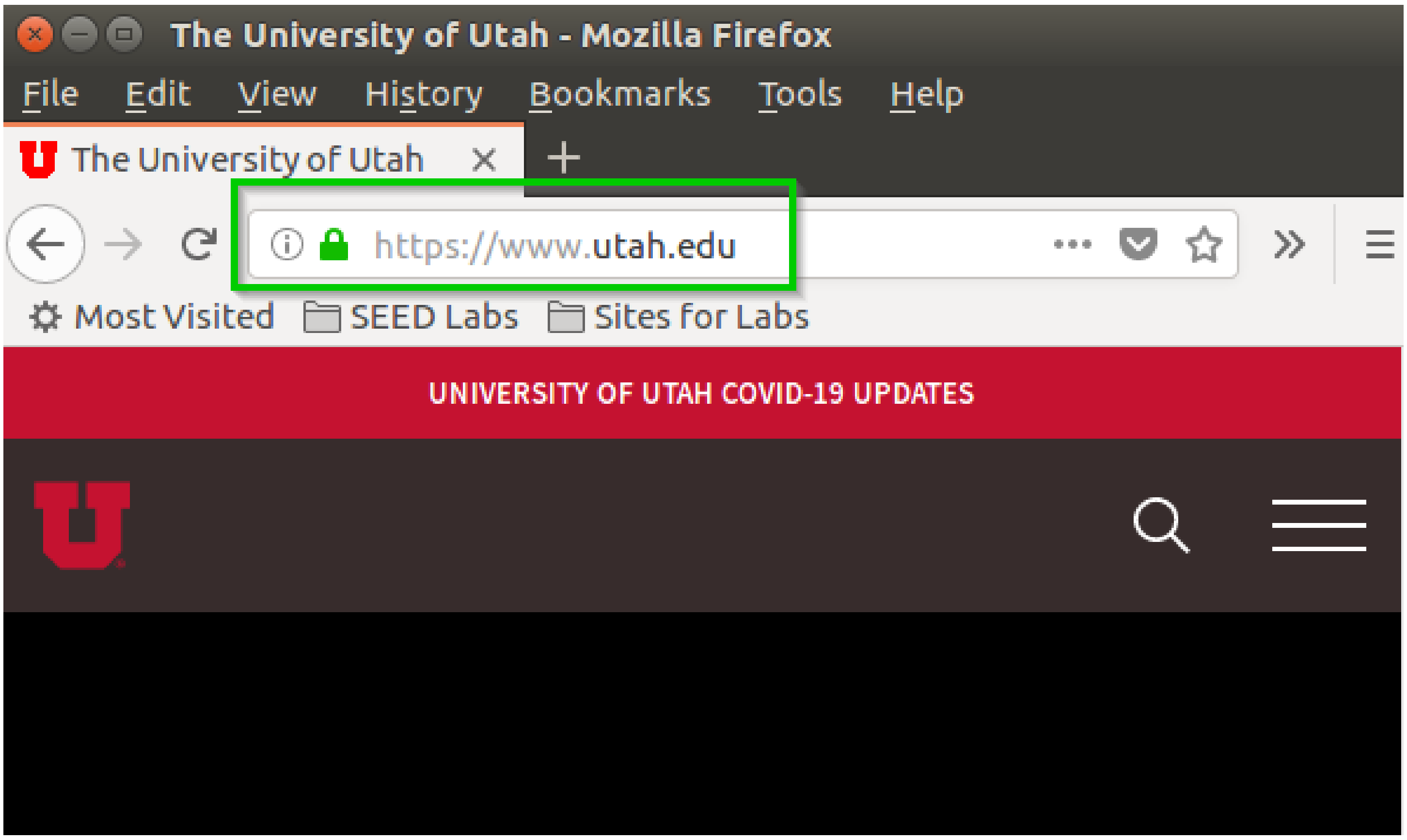
We can see unsuccessful attempt.
YouTube is to show network connectivity.

Deleting the iptables rule.



```
[Sun May 30|21:31@Lab_Attacker]:~$ sudo iptables -D OUTPUT 1
[Sun May 30|21:35@Lab_Attacker]:~$ 
```

We can browse now to Utah University website.


The University of Utah - Mozilla Firefox
File Edit View History Bookmarks Tools Help
The University of Utah
https://www.utah.edu
Most Visited
SEED Labs
Sites for Labs
UNIVERSITY OF UTAH COVID-19 UPDATES

We can see a successful connection.

Task 1 Summary

- I have learned using iptables as our Firewall.
- I opened Telnet and SSH connection between A and B. Then applied iptables rule to block the communication.
- I also blocked connection to Utah University website using iptables.
- I learned about iptables structure and rules.

Task 2: Implementing a Simple Firewall

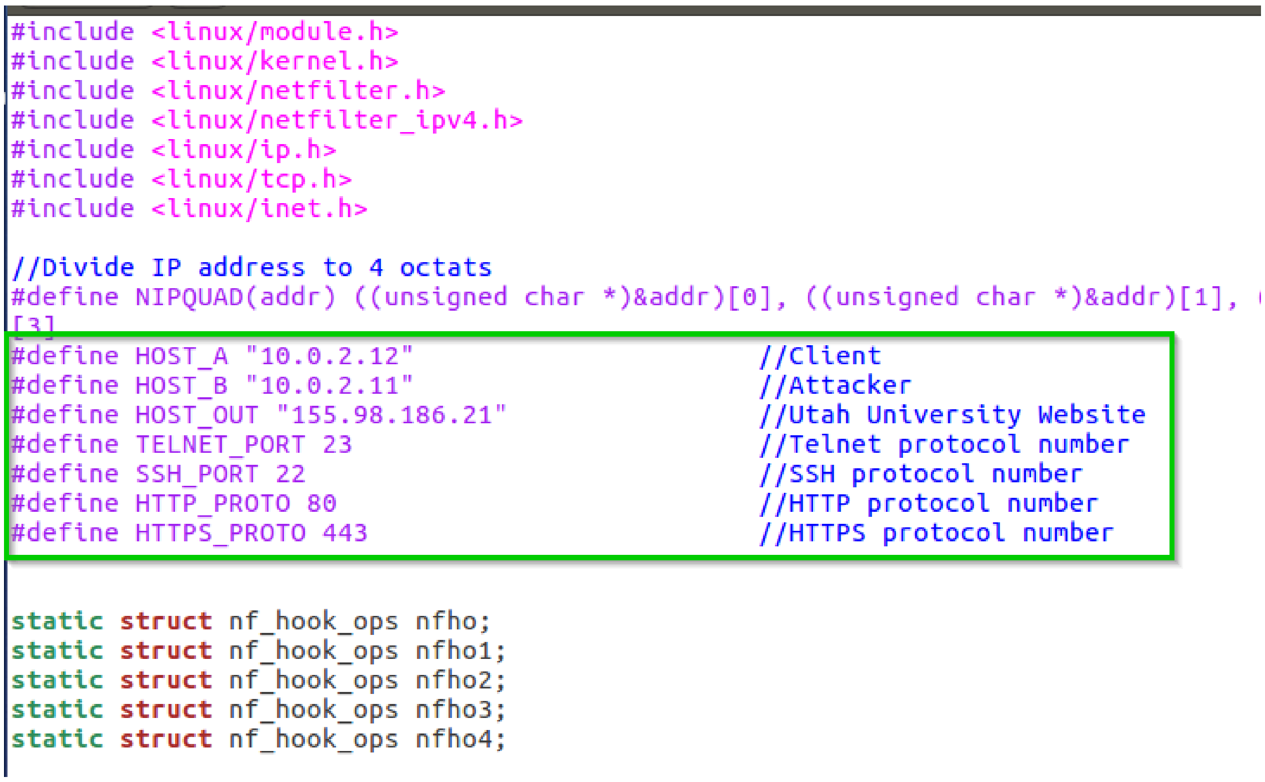
Task Description:

In this task I will use LKM and Netfilter to implement packet filtering.
We will filter:

- Outgoing SSH from 10.0.2.12 to 10.0.2.11
- Outgoing Telnet from 10.0.2.12 to 10.0.2.11
- Incoming SSH from 10.0.2.11 to 10.0.2.12
- Incoming Telnet from 10.0.2.11 to 10.0.2.12
- Outgoing Web communication with Utah University website.

Loadable Kernel Module are module we can inject directly to the OS kernel using Netfilter.
Netfilter implements ‘hooks’ on the operating system – thus, capturing system calls before they move from the kernel to the operating system. Therefore giving us the first control over the packets filtered.

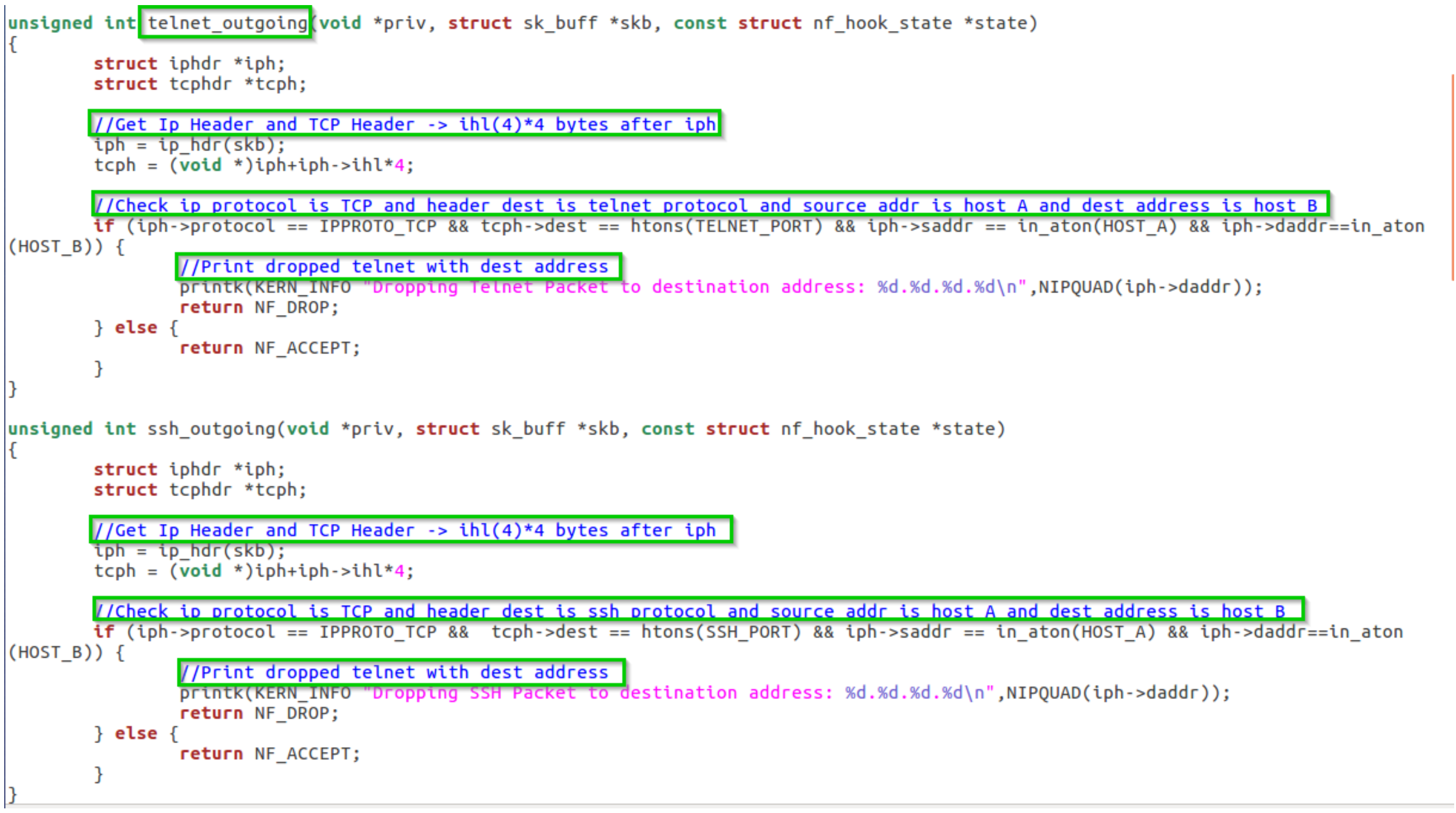
To achieve my goal, I wrote the following code:



```
#include <linux/module.h>
#include <linux/kernel.h>
#include <linux/netfilter.h>
#include <linux/netfilter_ipv4.h>
#include <linux/ip.h>
#include <linux/tcp.h>
#include <linux/inet.h>

//Divide IP address to 4 octats
#define NIPQUAD(addr) ((unsigned char *)&addr)[0], ((unsigned char *)&addr)[1],
[3]
#define HOST_A "10.0.2.12"                              //Client
#define HOST_B "10.0.2.11"                              //Attacker
#define HOST_OUT "155.98.186.21"                        //Utah University Website
#define TELNET_PORT 23                                  //Telnet protocol number
#define SSH_PORT 22                                     //SSH protocol number
#define HTTP_PROTO 80                                   //HTTP protocol number
#define HTTPS_PROTO 443                                 //HTTPS protocol number


static struct nf_hook_ops nfho;
static struct nf_hook_ops nfho1;
static struct nf_hook_ops nfho2;
static struct nf_hook_ops nfho3;
static struct nf_hook_ops nfho4;
```

Initiate variables for our IP’s, ports.
Note: the divide IP address define is cut in the screenshot for convenience - it does the same for ((unsinged char *)&addr)[2] and 3.

The following are Telnet outgoing and SSH outgoing filtering.

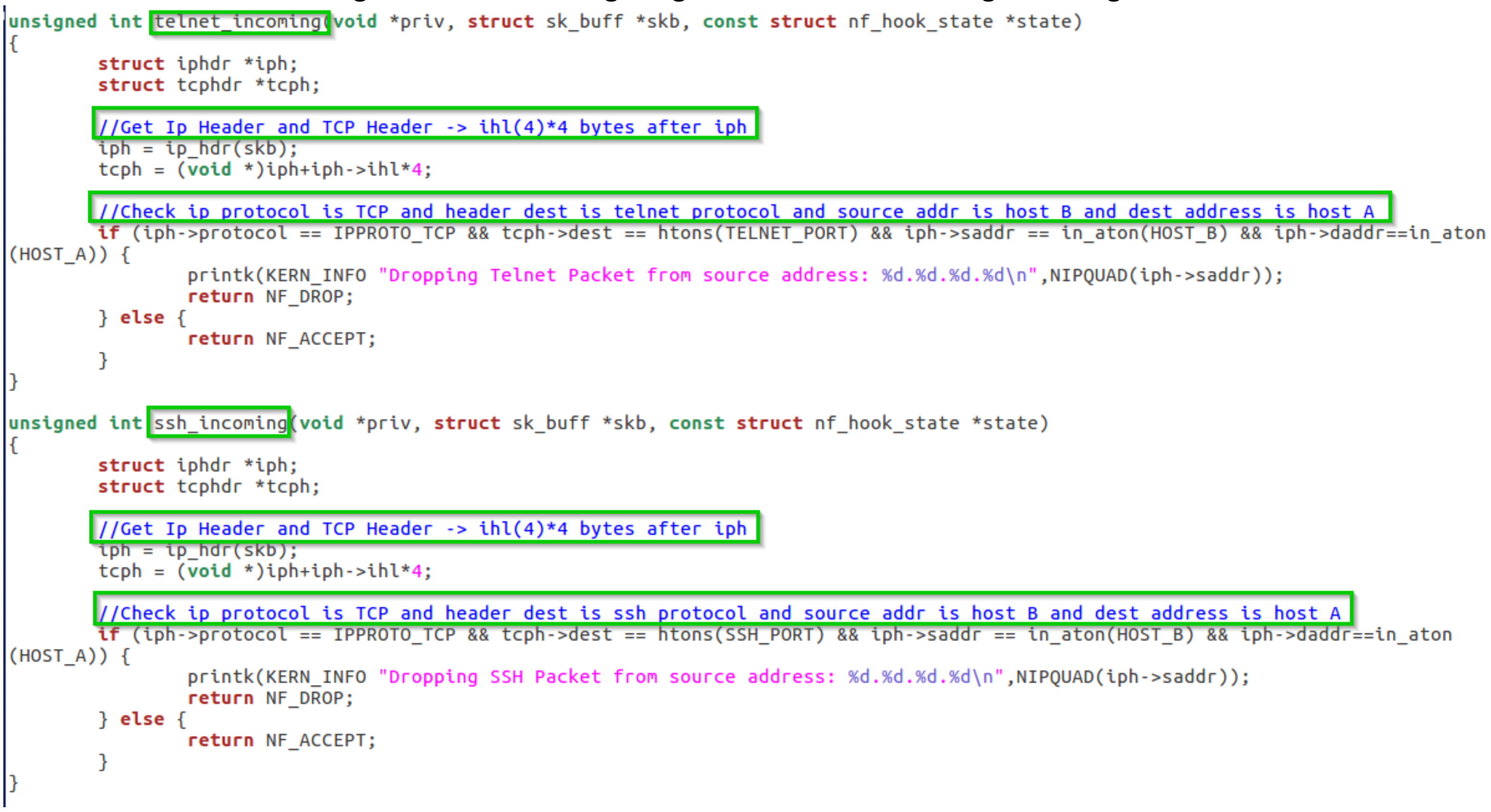


```
unsigned int telnet_outgoing(void *priv, struct sk_buff *skb, const struct nf_hook_state *state)
{
        struct iphdr *iph;
        struct tcphdr *tcph;

        //Get Ip Header and TCP Header -> ihl(4)*4 bytes after iph
        iph = ip_hdr(skb);
        tcph = (void *)iph+iph->ihl*4;

        //Check ip protocol is TCP and header dest is telnet protocol and source addr is host A and dest address is host B
        if (iph->protocol == IPPROTO_TCP && tcph->dest == htons(TELNET_PORT) && iph->saddr == in_aton(HOST_A) && iph->daddr==in_aton
(HOST_B)) {
                //Print dropped telnet with dest address
                printk(KERN_INFO "Dropping Telnet Packet to destination address: %d.%d.%d.%d\n",NIPQUAD(iph->daddr));
                return NF_DROP;
        } else {
                return NF_ACCEPT;
        }
}

unsigned int ssh_outgoing(void *priv, struct sk_buff *skb, const struct nf_hook_state *state)
{
        struct iphdr *iph;
        struct tcphdr *tcph;

        //Get Ip Header and TCP Header -> ihl(4)*4 bytes after iph
        iph = ip_hdr(skb);
        tcph = (void *)iph+iph->ihl*4;

        //Check ip protocol is TCP and header dest is ssh protocol and source addr is host A and dest address is host B
        if (iph->protocol == IPPROTO_TCP &&  tcph->dest == htons(SSH_PORT) && iph->saddr == in_aton(HOST_A) && iph->daddr==in_aton
(HOST_B)) {
                //Print dropped telnet with dest address
                printk(KERN_INFO "Dropping SSH Packet to destination address: %d.%d.%d.%d\n",NIPQUAD(iph->daddr));
                return NF_DROP;
        } else {
                return NF_ACCEPT;
        }
}
```

We can see we initiate the ip header and tcp header from the buffer skb.

We check if the protocol is tcp, the source is A and destination B, and the port is for ssh or telnet respectively for the functions.

The following are Telnet outgoing and SSH Incoming filtering.

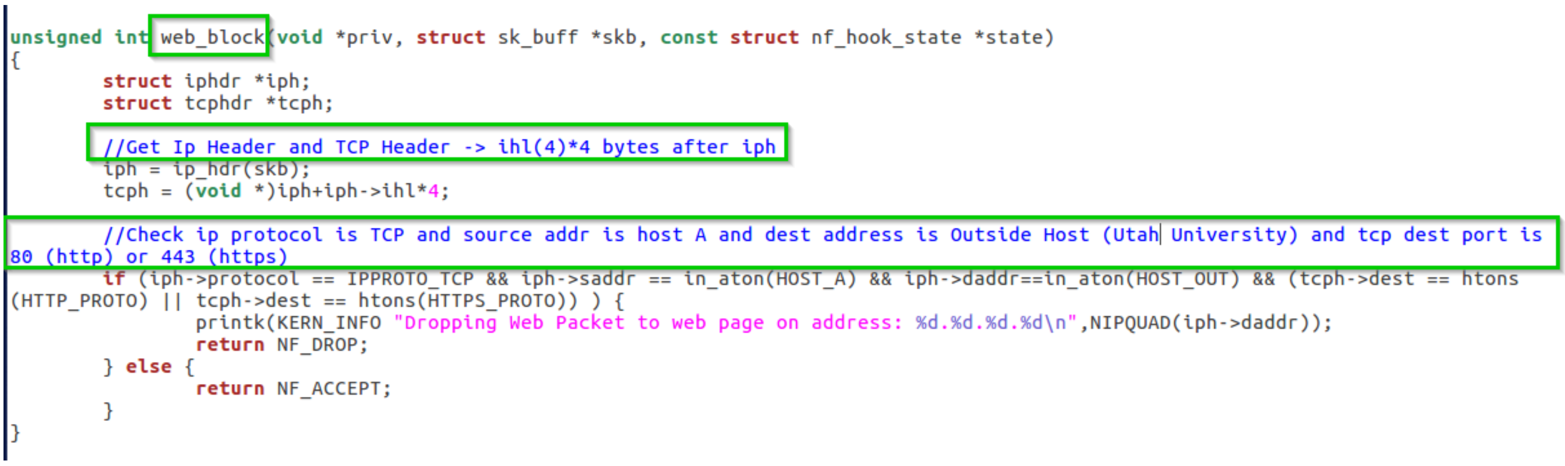


```
unsigned int telnet_incoming(void *priv, struct sk_buff *skb, const struct nf_hook_state *state)
{
        struct iphdr *iph;
        struct tcphdr *tcph;

        //Get Ip Header and TCP Header -> ihl(4)*4 bytes after iph
        iph = ip_hdr(skb);
        tcph = (void *)iph+iph->ihl*4;

        //Check ip protocol is TCP and header dest is telnet protocol and source addr is host B and dest address is host A
        if (iph->protocol == IPPROTO_TCP && tcph->dest == htons(TELNET_PORT) && iph->saddr == in_aton(HOST_B) && iph->daddr==in_aton
(HOST_A)) {
                printk(KERN_INFO "Dropping Telnet Packet from source address: %d.%d.%d.%d\n",NIPQUAD(iph->saddr));
                return NF_DROP;
        } else {
                return NF_ACCEPT;
        }
}

unsigned int ssh_incoming(void *priv, struct sk_buff *skb, const struct nf_hook_state *state)
{
        struct iphdr *iph;
        struct tcphdr *tcph;

        //Get Ip Header and TCP Header -> ihl(4)*4 bytes after iph
        iph = ip_hdr(skb);
        tcph = (void *)iph+iph->ihl*4;

        //Check ip protocol is TCP and header dest is ssh protocol and source addr is host B and dest address is host A
        if (iph->protocol == IPPROTO_TCP && tcph->dest == htons(SSH_PORT) && iph->saddr == in_aton(HOST_B) && iph->daddr==in_aton
(HOST_A)) {
                printk(KERN_INFO "Dropping SSH Packet from source address: %d.%d.%d.%d\n",NIPQUAD(iph->saddr));
                return NF_DROP;
        } else {
                return NF_ACCEPT;
        }
}
```

We can see we initiate the ip header and tcp header from the buffer skb.

We check if the protocol is tcp, the source is B and destination A, and the port is for ssh or telnet respectively for the functions.

The following is website filtering.

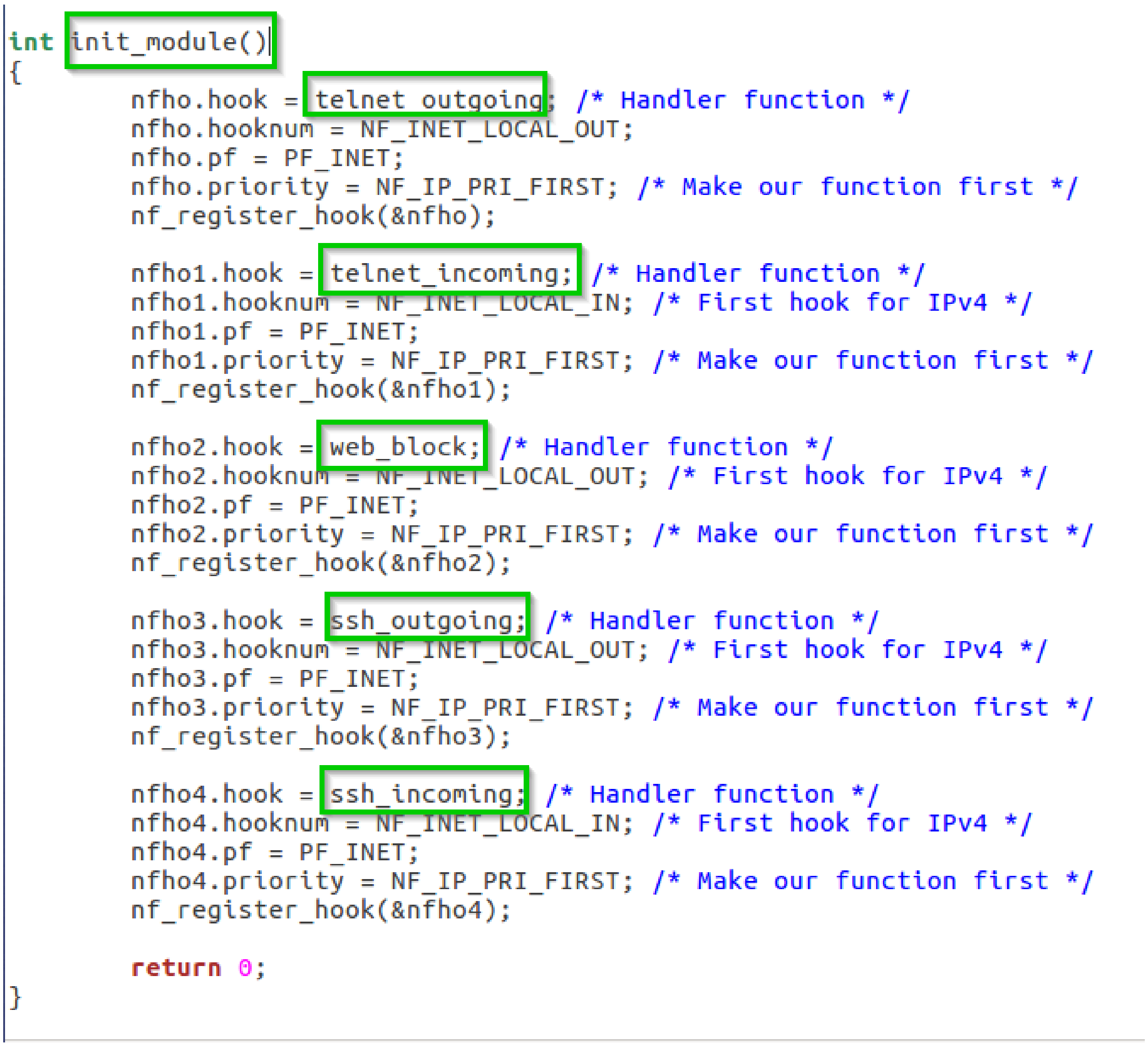


```
unsigned int web_block(void *priv, struct sk_buff *skb, const struct nf_hook_state *state)
{
        struct iphdr *iph;
        struct tcphdr *tcph;

        //Get Ip Header and TCP Header -> ihl(4)*4 bytes after iph
        iph = ip_hdr(skb);
        tcph = (void *)iph+iph->ihl*4;

        //Check ip protocol is TCP and source addr is host A and dest address is Outside Host (Utah University) and tcp dest port is
80 (http) or 443 (https)
        if (iph->protocol == IPPROTO_TCP && iph->saddr == in_aton(HOST_A) && iph->daddr==in_aton(HOST_OUT) && (tcph->dest == htons
(HTTP_PROTO) || tcph->dest == htons(HTTPS_PROTO)) ) {
                printk(KERN_INFO "Dropping Web Packet to web page on address: %d.%d.%d.%d\n",NIPQUAD(iph->daddr));
                return NF_DROP;
        } else {
                return NF_ACCEPT;
        }
}
```

We can see we initiate the ip header and tcp header from the buffer skb.

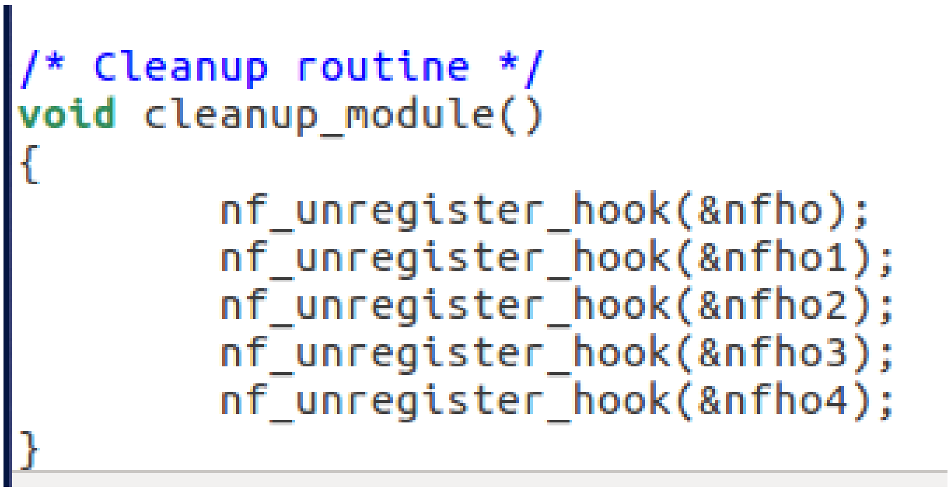
We check if the protocol is tcp, the ports are 443 or 80 and the source is A and destination is the Utah University website IP.

Initiate of the module and our hooks.



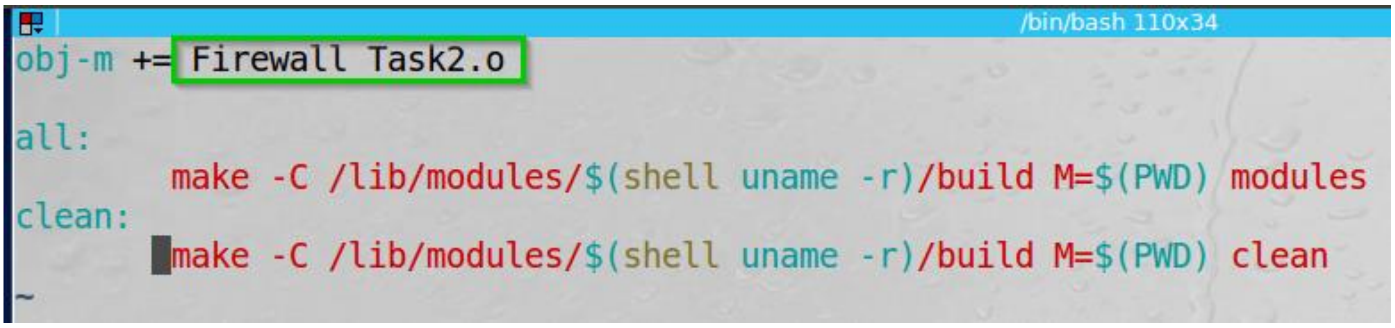
```
int init_module()
{
        nfho.hook = telnet_outgoing; /* Handler function */
        nfho.hooknum = NF_INET_LOCAL_OUT;
        nfho.pf = PF_INET;
        nfho.priority = NF_IP_PRI_FIRST; /* Make our function first */
        nf_register_hook(&nfho);

        nfho1.hook = telnet_incoming; /* Handler function */
        nfho1.hooknum = NF_INET_LOCAL_IN; /* First hook for IPv4 */
        nfho1.pf = PF_INET;
        nfho1.priority = NF_IP_PRI_FIRST; /* Make our function first */
        nf_register_hook(&nfho1);

        nfho2.hook = web_block; /* Handler function */
        nfho2.hooknum = NF_INET_LOCAL_OUT; /* First hook for IPv4 */
        nfho2.pf = PF_INET;
        nfho2.priority = NF_IP_PRI_FIRST; /* Make our function first */
        nf_register_hook(&nfho2);

        nfho3.hook = ssh_outgoing; /* Handler function */
        nfho3.hooknum = NF_INET_LOCAL_OUT; /* First hook for IPv4 */
        nfho3.pf = PF_INET;
        nfho3.priority = NF_IP_PRI_FIRST; /* Make our function first */
        nf_register_hook(&nfho3);

        nfho4.hook = ssh_incoming; /* Handler function */
        nfho4.hooknum = NF_INET_LOCAL_IN; /* First hook for IPv4 */
        nfho4.pf = PF_INET;
        nfho4.priority = NF_IP_PRI_FIRST; /* Make our function first */
        nf_register_hook(&nfho4);

        return 0;
}
```

Cleanup module to unregister our hooks.



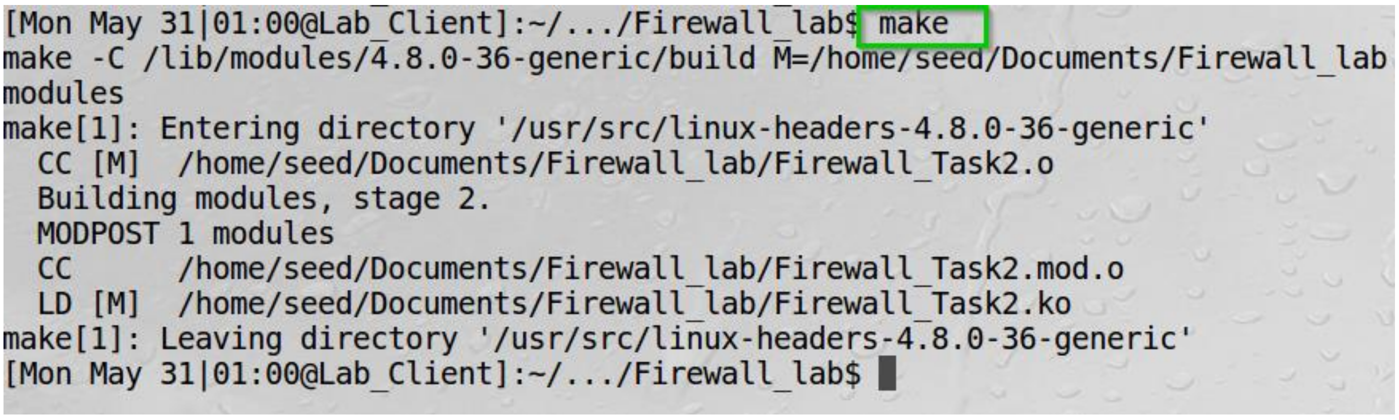
```
/* Cleanup routine */
void cleanup_module()
{
        nf_unregister_hook(&nfho);
        nf_unregister_hook(&nfho1);
        nf_unregister_hook(&nfho2);
        nf_unregister_hook(&nfho3);
        nf_unregister_hook(&nfho4);
}
```

On the same directory I created a file ("Makefile") to compile make our code.



```
obj-m += Firewall_Task2.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
~
```

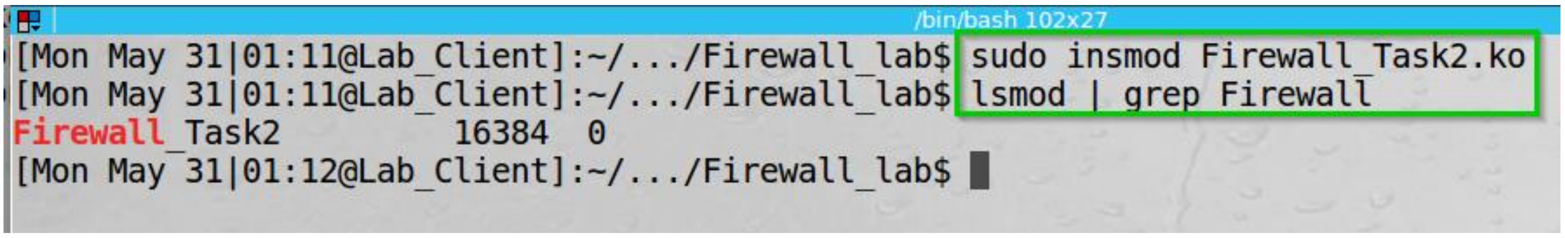
Executing "make" builds our code.



```
[Mon May 31|01:00@Lab_Client]:~/.../Firewall_lab$ make
make -C /lib/modules/4.8.0-36-generic/build M=/home/seed/Documents/Firewall_lab
modules
make[1]: Entering directory '/usr/src/linux-headers-4.8.0-36-generic'
  CC [M]  /home/seed/Documents/Firewall_lab/Firewall_Task2.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC      /home/seed/Documents/Firewall_lab/Firewall_Task2.mod.o
  LD [M]  /home/seed/Documents/Firewall_lab/Firewall_Task2.ko
make[1]: Leaving directory '/usr/src/linux-headers-4.8.0-36-generic'
[Mon May 31|01:00@Lab_Client]:~/.../Firewall_lab$
```

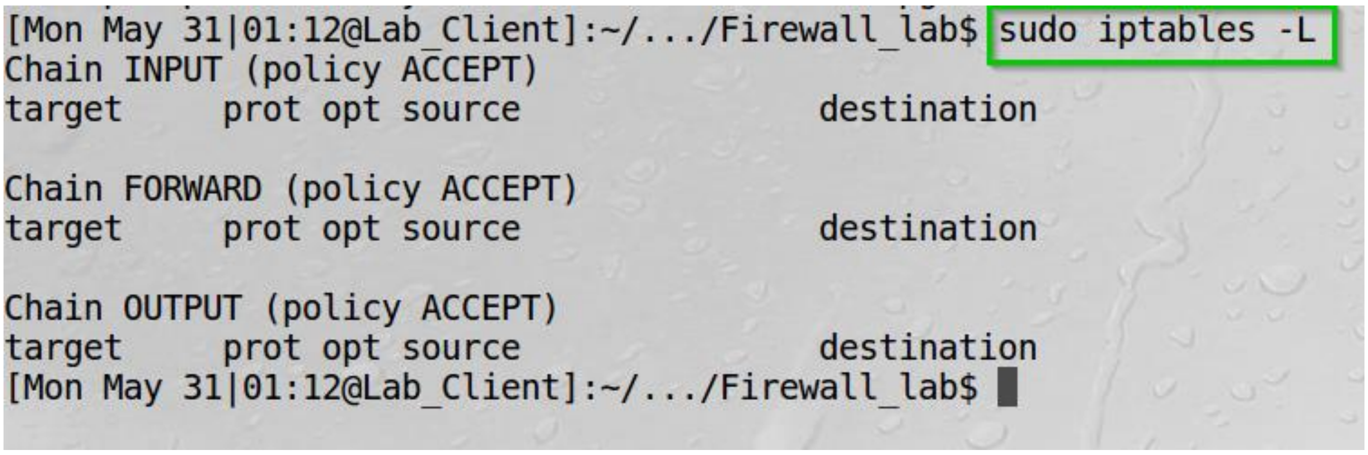
We can see no errors.
Note: when we tried it on the /home/see/host network mounted directory it did not work.

Inserted the module and see it exists.



```
[Mon May 31|01:11@Lab_Client]:~/.../Firewall_lab$ sudo insmod Firewall_Task2.ko
[Mon May 31|01:11@Lab_Client]:~/.../Firewall_lab$ lsmod | grep Firewall
Firewall_Task2          16384  0
[Mon May 31|01:12@Lab_Client]:~/.../Firewall_lab$
```

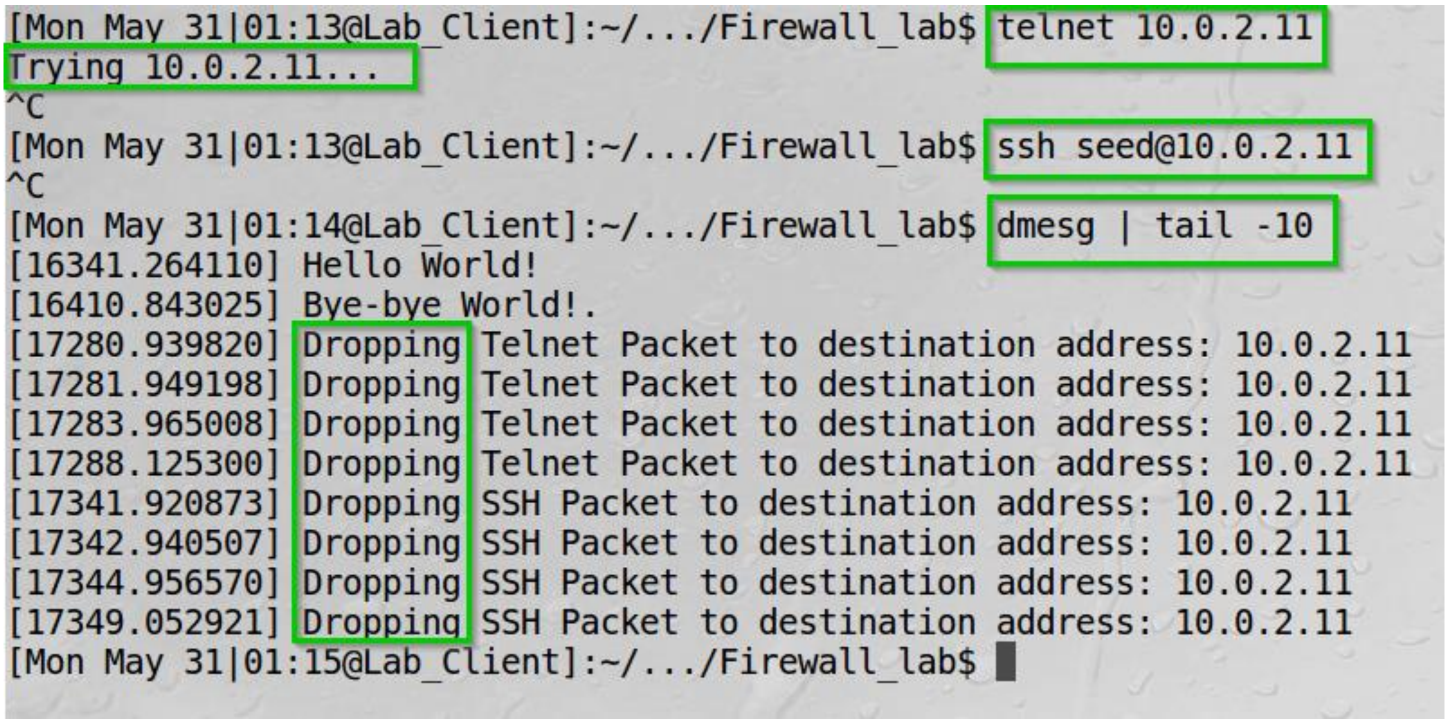
Checking the iptables rules.



```
[Mon May 31|01:12@Lab_Client]:~/.../Firewall_lab$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
[Mon May 31|01:12@Lab_Client]:~/.../Firewall_lab$
```

There are none so our modules are the one whom filtering the packets.

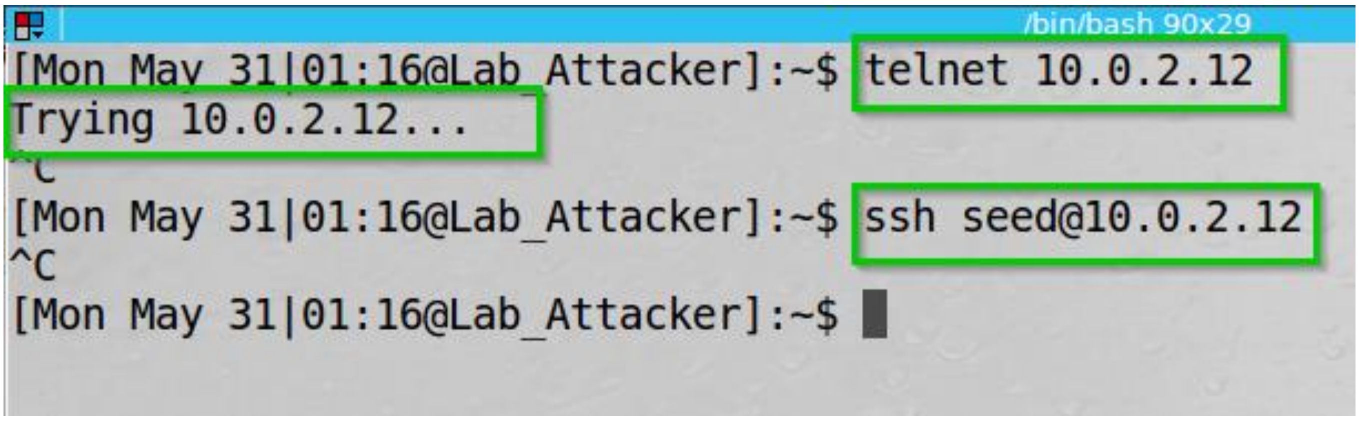
Trying Telnet and SSH from 10.0.2.12 to 10.0.2.11.



```
[Mon May 31|01:13@Lab_Client]:~/.../Firewall_lab$ telnet 10.0.2.11
Trying 10.0.2.11...
^C
[Mon May 31|01:13@Lab_Client]:~/.../Firewall_lab$ ssh seed@10.0.2.11
^C
[Mon May 31|01:14@Lab_Client]:~/.../Firewall_lab$ dmesg | tail -10
[16341.264110] Hello World!
[16410.843025] Bye-bye World!.
[17280.939820] Dropping Telnet Packet to destination address: 10.0.2.11
[17281.949198] Dropping Telnet Packet to destination address: 10.0.2.11
[17283.965008] Dropping Telnet Packet to destination address: 10.0.2.11
[17288.125300] Dropping Telnet Packet to destination address: 10.0.2.11
[17341.920873] Dropping SSH Packet to destination address: 10.0.2.11
[17342.940507] Dropping SSH Packet to destination address: 10.0.2.11
[17344.956570] Dropping SSH Packet to destination address: 10.0.2.11
[17349.052921] Dropping SSH Packet to destination address: 10.0.2.11
[Mon May 31|01:15@Lab_Client]:~/.../Firewall_lab$
```

We can see unsuccessful attempts and the log file showing our filtering.

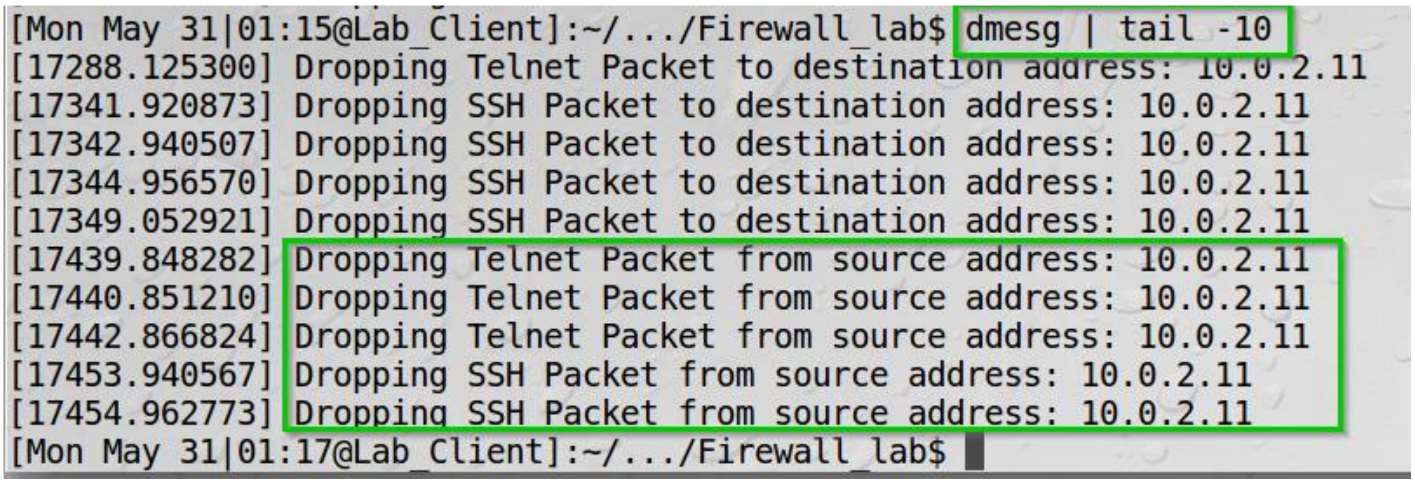
Trying Telnet and SSH from 10.0.2.11 to 10.0.2.12.



```
/bin/bash 90x29
[Mon May 31|01:16@Lab_Attacker]:~$ telnet 10.0.2.12
Trying 10.0.2.12...
^C
[Mon May 31|01:16@Lab_Attacker]:~$ ssh seed@10.0.2.12
^C
[Mon May 31|01:16@Lab_Attacker]:~$ 
```

We can see unsuccessful attempts.

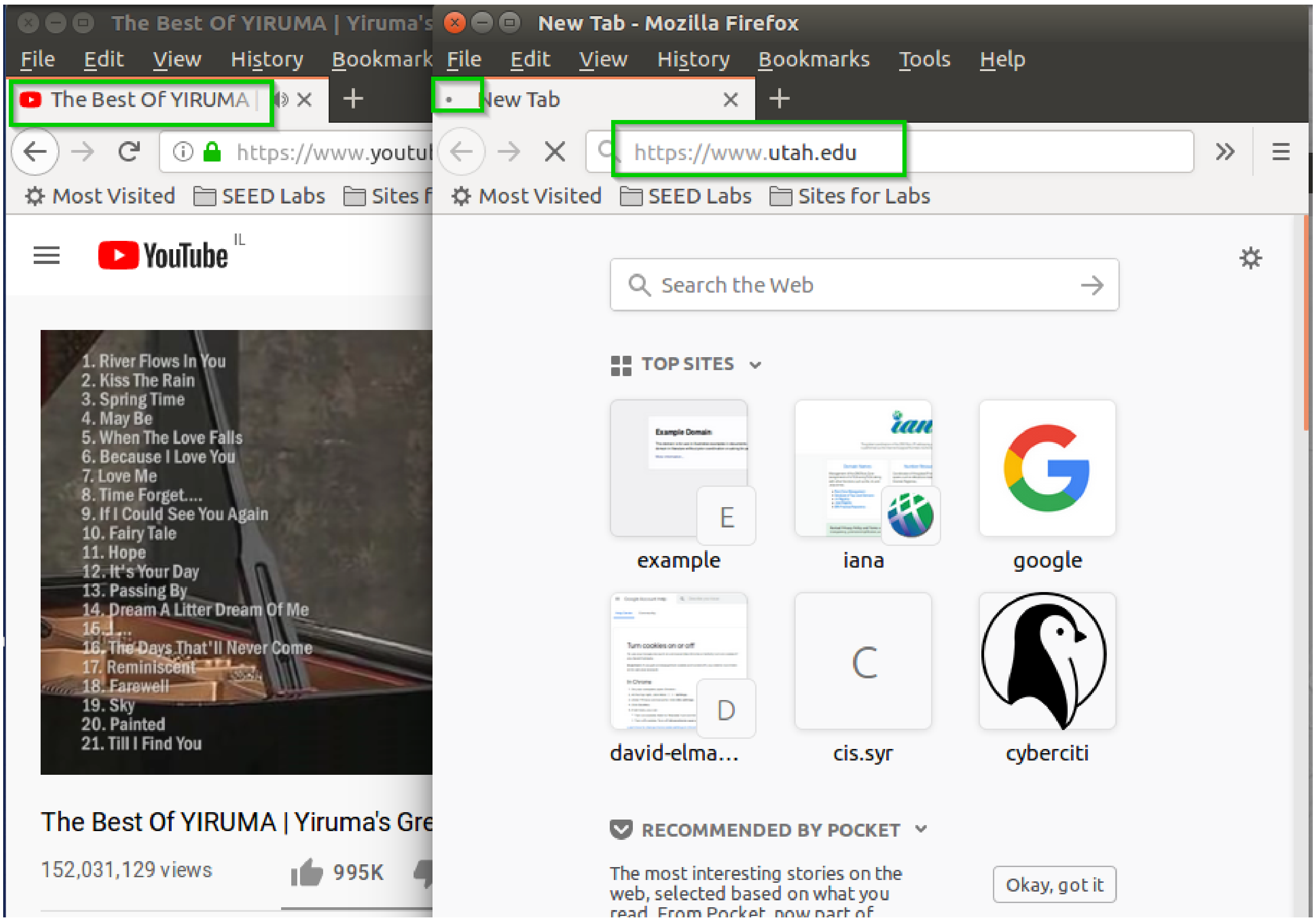
Viewing the log file.



```
[Mon May 31|01:15@Lab_Client]:~/.../Firewall_lab$ dmesg | tail -10
[17288.125300] Dropping Telnet Packet to destination address: 10.0.2.11
[17341.920873] Dropping SSH Packet to destination address: 10.0.2.11
[17342.940507] Dropping SSH Packet to destination address: 10.0.2.11
[17344.956570] Dropping SSH Packet to destination address: 10.0.2.11
[17349.052921] Dropping SSH Packet to destination address: 10.0.2.11
[17439.848282] Dropping Telnet Packet from source address: 10.0.2.11
[17440.851210] Dropping Telnet Packet from source address: 10.0.2.11
[17442.866824] Dropping Telnet Packet from source address: 10.0.2.11
[17453.940567] Dropping SSH Packet from source address: 10.0.2.11
[17454.962773] Dropping SSH Packet from source address: 10.0.2.11
[Mon May 31|01:17@Lab_Client]:~/.../Firewall_lab$ 
```

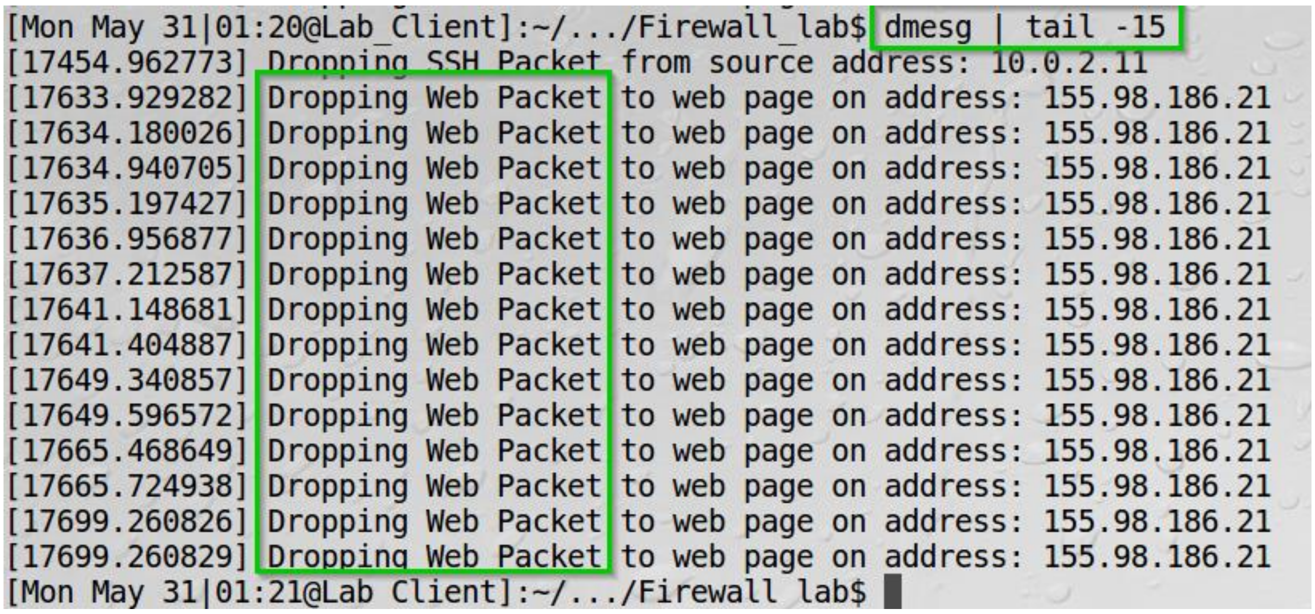
We can see the packets have dropped.

Trying to browse from 10.0.2.11 to Utah University website.



We can see unsuccessful attempt.

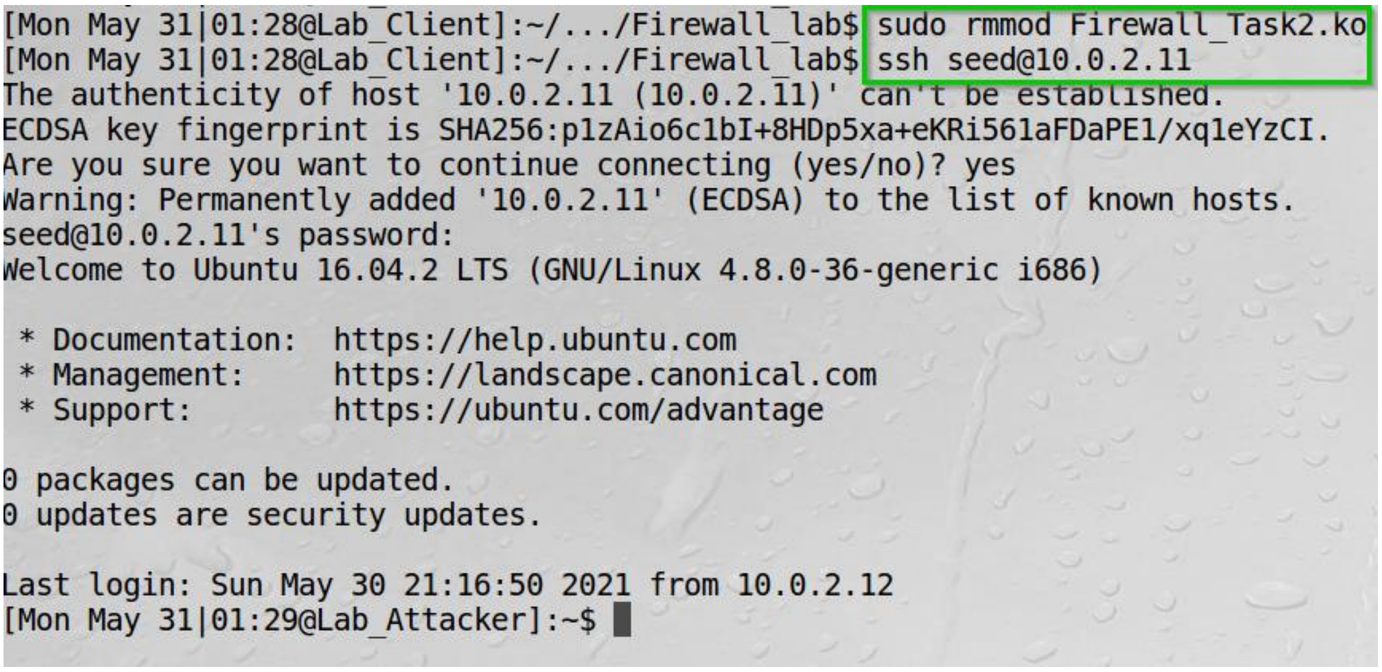
Viewing the log file.



```
[Mon May 31|01:20@Lab_Client]:~/.../Firewall_lab$ dmesg | tail -15
[17454.962773] Dropping SSH Packet from source address: 10.0.2.11
[17633.929282] Dropping Web Packet to web page on address: 155.98.186.21
[17634.180026] Dropping Web Packet to web page on address: 155.98.186.21
[17634.940705] Dropping Web Packet to web page on address: 155.98.186.21
[17635.197427] Dropping Web Packet to web page on address: 155.98.186.21
[17636.956877] Dropping Web Packet to web page on address: 155.98.186.21
[17637.212587] Dropping Web Packet to web page on address: 155.98.186.21
[17641.148681] Dropping Web Packet to web page on address: 155.98.186.21
[17641.404887] Dropping Web Packet to web page on address: 155.98.186.21
[17649.340857] Dropping Web Packet to web page on address: 155.98.186.21
[17649.596572] Dropping Web Packet to web page on address: 155.98.186.21
[17665.468649] Dropping Web Packet to web page on address: 155.98.186.21
[17665.724938] Dropping Web Packet to web page on address: 155.98.186.21
[17699.260826] Dropping Web Packet to web page on address: 155.98.186.21
[17699.260829] Dropping Web Packet to web page on address: 155.98.186.21
[Mon May 31|01:21@Lab_Client]:~/.../Firewall_lab$ 
```

We can see the packets have dropped.

Removing the module and SSH to 10.0.2.11.



```
[Mon May 31|01:28@Lab_Client]:~/.../Firewall_lab$ sudo rmmod Firewall_Task2.ko
[Mon May 31|01:28@Lab_Client]:~/.../Firewall_lab$ ssh seed@10.0.2.11
The authenticity of host '10.0.2.11 (10.0.2.11)' 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 '10.0.2.11' (ECDSA) to the list of known hosts.
seed@10.0.2.11'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

0 packages can be updated.
0 updates are security updates.

Last login: Sun May 30 21:16:50 2021 from 10.0.2.12
[Mon May 31|01:29@Lab_Attacker]:~$
```

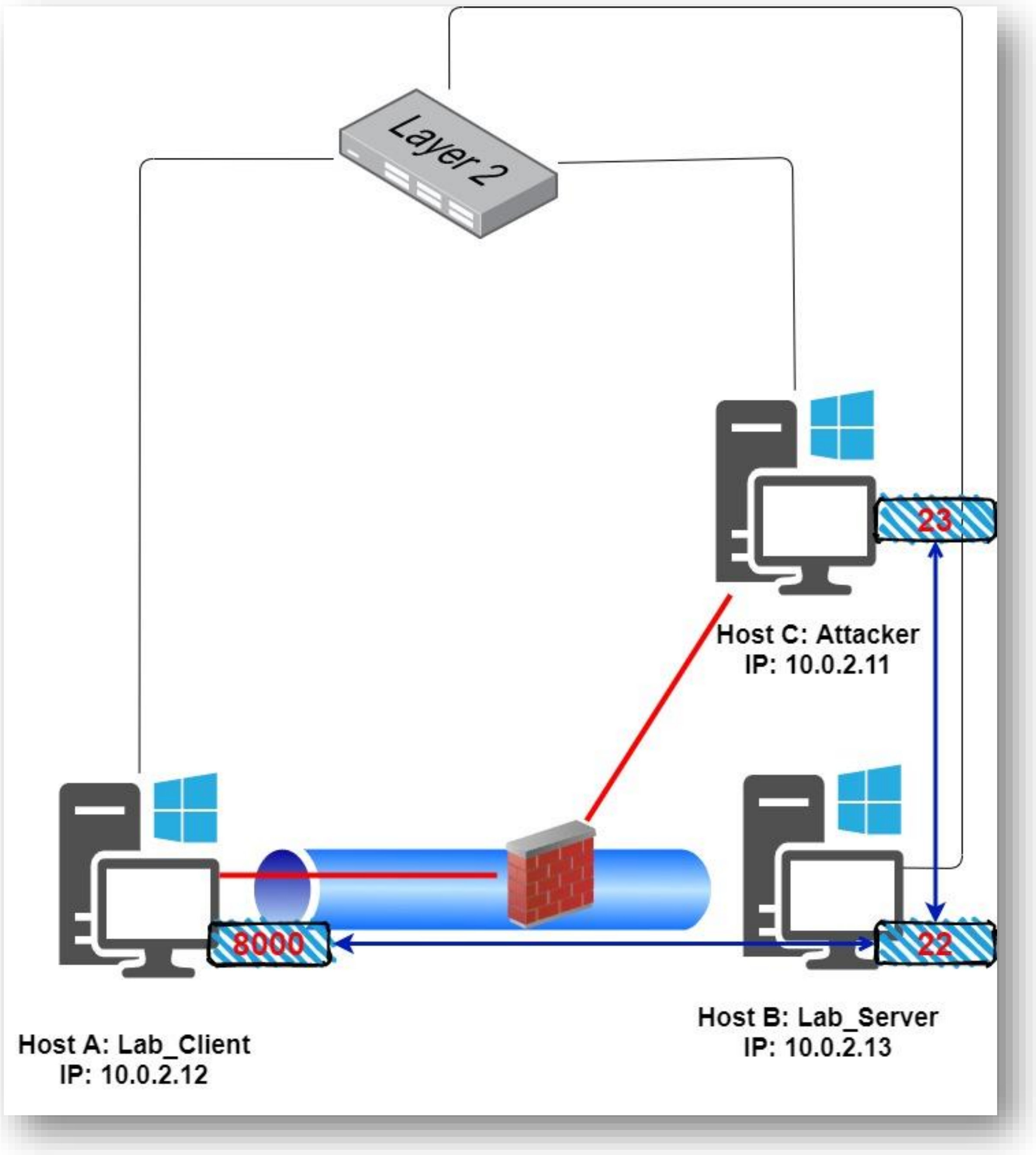
We can see a successful connection.

<u>**Task 2 Summary**</u>

- I have learned about LKM – Loadable Kernel Modules which are modules we can insert to our kernel on the fly.
- I have learned about Netfilter – which allows us to load the LKM and call hooks on the operating system – Meaning we catch the system calls before the kernel decides what to do with it and forward it to the operating system.
- I have learned about Makefiles and how to build our code and then use these files and insert modules to the Linux kernel.
- I implemented hooks to catch SSH, Telnet and connection to Utah University website and decide what to do with it.

Task 3: Evading Egress Filter

Network physical topology:



<u>Task Description:</u>

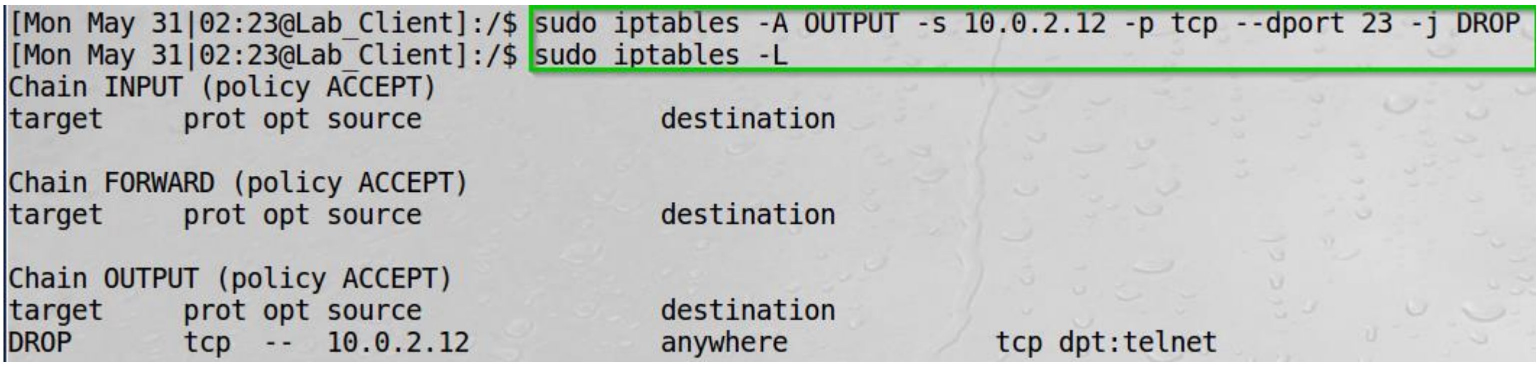
Task 3.a - Iptables rules will deny us from initiating telnet to a remote server.

I will evade iptables filtering by using SSH connection from one host to another through intermediate host.

Task 3.b - I will also block communication with Utah University website and set Firefox proxy as localhost:9000 to bypass the block (using dynamic port forwarding).

<u>Task 3.a:</u>

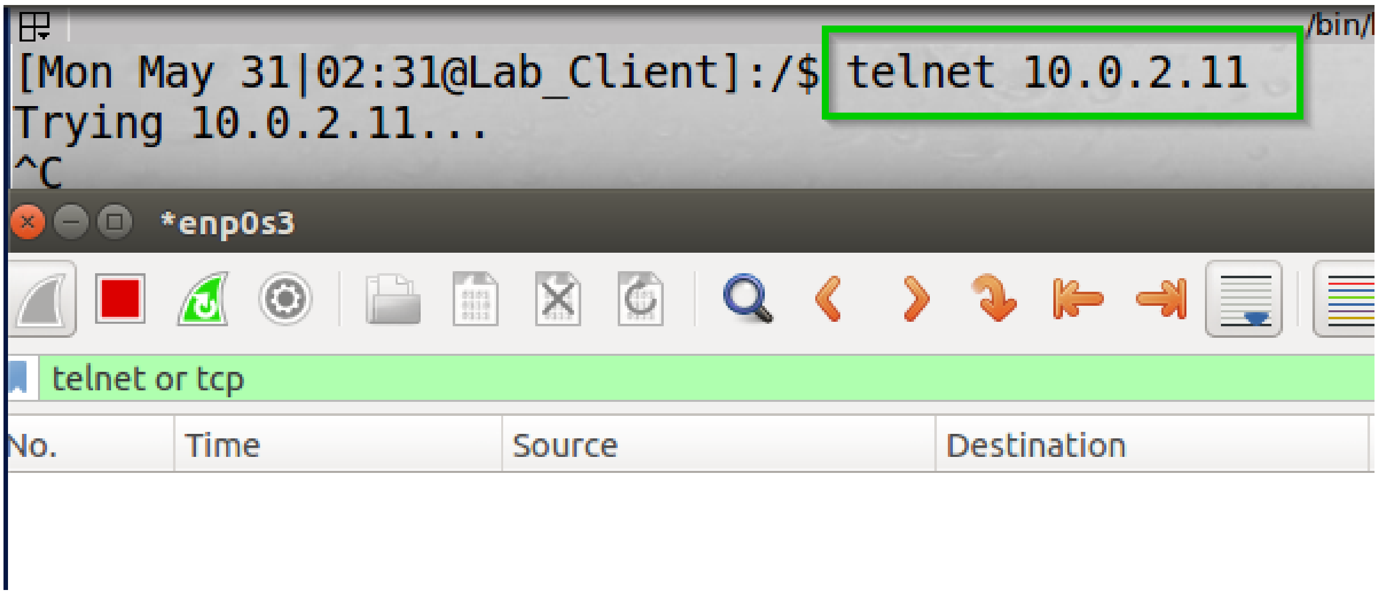
The following rule filter outgoing telnet session from 10.0.2.12 to anywhere.



```
[Mon May 31|02:23@Lab_Client]:/$ sudo iptables -A OUTPUT -s 10.0.2.12 -p tcp --dport 23 -j DROP
[Mon May 31|02:23@Lab_Client]:/$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
DROP       tcp  --  10.0.2.12            anywhere             tcp dpt:telnet
```

We can see the rule updated in the iptables.

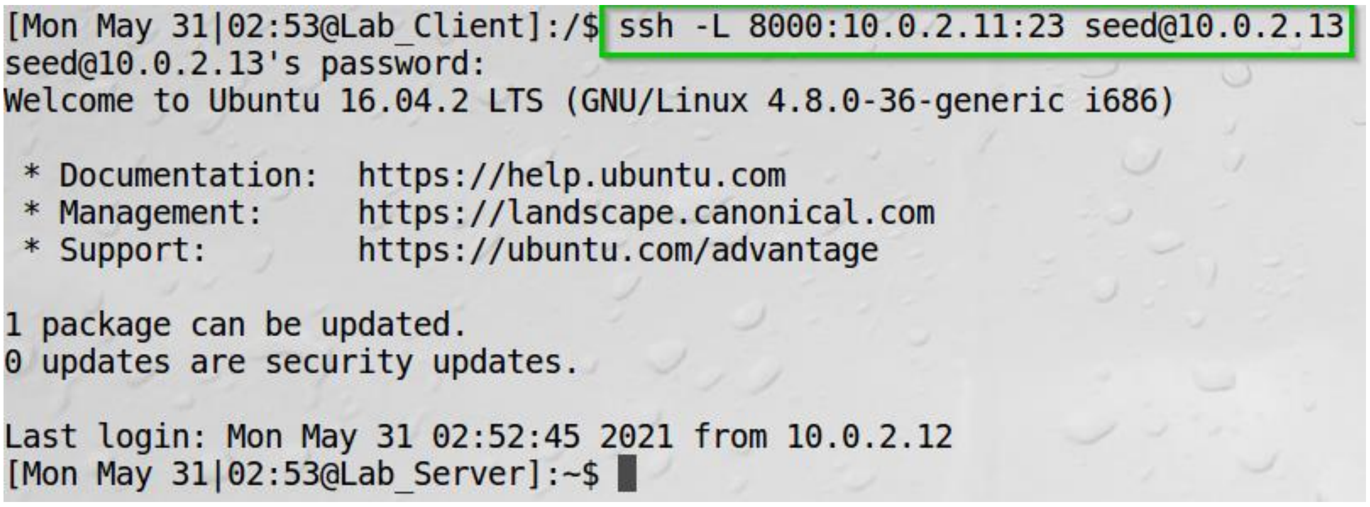
I tried to initiate telnet session to 10.0.2.11.



```
[Mon May 31|02:31@Lab_Client]:/$ telnet 10.0.2.11
Trying 10.0.2.11...
^C
```

We can see unsuccessful attempt and no packets on Wireshark.

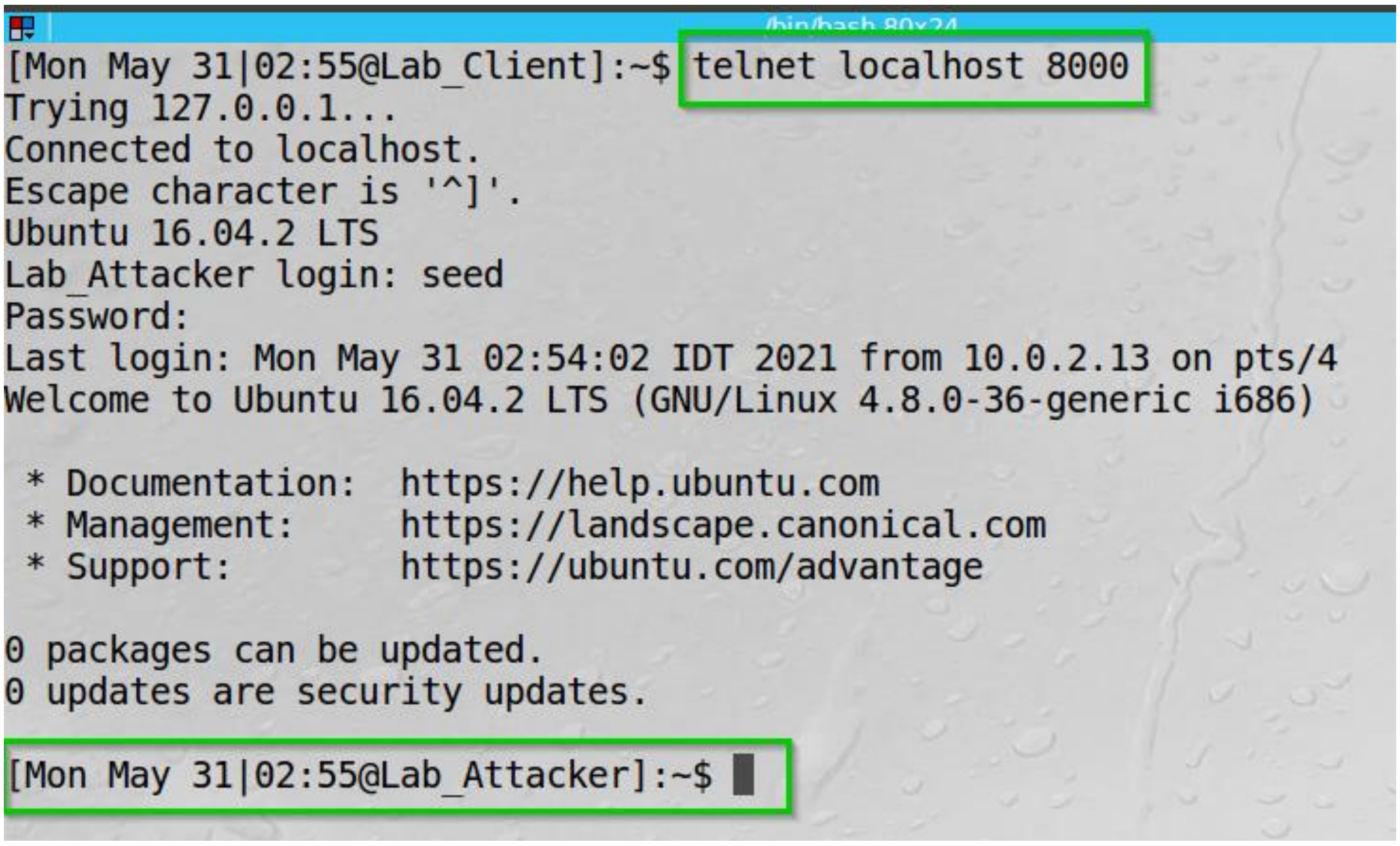
I established SSH tunnel between host A and B.



```
[Mon May 31|02:53@Lab_Client]:/$ ssh -L 8000:10.0.2.11:23 seed@10.0.2.13
seed@10.0.2.13'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 May 31 02:52:45 2021 from 10.0.2.12
[Mon May 31|02:53@Lab_Server]:~$
```

We can see successful login to 10.0.2.13 using SSH.

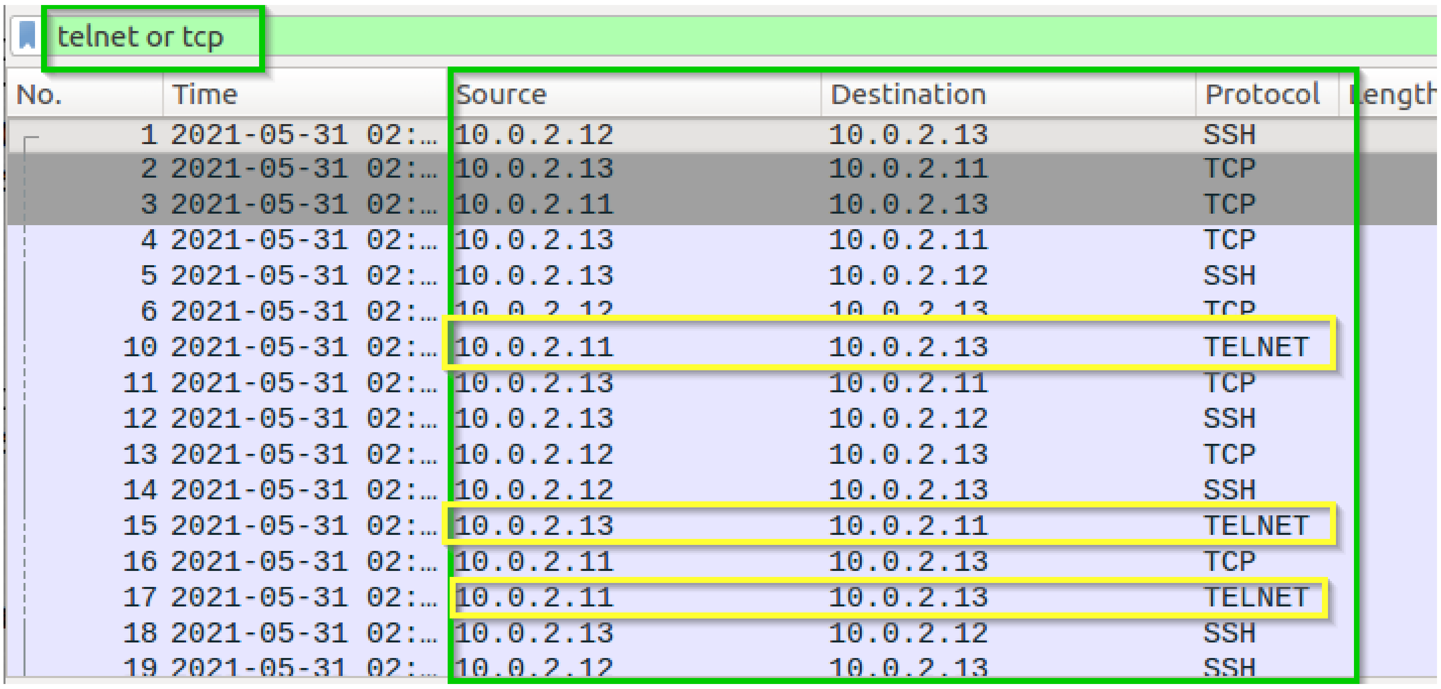
Then I used the tunnel we created – telnet to localhost on port 8000.



```
[Mon May 31|02:55@Lab_Client]:~$ telnet localhost 8000
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
Lab_Attacker login: seed
Password:
Last login: Mon May 31 02:54:02 IDT 2021 from 10.0.2.13 on pts/4
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

0 packages can be updated.
0 updates are security updates.

[Mon May 31|02:55@Lab_Attacker]:~$
```

We can see successful connection to C (10.0.2.11).

Wireshark view.

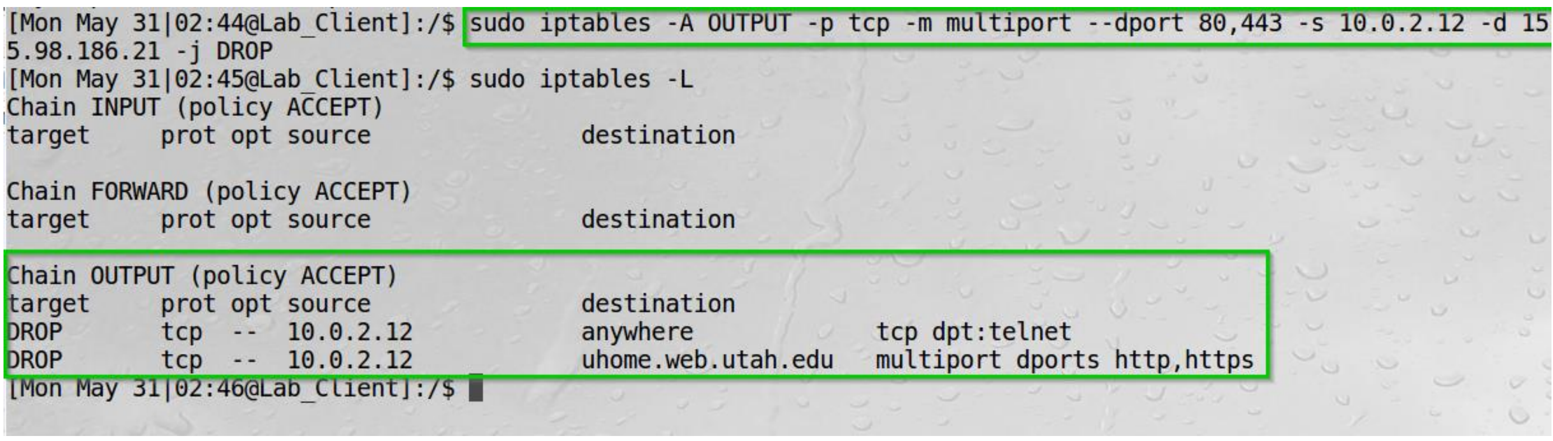


```
telnet or tcp
No.    Time               Source        Destination   Protocol  Length
 1 2021-05-31 02:…  10.0.2.12     10.0.2.13     SSH
 2 2021-05-31 02:…  10.0.2.13     10.0.2.11     TCP
 3 2021-05-31 02:…  10.0.2.11     10.0.2.13     TCP
 4 2021-05-31 02:…  10.0.2.13     10.0.2.11     TCP
 5 2021-05-31 02:…  10.0.2.13     10.0.2.12     SSH
 6 2021-05-31 02:…  10.0.2.12     10.0.2.13     TCP
10 2021-05-31 02:…  10.0.2.11     10.0.2.13     TELNET
11 2021-05-31 02:…  10.0.2.13     10.0.2.11     TCP
12 2021-05-31 02:…  10.0.2.13     10.0.2.12     SSH
13 2021-05-31 02:…  10.0.2.12     10.0.2.13     TCP
14 2021-05-31 02:…  10.0.2.12     10.0.2.13     SSH
15 2021-05-31 02:…  10.0.2.13     10.0.2.11     TELNET
16 2021-05-31 02:…  10.0.2.11     10.0.2.13     TCP
17 2021-05-31 02:…  10.0.2.11     10.0.2.13     TELNET
18 2021-05-31 02:…  10.0.2.13     10.0.2.12     SSH
19 2021-05-31 02:…  10.0.2.12     10.0.2.13     SSH
```

We can see we (10.0.2.12) communicate with B (10.0.2.13) through SSH, and B (10.0.2.13) communicate with C (10.0.2.11) through telnet – thus creating a tunnel for us to communicate with 10.0.2.11.

Task 3.b:

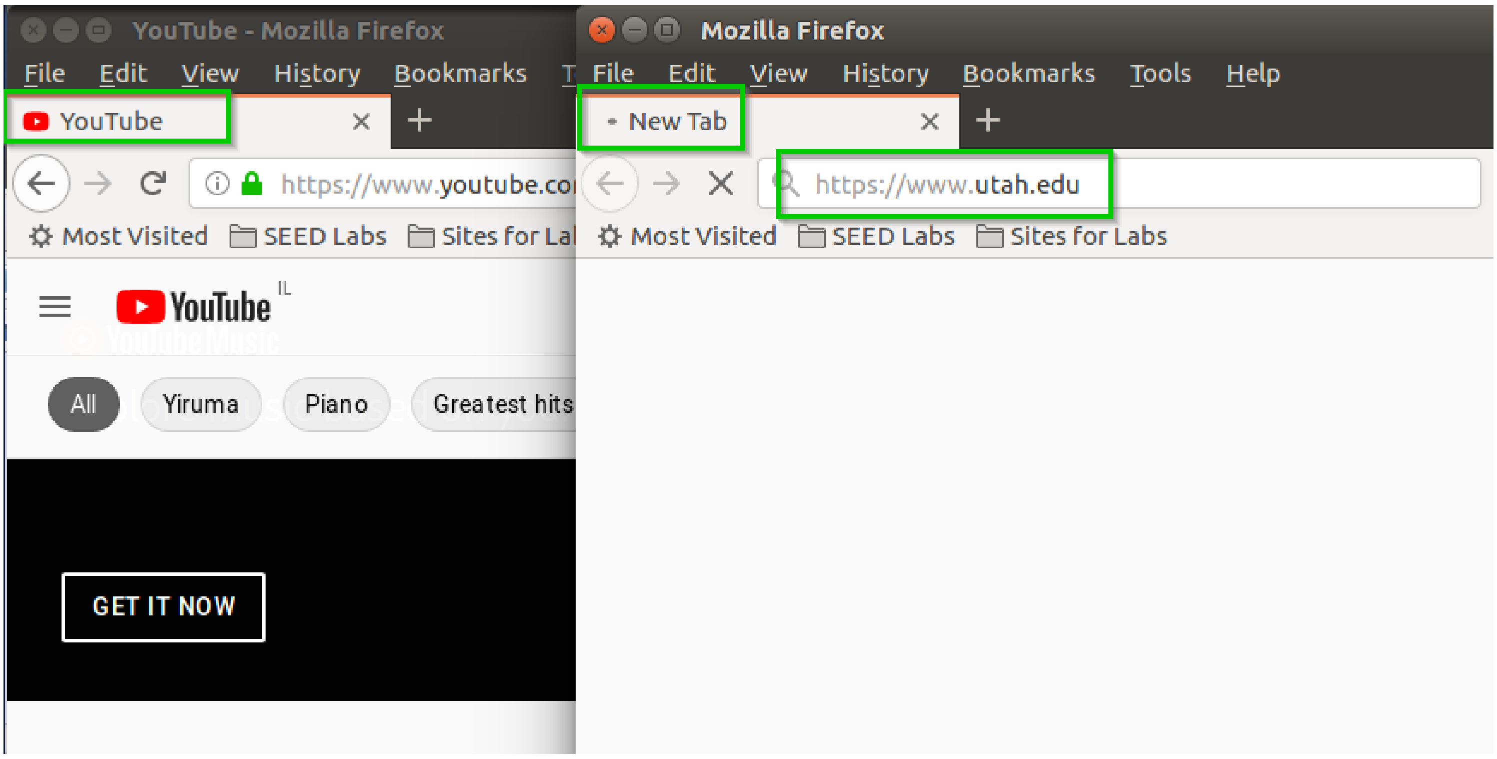
The following rule block out going traffic on ports 80,443 from 10.0.2.12 to Utah University website.



```
[Mon May 31|02:44@Lab_Client]:/$ sudo iptables -A OUTPUT -p tcp -m multiport --dport 80,443 -s 10.0.2.12 -d 15
5.98.186.21 -j DROP
[Mon May 31|02:45@Lab_Client]:/$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
DROP       tcp  --  10.0.2.12            anywhere             tcp dpt:telnet
DROP       tcp  --  10.0.2.12            uhome.web.utah.edu   multiport dports http,https
[Mon May 31|02:46@Lab_Client]:/$ 
```

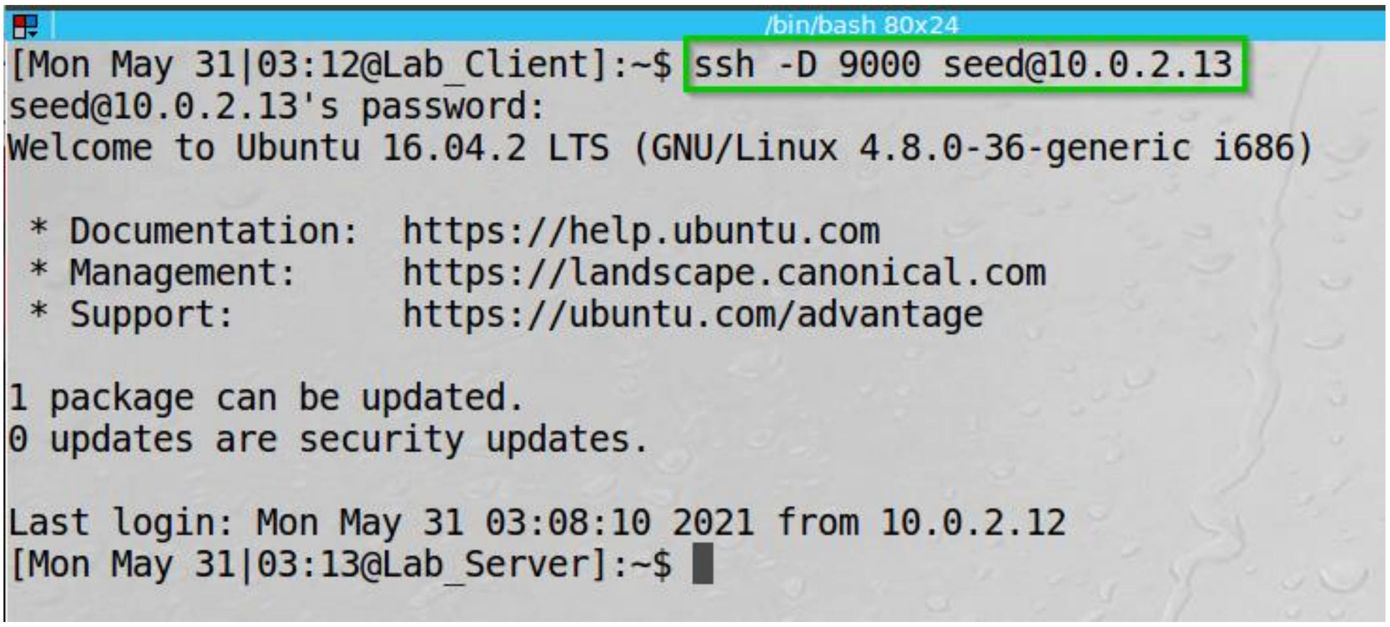
We can see the rule in iptables.

I now try browsing to Utah University website.


YouTube - Mozilla Firefox
File Edit View History Bookmarks
YouTube
https://www.youtube.co
Most Visited SEED Labs Sites for La
YouTube
All Yiruma Piano Greatest hits
GET IT NOW
Mozilla Firefox
File Edit View History Bookmarks Tools Help
New Tab
https://www.utah.edu
Most Visited SEED Labs Sites for Labs

We can see unsuccessful attempt.
YouTube is to show network connectivity.

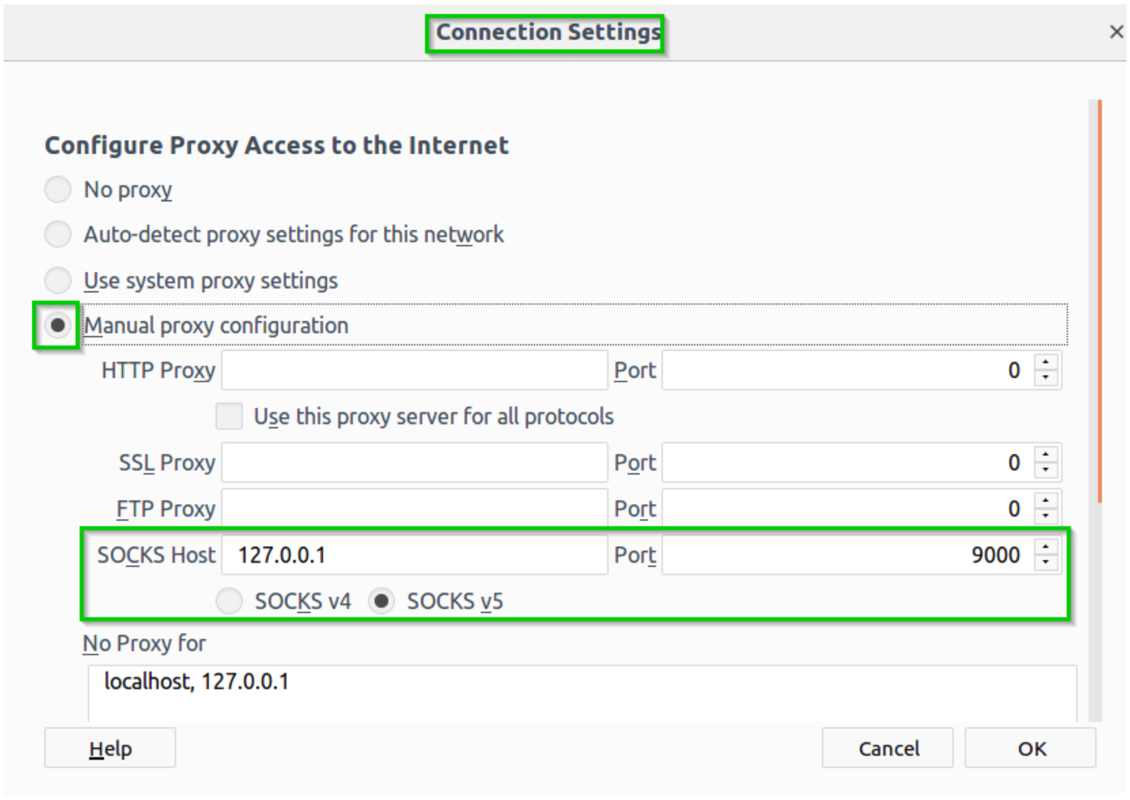
I then SSH to 10.0.2.13 on port 9000 and later configure Firefox to use this port as the port forwarding port.



```
[Mon May 31|03:12@Lab_Client]:~$ ssh -D 9000 seed@10.0.2.13
seed@10.0.2.13'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 May 31 03:08:10 2021 from 10.0.2.12
[Mon May 31|03:13@Lab_Server]:~$ 
```

We can see successful connection.

I configured Firefox proxy settings.



Connection Settings

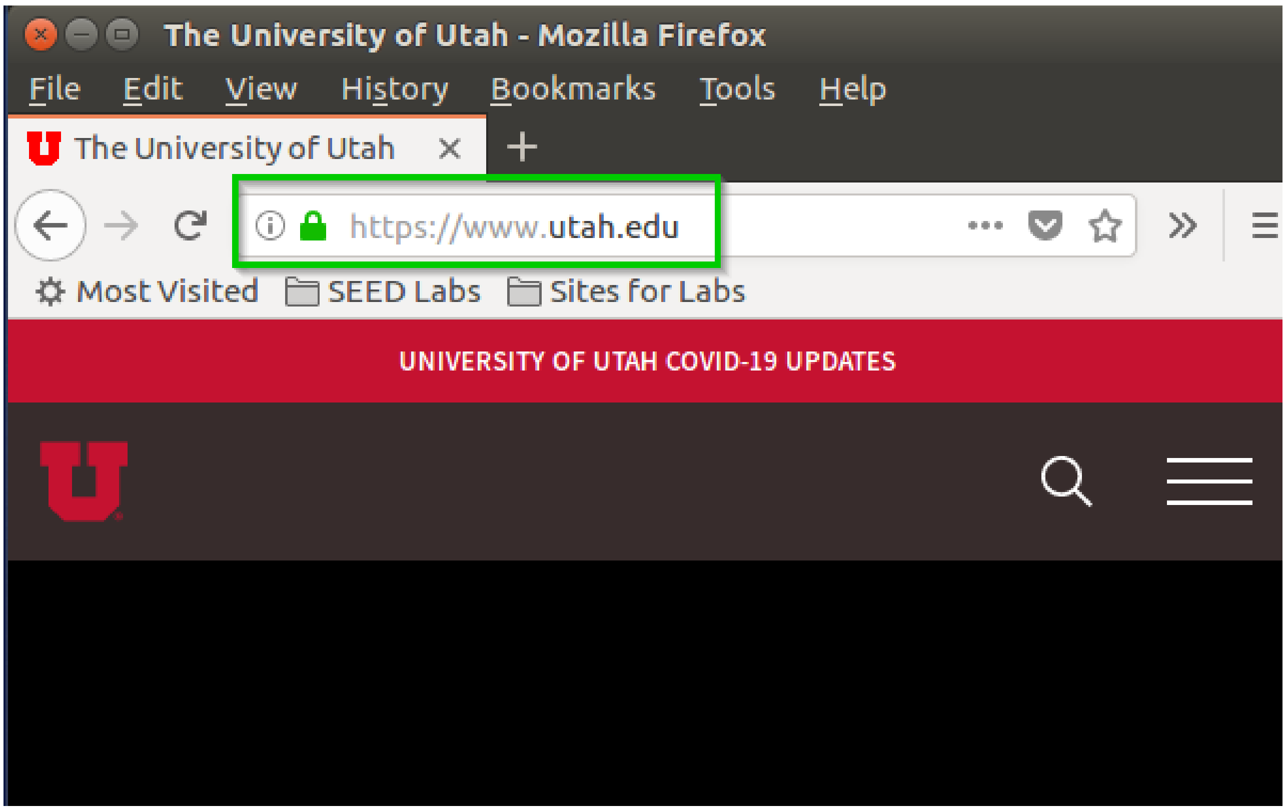
Configure Proxy Access to the Internet

No proxy
Auto-detect proxy settings for this network
Use system proxy settings
[✓ Manual proxy configuration]

HTTP Proxy: Port: 0
Use this proxy server for all protocols
SSL Proxy: Port: 0
FTP Proxy: Port: 0
SOCKS Host: 127.0.0.1 Port: 9000
SOCKS v4 [✓ SOCKS v5]

No Proxy for: localhost, 127.0.0.1

Then browsed again to Utah University website.

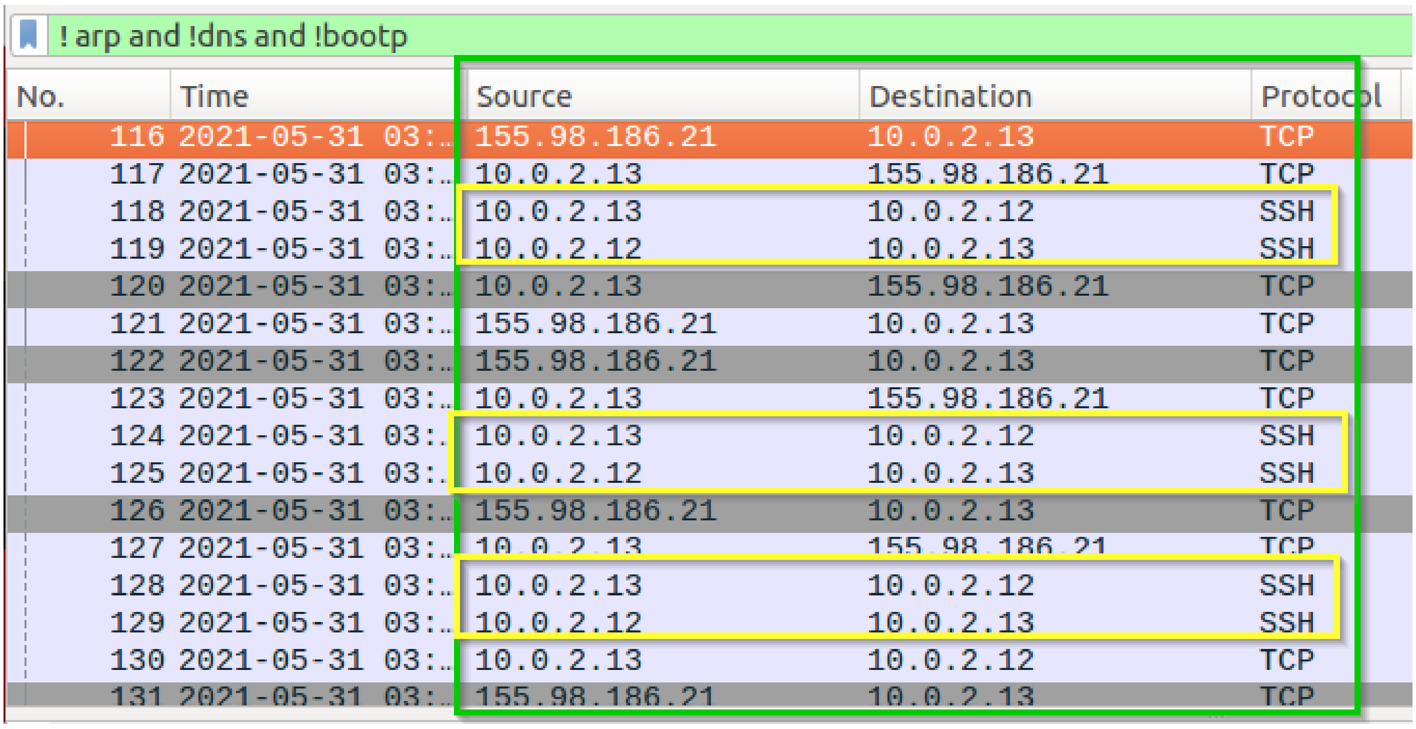


The University of Utah - Mozilla Firefox

https://www.utah.edu

We can see a successful connection.

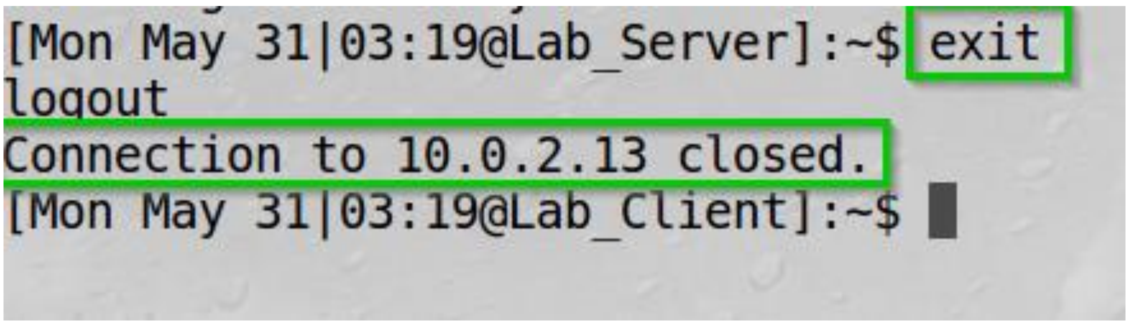
Wireshark view.



```
! arp and !dns and !bootp
No.  Time                Source          Destination     Protocol
 116 2021-05-31 03:…     155.98.186.21   10.0.2.13       TCP
 117 2021-05-31 03:…     10.0.2.13       155.98.186.21   TCP
 118 2021-05-31 03:…     10.0.2.13       10.0.2.12       SSH
 119 2021-05-31 03:…     10.0.2.12       10.0.2.13       SSH
 120 2021-05-31 03:…     10.0.2.13       155.98.186.21   TCP
 121 2021-05-31 03:…     155.98.186.21   10.0.2.13       TCP
 122 2021-05-31 03:…     155.98.186.21   10.0.2.13       TCP
 123 2021-05-31 03:…     10.0.2.13       155.98.186.21   TCP
 124 2021-05-31 03:…     10.0.2.13       10.0.2.12       SSH
 125 2021-05-31 03:…     10.0.2.12       10.0.2.13       SSH
 126 2021-05-31 03:…     155.98.186.21   10.0.2.13       TCP
 127 2021-05-31 03:…     10.0.2.13       155.98.186.21   TCP
 128 2021-05-31 03:…     10.0.2.13       10.0.2.12       SSH
 129 2021-05-31 03:…     10.0.2.12       10.0.2.13       SSH
 130 2021-05-31 03:…     10.0.2.13       10.0.2.12       TCP
 131 2021-05-31 03:…     155.98.186.21   10.0.2.13       TCP
```

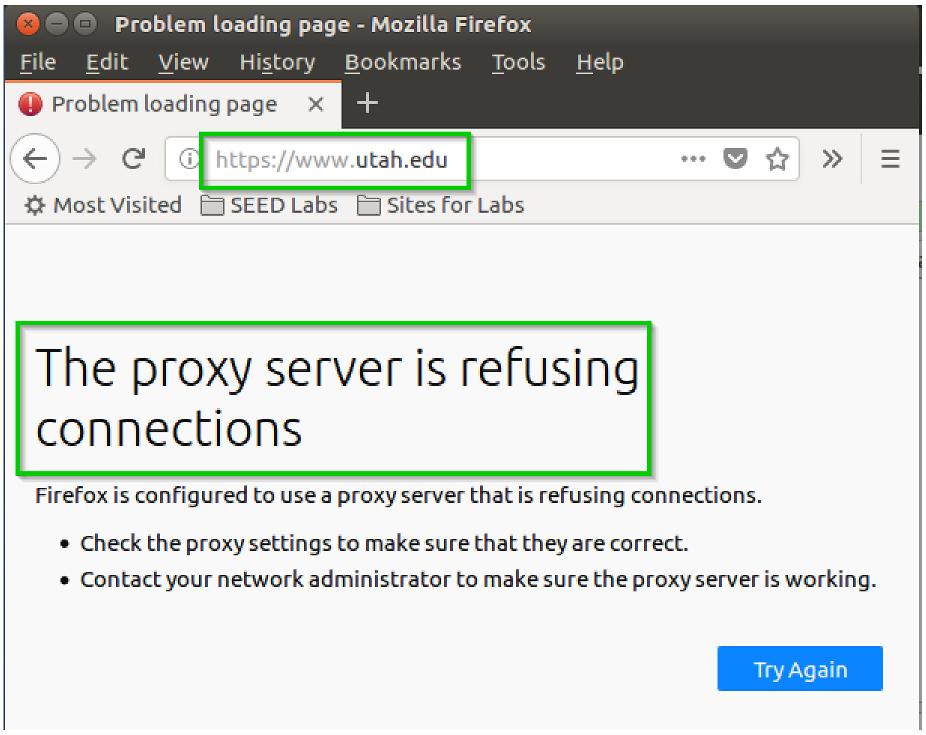
We can see we communicate with 10.0.2.13 through SSH and 10.0.2.13 communicate with 155.98.186.21 for us.

I broke the connection.



```
[Mon May 31|03:19@Lab_Server]:~$ exit
logout
Connection to 10.0.2.13 closed.
[Mon May 31|03:19@Lab_Client]:~$ 
```

Then tried browsing again.



Problem loading page - Mozilla Firefox

https://www.utah.edu

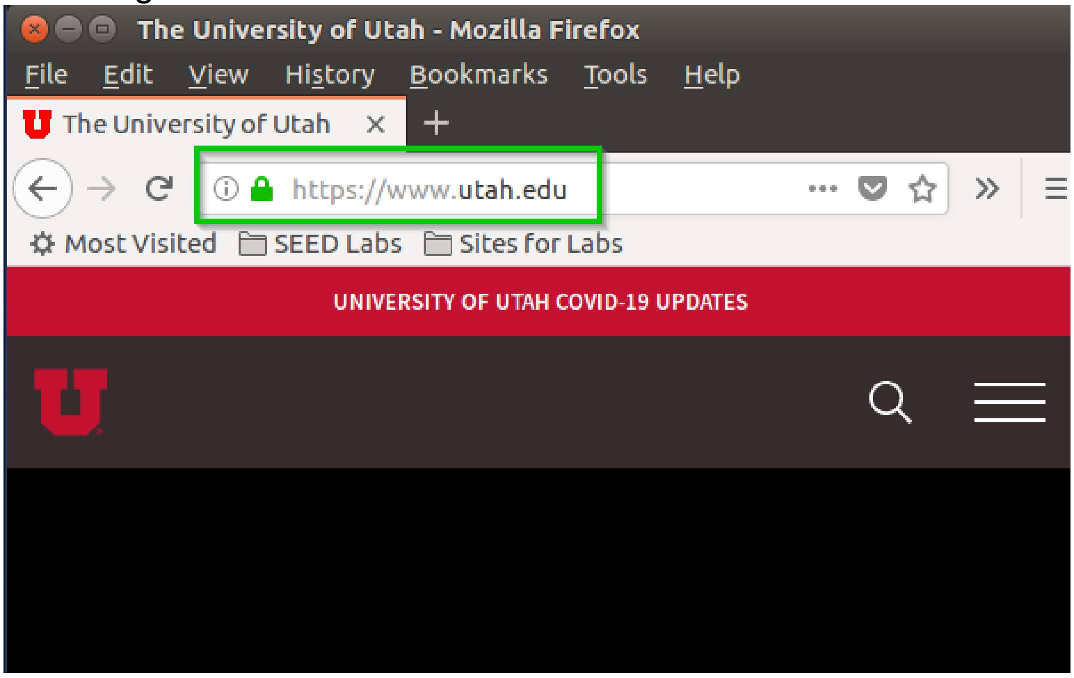
The proxy server is refusing connections

Firefox is configured to use a proxy server that is refusing connections.

- Check the proxy settings to make sure that they are correct.
- Contact your network administrator to make sure the proxy server is working.

We can see that we cannot connect to the University website. Firefox tries to communicate with localhost at port 9000, however it is closed.

After initiating the SSH tunnel again I can browse to the university website again.



The University of Utah - Mozilla Firefox

https://www.utah.edu

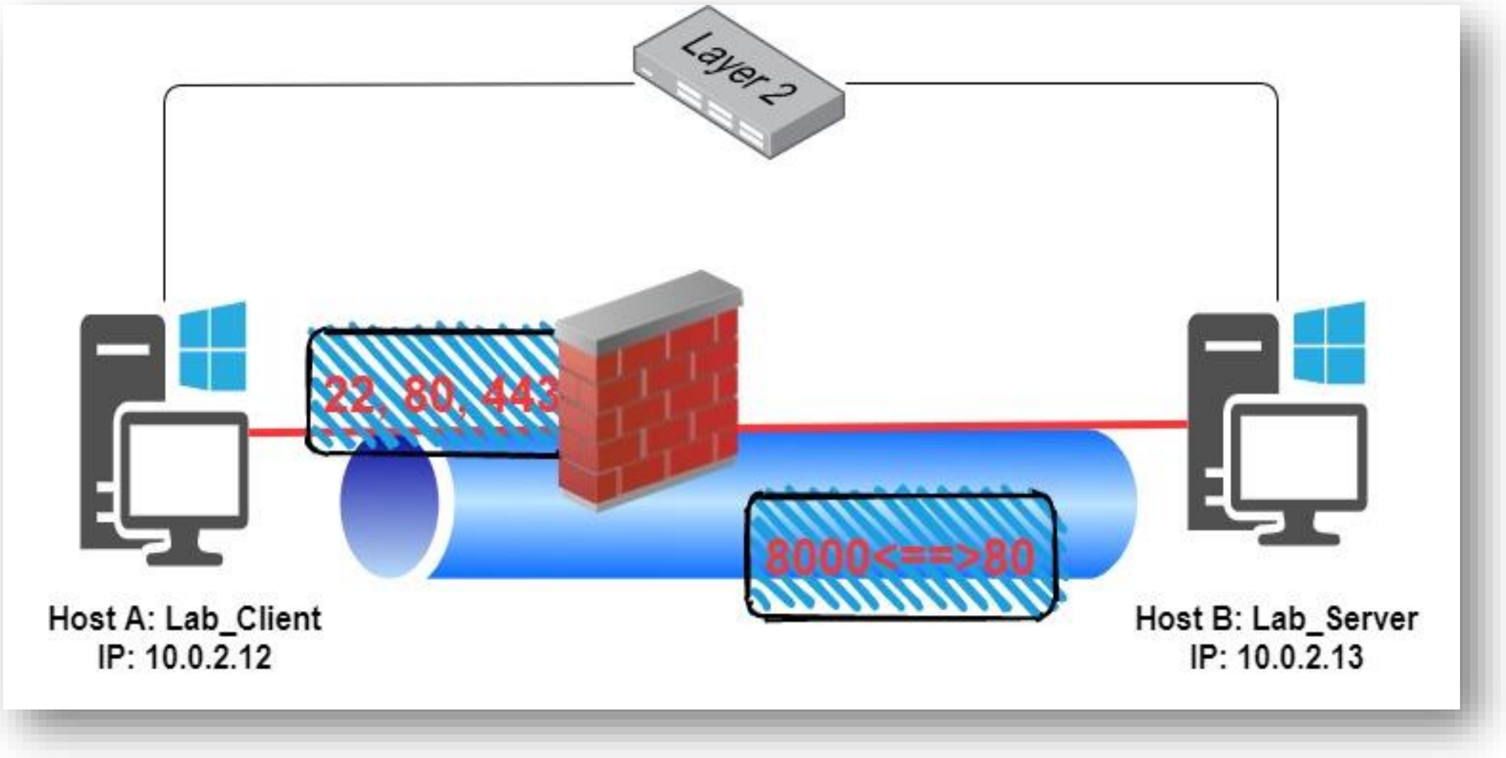
File Edit View History Bookmarks Tools Help
The University of Utah
Most Visited
SEED Labs
Sites for Labs
UNIVERSITY OF UTAH COVID-19 UPDATES

Task 3 summary

- I blocked outgoing Telnet from 10.0.2.12, then executed ‘ssh -L 8000:10.0.2.11:23 seed@10.0.2.13’ to create a tunnel from 10.0.2.12 to 10.0.2.11 through 10.0.2.13.
- I blocked ports 80 and 443 to Utah University website, the executed ‘ssh -D 9000 seed@10.0.2.13’ and configured Firefox proxy through port 9000
  Then connected to 9000 port to SSH and could browse to Utah University website.

Task 4: Evading Egress Filter

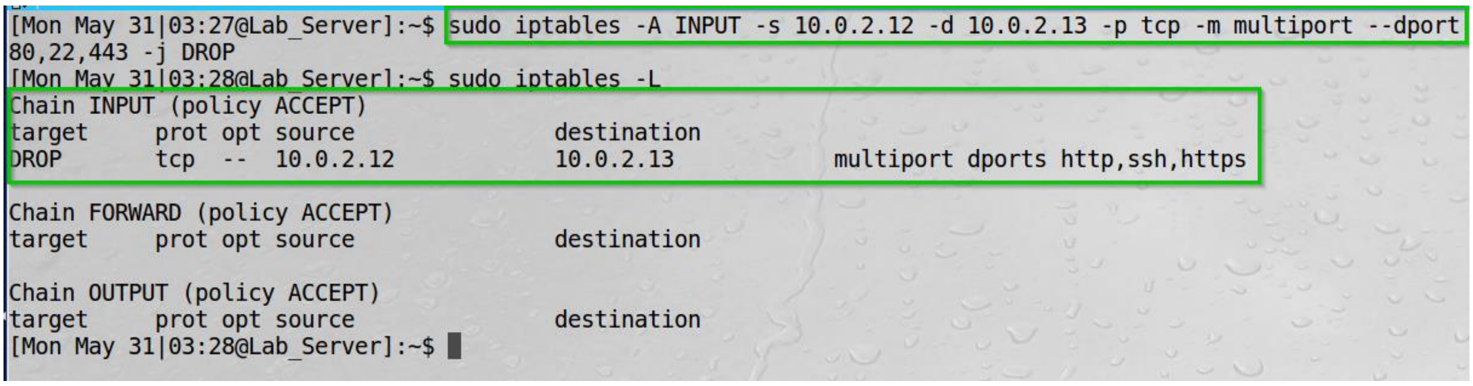
Network physical topology:



Task Description:

Similarly, to Task 3, I block ports 22,80,443 from A (10.0.2.12) to B (10.0.2.13), however the tunnel now would be SSH Reverse Path from B to A.

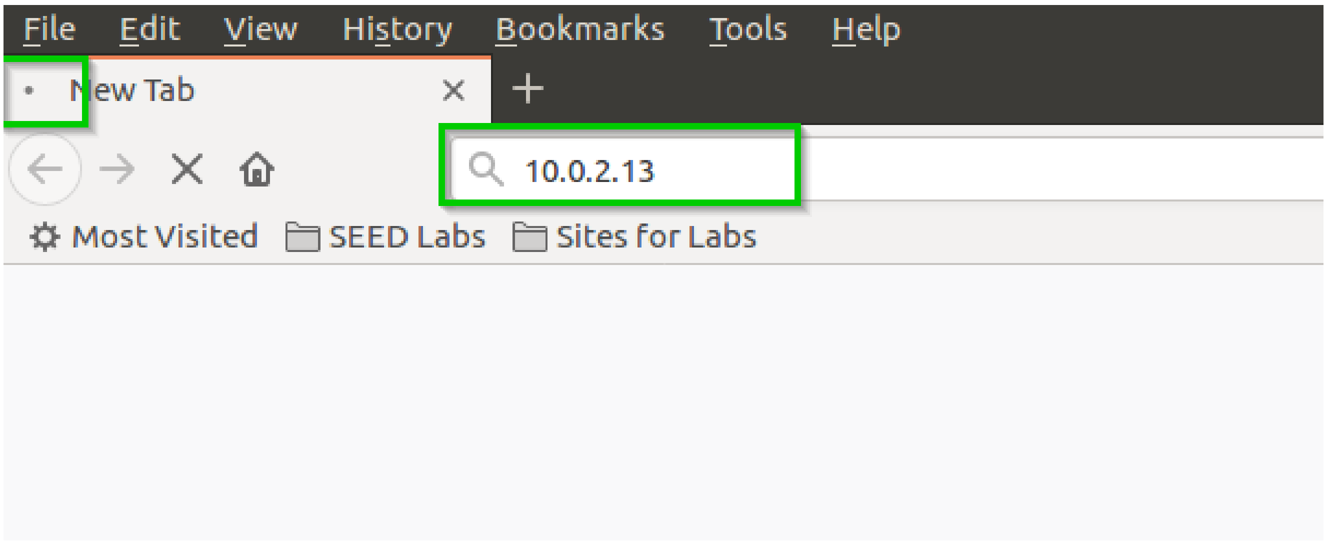
The following rule filter Incoming traffic on ports 22,80,443 from 10.0.2.12 to 10.0.2.13.



```
[Mon May 31|03:27@Lab_Server]:~$ sudo iptables -A INPUT -s 10.0.2.12 -d 10.0.2.13 -p tcp -m multiport --dport
80,22,443 -j DROP
[Mon May 31|03:28@Lab_Server]:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       tcp  --  10.0.2.12            10.0.2.13            multiport dports http,ssh,https

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
[Mon May 31|03:28@Lab_Server]:~$
```

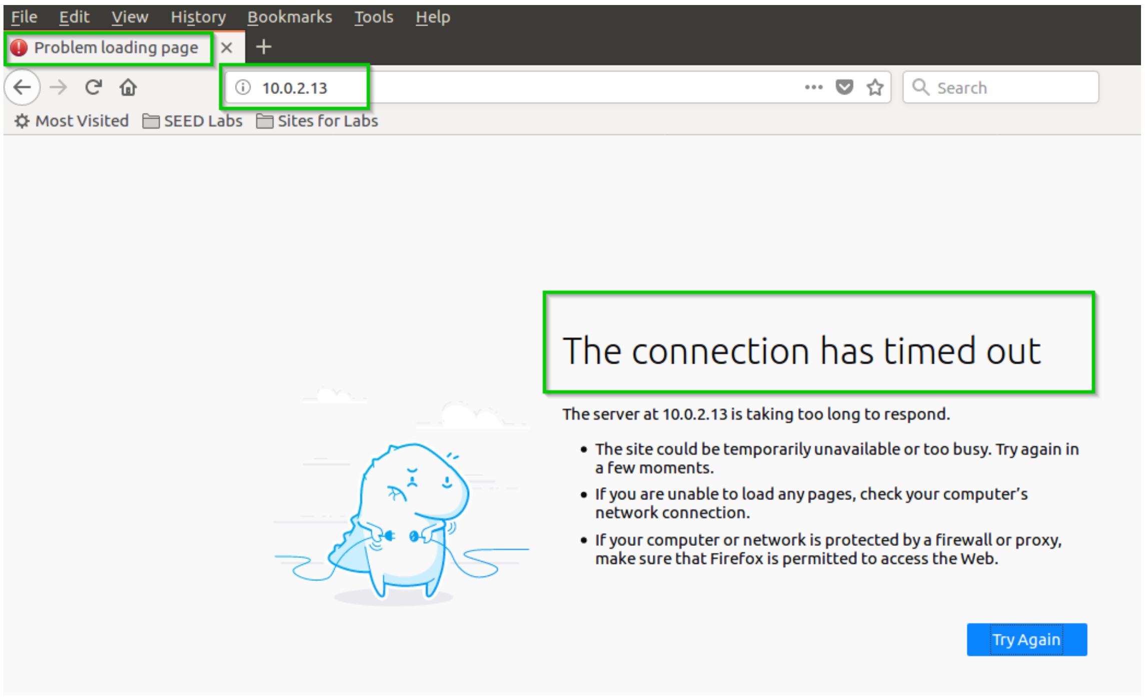
We can see the filter updated in the iptables.

I browsed to 10.0.2.13 website.



We can see unsuccessful attempt.

We can even see the connection is timed out.

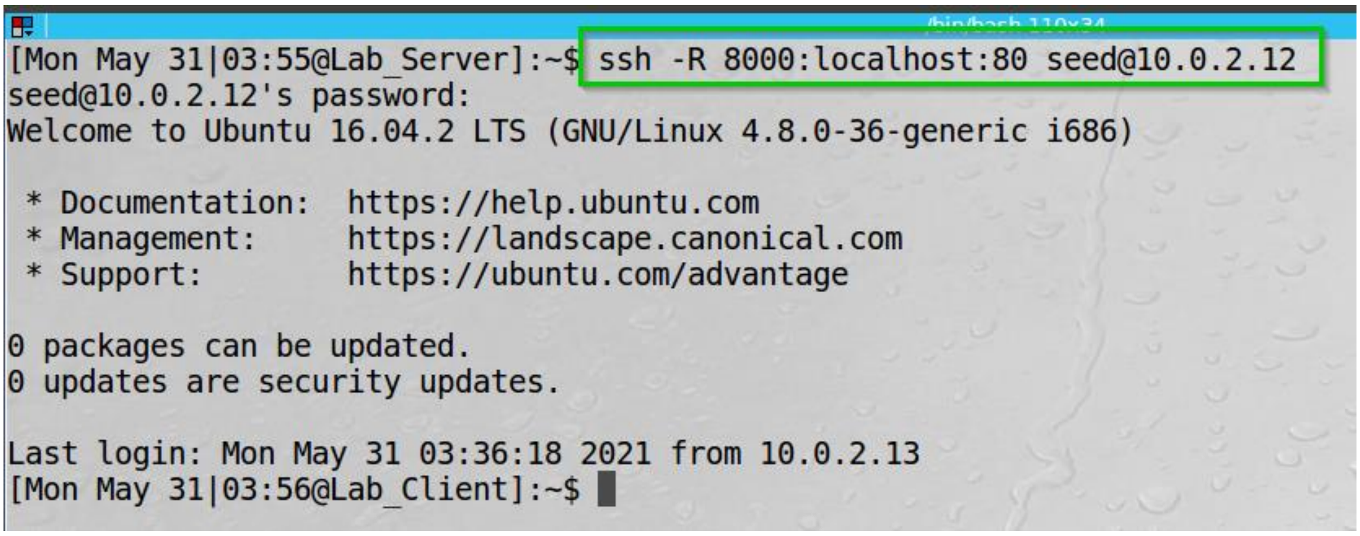


The connection has timed out

The server at 10.0.2.13 is taking too long to respond.

- The site could be temporarily unavailable or too busy. Try again in a few moments.
- If you are unable to load any pages, check your computer's network connection.
- If your computer or network is protected by a firewall or proxy, make sure that Firefox is permitted to access the Web.

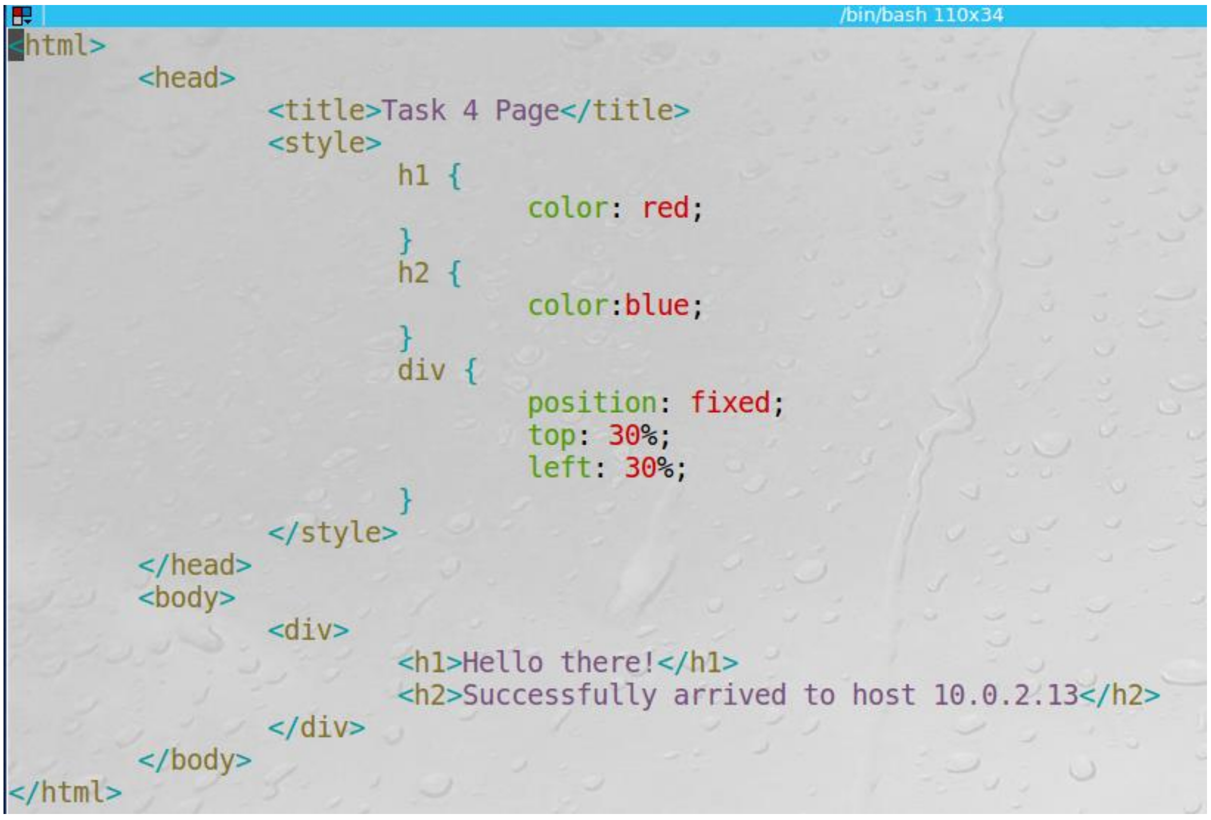
I created SSH Reversed Path from 10.0.2.13 to 10.0.2.12 on port 8000 accessing localhost (10.0.2.13) on port 80.



```
[Mon May 31|03:55@Lab_Server]:~$ ssh -R 8000:localhost:80 seed@10.0.2.12
seed@10.0.2.12'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

0 packages can be updated.
0 updates are security updates.

Last login: Mon May 31 03:36:18 2021 from 10.0.2.13
[Mon May 31|03:56@Lab_Client]:~$ 
```

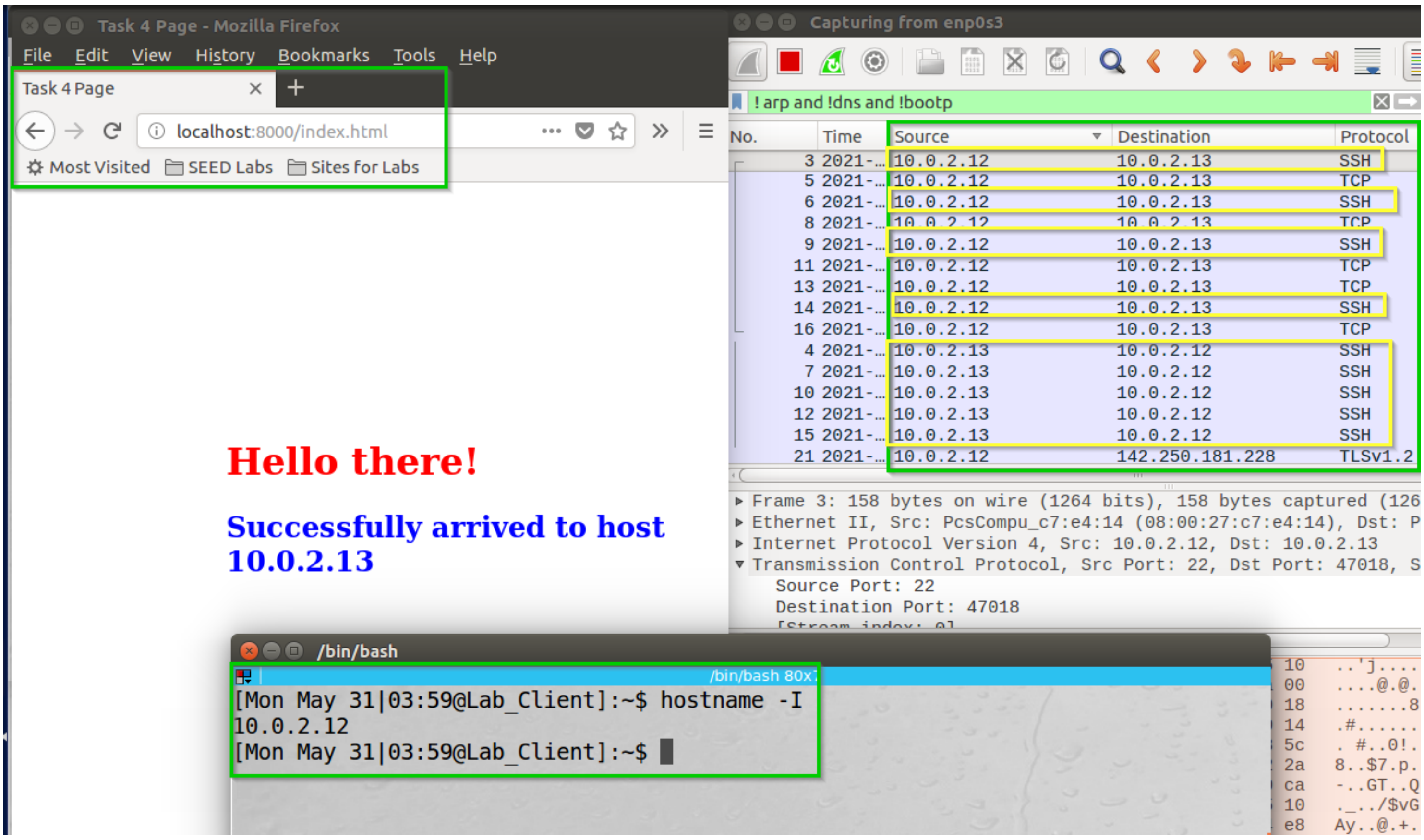
The website is hosted under /var/www/html as index.html

I changed the content of the file.



```
<html>
        <head>
                <title>Task 4 Page</title>
                <style>
                        h1 {
                                color: red;
                        }
                        h2 {
                                color:blue;
                        }
                        div {
                                position: fixed;
                                top: 30%;
                                left: 30%;
                        }
                </style>
        </head>
        <body>
                <div>
                        <h1>Hello there!</h1>
                        <h2>Successfully arrived to host 10.0.2.13</h2>
                </div>
        </body>
</html>
```

Then I browsed to localhost:8000/index.html from 10.0.2.12 and got to the 10.0.2.13 website.


Task 4 Page - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Task 4 Page
localhost:8000/index.html
Most Visited
SEED Labs
Sites for Labs
Capturing from enp0s3
! arp and !dns and !bootp
/bin/bash
/bin/bash 80x

Hello there!

Successfully arrived to host 10.0.2.13

No.	Time	Source	Destination	Protocol
3	2021-…	10.0.2.12	10.0.2.13	SSH
5	2021-…	10.0.2.12	10.0.2.13	TCP
6	2021-…	10.0.2.12	10.0.2.13	SSH
8	2021-…	10.0.2.12	10.0.2.13	TCP
9	2021-…	10.0.2.12	10.0.2.13	SSH
11	2021-…	10.0.2.12	10.0.2.13	TCP
13	2021-…	10.0.2.12	10.0.2.13	TCP
14	2021-…	10.0.2.12	10.0.2.13	SSH
16	2021-…	10.0.2.12	10.0.2.13	TCP
4	2021-…	10.0.2.13	10.0.2.12	SSH
7	2021-…	10.0.2.13	10.0.2.12	SSH
10	2021-…	10.0.2.13	10.0.2.12	SSH
12	2021-…	10.0.2.13	10.0.2.12	SSH
15	2021-…	10.0.2.13	10.0.2.12	SSH
21	2021-…	10.0.2.12	142.250.181.228	TLSv1.2

```
▸ Frame 3: 158 bytes on wire (1264 bits), 158 bytes captured (126
▸ Ethernet II, Src: PcsCompu_c7:e4:14 (08:00:27:c7:e4:14), Dst: P
▸ Internet Protocol Version 4, Src: 10.0.2.12, Dst: 10.0.2.13
▾ Transmission Control Protocol, Src Port: 22, Dst Port: 47018, S
    Source Port: 22
    Destination Port: 47018
```

```
10   ..'j....
00   ....@.@.
18   .......8
14   .#......
5c   . #..0!.
2a   8..$7.p.
ca   -..GT..Q
10   ._../$vG
e8   Ay..@.+.
```

```
[Mon May 31|03:59@Lab_Client]:~$ hostname -I
10.0.2.12
[Mon May 31|03:59@Lab_Client]:~$
```

We can see I’m on 10.0.2.12 and got to the 10.0.2.13 website by browsing to localhost 8000.
We can also see the behavior of our connection on Wireshark - only SSH between 10.0.2.12 and 10.0.2.13

Task 4 summary

- I blocked port 20,80,443 from A to B.
  Then we tunneled using SSH Reverse Path from B to A:
  ‘ssh -R 8000:localhost:80 seed@10.0.2.12’
- I edited the index.html file on 10.0.2.13 and browsed from 10.0.2.12 to 10.0.2.13:8000/index.html and was able to reach the website.

Lab Summary

Task 1: I have learned using iptables as our Firewall.

I opened Telnet and SSH connection between A and B and Utah University website. Then, applied iptables rule to block the communication.

Task 2: I have learned about LKM – Loadable Kernel Modules which are modules we can insert to our kernel on the fly.

I have learned about Netfilter – which allows me to load the LKM and call hooks on the operating system – Meaning we catch the system calls before the kernel decides what to do with it and forward it to the operating system.

I have learned about Makefiles and how to build my code and then use these files and insert modules to the Linux kernel.

I implemented hooks to catch SSH, Telnet and connection to Utah University website and decide what to do with it.

Task 3: I blocked outgoing Telnet from 10.0.2.12, then executed 'ssh -L 8000:10.0.2.11:23 seed@10.0.2.13' to create a tunnel from 10.0.2.12 to 10.0.2.11 through 10.0.2.13.

I blocked ports 80 and 443 to Utah University website, the executed 'ssh -D 9000 seed@10.0.2.13' and configured Firefox proxy through port 9000
Then connected to 9000 port to SSH and could browse to Utah University website.

Task 4: I blocked port 20,80,443 from A to B.
Then, tunneled using SSH Reverse Path from B to A:
'ssh -R 8000:localhost:80 seed@10.0.2.12'

I edited the index.html file on 10.0.2.13 and browsed from 10.0.2.12 to 10.0.2.13:8000/index.html and was able to reach the website.