PARSHVANATH CHARITABLE TRUST'S
A. P. SHAH INSTITUTE OF TECHNOLOGY
Department of Information Technology
(NBA Accredited)

**Academic Year: 2025-26**
**Semester: V**
**Class / Branch: TE IT**
**Subject: Security Lab**
**Name of Instructor: Prof. Vishal Badgujar**

**Name of Student: Suyash Y Mane**
**Student ID: 24204008**
**Date of Performance: 30/07/2025**
**Date of Submission: 30/07/2025**

---

## Experiment No. 3

## Aim: To study installation and configuration of Linux Kernel firewall iptables.

## Code Screen Shots:

To list the current rules that are configured for iptables:

```
apsit@apsit-HP-280-Pro-G6-Microtower-PC:~$ sudo iptables -L
[sudo] password for apsit:
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy DROP)
target     prot opt source               destination
DOCKER-USER  all  --  anywhere             anywhere
DOCKER-ISOLATION-STAGE-1  all  --  anywhere             anywhere
ACCEPT     all  --  anywhere             anywhere             ctstate RELATED,ESTABLISHED
DOCKER     all  --  anywhere             anywhere
ACCEPT     all  --  anywhere             anywhere
ACCEPT     all  --  anywhere             anywhere

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain DOCKER (1 references)
target     prot opt source               destination

Chain DOCKER-ISOLATION-STAGE-1 (1 references)
target     prot opt source               destination
DOCKER-ISOLATION-STAGE-2  all  --  anywhere             anywhere
RETURN     all  --  anywhere             anywhere

Chain DOCKER-ISOLATION-STAGE-2 (1 references)
target     prot opt source               destination
DROP       all  --  anywhere             anywhere
RETURN     all  --  anywhere             anywhere

Chain DOCKER-USER (1 references)
target     prot opt source               destination
RETURN     all  --  anywhere             anywhere
```

The command allows incoming traffic for established and related connections by using the conntrack module to track connection states.

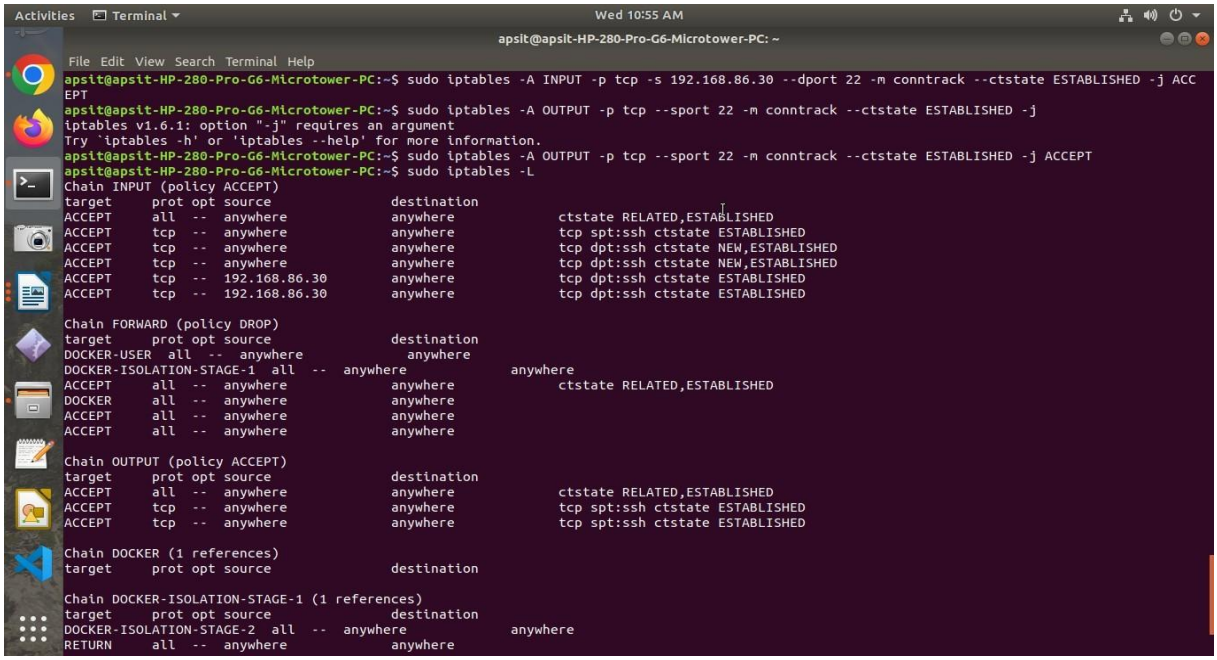
```
apsit@apsit-HP-280-Pro-G6-Microtower-PC:~$ sudo iptables -A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
apsit@apsit-HP-280-Pro-G6-Microtower-PC:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere             ctstate RELATED,ESTABLISHED

Chain FORWARD (policy DROP)
target     prot opt source               destination
DOCKER-USER  all  --  anywhere             anywhere
DOCKER-ISOLATION-STAGE-1  all  --  anywhere             anywhere
ACCEPT     all  --  anywhere             anywhere             ctstate RELATED,ESTABLISHED
DOCKER     all  --  anywhere             anywhere
ACCEPT     all  --  anywhere             anywhere
ACCEPT     all  --  anywhere             anywhere

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain DOCKER (1 references)
target     prot opt source               destination

Chain DOCKER-ISOLATION-STAGE-1 (1 references)
target     prot opt source               destination
DOCKER-ISOLATION-STAGE-2  all  --  anywhere             anywhere
RETURN     all  --  anywhere             anywhere

Chain DOCKER-ISOLATION-STAGE-2 (1 references)
target     prot opt source               destination
DROP       all  --  anywhere             anywhere
RETURN     all  --  anywhere             anywhere

Chain DOCKER-USER (1 references)
target     prot opt source               destination
RETURN     all  --  anywhere             anywhere
```

```
apsit@apsit-HP-280-Pro-G6-Microtower-PC:~$ sudo iptables -A OUTPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
apsit@apsit-HP-280-Pro-G6-Microtower-PC:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere             ctstate RELATED,ESTABLISHED

Chain FORWARD (policy DROP)
target     prot opt source               destination
DOCKER-USER  all  --  anywhere             anywhere
DOCKER-ISOLATION-STAGE-1  all  --  anywhere             anywhere
ACCEPT     all  --  anywhere             anywhere             ctstate RELATED,ESTABLISHED
DOCKER     all  --  anywhere             anywhere
ACCEPT     all  --  anywhere             anywhere
ACCEPT     all  --  anywhere             anywhere

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere             ctstate RELATED,ESTABLISHED

Chain DOCKER (1 references)
target     prot opt source               destination

Chain DOCKER-ISOLATION-STAGE-1 (1 references)
target     prot opt source               destination
DOCKER-ISOLATION-STAGE-2  all  --  anywhere             anywhere
RETURN     all  --  anywhere             anywhere

Chain DOCKER-ISOLATION-STAGE-2 (1 references)
target     prot opt source               destination
DROP       all  --  anywhere             anywhere
RETURN     all  --  anywhere             anywhere

Chain DOCKER-USER (1 references)
target     prot opt source               destination
RETURN     all  --  anywhere             anywhere
```

The first command allows incoming SSH (TCP port 22) traffic for new and established connections, while the second allows outgoing traffic for established SSH connections.

```
apsit@apsit-HP-280-Pro-G6-Microtower-PC:~$ sudo iptables -A INPUT -p tcp --dport 22 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT
apsit@apsit-HP-280-Pro-G6-Microtower-PC:~$ sudo iptables -A OUTPUT -p tcp --sport 22 -m conntrack --ctstate ESTABLISHED -j ACCEPT
apsit@apsit-HP-280-Pro-G6-Microtower-PC:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere             ctstate RELATED,ESTABLISHED
ACCEPT     tcp  --  anywhere             anywhere             tcp spt:ssh ctstate ESTABLISHED
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh ctstate NEW,ESTABLISHED
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh ctstate NEW,ESTABLISHED

Chain FORWARD (policy DROP)
target     prot opt source               destination
DOCKER-USER  all  --  anywhere             anywhere
DOCKER-ISOLATION-STAGE-1  all  --  anywhere             anywhere
ACCEPT     all  --  anywhere             anywhere             ctstate RELATED,ESTABLISHED
DOCKER     all  --  anywhere             anywhere
ACCEPT     all  --  anywhere             anywhere
ACCEPT     all  --  anywhere             anywhere

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere             ctstate RELATED,ESTABLISHED
ACCEPT     tcp  --  anywhere             anywhere             tcp spt:ssh ctstate ESTABLISHED

Chain DOCKER (1 references)
target     prot opt source               destination

Chain DOCKER-ISOLATION-STAGE-1 (1 references)
target     prot opt source               destination
DOCKER-ISOLATION-STAGE-2  all  --  anywhere             anywhere
RETURN     all  --  anywhere             anywhere

Chain DOCKER-ISOLATION-STAGE-2 (1 references)
target     prot opt source               destination
DROP       all  --  anywhere             anywhere
RETURN     all  --  anywhere             anywhere

Chain DOCKER-USER (1 references)
```

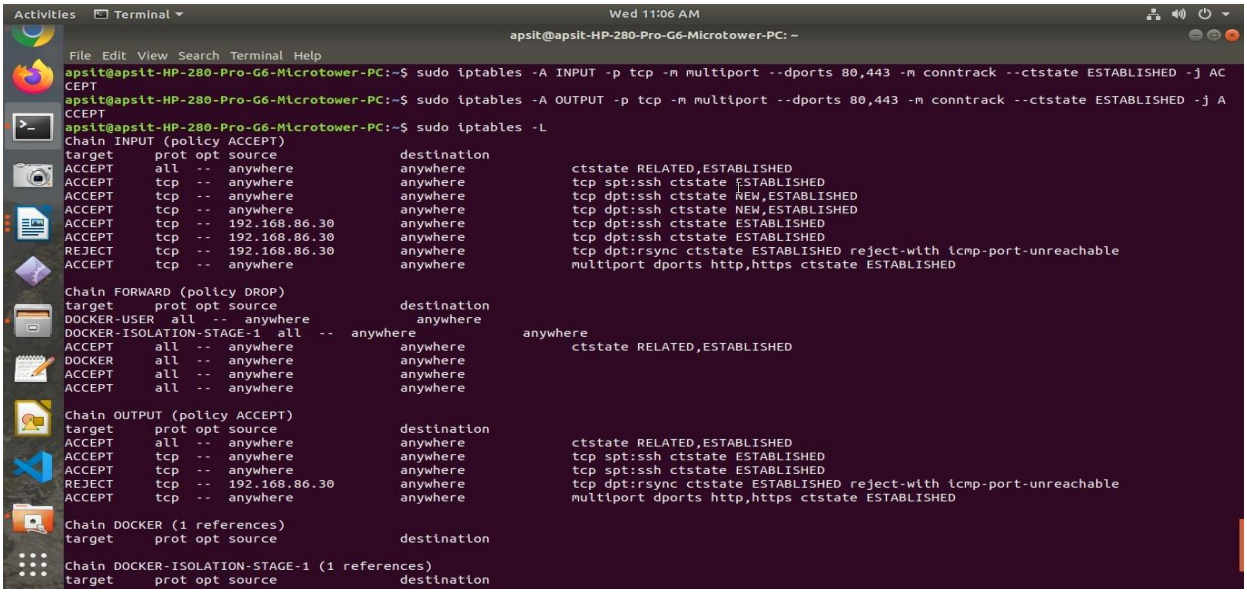
The first command allows incoming SSH (TCP port 22) traffic from the IP range 192.168.86.30 for new and established connections, while the second allows outgoing traffic for established SSH connections.



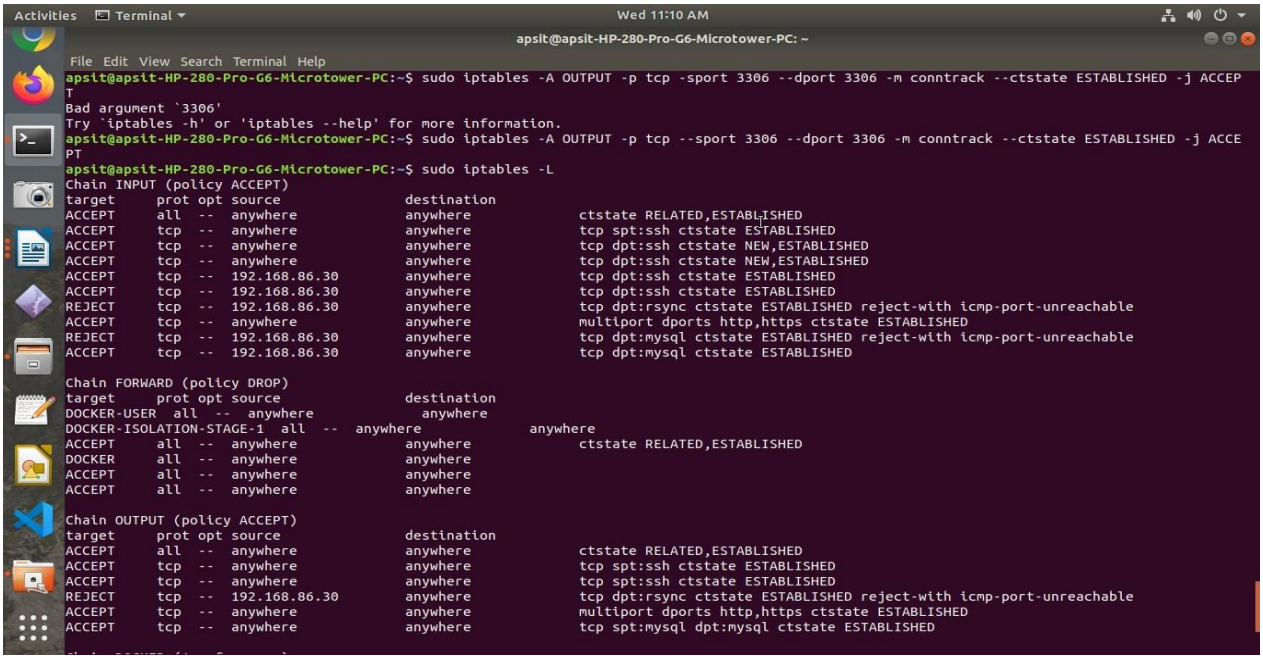
This iptables command adds a rule to reject outgoing TCP traffic from IP 192.168.86.30 to destination port 873 (typically used by rsync) if the connection is already ESTABLISHED. It uses conntrack module to match existing connections and applies the REJECT action instead of just dropping the packet.

```
apsit@apsit-HP-280-Pro-G6-Microtower-PC:~$ sudo iptables -A OUTPUT -p tcp -s 192.168.86.30 --dport 873 -m conntrack --ctstate ESTABLISHED -j R
EJECT
apsit@apsit-HP-280-Pro-G6-Microtower-PC:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere             ctstate RELATED,ESTABLISHED
ACCEPT     tcp  --  anywhere             anywhere             tcp spt:ssh ctstate ESTABLISHED
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh ctstate NEW,ESTABLISHED
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh ctstate NEW,ESTABLISHED
ACCEPT     tcp  --  192.168.86.30        anywhere             tcp dpt:ssh ctstate ESTABLISHED
ACCEPT     tcp  --  192.168.86.30        anywhere             tcp dpt:ssh ctstate ESTABLISHED
REJECT     tcp  --  192.168.86.30        anywhere             tcp dpt:rsync ctstate ESTABLISHED reject-with icmp-port-unreachable

Chain FORWARD (policy DROP)
target     prot opt source               destination
DOCKER-USER  all  --  anywhere             anywhere
DOCKER-ISOLATION-STAGE-1  all  --  anywhere             anywhere
ACCEPT     all  --  anywhere             anywhere             ctstate RELATED,ESTABLISHED
DOCKER     all  --  anywhere             anywhere
ACCEPT     all  --  anywhere             anywhere
ACCEPT     all  --  anywhere             anywhere

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere             ctstate RELATED,ESTABLISHED
ACCEPT     tcp  --  anywhere             anywhere             tcp spt:ssh ctstate ESTABLISHED
ACCEPT     tcp  --  anywhere             anywhere             tcp spt:ssh ctstate ESTABLISHED
REJECT     tcp  --  192.168.86.30        anywhere             tcp dpt:rsync ctstate ESTABLISHED reject-with icmp-port-unreachable

Chain DOCKER (1 references)
target     prot opt source               destination

Chain DOCKER-ISOLATION-STAGE-1 (1 references)
target     prot opt source               destination
DOCKER-ISOLATION-STAGE-2  all  --  anywhere             anywhere
RETURN     all  --  anywhere             anywhere

Chain DOCKER-ISOLATION-STAGE-2 (1 references)
```

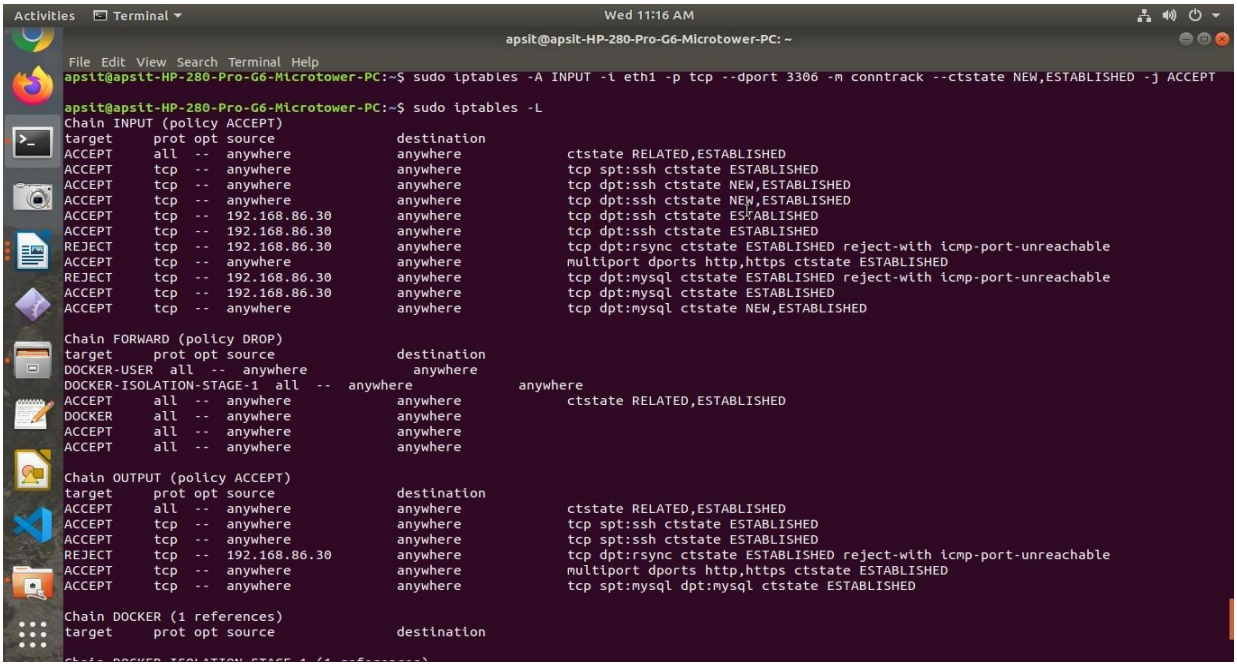
These iptables rules allow incoming and outgoing HTTP (port 80) and HTTPS (port 443) traffic only if the connection is already established, ensuring secure web communication continuity.
It uses multiport to specify multiple destination ports and conntrack to match connections in the ESTABLISHED state.



This iptables rule allows outgoing MySQL traffic (port 3306) if the connection is already ESTABLISHED, ensuring database response packets are allowed.
It uses --sport and --dport to filter traffic on MySQL port and tracks connections using conntrack.



The rule sudo iptables -A INPUT -i eth1 -p tcp --dport 3306 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT allows incoming MySQL traffic on port 3306 from interface eth1 for new and established connections.



## Conclusion:

**In this experiment, we learnt and understood how To study installation and configuration of Linux Kernel firewall iptables.**