

**COMPUTER NETWORKS SECURITY**  
**LABORATORY**

**LAB 3**

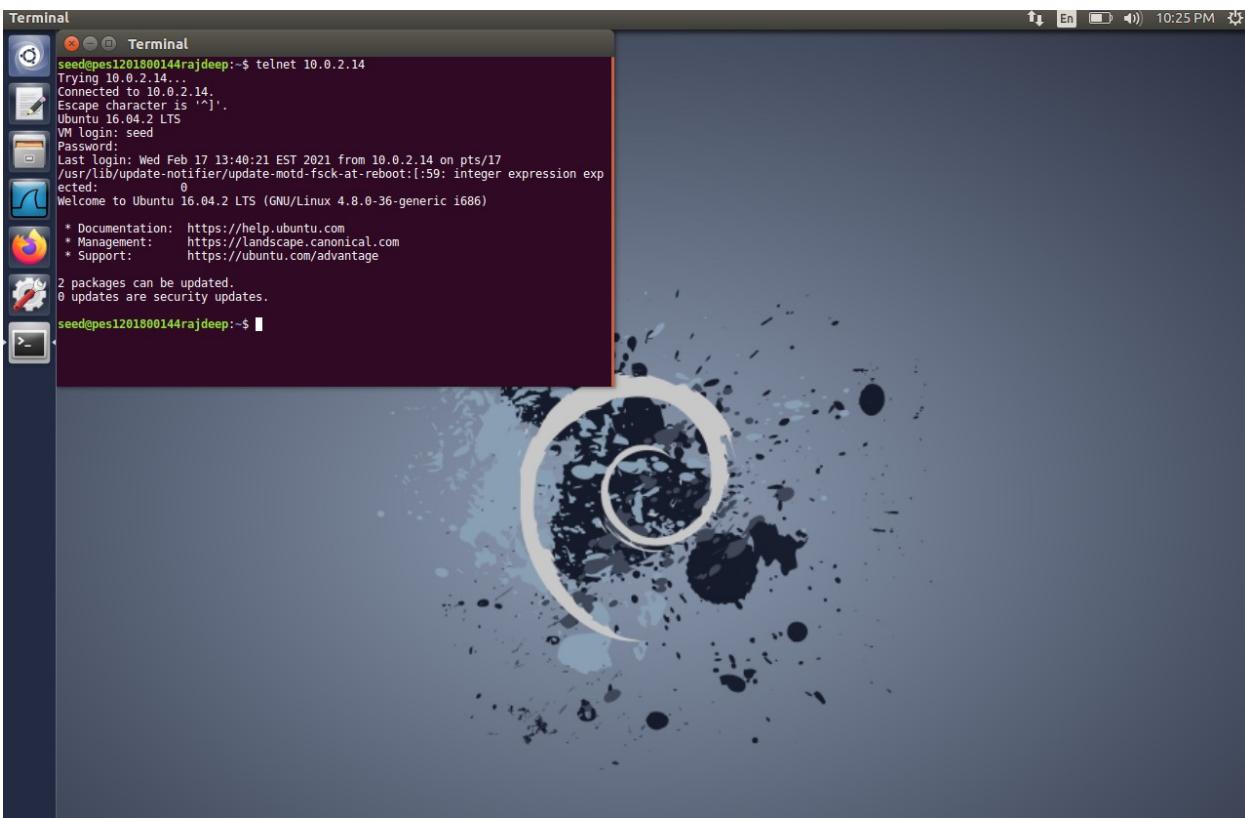
**BY: RAJDEEP SENGUPTA**

**SRN: PES1201800144**

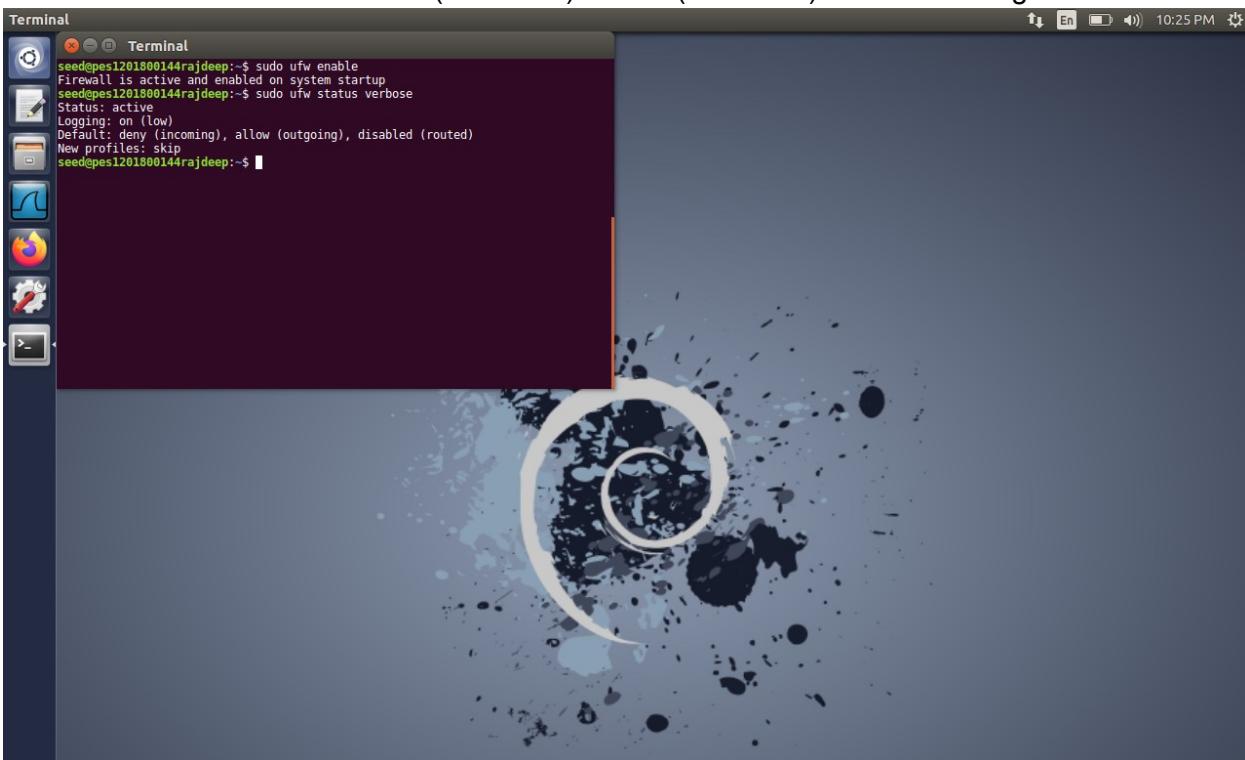
**SECTION: C**

**NOTE: Please find my SRN 'PES1201800144rajdeep' as the terminal  
username.**

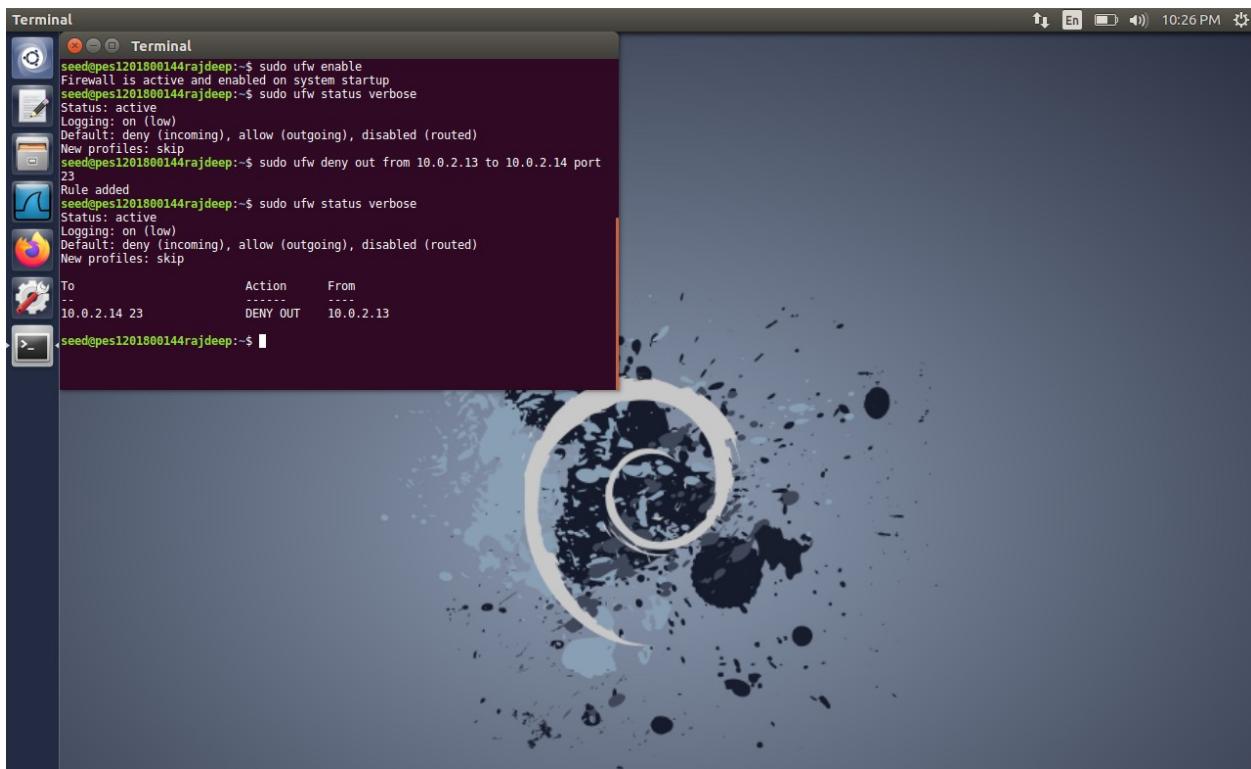
## TASK 1:



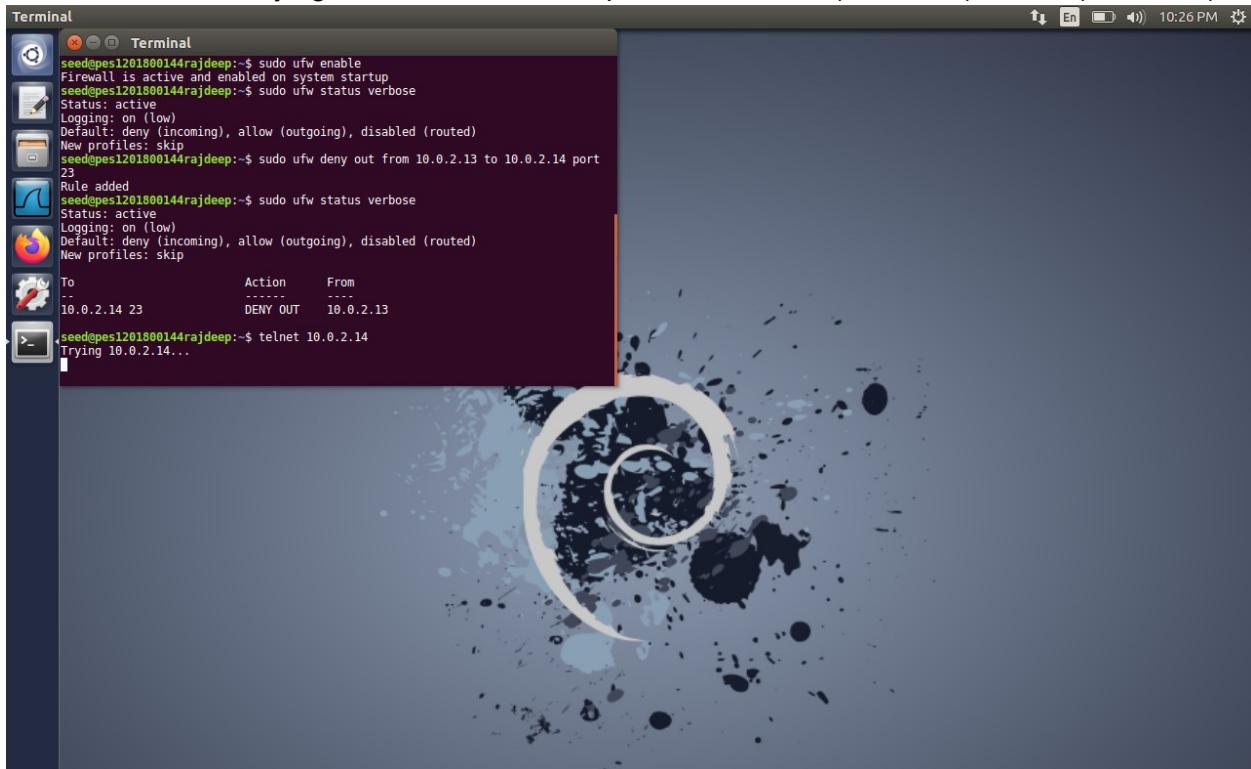
Screenshot 1.1: Telnet from VM1(10.0.2.13) to VM2(10.0.2.14) before enabling ufw



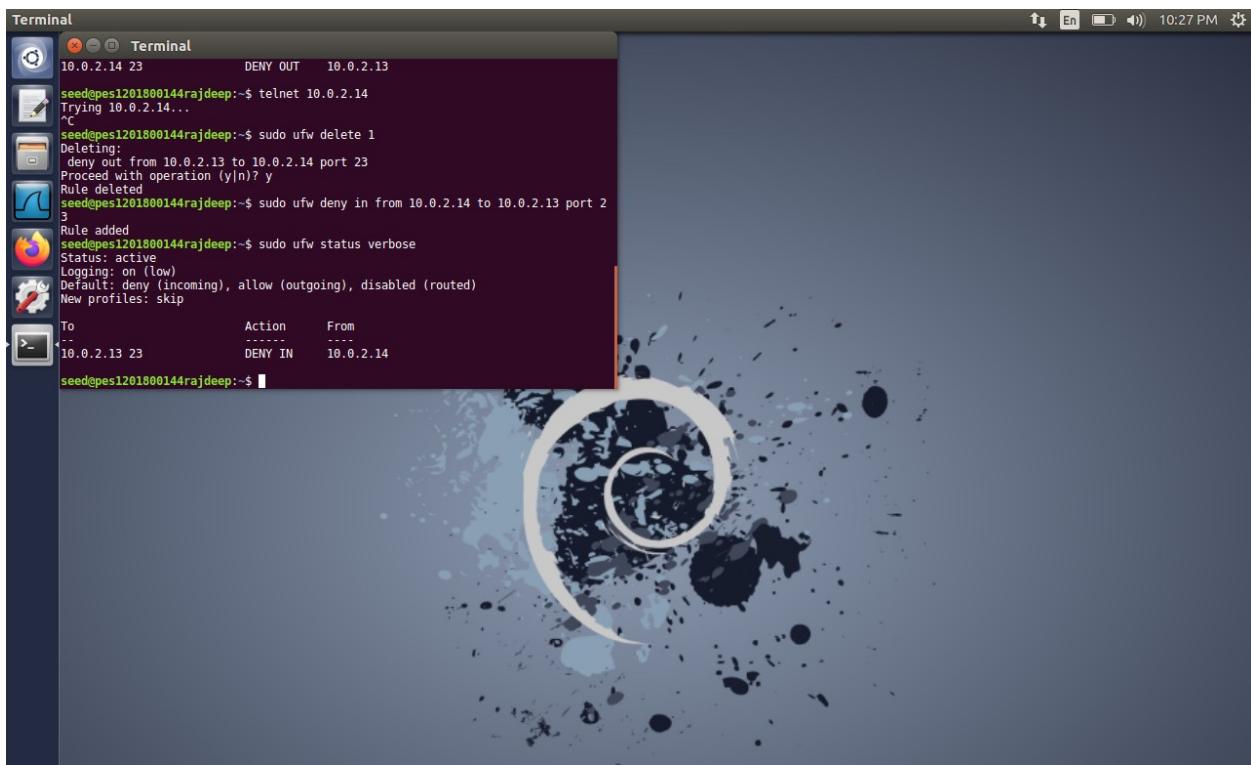
Screenshot 1.2: Enabling firewall and displaying status



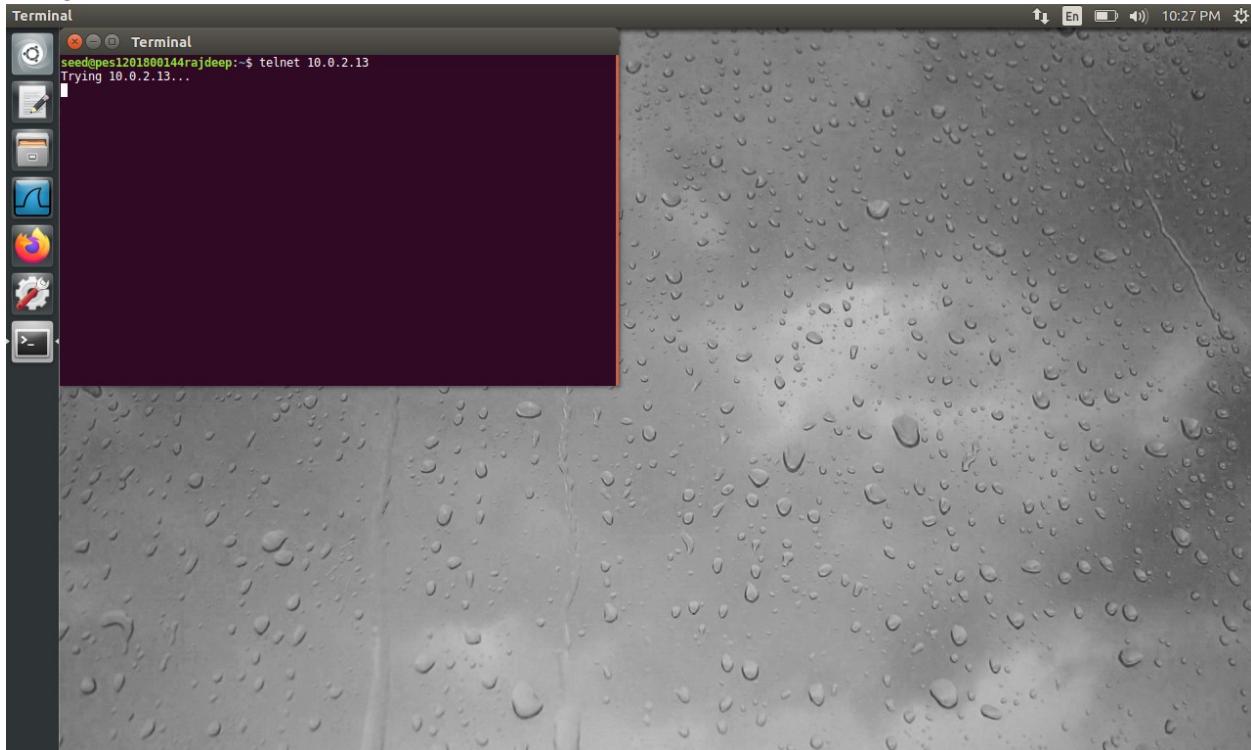
Screenshot 1.3: Denying telnet connection on port 23 from VM1(10.0.2.13) to VM2(10.0.2.14)



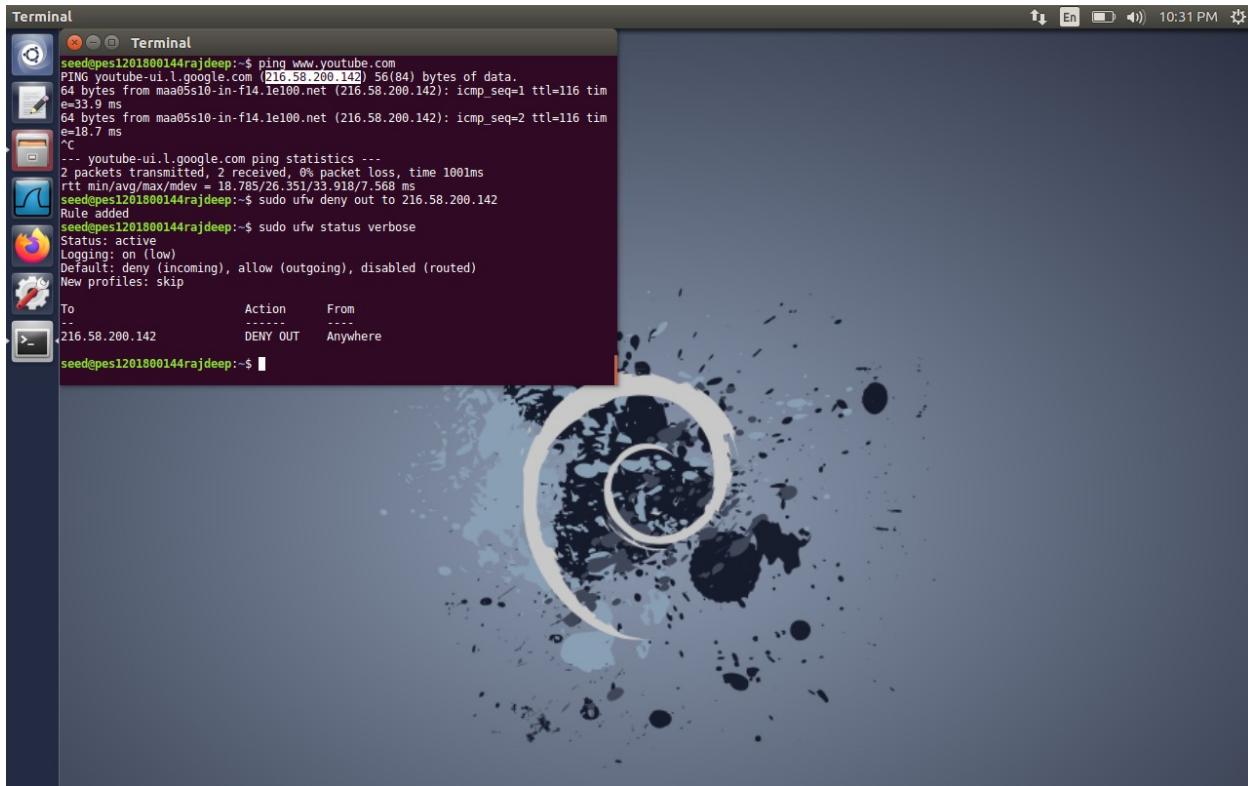
Screenshot 1.4: Trying telnet connection which is unsuccessful since it is denied by firewall



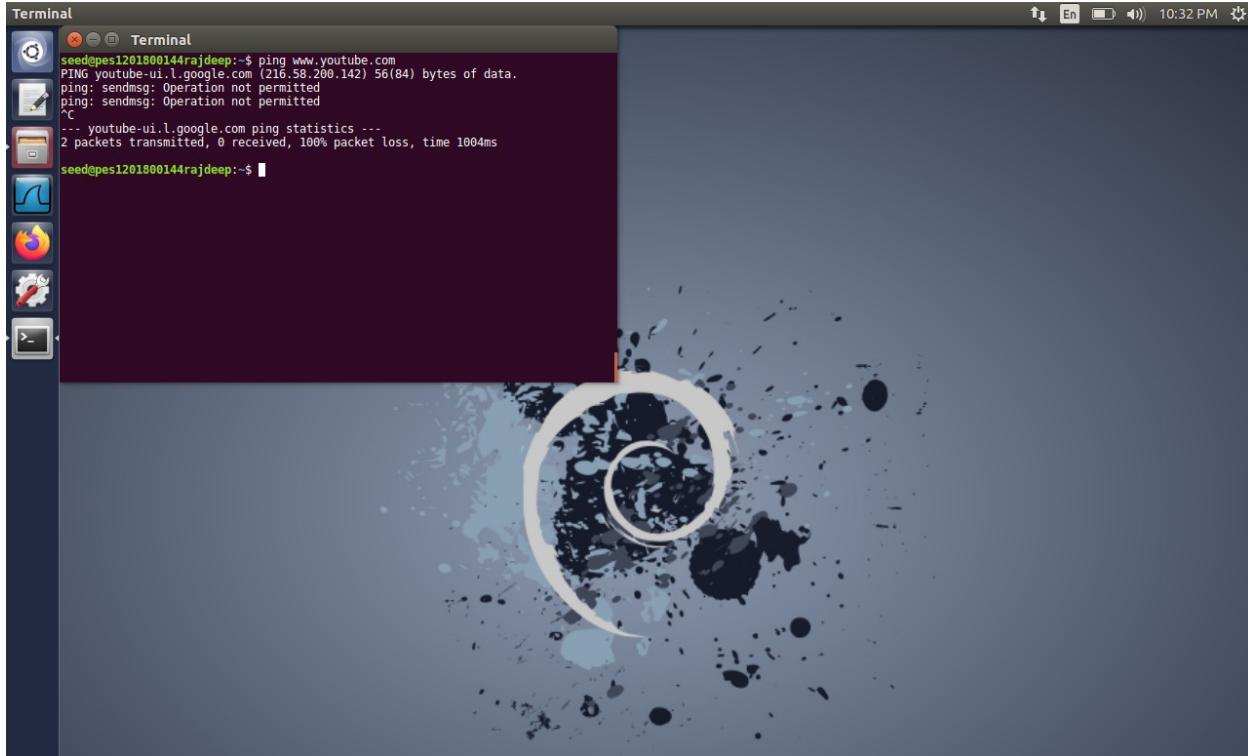
Screenshot 1.5: Denying telnet connection on port 23 from VM2(10.0.2.14) to VM1(10.0.2.13) using firewall



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

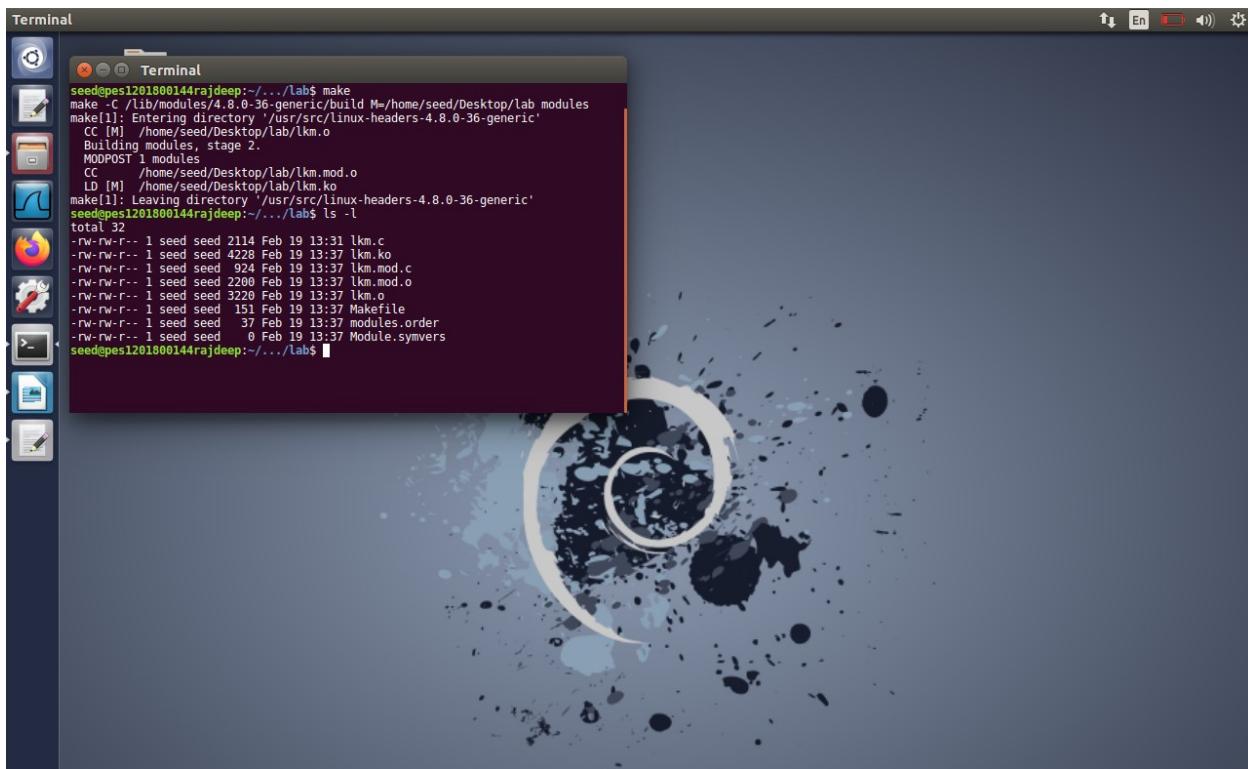


Screenshot 1.7: Pinging [www.youtube.com](http://www.youtube.com) to find it's IP address and then denying outgoing connections to that IP address

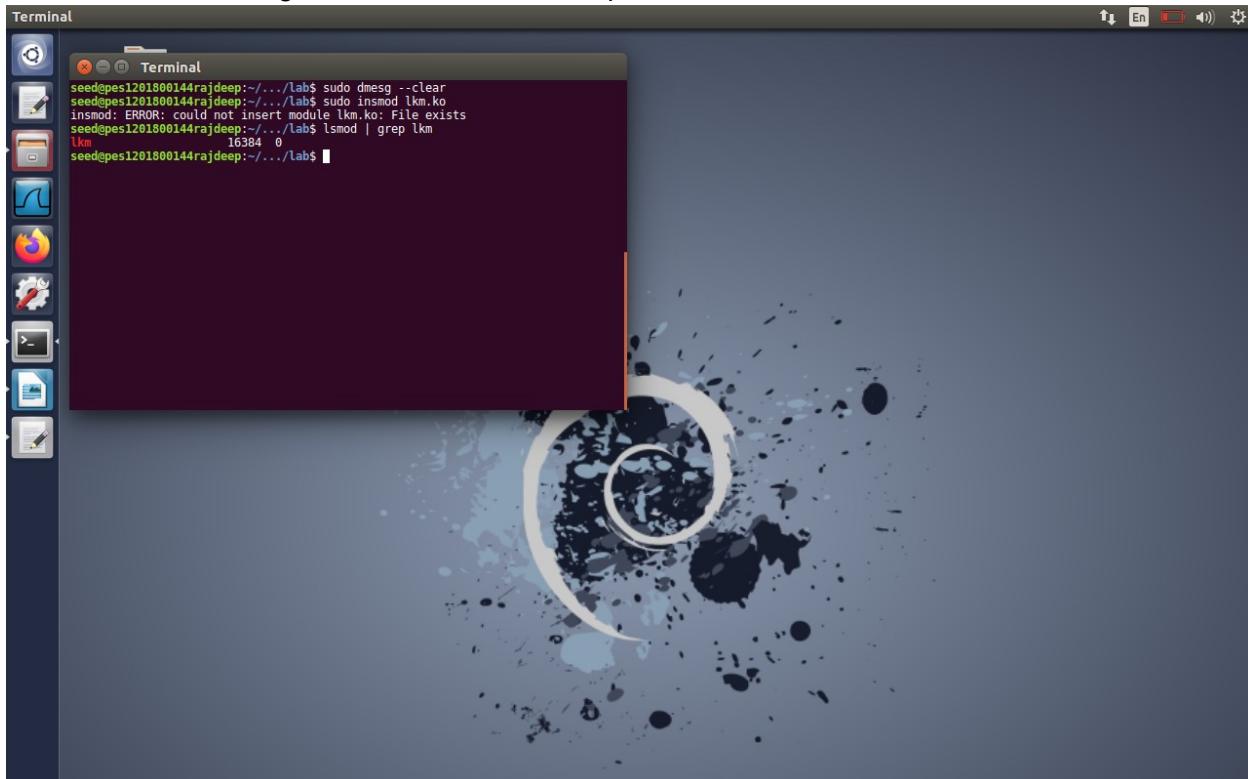


Screenshot 1.8: Pinging to [www.youtube.com](http://www.youtube.com) but it shows “Operation not permitted” since the connection is denied by firewall of VM1

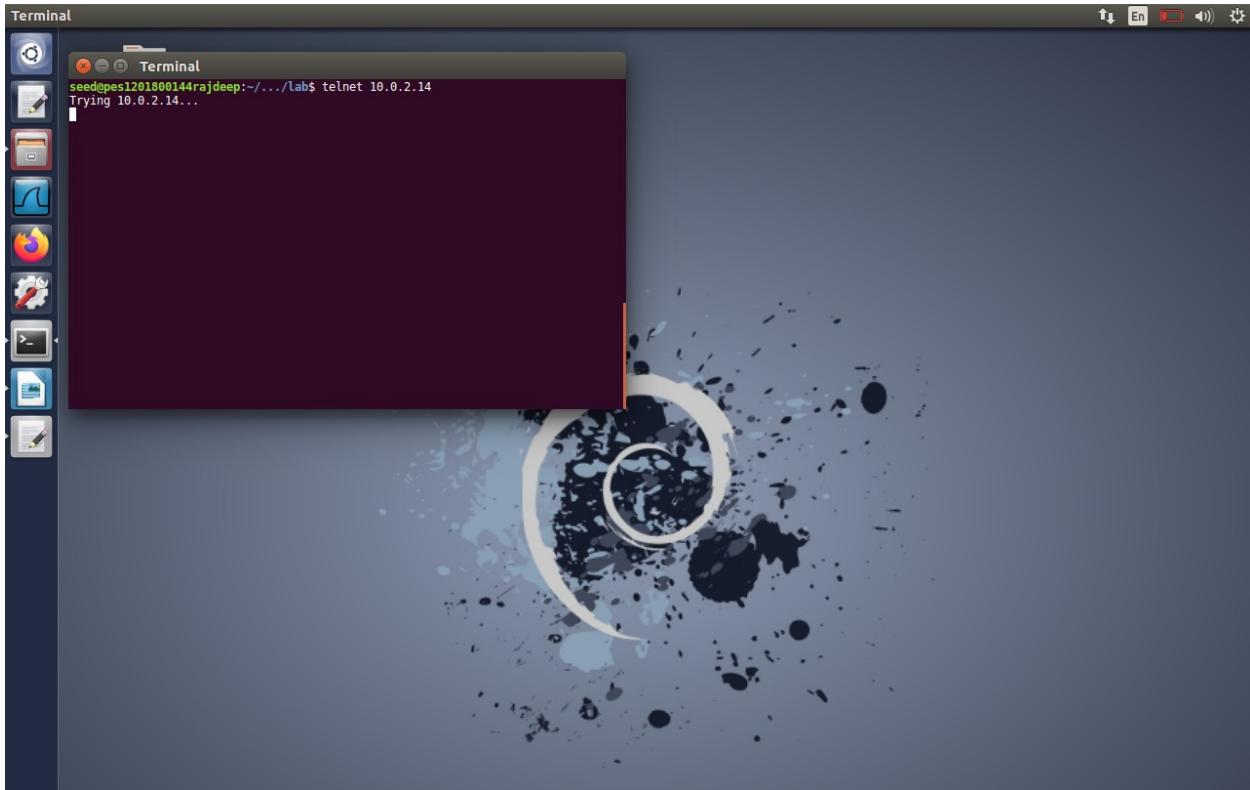
## TASK 2:



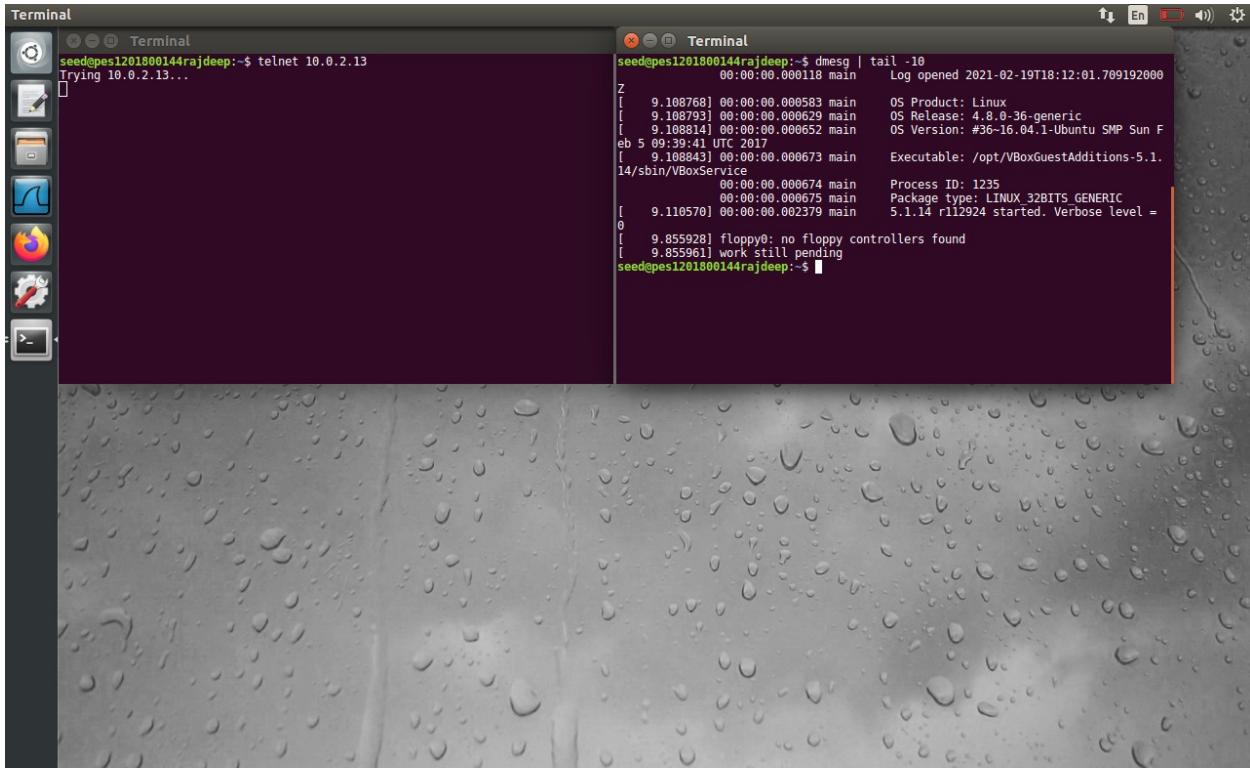
Screenshot 2.1: Using make command to compile the kernel module



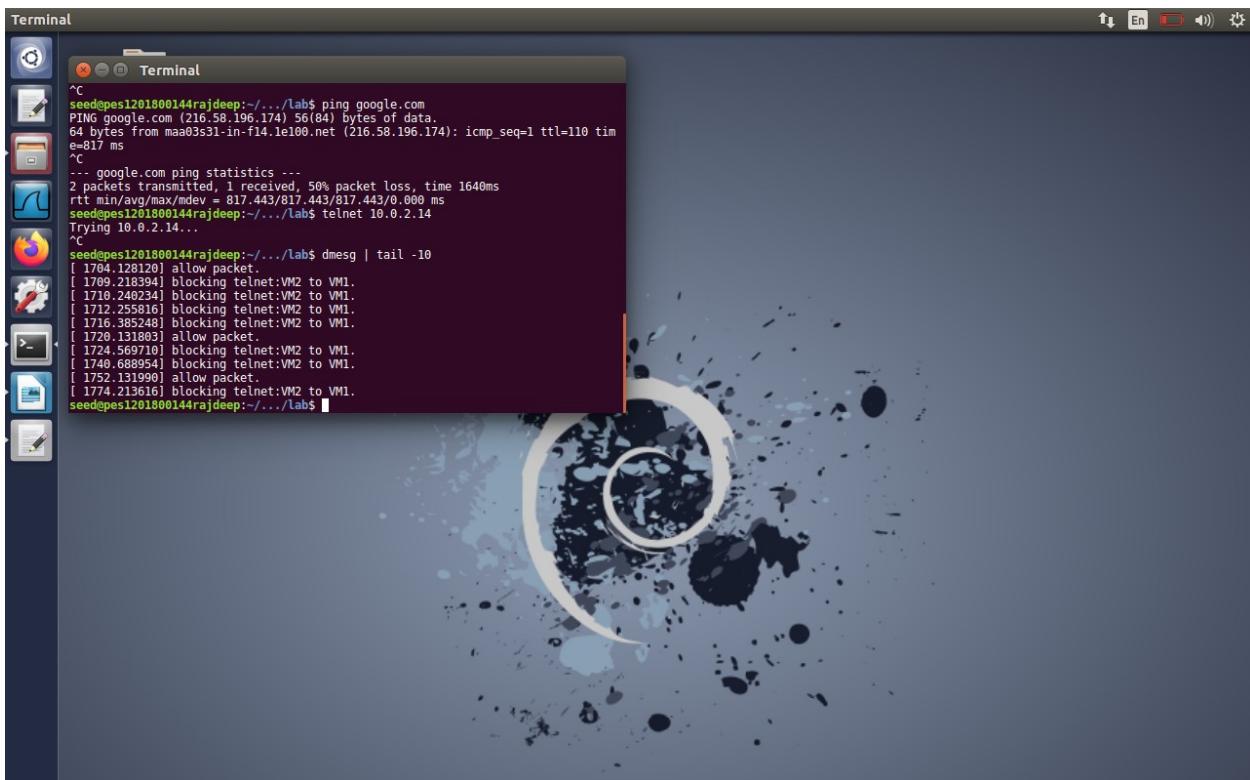
Screenshot 2.2: Clearing the dmesg buffer



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



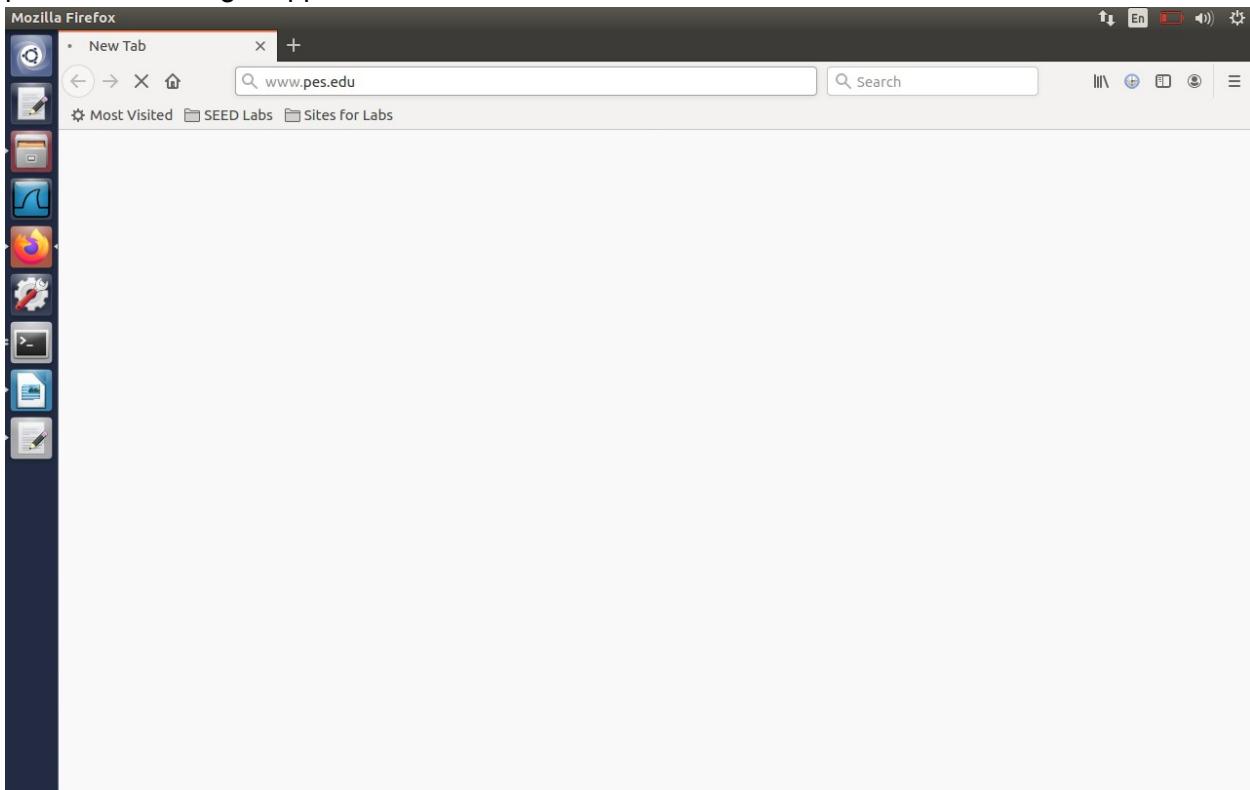
Screenshot 2.4: Trying telnet from VM2(10.0.2.14) to VM1(10.0.2.13) but it is blocked



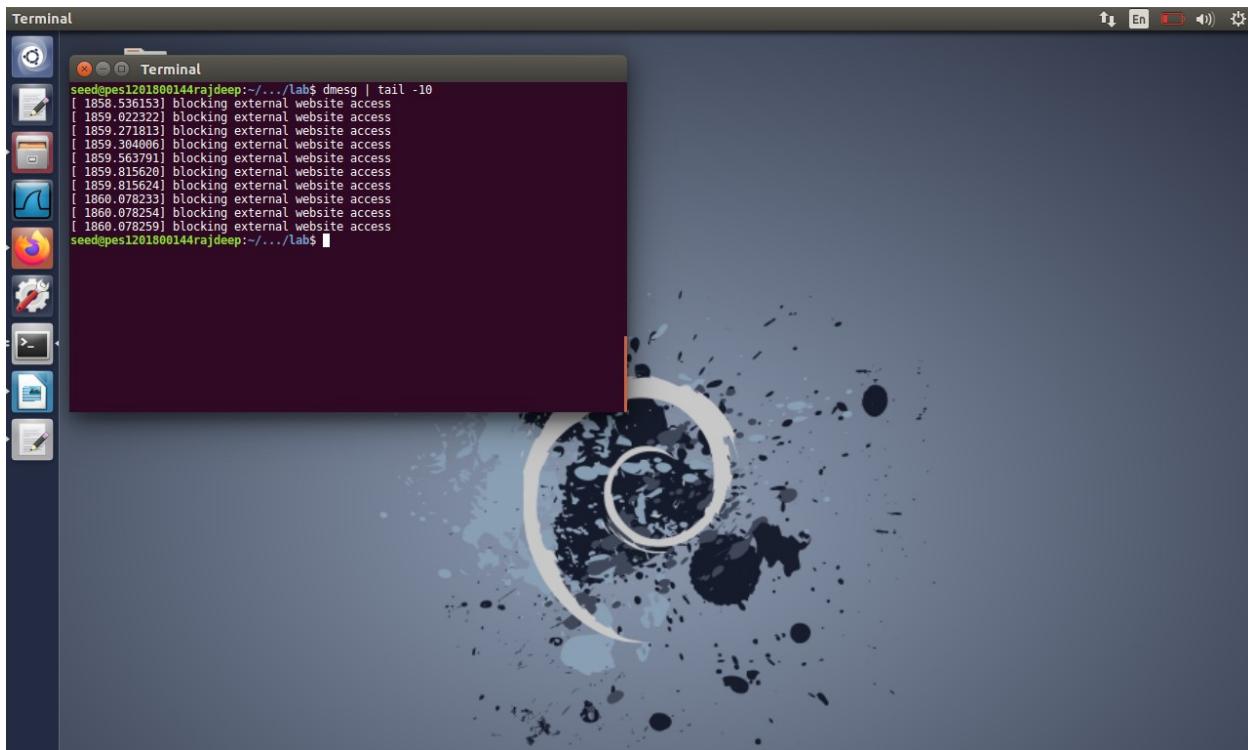
A screenshot of a Linux desktop environment. In the foreground, a terminal window titled "Terminal" is open, displaying command-line output. The output shows a ping to google.com and a dmesg log with numerous entries like "blocking telnet:VM2 to VM1". The desktop background features a blue and white abstract design.

```
^C
seed@pes1201800144rajdeep:~/.../lab$ ping google.com
PING google.com (216.58.196.174) 56(84) bytes of data.
64 bytes from maa03s31-in-f14.1e100.net (216.58.196.174): icmp_seq=1 ttl=110 time=817 ms
^C
... google.com ping statistics ...
2 packets transmitted, 1 received, 50% packet loss, time 1640ms
rtt min/avg/max/mdev = 817.443/817.443/817.443/0.000 ms
seed@pes1201800144rajdeep:~/.../lab$ telnet 10.0.2.14...
Trying 10.0.2.14...
^C
seed@pes1201800144rajdeep:~/.../lab$ dmesg | tail -10
[ 1784.128128] allow_packet
[ 1789.218394] blocking telnet:VM2 to VM1.
[ 1710.249234] blocking telnet:VM2 to VM1.
[ 1712.255816] blocking telnet:VM2 to VM1.
[ 1716.385248] blocking telnet:VM2 to VM1.
[ 1720.131803] allow_packet.
[ 1724.569710] blocking telnet:VM2 to VM1.
[ 1740.688954] blocking telnet:VM2 to VM1.
[ 1752.131990] allow_packet.
[ 1774.213616] blocking telnet:VM2 to VM1.
seed@pes1201800144rajdeep:~/.../lab$
```

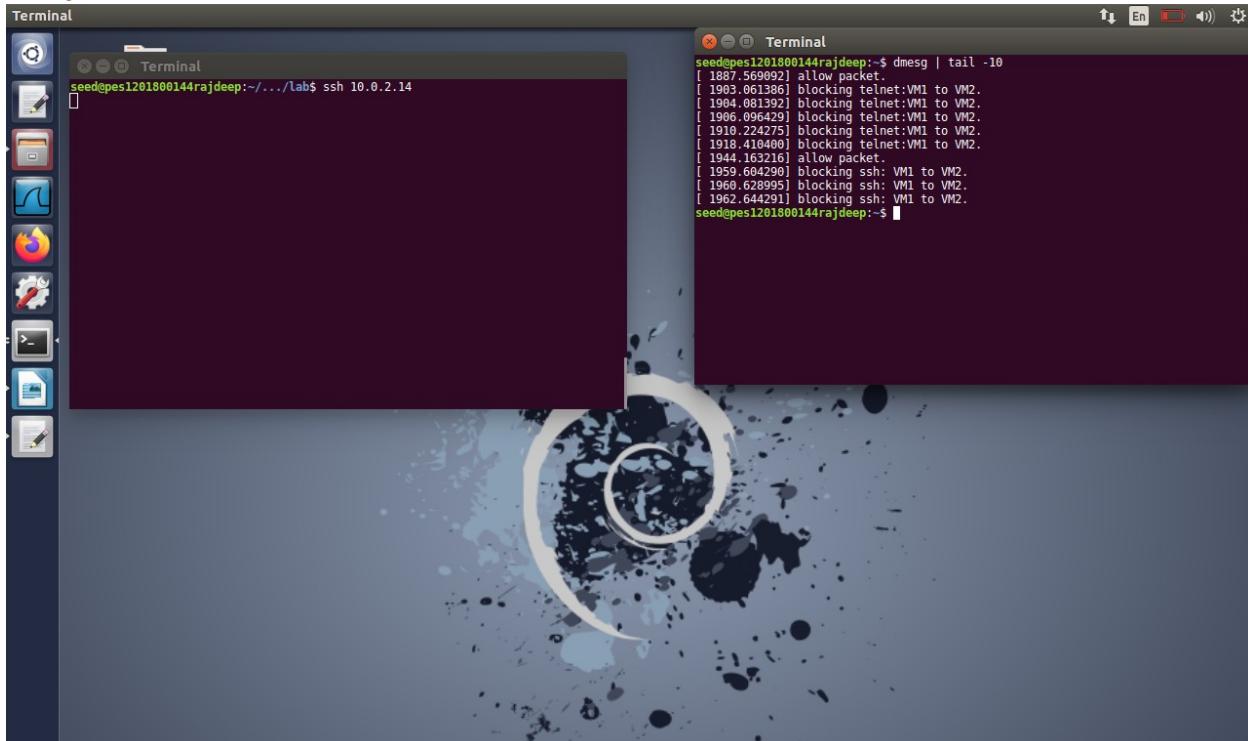
Screenshot 2.6: It can be seen that the telnet connections from VM2 to VM1 are blocked as the packets are being dropped



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



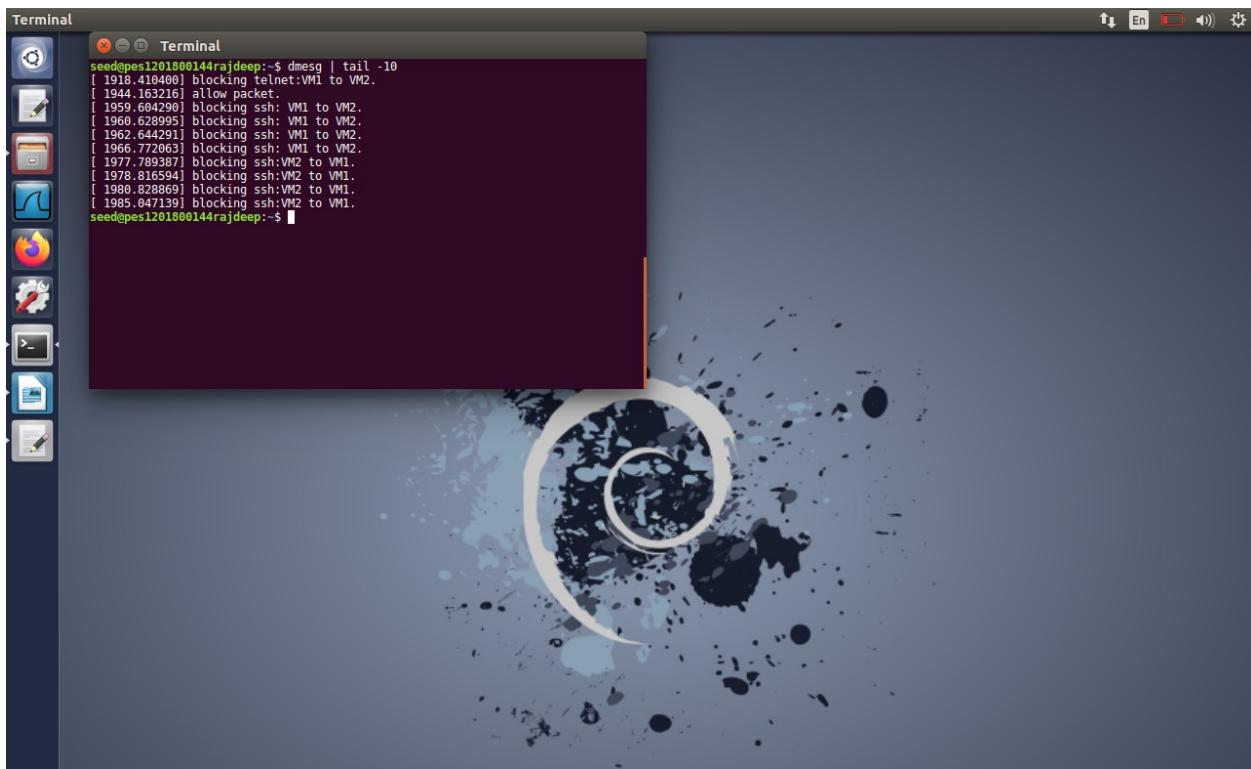
Screenshot 2.8: It can be seen that the external website access is blocked since the packets are being dropped



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

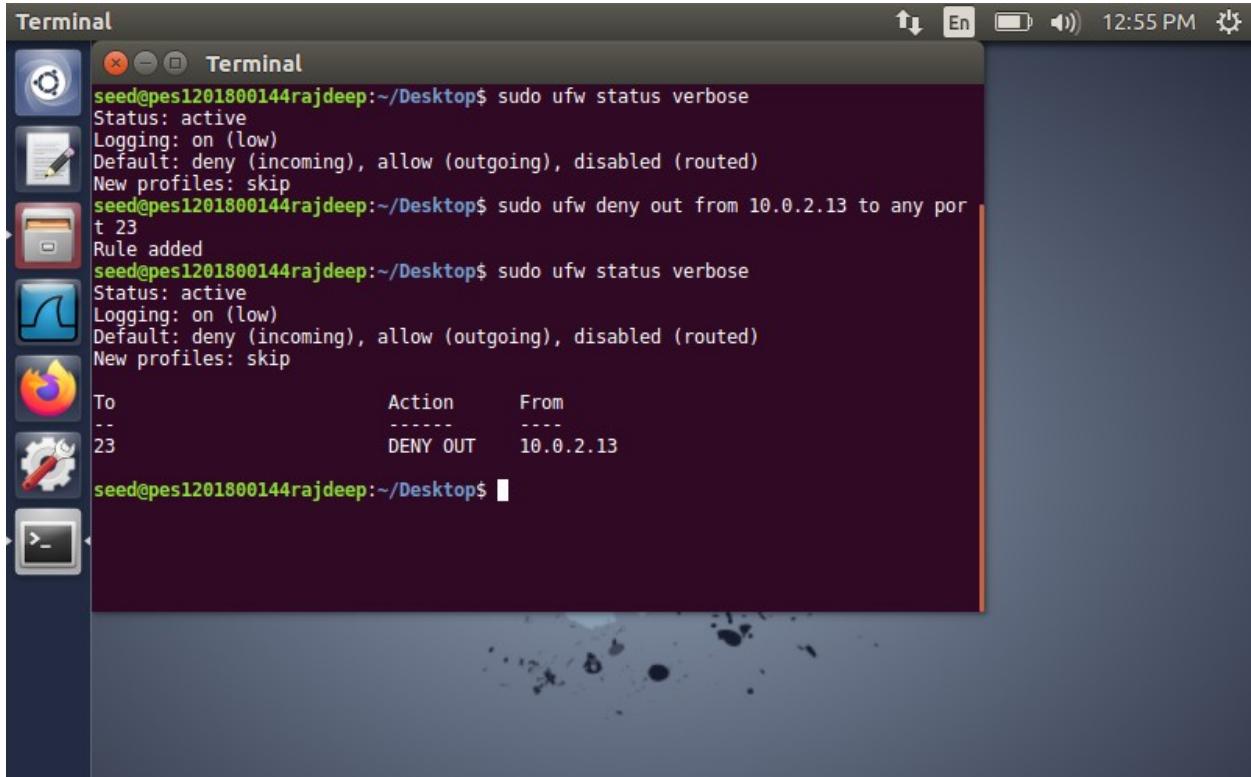


Screenshot 2.11: Above SSH connection from VM2 to VM1 is blocked and the packets are dropped

## OBSERVATIONS:

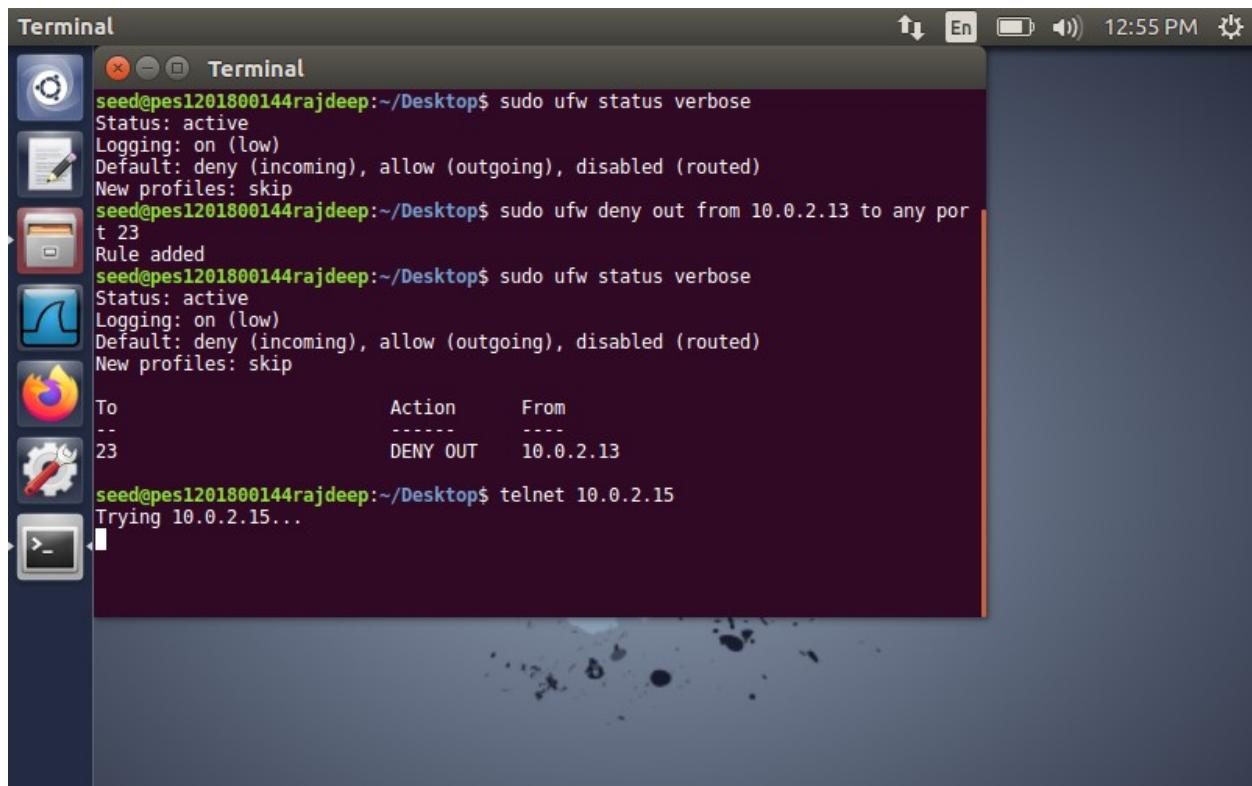
A simple firewall is implemented in this task using netfilter. This is done by loading the code into the kernel using the hooks in netfilter. The packets are being dropped according to the rules written in the lkm file. This is proved by the evident kernel logs shown in the above screenshots showing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

## **TASK 3a:**



```
seed@pes1201800144rajdeep:~/Desktop$ sudo ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip
seed@pes1201800144rajdeep:~/Desktop$ sudo ufw deny out from 10.0.2.13 to any port 23
Rule added
seed@pes1201800144rajdeep:~/Desktop$ sudo ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip
To           Action      From
--           ----      --
23          DENY OUT    10.0.2.13
seed@pes1201800144rajdeep:~/Desktop$
```

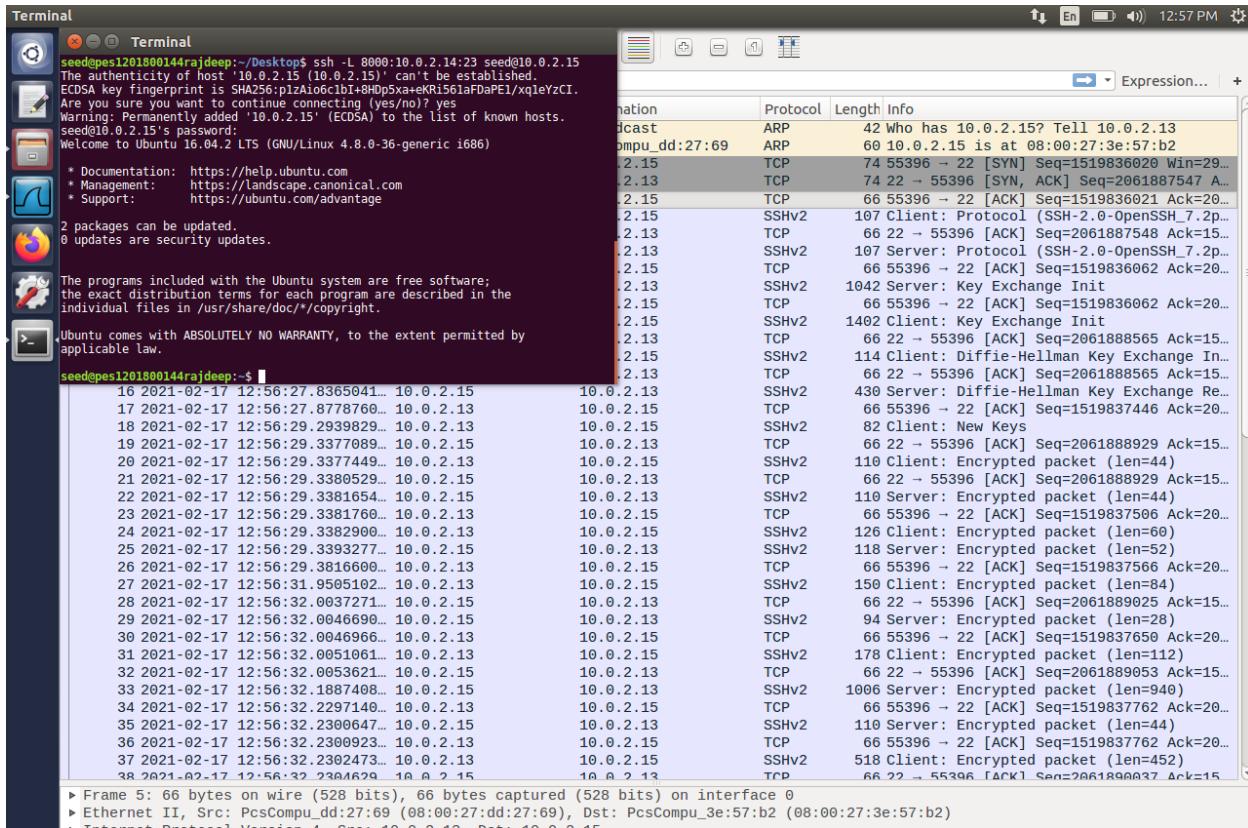
Screenshot 3.a.1: On VM1(10.0.2.13), denying outgoing telnet connections on port 23.

A screenshot of a Linux desktop environment, likely Kali Linux, showing a terminal window titled "Terminal". The terminal window contains the following text:

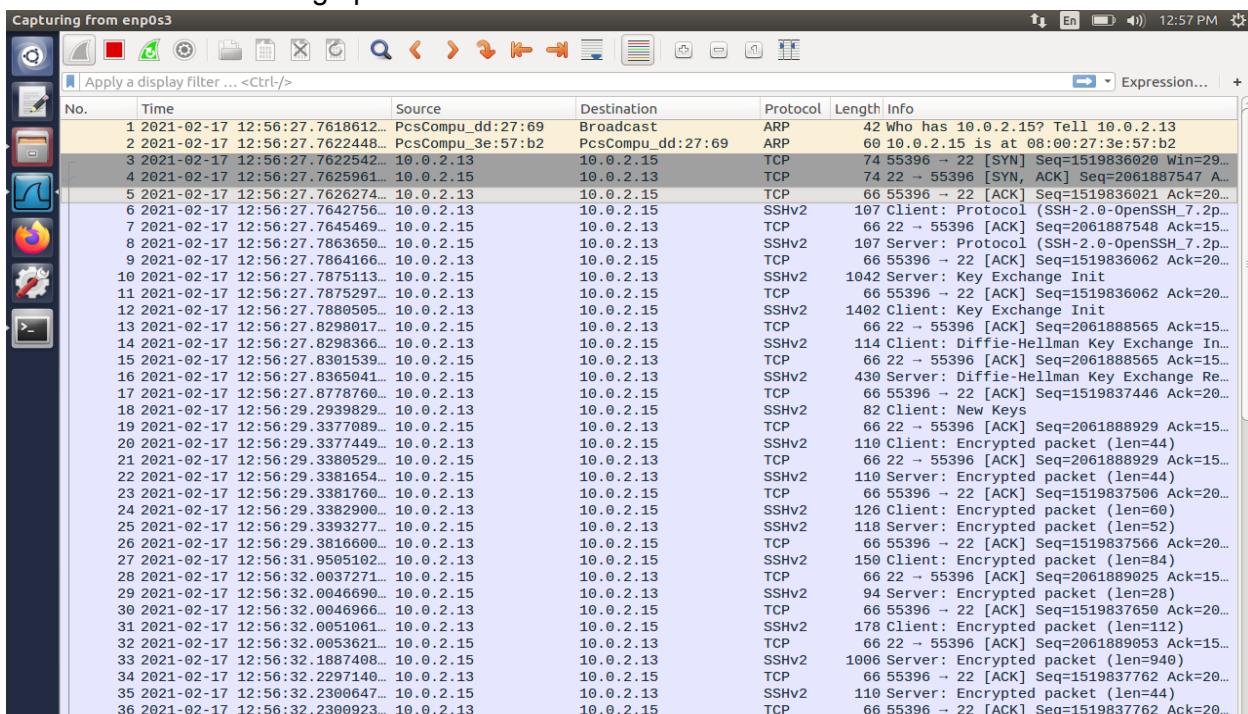
```
seed@pes1201800144rajdeep:~/Desktop$ sudo ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip
seed@pes1201800144rajdeep:~/Desktop$ sudo ufw deny out from 10.0.2.13 to any port 23
Rule added
seed@pes1201800144rajdeep:~/Desktop$ sudo ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip
To          Action      From
--          ----      ---
23          DENY OUT   10.0.2.13
seed@pes1201800144rajdeep:~/Desktop$ telnet 10.0.2.15
Trying 10.0.2.15...
```

The terminal window has a dark background and light-colored text. The desktop environment includes icons for various applications like a file manager, terminal, and browser.

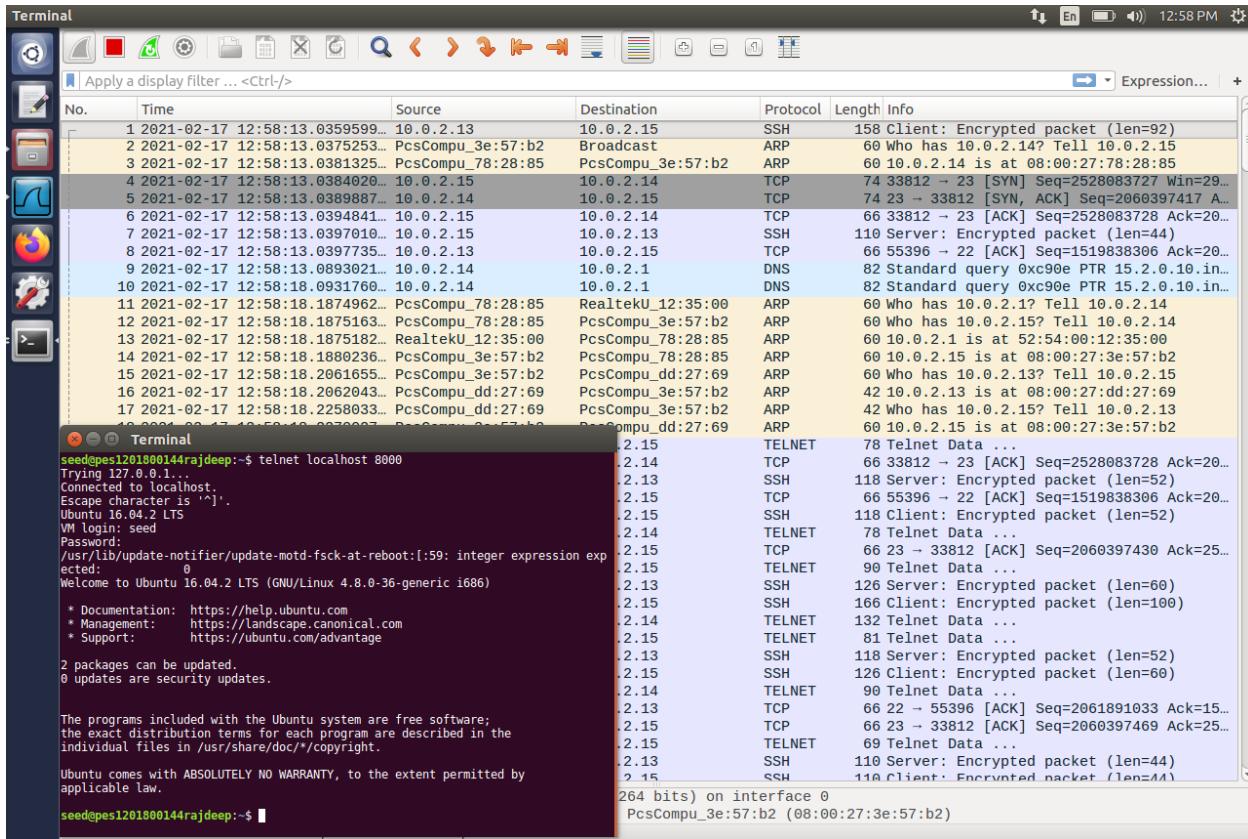
Screenshot 3.a.2: Trying telnet from VM1 to VM3 but unsuccessful due to firewall



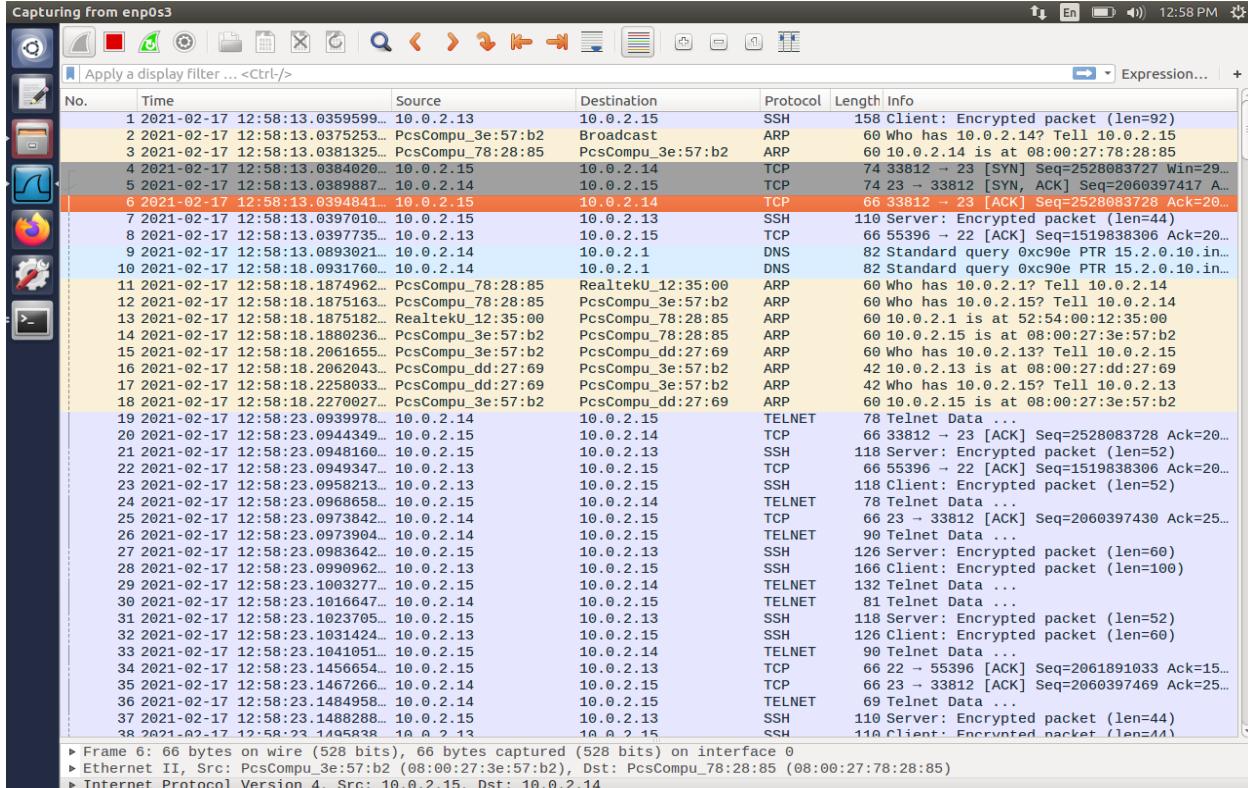
Screenshot 3.a.3: Setting up ssh tunnel from between VM1(10.0.2.13) and VM2(10.0.2.14) to get connection through telnet to VM3(10.0.2.15) via VM2. Also, it is specified that the telnet connection will be through port 8000



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

Capturing from enp0s3

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	2021-02-17 12:58:01.1067621...	fe80::ed69:46d:d15:... ff02::fb	MDNS	180	Standard query 0x0000 PTR _ftp._tcp.lo...	
2	2021-02-17 12:58:01.1068484...	10.0.2.13	224.0.0.251	MDNS	160	Standard query 0x0000 PTR _ftp._tcp.lo...
3	2021-02-17 12:58:13.3166956...	10.0.2.13	10.0.2.15	SSH	158	Client: Encrypted packet (len=92)
4	2021-02-17 12:58:13.3177236...	PcsCompu_3e:57:b2	Broadcast	ARP	60	Who has 10.0.2.14? Tell 10.0.2.15
5	2021-02-17 12:58:13.3178322...	PcsCompu_78:28:85	PcsCompu_3e:57:b2	ARP	42	10.0.2.14 is at 08:00:27:78:28:85
6	2021-02-17 12:58:13.3187020...	10.0.2.15	10.0.2.14	TCP	74	33812 → 23 [SYN] Seq=2528083727 Win=29...
7	2021-02-17 12:58:13.3187535...	10.0.2.14	10.0.2.15	TCP	74	23 → 33812 [SYN, ACK] Seq=2060397417 A...
8	2021-02-17 12:58:13.3196742...	10.0.2.15	10.0.2.14	TCP	66	33812 → 23 [ACK] Seq=2528083728 Ack=20...
9	2021-02-17 12:58:13.3197338...	10.0.2.15	10.0.2.13	SSH	110	Server: Encrypted packet (len=44)
10	2021-02-17 12:58:13.3202999...	10.0.2.13	10.0.2.15	TCP	66	55396 → 22 [ACK] Seq=1519838306 Ack=20...
11	2021-02-17 12:58:13.3692249...	10.0.2.14	10.0.2.1	DNS	82	Standard query 0xc90e PTR 15.2.0.10.in...
12	2021-02-17 12:58:18.3726526...	10.0.2.14	10.0.2.1	DNS	82	Standard query 0xc90e PTR 15.2.0.10.in...
13	2021-02-17 12:58:18.4670725...	PcsCompu_78:28:85	RealtekU_12:35:00	ARP	42	Who has 10.0.2.1? Tell 10.0.2.14
14	2021-02-17 12:58:18.4671507...	PcsCompu_78:28:85	PcsCompu_3e:57:b2	ARP	42	Who has 10.0.2.15? Tell 10.0.2.14
15	2021-02-17 12:58:18.4676142...	RealtekU_12:35:00	PcsCompu_78:28:85	ARP	60	10.0.2.1 is at 52:54:00:12:35:00
16	2021-02-17 12:58:18.4686771...	PcsCompu_3e:57:b2	PcsCompu_78:28:85	ARP	60	10.0.2.15 is at 08:00:27:3e:57:b2
17	2021-02-17 12:58:18.4864189...	PcsCompu_3e:57:b2	PcsCompu_dd:27:69	ARP	60	Who has 10.0.2.13? Tell 10.0.2.15
18	2021-02-17 12:58:18.4868883...	PcsCompu_dd:27:69	PcsCompu_3e:57:b2	ARP	60	10.0.2.13 is at 08:00:27:dd:27:69
19	2021-02-17 12:58:18.5065307...	PcsCompu_dd:27:69	PcsCompu_3e:57:b2	ARP	60	Who has 10.0.2.15? Tell 10.0.2.13
20	2021-02-17 12:58:18.5071927...	PcsCompu_3e:57:b2	PcsCompu_dd:27:69	ARP	60	10.0.2.15 is at 08:00:27:3e:57:b2
21	2021-02-17 12:58:23.3736422...	10.0.2.14	10.0.2.15	TELNET	78	TelNet Data ...
22	2021-02-17 12:58:23.3749864...	10.0.2.14	10.0.2.14	TCP	66	33812 → 23 [ACK] Seq=2528083728 Ack=20...
23	2021-02-17 12:58:23.3750175...	10.0.2.15	10.0.2.13	SSH	118	Server: Encrypted packet (len=52)
24	2021-02-17 12:58:23.3754453...	10.0.2.13	10.0.2.15	TCP	66	55396 → 22 [ACK] Seq=1519838306 Ack=20...
25	2021-02-17 12:58:23.3764130...	10.0.2.13	10.0.2.15	SSH	118	Client: Encrypted packet (len=52)
26	2021-02-17 12:58:23.3771883...	10.0.2.15	10.0.2.14	TELNET	78	TelNet Data ...
27	2021-02-17 12:58:23.3772803...	10.0.2.14	10.0.2.15	TCP	66	23 → 33812 [ACK] Seq=2060397430 Ack=25...
28	2021-02-17 12:58:23.3773969...	10.0.2.14	10.0.2.15	TELNET	90	TelNet Data ...
29	2021-02-17 12:58:23.3784315...	10.0.2.15	10.0.2.13	SSH	126	Server: Encrypted packet (len=60)

Frame 1: 180 bytes on wire (1440 bits), 180 bytes captured (1440 bits) on interface 0  
Ethernet II, Src: PcsCompu\_dd:27:69 (08:00:27:dd:27:69), Dst: IPv6mcast\_fb (33:33:00:00:00:fb)  
Internet Protocol Version 6, Src: fe80::ed69:46d:d15:2ec8, Dst: ff02::fb  
User Datagram Protocol, Src Port: 5353, Dst Port: 5353  
Multicast Domain Name System (query)

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

Capturing from enp0s3

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	2021-02-17 12:58:01.2173810...	fe80::ed69:46d:d15:... ff02::fb	MDNS	180	Standard query 0x0000 PTR _ftp._tcp.lo...	
2	2021-02-17 12:58:01.2174554...	10.0.2.13	224.0.0.251	MDNS	160	Standard query 0x0000 PTR _ftp._tcp.lo...
3	2021-02-17 12:58:13.4273061...	10.0.2.13	10.0.2.15	SSH	158	Client: Encrypted packet (len=92)
4	2021-02-17 12:58:13.4278874...	PcsCompu_3e:57:b2	Broadcast	ARP	42	Who has 10.0.2.14? Tell 10.0.2.15
5	2021-02-17 12:58:13.4289356...	PcsCompu_78:28:85	PcsCompu_3e:57:b2	ARP	60	10.0.2.14 is at 08:00:27:78:28:85
6	2021-02-17 12:58:13.4289578...	10.0.2.15	10.0.2.14	TCP	74	33812 → 23 [SYN] Seq=2528083727 Win=29...
7	2021-02-17 12:58:13.4298476...	10.0.2.14	10.0.2.15	TCP	74	23 → 33812 [SYN, ACK] Seq=2060397417 A...
8	2021-02-17 12:58:13.4299509...	10.0.2.15	10.0.2.14	TCP	66	33812 → 23 [ACK] Seq=2528083728 Ack=20...
9	2021-02-17 12:58:13.4301361...	10.0.2.15	10.0.2.13	SSH	110	Server: Encrypted packet (len=44)
10	2021-02-17 12:58:13.4310481...	10.0.2.13	10.0.2.15	TCP	66	55396 → 22 [ACK] Seq=1519838306 Ack=20...
11	2021-02-17 12:58:13.4801327...	10.0.2.14	10.0.2.1	DNS	82	Standard query 0xc90e PTR 15.2.0.10.in...
12	2021-02-17 12:58:18.4839768...	10.0.2.14	10.0.2.1	DNS	82	Standard query 0xc90e PTR 15.2.0.10.in...
13	2021-02-17 12:58:18.5782912...	PcsCompu_78:28:85	RealtekU_12:35:00	ARP	60	Who has 10.0.2.1? Tell 10.0.2.14
14	2021-02-17 12:58:18.5783106...	PcsCompu_78:28:85	PcsCompu_3e:57:b2	ARP	60	Who has 10.0.2.15? Tell 10.0.2.14
15	2021-02-17 12:58:18.5783433...	PcsCompu_3e:57:b2	PcsCompu_78:28:85	ARP	42	10.0.2.15 is at 08:00:27:3e:57:b2
16	2021-02-17 12:58:18.5784127...	RealtekU_12:35:00	PcsCompu_78:28:85	ARP	60	10.0.2.1 is at 52:54:00:12:35:00
17	2021-02-17 12:58:18.5964885...	PcsCompu_3e:57:b2	PcsCompu_dd:27:69	ARP	42	Who has 10.0.2.13? Tell 10.0.2.15
18	2021-02-17 12:58:18.5976009...	PcsCompu_dd:27:69	PcsCompu_3e:57:b2	ARP	60	10.0.2.13 is at 08:00:27:dd:27:69
19	2021-02-17 12:58:18.6172444...	PcsCompu_dd:27:69	PcsCompu_3e:57:b2	ARP	60	Who has 10.0.2.15? Tell 10.0.2.13
20	2021-02-17 12:58:18.6172860...	PcsCompu_3e:57:b2	PcsCompu_dd:27:69	ARP	42	10.0.2.15 is at 08:00:27:3e:57:b2
21	2021-02-17 12:58:23.4848555...	10.0.2.14	10.0.2.15	TELNET	78	TelNet Data ...
22	2021-02-17 12:58:23.4849638...	10.0.2.15	10.0.2.14	TCP	66	33812 → 23 [ACK] Seq=2528083728 Ack=20...
23	2021-02-17 12:58:23.4853622...	10.0.2.15	10.0.2.13	SSH	118	Server: Encrypted packet (len=52)
24	2021-02-17 12:58:23.4861645...	10.0.2.13	10.0.2.15	TCP	66	55396 → 22 [ACK] Seq=1519838306 Ack=20...
25	2021-02-17 12:58:23.4870810...	10.0.2.13	10.0.2.15	SSH	118	Client: Encrypted packet (len=52)

Frame 1: 180 bytes on wire (1440 bits), 180 bytes captured (1440 bits) on interface 0  
Ethernet II, Src: PcsCompu\_dd:27:69 (08:00:27:dd:27:69), Dst: IPv6mcast\_fb (33:33:00:00:00:fb)  
Internet Protocol Version 6, Src: fe80::ed69:46d:d15:2ec8, Dst: ff02::fb  
User Datagram Protocol, Src Port: 5353, Dst Port: 5353  
Multicast Domain Name System (query)

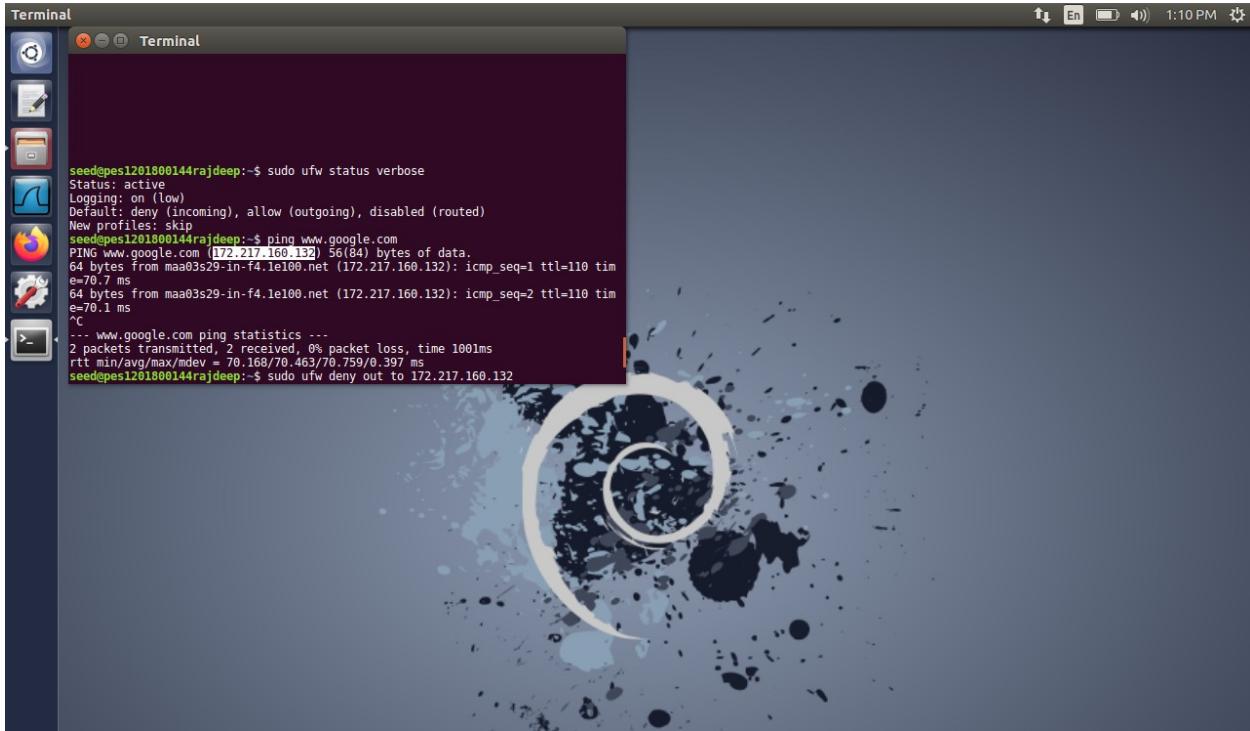
0000	33	33	00	00	00	fb	08	00	27	dd	27	69	86	dd	60	06	33.....'.'i...
0010	7a	d0	00	7e	11	ff	fe	80	00	00	00	00	00	ed	69	z.....'.....i	
0020	04	6d	0d	15	2e	c8	ff	02	00	00	00	00	00	00	00	00	.m.....'.....

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

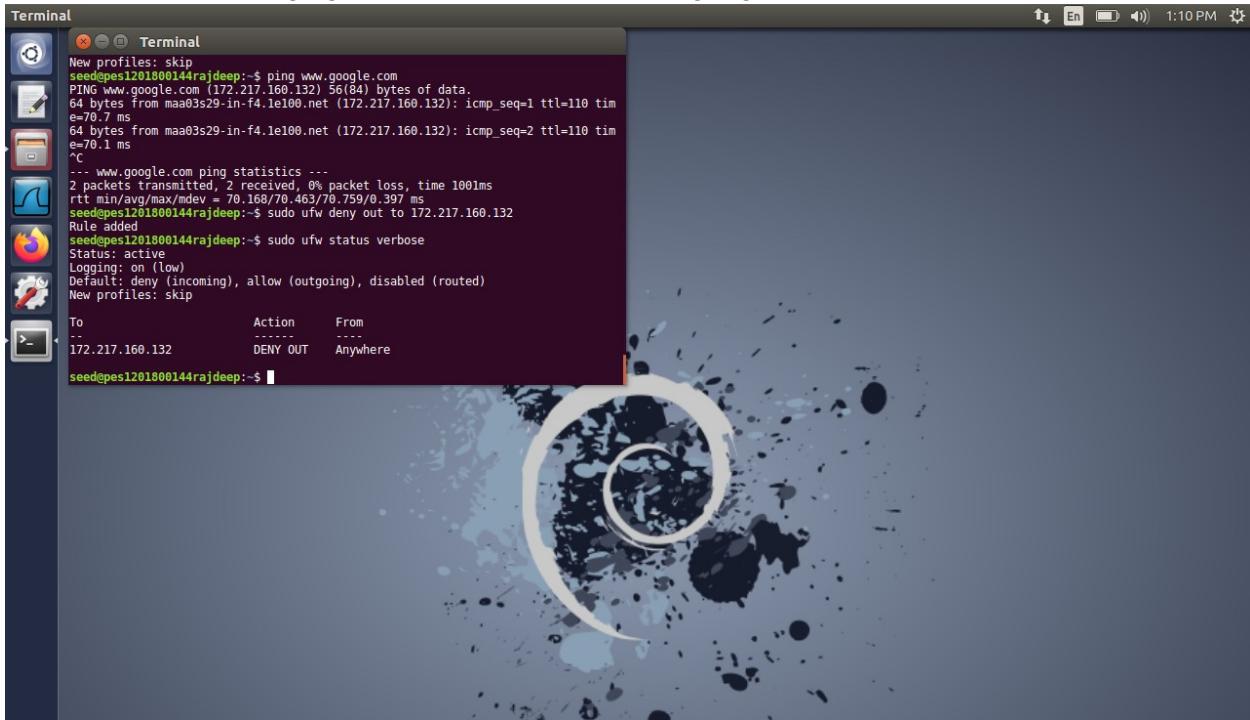
**OBSERVATION:**

**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](http://www.google.com)

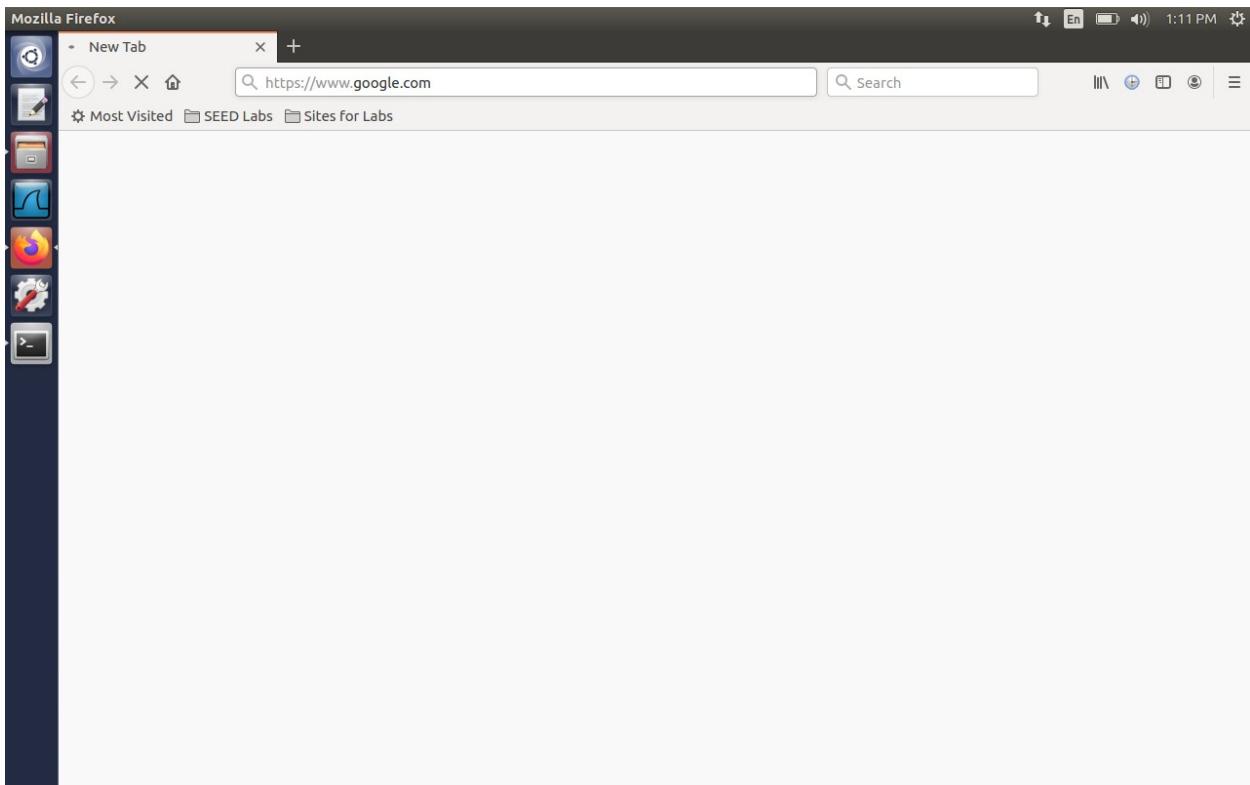


Screenshot 3.b.2: Denying connections to google.com IP address

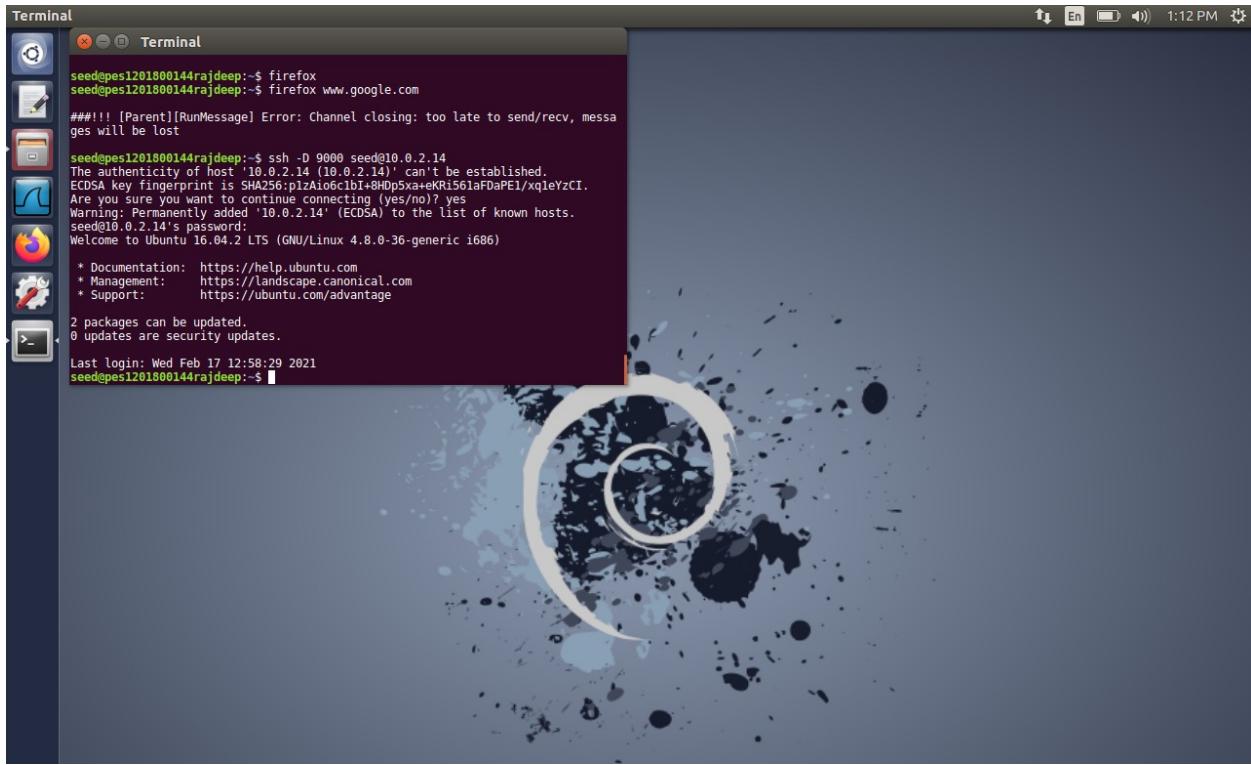
The screenshot shows a Linux desktop environment with a terminal window open. The terminal window title is "Terminal". The session content is as follows:

```
seed@pes1201800144rajddeep:~$ ping www.google.com
PING www.google.com (172.217.160.132) 56(84) bytes of data.
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
^C
--- www.google.com ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1003ms
seed@pes1201800144rajddeep:~$
```

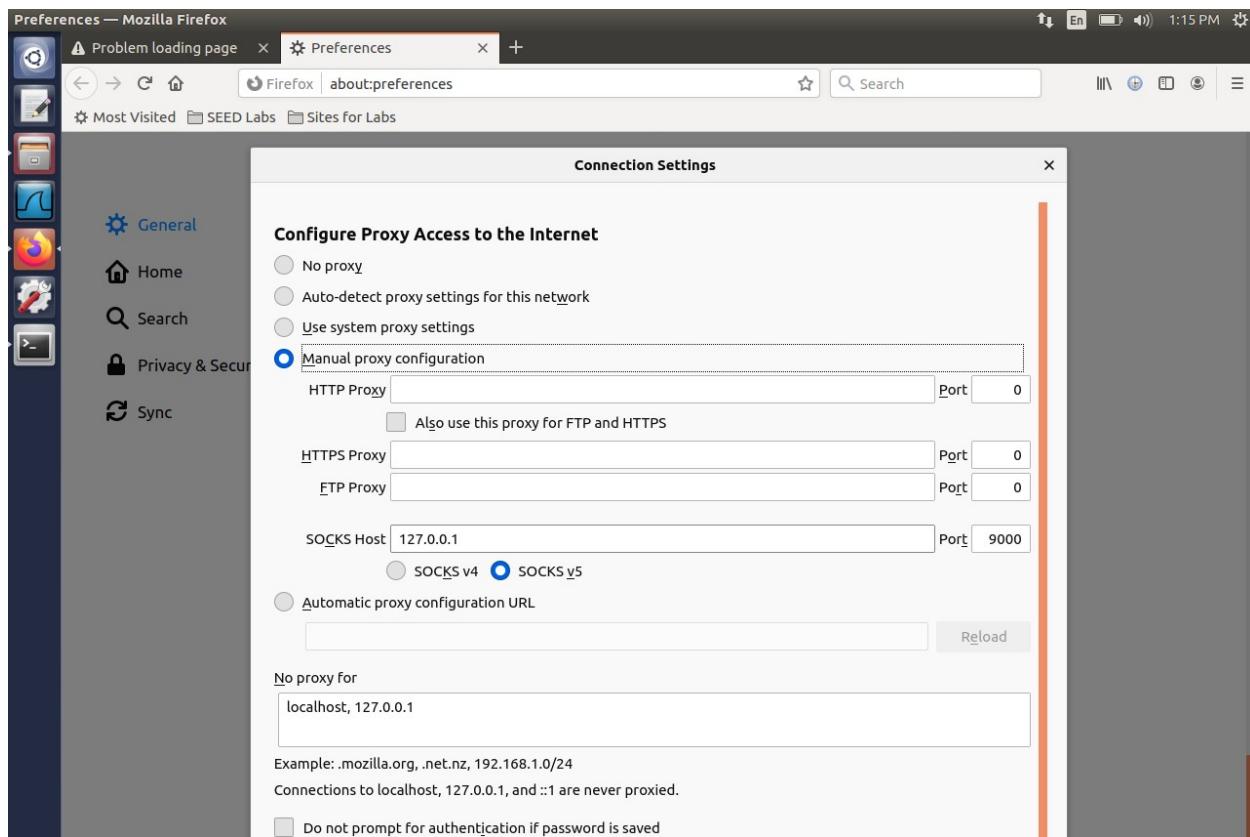
Screenshot 3.b.3: Ping operation blocked because of firewall



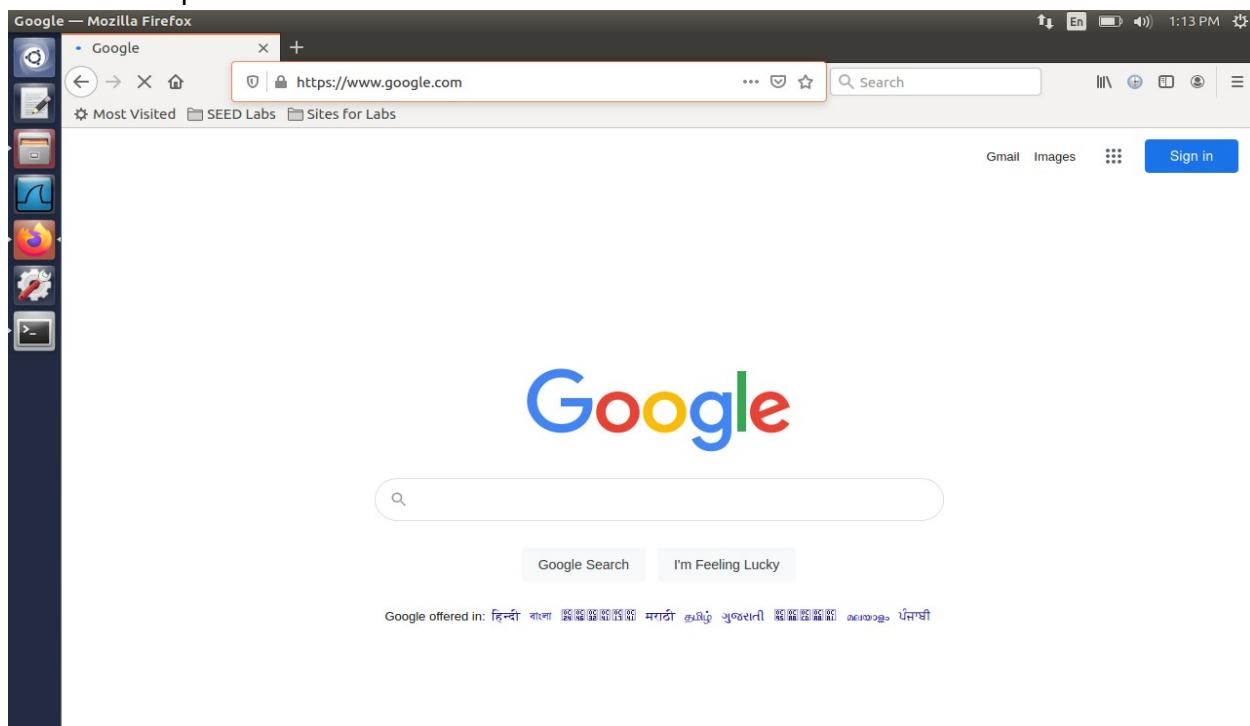
Screenshot 3.b.4: Unable to access google.com on browser too



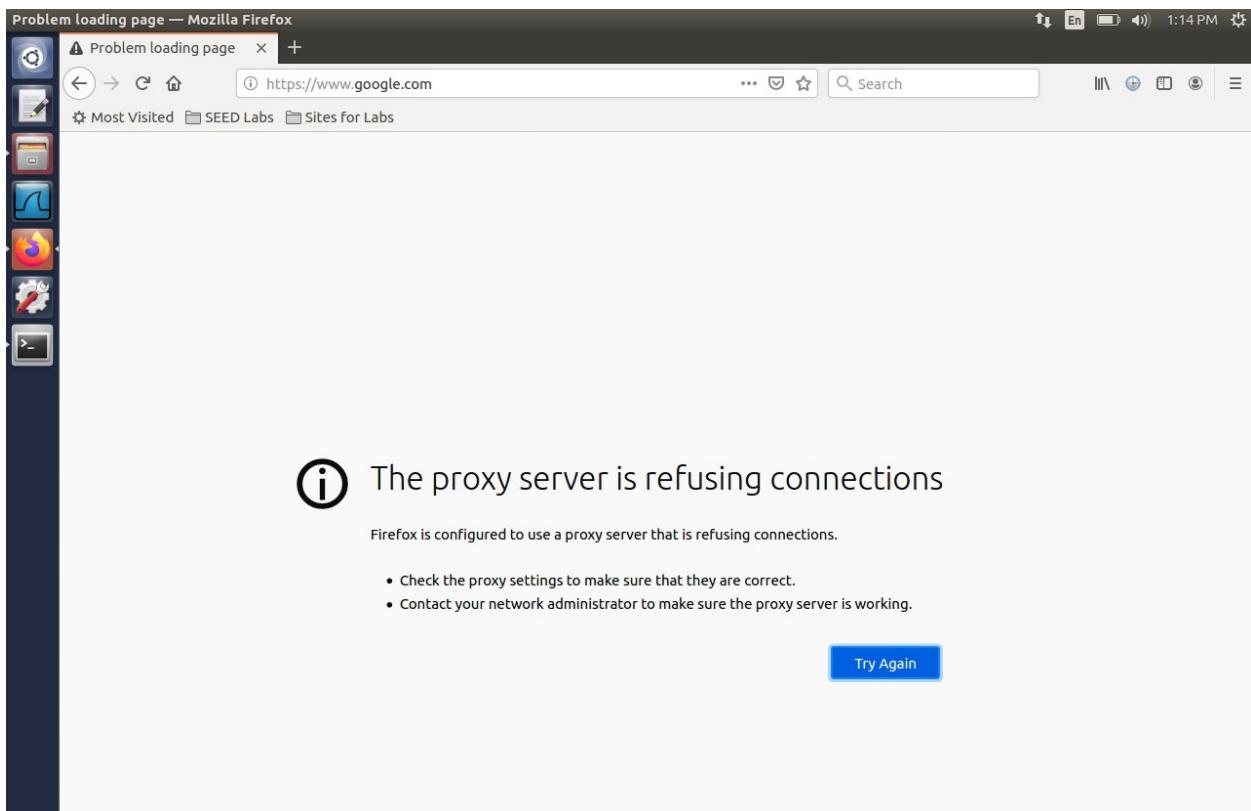
Screenshot 3.b.5: Creating SSH tunnel from VM1(10.0.2.13) to VM2(10.0.2.14) with dynamic port forwarding through port 9000



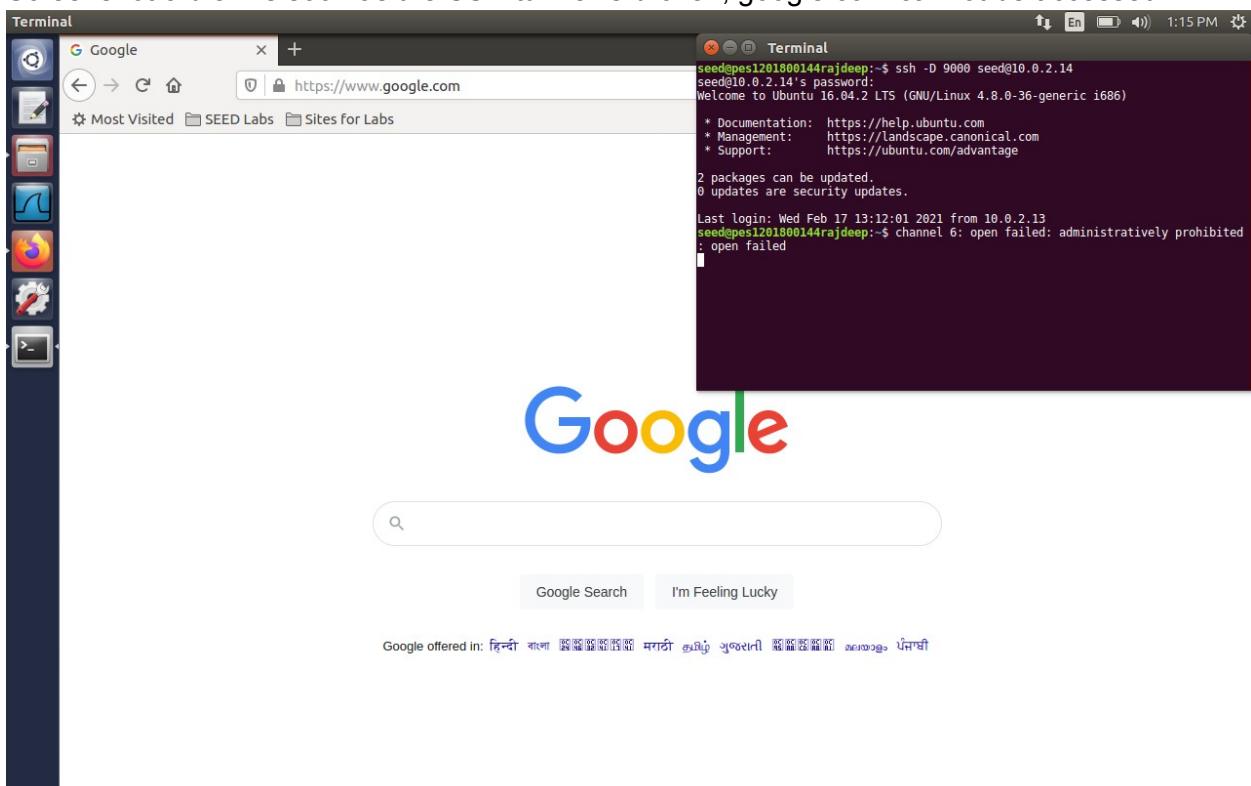
Screenshot 3.b.6: Setting up firefox network connections on VM1, SOCKS host is set to localhost but port is set to 9000



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

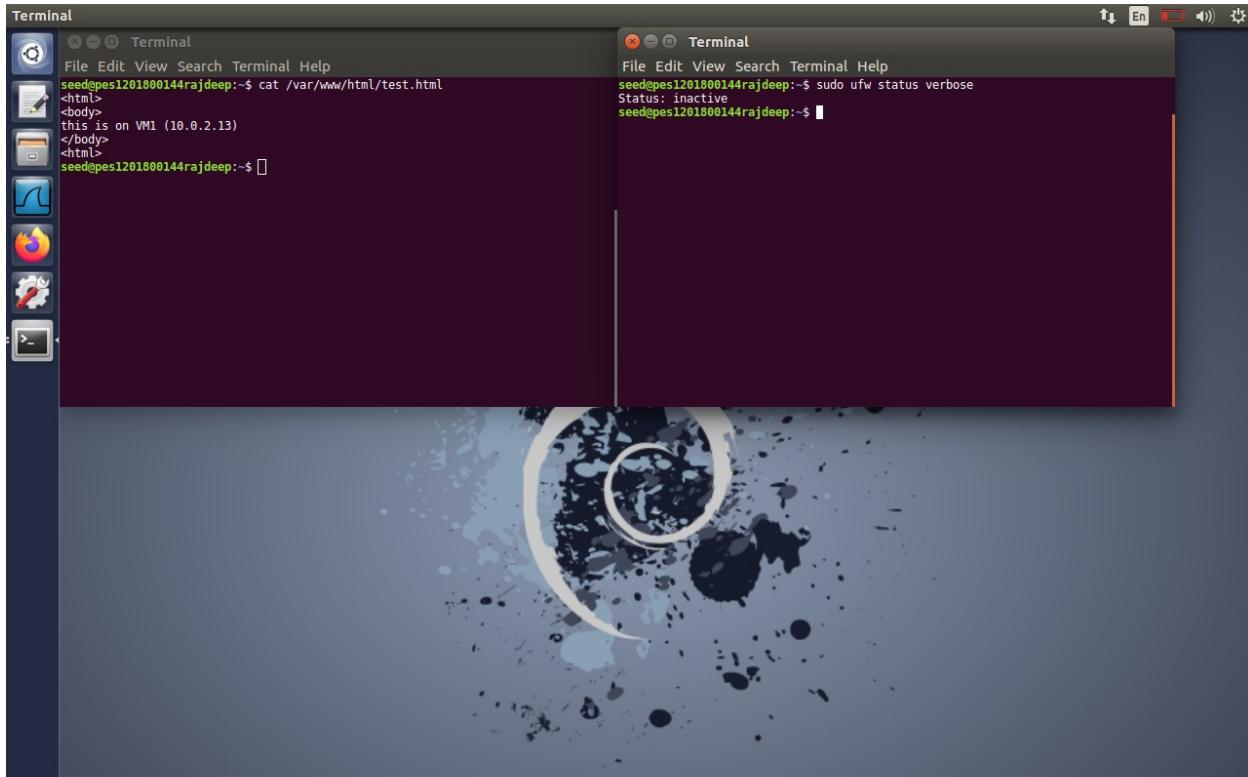


Screenshot 3.b.9: Again when the SSH tunnel is created from VM1 to VM2, google.com can be accessed

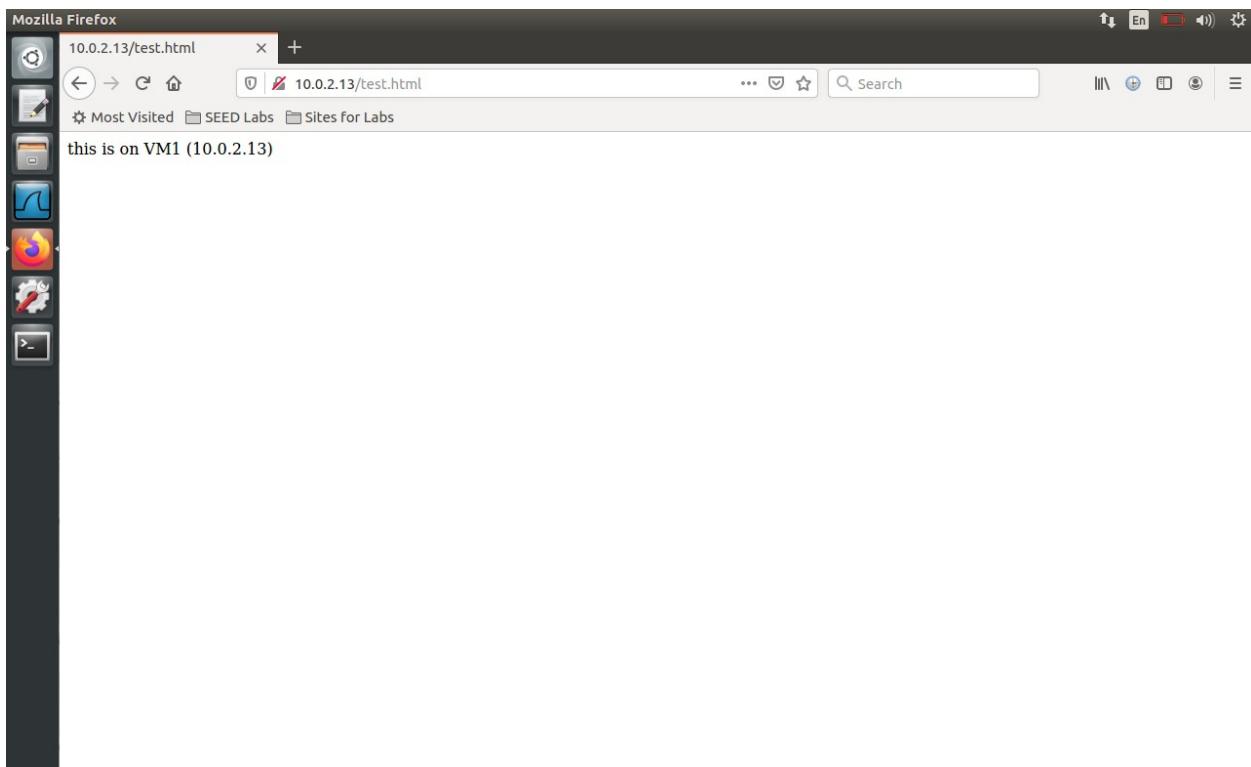
## OBSERVATION:

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.

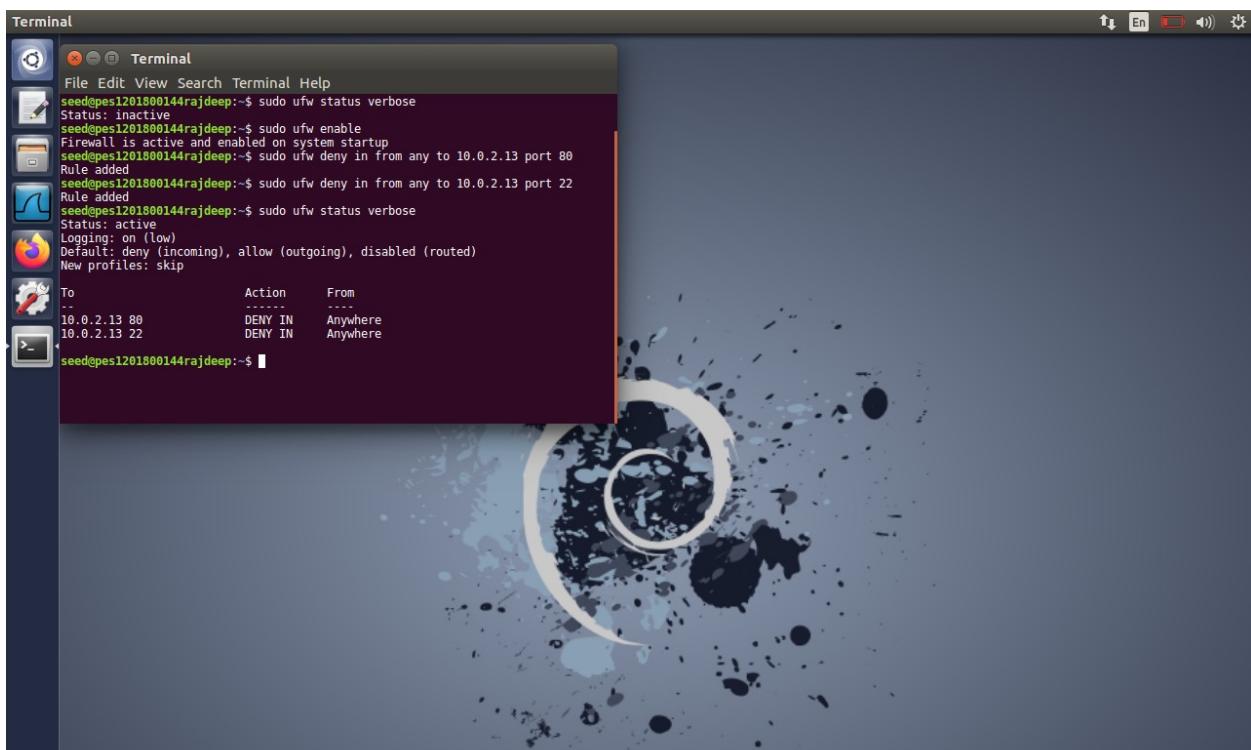
## **TASK 4:**



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

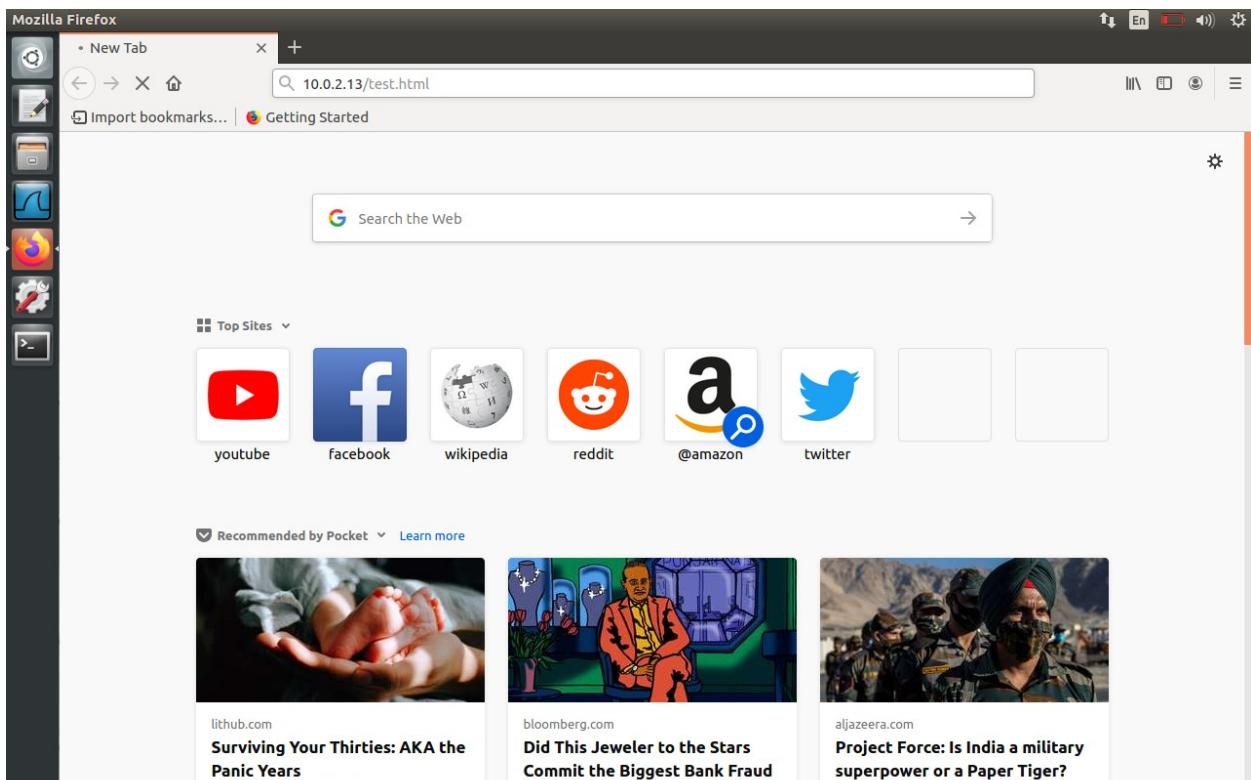


Screenshot 4.2: On VM2(10.0.2.14), the HTML file webpage can be accessed

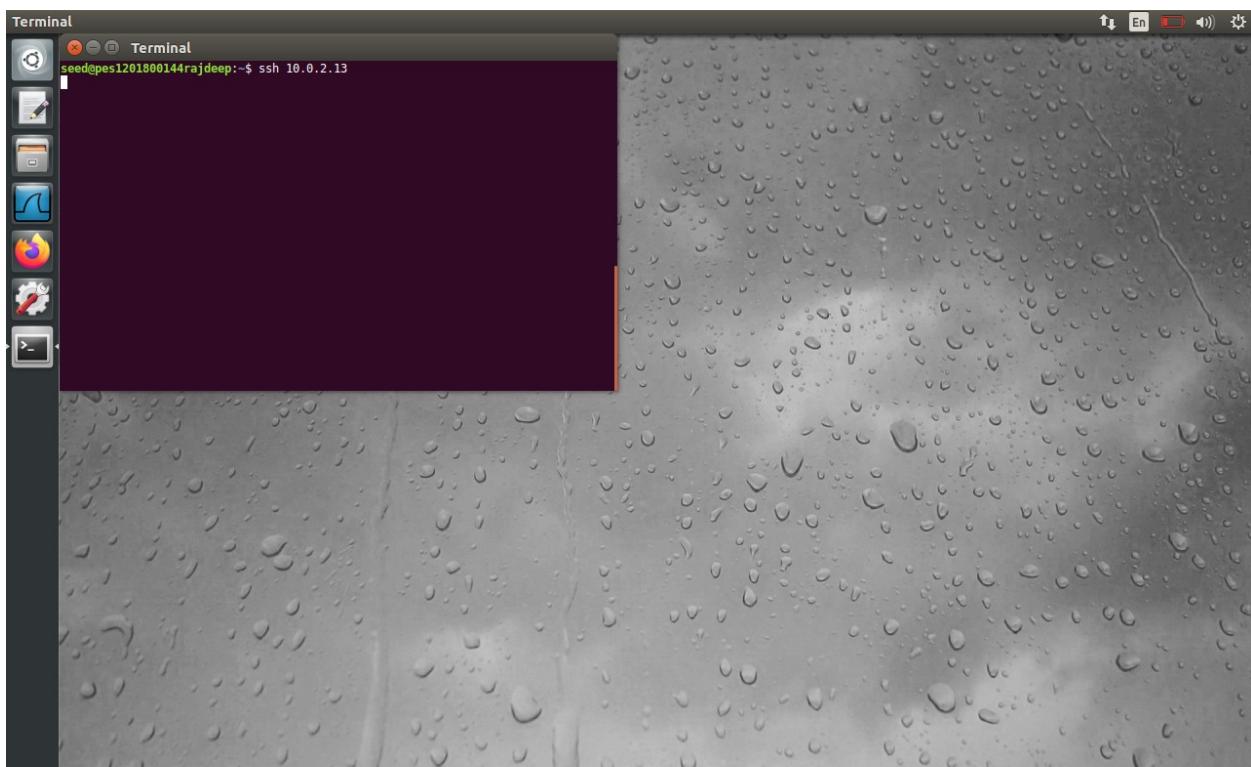


To	Action	From
...	-----	....
10.0.2.13 80	DENY IN	Anywhere
10.0.2.13 22	DENY IN	Anywhere

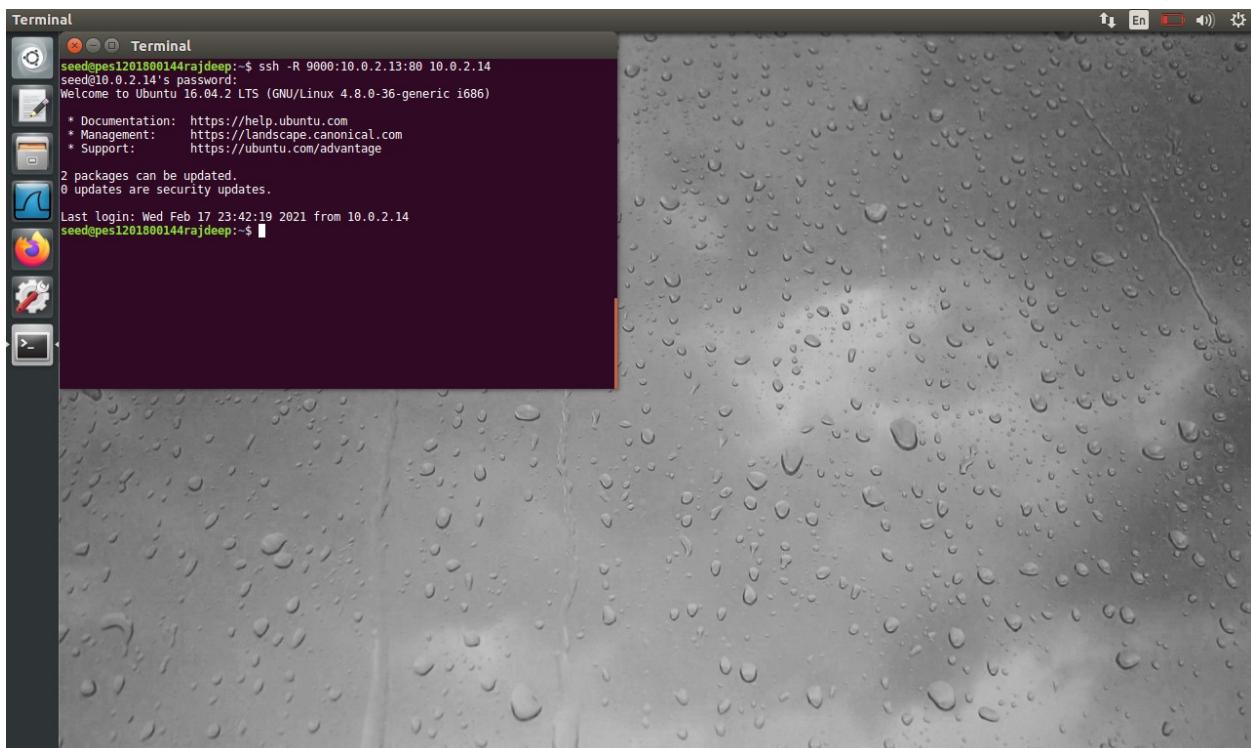
Screenshot 4.3: All connections to VM1(10.0.2.13) are denied through port 22(SSH) and port 80(HTTP) through firewall on VM1



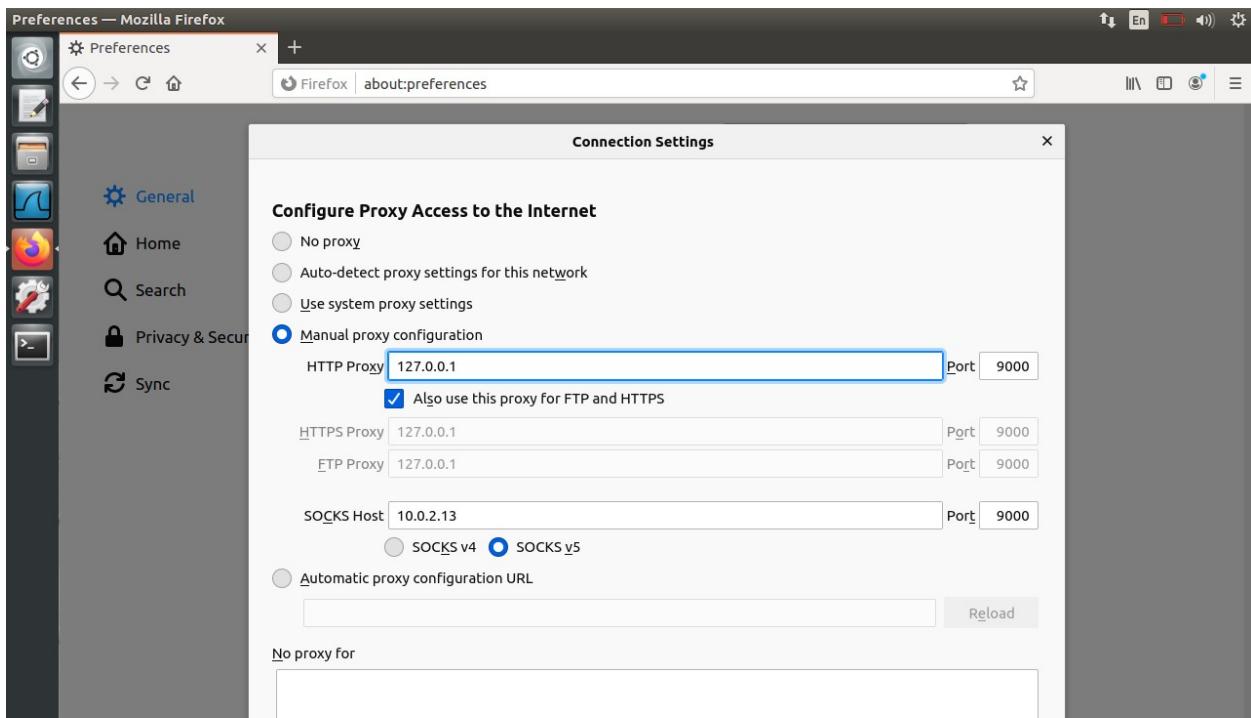
Screenshot 4.4: VM2(10.0.2.14) can no longer access the HTML file on VM1 through HTTP port 80



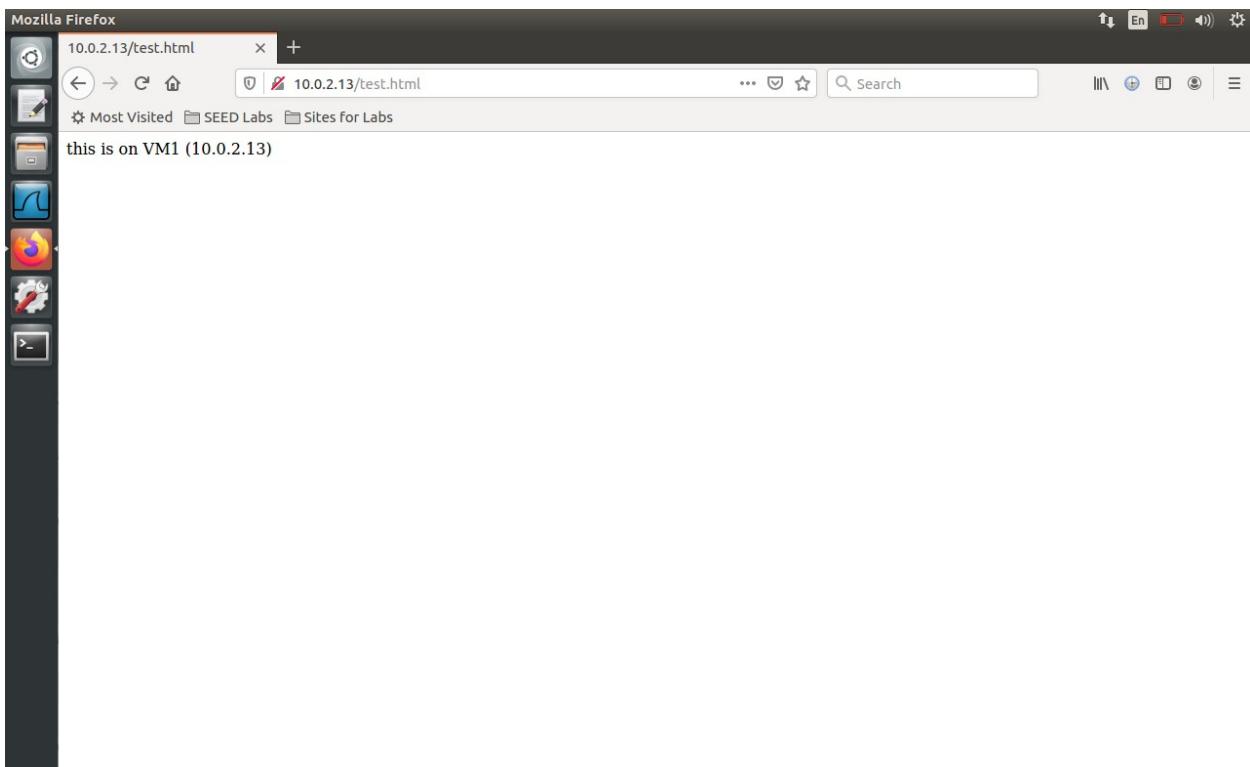
Screenshot 4.5: Also, SSH connection is unsuccessful from VM2 to VM1



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.