COMPUTER NETWORKS SECURITY LABORATORY

LAB 3

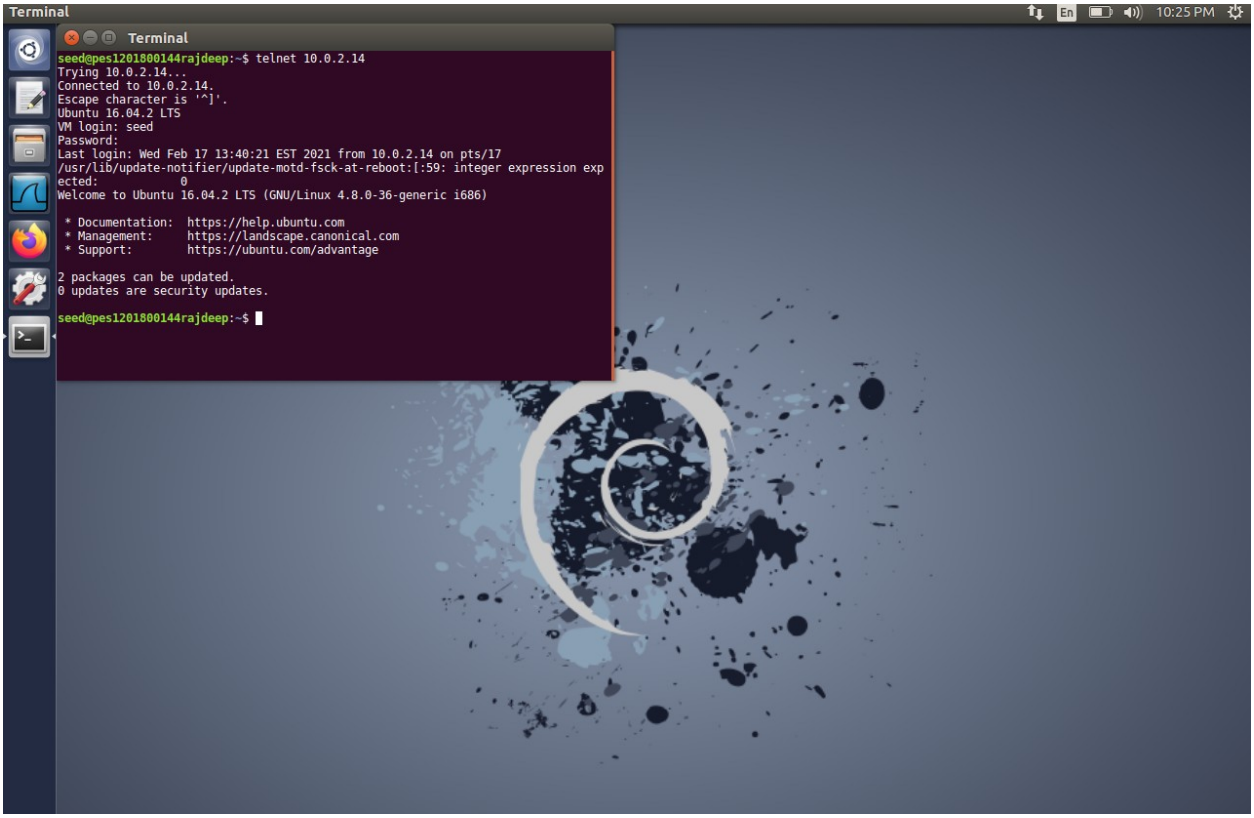
BY: RAJDEEP SENGUPTA

SRN: PES1201800144

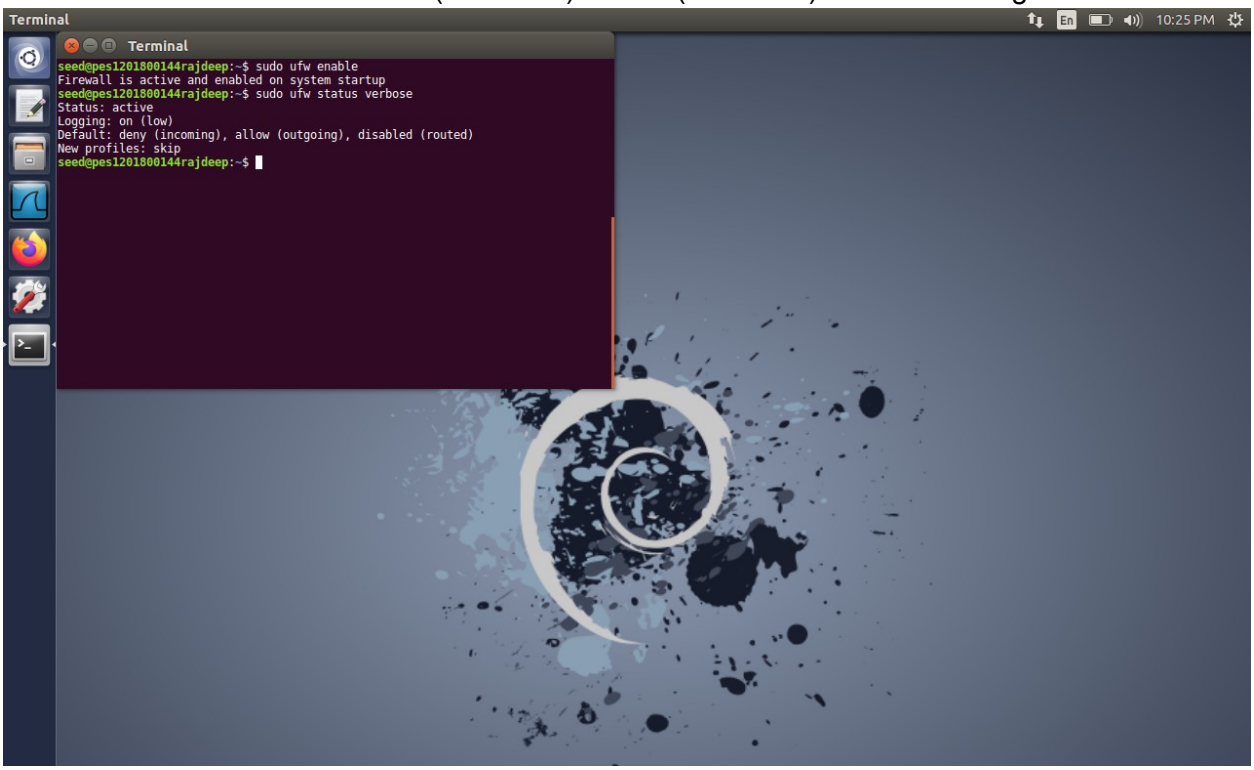
SECTION: C

NOTE: Please find my SRN ‘PES1201800144rajdeep’ as the terminal username.

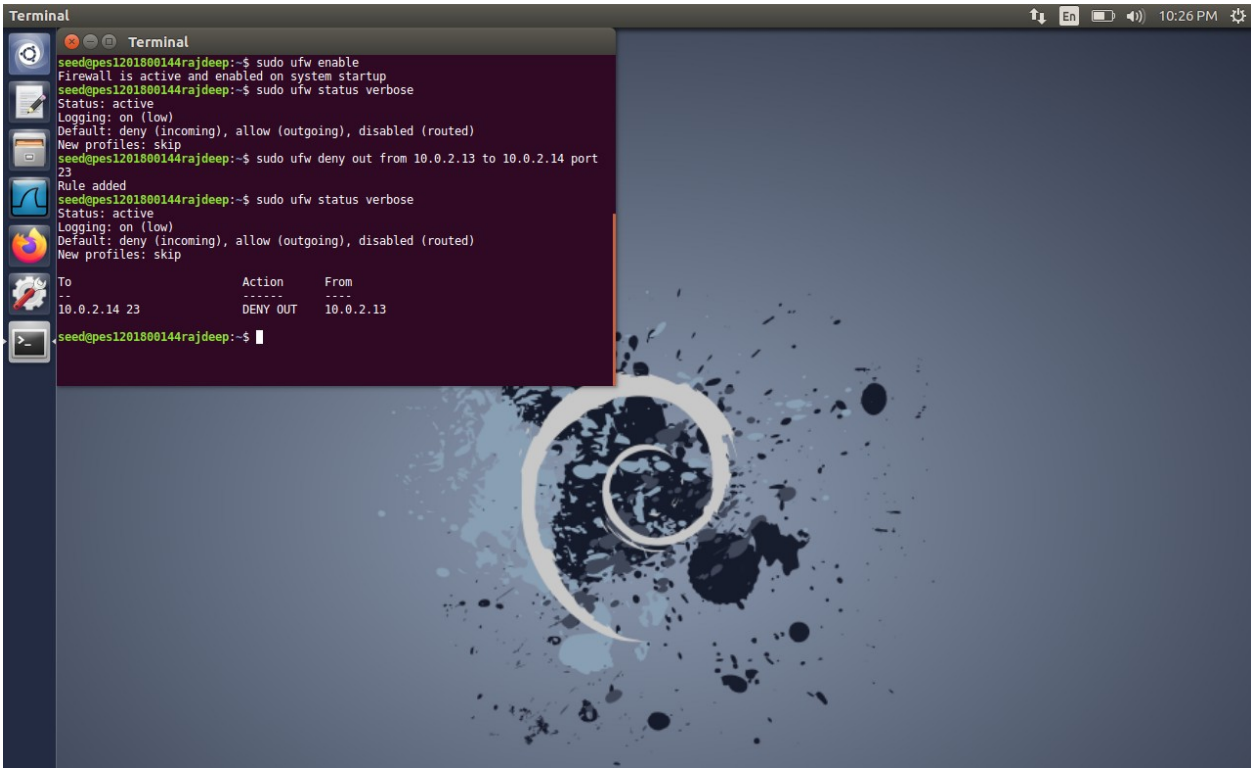
TASK 1:



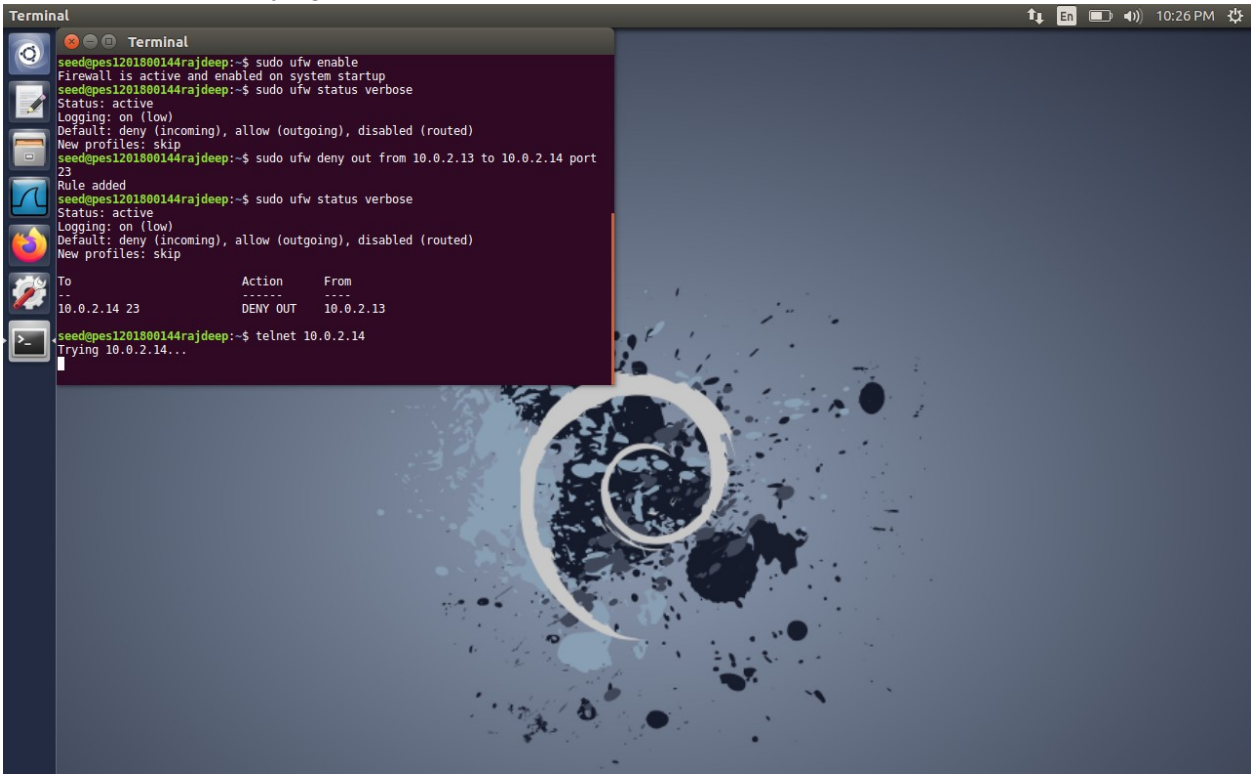
Screenshot 1.1: Telnet from VM1(10.0.2.13) to VM2(10.0.2.14) before enabling ufw



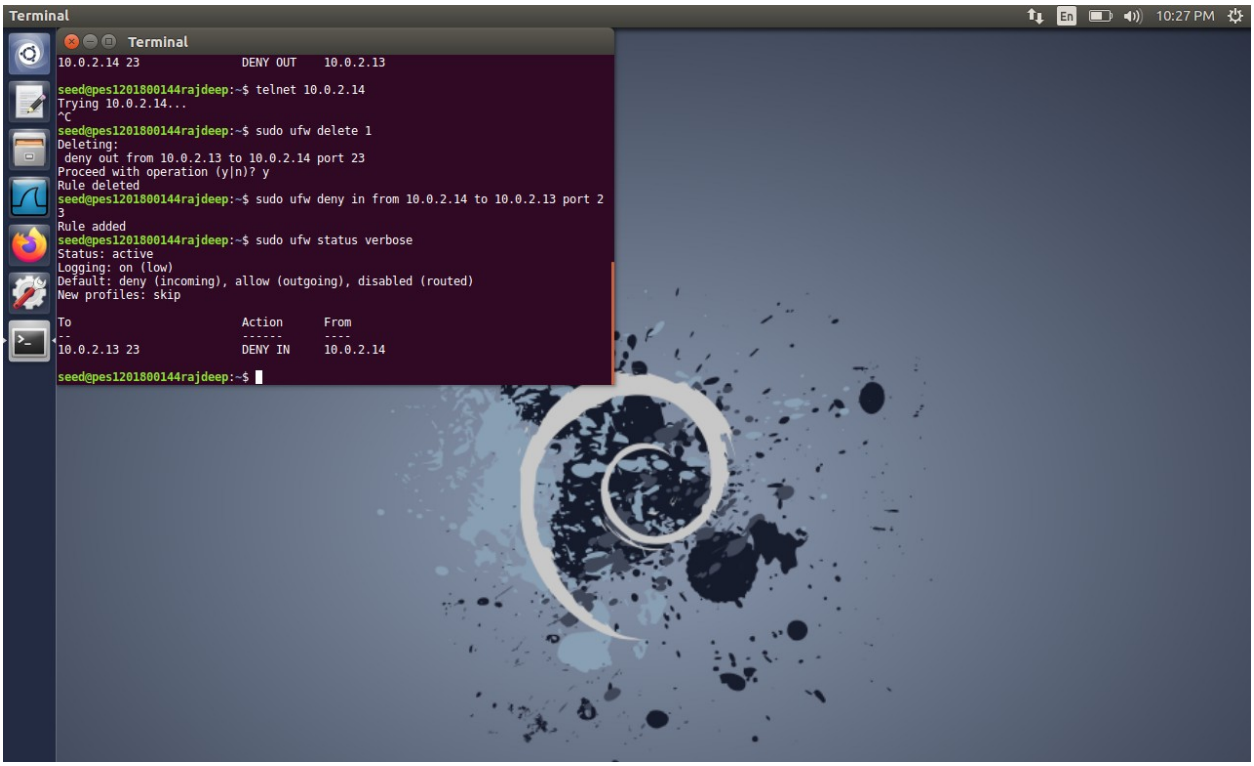
Screenshot 1.2: Enabling firewall and displaying status



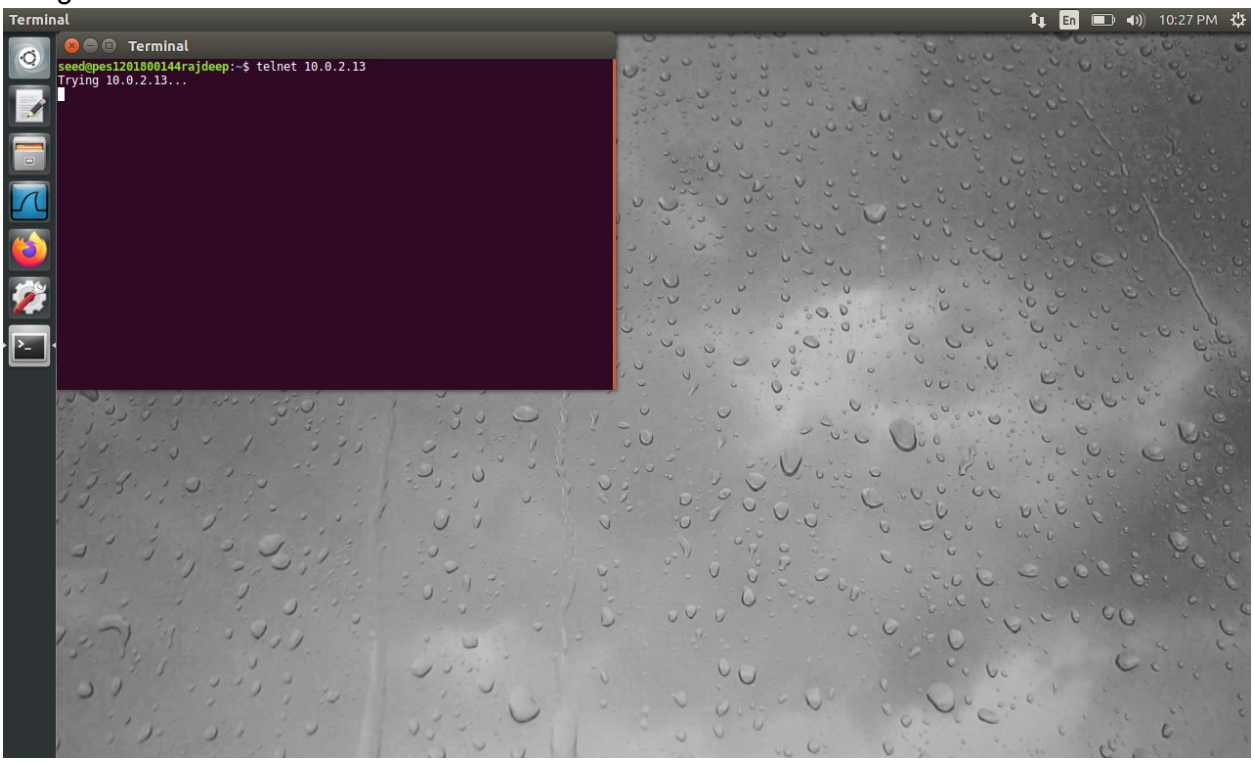
Screenshot 1.3: Denying telnet connection on port 23 from VM1(10.0.2.13) to VM2(10.0.2.14)



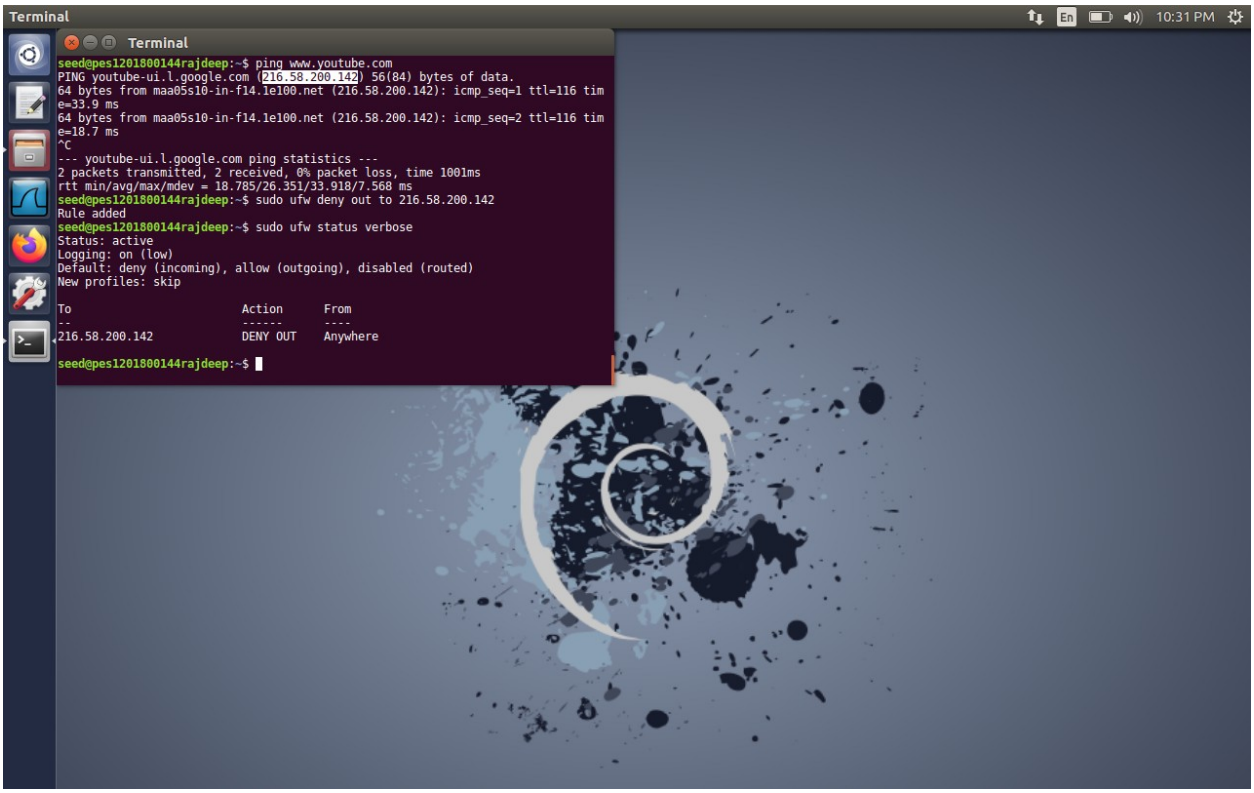
Screenshot 1.4: Trying telnet connection which is unsuccessful since it is denied by firewall



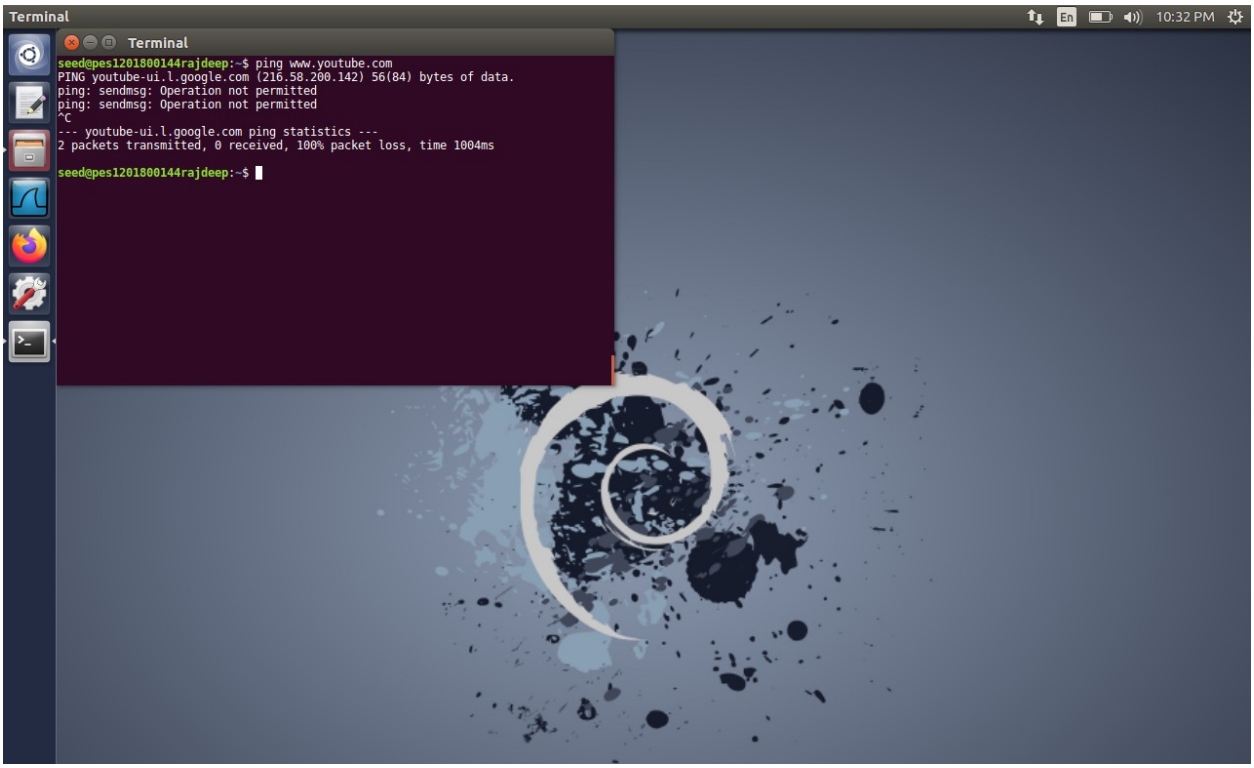
Screenshot 1.5: Denying telnet connection on port 23 from VM2(10.0.2.14) to VM1(10.0.2.13) using firewall



Screenshot 1.6: Trying telnet from VM2 to VM1 but unsuccessful since the connection is denied by firewall of VM1

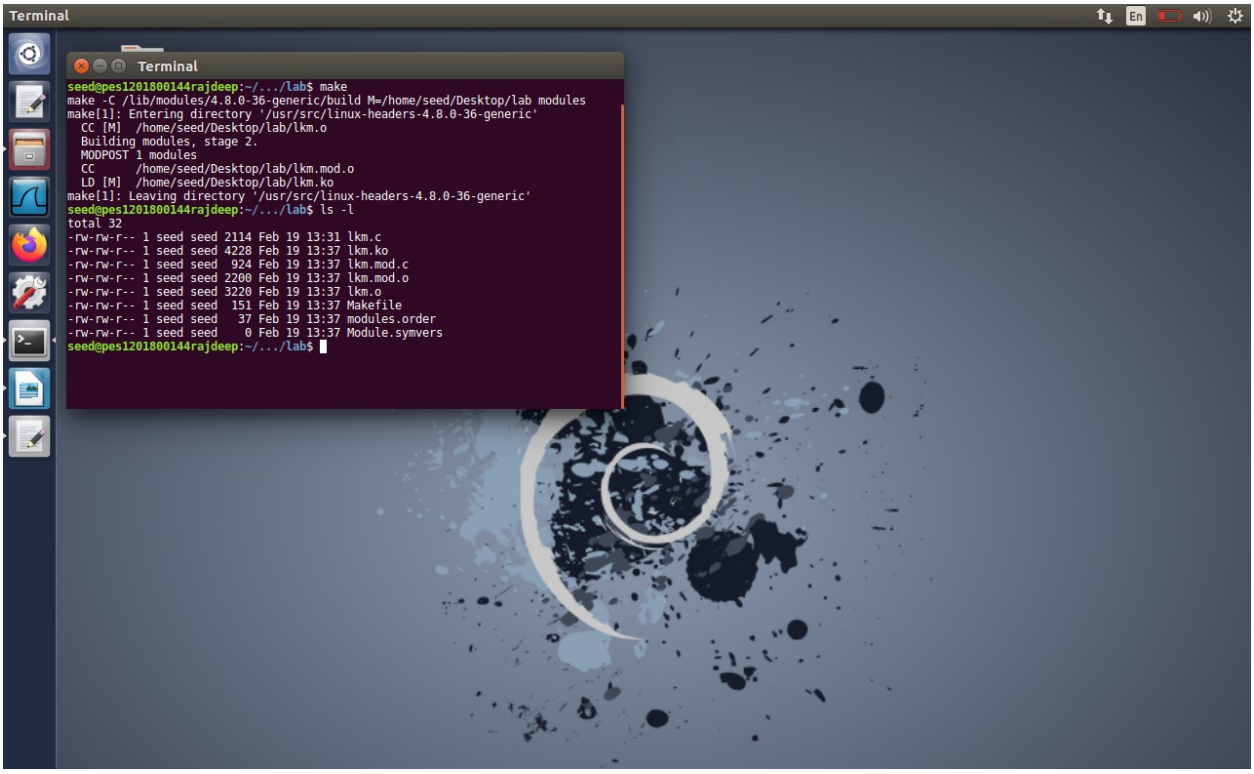


Screenshot 1.7: Pinging www.youtube.com to find it's IP address and then denying outgoing connections to that IP address

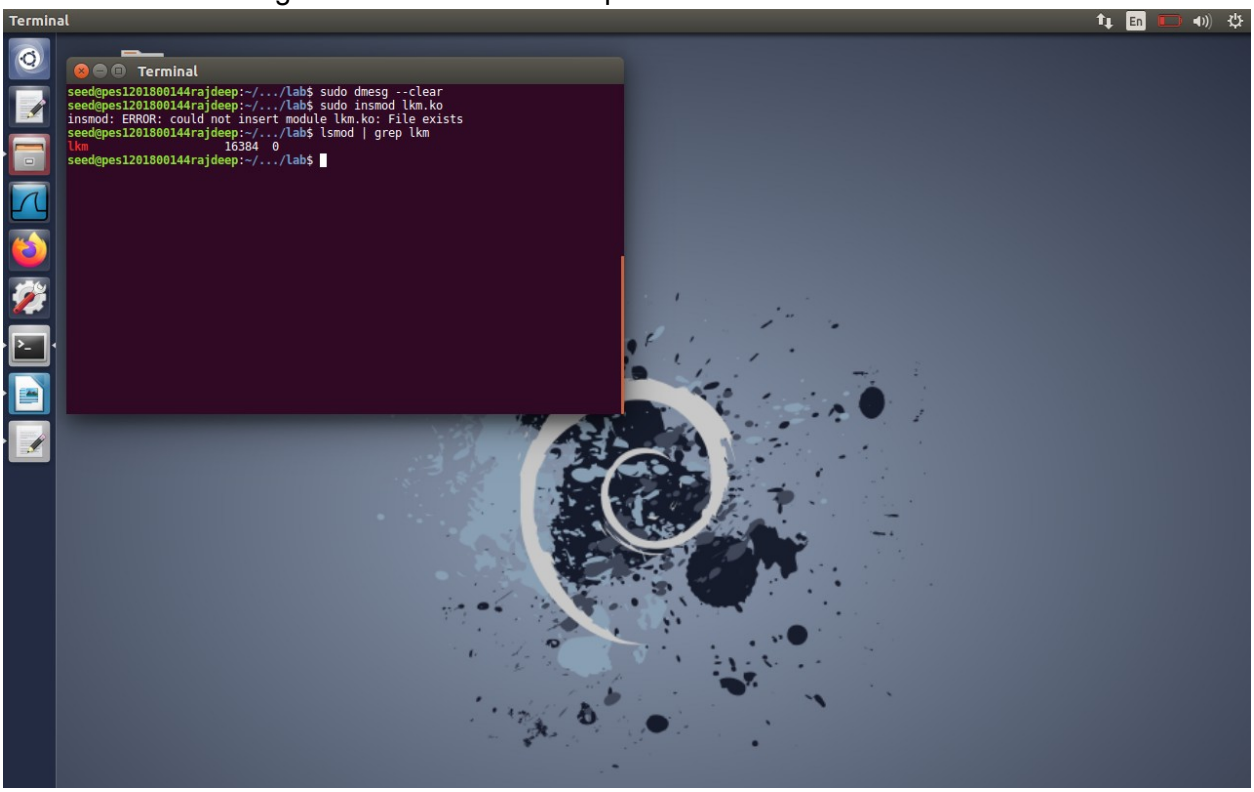


Screenshot 1.8: Pinging to www.youtube.com but it shows "Operation not permitted" since the connection is denied by firewall of VM1

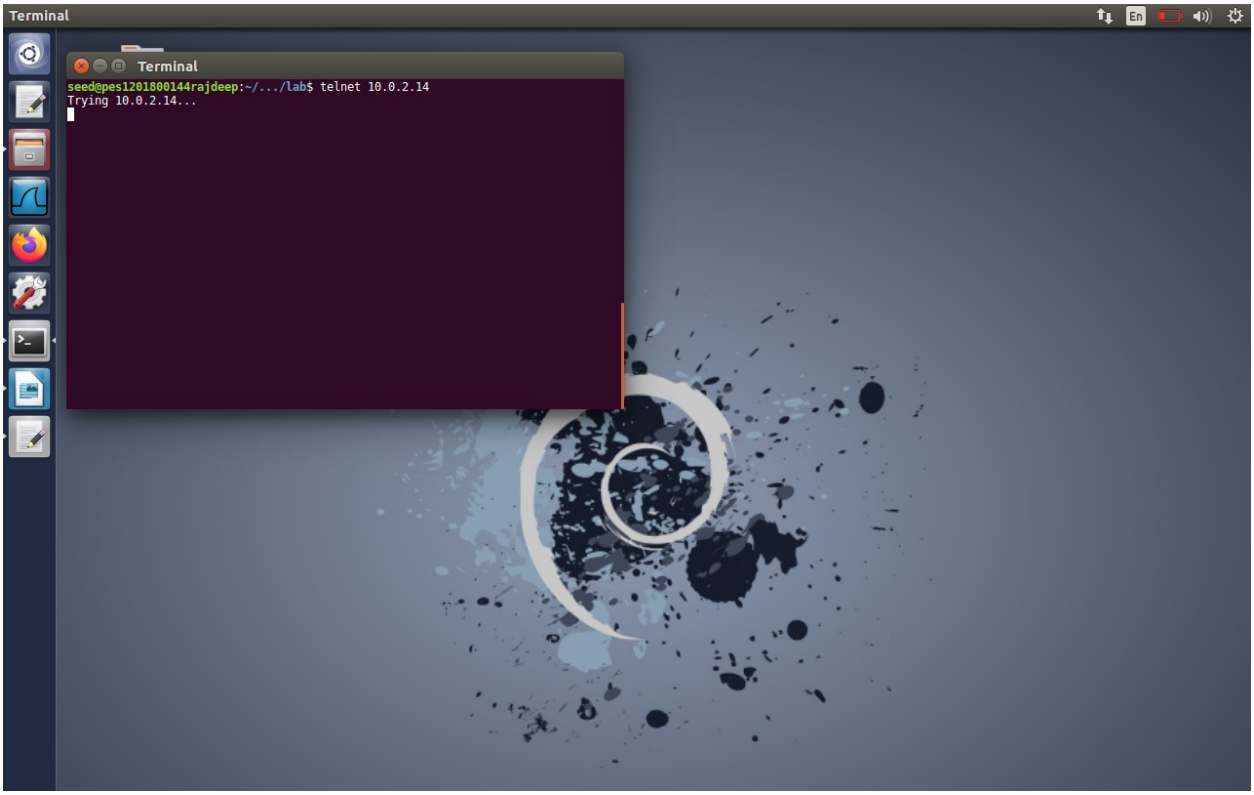
TASK 2:



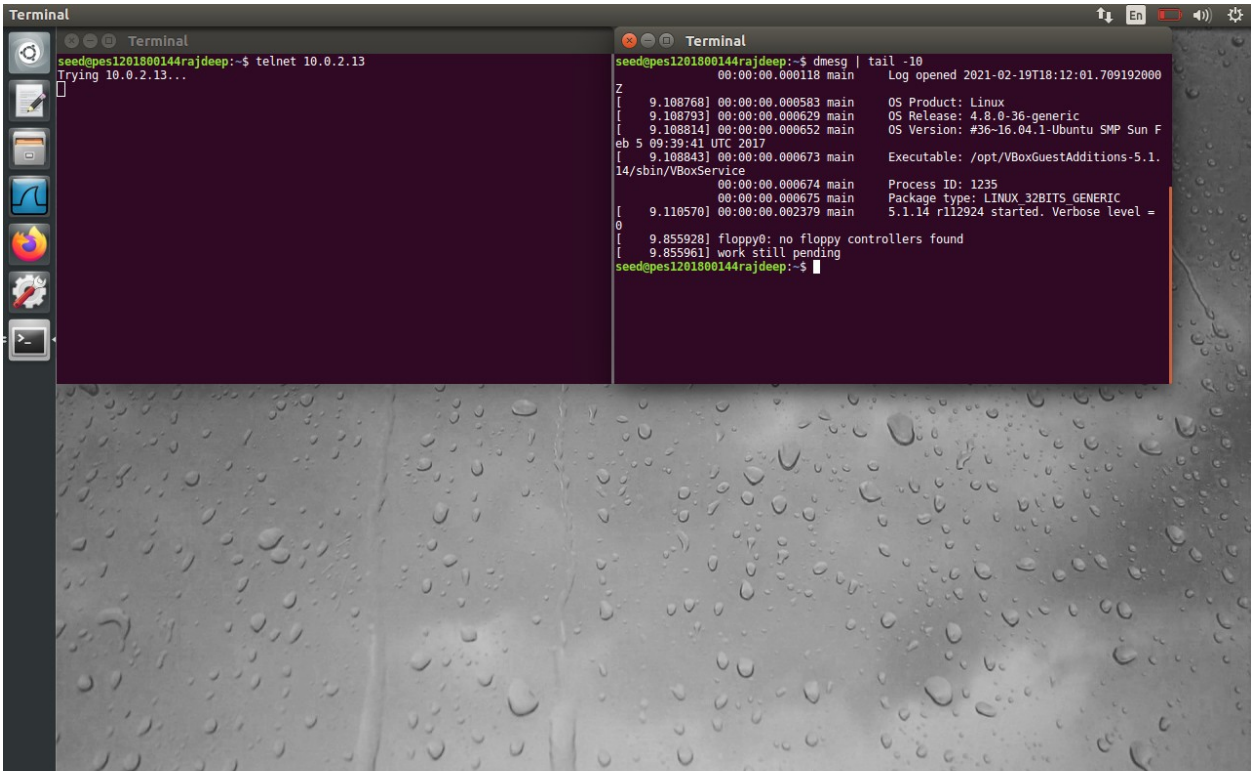
Screenshot 2.1: Using make command to compile the kernel module



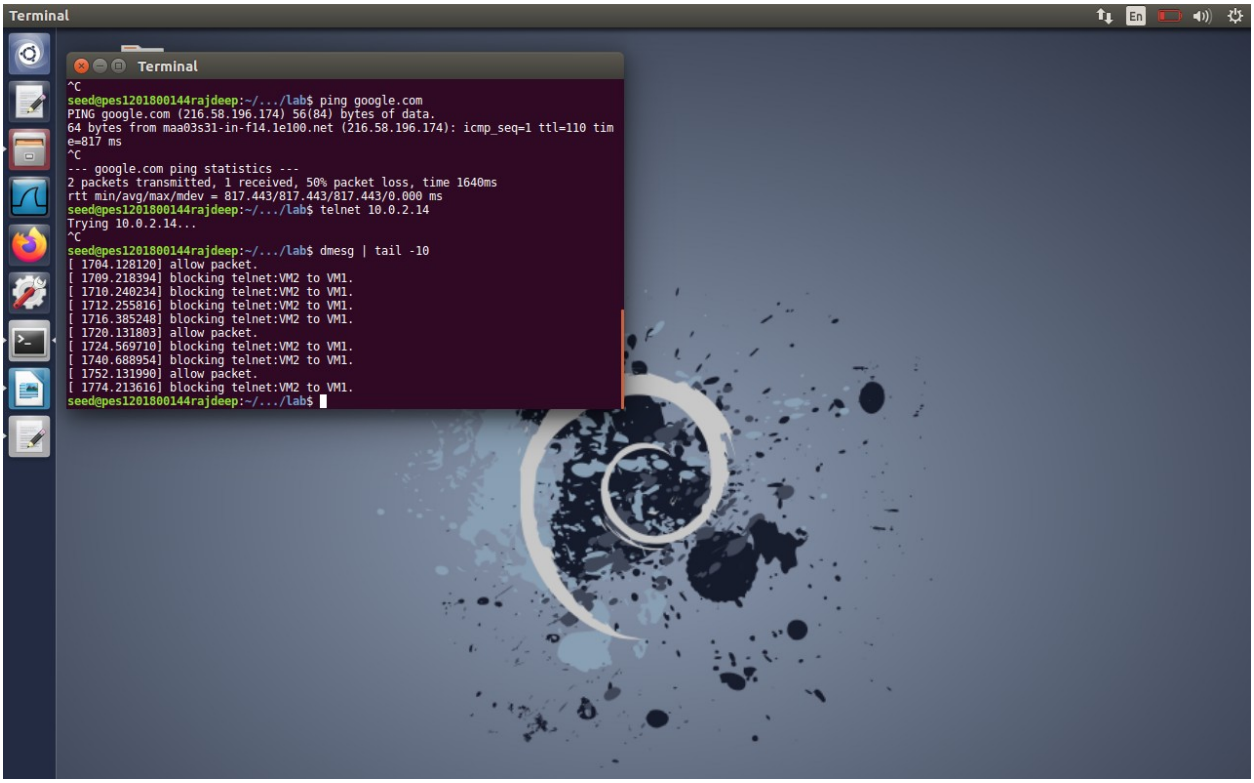
Screenshot 2.2: Clearing the dmesg buffer



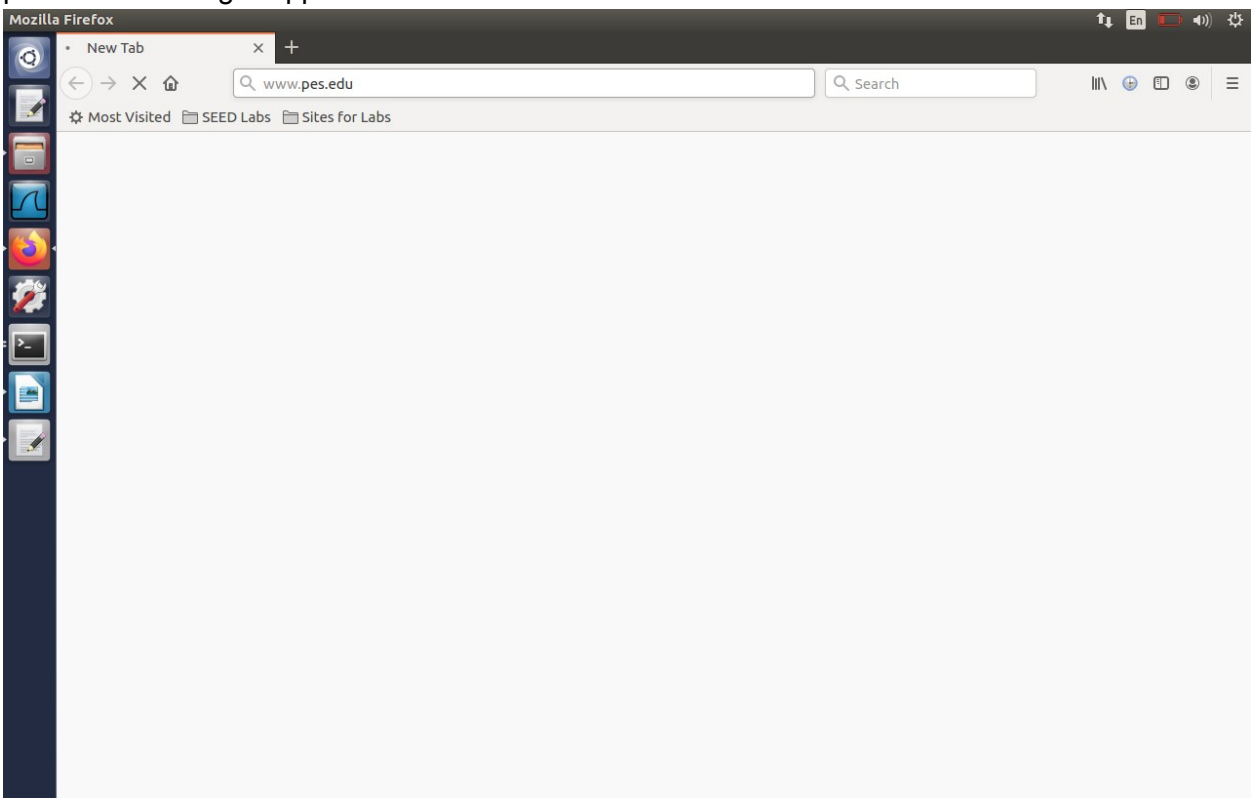
Screenshot 2.3: Trying telnet connection from VM1(10.0.2.13) to VM2(10.0.2.14) but it is blocked



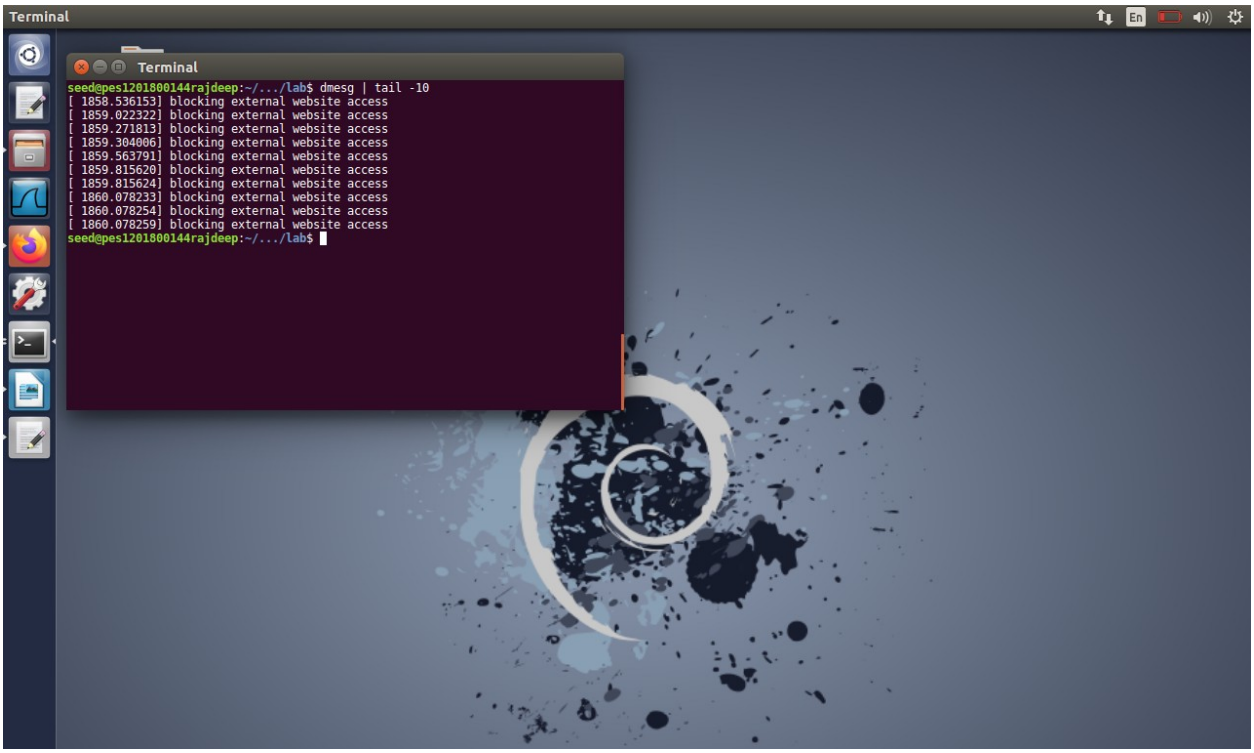
Screenshot 2.4: Trying telnet from VM2(10.0.2.14) to VM1(10.0.2.13) but it is blocked



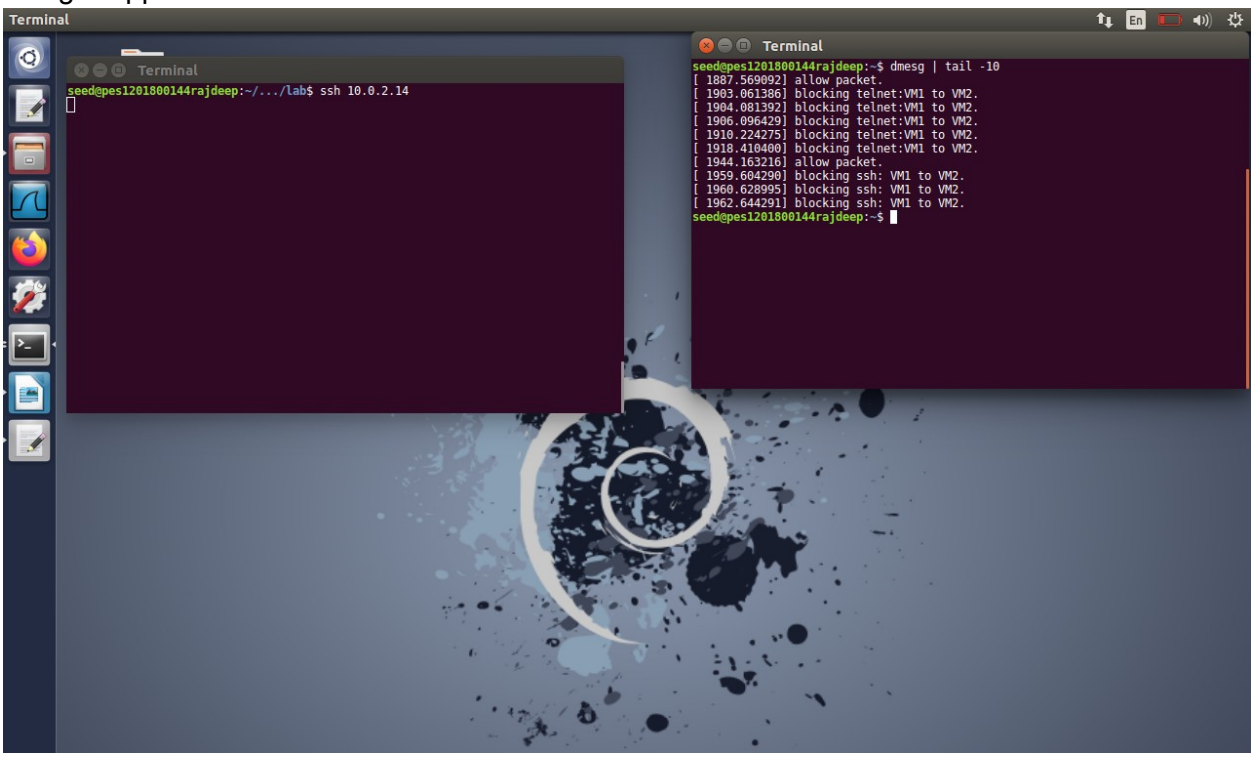
Screenshot 2.6: It can be seen that the telnet connections from VM2 to VM1 are blocked as the packets are being dropped



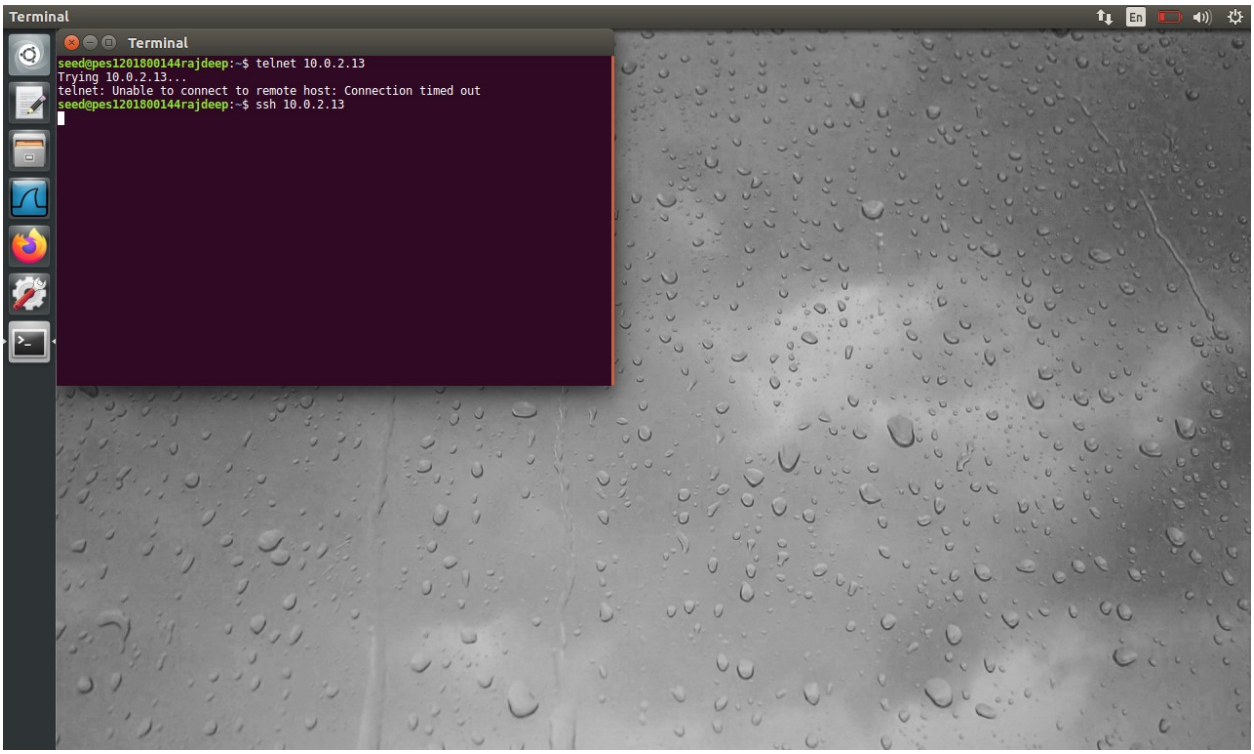
Screenshot 2.7: Trying to open a website(www.pes.edu) but it is again blocked



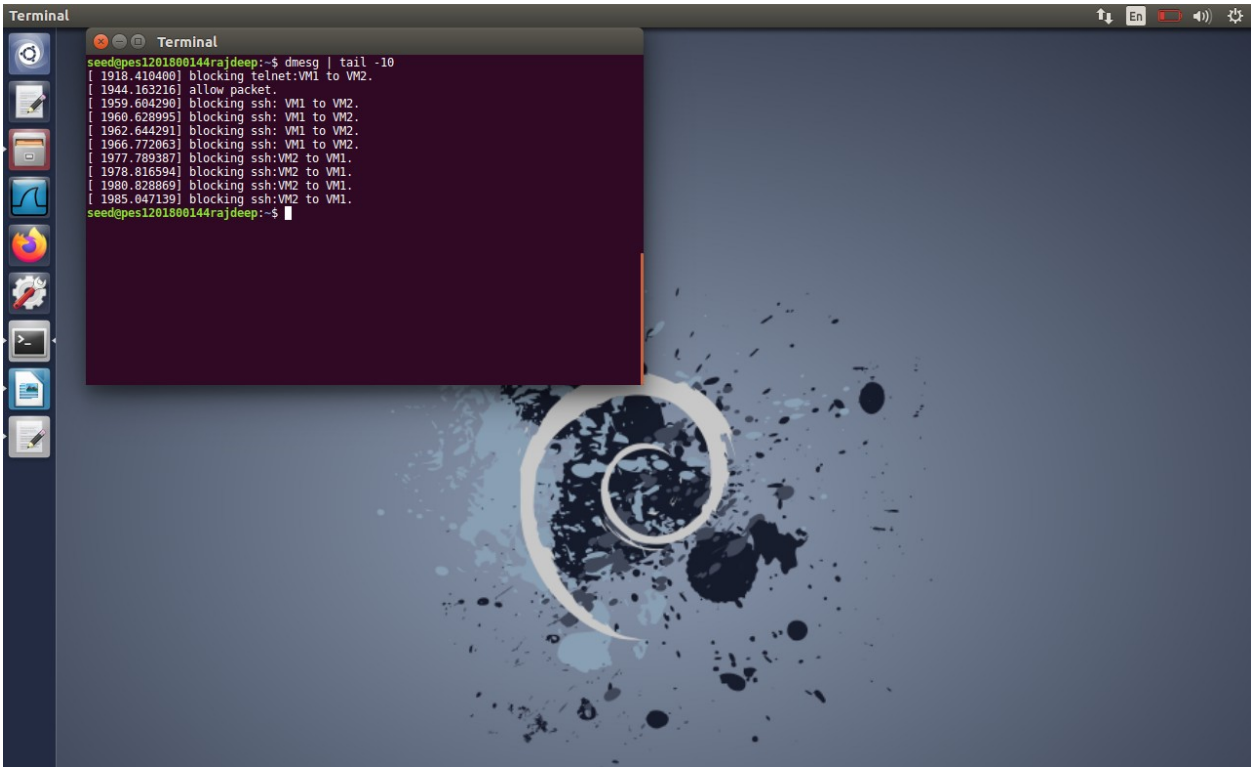
Screenshot 2.8: It can be seen that the external website access is blocked since the packets are being dropped



Screenshot 2.9: It can be seen that VM1(10.0.2.13) is trying to connect through SSH to VM2(10.0.2.14) but the packets are being dropped



Screenshot 2.10: VM2 tries SSH connection to VM1 but it is blocked

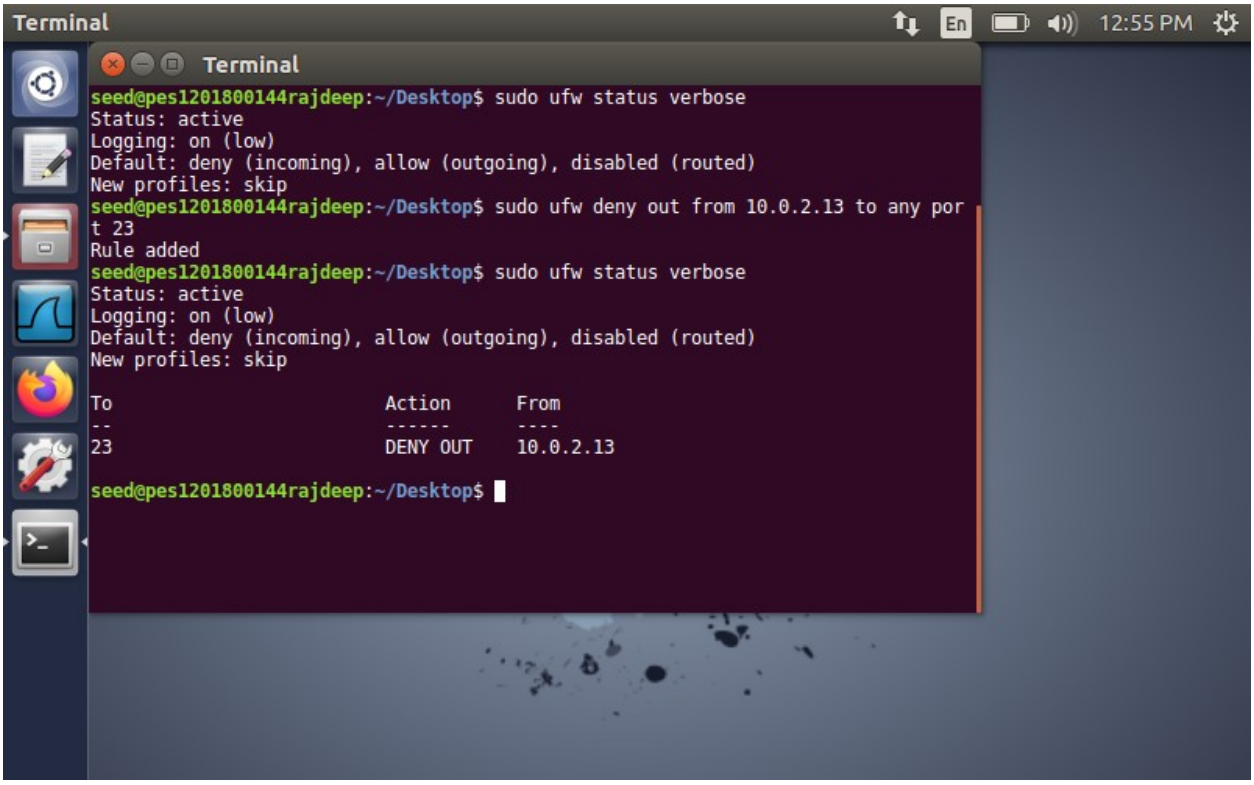


Screenshot 2.11: Above SSH connection from VM2 to VM1 is blocked and the packets are dropped

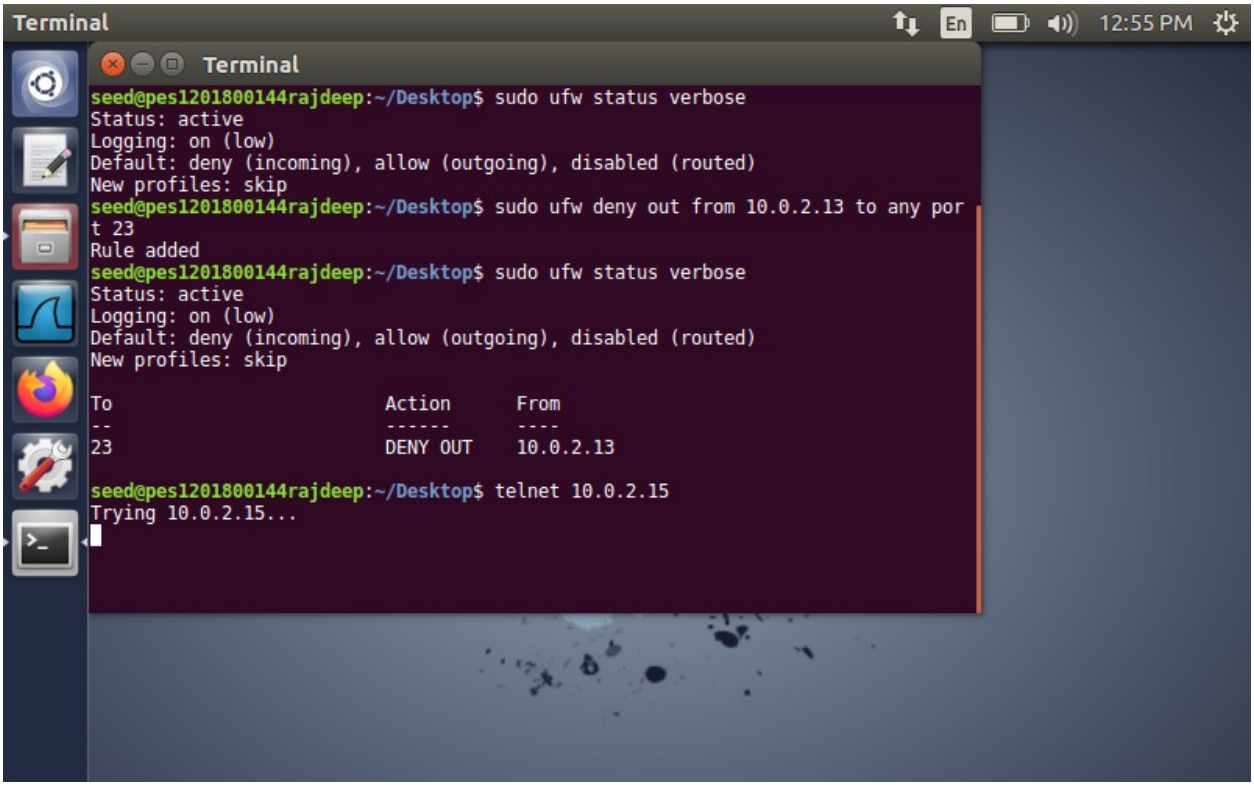
OBSERVATIONS:

A simple firewall is implemented in this task using netfilter. This is done by loading the code into the kernel using the hooks in netfilter. The packets are being dropped according to the rules written in the lkm file. This is proved by the evident kernel logs shown in the above screenshots showing the packets being dropped.

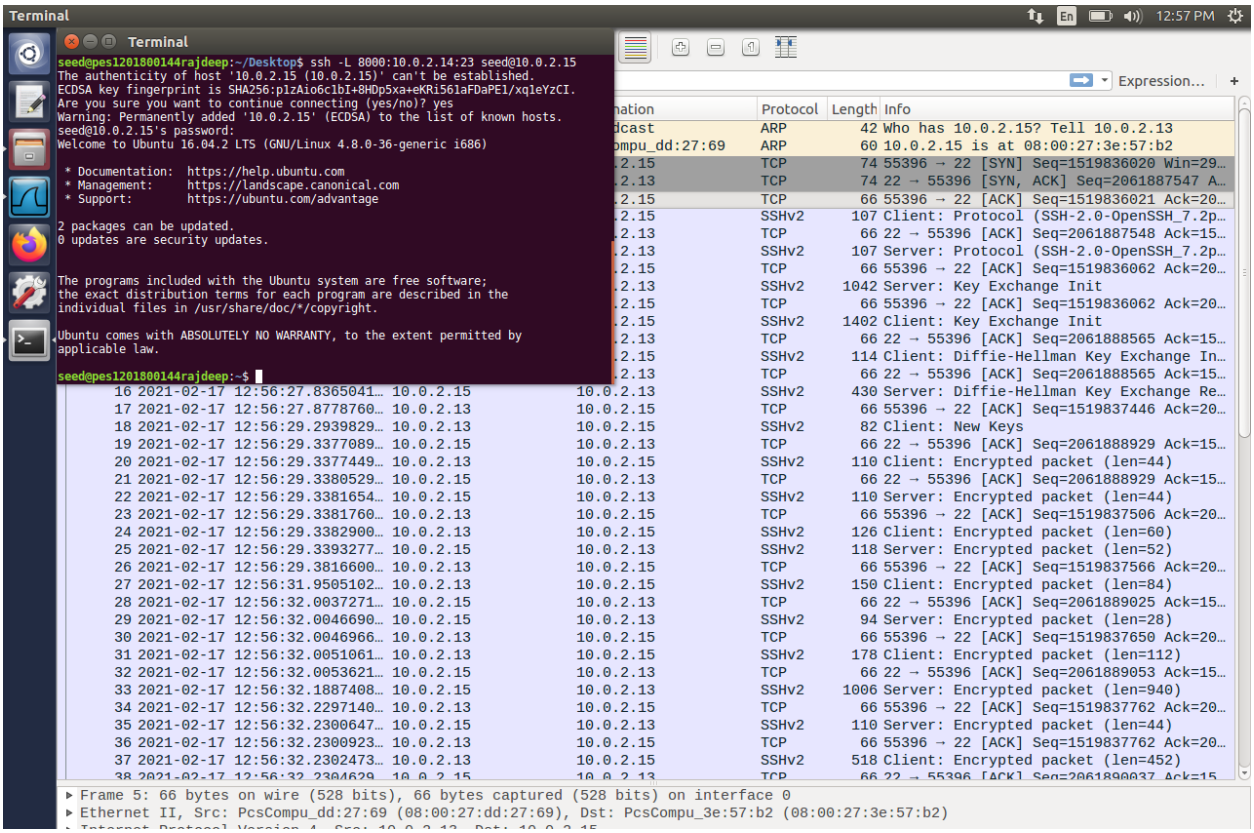
TASK 3a:



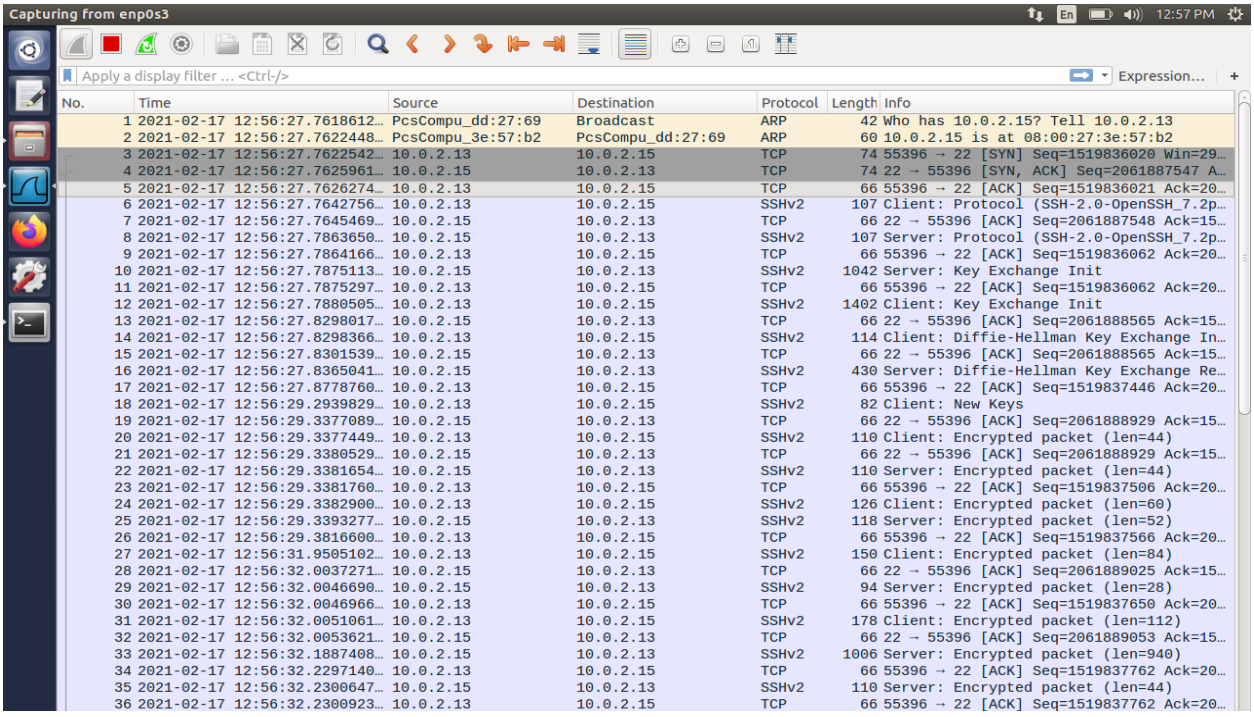
Screenshot 3.a.1: On VM1(10.0.2.13), denying outgoing telnet connections on port 23.



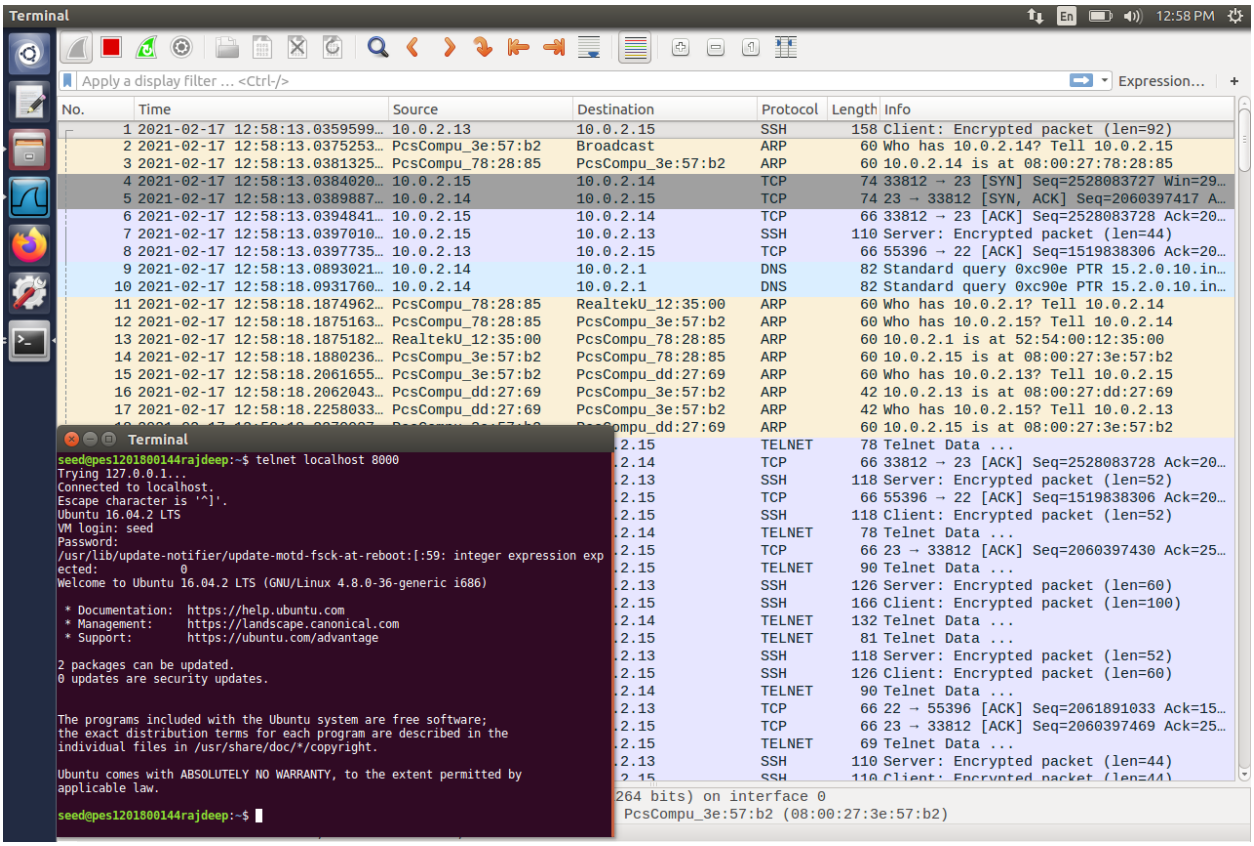
Screenshot 3.a.2: Trying telnet from VM1 to VM3 but unsuccessful due to firewall



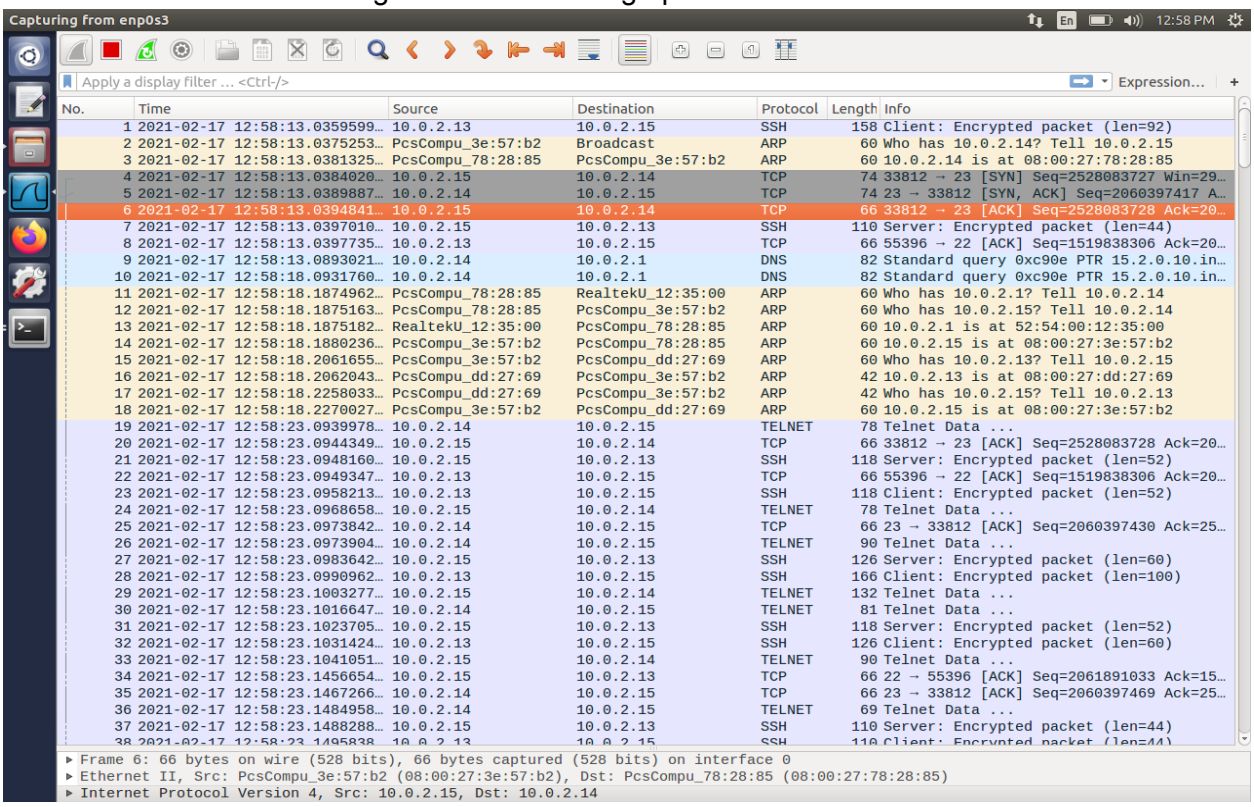
Screenshot 3.a.3: Setting up ssh tunnel from between VM1(10.0.2.13) and VM2(10.0.2.14) to get connection through telnet to VM3(10.0.2.15) via VM2. Also, it is specified that the telnet connection will be through port 8000



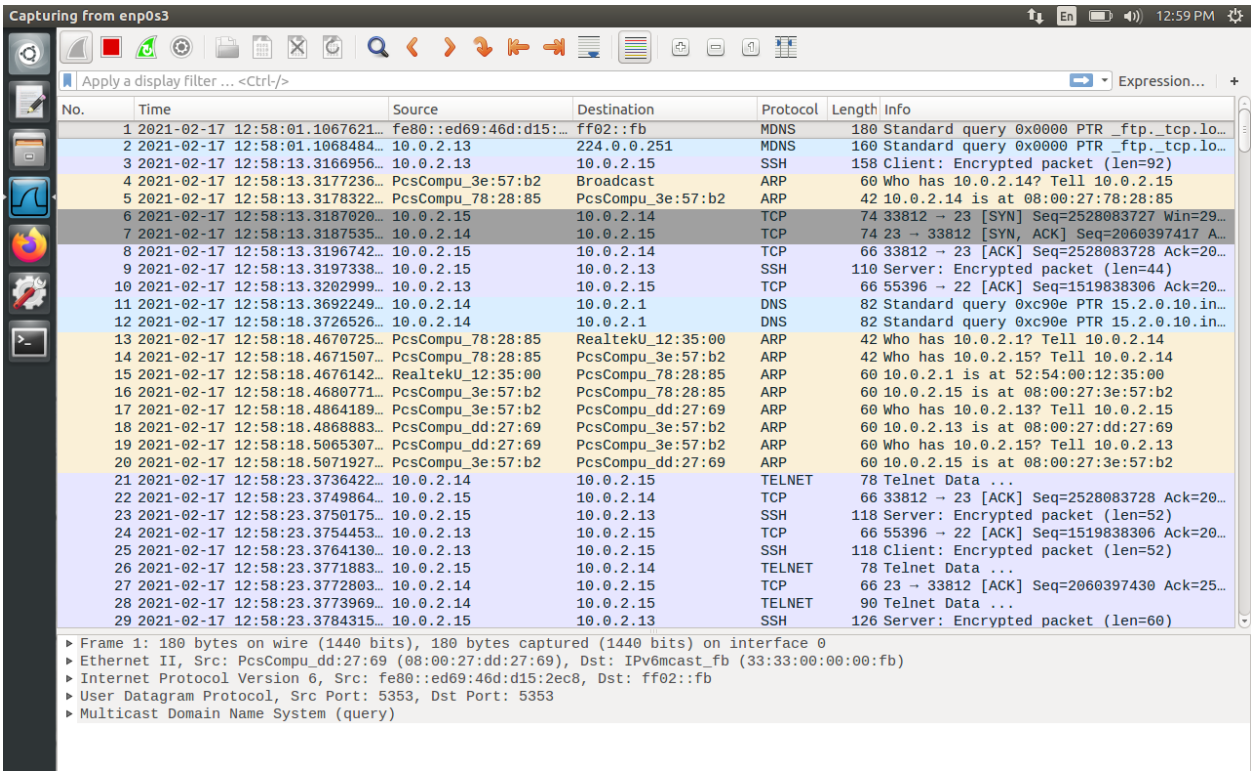
Screenshot 3.a.4: SSH connection successful from VM1(10.0.13) to VM3(10.0.2.15)



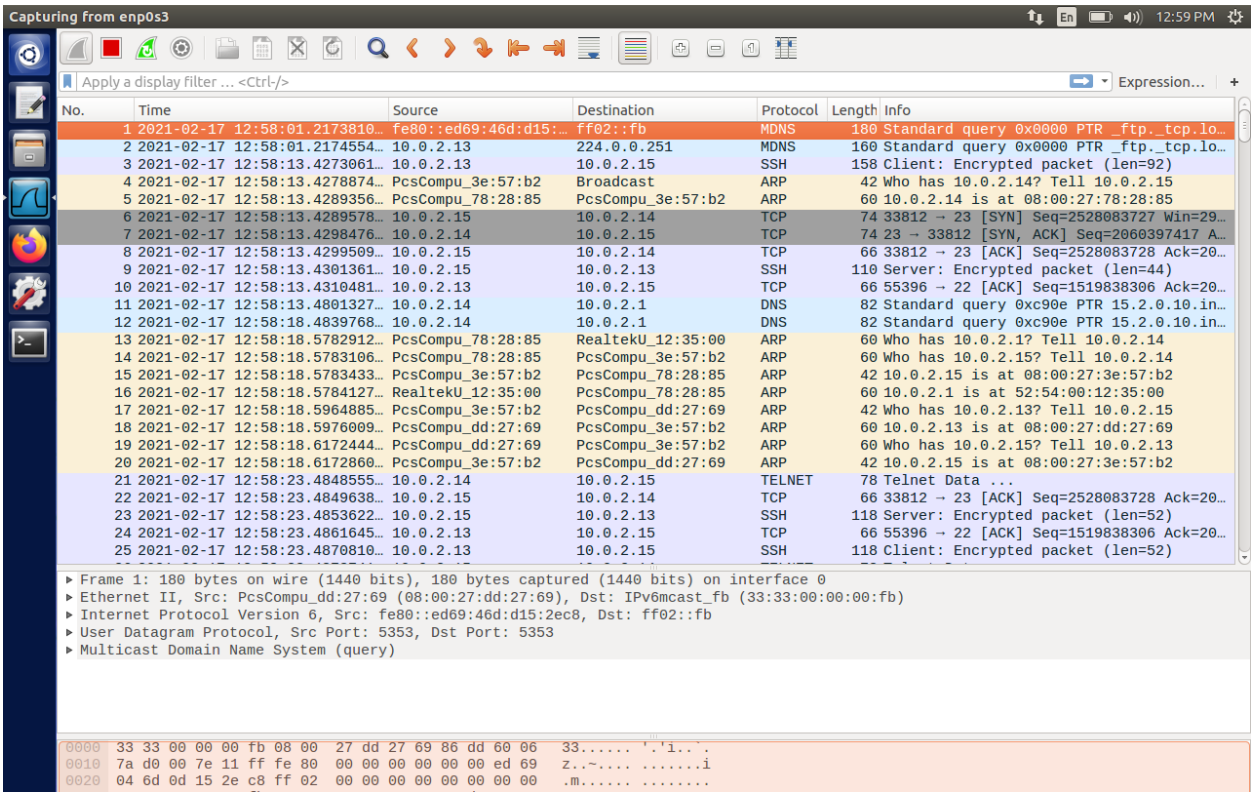
Screenshot 3.a.5: Connecting to localhost through port 8000



Screenshot 3.a.6: VM1(10.0.2.13) wireshark packet capture



Screenshot 3.a.7: VM2(10.0.2.14) wireshark packet capture

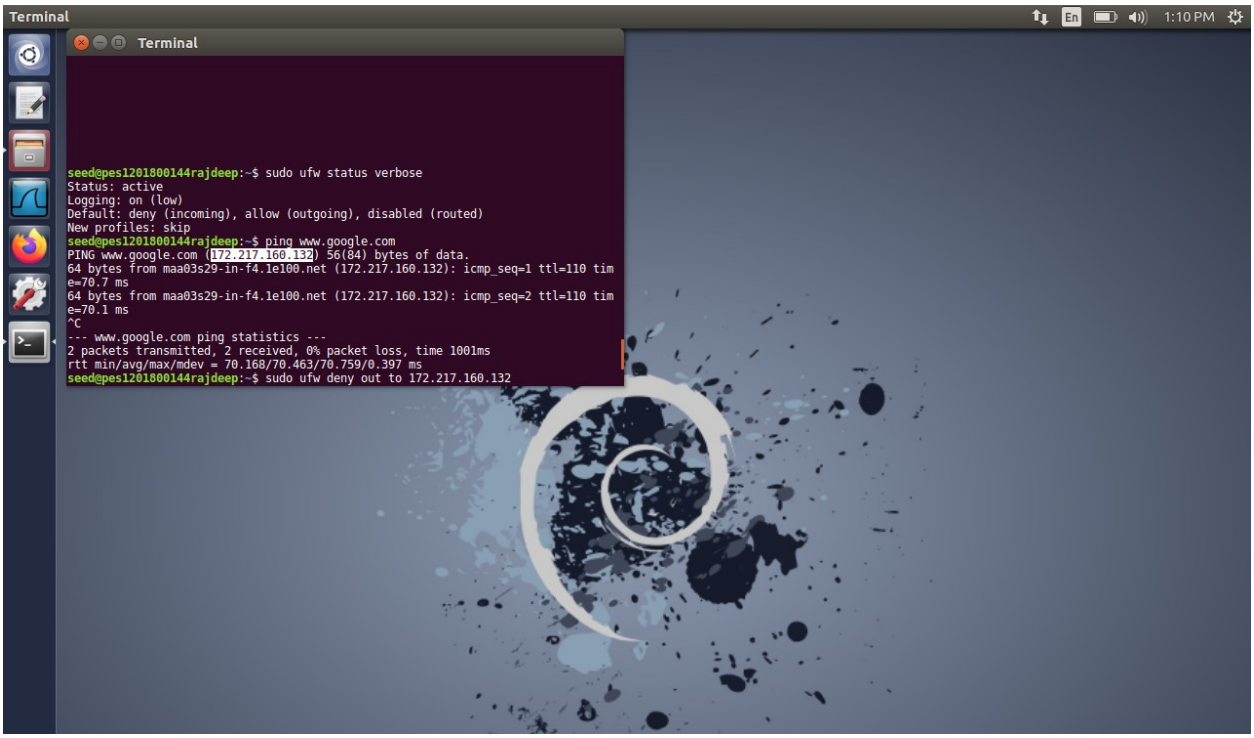


Screenshot 3.a.8: VM3(10.0.2.15) wireshark packet capture

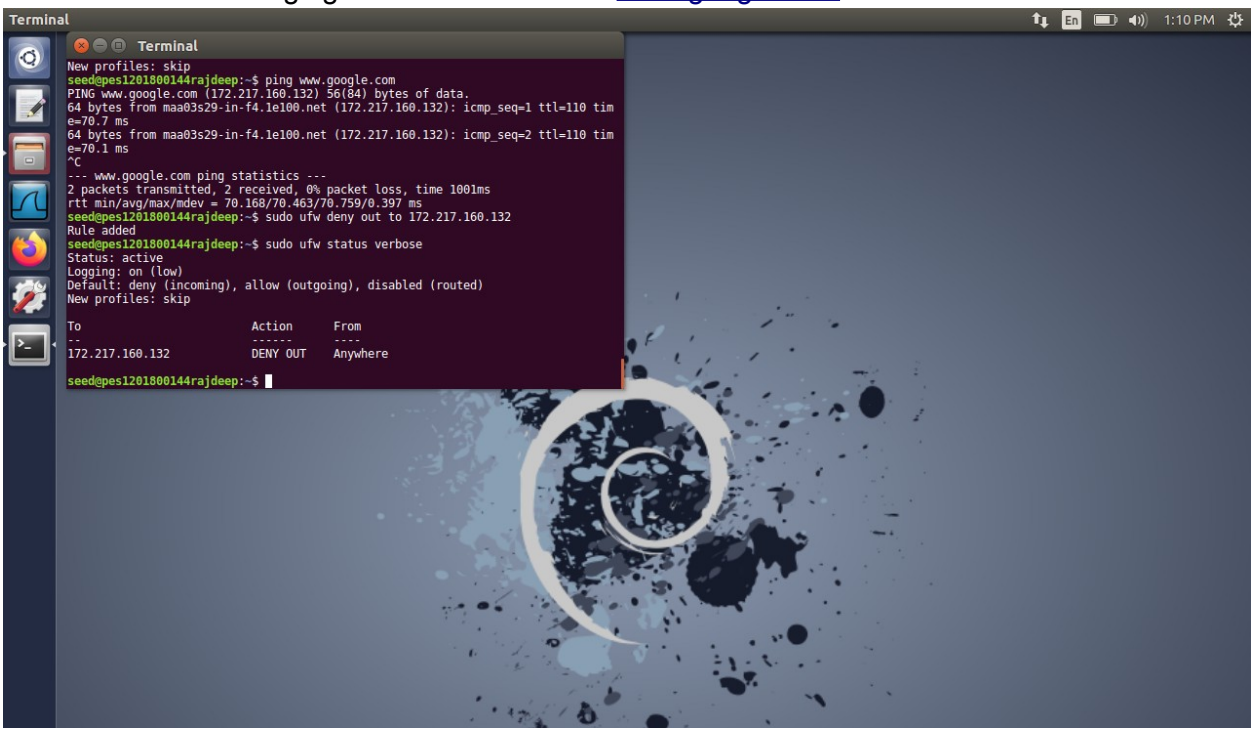
OBSERVATION:

ufw on VM1(10.0.2.13) blocks all telnet connections from port 23. To bypass this, a SSH tunnel is created between VM1 and VM2 and then telnetted to VM3(port 23). So the connection is telnet from VM1(port 8000) to VM3(port 23).

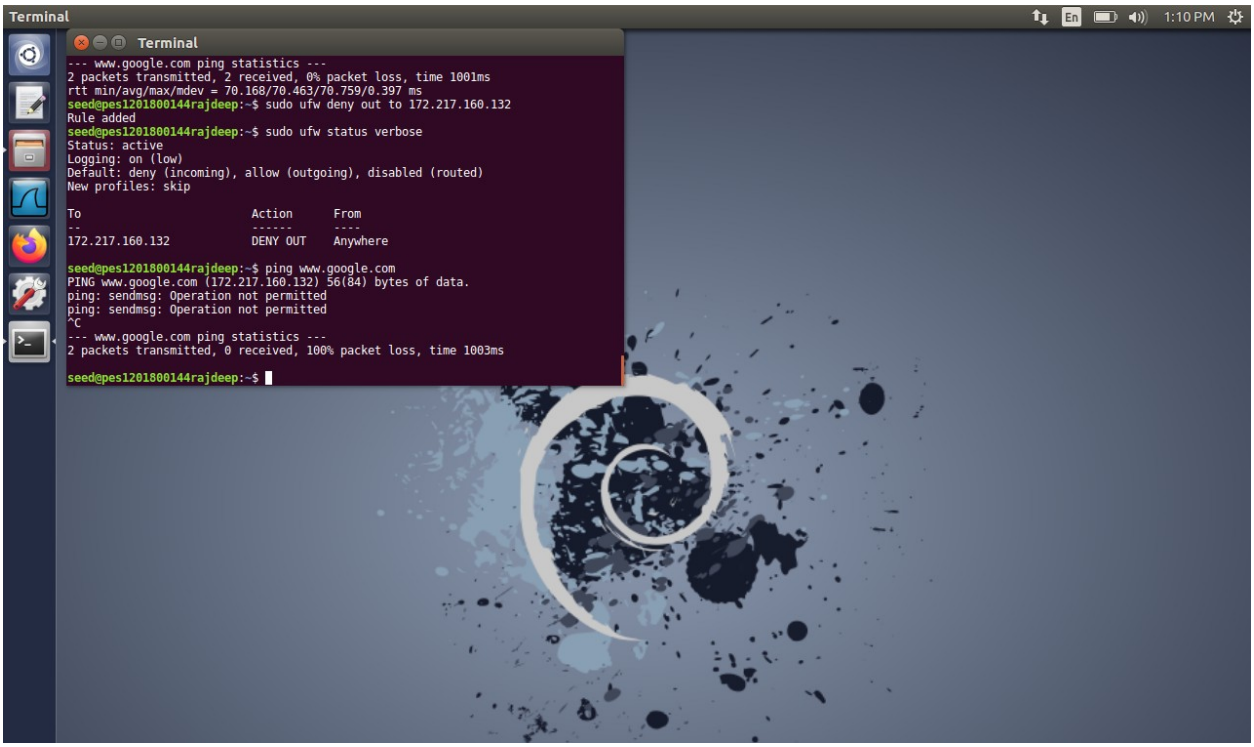
TASK 3b:



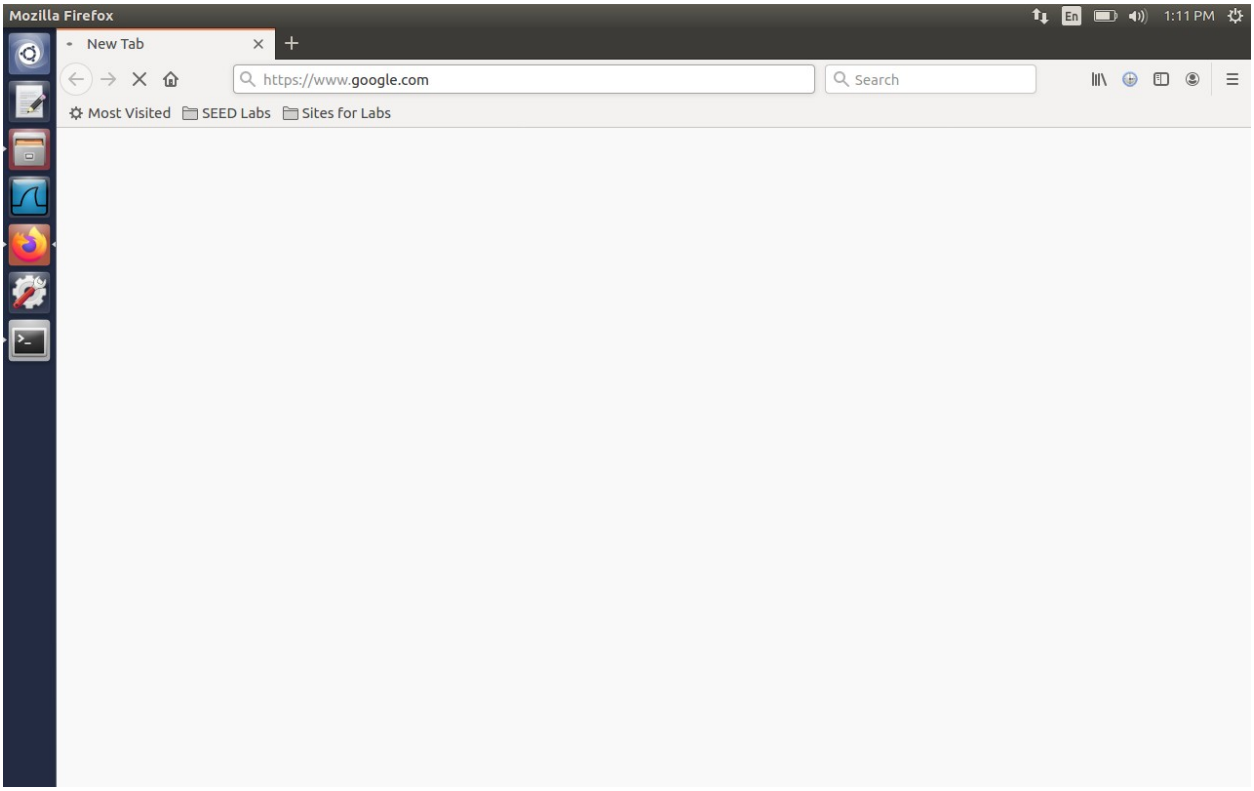
Screenshot 3.b.1: Pinging to find IP address of www.google.com



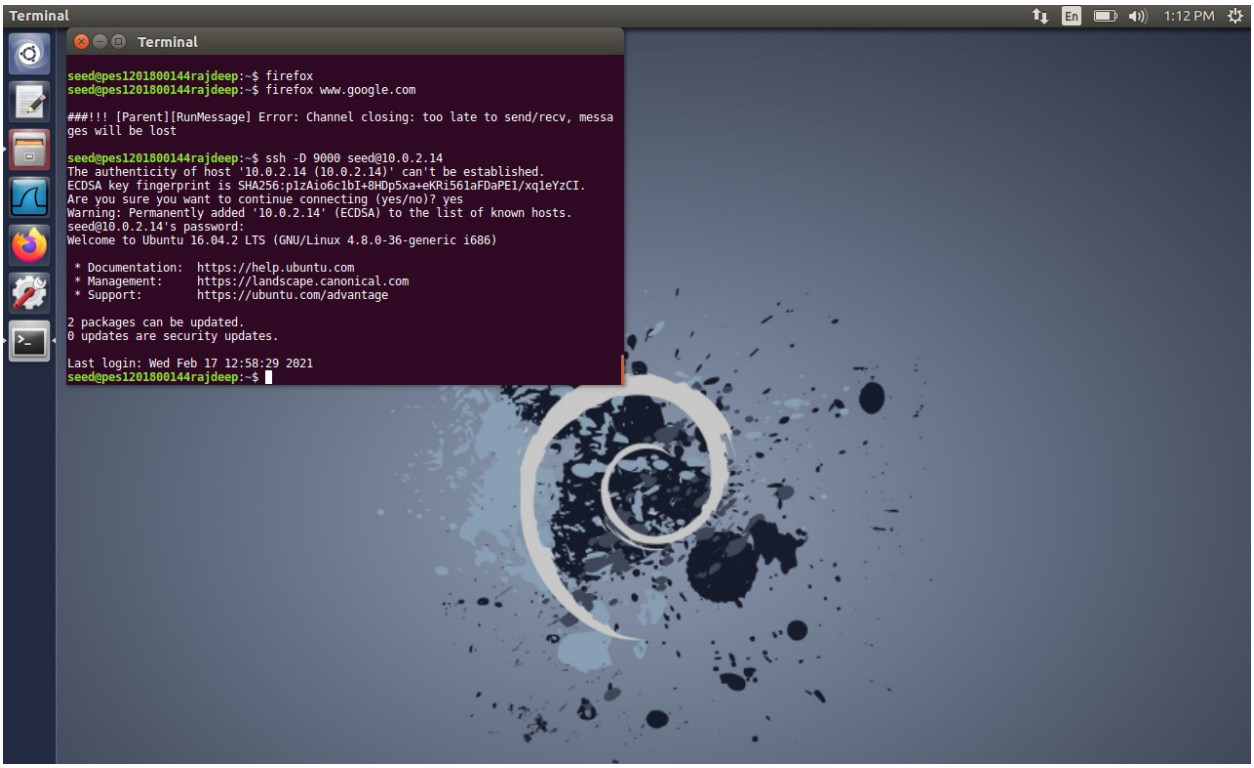
Screenshot 3.b.2: Denying connections to google.com IP address



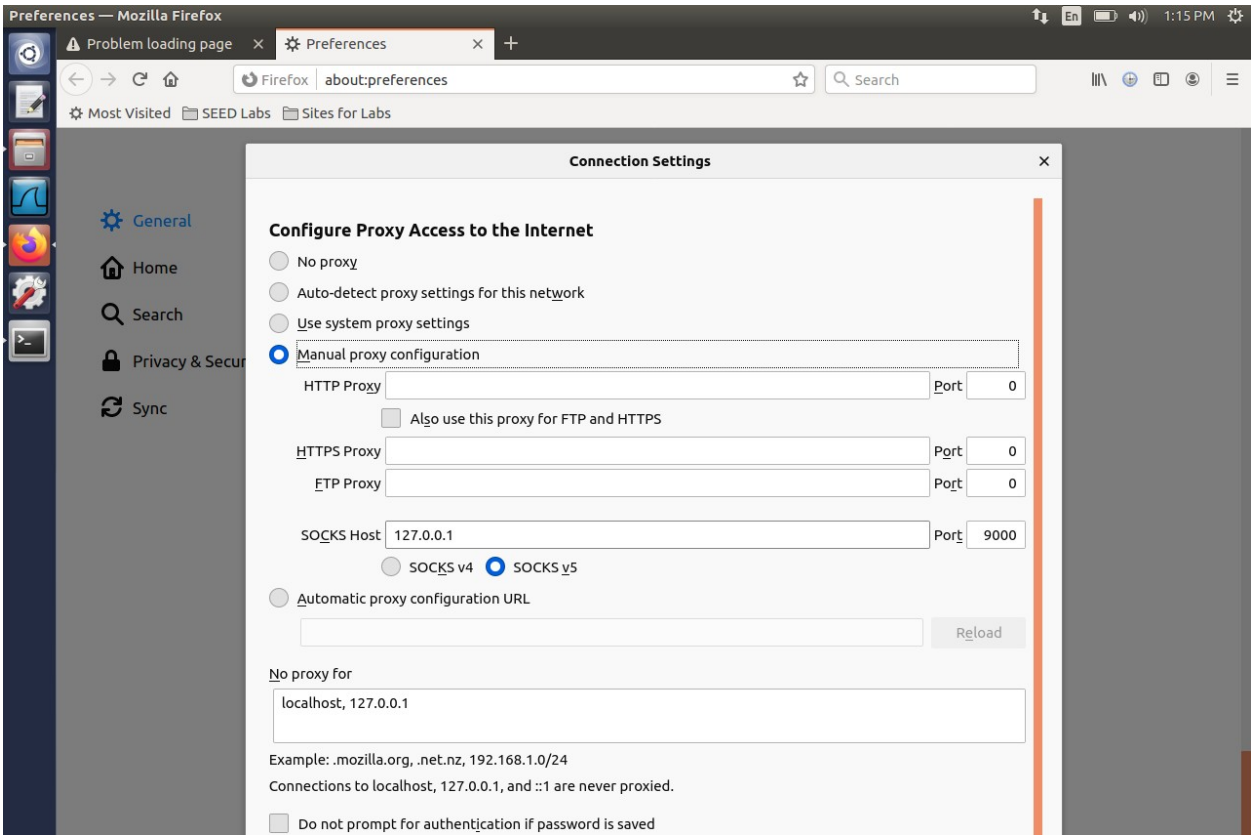
Screenshot 3.b.3: Ping operation blocked because of firewall



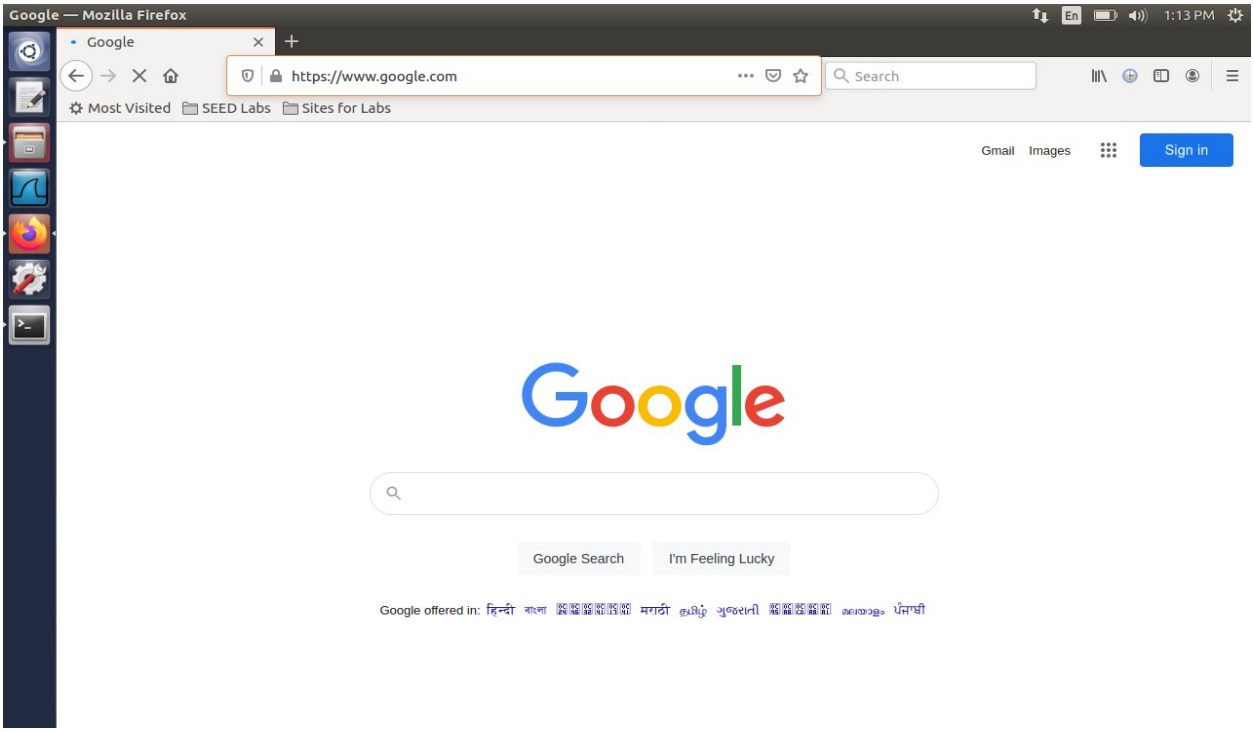
Screenshot 3.b.4: Unable to access google.com on browser too



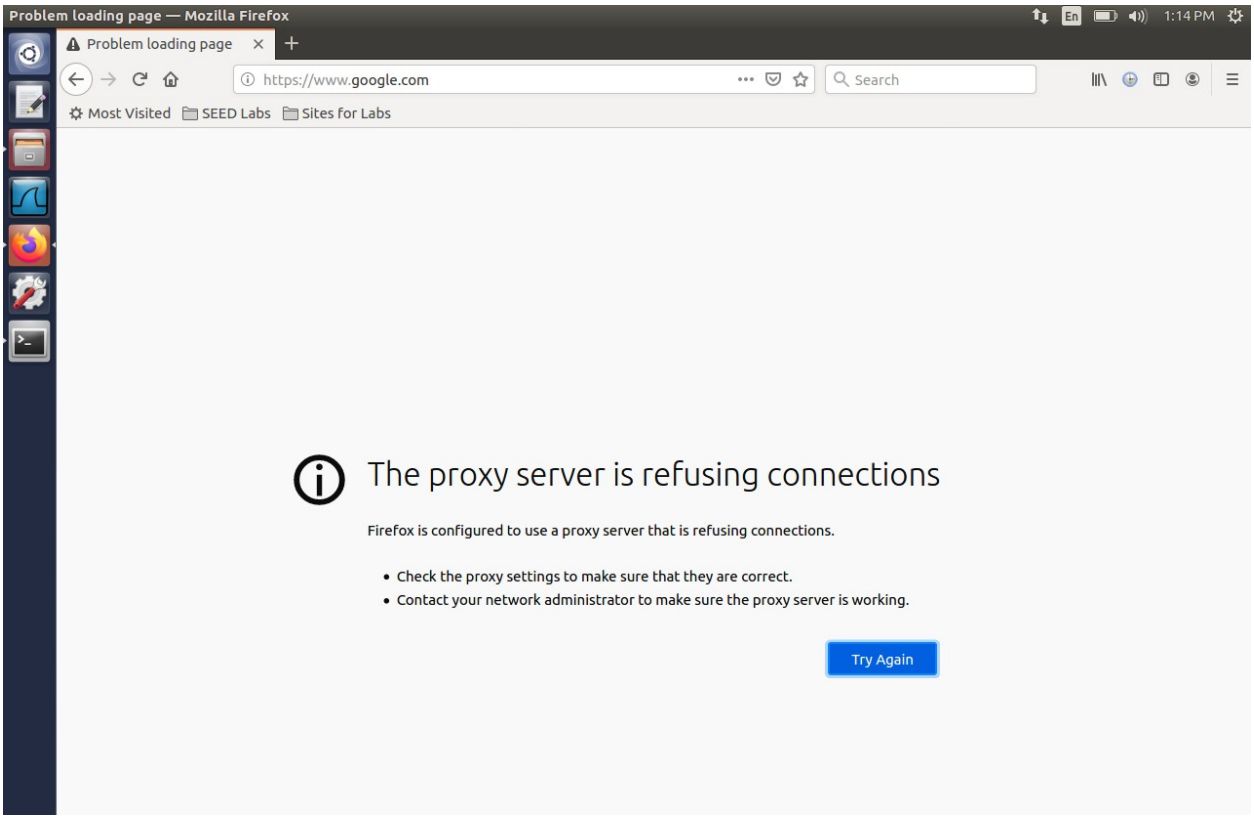
Screenshot 3.b.5: Creating SSH tunnel from VM1(10.0.2.13) to VM2(10.0.2.14) with dynamic port forwarding through port 9000



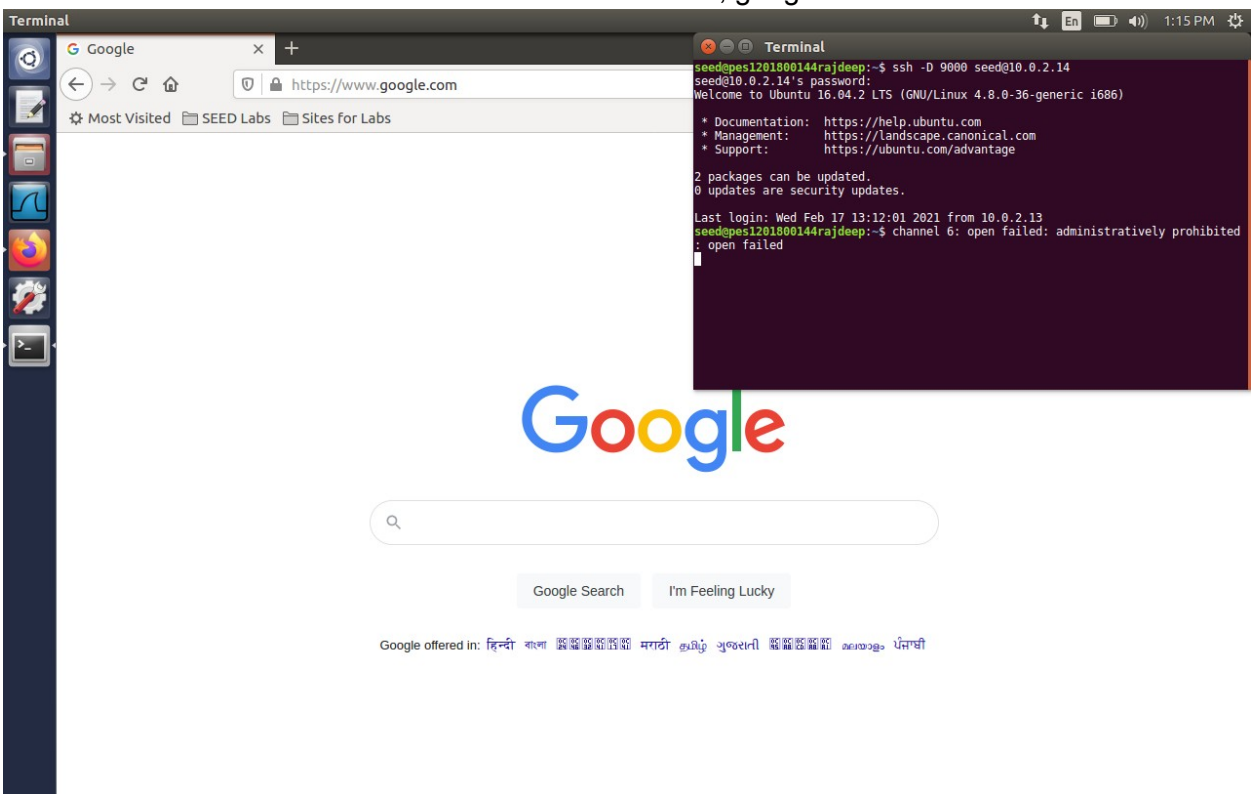
Screenshot 3.b.6: Setting up firefox network connections on VM1, SOCKS host is set to localhost but port is set to 9000



Screenshot 3.b.7: Able to access google.com even though firewall blocks direct connection to google.com



Screenshot 3.b.8: As soon as the SSH tunnel is broken, google.com cannot be accessed

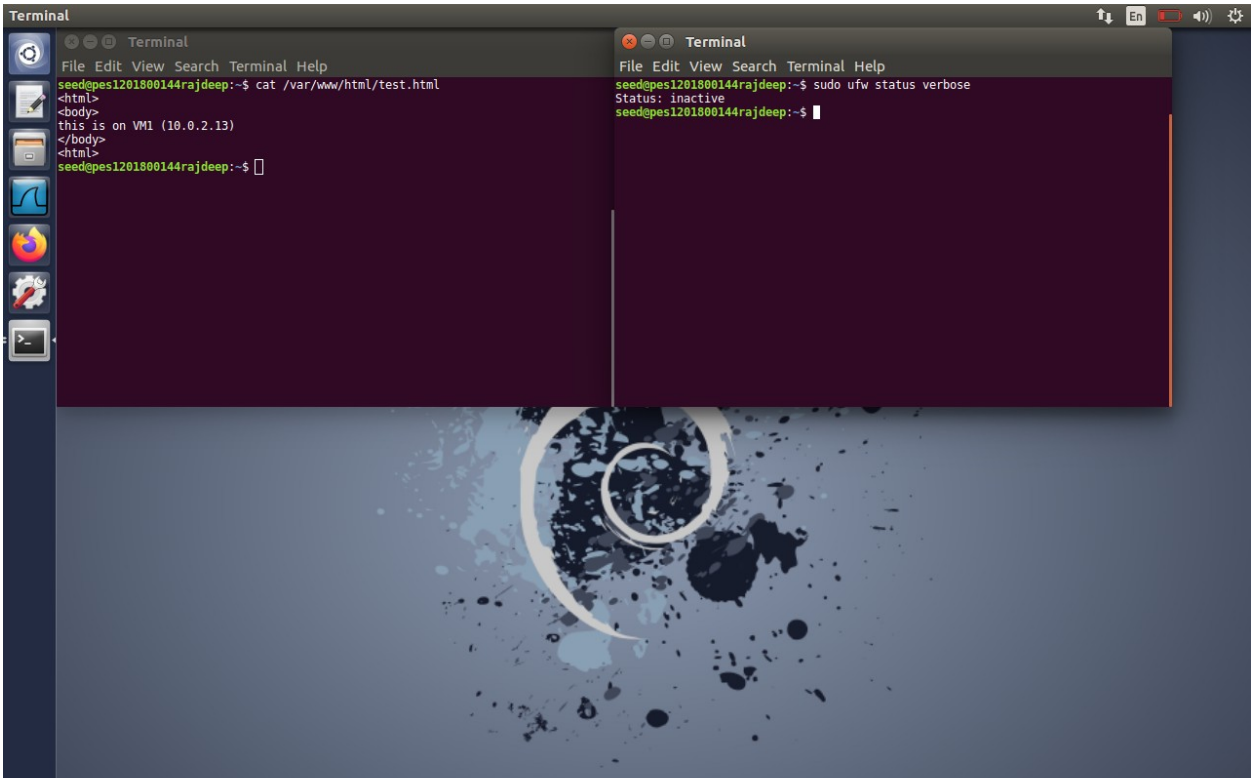


Screenshot 3.b.9: Again when the SSH tunnel is created from VM1 to VM2, google.com can be accessed

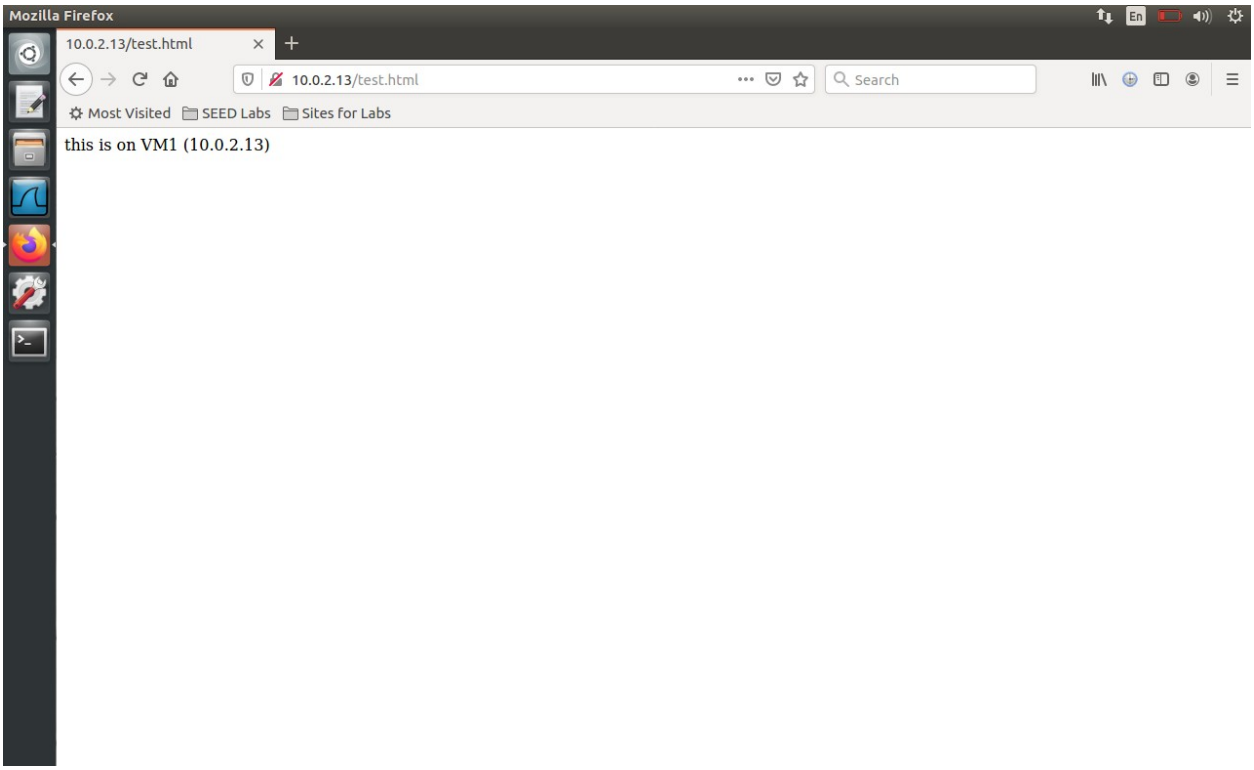
OBSERVATION:

Initially, firewall blocks connection from VM1(10.0.2.13) to google.com. So a SSH tunnel is created between VM1(10.0.2.13) and VM2(10.0.2.14) with dynamic port forwarding through port 9000. Then the firefox configurations are changed. The SOCKS port is set to 9000 on firefox. This allows HTTP request and response packets from VM1 to travel to VM2 and then connect to internet. Since VM2 can access google.com so it sends the response page(google.com) to VM1 which it displays. As soon as the SSH tunnel is broken, the google.com website is again inaccessible.

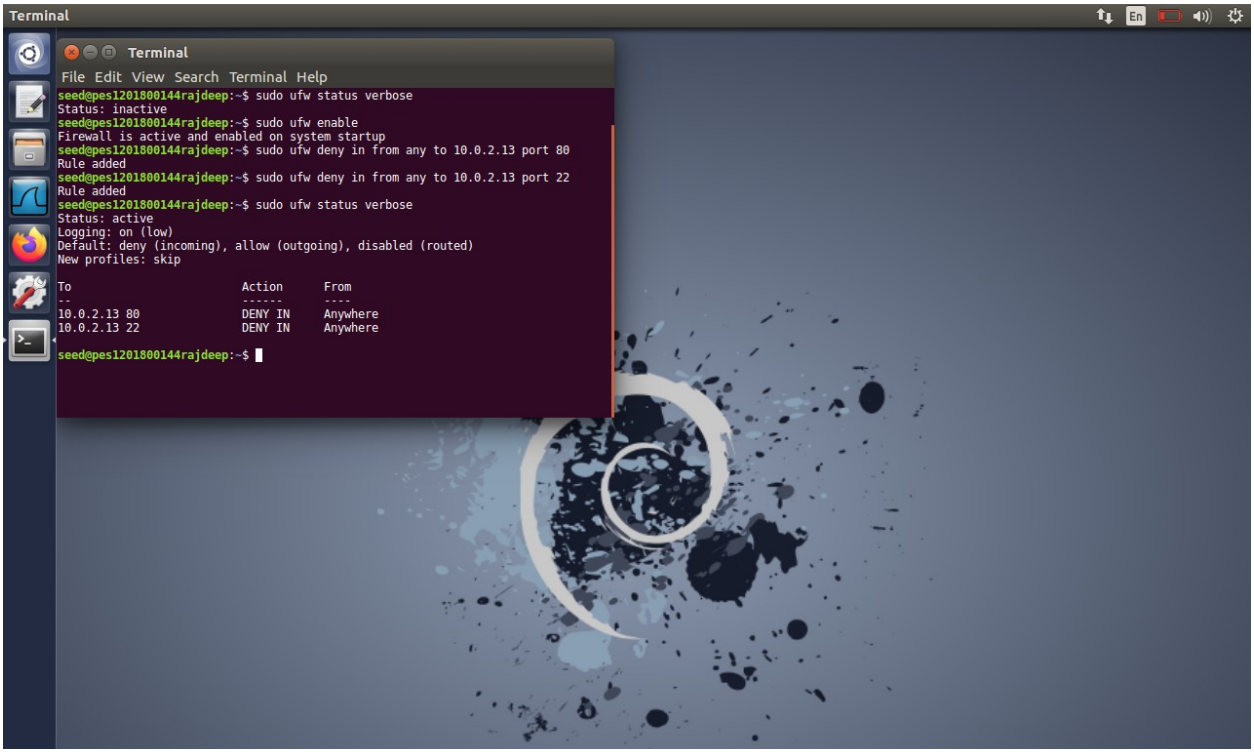
TASK 4:



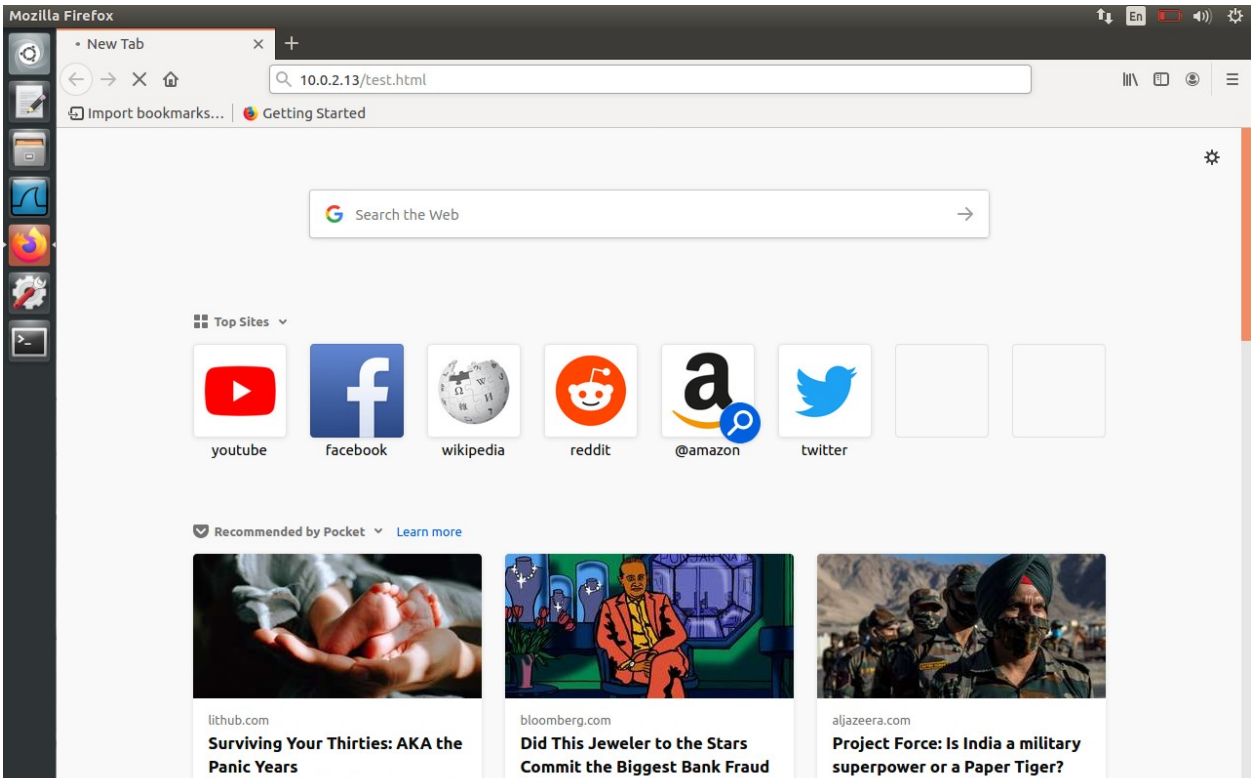
Screenshot 4.1: On VM1(10.0.2.13), an HTML file is hosted on apache2 server and firewall is disabled



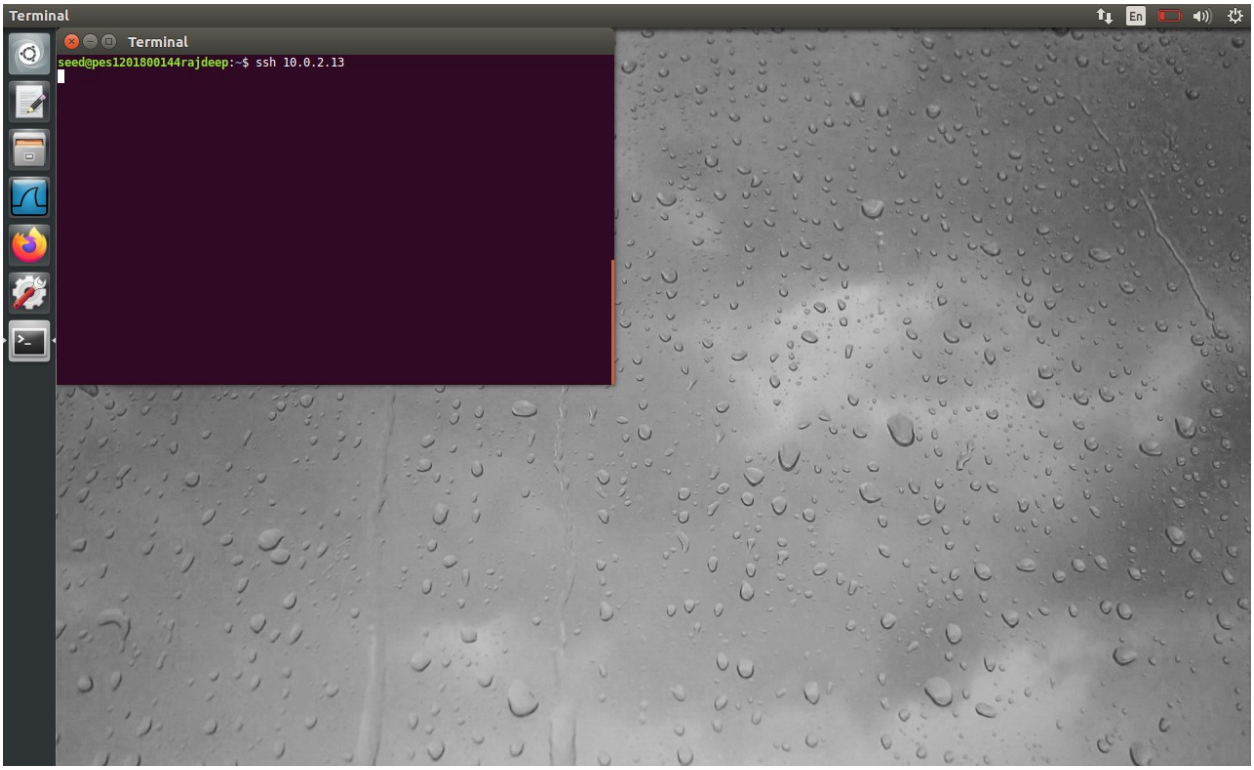
Screenshot 4.2: On VM2(10.0.2.14), the HTML file webpage can be accessed



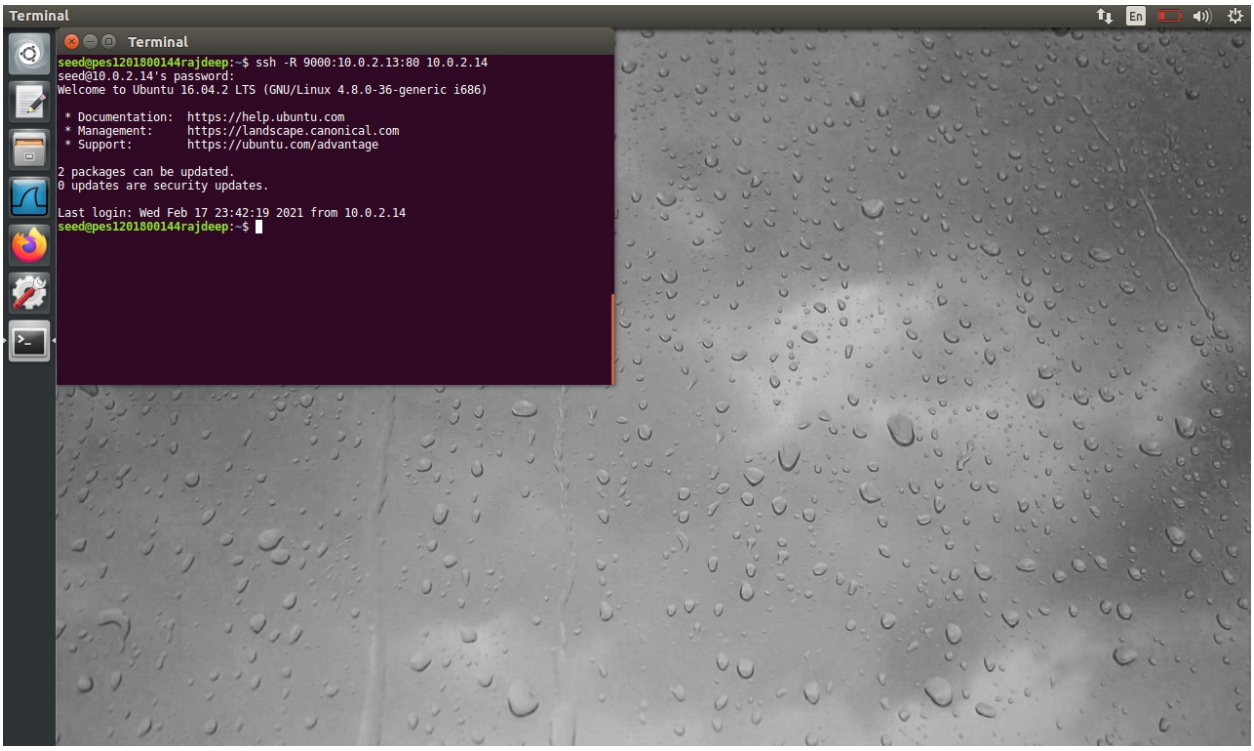
Screenshot 4.3: All connections to VM1(10.0.2.13) are denied through port 22(SSH) and port 80(HTTP) through firewall on VM1



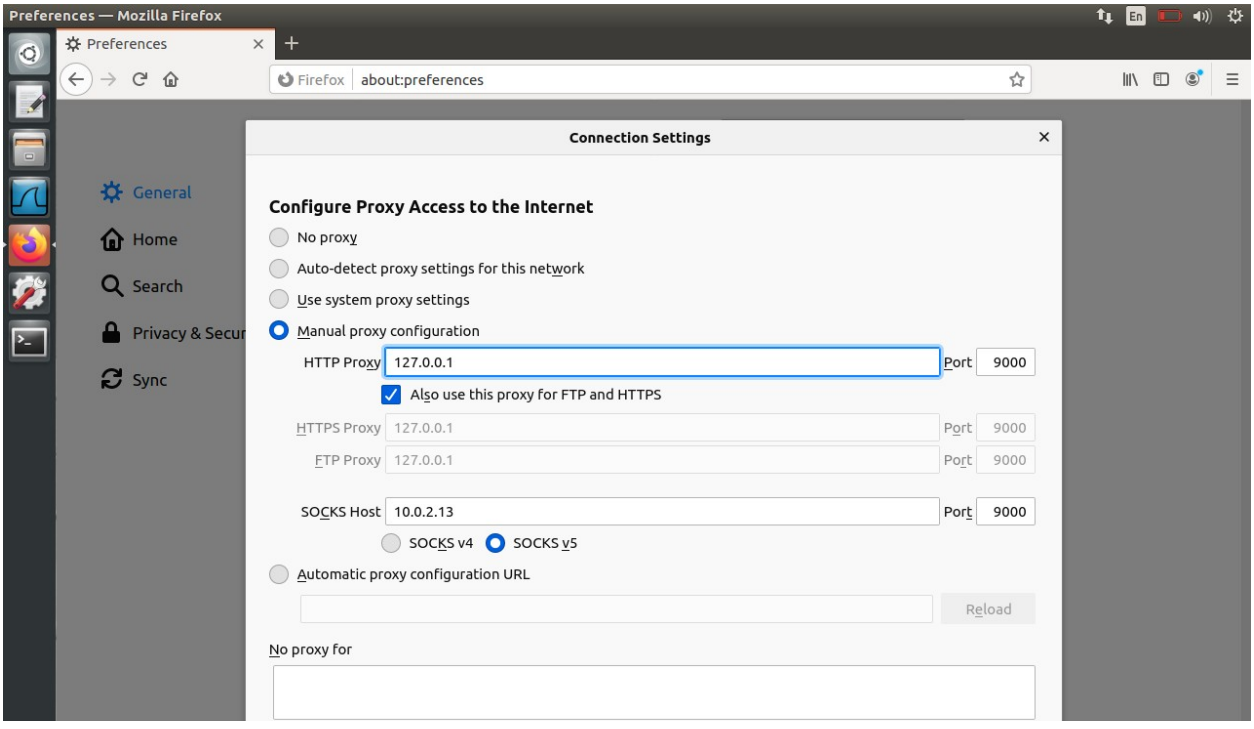
Screenshot 4.4: VM2(10.0.2.14) can no longer access the HTML file on VM1 through HTTP port 80



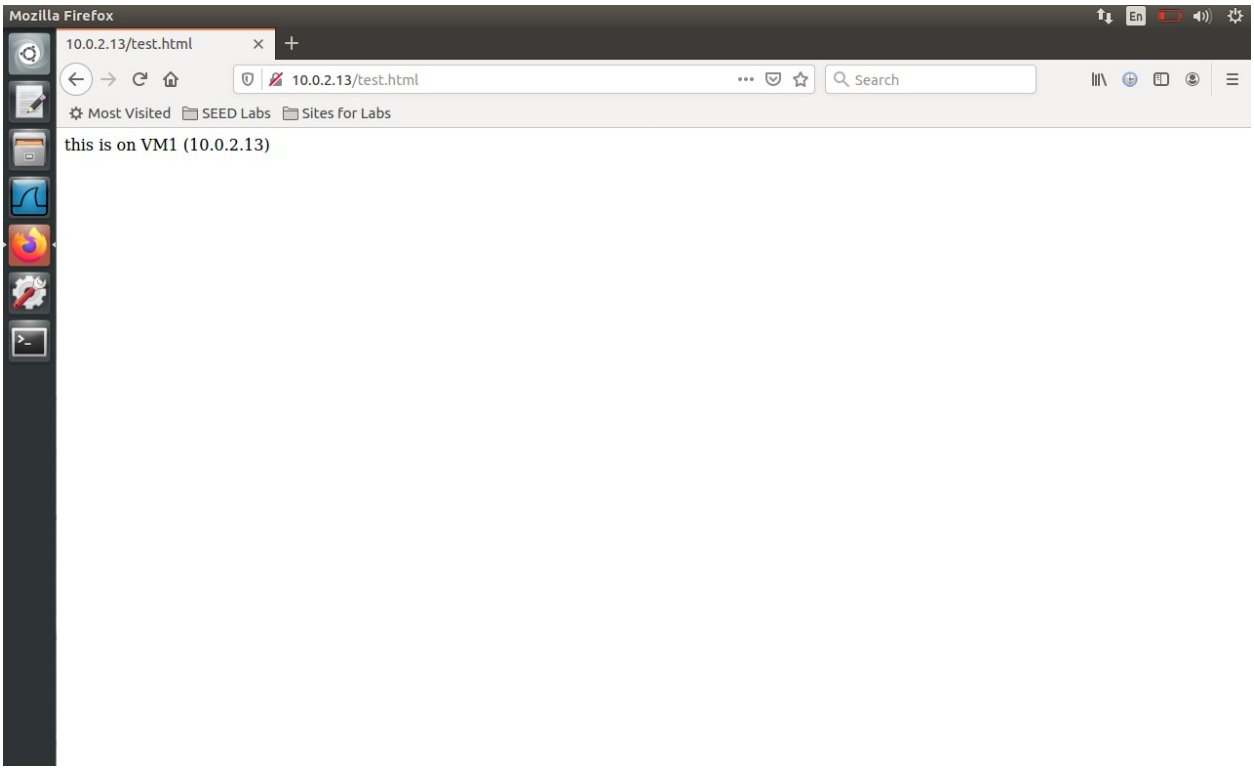
Screenshot 4.5: Also, SSH connection is unsuccessful from VM2 to VM1



Screenshot 4.6: A reverse SSH tunnel is created



Screenshot 4.7: Receiving connections from port 9000



Screenshot 4.8: HTML file on VM1 can be accessed by VM2

OBSERVATION:

A HTML file is hosted on apache server on VM1 and it's access through ports 80(HTTP) and 22(SSH) are blocked by firewall on VM1. Now VM2 cannot access this HTML file on VM1 through HTTP and SSH protocols. So to bypass this, VM2 creates a reverse tunnel. Now, all the HTTP requests and responses to and from VM1 through port 80 will happen through port 9000. This is achieved by reverse SSH tunnel. So now on firefox browser of VM2, changes are made so that the connections received are from port 9000, this makes connection to VM1 possible. Hence VM2 can access the HTML file on VM1.