BHARATIYA VIDYA BHAVAN'S
SARDAR PATEL INSTITUTE OF TECHNOLOGY
Bhavan's Campus, Munshi Nagar, Andheri (West), Mumbai – 400058-India

DEPARTMENT OF COMPUTER ENGINEERING
SUBJECT: Artificial Intelligence and Machine Learning

Name	Atharva Tamhankar
UID no.	2022300127

Experiment 8

AIM : Setting Up a Personal Firewall Using iptables

STEPS:

Step 1: Check Current iptables Rules

- Command: `sudo iptables -L`
  - Lists (`-L`) all current firewall rules.
  - Helps see existing rules before modifications.

Step 2: Set Default Policies

- Commands:
  - `sudo iptables -P INPUT DROP` — Default: Block all incoming packets.
  - `sudo iptables -P FORWARD DROP` — Default: Block all forwarded packets.
  - `sudo iptables -P OUTPUT ACCEPT` — Default: Allow all outgoing packets.
- Explanation:
  - `-P CHAIN TARGET` sets policy on chain (INPUT, FORWARD, OUTPUT).
  - `DROP` means deny packets; `ACCEPT` means allow packets.

Step 3: Allow Specific Incoming Connections

- Commands:

  -
    ```
    sudo iptables -A INPUT -p tcp --dport 22 -j ACCEPT
    ```

  -
    ```
    sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT
    ```
- Explanation:
  - `-A INPUT`: Append rule to INPUT chain (for incoming packets).
  - `-p tcp`: Match TCP protocol.
  - `--dport 22` or `--dport 80`: Match destination port 22 (SSH) or 80 (HTTP).
  - `-j ACCEPT`: Accept matching packets.

Step 4: Block Traffic from a Specific IP

- Command:

  -
    ```
    sudo iptables -A INPUT -s 192.168.1.100 -j DROP
    ```
- Explanation:
  - `-s 192.168.1.100`: Match packets originating from this IP.
  - `-j DROP`: Drop (block) these packets.

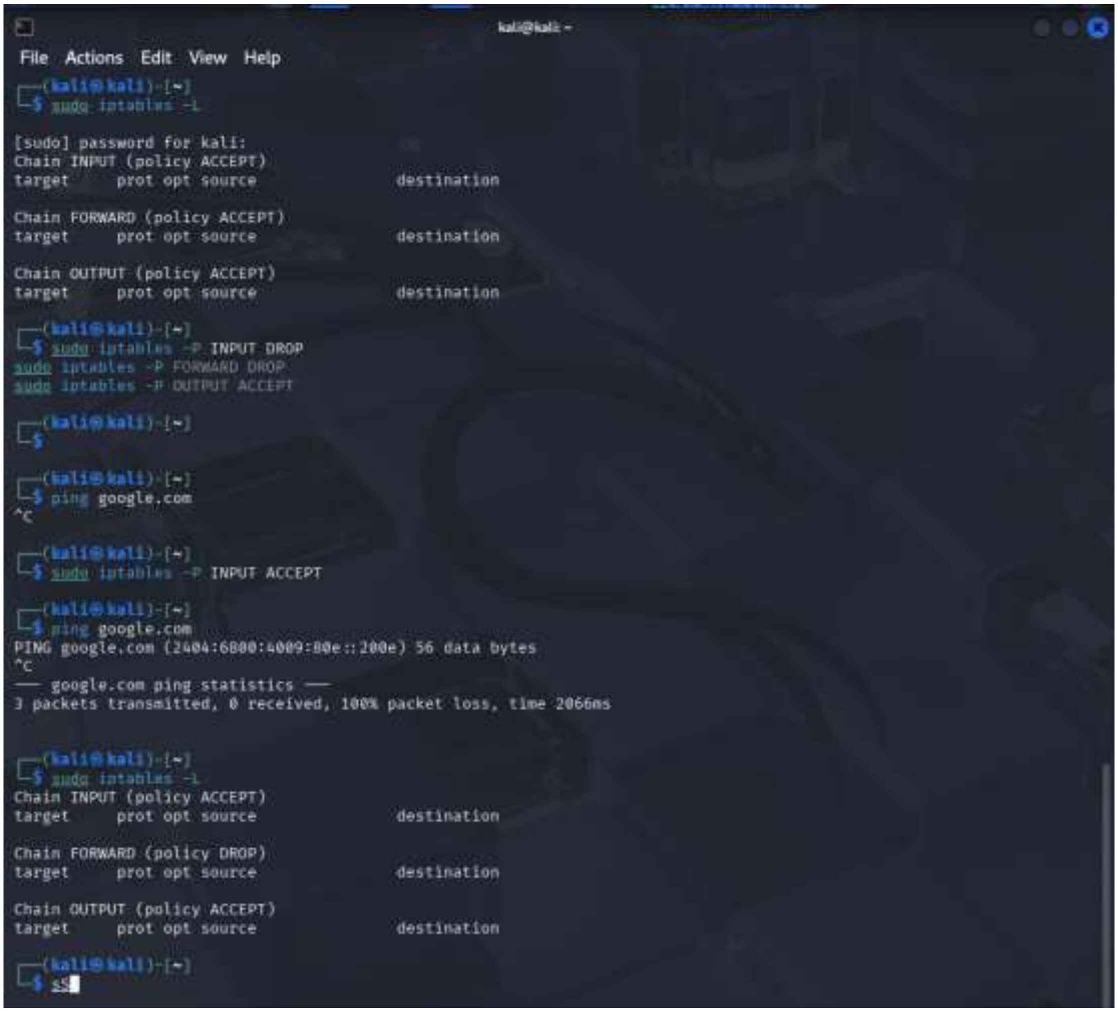
Step 5: Save iptables Rules for Persistence

- Debian/Ubuntu:
  - `sudo iptables-save > /etc/iptables/rules.v4`
  - Saves current rules to a file loaded on boot.
- CentOS/RHEL:
  - `sudo service iptables save`
  - Saves current rules to keep after reboot.

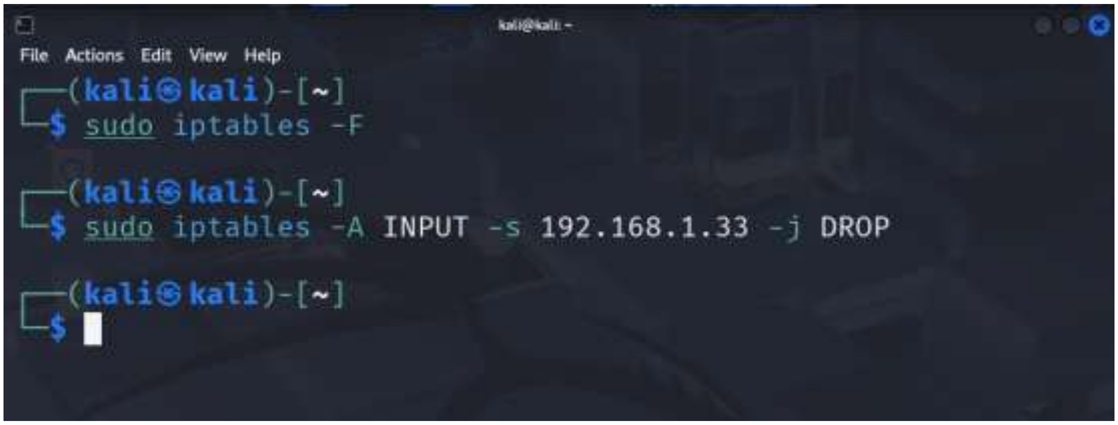
Step 6: Test the Firewall

- Check allowed services:
  - SSH: `ssh user@target_ip` — should connect.
  - HTTP: `curl http://target_ip` — webpage should load.
- Check blocked IP:
  - Ping or connection attempts from `192.168.1.100` should fail.

OUTPUT:

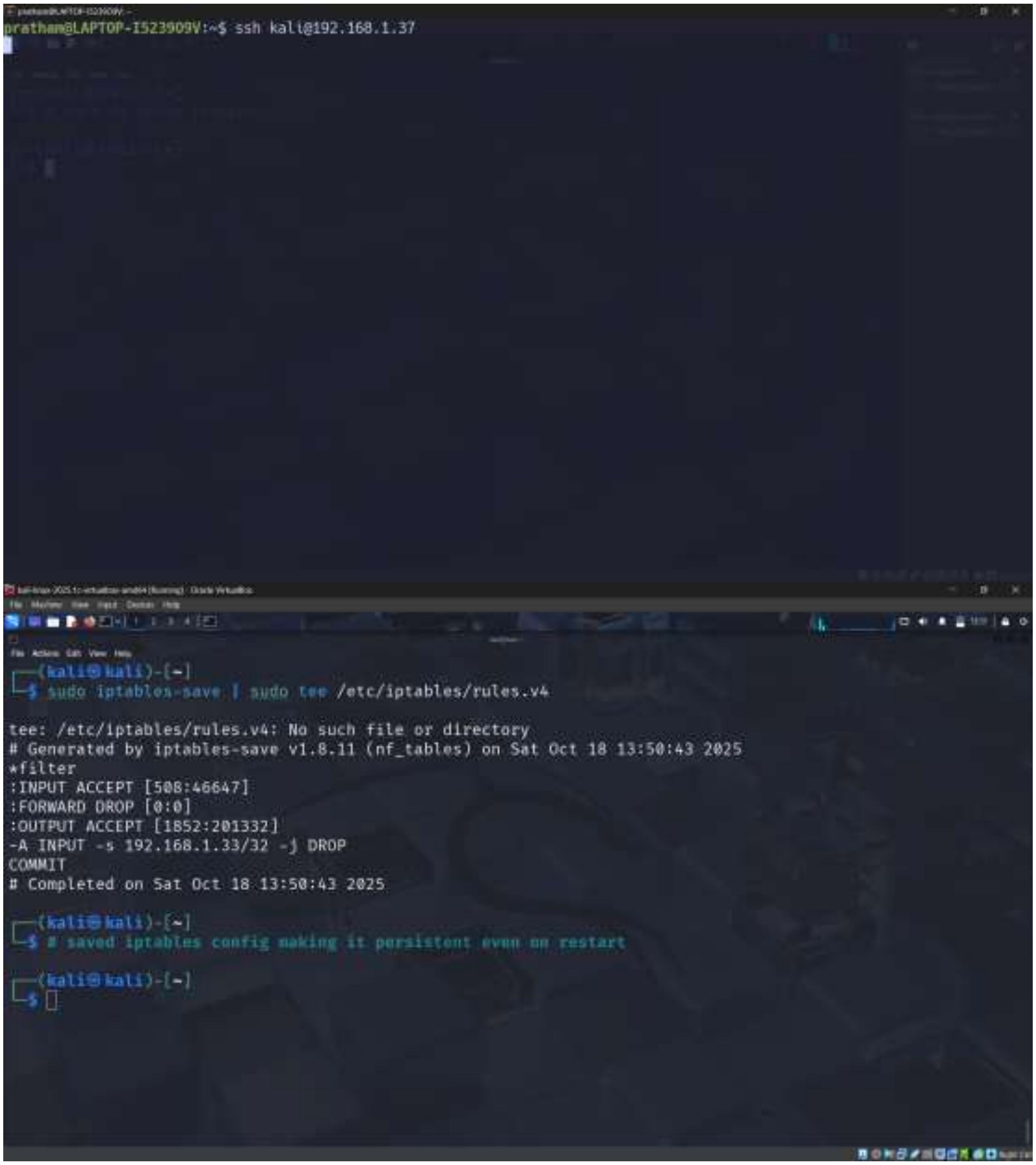


```
┌──(kali㉿kali)-[~]
└─$ sudo iptables -L

[sudo] password for kali:
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

┌──(kali㉿kali)-[~]
└─$ sudo iptables -P INPUT DROP
sudo iptables -P FORWARD DROP
sudo iptables -P OUTPUT ACCEPT

┌──(kali㉿kali)-[~]
└─$

┌──(kali㉿kali)-[~]
└─$ ping google.com
^C

┌──(kali㉿kali)-[~]
└─$ sudo iptables -P INPUT ACCEPT

┌──(kali㉿kali)-[~]
└─$ ping google.com
PING google.com (2404:6800:4009:80e::200e) 56 data bytes
^C
--- google.com ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2066ms


┌──(kali㉿kali)-[~]
└─$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy DROP)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

┌──(kali㉿kali)-[~]
└─$ ss
```



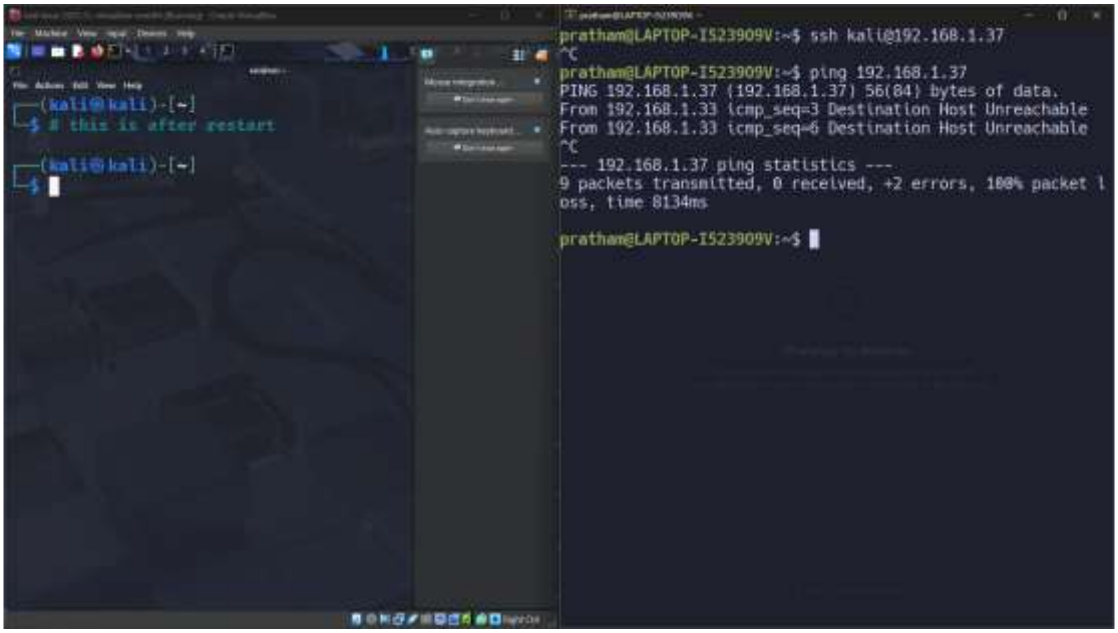
```
┌──(kali㉿kali)-[~]
└─$ sudo iptables -F

┌──(kali㉿kali)-[~]
└─$ sudo iptables -A INPUT -s 192.168.1.33 -j DROP

┌──(kali㉿kali)-[~]
└─$
```

Can’t ssh from another host now

```
pratham@LAPTOP-I523909V:~$ ssh kali@192.168.1.37
```

```
┌──(kali㉿kali)-[~]
└─$ sudo iptables-save | sudo tee /etc/iptables/rules.v4

tee: /etc/iptables/rules.v4: No such file or directory
# Generated by iptables-save v1.8.11 (nf_tables) on Sat Oct 18 13:50:43 2025
*filter
:INPUT ACCEPT [508:46647]
:FORWARD DROP [0:0]
:OUTPUT ACCEPT [1852:201332]
-A INPUT -s 192.168.1.33/32 -j DROP
COMMIT
# Completed on Sat Oct 18 13:50:43 2025

┌──(kali㉿kali)-[~]
└─$ # saved iptables config making it persistent even on restart

┌──(kali㉿kali)-[~]
└─$ 
```

<table>
<tr>
<td></td>
<td>

<pre>
┌──(kali㉿kali)-[~]
└─$ # this is after restart

┌──(kali㉿kali)-[~]
└─$
</pre>
<pre>
pratham@LAPTOP-I523909V:~$ ssh kali@192.168.1.37
^C
pratham@LAPTOP-I523909V:~$ ping 192.168.1.37
PING 192.168.1.37 (192.168.1.37) 56(84) bytes of data.
From 192.168.1.33 icmp_seq=3 Destination Host Unreachable
From 192.168.1.33 icmp_seq=6 Destination Host Unreachable
^C
--- 192.168.1.37 ping statistics ---
9 packets transmitted, 0 received, +2 errors, 100% packet l
oss, time 8134ms

pratham@LAPTOP-I523909V:~$
</pre>
</td>
</tr>
<tr>
<td><b>CONCLUSION:</b></td>
<td>This experiment teaches the fundamentals of securing a Linux system’s network with iptables by filtering traffic through customizable rules. You learn to set default firewall policies that block all incoming connections and allow outgoing traffic by default, creating a secure environment. Then, you build upon this secure foundation by selectively allowing necessary services such as SSH and web traffic while blocking specific unwanted IP addresses. Additionally, you understand how to list, modify, and delete firewall rules, recognizing the importance of rule order. The experiment emphasizes saving firewall configurations so that protection remains intact after system reboots. Overall, it provides essential knowledge and practical skills for managing network security effectively using iptables.</td>
</tr>
</table>