Firewalls

CSE644

Internet Security

Chirag Sachdev

680231131

Homework 3

Task 1:

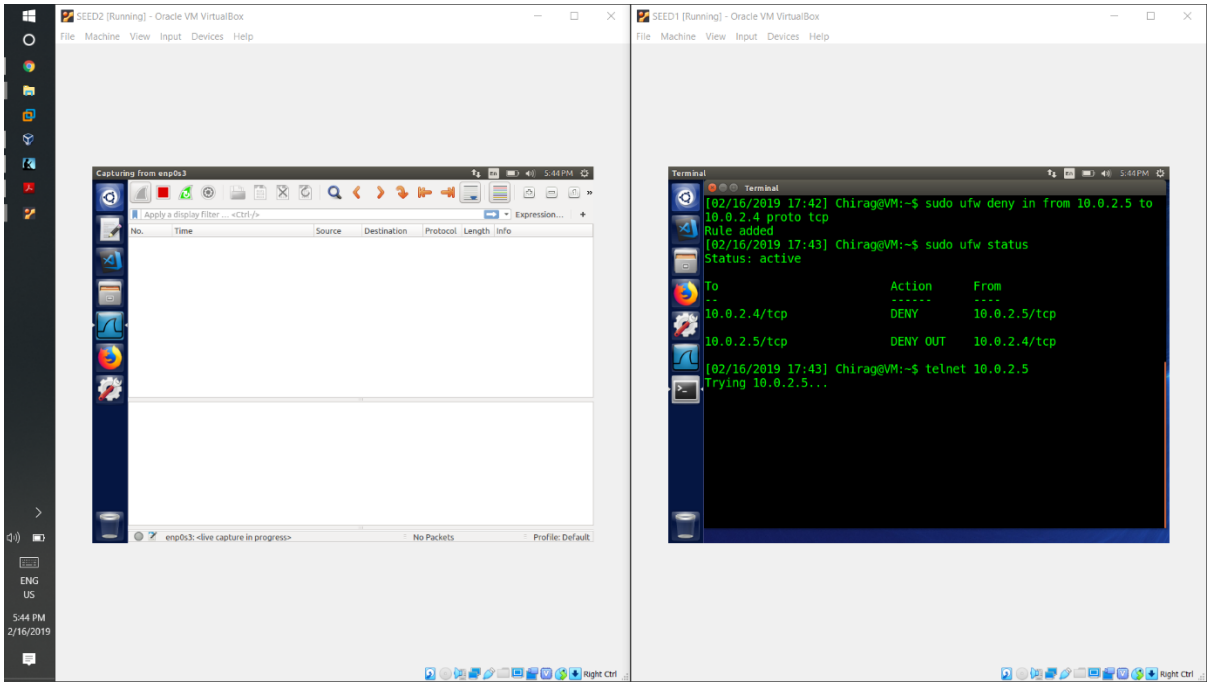
Setting up firewall policies using ufw.

Host A is set up at 10.0.2.4 which runs the firewall. Host B is the second VM which has to be connected to the telnet.

Code:

ufw deny out from 10.0.2.4 to 10.0.2.5 proto tcp

Output:



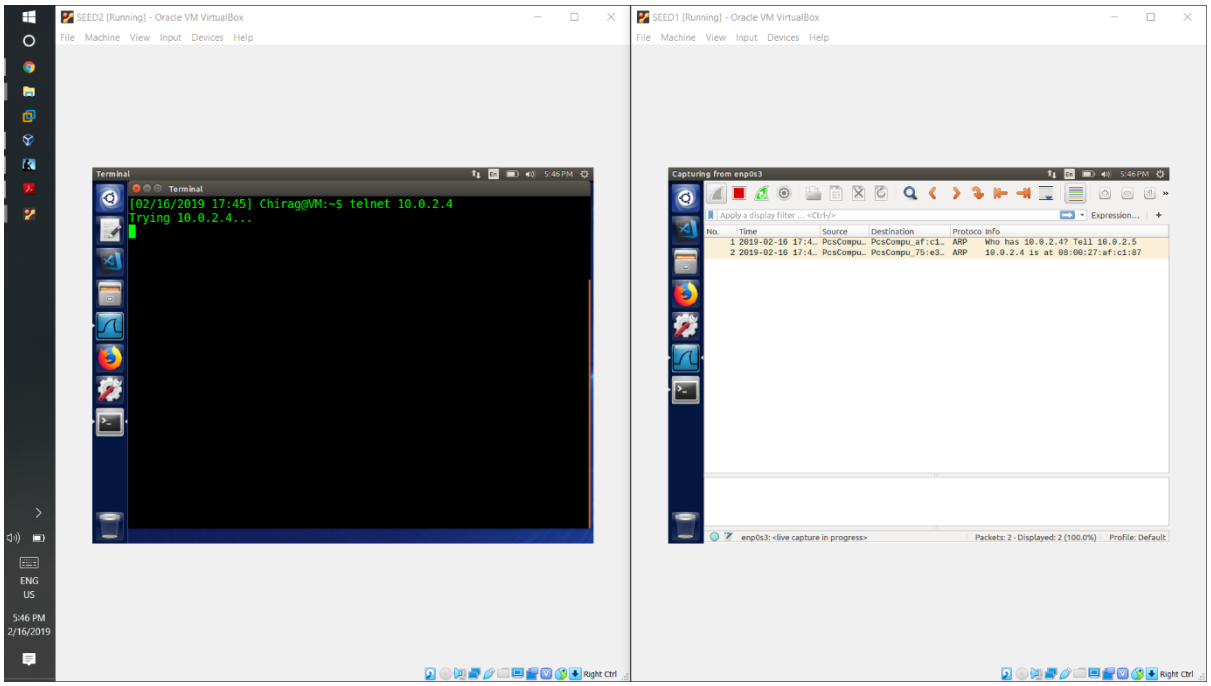
Observation:

After enabling a rule which denies tcp traffic from A to B, we see that the telnet connection gets stuck and hence is blocked. Machine B runs wireshark, which does not receive any packets.

To deny an incoming connection from B to A we can use the command

ufw deny from 10.0.2.5 to 10.0.2.4 proto tcp

Observation:



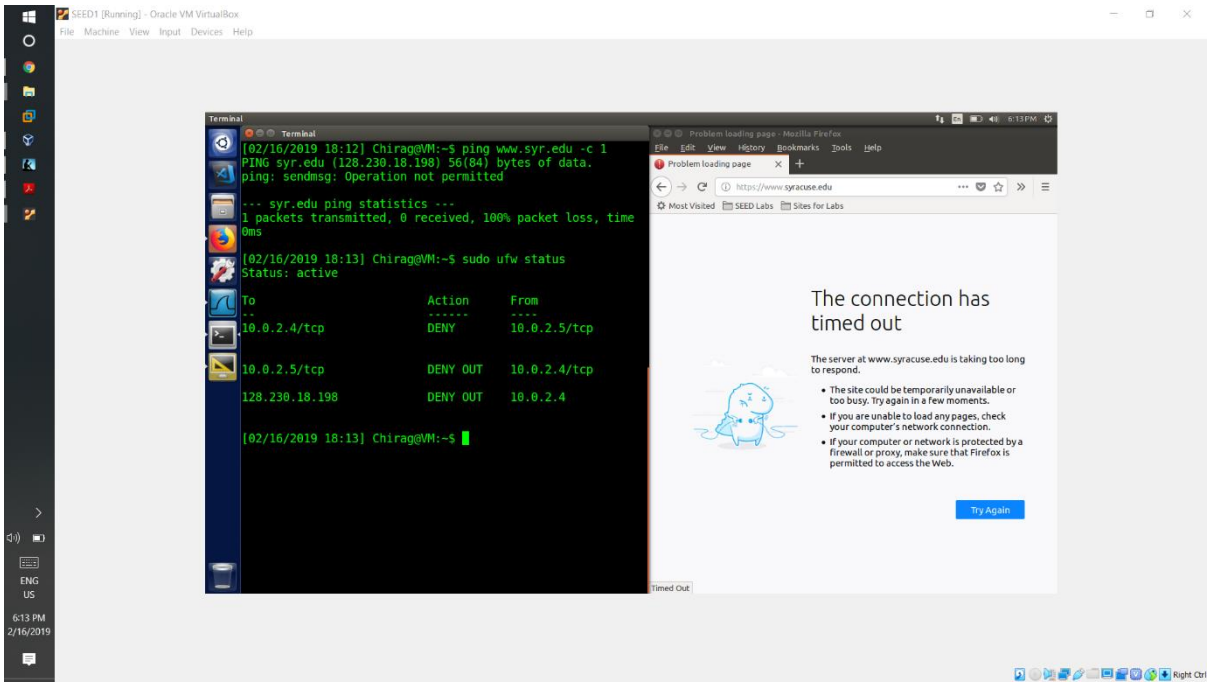
After enabling this policy, we try to do a telnet from B to A. We run wireshark on A to capture packets. We can see that by trying a telnet from B to A and see that no packets are captured due to the firewall policies.

Blocking A website.

For the purpose of this task, I chose to block ww.syr.edu(130.230.18.198). We can block the website using:

ufw deny out to 130.230.18.198

Output:



Observation:

After enabling the filter, access to www.syr.edu cannot be established.

Task 2

Coding a simple firewall program by attaching hooks to the kernel using netfilter modules.

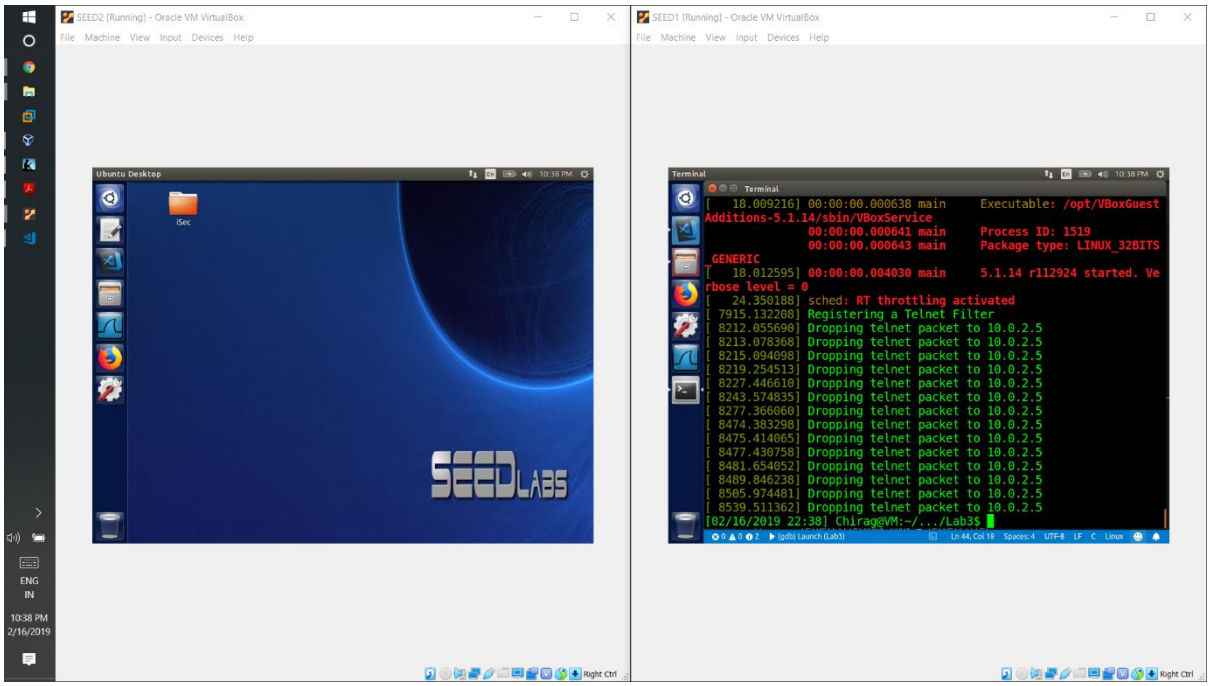
With the following code we can establish the Filters as seen in task 1.

Code:

```
#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>


static struct nf_hook_ops telnetFilterHook;

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

    iph=ip_hdr(skb);
    tcph = (void *)iph+iph->ihl*4;
    if(iph->protocol == IPPROTO_TCP)
    {
        if (tcph->dest == htons(23))
        {
            printk(KERN_INFO "Dropping telnet packet to %d.%d.%d.%d\n",
                ((unsigned char *)&iph->daddr)[0],
                ((unsigned char *)&iph->daddr)[1],
                ((unsigned char *)&iph->daddr)[2],
                ((unsigned char *)&iph->daddr)[3]);
            return NF_DROP;
        }

        else if (tcph->source == htons(23))
        {
            printk(KERN_INFO "Dropping telnet packet from
%d.%d.%d.%d\n",
                ((unsigned char *)&iph->saddr)[0],
                ((unsigned char *)&iph->saddr)[1],
                ((unsigned char *)&iph->saddr)[2],
                ((unsigned char *)&iph->saddr)[3]);
            return NF_DROP;
        }
    }
    else if (((((unsigned char *)&iph->daddr)[0] == 128)  &&
```

```
              (((unsigned char *)&iph->daddr)[1] == 230)  &&
              (((unsigned char *)&iph->daddr)[2] == 18)   &&
              (((unsigned char *)&iph->daddr)[3] == 198)) &&
              (iph->protocol == IPPROTO_TCP ||
                    iph->protocol == IPPROTO_UDP))
    {
      printk(KERN_INFO "Website at %d.%d.%d.%d is blocked\n",
        ((unsigned char *)&iph->daddr)[0],
        ((unsigned char *)&iph->daddr)[1],
        ((unsigned char *)&iph->daddr)[2],
        ((unsigned char *)&iph->daddr)[3]);
        return NF_DROP;
    }
    else
    {
        return NF_ACCEPT;
    }
}


int setUpFilter(void)
{
    printk(KERN_INFO "Registering a Filter\n");
    telnetFilterHook.hook = telnetFilter;
    telnetFilterHook.hooknum = NF_INET_POST_ROUTING;
    telnetFilterHook.pf = PF_INET;
    telnetFilterHook.priority = NF_IP_PRI_FIRST;
    nf_register_hook(&telnetFilterHook);
    return 0;
}

void removeFilter(void)
{
    printk(KERN_INFO "Filter is being removed\n");
    nf_unregister_hook(&telnetFilterHook);
}

module_init(setUpFilter);
module_exit(removeFilter);

MODULE_LICENSE("GPL");
```

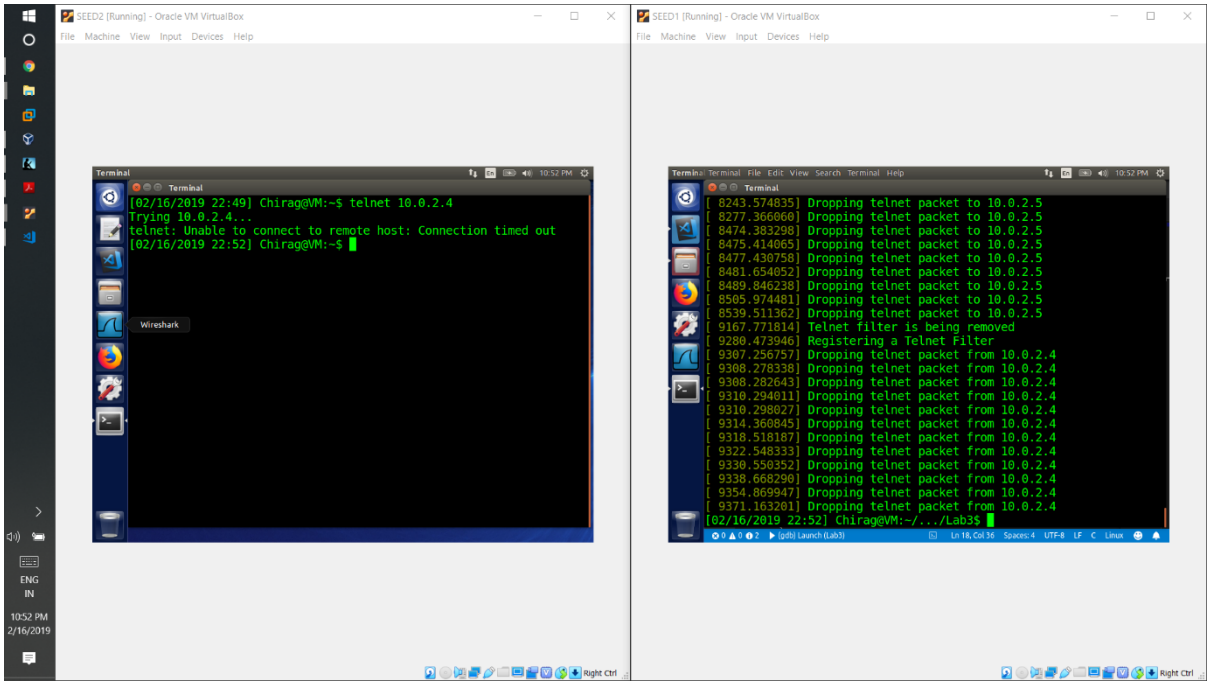
Output:



Observation:

Here we see that the telnet packets get dropped, from A to B

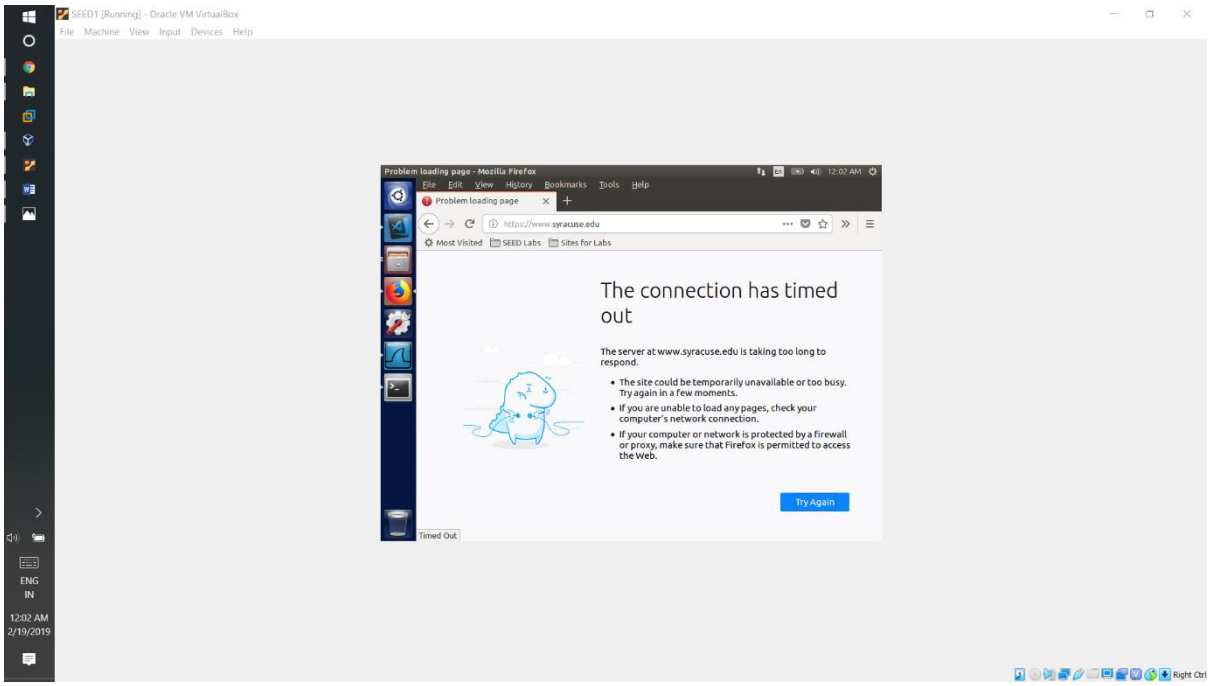
Output:



Observation:

Here we see the telnet packets get dropped, B to A.

Output:



Observation:

Here we see that www.syr.edu cannot be accessed after thr filter is hooked on.

Task 3:

Bypassing firewall by using ssh

We use the command:

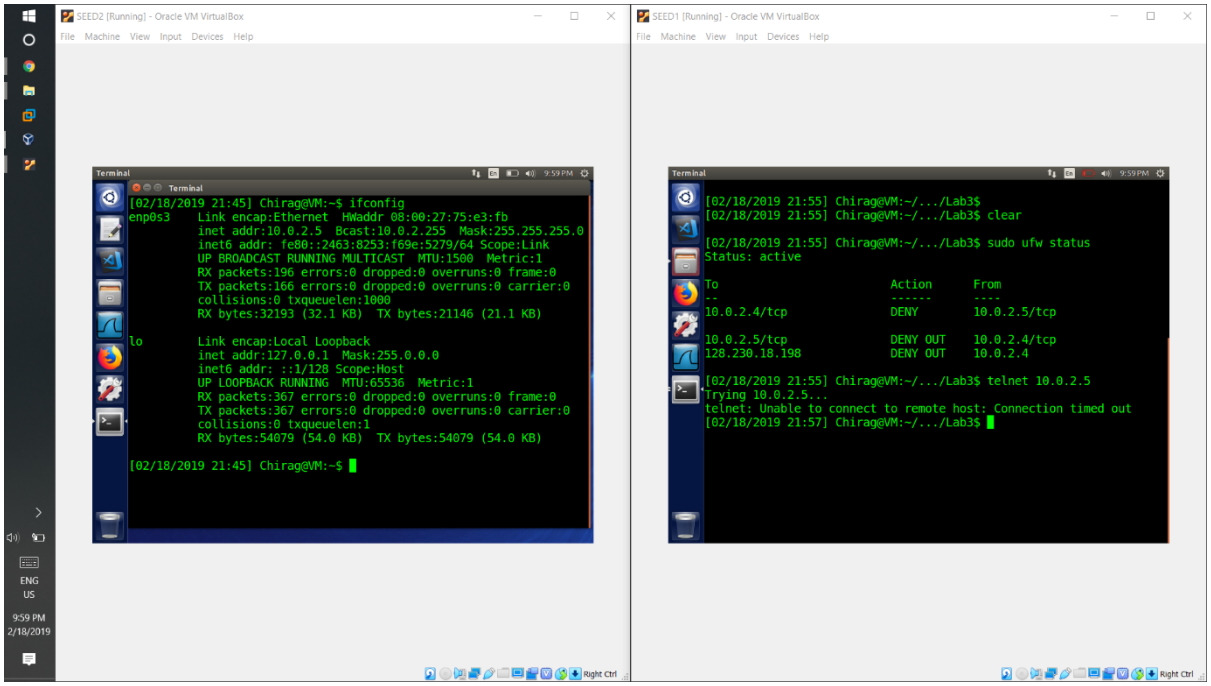
Ssh -L 8000:10.0.2.5:23 seed@10.0.2.5

To establish a ssh tunnel.

We send our packets through the localhost through the port 8000

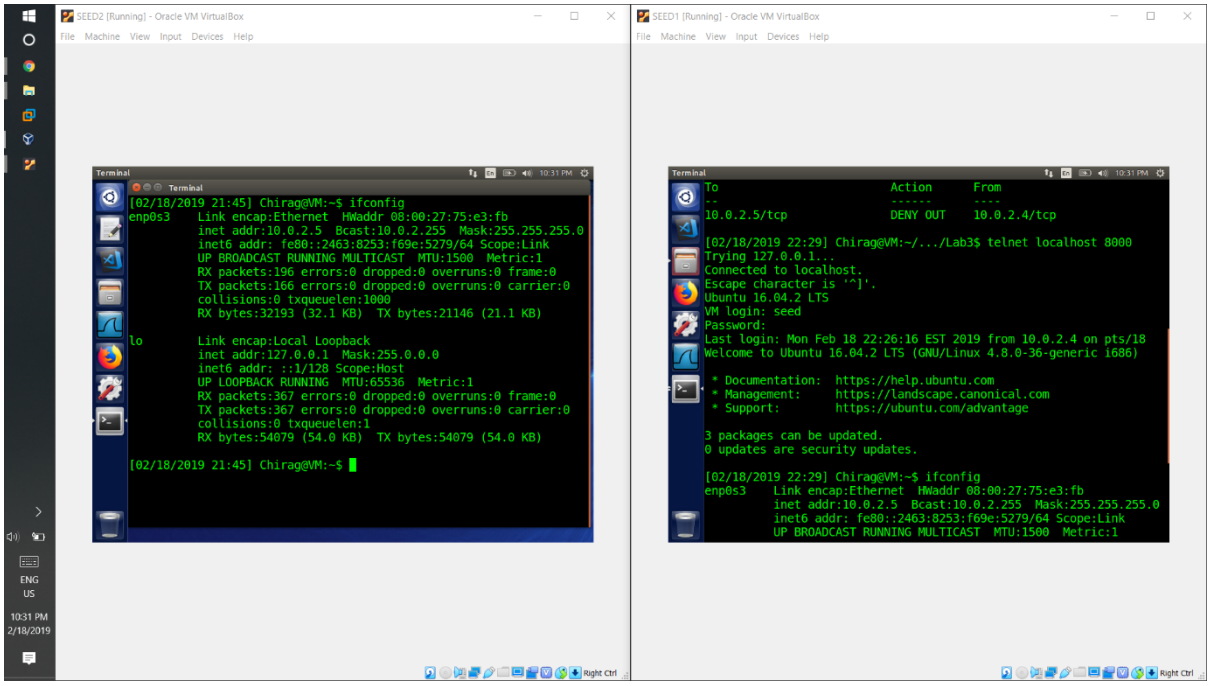
We use the command: telnet localhost 8000, to route traffic through the localhost.

Output:



Observation:

Here we see that the connection doesn't go through without enabling a tunnel using ssh.

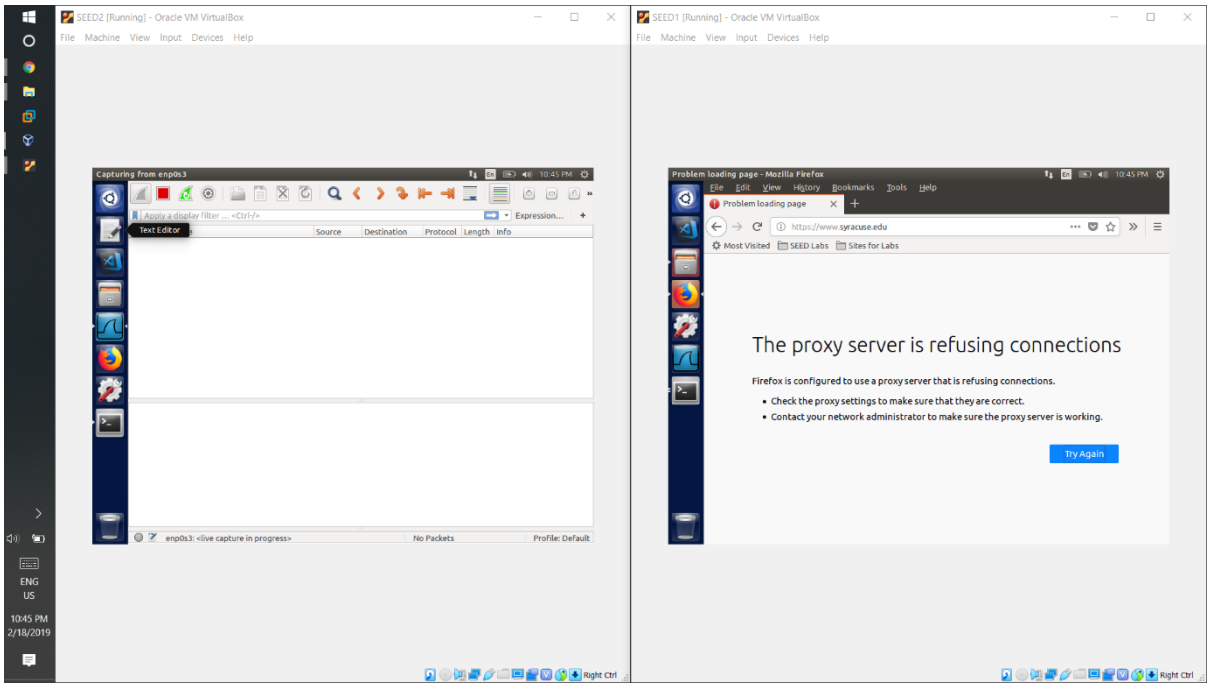


Observation: After establishing a tunnel, the connection goes through and bypasses the firewall.

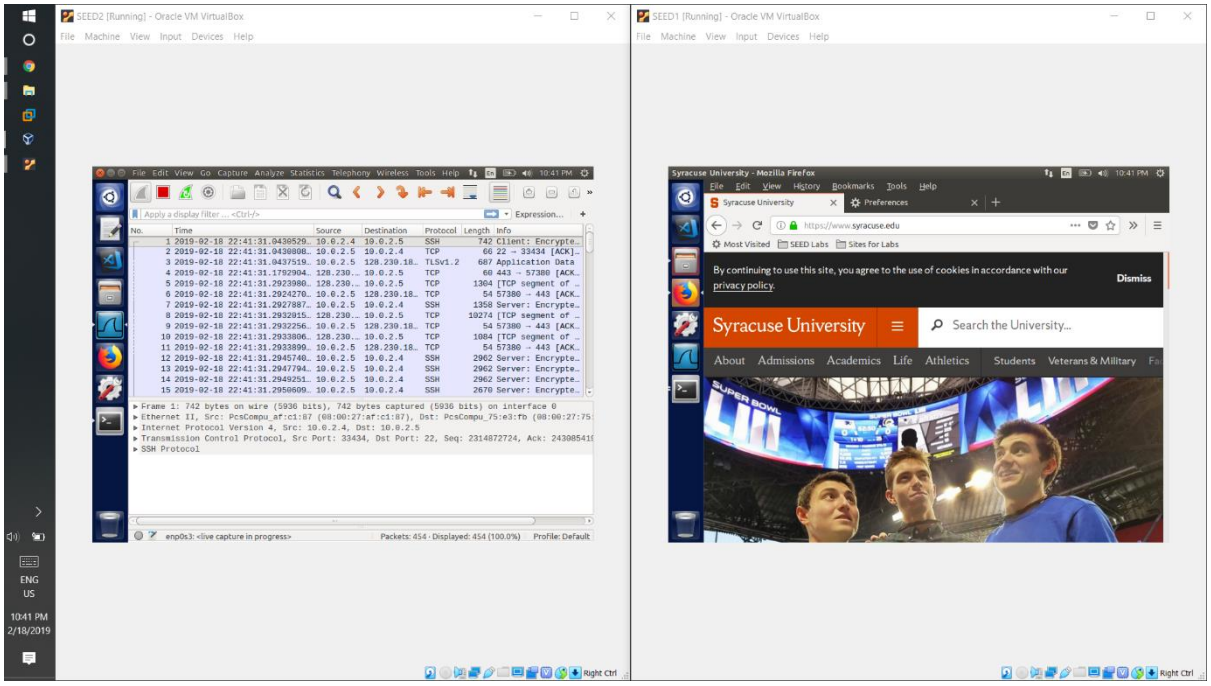
Bypassing websites using dynamic ports.

Ssh -D 9000 -C seed@10.0.2.5

After enabling the firewall but the proxy connection is not established in firefox, we get the following output:



Output:



Observation:

Here we see that the website is accessed even though the firewall denys connections. Here we can see the packets captured in wireshark route the traffic from A to the website and back. This happens when the proxy is enabled in on firefox.

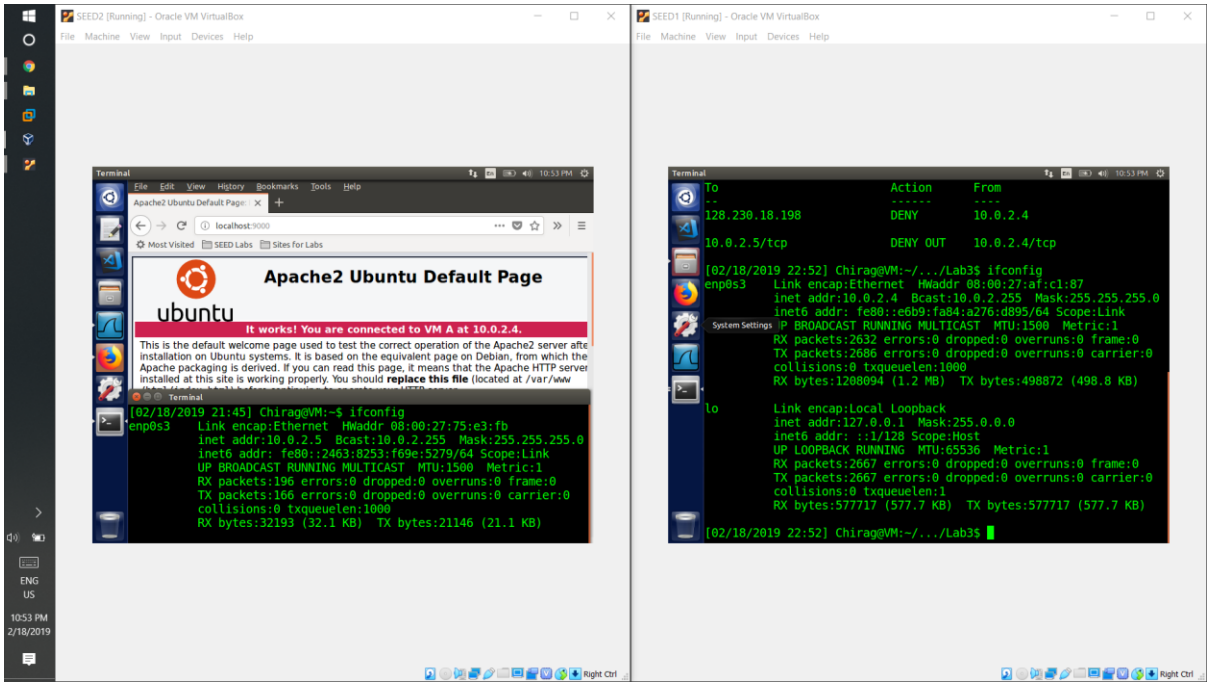
Task 4

Reverse ssh to access local files of A from B.

Command:

Ssh -R 8000:localhost:80 seed@10.0.2.5

Output:



Observation:

Here we see that the, the localhost, index files is accessed from machine B.