

New York Institute of Technology



School of Engineering and Computing Sciences

INCS 745 : Intrusion Detection and Hacker Exploits

Linux Firewall Exploration Lab

Harshita Grover (1276595)

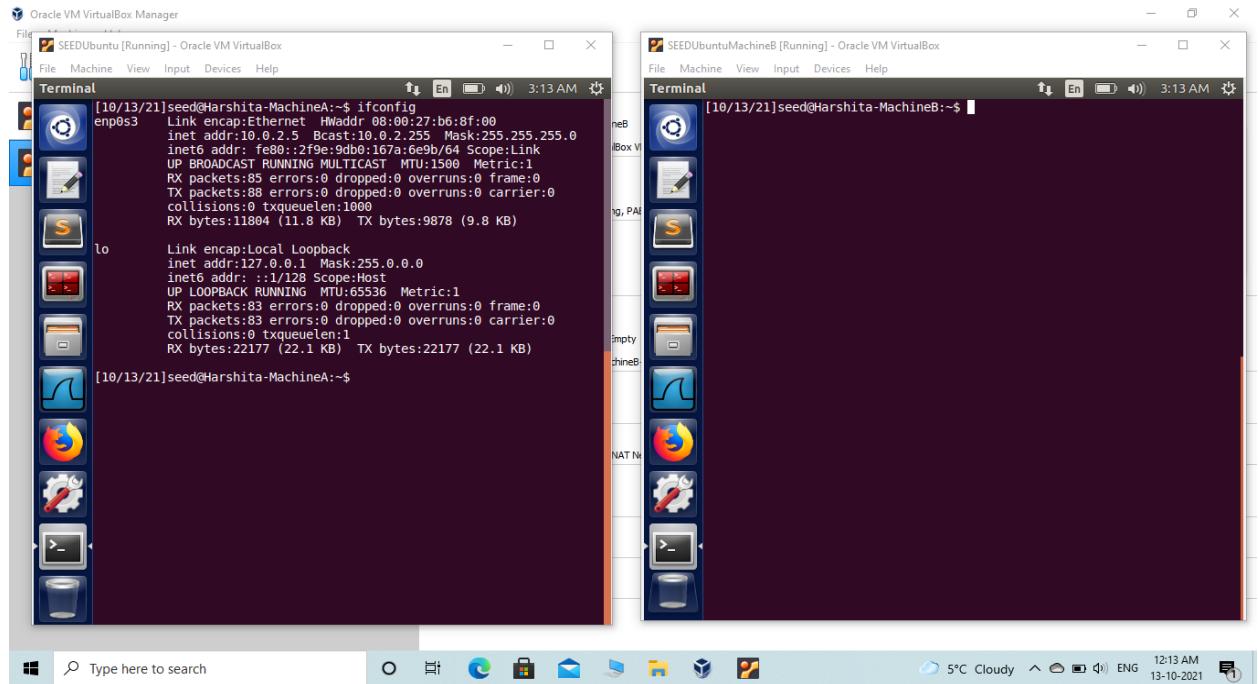
Professor: Yunlong Shao

Date Submitted:

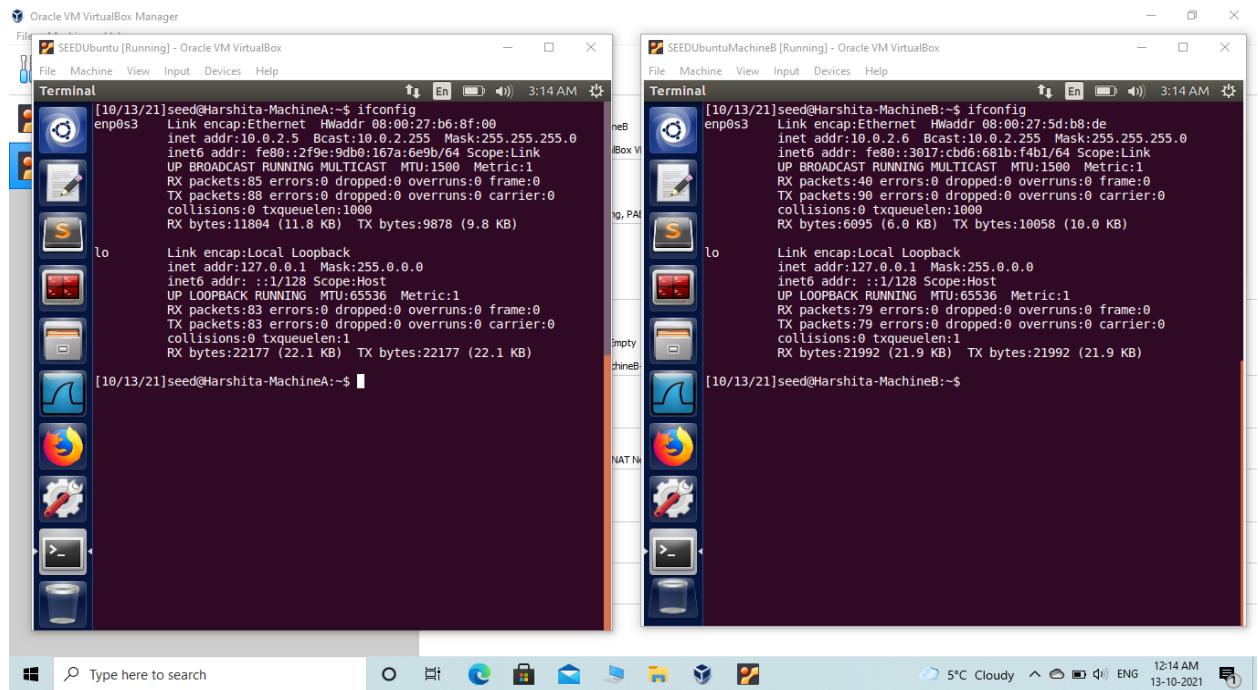
October 15, 2021

Lab 1: Linux Firewall Exploration Lab

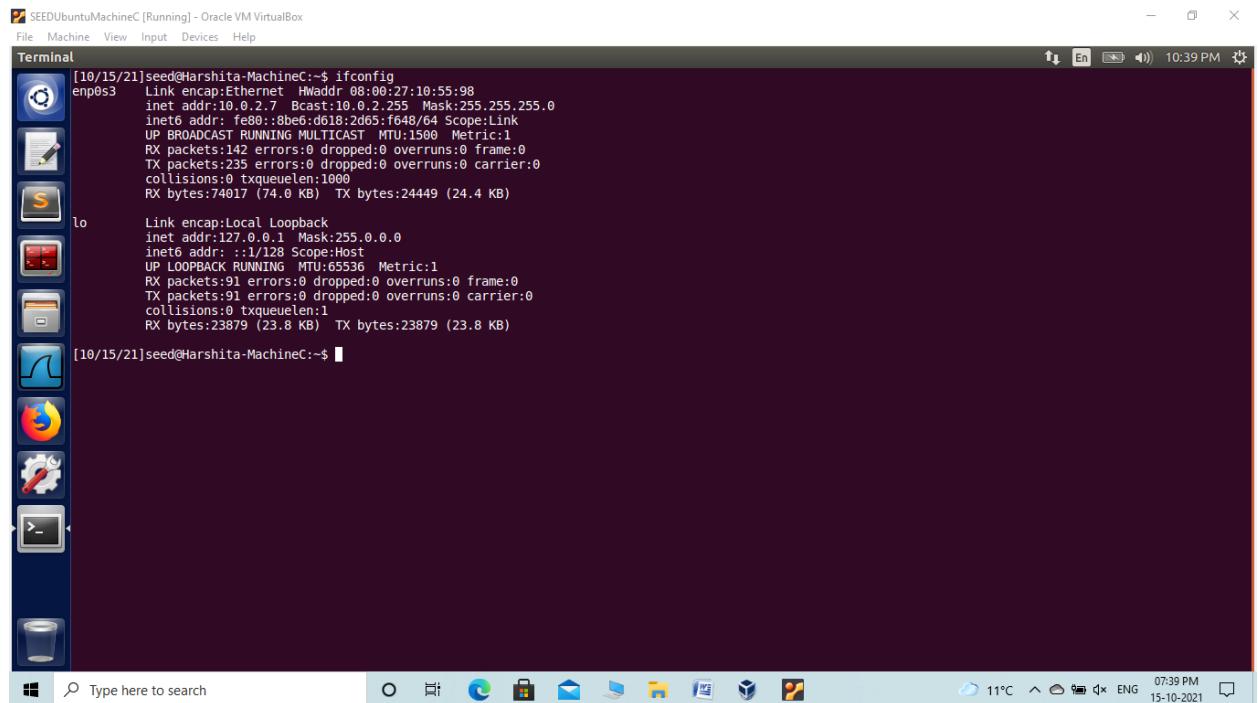
IP address of Machine A : 10.0.2.5



IP address of Machine B : 10.0.2.6



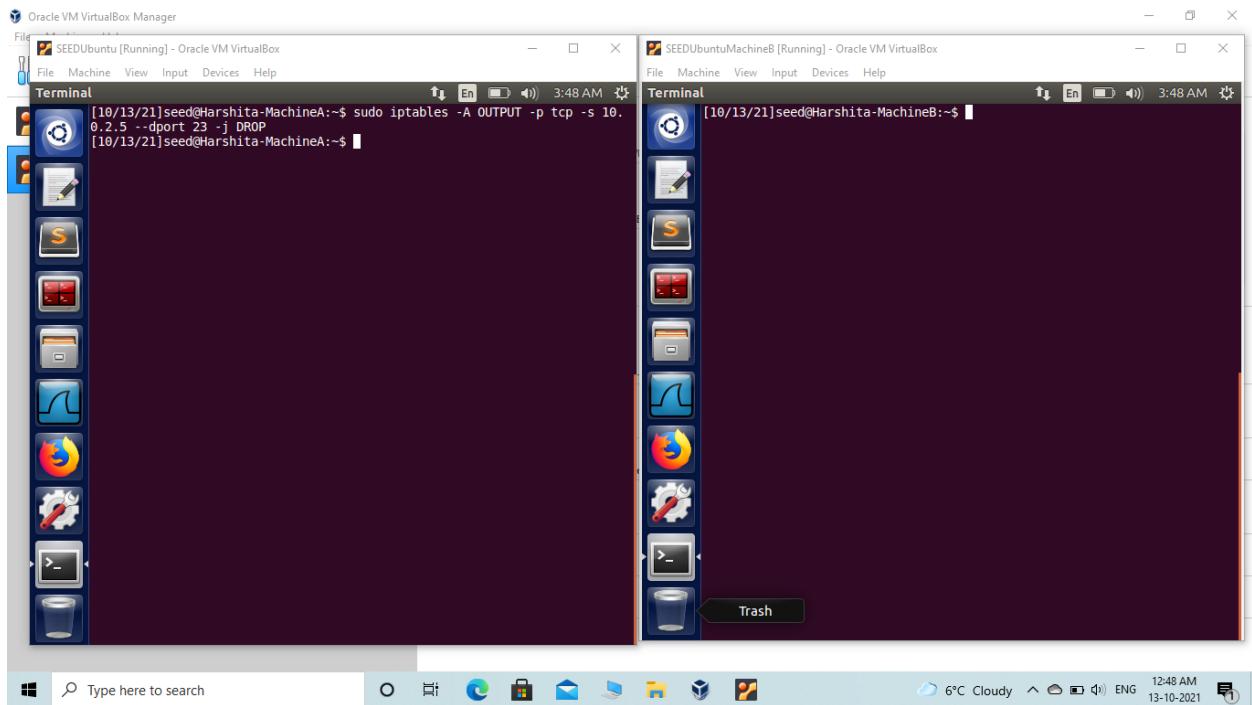
IP address of Machine C: 10.0.2.7



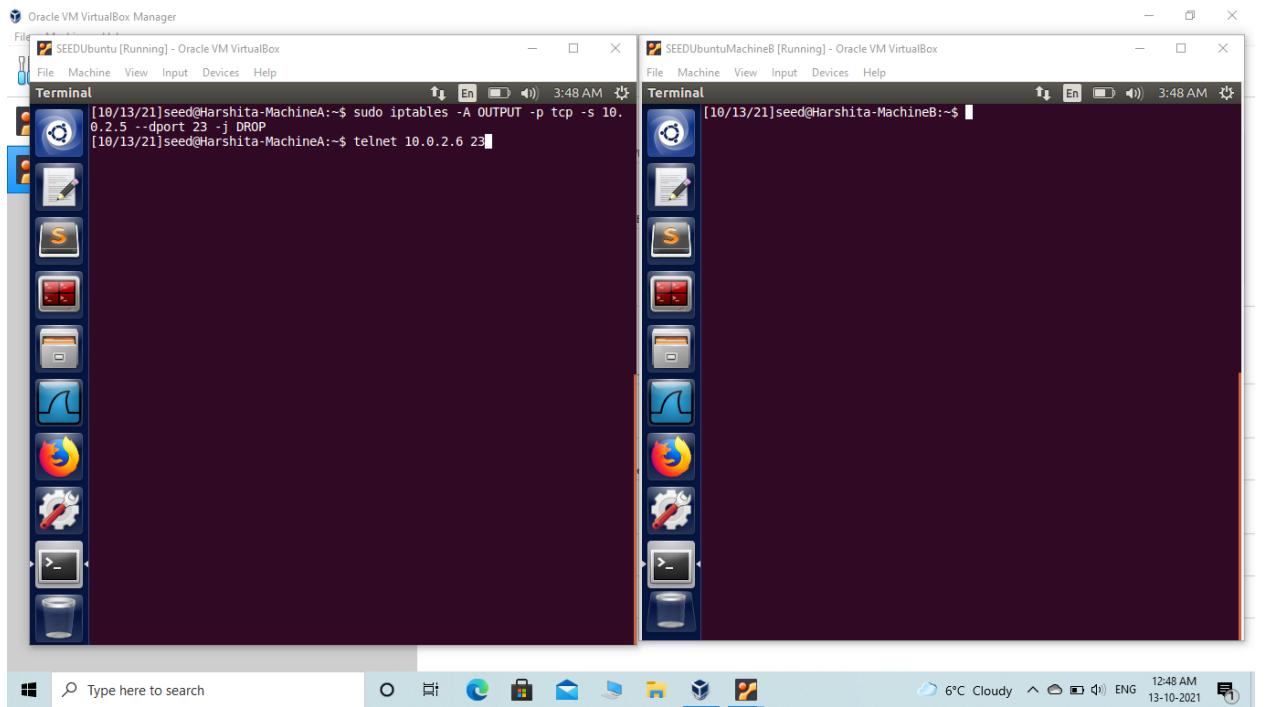
Task 1: Using Firewall

(a) Prevent Machine A from doing telnet to Machine B.

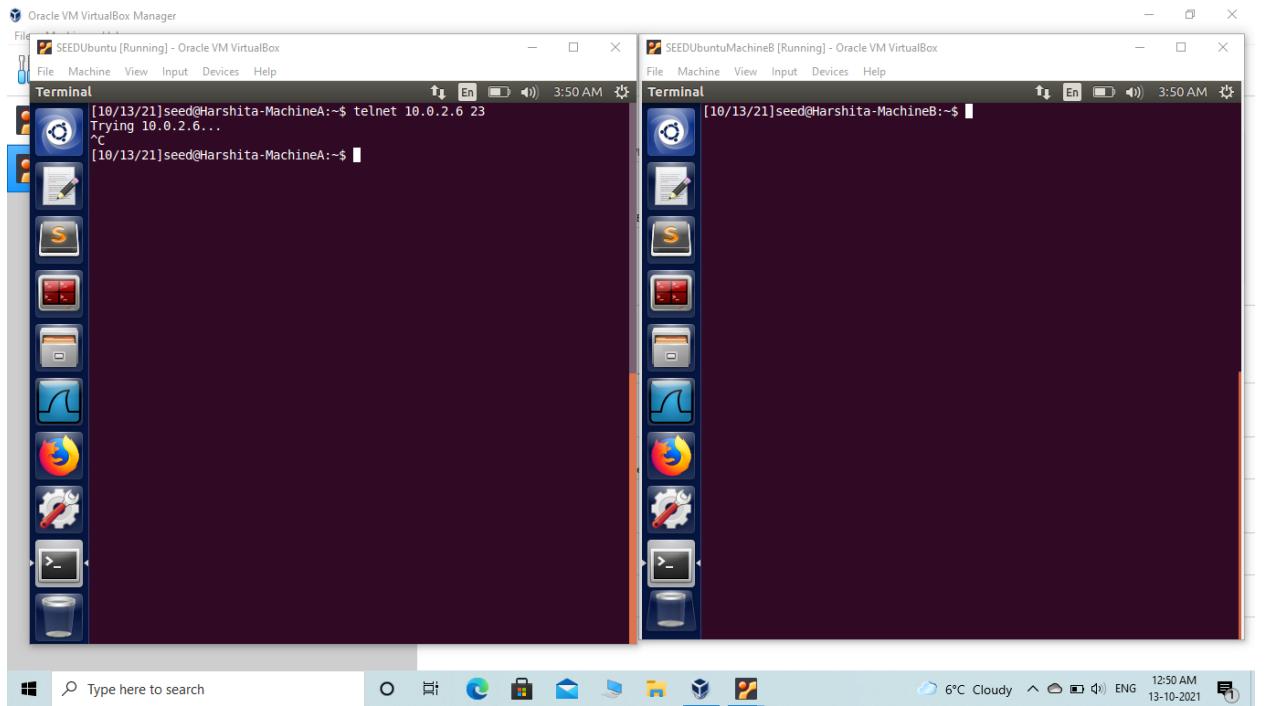
Command: sudo iptables -A OUTPUT -p tcp -s 10.0.2.5 --dport 23 -j DROP



We then try to telnet machine B as shown in the following figure.

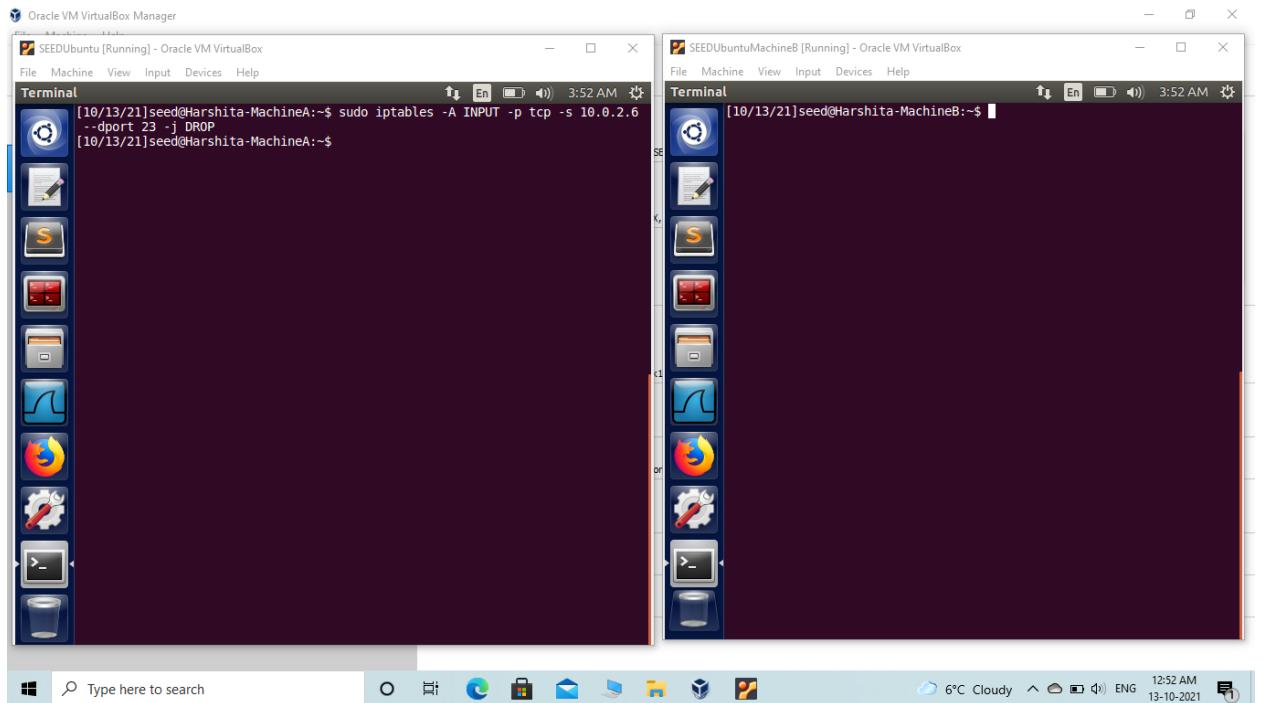


But we are not able to telnet to machine B.

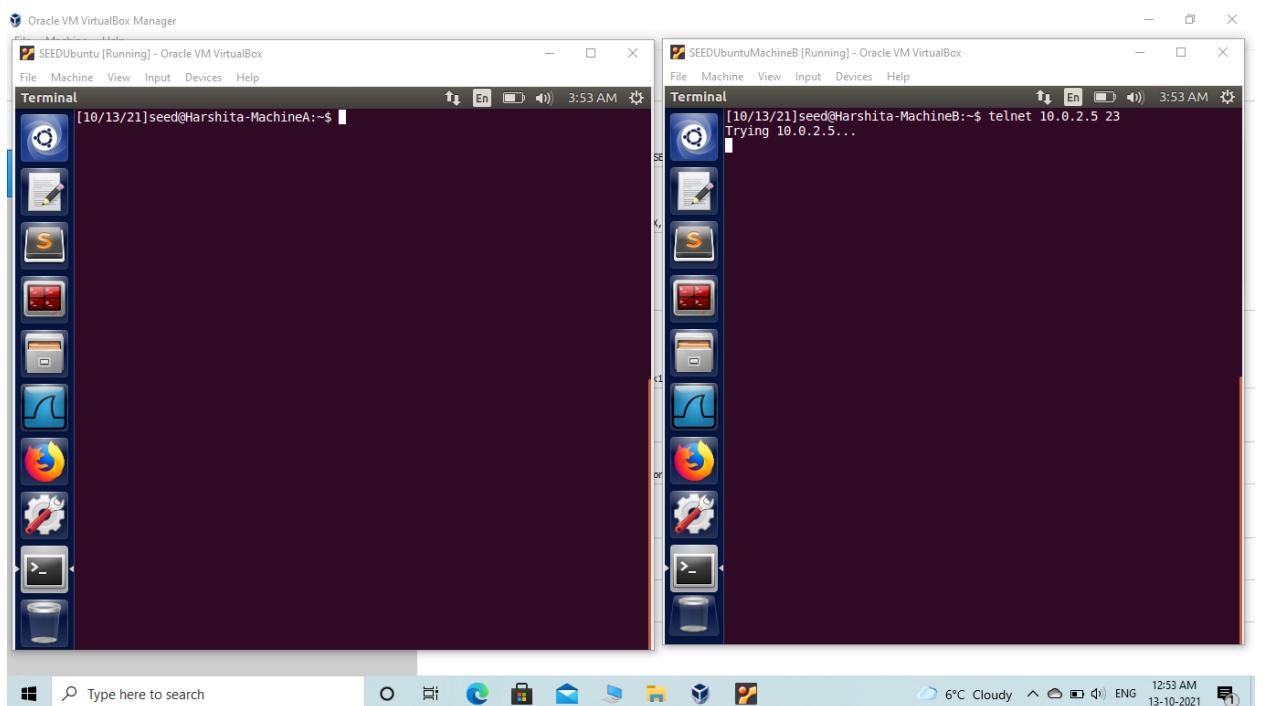


(b) Prevent Machine B from doing telnet to Machine A.

Command: sudo iptables -A INPUT -p tcp --sport 23 -j DROP

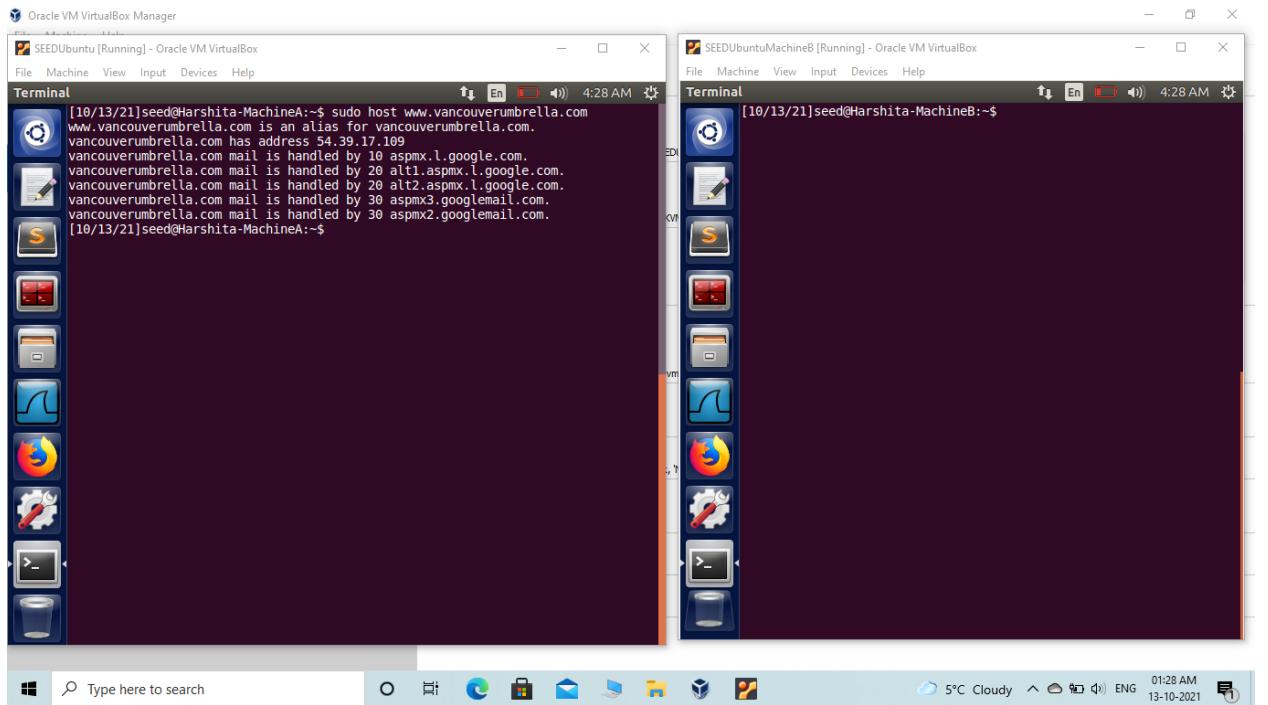


We try to telnet machine A from machine B but we are unable to telnet.

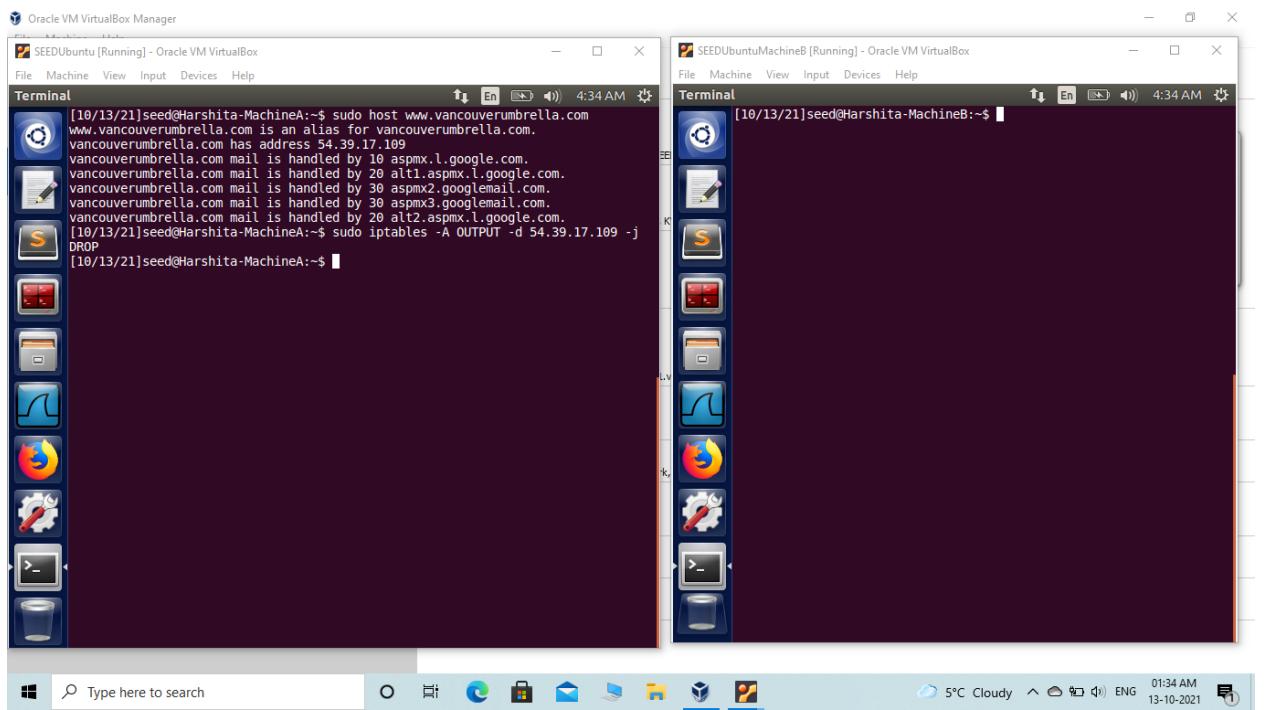


(c) Prevent A from visiting an external web site.

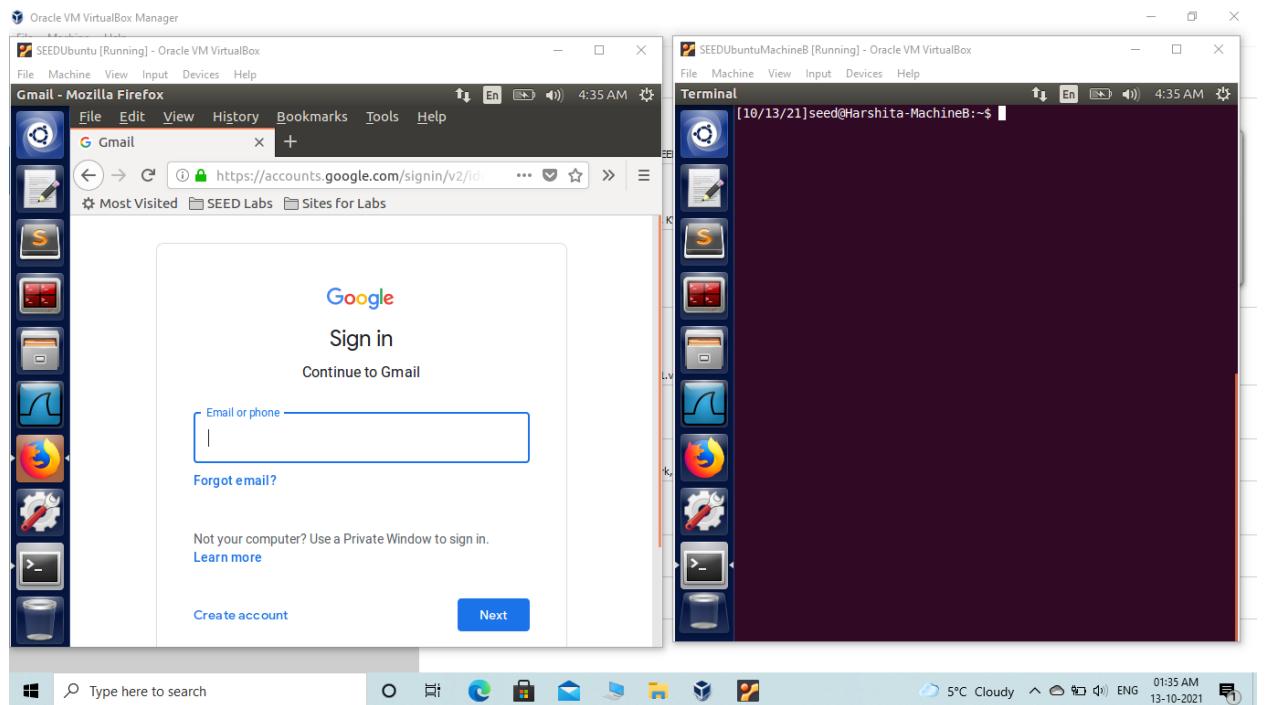
We firstly find ip address of an external website “www.vancouverumbrella.com” .



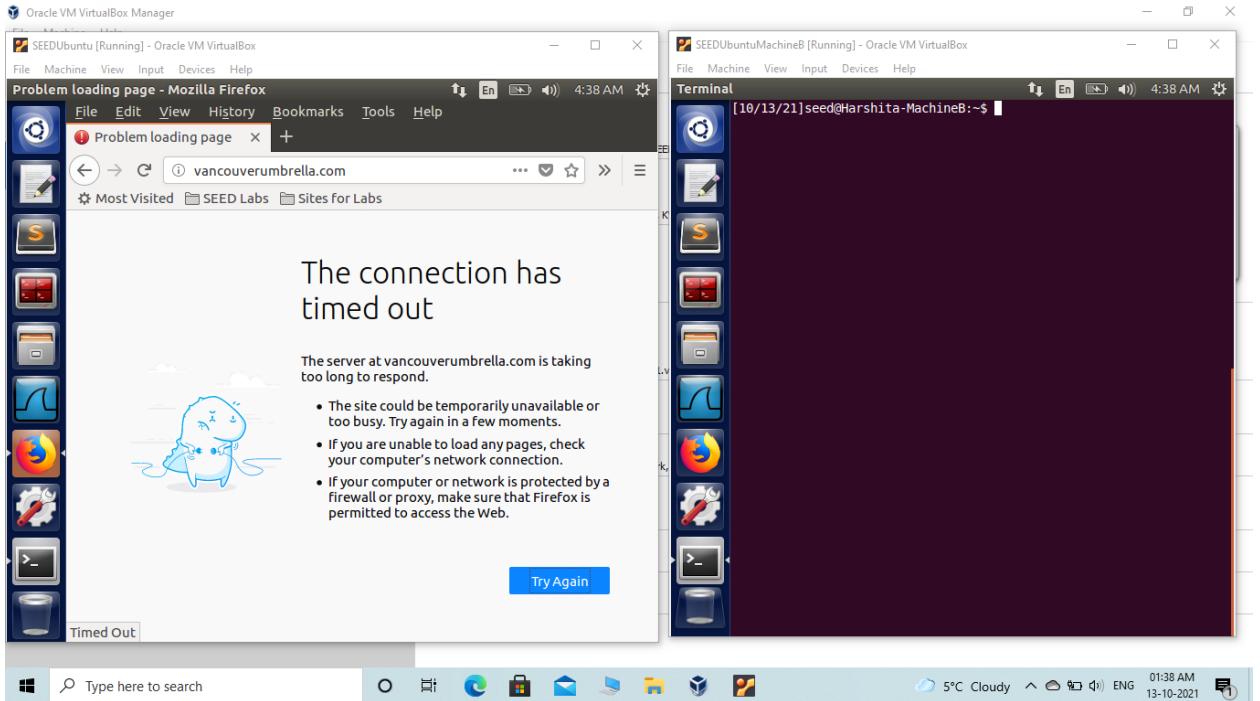
We block access of machine A to external website using iptables.



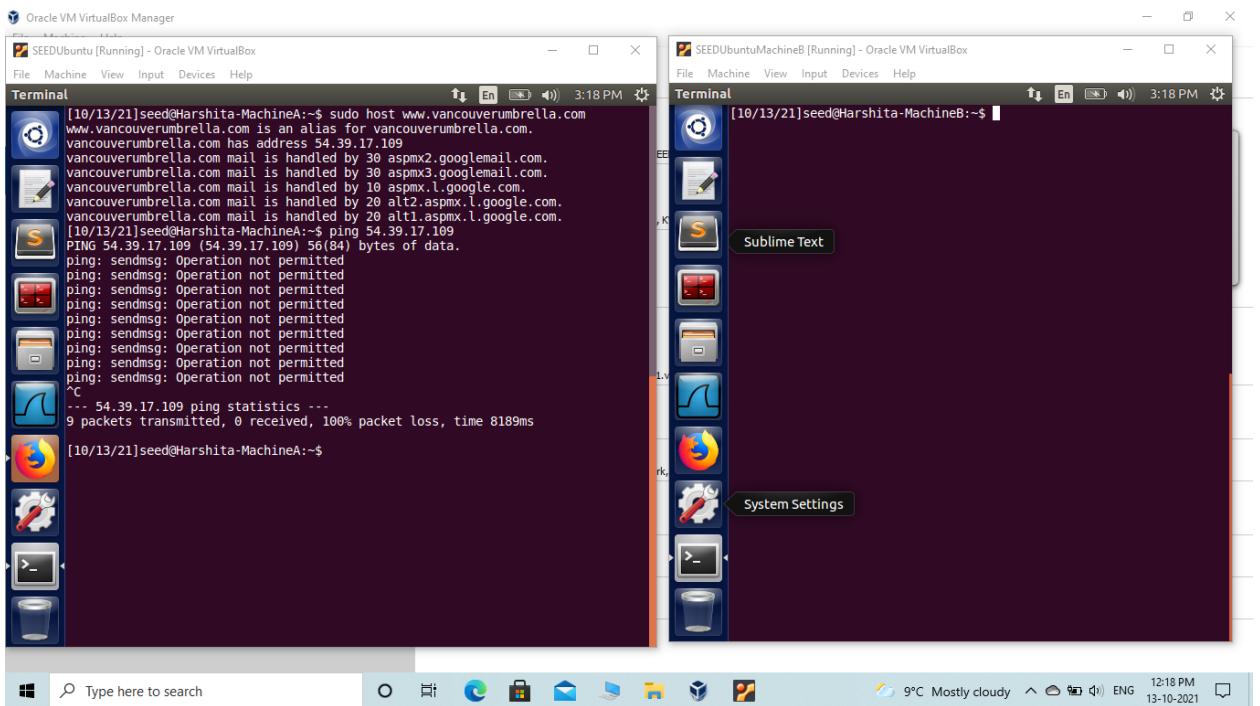
We now try to access “gmail.com” just to see that we can access any other website.



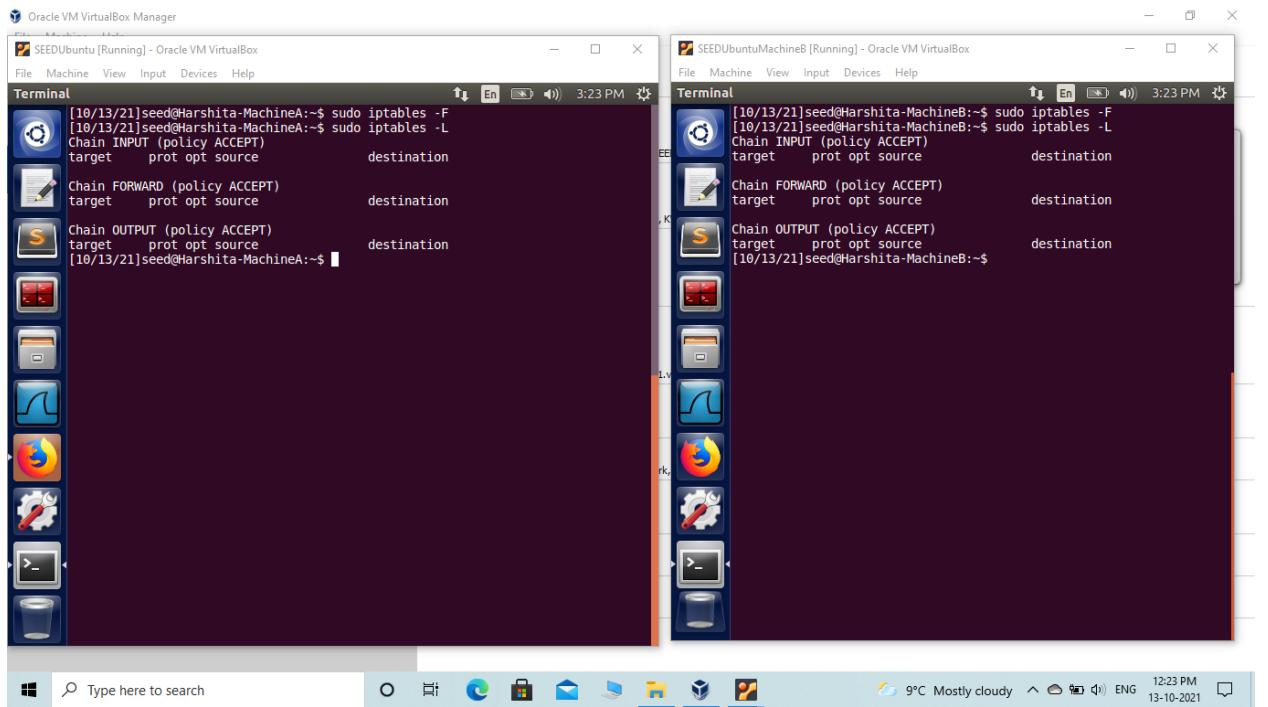
But we cannot access the external webpage that we blocked that is
“www.vancouverumbrella.com”.



Similarly, we try to ping the external website “www.vancouverumbrella.com” which is not permitted as we saw in the previous screenshots.

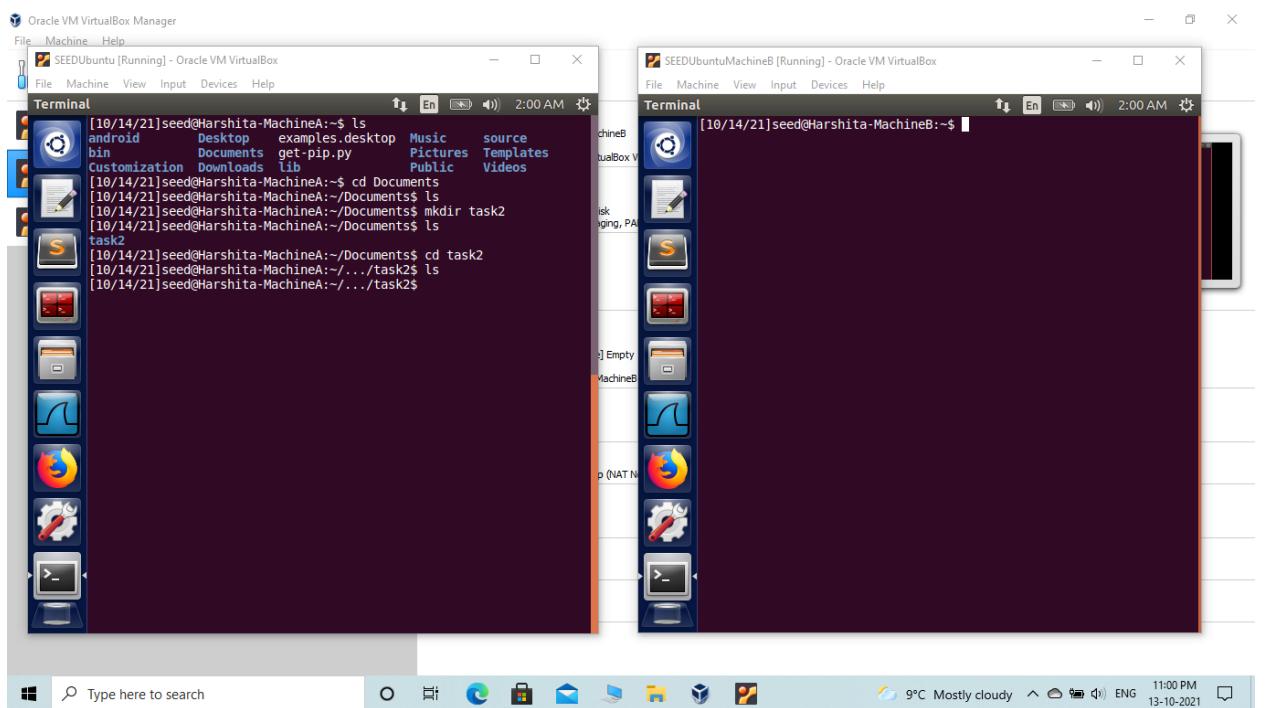


Now, we remove all the iptable rules using “iptables -F”.

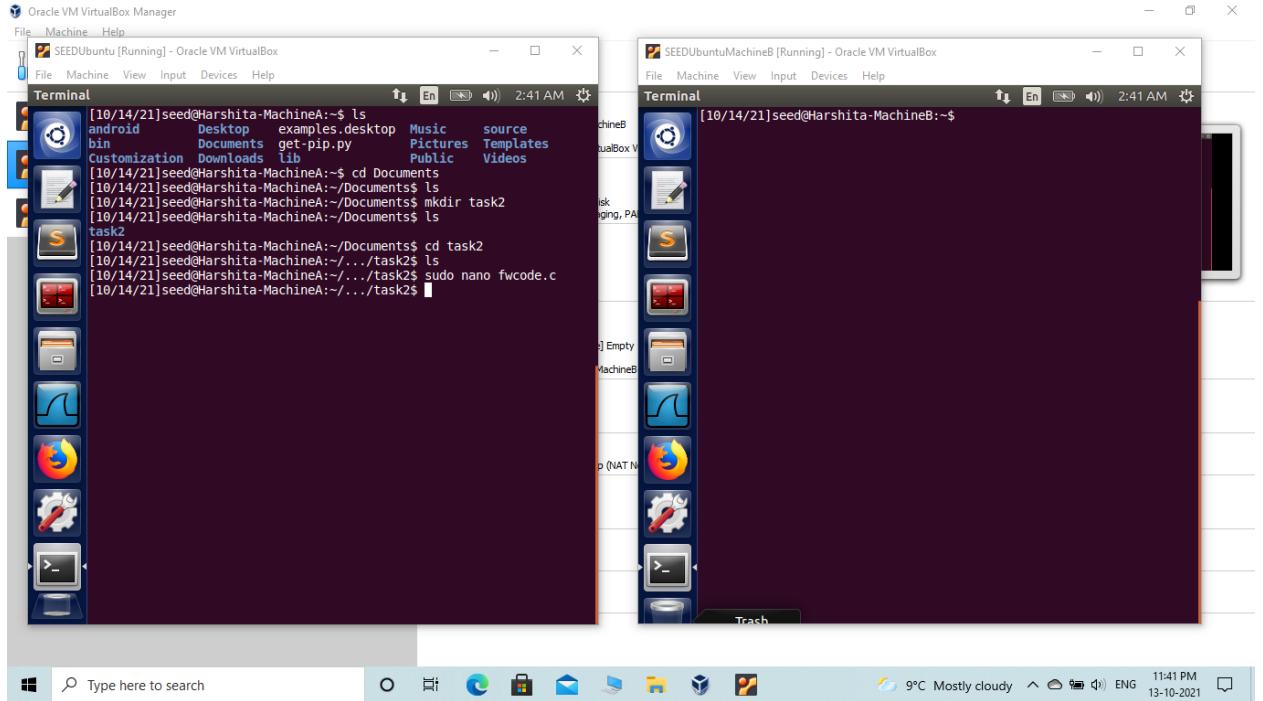


Task 2: Implementing a Simple Firewall

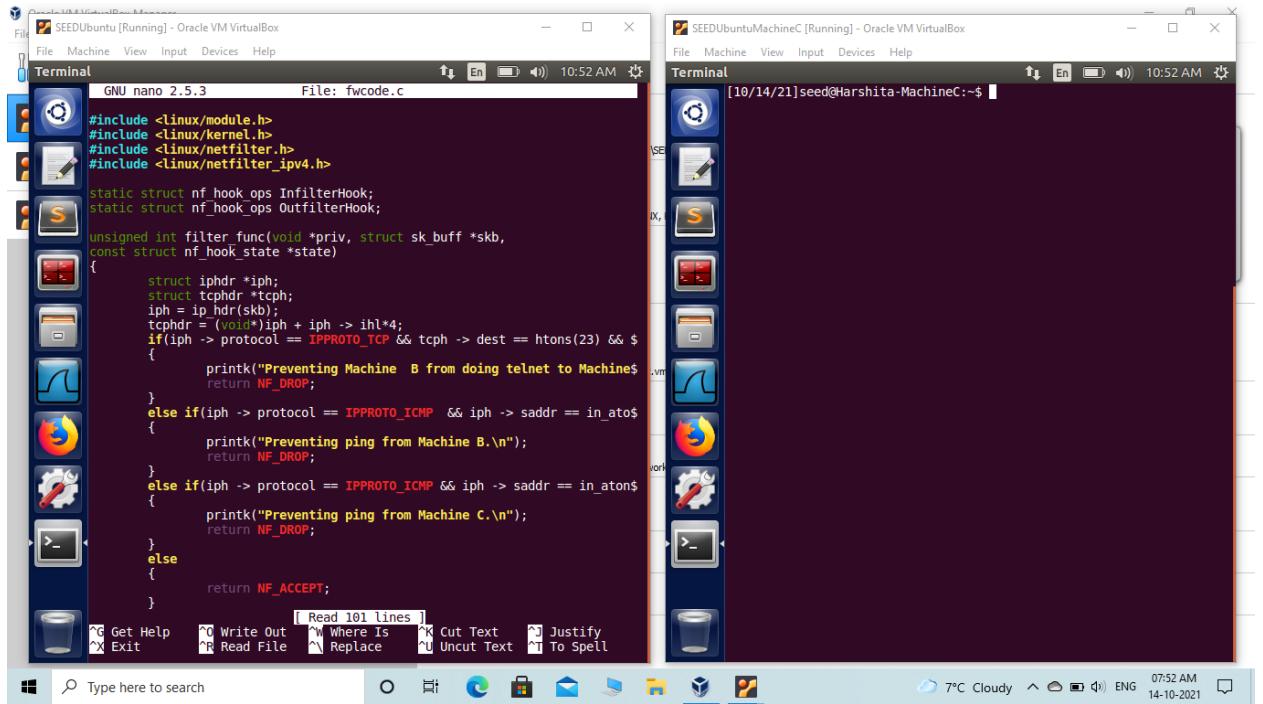
We firstly create a folder named “task2” in Documents folder.

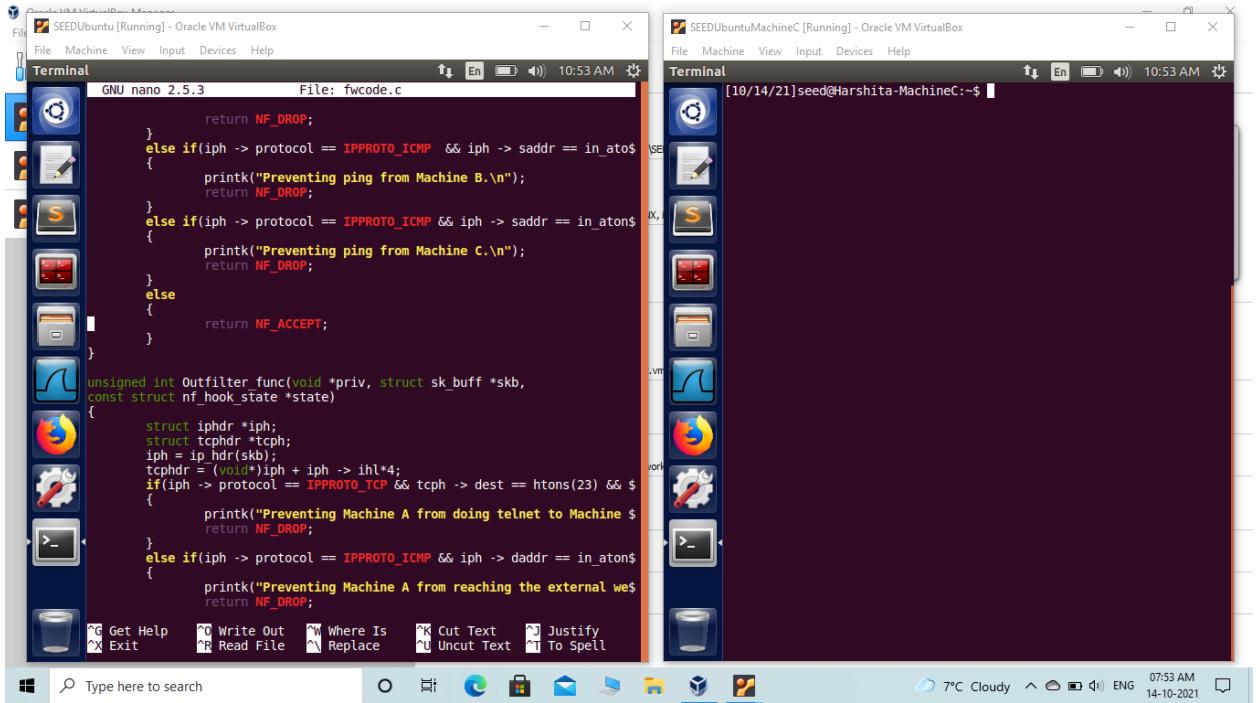


Now, we create a “fwcode.c” file for our netfilter firewall rules.



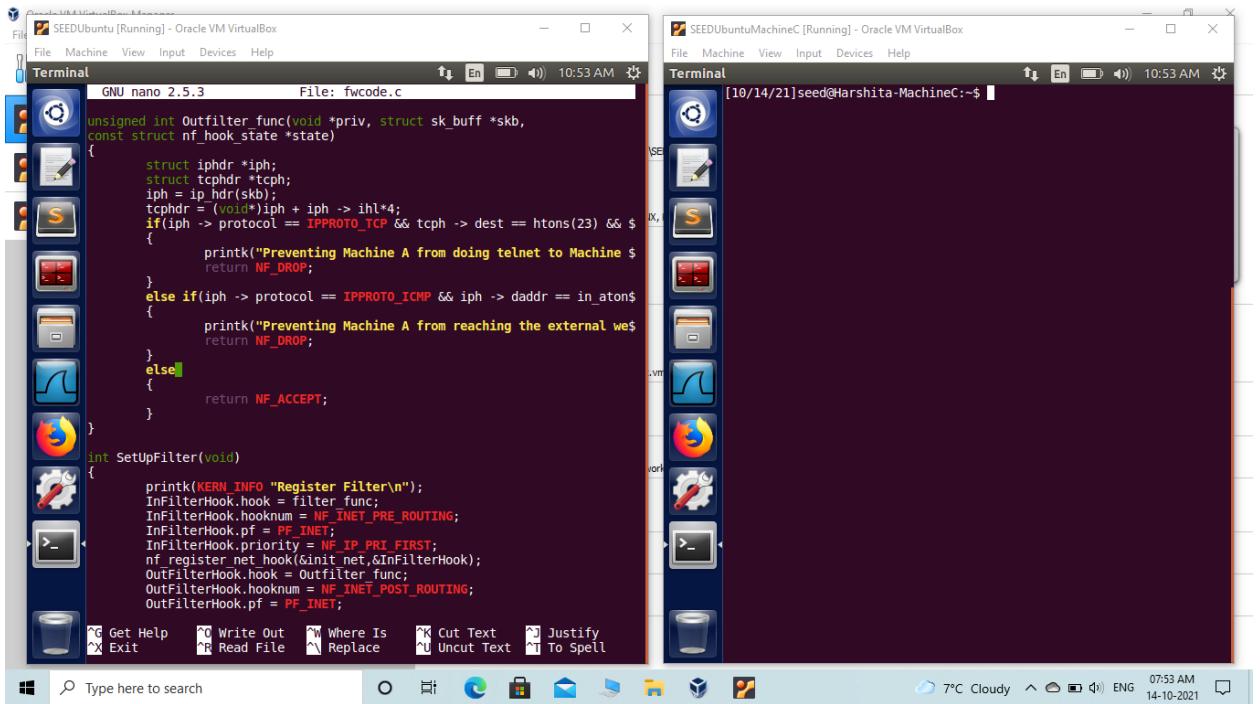
This is the code for our file “fwcode.c”.





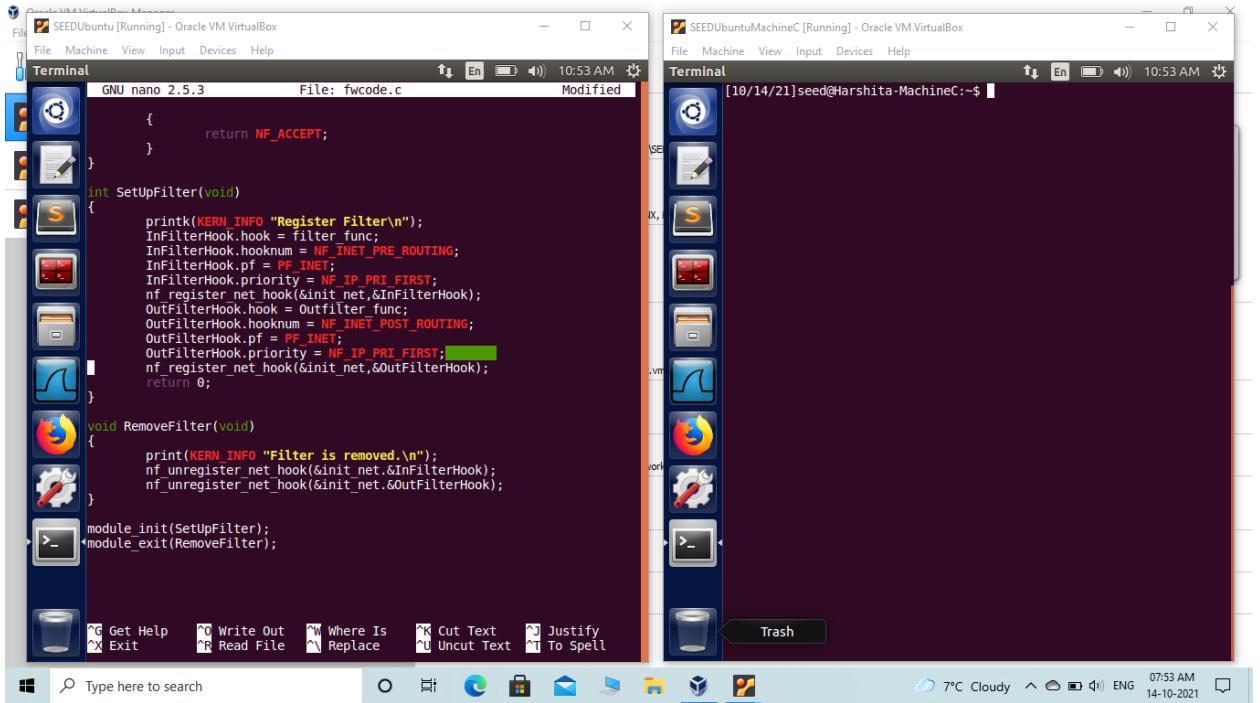
```
return NF_DROP;
}
else if(iph->protocol == IPPROTO_ICMP && iph->saddr == in_atos
{
    printk("Preventing ping from Machine B.\n");
    return NF_DROP;
}
else if(iph->protocol == IPPROTO_ICMP && iph->saddr == in_atos
{
    printk("Preventing ping from Machine C.\n");
    return NF_DROP;
}
else
{
    return NF_ACCEPT;
}

unsigned int Outfilter func(void *priv, struct sk_buff *skb,
const struct nf_hook_state *state)
{
    struct iphdr *iph;
    struct tcphdr *tcp;
    iph = ip_hdr(skb);
    tcphdr = (void*)iph + iph->ihl*4;
    if(iph->protocol == IPPROTO_TCP && tcph->dest == htons(23) && $)
    {
        printk("Preventing Machine A from doing telnet to Machine $");
        return NF_DROP;
    }
    else if(iph->protocol == IPPROTO_ICMP && iph->daddr == in_atos
    {
        printk("Preventing Machine A from reaching the external we$");
        return NF_DROP;
    }
    else
    {
        return NF_ACCEPT;
    }
}
```

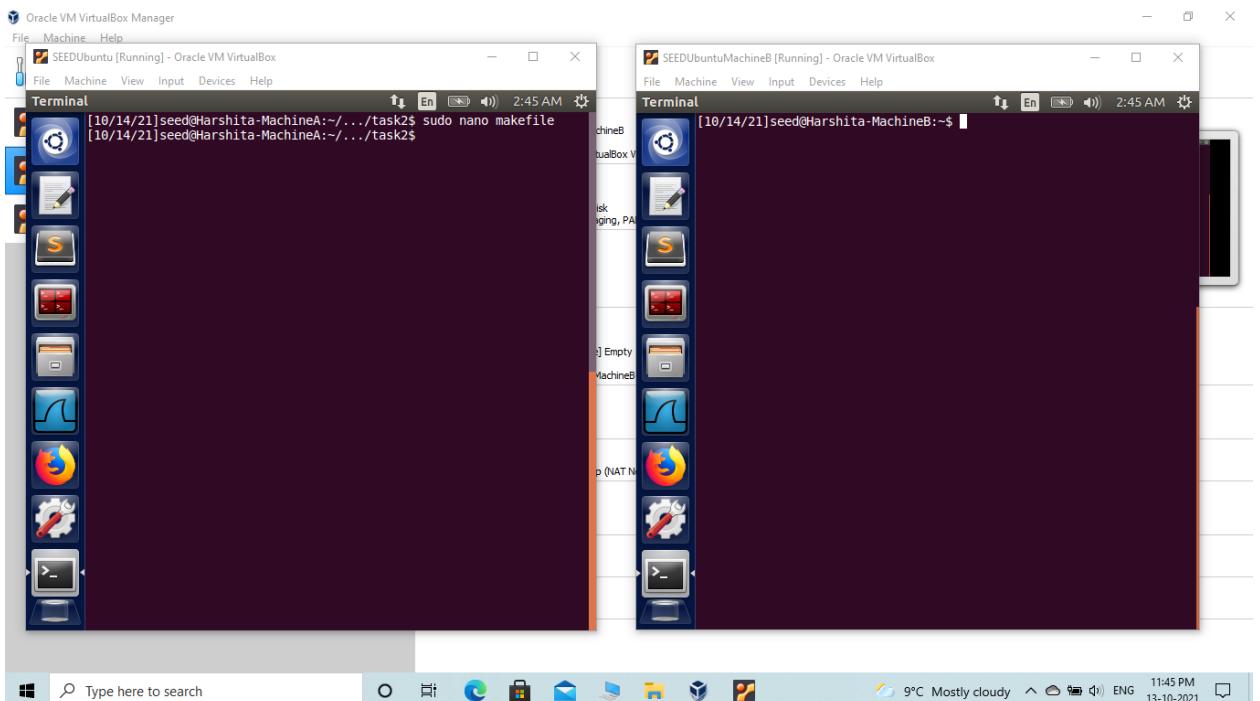


```
unsigned int Outfilter func(void *priv, struct sk_buff *skb,
const struct nf_hook_state *state)
{
    struct iphdr *iph;
    struct tcphdr *tcp;
    iph = ip_hdr(skb);
    tcphdr = (void*)iph + iph->ihl*4;
    if(iph->protocol == IPPROTO_TCP && tcph->dest == htons(23) && $)
    {
        printk("Preventing Machine A from doing telnet to Machine $");
        return NF_DROP;
    }
    else if(iph->protocol == IPPROTO_ICMP && iph->daddr == in_atos
    {
        printk("Preventing Machine A from reaching the external we$");
        return NF_DROP;
    }
    else
    {
        return NF_ACCEPT;
    }
}

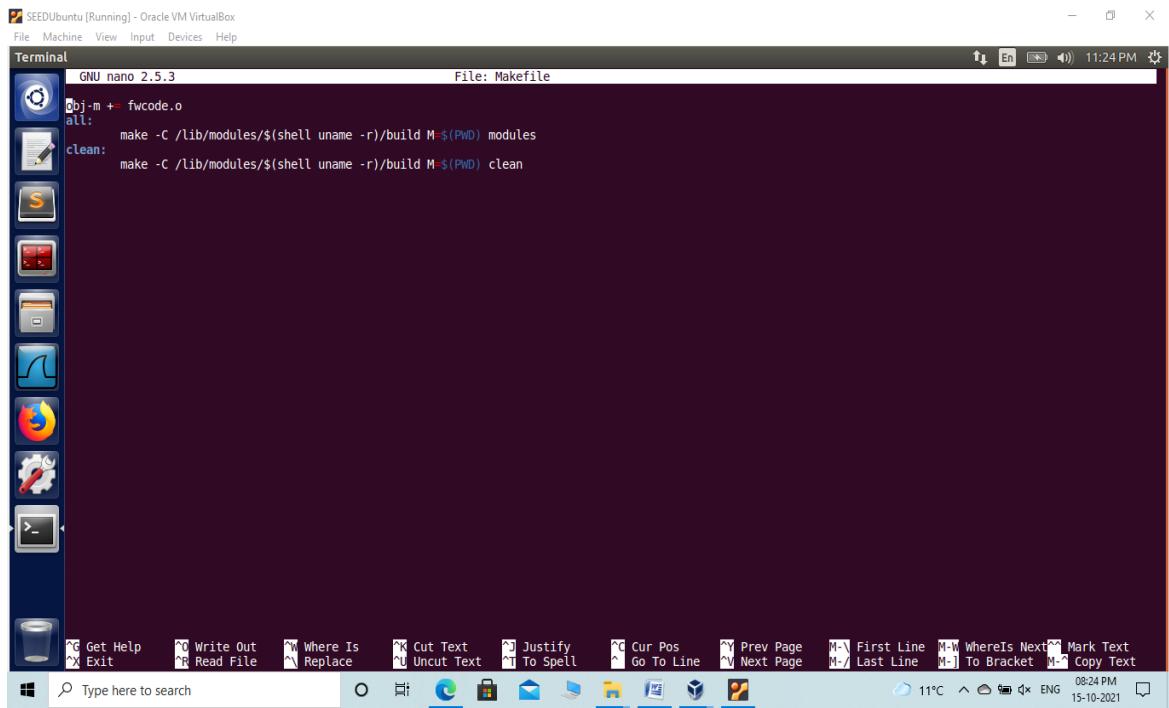
int SetUpFilter(void)
{
    printk(KERN_INFO "Register Filter\n");
    InFilterHook.hook = filter func;
    InFilterHook.hooknum = NF_INET_PRE_ROUTING;
    InFilterHook.pf = PF_INET;
    InFilterHook.priority = NF_IP_PRI_FIRST;
    nf_register_net_hook(&init_net,&InFilterHook);
    OutFilterHook.hook = Outfilter func;
    OutFilterHook.hooknum = NF_INET_POST_ROUTING;
    OutFilterHook.pf = PF_INET;
```



Now, we create a “Makefile”.



Here is the code for “Makefile”.

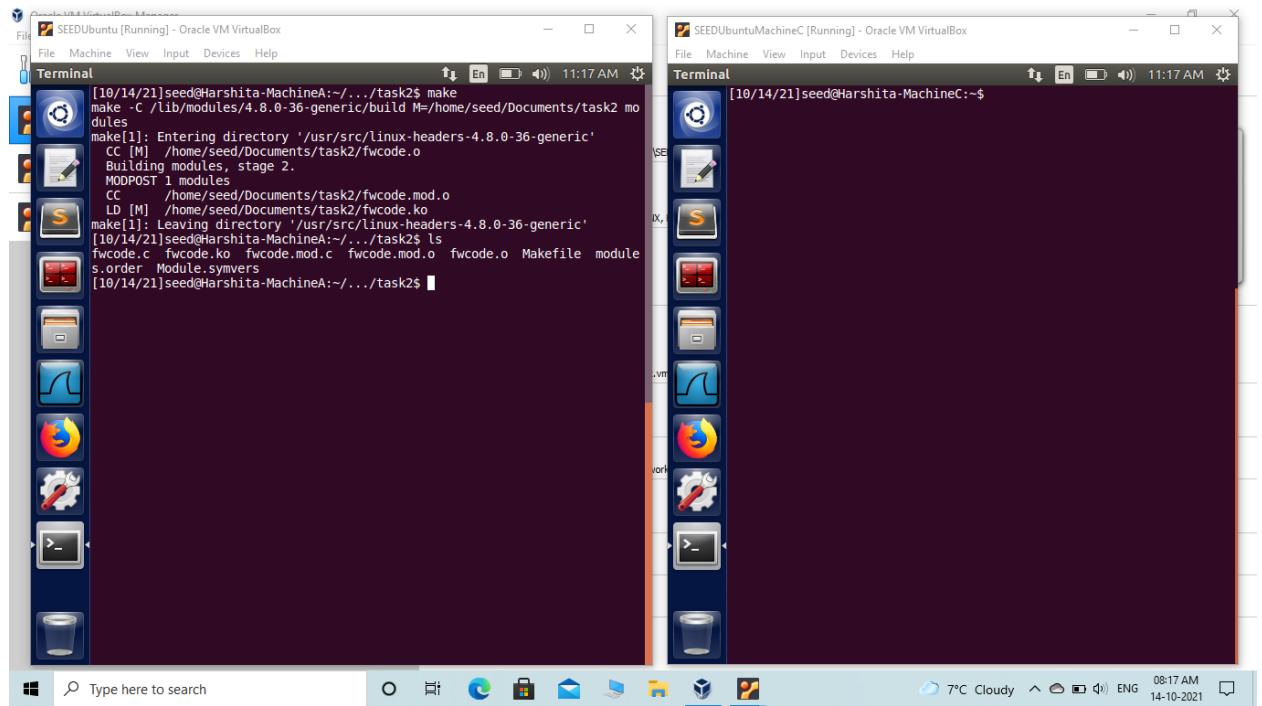


The screenshot shows a Linux desktop environment with a dark theme. A terminal window titled "File: Makefile" is open, displaying the following content:

```
obj-m += fwcode.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
```

The desktop interface includes a dock with icons for various applications like a file manager, terminal, browser, and system tools. The taskbar at the bottom shows the date and time as 15-10-2021 08:24 PM.

Now, we use “make” command.

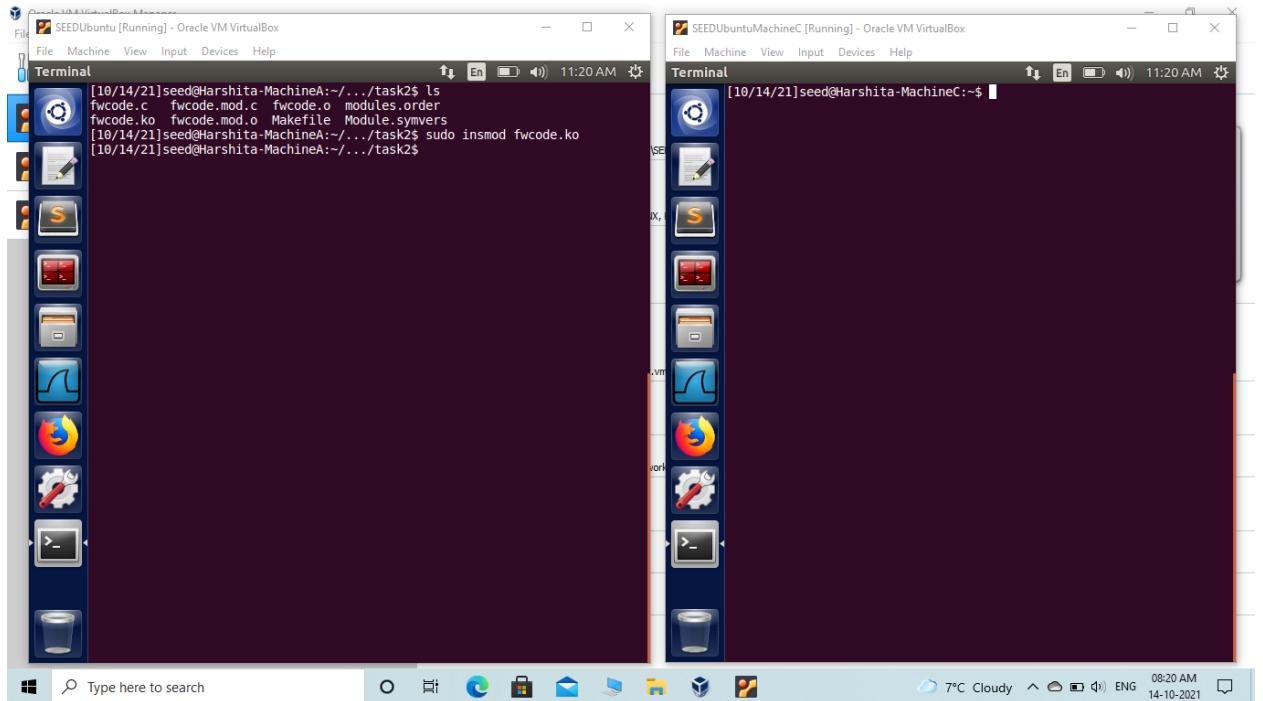


The screenshot shows a Windows desktop environment with two terminal windows side-by-side. Both terminals are titled "Terminal" and show the same command-line session:

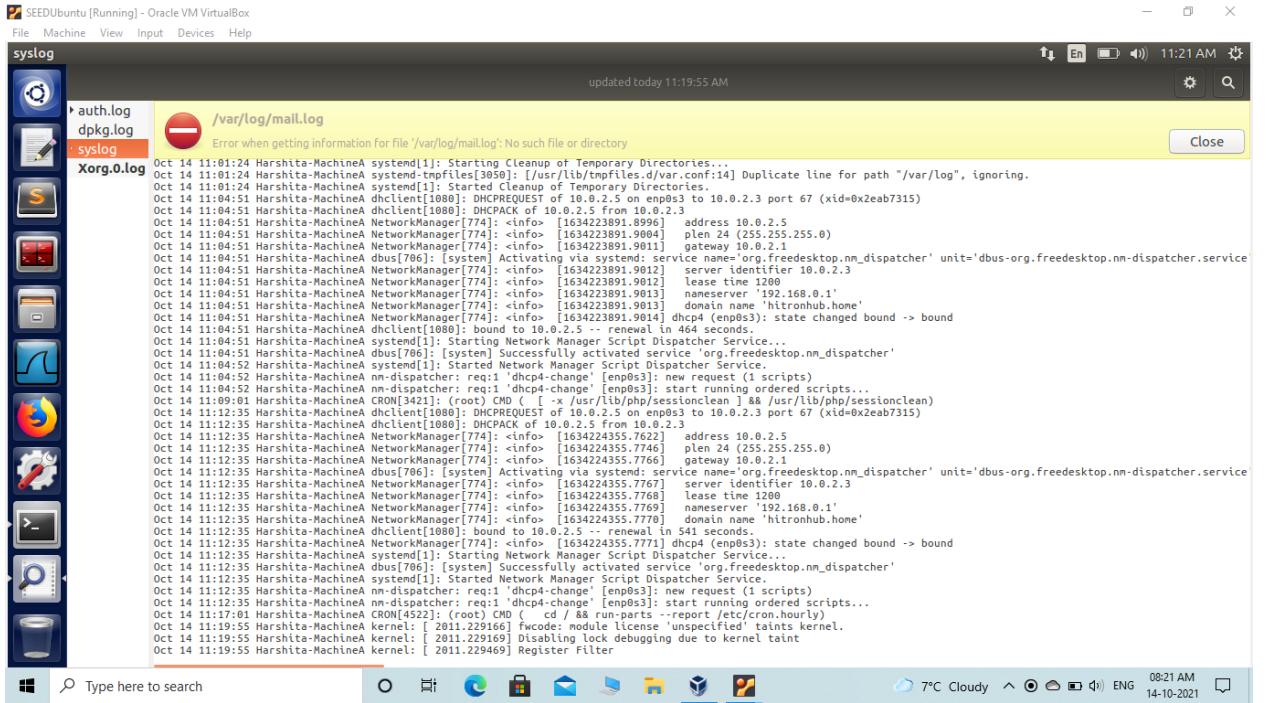
```
[10/14/21]seed@Harshita-MachineA:~/.../task2$ make
make -C /lib/modules/4.8.0-36-generic/build M=/home/seed/Documents/task2 modules
make[1]: Entering directory '/usr/src/linux-headers-4.8.0-36-generic'
  CC [M] /home/seed/Documents/task2/fwcode.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC [M] /home/seed/Documents/task2/fwcode.ko
make[1]: Leaving directory '/usr/src/linux-headers-4.8.0-36-generic'
[10/14/21]seed@Harshita-MachineA:~/.../task2$ ls
fwcode.c fwcode.ko fwcode.mod.c fwcode.mod.o fwcode.o Makefile module
s.order Module.symvers
[10/14/21]seed@Harshita-MachineA:~/.../task2$
```

The desktop interface includes a taskbar with various application icons and a system tray showing the date and time as 14-10-2021 08:17 AM.

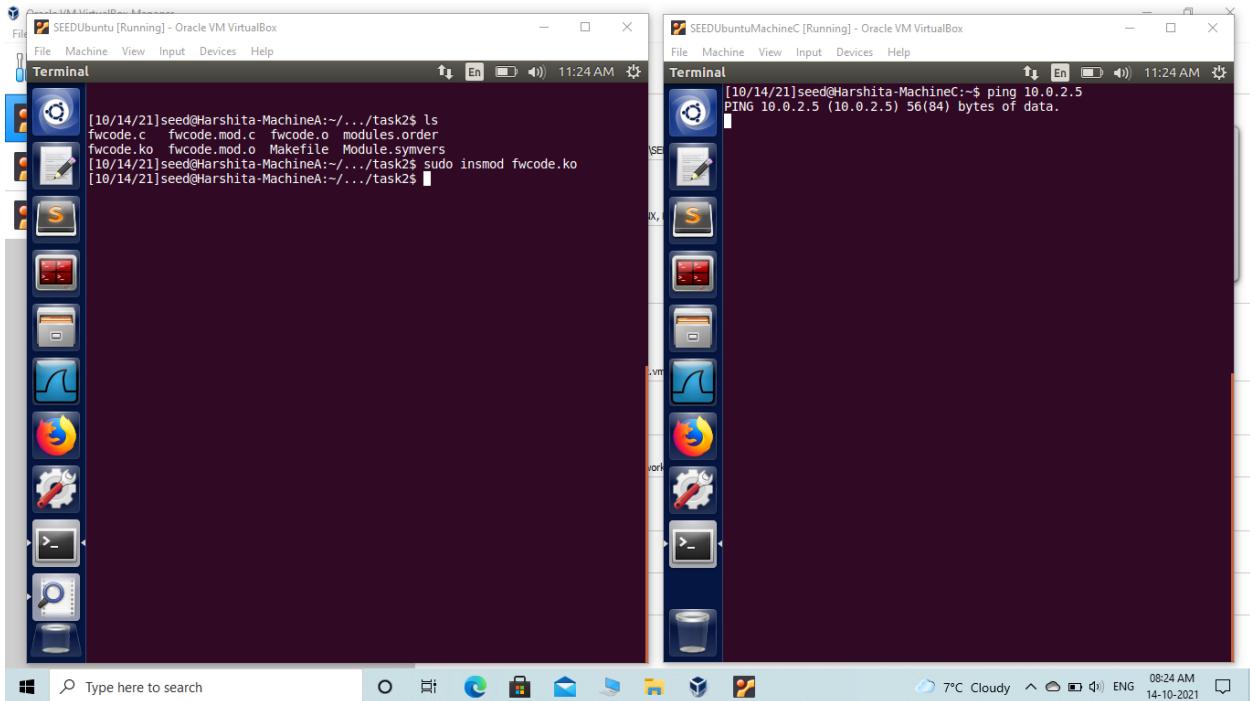
We use “sudo insmod fwcode.ko” to insert the module “fwcode.ko” into the kernel space.



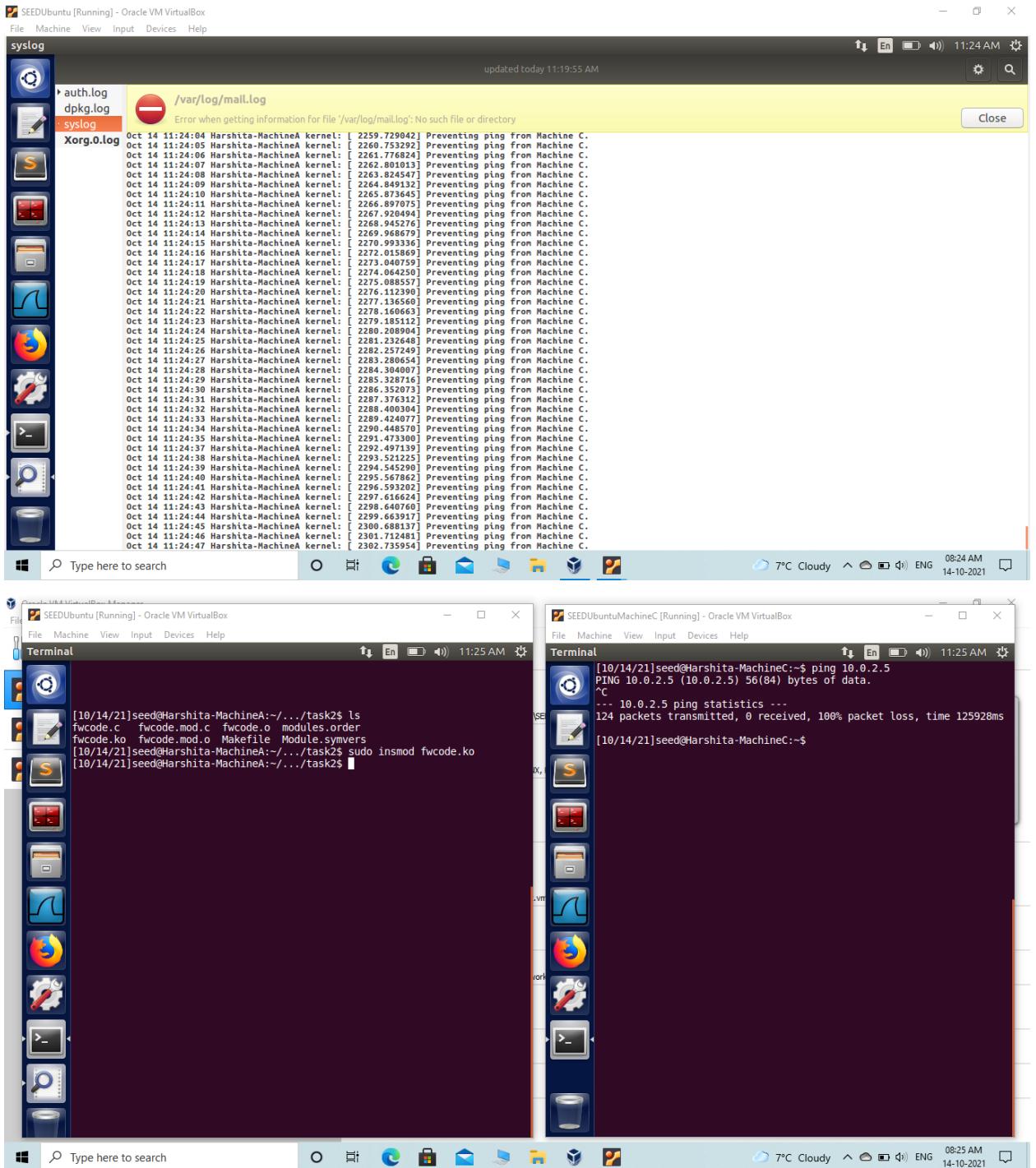
We see in the syslog that the filter is registered.



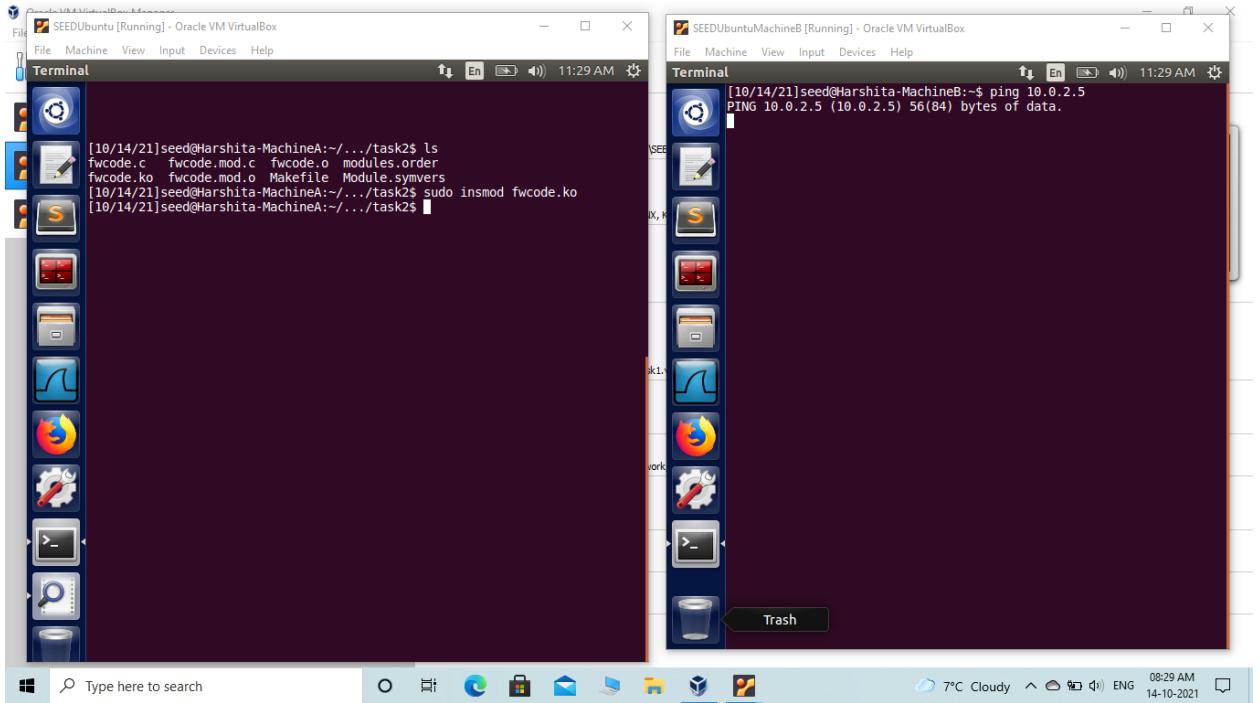
(a) Prevent Machine C to ping Machine A.



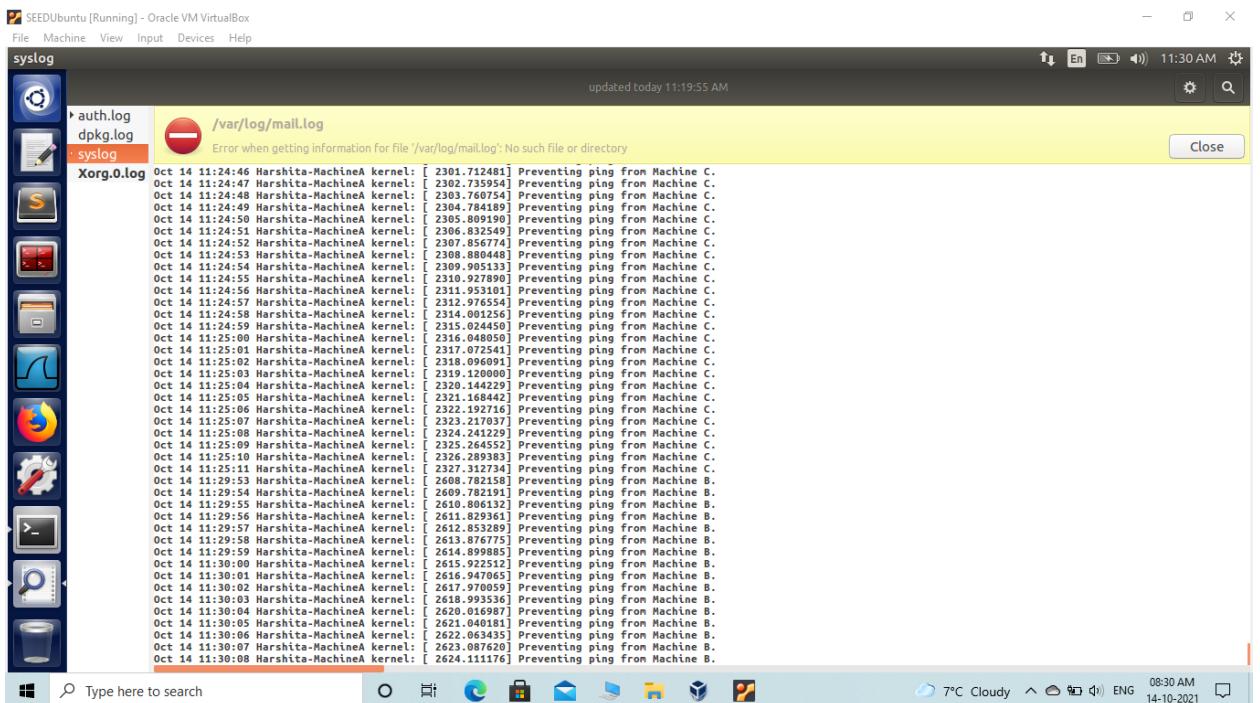
Messages in the syslog “Preventing ping from Machine C”.



(b) Preventing Machine B from pinging Machine A.



Messages in the syslog : “Preventing ping from Machine C”.



The image shows two side-by-side terminal windows running on different hosts. The left window is titled 'SEEDUbuntu [Running] - Oracle VM VirtualBox' and shows a terminal session on 'MachineA'. The right window is titled 'SEEDubuntuMachineB [Running] - Oracle VM VirtualBox' and shows a terminal session on 'MachineB'. Both hosts have a similar desktop environment with a taskbar at the bottom.

MachineA Terminal Session:

```
[10/14/21]seed@Harshita-MachineA:~/.../task2$ ls
fwcode.c fwcode.mod.c fwcode.o modules.order
fwcode.ko fwcode.mod.o Makefile Module.symvers
[10/14/21]seed@Harshita-MachineA:~/.../task2$ sudo insmod fwcode.ko
[10/14/21]seed@Harshita-MachineA:~/.../task2$
```

MachineB Terminal Session:

```
[10/14/21]seed@Harshita-MachineB:~$ ping 10.0.2.5
PING 10.0.2.5 (10.0.2.5) 56(84) bytes of data.
^C
--- 10.0.2.5 ping statistics ---
32 packets transmitted, 0 received, 100% packet loss, time 31720ms
[10/14/21]seed@Harshita-MachineB:~$
```

(c) Preventing telnet from Machine B to Machine A.

The image shows two side-by-side terminal windows running on different hosts. The left window is titled 'SEEDUbuntu [Running] - Oracle VM VirtualBox' and shows a terminal session on 'MachineA'. The right window is titled 'SEEDubuntuMachineB [Running] - Oracle VM VirtualBox' and shows a terminal session on 'MachineB'. Both hosts have a similar desktop environment with a taskbar at the bottom.

MachineA Terminal Session:

```
[10/14/21]seed@Harshita-MachineA:~/.../task2$ ls
fwcode.c fwcode.mod.c fwcode.o modules.order
fwcode.ko fwcode.mod.o Makefile Module.symvers
[10/14/21]seed@Harshita-MachineA:~/.../task2$ sudo insmod fwcode.ko
[10/14/21]seed@Harshita-MachineA:~/.../task2$
```

MachineB Terminal Session:

```
[10/14/21]seed@Harshita-MachineB:~$ telnet 10.0.2.5
Trying 10.0.2.5...
```

Messages in syslog: “Preventing Machine B from doing telnet to Machine A”.

SEEDUbuntu [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

updated today 11:19:55 AM

syslog

/var/log/mail.log

Error when getting information for file '/var/log/mail.log': No such file or directory

Close

```

Oct 14 11:30:04 Harshita-MachineA kernel: [ 2620.010997] Preventing ping from Machine B.
Oct 14 11:30:05 Harshita-MachineA kernel: [ 2621.064018] Preventing ping from Machine B.
Oct 14 11:30:06 Harshita-MachineA kernel: [ 2622.063435] Preventing ping from Machine B.
Oct 14 11:30:07 Harshita-MachineA kernel: [ 2623.087620] Preventing ping from Machine B.
Oct 14 11:30:08 Harshita-MachineA kernel: [ 2624.111176] Preventing ping from Machine B.
Oct 14 11:30:09 Harshita-MachineA kernel: [ 2625.135217] Preventing ping from Machine B.
Oct 14 11:30:10 Harshita-MachineA kernel: [ 2626.157956] Preventing ping from Machine B.
Oct 14 11:30:11 Harshita-MachineA kernel: [ 2627.180645] Preventing ping from Machine B.
Oct 14 11:30:12 Harshita-MachineA kernel: [ 2628.203687] Preventing ping from Machine B.
Oct 14 11:30:13 Harshita-MachineA kernel: [ 2629.227874] Preventing ping from Machine B.
Oct 14 11:30:14 Harshita-MachineA kernel: [ 2630.252045] Preventing ping from Machine B.
Oct 14 11:30:15 Harshita-MachineA kernel: [ 2631.276149] Preventing ping from Machine B.
Oct 14 11:30:16 Harshita-MachineA kernel: [ 2632.298603] Preventing ping from Machine B.
Oct 14 11:30:18 Harshita-MachineA kernel: [ 2633.320625] Preventing ping from Machine B.
Oct 14 11:30:19 Harshita-MachineA kernel: [ 2634.344627] Preventing ping from Machine B.
Oct 14 11:30:20 Harshita-MachineA kernel: [ 2635.369458] Preventing ping from Machine B.
Oct 14 11:30:21 Harshita-MachineA kernel: [ 2636.392178] Preventing ping from Machine B.
Oct 14 11:30:22 Harshita-MachineA kernel: [ 2637.416591] Preventing ping from Machine B.
Oct 14 11:30:23 Harshita-MachineA kernel: [ 2638.439041] Preventing ping from Machine B.
Oct 14 11:30:24 Harshita-MachineA kernel: [ 2639.462777] Preventing ping from Machine B.
Oct 14 11:30:25 Harshita-MachineA kernel: [ 2640.486441] Preventing ping from Machine B.
Oct 14 11:30:40 Harshita-MachineA dhclient[1080]: DHCPACK of 10.0.2.5 from 10.0.2.3
Oct 14 11:30:40 Harshita-MachineA dhclient[1080]: DHCPACK of 10.0.2.5 from 10.0.2.3
Oct 14 11:30:40 Harshita-MachineA NetworkManager[774]: <info> [1634225440.1256] address 10.0.2.5
Oct 14 11:30:40 Harshita-MachineA NetworkManager[774]: <info> [1634225440.1264] plen 24 (255.255.255.0)
Oct 14 11:30:40 Harshita-MachineA NetworkManager[774]: <info> [1634225440.1272] gateway 10.0.2.1
Oct 14 11:30:40 Harshita-MachineA NetworkManager[774]: <info> [1634225440.1278] server identifier 10.0.2.3
Oct 14 11:30:40 Harshita-MachineA NetworkManager[774]: <info> [1634225440.1286] lease 10.0.2.5
Oct 14 11:30:40 Harshita-MachineA NetworkManager[774]: <info> [1634225440.1292] nameserver '192.168.0.1'
Oct 14 11:30:40 Harshita-MachineA NetworkManager[774]: <info> [1634225440.1292] domain name 'hitronhub.home'
Oct 14 11:30:40 Harshita-MachineA NetworkManager[774]: <info> [1634225440.1293] dhcpc (enp0s3): state changed bound -> bound
Oct 14 11:30:40 Harshita-MachineA dhclient[1080]: bound to 10.0.2.5 -- renewal in 552 seconds.
Oct 14 11:30:40 Harshita-MachineA dbus[706]: [system] activating via systemd: service name='org.freedesktop.nm-dispatcher' unit='dbus-org.freedesktop.nm-dispatcher.service'
Oct 14 11:30:40 Harshita-MachineA dbus[706]: [system] Started Network Manager Script Dispatcher Service.
Oct 14 11:30:40 Harshita-MachineA systemd[1]: Started Network Manager Script Dispatcher Service.
Oct 14 11:30:40 Harshita-MachineA nn-dispatcher: req:1 'dhcp4-change' [enp0s3]: new request (1 scripts)
Oct 14 11:30:40 Harshita-MachineA nn-dispatcher: req:1 'dhcp4-change' [enp0s3]: start running ordered scripts...
Oct 14 11:31:16 Harshita-MachineA kernel: [ 2691.547640] Preventing Machine B from doing telnet to Machine A.
Oct 14 11:31:17 Harshita-MachineA kernel: [ 2692.556050] Preventing Machine B from doing telnet to Machine A.
Oct 14 11:31:19 Harshita-MachineA kernel: [ 2694.571554] Preventing Machine B from doing telnet to Machine A.
Oct 14 11:31:23 Harshita-MachineA kernel: [ 2698.463340] Preventing Machine B from doing telnet to Machine A.

```

Type here to search

7°C Cloudy 14-10-2021

SEEDUbuntuMachineB [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

updated today 11:32 AM

Terminal

```

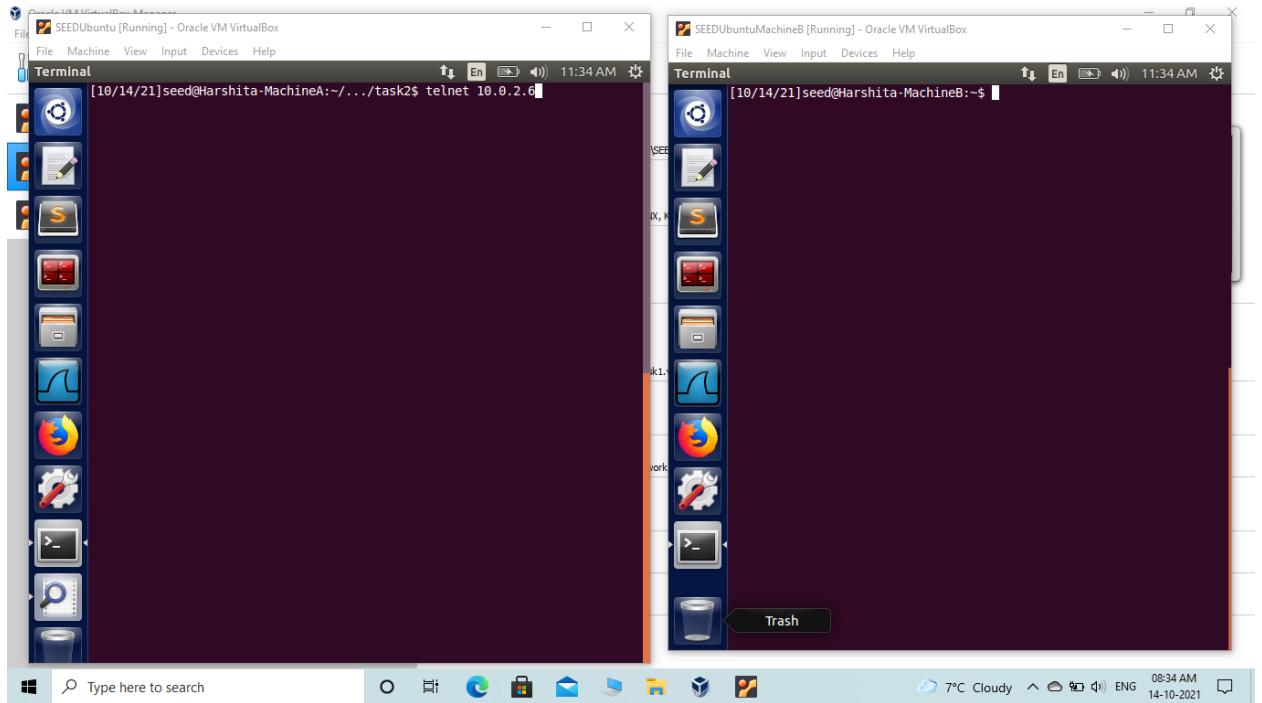
[10/14/21]seed@Harshita-MachineB:~$ telnet 10.0.2.5
Trying 10.0.2.5...
[10/14/21]seed@Harshita-MachineB:~$ 

```

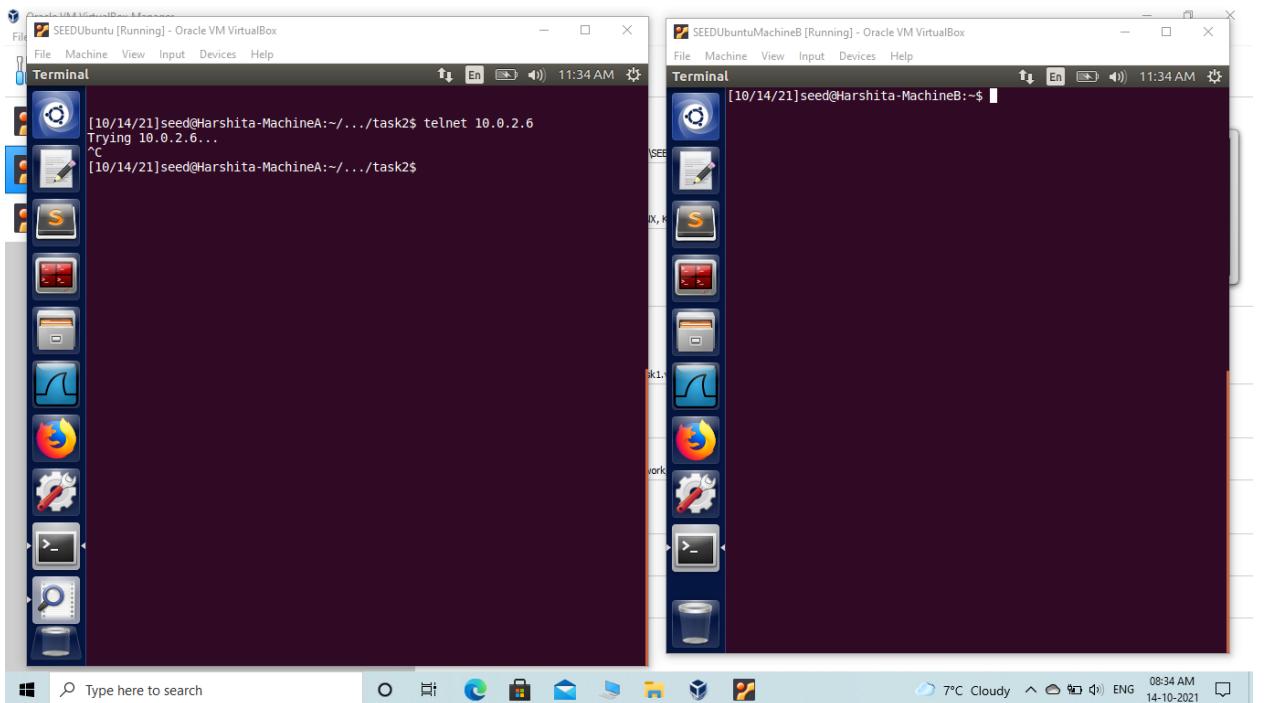
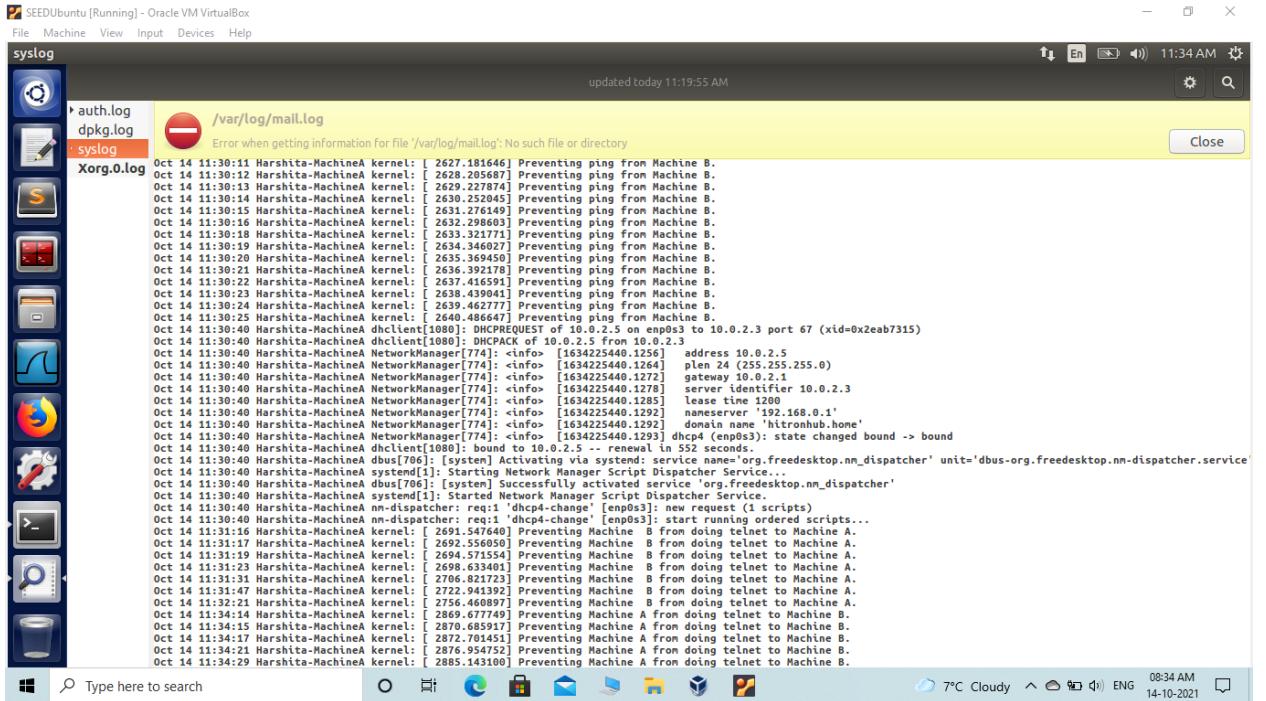
Type here to search

7°C Cloudy 14-10-2021

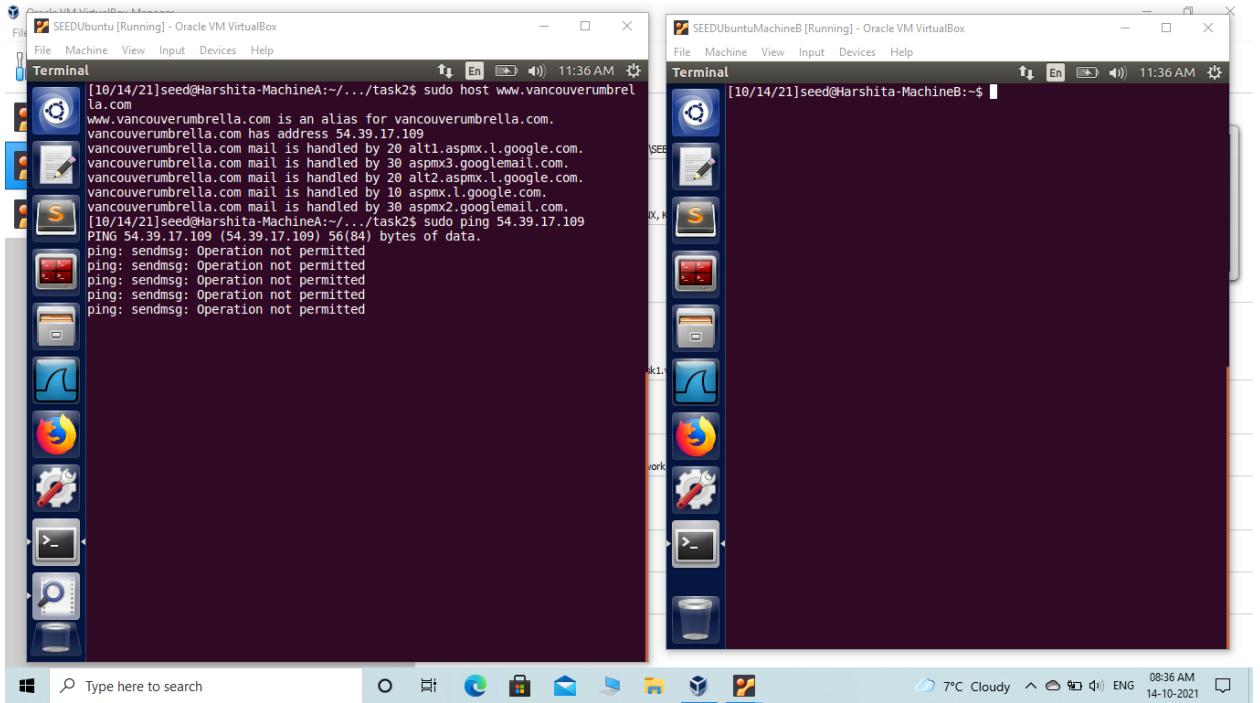
(d) Prevent machine A from doing telnet to Machine B



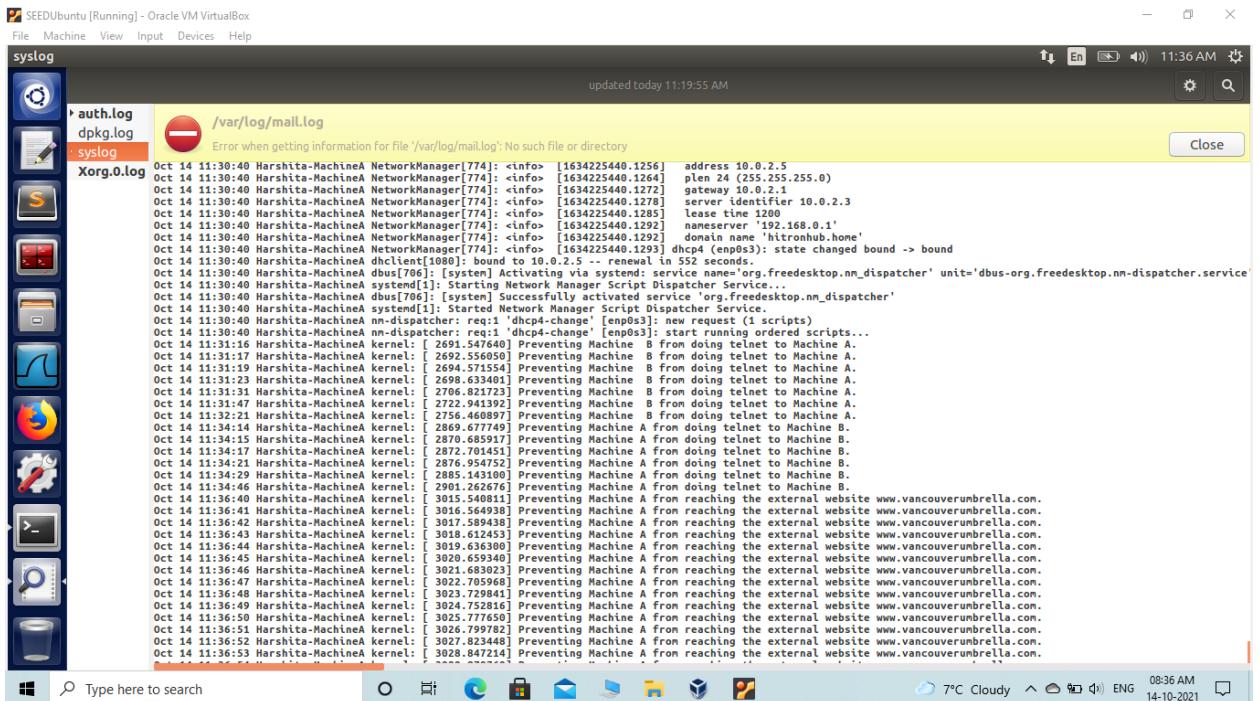
Messages in syslog: “Preventing machine A from doing telnet to Machine B”

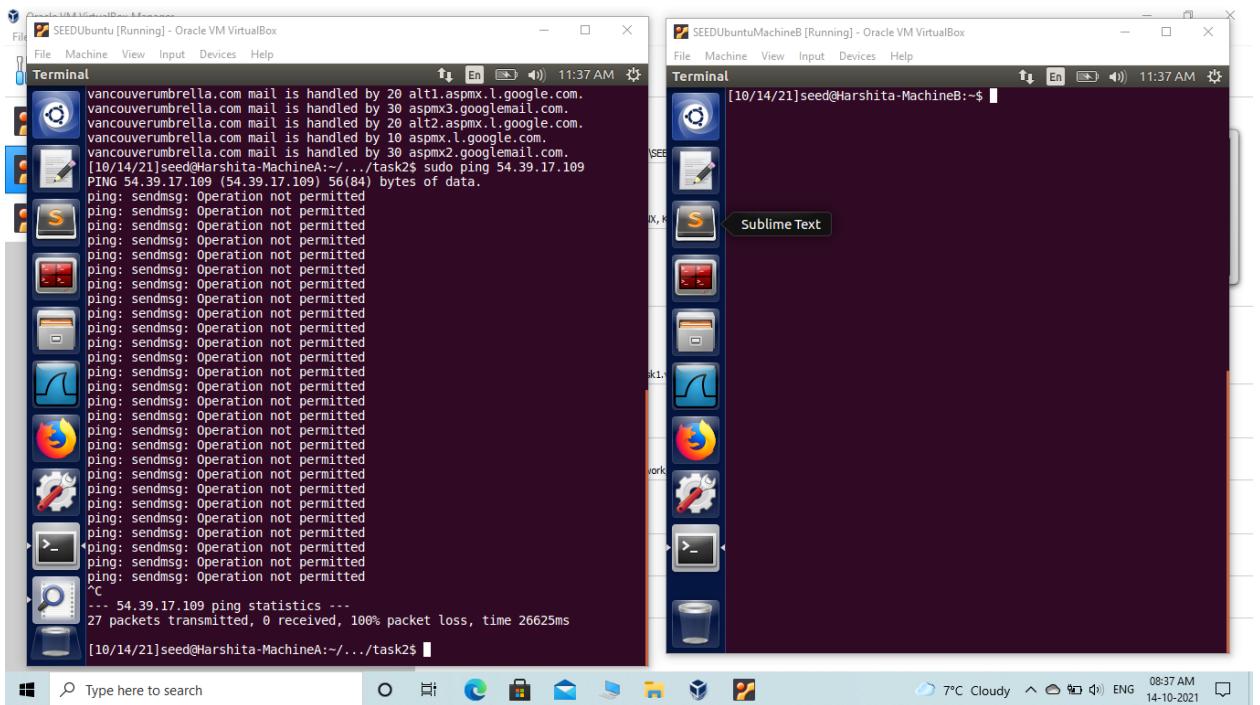


**(e) Prevent Machine A from accessing an external website
“www.vancouverumbrella.com”**

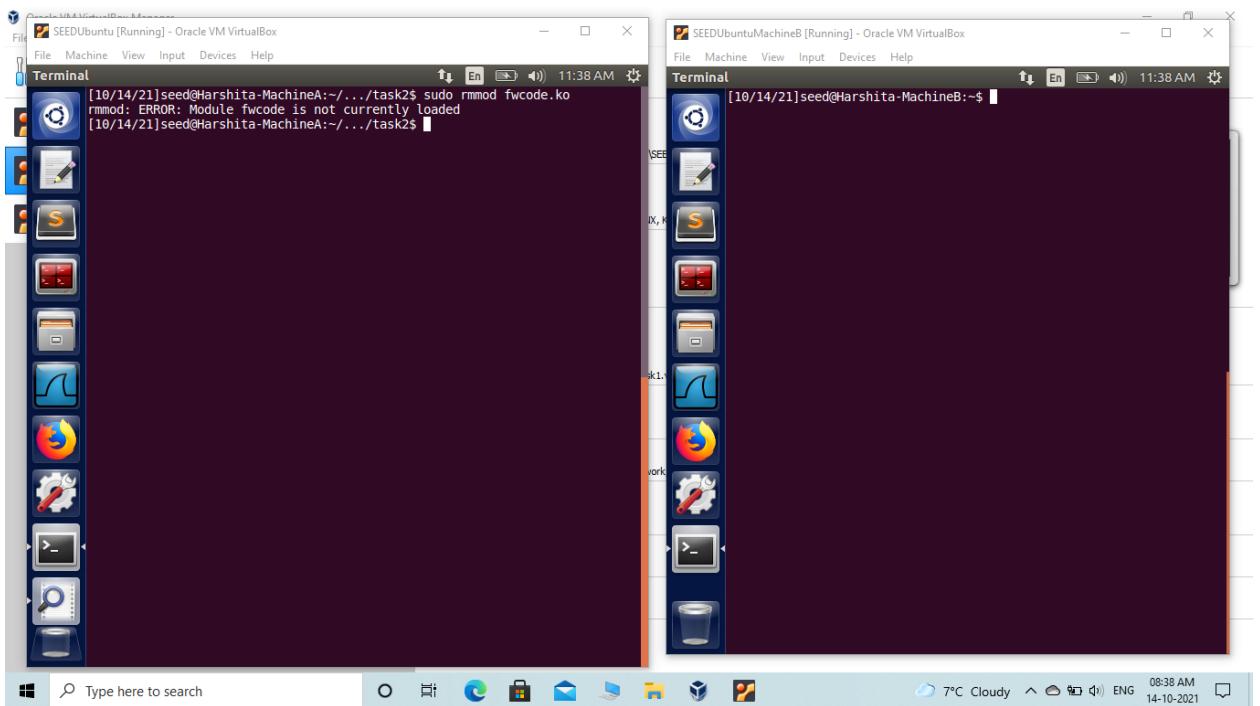


Messages in the syslog: “Preventing machine A from telneting external website”



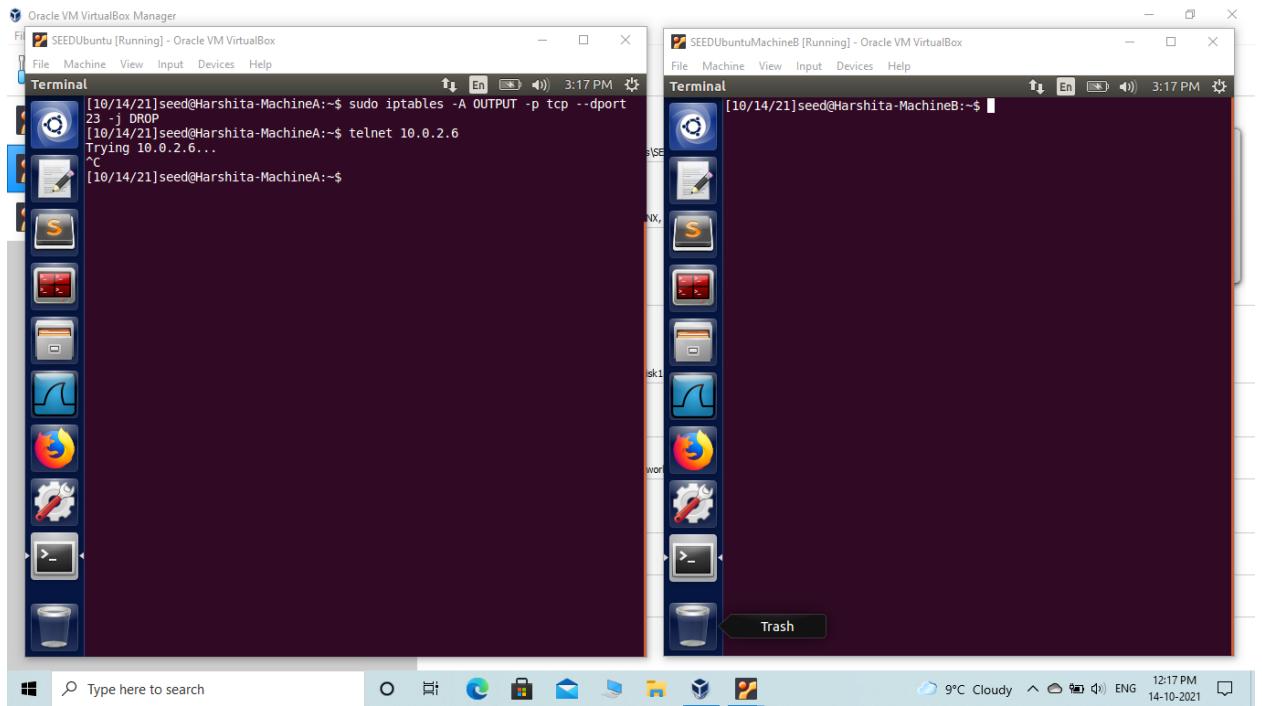


Now, removing the module using “rmmod” command.

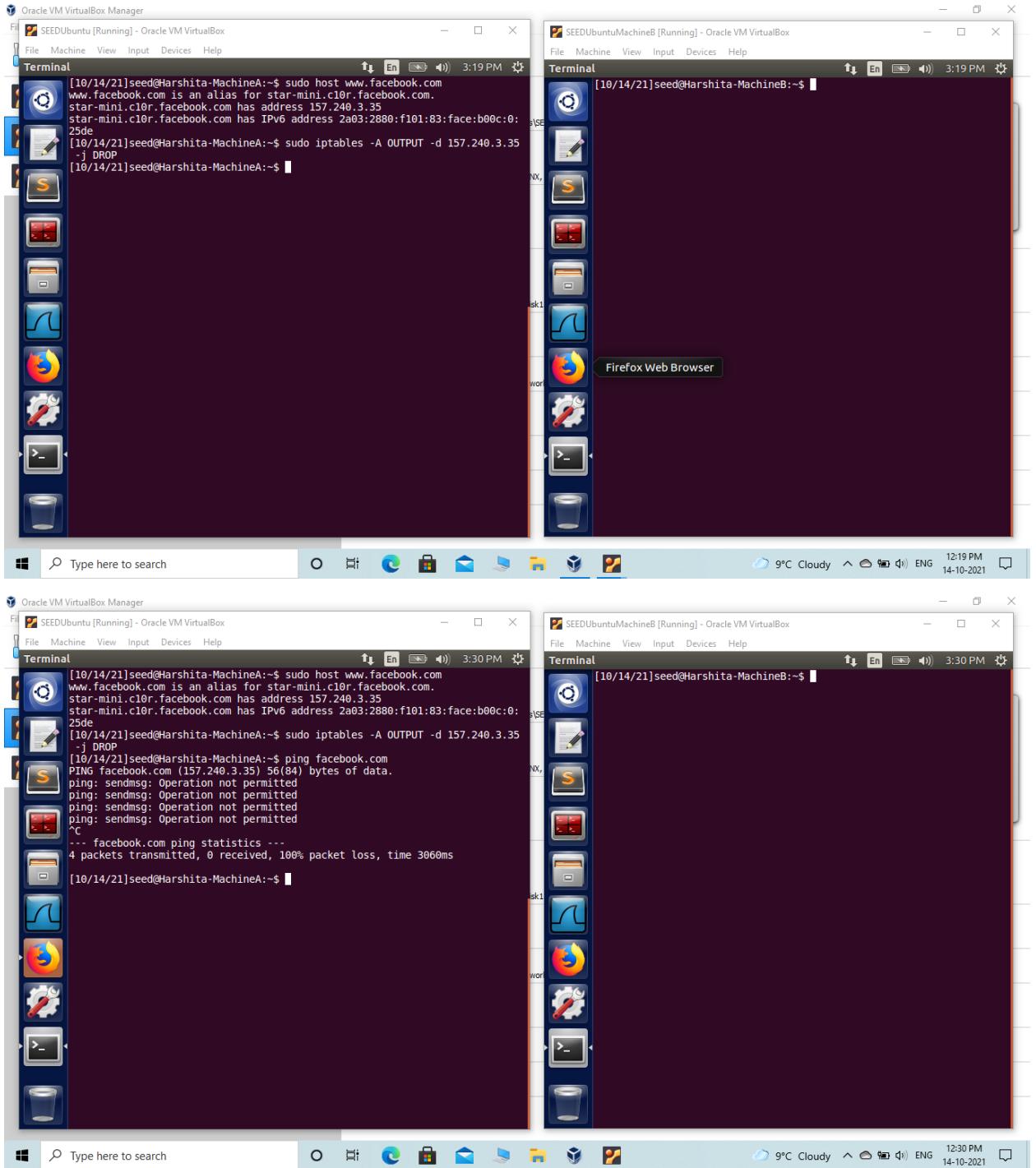


Task 3: Evading Egress Filtering

- Block all the outgoing traffic to external telnet servers.

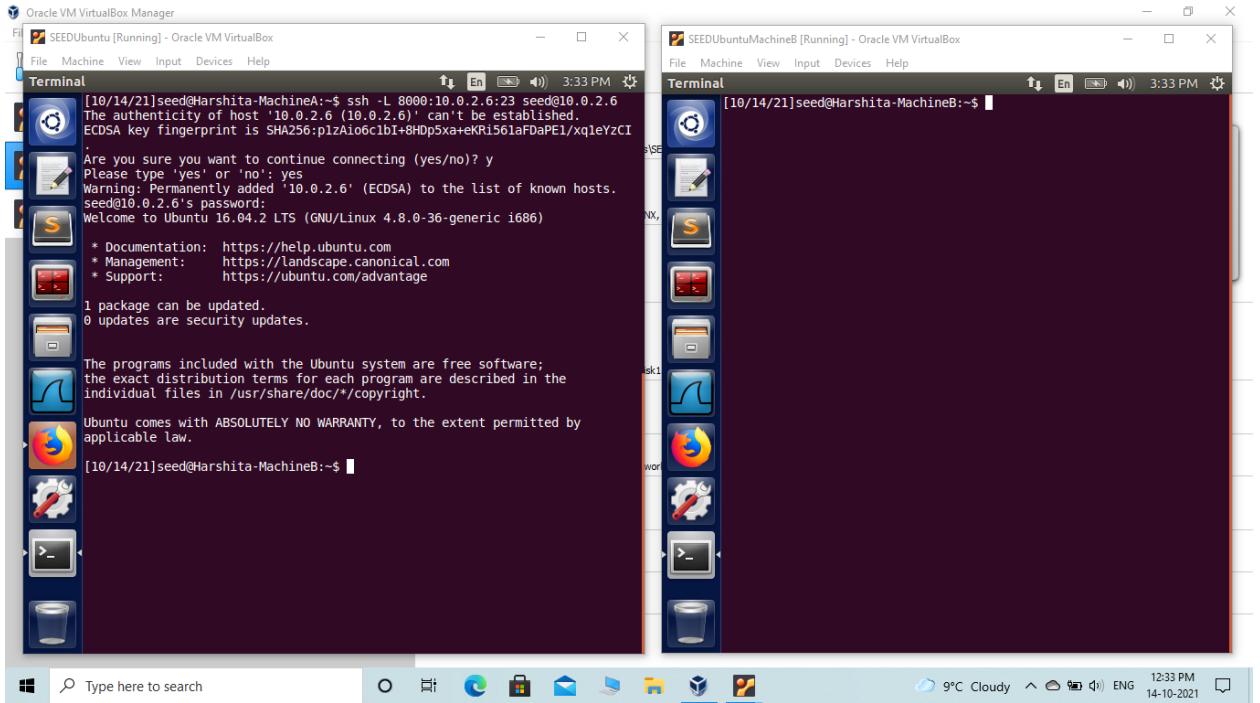


- Block all the outgoing traffic to www.facebook.com

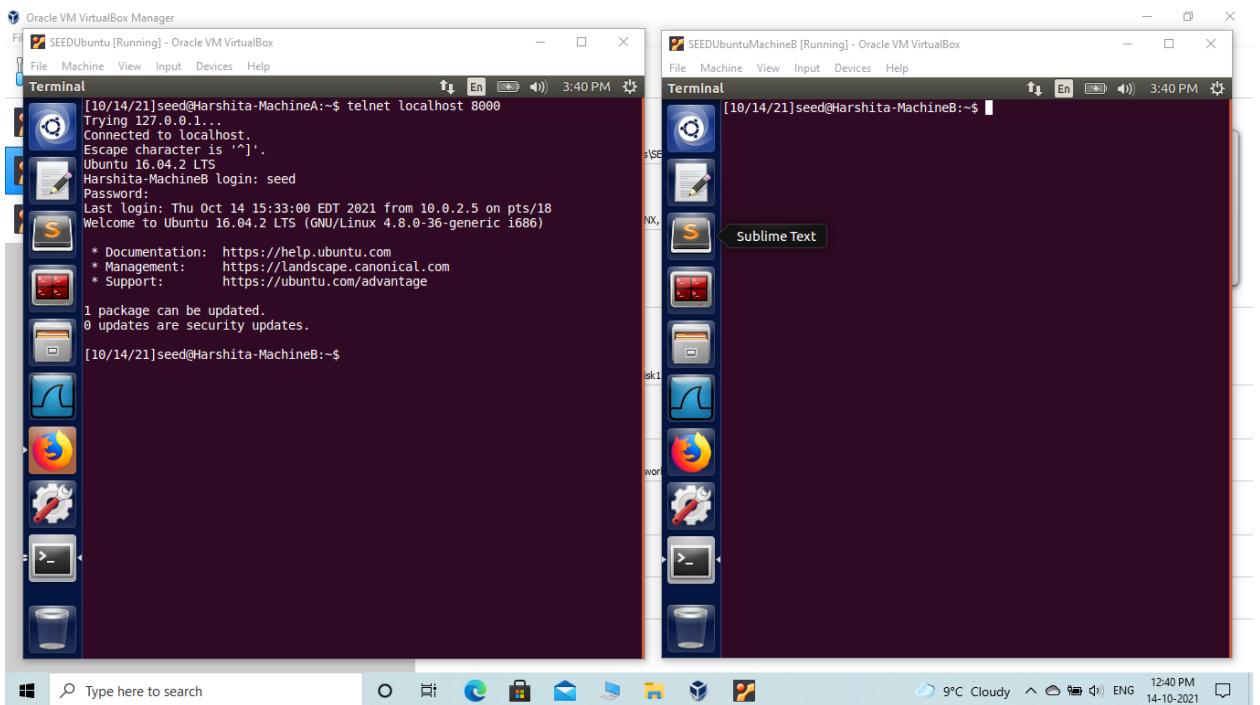


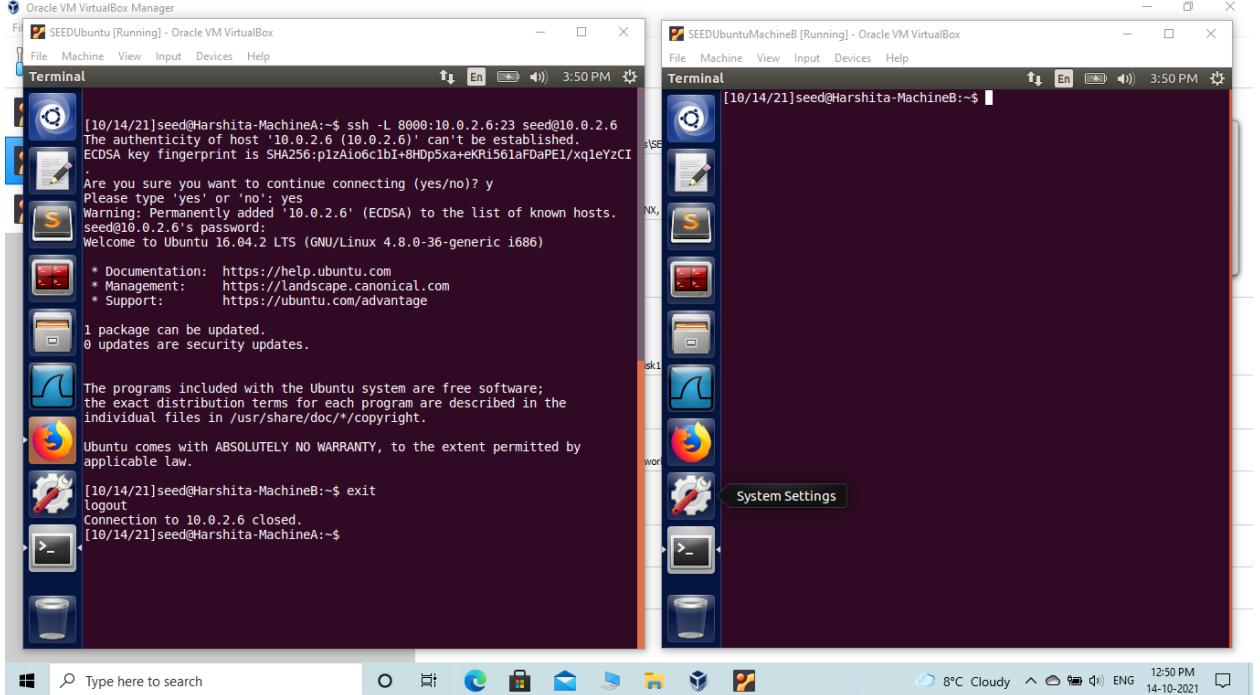
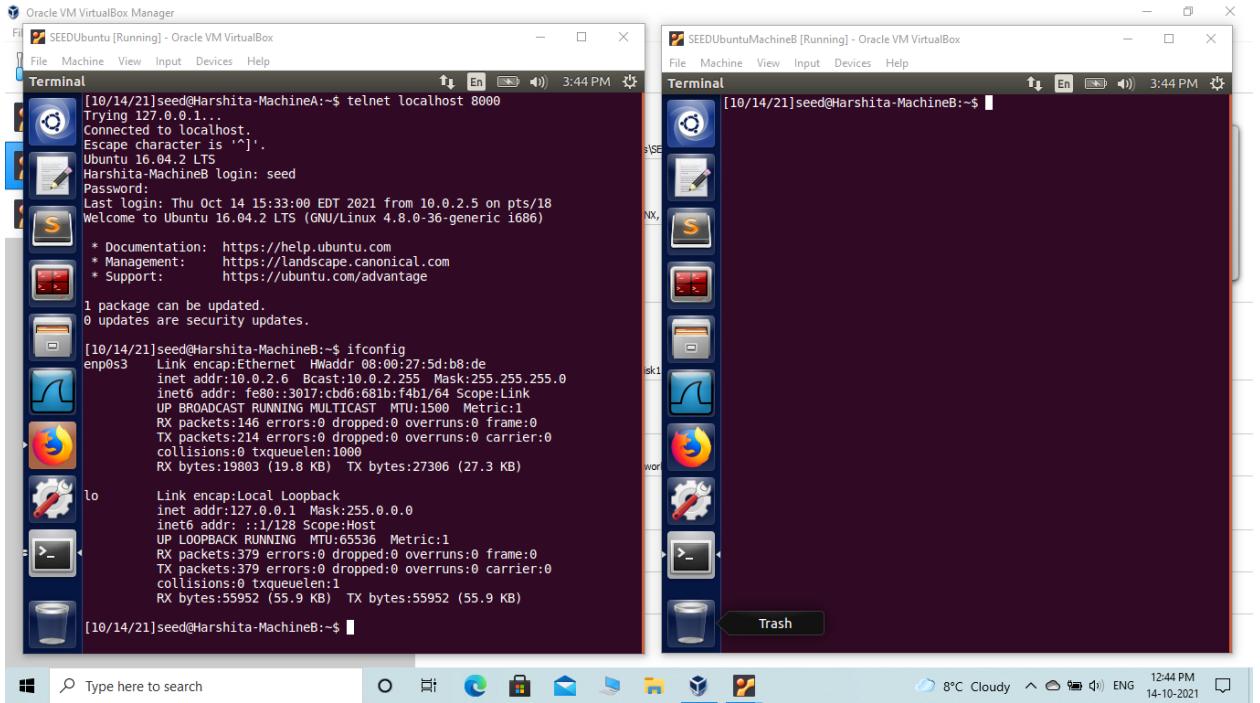
(a) Telnet to Machine B through the firewall

Now, we are connecting to machine B.



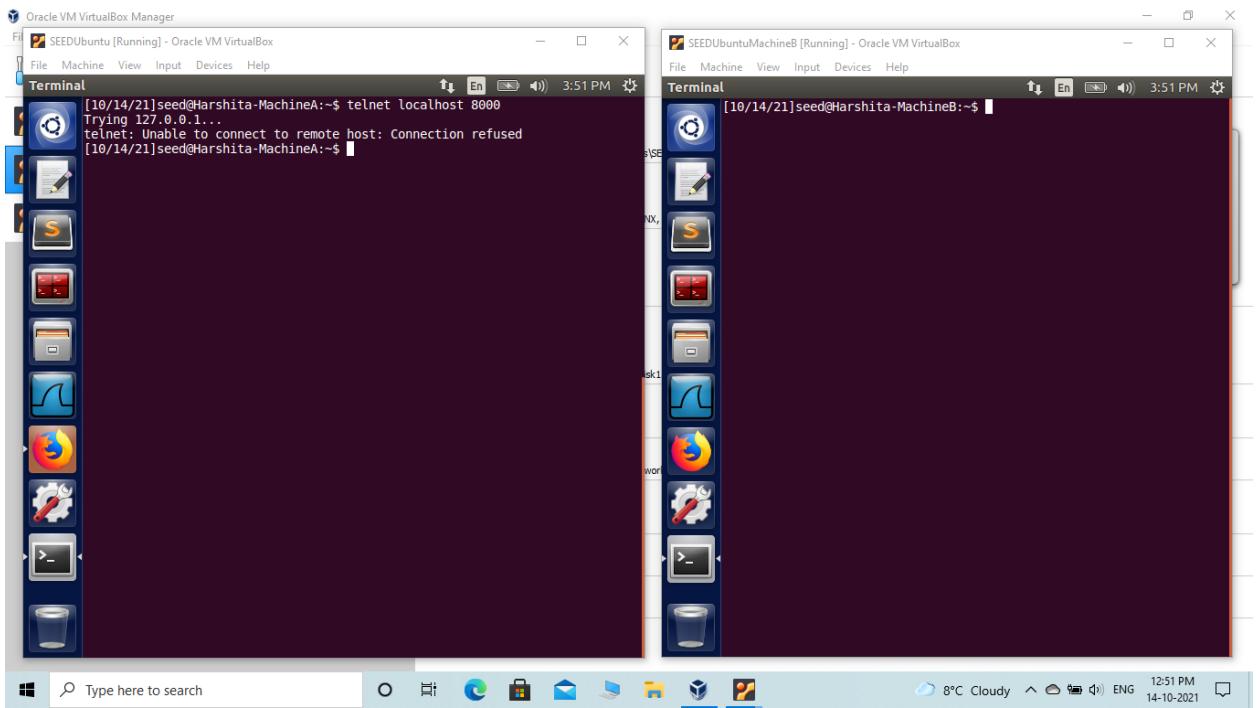
Now, we can telnet Machine B.





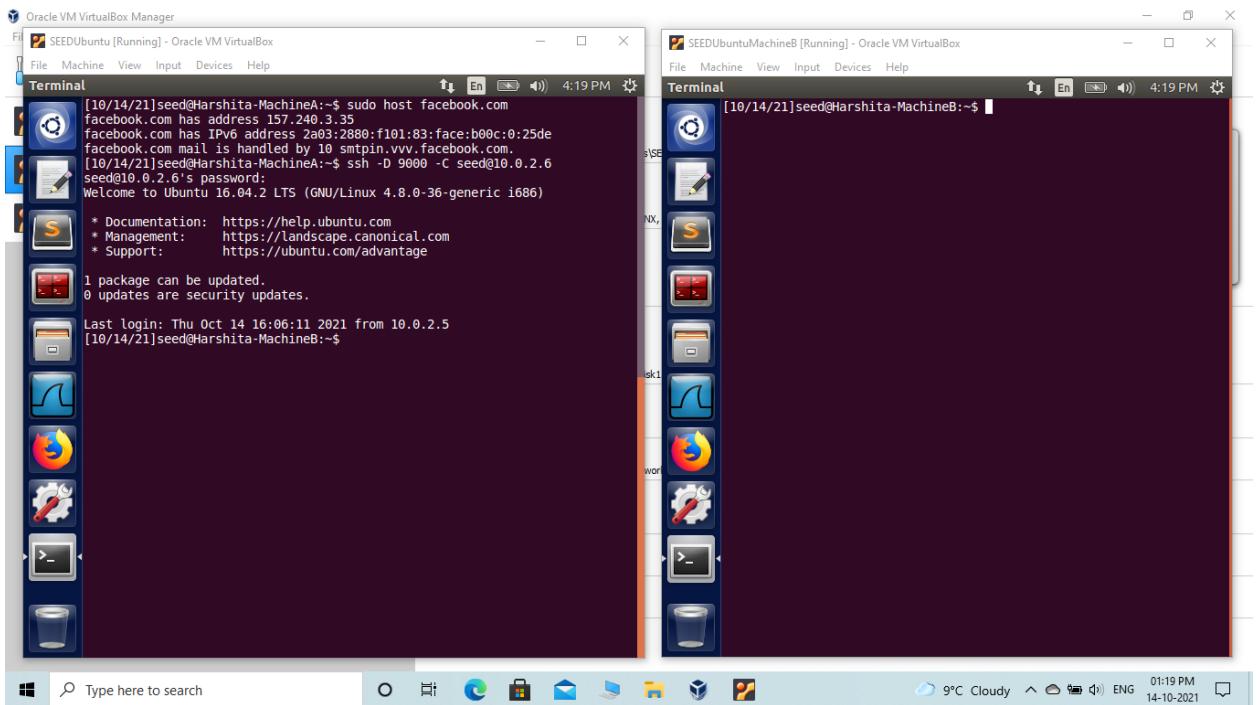
Now, We are again trying to connect to machine B.

But we are unable to connect because we have closed the connection.

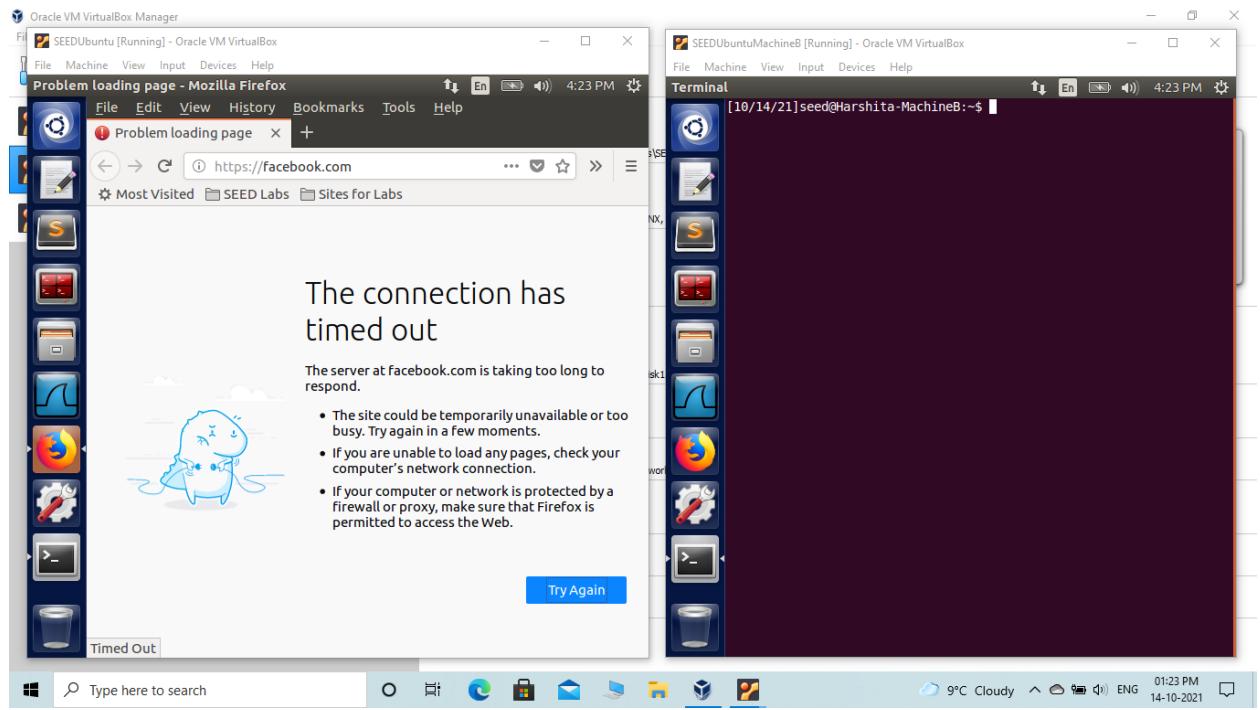


(b) Connect to Facebook using SSH Tunnel

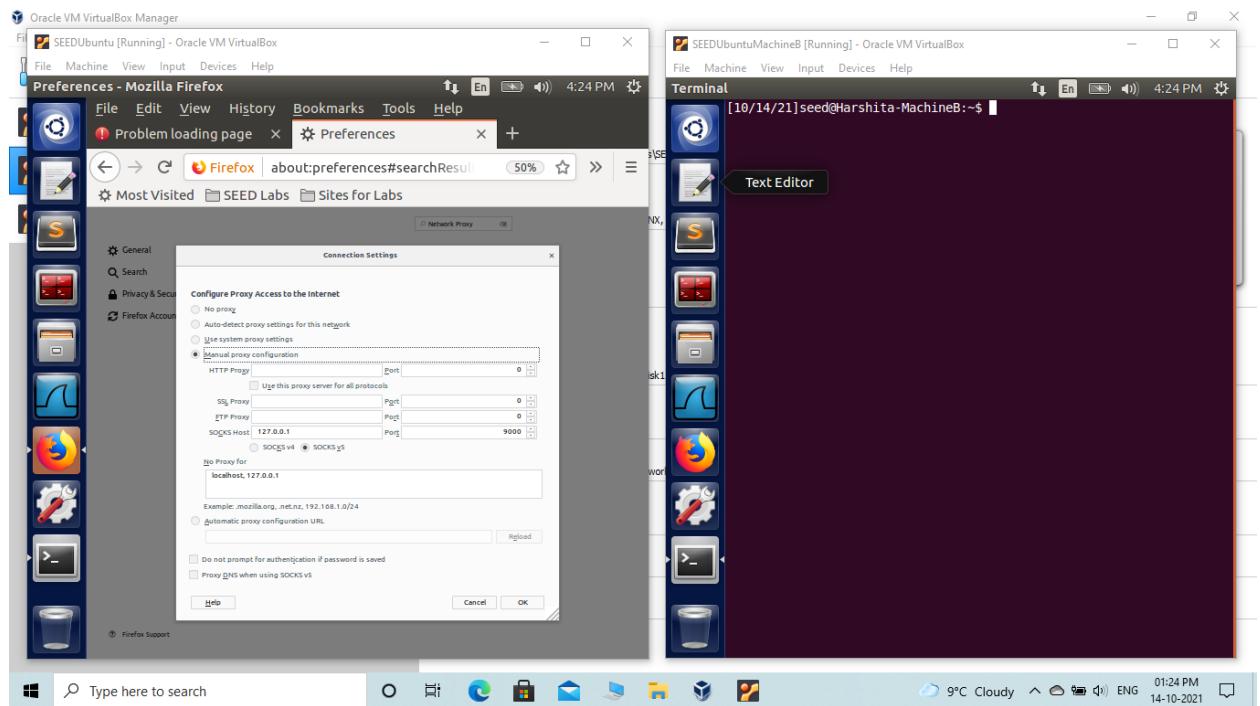
Now, we are connecting to Machine B.



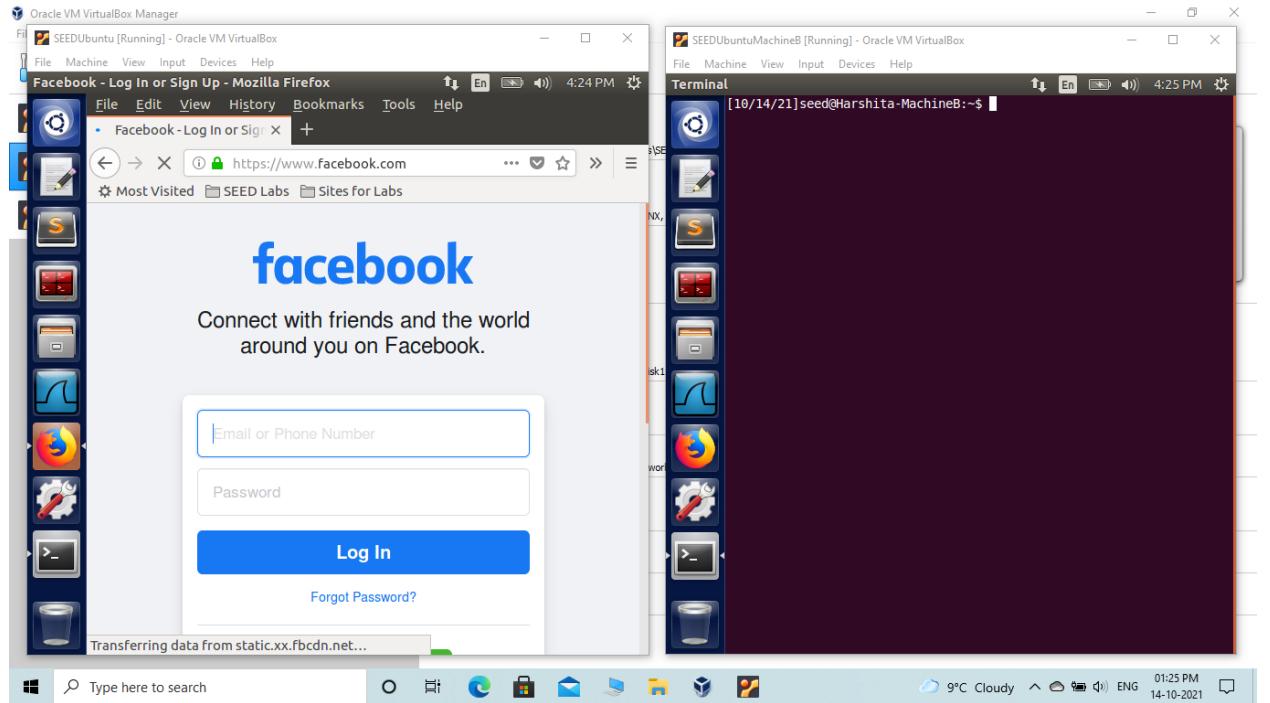
When we tried to access facebook on firefox, we are unable to access it.



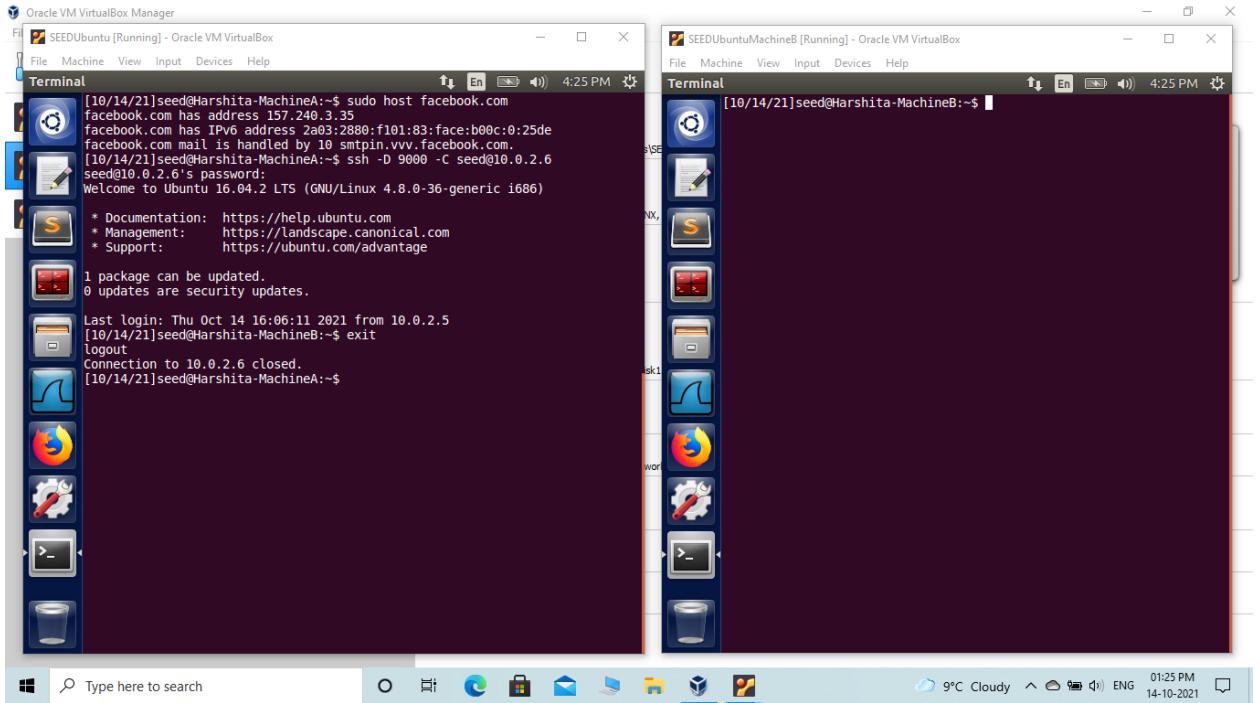
Now, we change the network proxy settings.



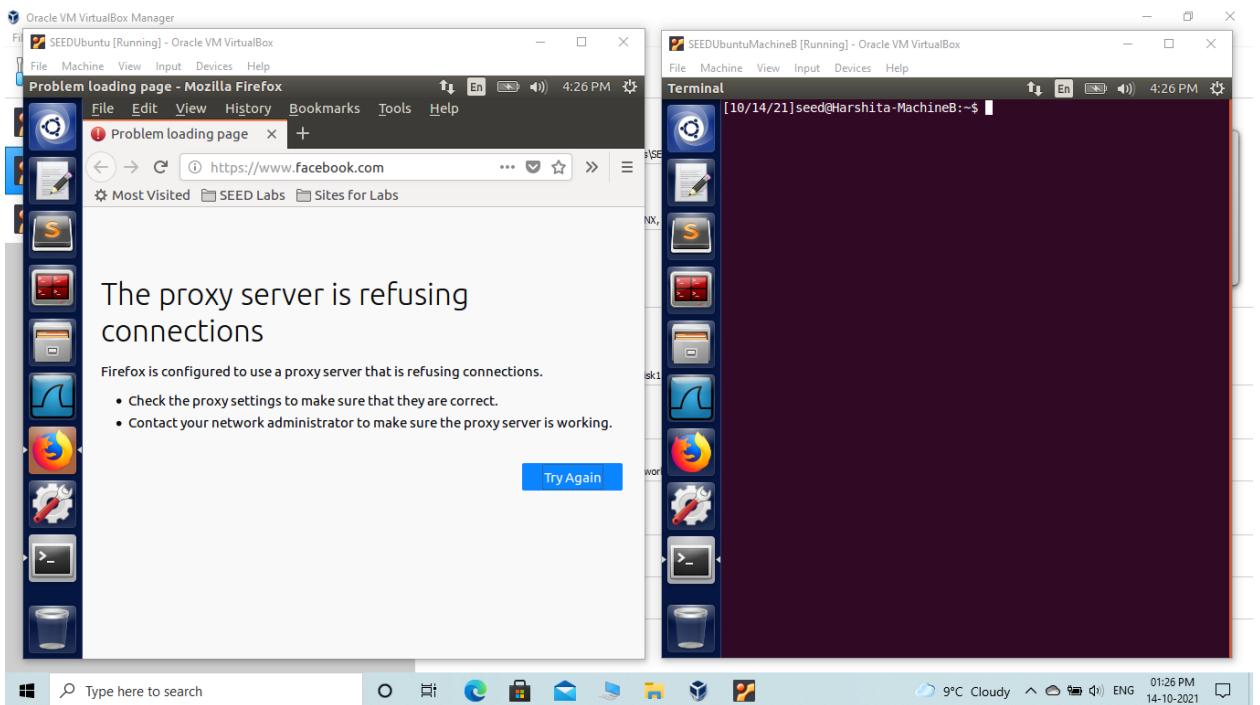
Now, we are able to access the facebook.



We exit and close the connection.

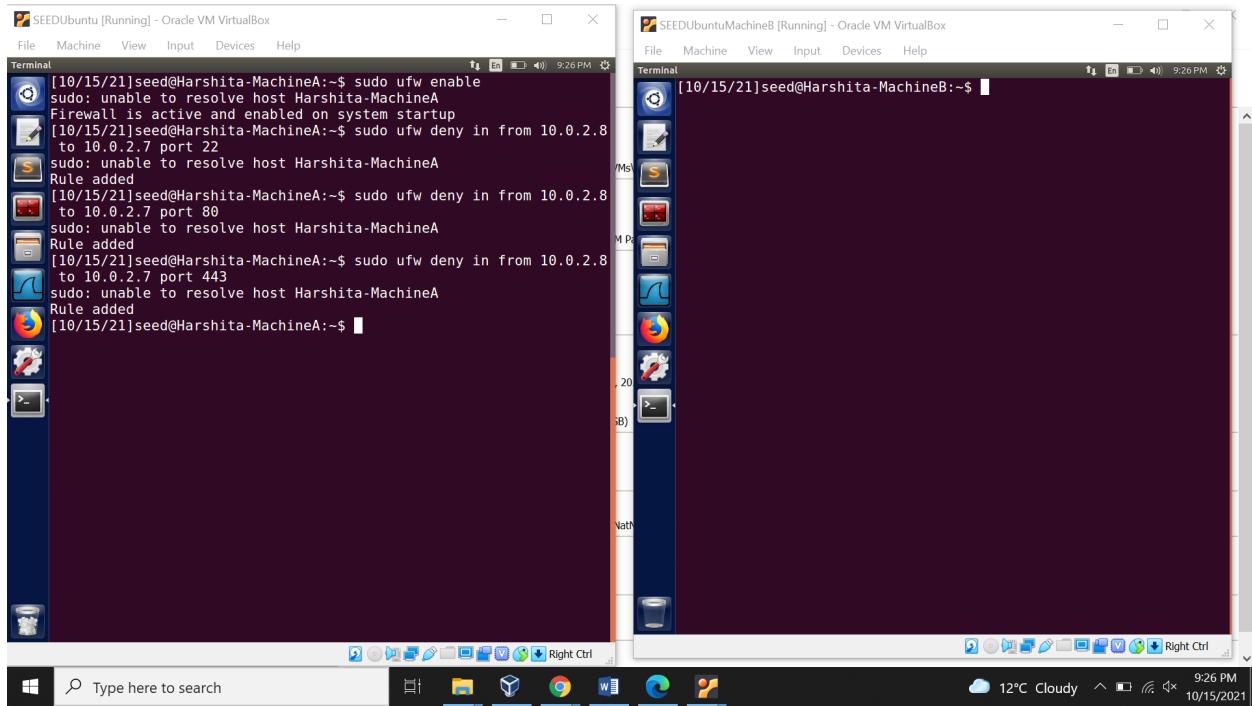


Again, we tried to access the facebook, but we are unable to access it again.

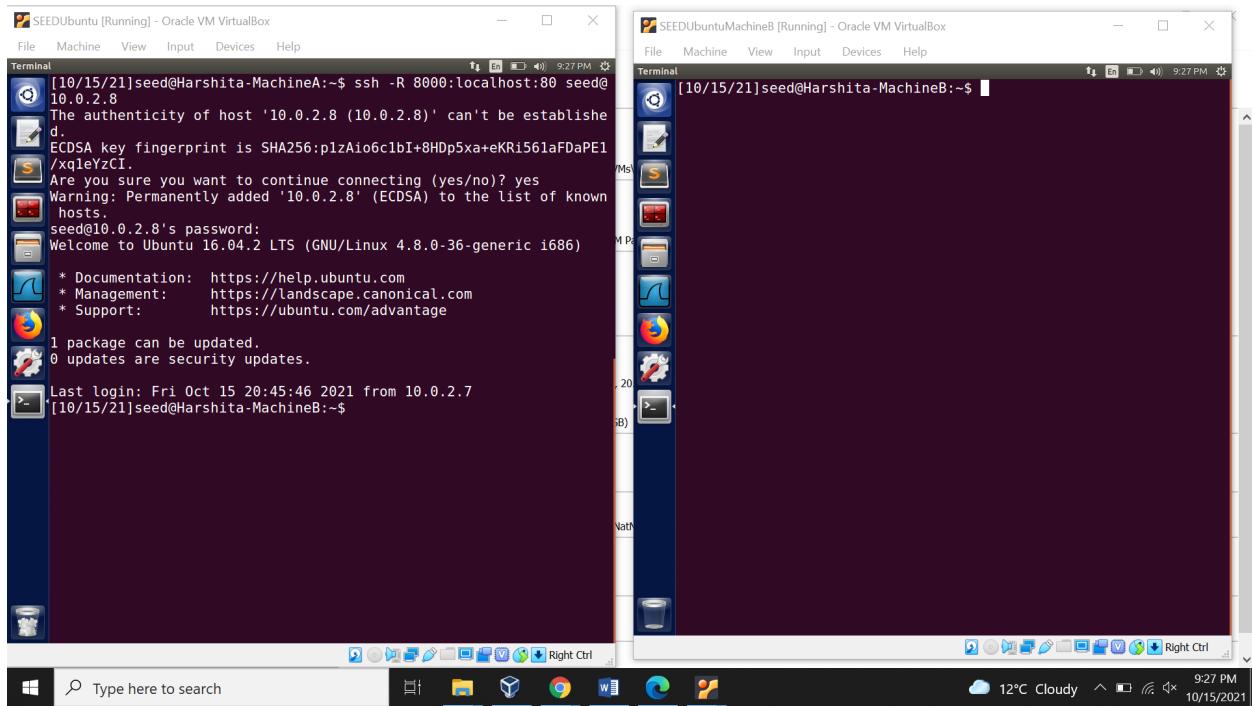


Task 4: Evading Ingress Filtering

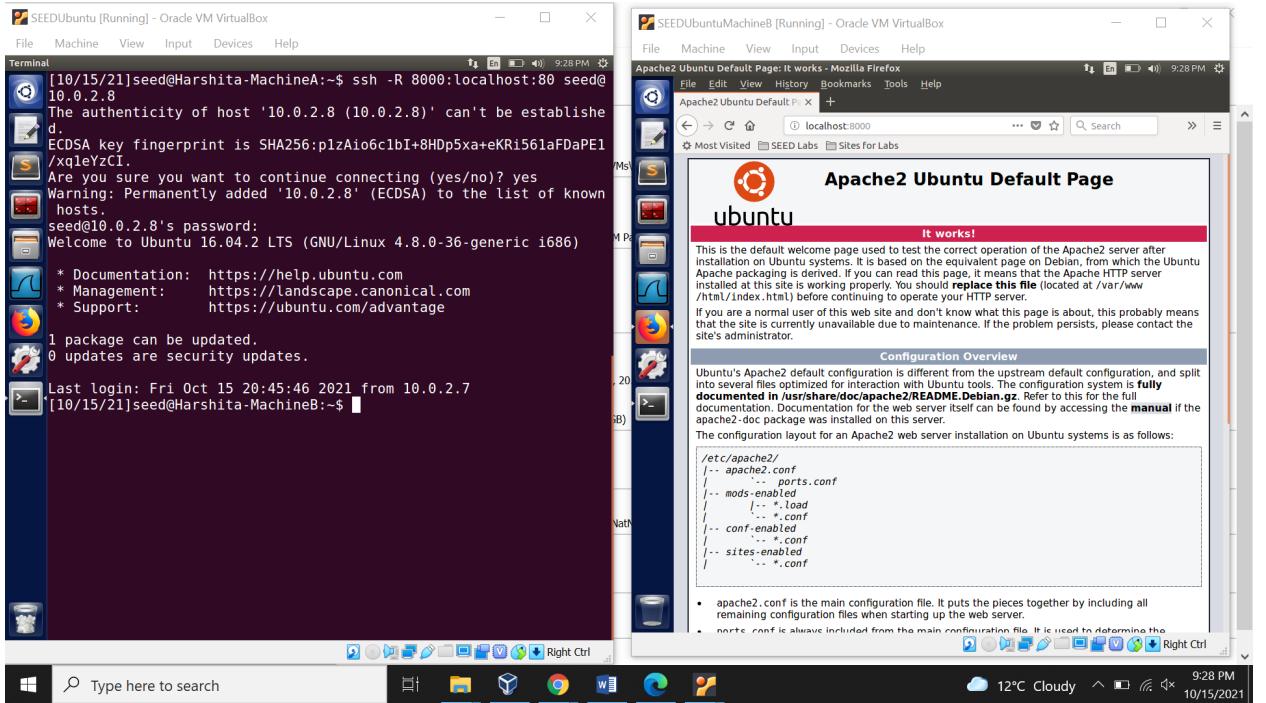
We block Machine B from accessing its port 80 (web server), 22 (SSH server) and port 443.



Set up a reverse SSH tunnel on Machine A.



Accessing the web server on Machine A from Machine B.



Appendix:

Code for the file fwcode.c

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

static struct nf_hook_ops InfilterHook;
static struct nf_hook_ops OutfilterHook;

unsigned int filter_func(void *priv, struct sk_buff *skb,
const struct nf_hook_state *state)
{
    struct iphdr *iph;
    struct tcphdr *tceph;
    iph = ip_hdr(skb);
```

```

tcpiph = (void*)iph + iph -> ihl*4;
if(iph -> protocol == IPPROTO_TCP && tcpiph -> dest == htons(23) && iph -> saddr ==
in_aton("10.0.2.6"))
{
    printk("Preventing Machine B from doing telnet to Machine A.\n");
    return NF_DROP;
}
else if(iph -> protocol == IPPROTO_ICMP && iph -> saddr == in_aton("10.0.2.6"))
{
    printk("Preventing ping from Machine B.\n");
    return NF_DROP;
}
else if(iph -> protocol == IPPROTO_ICMP && iph -> saddr == in_aton("10.0.2.7"))
{
    printk("Preventing ping from Machine C.\n");
    return NF_DROP;
}
else
{
    return NF_ACCEPT;
}
}

```

```

unsigned int Outfilter_func(void *priv, struct sk_buff *skb,
const struct nf_hook_state *state)
{
    struct iphdr *iph;
    struct tcphdr *tcpiph;
    iph = ip_hdr(skb);
    tcpiph = (void*)iph + iph -> ihl*4;
    if(iph -> protocol == IPPROTO_TCP && tcpiph -> dest == htons(23) && iph -> saddr ==
in_aton("10.0.2.5"))
    {
        printk("Preventing Machine A from doing telnet to Machine B.\n");
        return NF_DROP;
    }
    else if(iph -> protocol == IPPROTO_ICMP && iph -> daddr == in_aton("54.39.17.109"))
    {
        printk("Preventing Machine A from reaching the external website
www.vancouverumbrella.com.\n");
        return NF_DROP;
    }
    else

```

```

    {
        return NF_ACCEPT;
    }
}

int SetUpFilter(void)
{
    printk(KERN_INFO "Register Filter\n");
    InfilterHook.hook = filter_func;
    InfilterHook.hooknum = NF_INET_PRE_ROUTING;
    InfilterHook(pf = PF_INET;
    InfilterHook.priority = NF_IP_PRI_FIRST;
    nf_register_net_hook(&init_net, &InfilterHook);
    OutfilterHook.hook = Outfilter_func;
    OutfilterHook.hooknum = NF_INET_POST_ROUTING;
    OutfilterHook(pf = PF_INET;
    OutfilterHook.priority = NF_IP_PRI_FIRST;
    nf_register_net_hook(&init_net, &OutfilterHook);
    return 0;
}

void RemoveFilter(void)
{
    printk(KERN_INFO "Filter is removed.\n");
    nf_unregister_net_hook(&init_net, &InfilterHook);
    nf_unregister_net_hook(&init_net, &OutfilterHook);
}

module_init(SetUpFilter);
module_exit(RemoveFilter);

```

Code for Makefile:

```

obj-m += fwcode.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

```