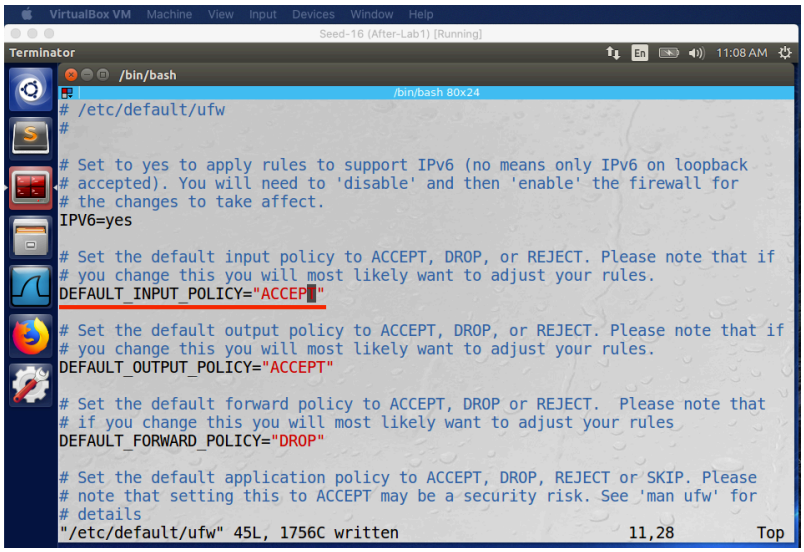
Name: Bharath Karumudi
Lab: Linux Firewall Exploration

Task 1: Using Firewall



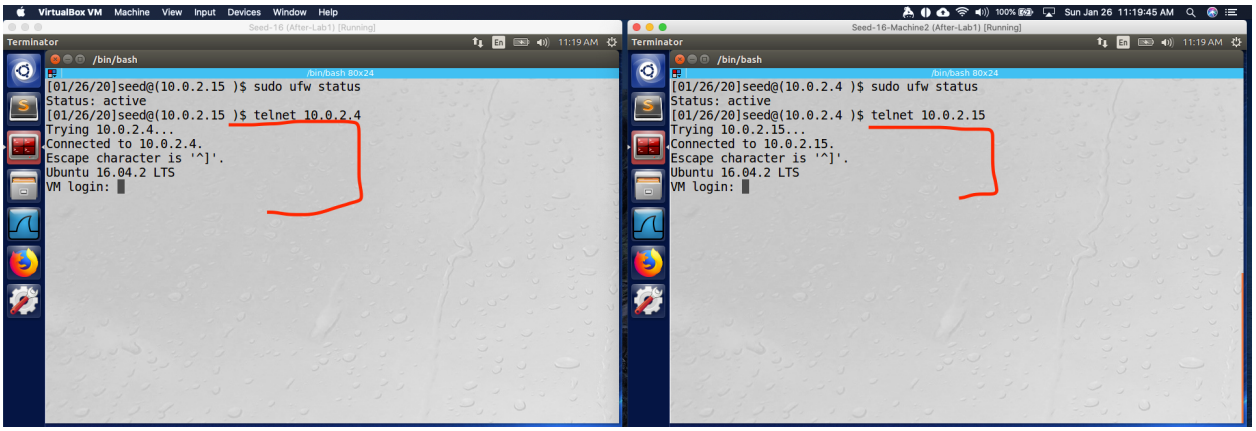


Fig: Telnet was working before modifying the firewall rules between two machines

A. Prevent A from doing telnet to Machine B

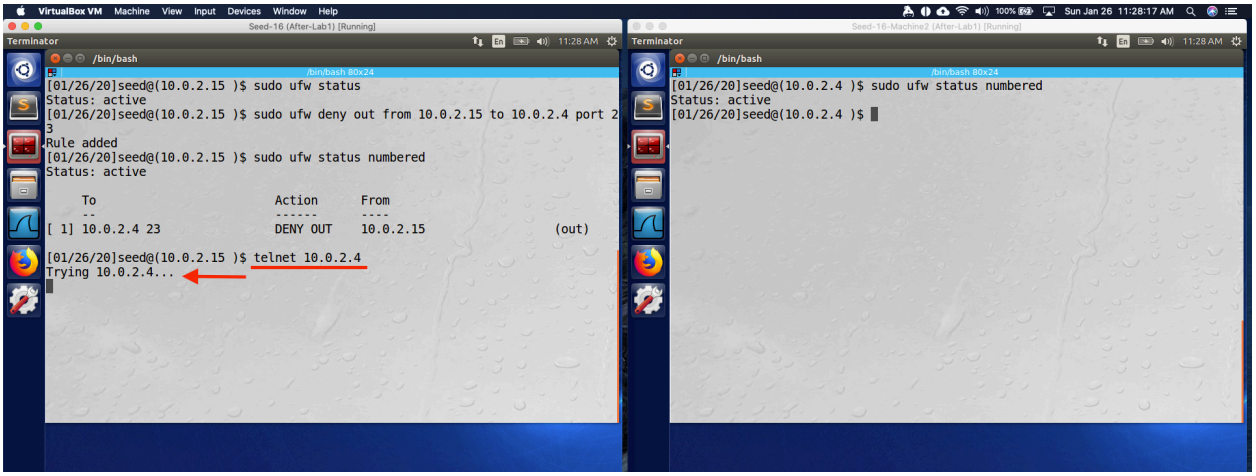


Fig: Telnet blocked from Machine A to Machine B; with a firewall rule on Machine A.

B. Prevent B from doing telnet to Machine A

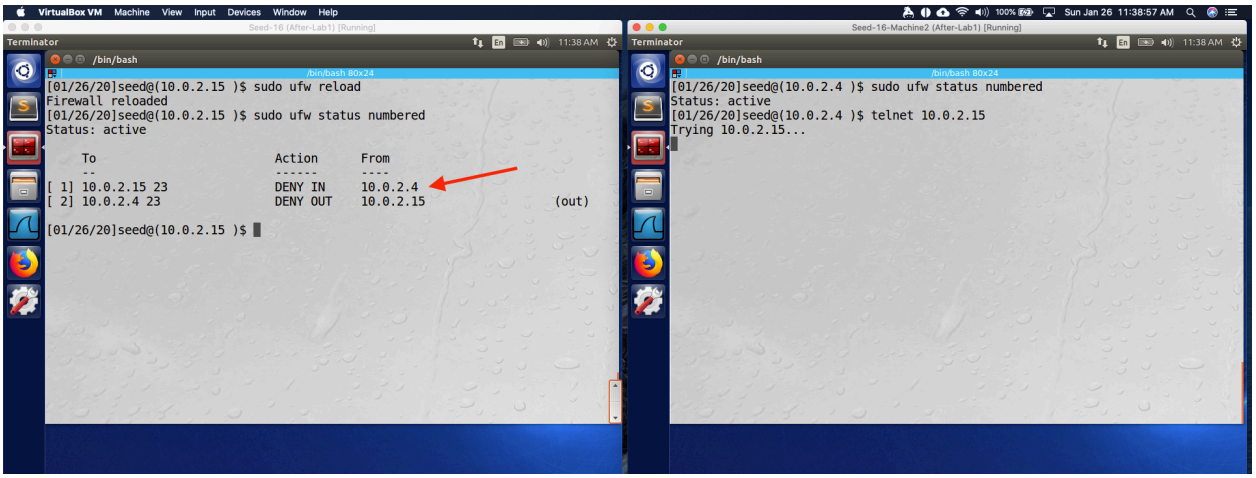


Fig: Telnet blocked from Machine B to Machine A; with a firewall rule on Machine A.

C. Prevent A from visiting an external web site

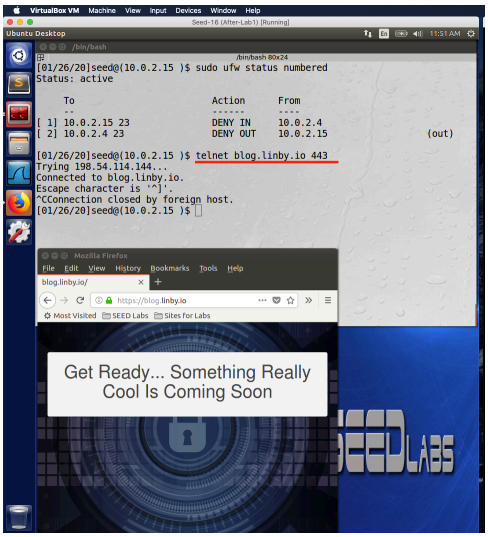


Fig: External website is working before making the firewall change

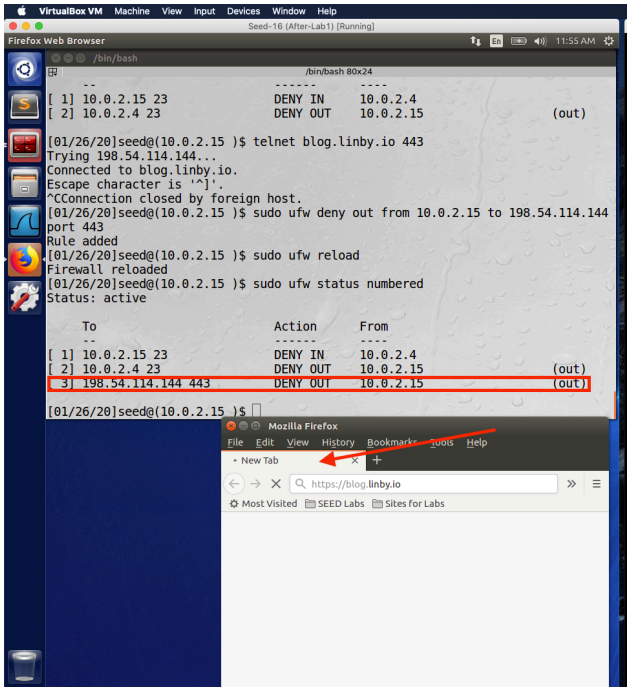


Fig: External website was blocked from Machine A after adding the rule to Firewall.

Observation: Without modifying the firewalls, I was able to do the telnet between two machines and also able to visit the external website. After modifying the firewalls using ufw to deny the traffic, I was unable to.

Explanation: ufw is a front end to the Linux inbuilt firewall – iptables. Using ufw, in case:

a. We added a rule in Machine A to deny the telnetting from Machine A to Machine B.
sudo ufw deny out from 10.0.2.15 to 10.0.2.4 port 23

b. We added a rule in Machine B to deny the telnetting from Machine B to Machine A.
sudo ufw deny out from 10.0.2.15 to 10.0.2.4 port 23

c. We added a rule in Machine A to deny vising a website (blog.linby.io) on port 443
sudo ufw deny out from 10.0.2.15 to 198.54.114.144 port 443

Using ufw, I was able to modify the firewalls and block the respective traffic.

Task 2: Implementing a Simple Firewall

A. Block telnet from Machine A to Machine B

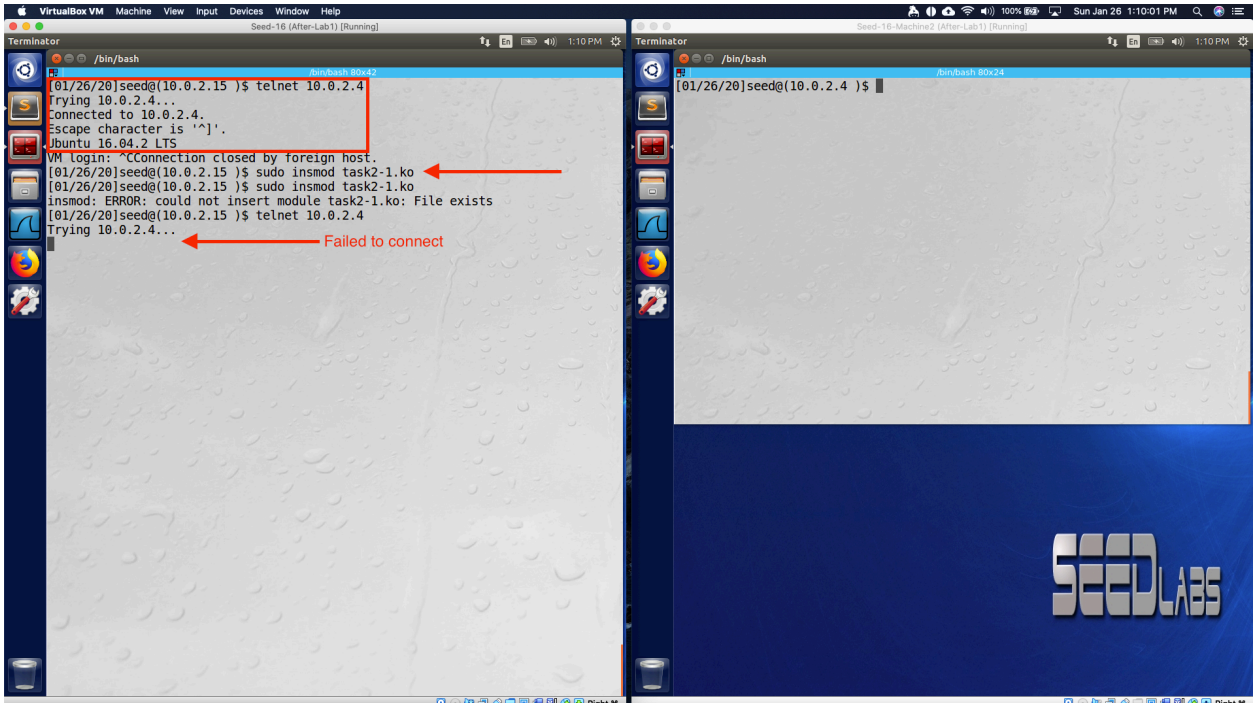


Fig: Machine A failed to connect to Machine B over telnet after module addition

B. Block telnet from Machine B to Machine A.

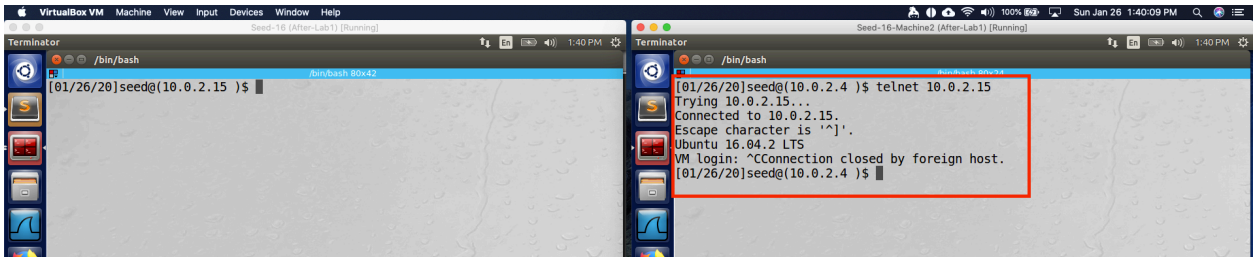


Fig: Machine B is able to connect before module addition

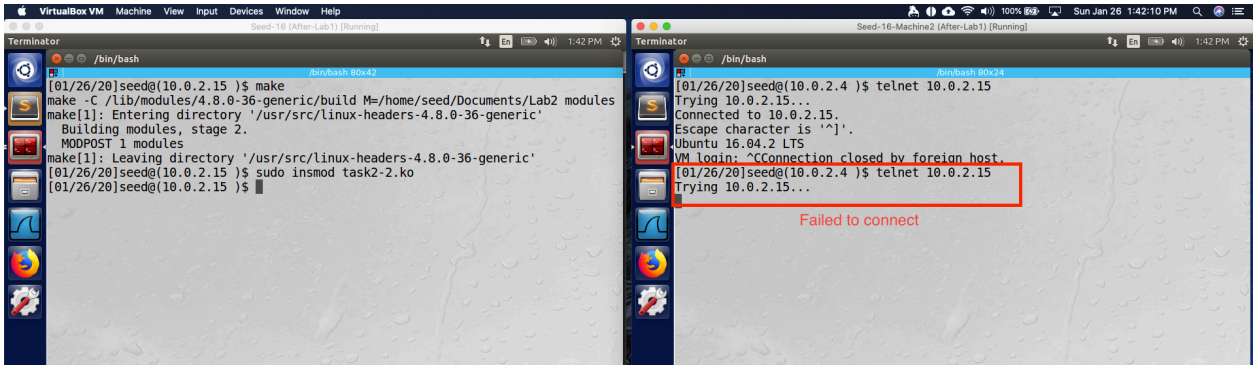


Fig: Machine B is failed to connect to Machine A over telnet

C. Block a Website on port 443

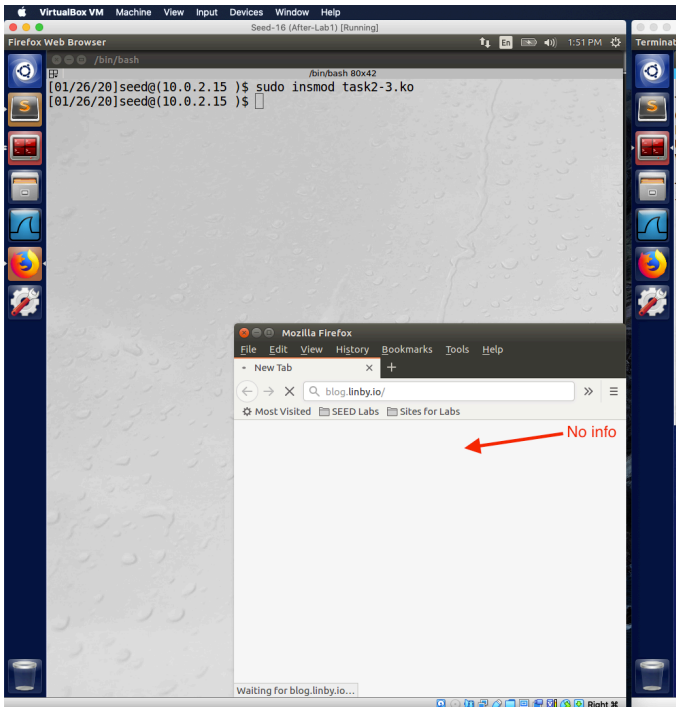


Fig: Blocked a specific website on port 443 from Machine A.

D. Block ICMP traffic

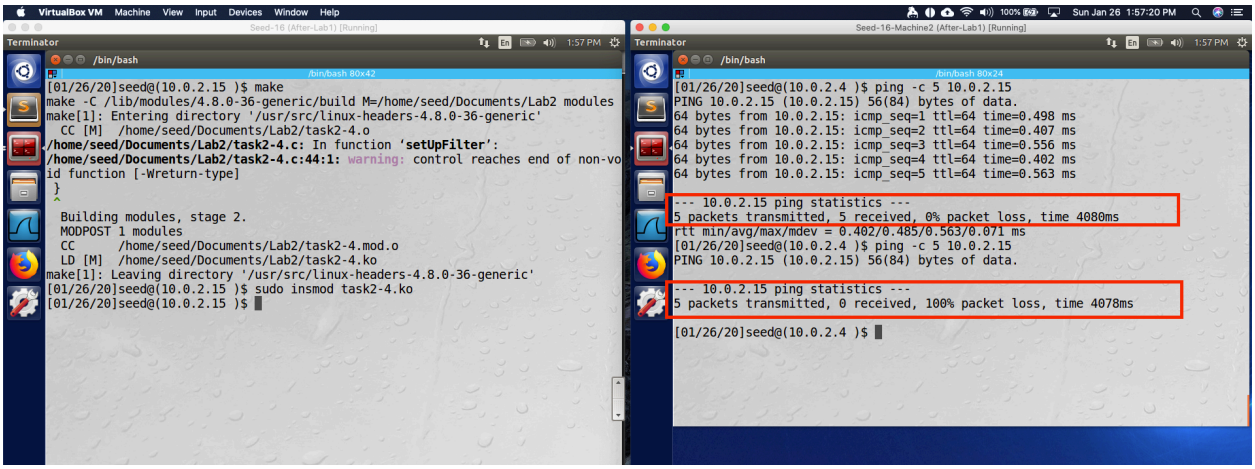


Fig: Blocking ICMP traffic from Machine B

E. Block accessing http websites (port 80)

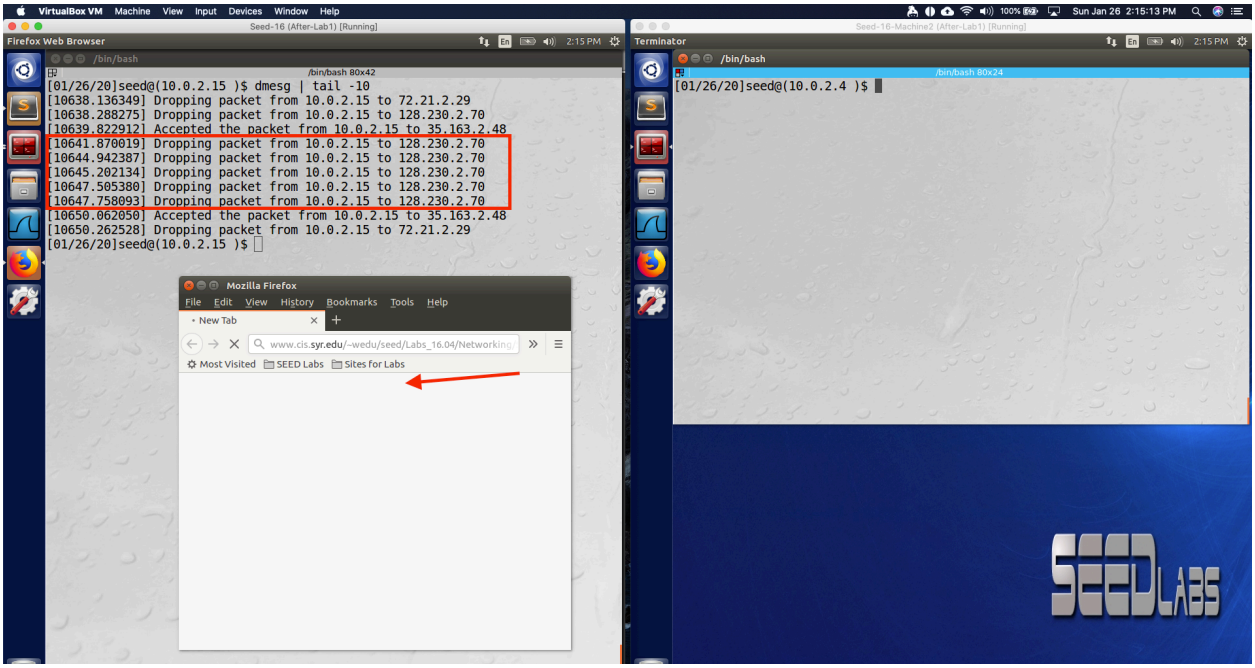


Fig: Blocking all http traffic from Machine A

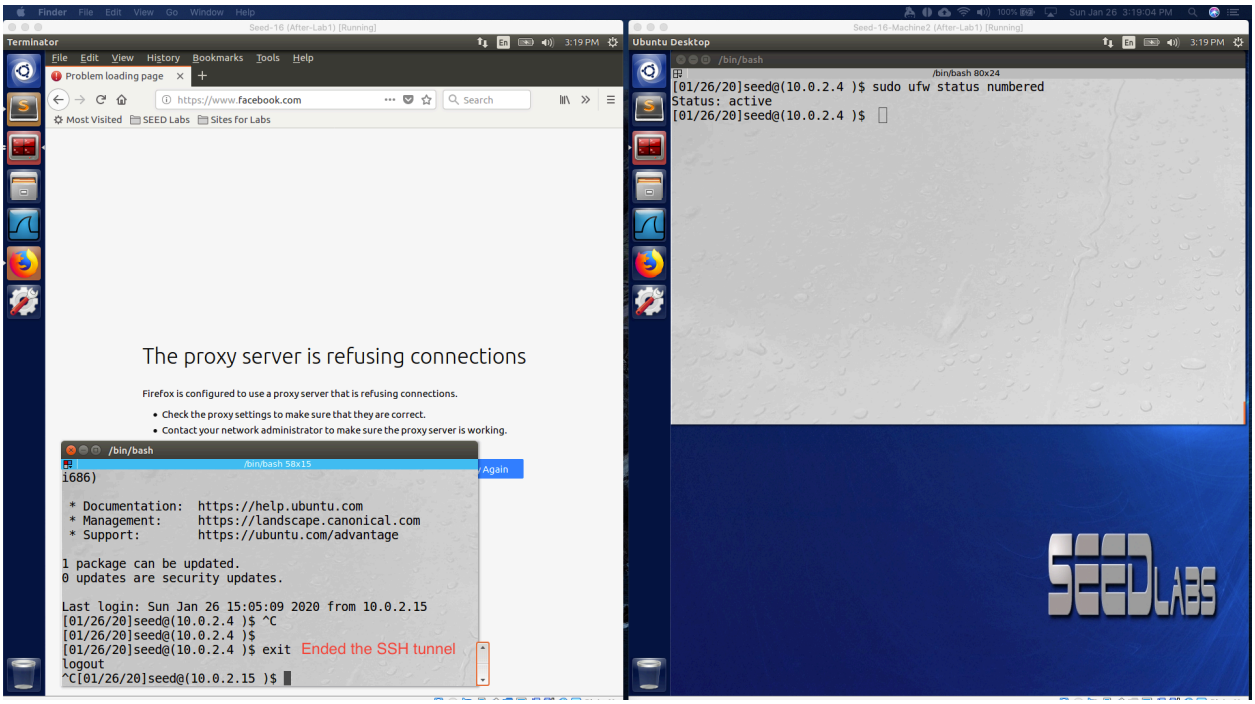


Fig: Unable to access the Facebook after terminating the SSH tunnel

Observation: Built a simple firewall using Loadable Kernel Module and Netfilter, when added the modules to the Kernel with conditions, the packets are being dropped.

Explanation: Using the Loadable Kernel Module, we can add the modules to the Kernel without compiling the whole Kernel. Also, by Netfilter, we add the hooks and registered, so our custom program executes when hook was invoked by Kernel. Using this, we blocked the traffic:

a. Blocking Telnet from Machine A to Machine B
b. Blocking Telnet from Machine B to Machine A
c. Blocking the website from Machine A
d. Blocking the ICMP traffic between Machine A and Machine B
e. Blocking all the HTTP traffic

Task 3: Evading Egress Filtering

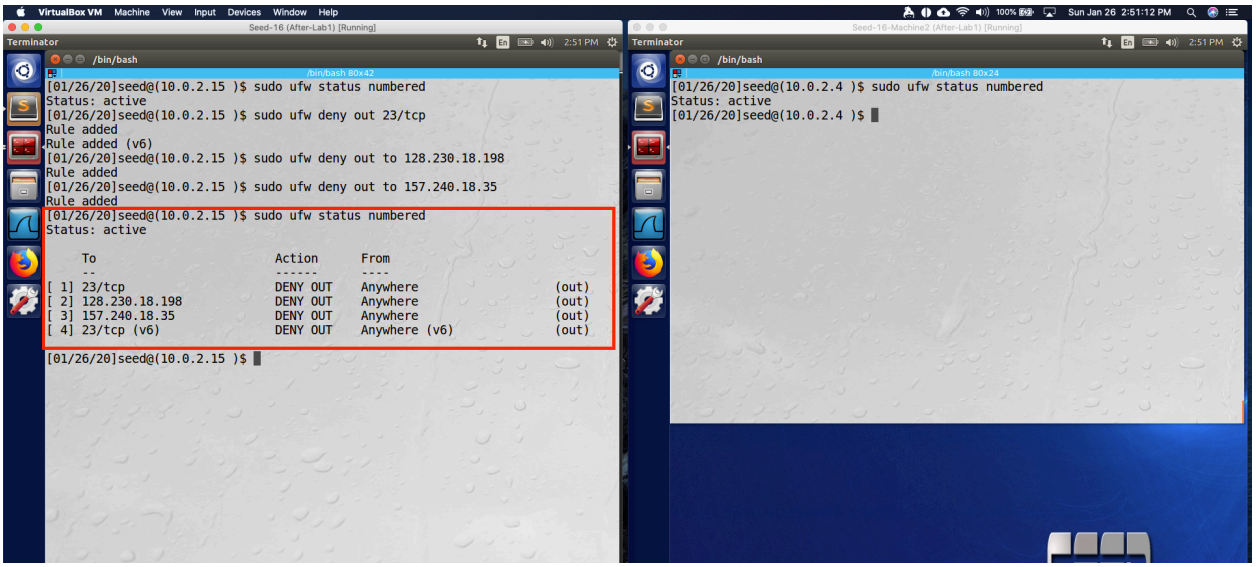
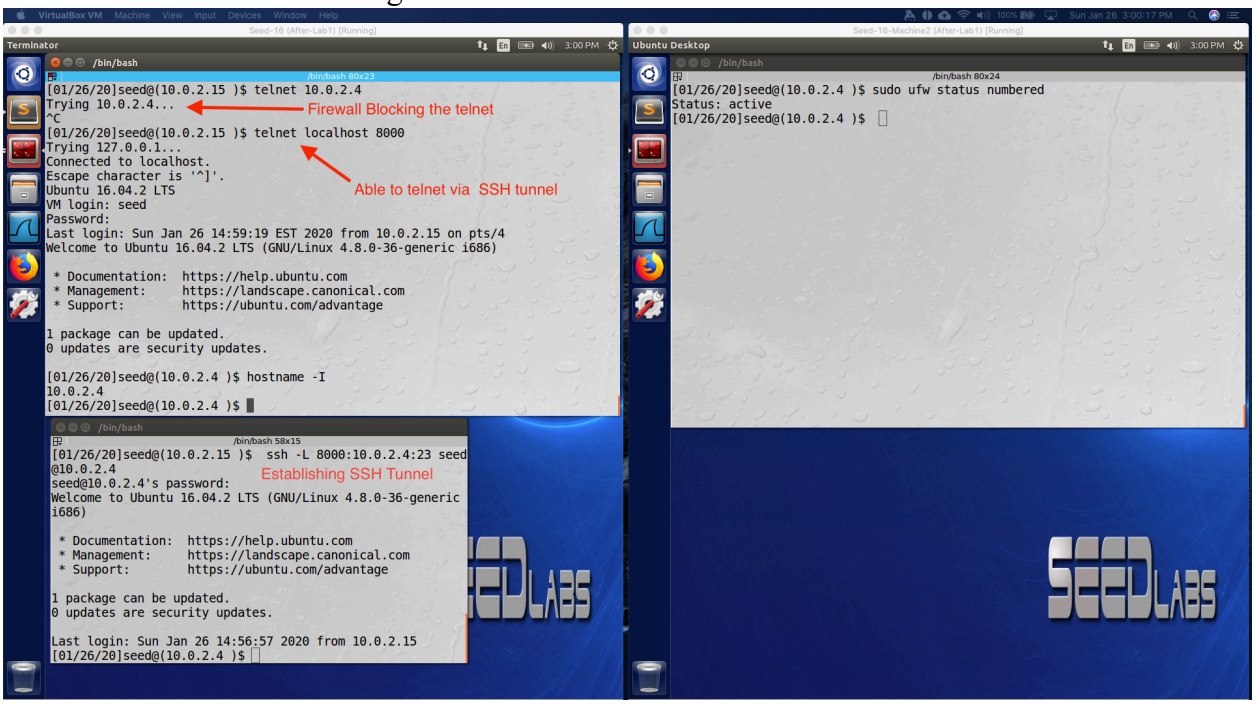


Fig: Adding the required firewall rules which block outgoing telnet, facebook and syr.edu

3a. Telnet to Machine B through the Firewall



Observation: Enabled the firewalls not to telnet the Machine B(10.0.2.4) when tried, the firewall blocked. But after enabling the SSH tunnel on port 8000, successfully able to telnet from machine A.

Explanation: Setup the firewalls using ufw to block the all outgoing telnet communications. Verified by trying the telnet and firewall blocked the connection. Then using the SSH tunnel which runs on port 22 with Machine B and using the local port 8000, then sent the telnet to

localhost 8000 which directed to Machine B and successfully able to telnet to Machine B. This was successful, because we are having the port 22 open on the firewall which can help in establishing the tunnels.

3b. Connect to Facebook using SSH Tunnel

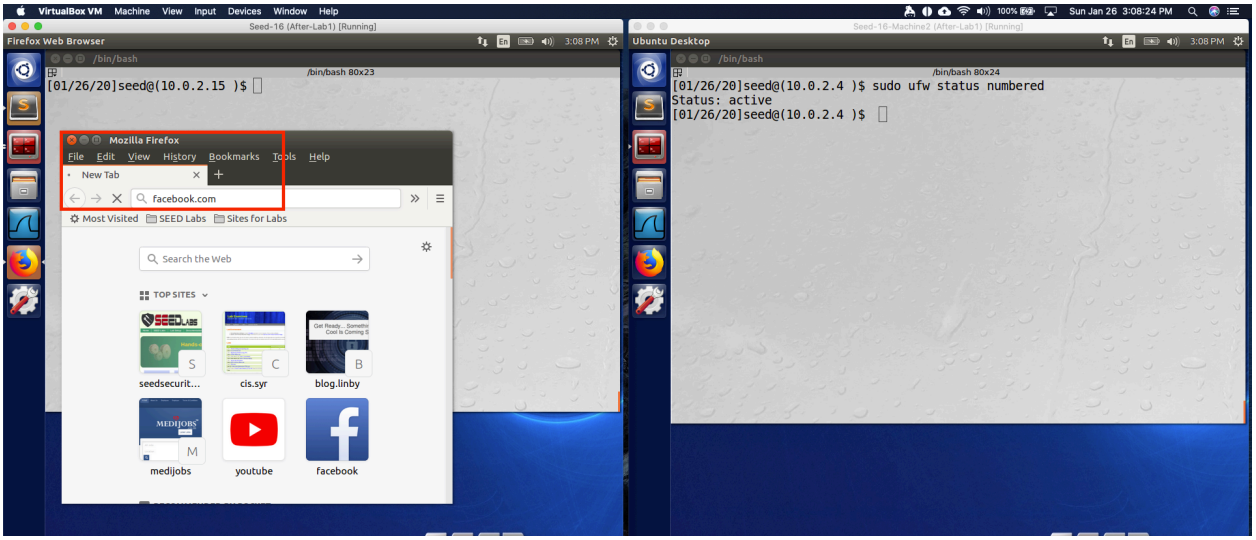


Fig: Unable to access the Facebook

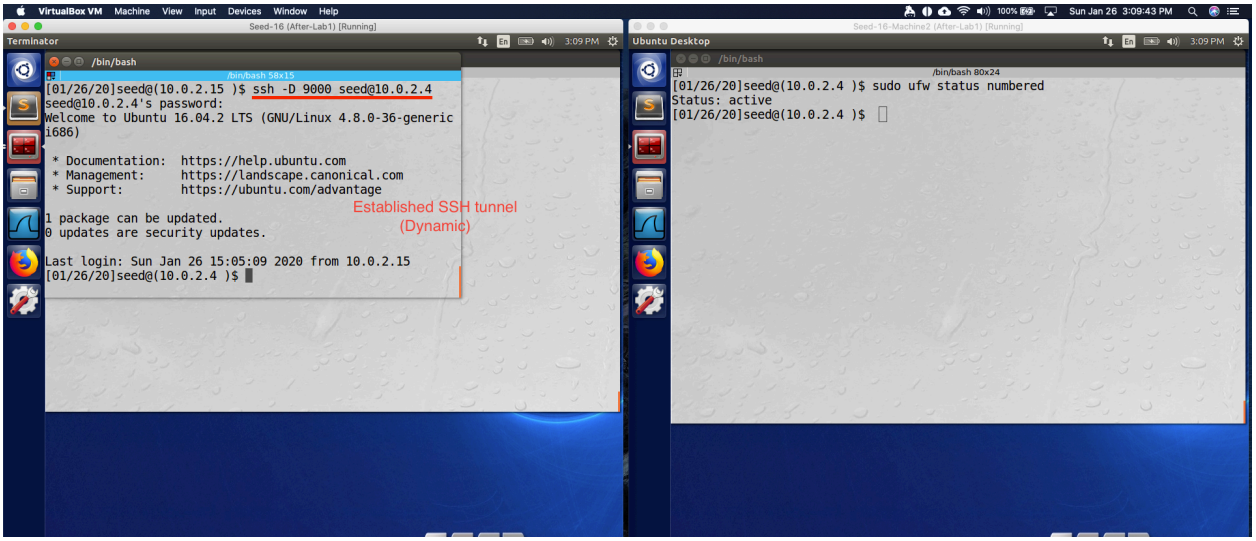


Fig: Establishing the SSH tunnel (Dynamic)

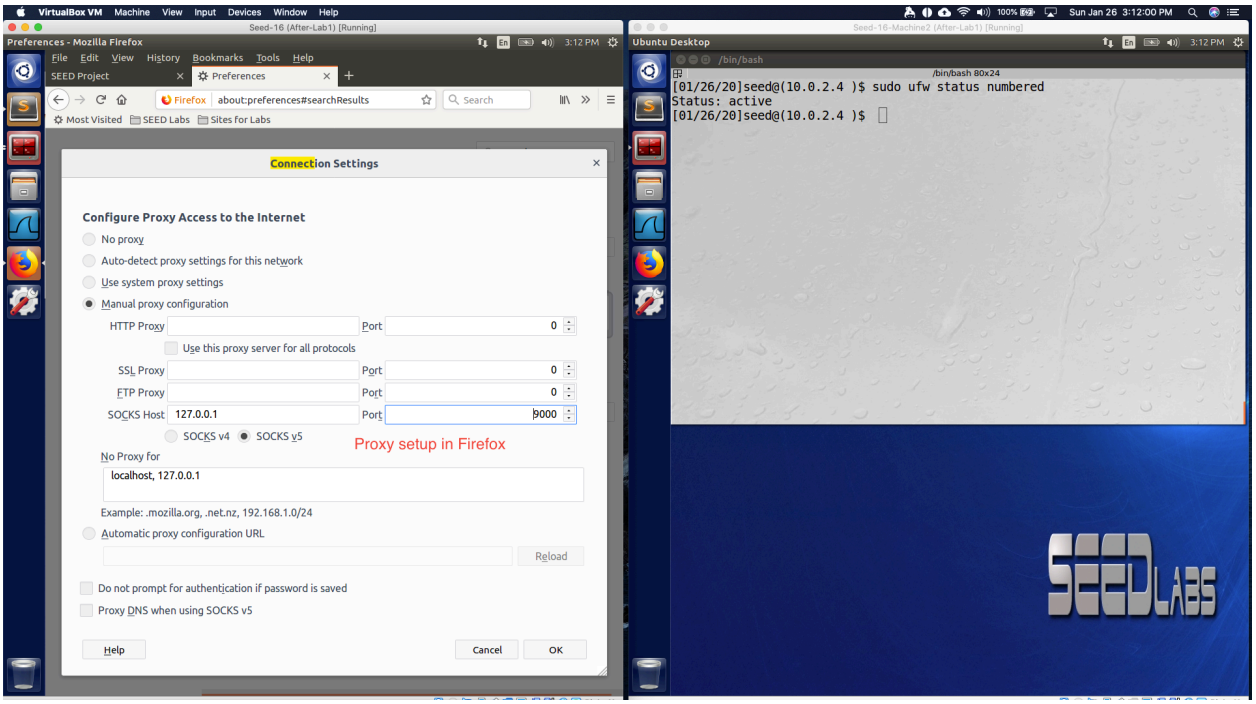


Fig: Proxy setup in Firefox to use SSH tunnel

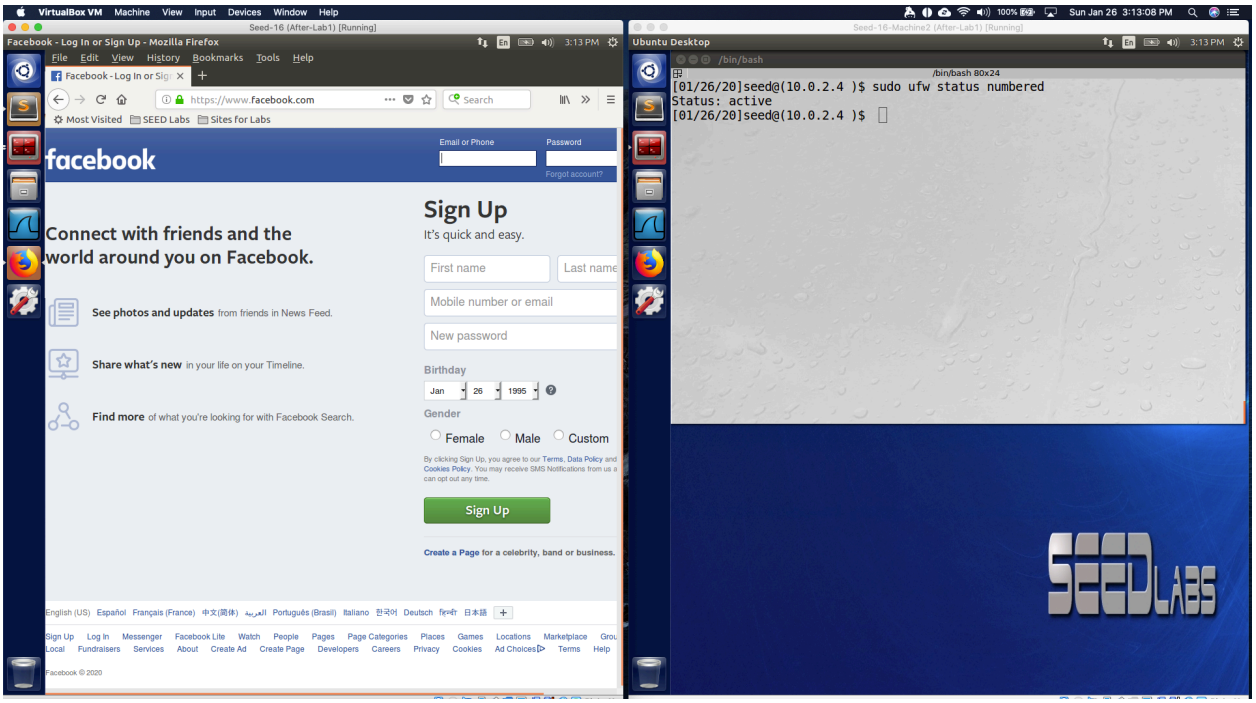


Fig: Able to access the Facebook

Observation: Added the rule to firewall to block the Facebook and then using the SSH dynamic tunneling established the proxy on localhost 9000 to Machine B which has access to Facebook. After updating the proxy setting in Firefox to use the proxy, successfully connected to Facebook.

Explanation: Blocked the firewall using ufw utility and verified that Facebook is blocked from machine A. But Machine B has the access to Facebook and Machine A can establish SSH to Machine B. So, established an SSH tunnel from Machine A to Machine B on port 9000. Once established all the connections on 9000 will be sent to Machine B over SSH and get the responses back to the same. So, using this, updated the Firefox proxy settings to use the localhost port 9000 and sent the web traffic through it. Thus, the access to Facebook was successful.