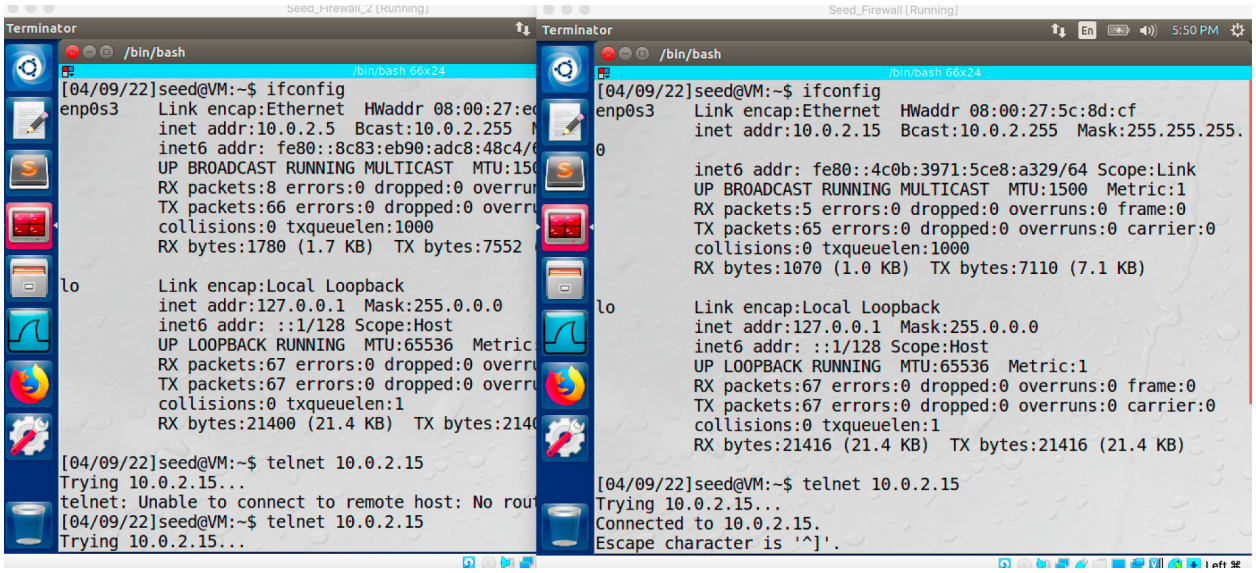
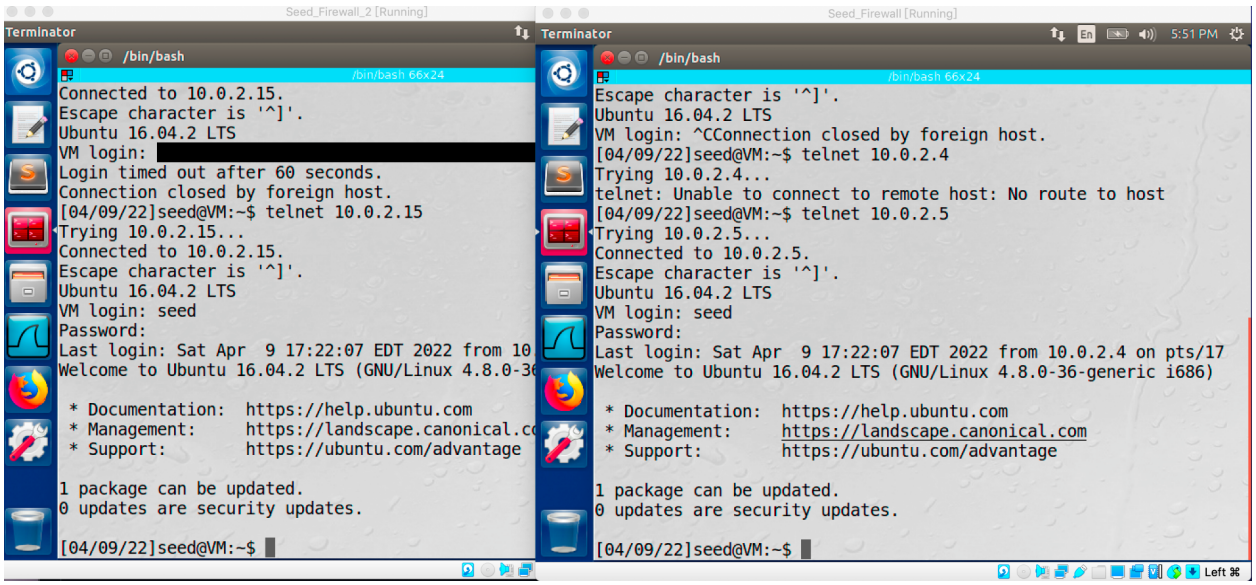
Task 1:

- VM A: 10.0.2.5
- VM B: 10.0.2.15

a. *The left screenshot is VM A, and the right one is VM B*

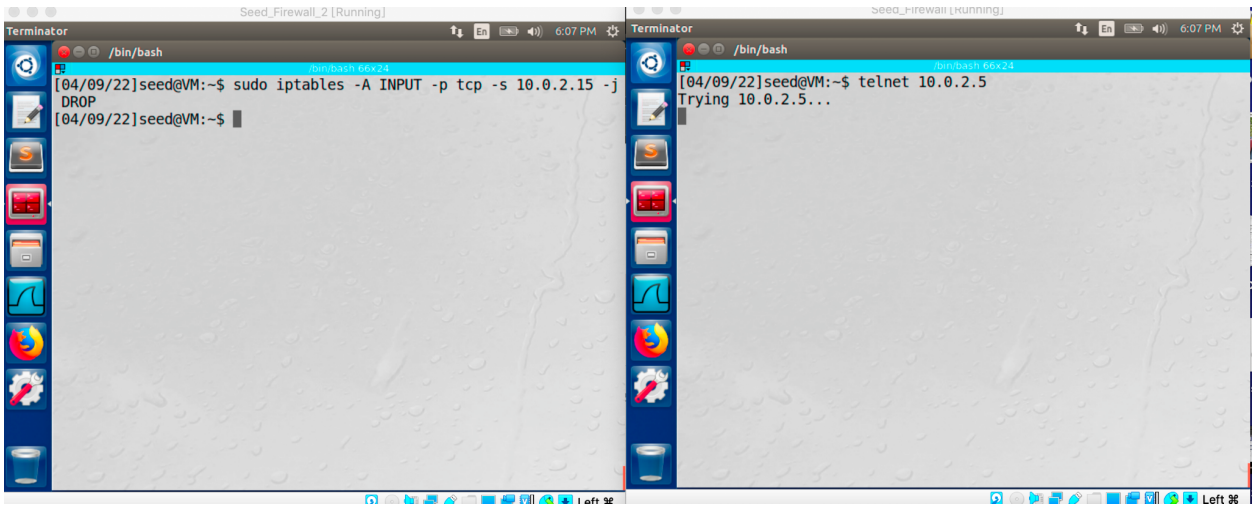


b. *The below screenshot shows that both VM A(10.0.2.5) and VM B (10.0.2.15) do not configure firewall policy, and both of them can access each other via telnet and vice versa.*



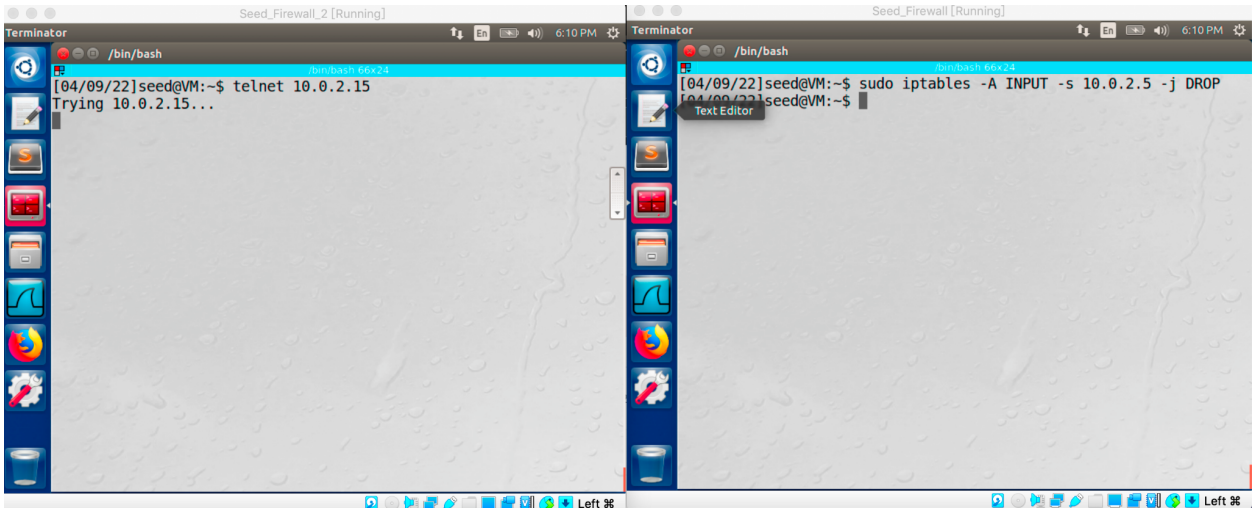
Prevent B from doing telnet to Machine A

c. *Based on the iptables firewall policy, it is used to prevent VM B (10.0.2.15) from doing telnet to VM A (10.0.2.5)*



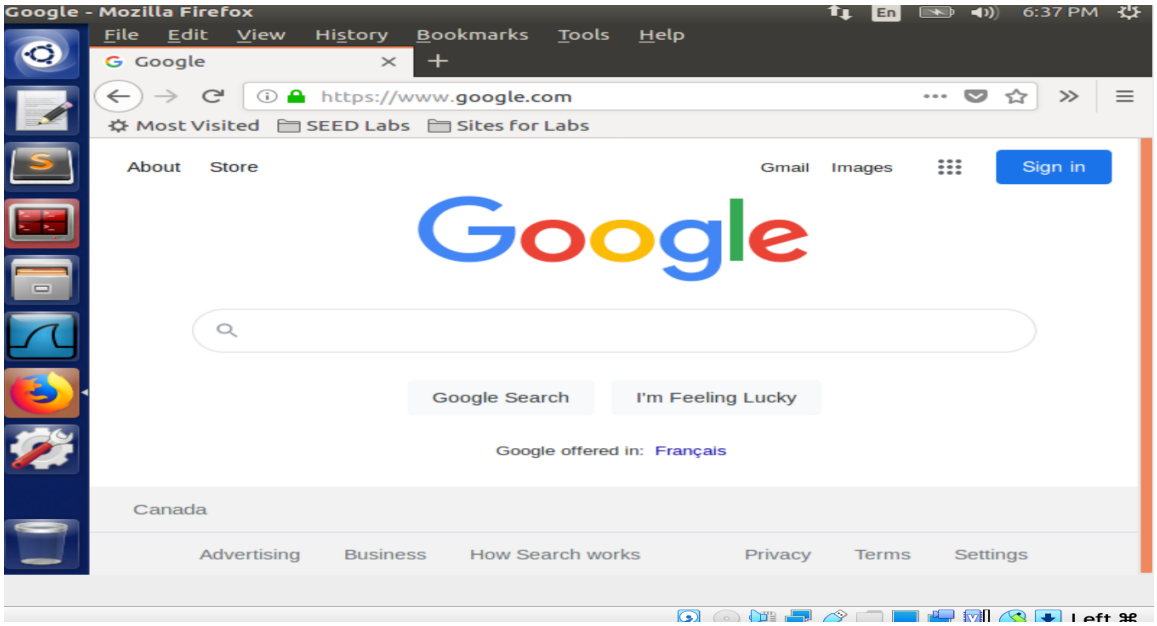
Prevent A from doing telnet to Machine B

d. *Based on the iptables firewall policy, it is used to prevent VM A (10.0.2.5) from doing telnet to VM B (10.0.2.15)*

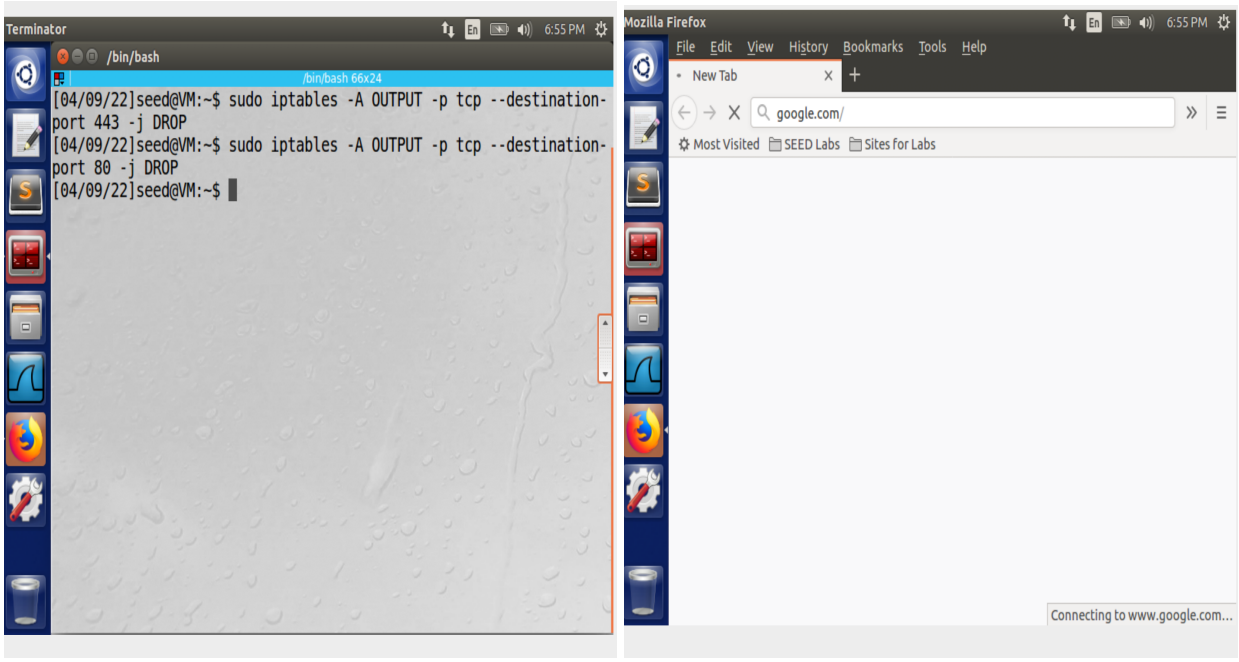


Prevent A from visiting an external website

e. *On the VMA (10.0.2.15), it can access the google website without configuring the firewall policy.*



f. *If VM A (10.0.2.15) configures the iptables firewall policies in terms of the left screenshot, accessing the google website will be loading continually that means can not access.*

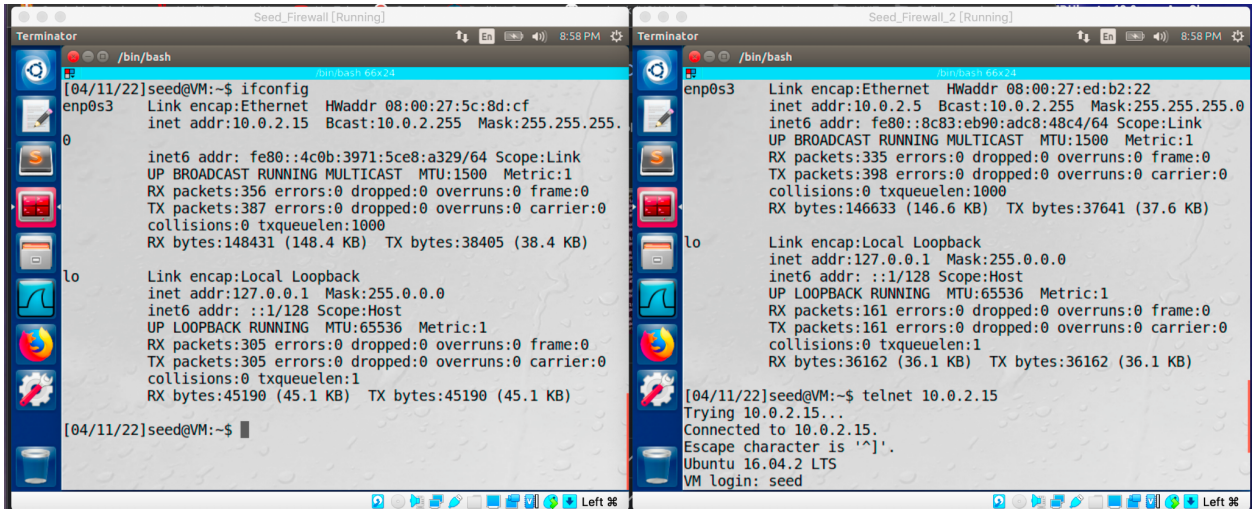


Task 3

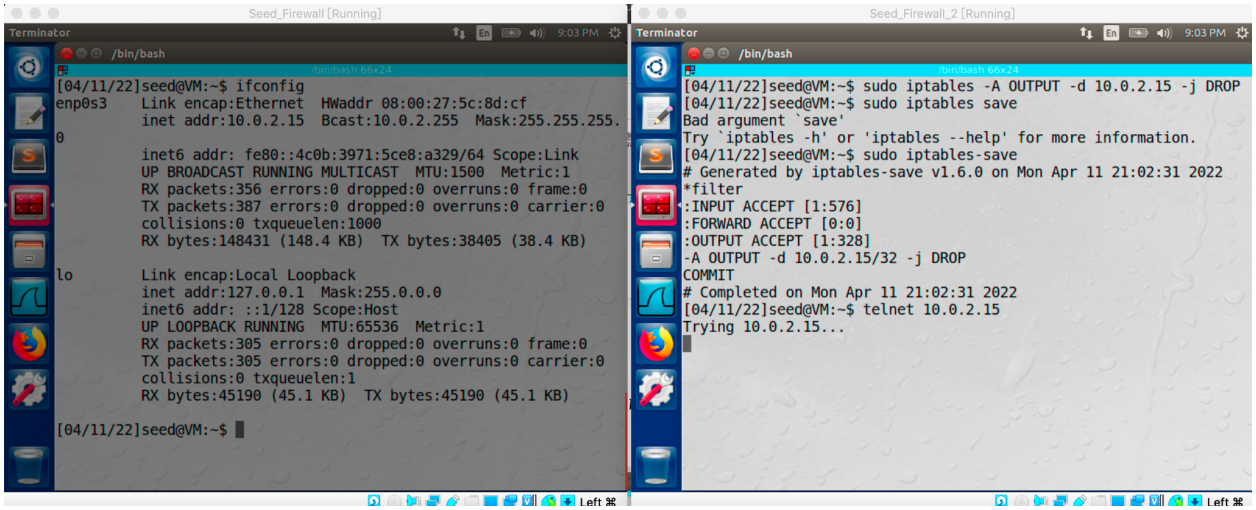
Block all the outgoing traffic to external telnet servers

- VM A: 10.0.2.5
- VM B: 10.0.2.15

g. *On the VM A (10.0.2.5) in the right screenshot, it can access VM B(10.0.2.15) via telnet without configuring the firewall policy.*



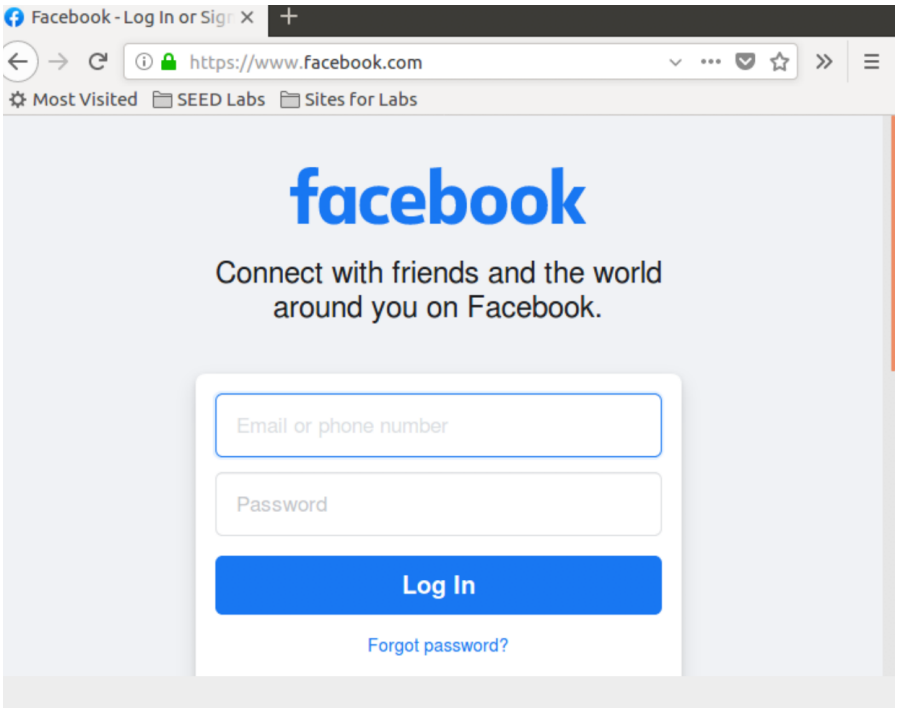
h. *After configuring the firewall policy which blocks outgoing packets to the external telnet, VM A (10.0.2.5) can not access VM B(10.0.2.15).*



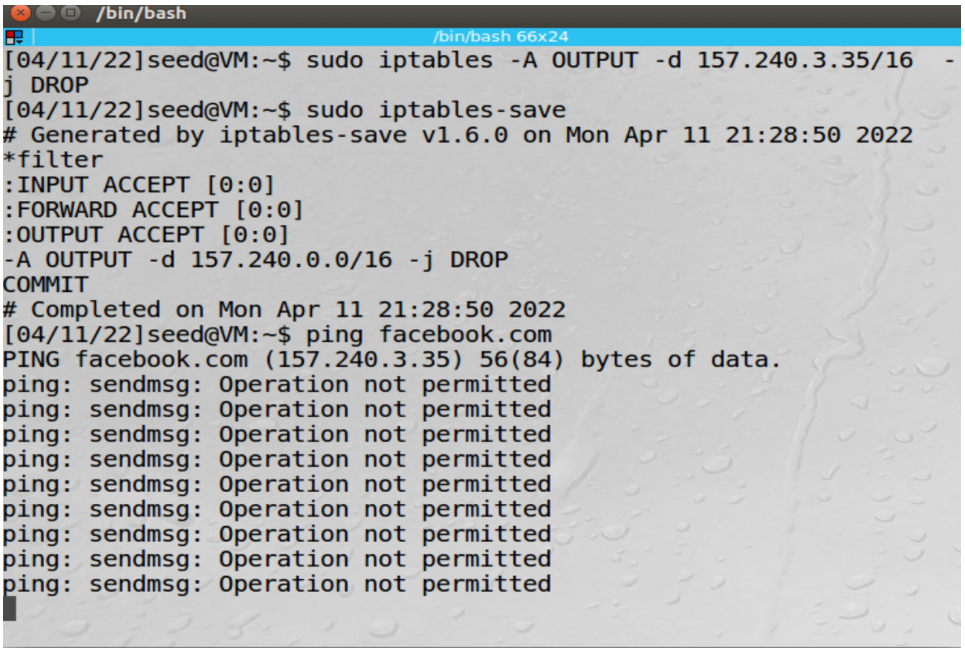
Block all the outgoing traffic to *www.facebook.com* or Static IP like *www.syr.edu*

- VM A: 10.0.2.5
- VM B: 10.0.2.15

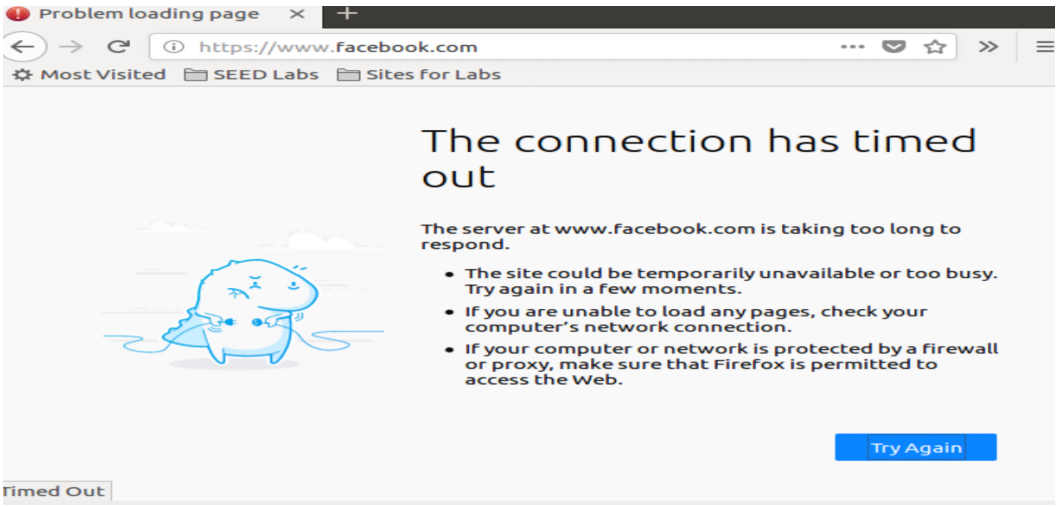
i. *On VM A(10.0.2.5), it can access the normal Facebook page without blocking outgoing packets.*



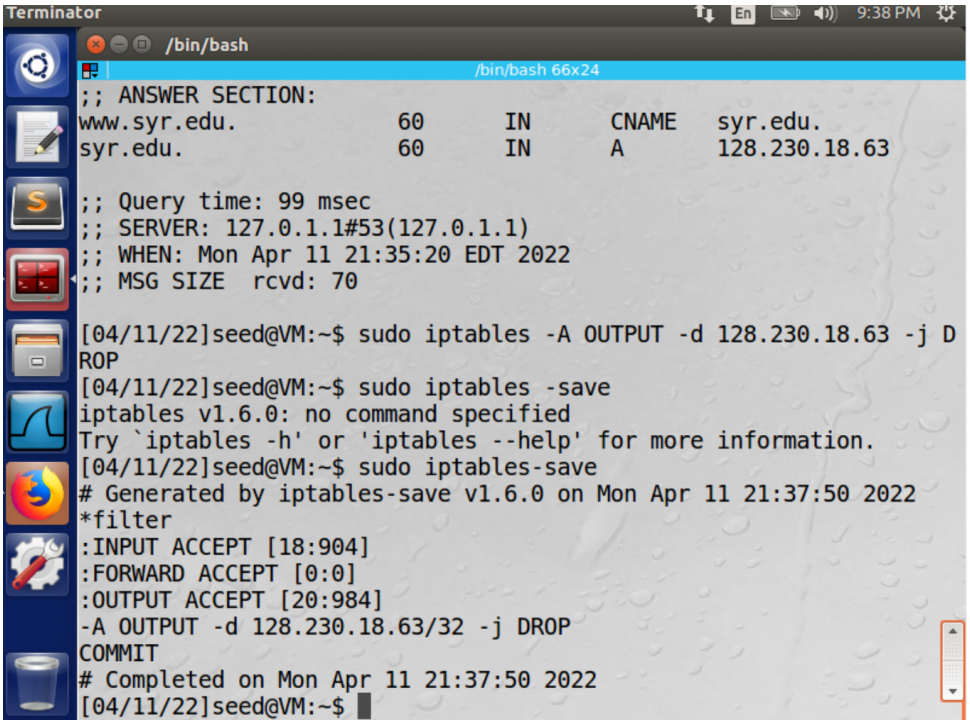
j. *Find the Facebook IP via "dig" command, and then block the Facebook IP(157.240.3.35). Considering Facebook has many IP addresses, it can block with submask 16 (255.255.0.0) to ensure this can block all Facebook IP. After configuring the firewall policy, use "ping" to check the connection with facebook.com.*



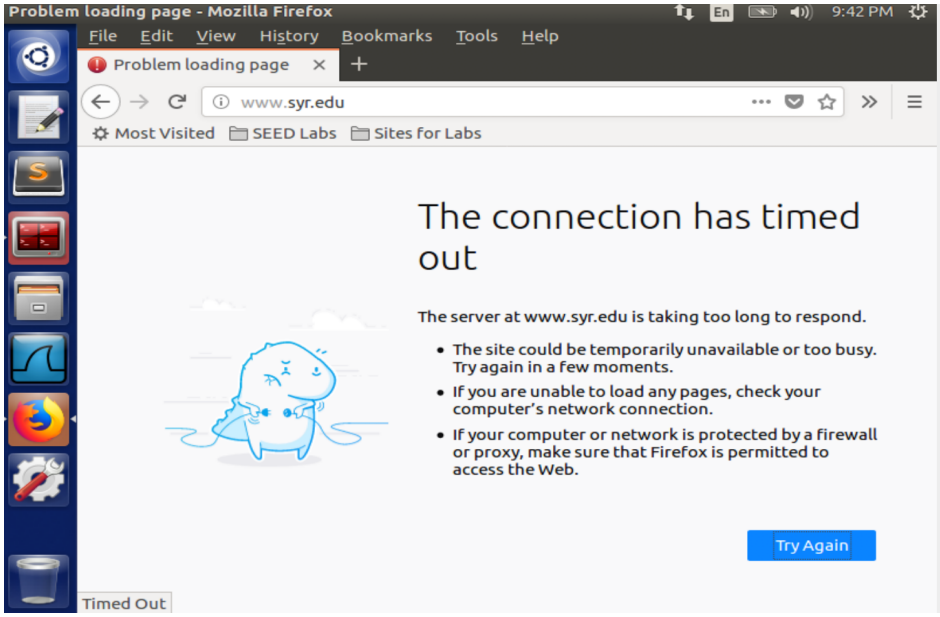
k. *Ths below screenshot shows the firewall policy is configuring properly and VM can not access Facebook.*



l. *From the lab description, the website www.syr.edu has a static IP (128.230.18.63), so we can also block this IP.*



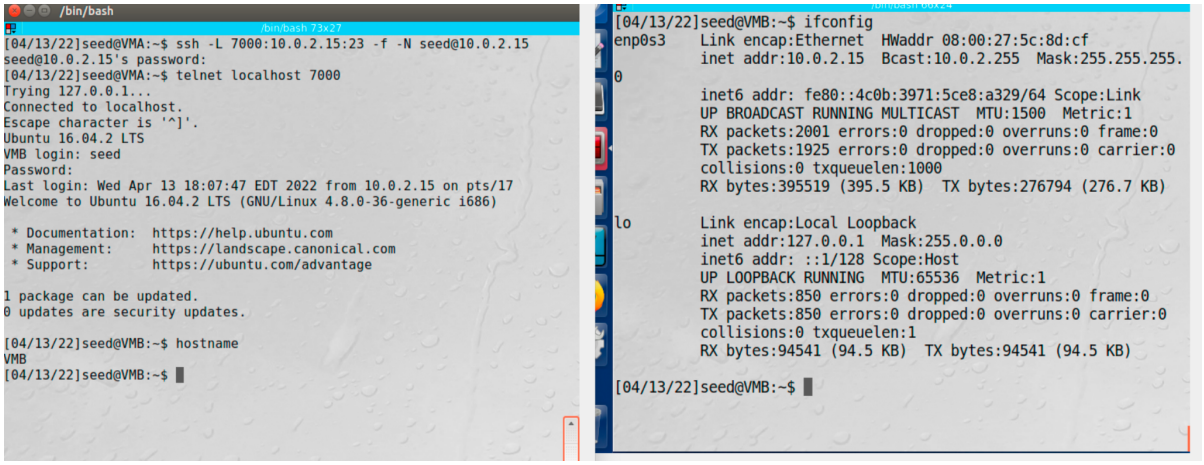
m. *The below screenshot shows the firewall policy works successfully.*



Task 3a. Telnet to Machine B through the firewall

- VM A: 10.0.2.5
- VM B: 10.0.2.15

n. *Considering it only has two virtual machines, VM B is used for the telnet and ssh server, and VM A is used to be a remote client.*



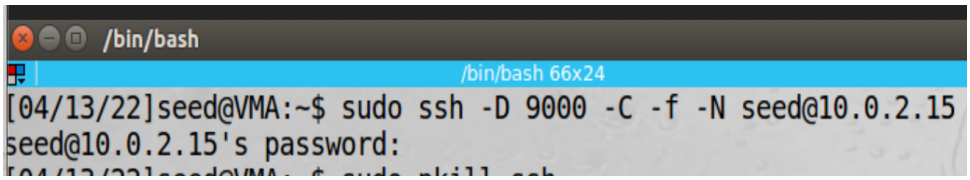
o. *After establishing an ssh tunnel, VM A(10.0.2.5) telnet VM B(10.0.2.15). Using Wireshark captures the packet which can see the communication is use SSH protocol.*

	Source	Destination	Protocol	Length
-04-13 18:10:30.0734319…	10.0.2.5	10.0.2.15	SSH	158
-04-13 18:10:30.0740350…	10.0.2.15	10.0.2.5	TCP	66
-04-13 18:10:30.0742109…	10.0.2.15	10.0.2.5	SSH	110

Task 3b. Connect to Facebook using SSH Tunnel

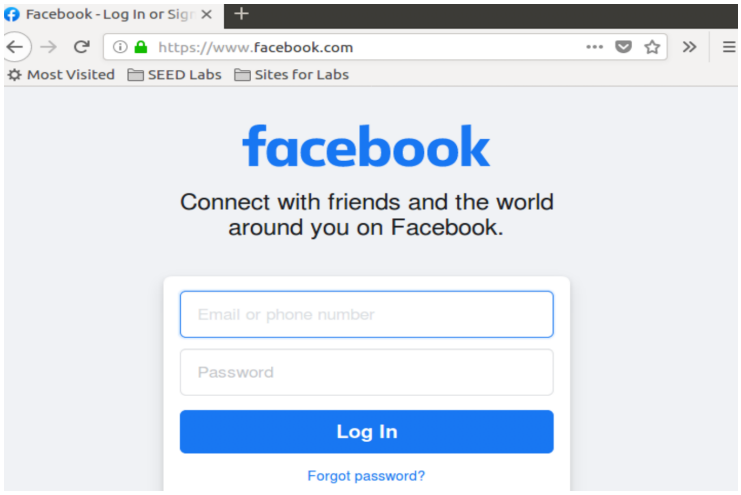
- VM A: 10.0.2.5
- VM B: 10.0.2.15
- Configure ssh dynamic port forwarding and socks proxy in the browser

p. *Establish an ssh tunnel between VMA and VMB, which can use this tunnel to forward packets.*



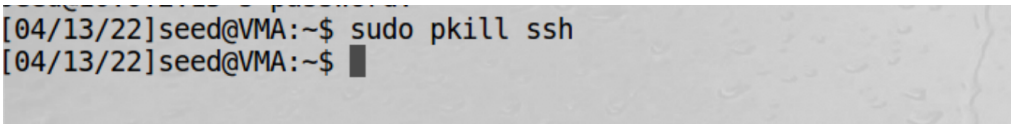
- Run Firefox and go visit the Facebook page

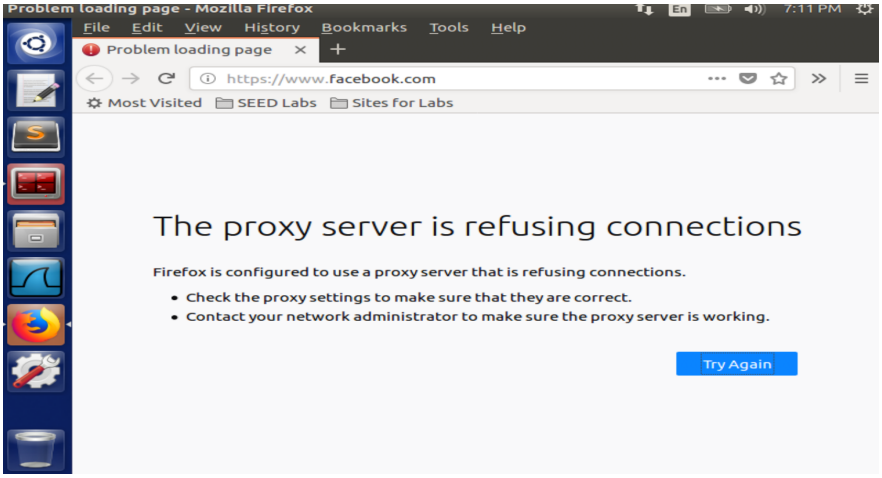
q. *Access successfully*



- Break the SSH tunnel, clear the Firefox cache, and try the connection again

r. *Using "pkill" command to kill all ssh sessions, and then accessing the Facebook page will be refused by the proxy server (127.0.0.1:9000).*





- Establish the SSH tunnel again and connect to Facebook

    s. *Reconnect the ssh session to VM B(10.0.2.15); facebook.com can be accessed.*

- Explain why the SSH tunnel can help bypass the egress filtering and capture the packets with Wireshark.

    t. *Facebook Public IP (157.240.22.35).*

    u. *Remote side: VM A (10.0.2.5) firstly connect to VM B (10.0.2.15) via an ssh tunnel, and then VM B will forward the packet to the Facebook page establishing TCP 3 ways handshake.*

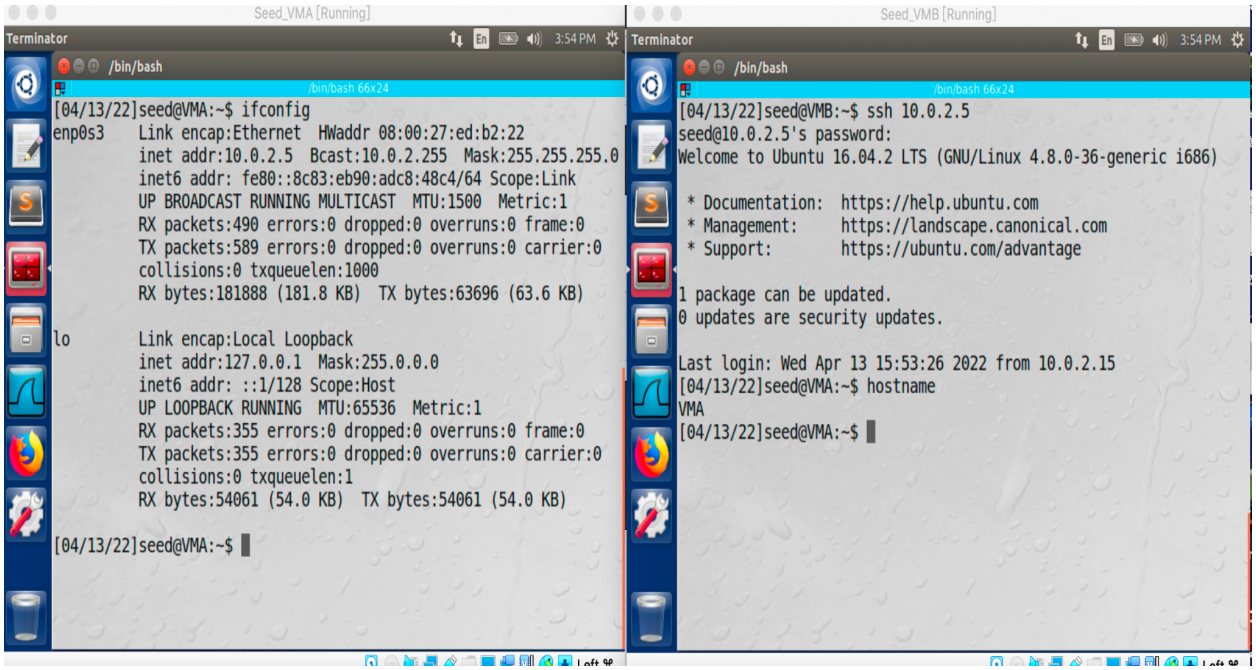
```
10.0.2.5        10.0.2.15       SSH      110 Client: Encrypted packet (le
10.0.2.15       157.240.22.35   TCP       74 58628 → 443 [SYN] Seq=275391
10.0.2.15       10.0.2.5        TCP       66 22 → 41288 [ACK] Seq=1450762
157.240.22.35   10.0.2.15       TCP       60 443 → 58628 [SYN, ACK] Seq=1
10.0.2.15       157.240.22.35   TCP       60 58628 → 443 [ACK] Seq=275391
10.0.2.15       10.0.2.5        SSH      102 Server: Encrypted packet (le
10.0.2.5        10.0.2.15       TCP       66 41288 → 22 [ACK] Seq=1030764
10.0.2.5        10.0.2.15       SSH      286 Client: Encrypted packet (le
10.0.2.15       10.0.2.5        TCP       66 22 → 41288 [ACK] Seq=1450762
```

Task 4:

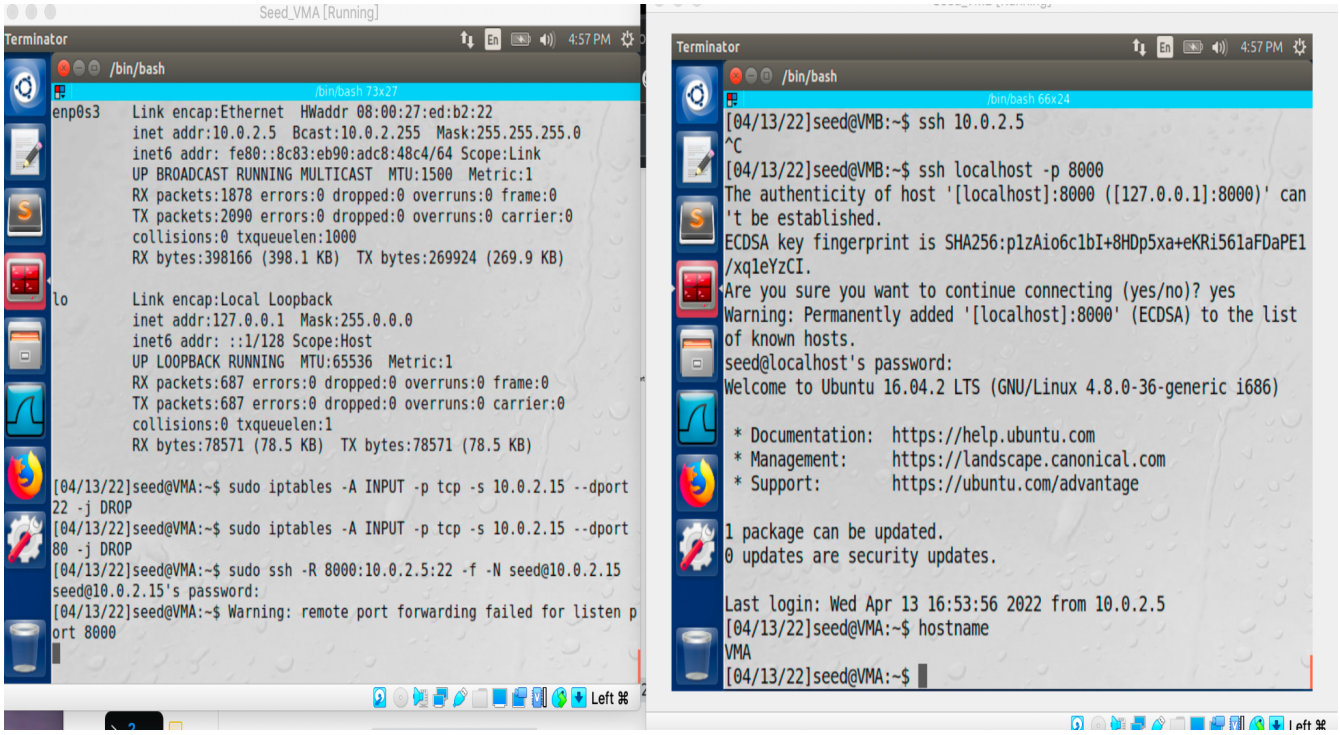
Reverse SSH tunnel

- The goal of this task is to set up a reverse SSH tunnel on VM A that is protected by the firewall. VM A configures firewall policy with blocking ports 22 and 80. And, this VM A is accessible from the outside VM B.
- VM A: 10.0.2.5
- VM B: 10.0.2.15

v. *Without configuring blocking ports 80,22 and establishing an ssh reverse tunnel, VM B can ssh to VM A.*



w. *Blocking ports 80,22 and establishing an ssh reverse tunnel on VMA, which allows VMB ssh to VMA.*



Based on the configuration, the below architecture shows how does the ssh reverse tunnel work.