Aarna Bafna
T11 - 07
Security Lab

Assignment No.10

Aim : To study and configure firewalls using IP table.

Theory :

Firewall:

A firewall is a system designed to prevent unauthorized access to or from a private network. You can implement a firewall in either hardware or software form, or a combination of both. Generally the firewall has two network interfaces: one for the external side of the network, one for the internal side. Its purpose is to control what traffic is allowed to traverse from one side to the other. As the most basic level, firewalls can block traffic intended for particular IP addresses or server ports.

TCP network traffic moves around a network in packets, which are containers that consist of a packet header—this contains control information such as source and destination addresses, and packet sequence information—and the data (also known as a payload). While the control information in each packet helps to ensure that its associated data gets delivered properly, the elements it contains also provides firewalls a variety of ways to match packets against firewall rules.

Types of Firewalls

Three basic types of network firewalls: packet filtering (stateless), stateful, and application layer.

Packet filtering, or stateless, firewalls work by inspecting individual packets in isolation. As such, they are unaware of connection state and can only allow or deny packets based on individual packet headers.

Stateful firewalls are able to determine the connection state of packets, which makes them much more flexible than stateless firewalls. They work by collecting related packets until the connection state can be determined before any firewall rules are applied to the traffic.

Application firewalls go one step further by analyzing the data being transmitted, which allows network traffic to be matched against firewall rules that are specific to individual services or applications. These are also known as proxy-based firewalls.

Basic of iptables:

Iptables is a firewall, installed by default on all official Ubuntu distributions (Ubuntu, Kubuntu, Xubuntu). When you install Ubuntu, iptables is there, but it allows all traffic by default.

The rules in IPTables are written to deal 3 different scenarios:

1.Those packets entering your machine that are destined for your machine. (INPUT) 2.Those packets leaving your machine. (OUTPUT)

3.Those packets entering your machine, but are destined for another machine and will pass through your machine (FORWARD).

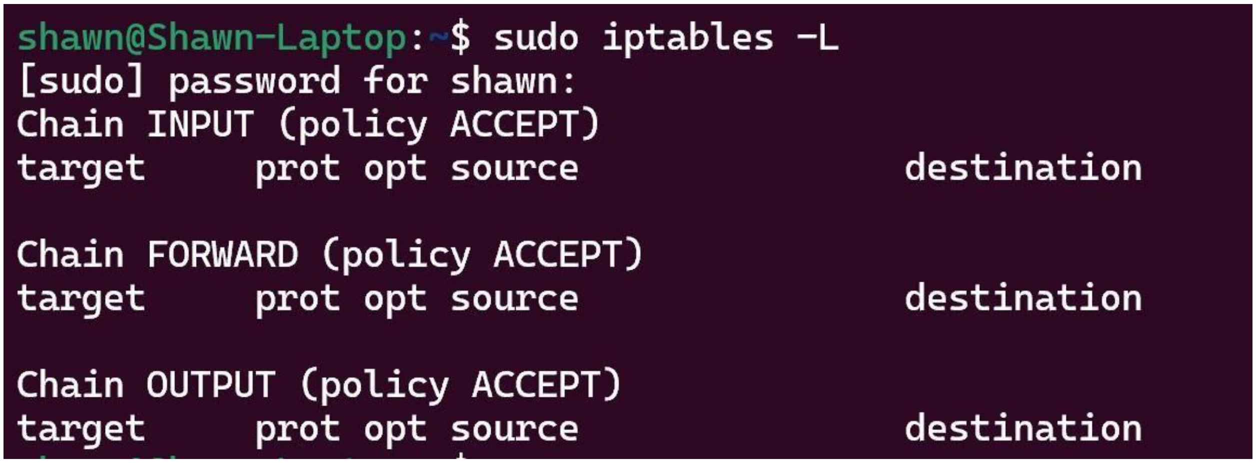
In Iptables, these scenarios are referred to as INPUT, OUTPUT, and FORWARD, respectively.

Once the traffic type has been specified, three actions may be taken:

1.ACCEPT allows packets to pass through the firewall.

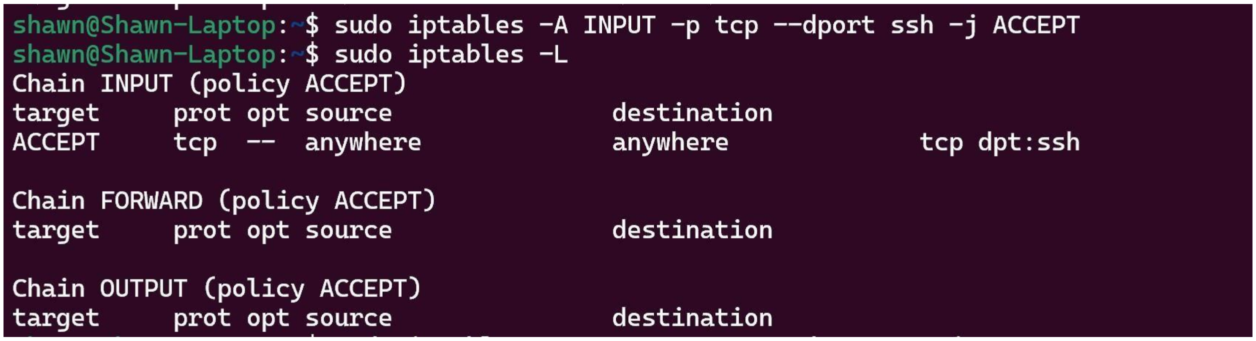
2.DROP ignores the packet and sends no response to the request.

3.REJECT ignores the packet, but responds to the request with a packet denied message.

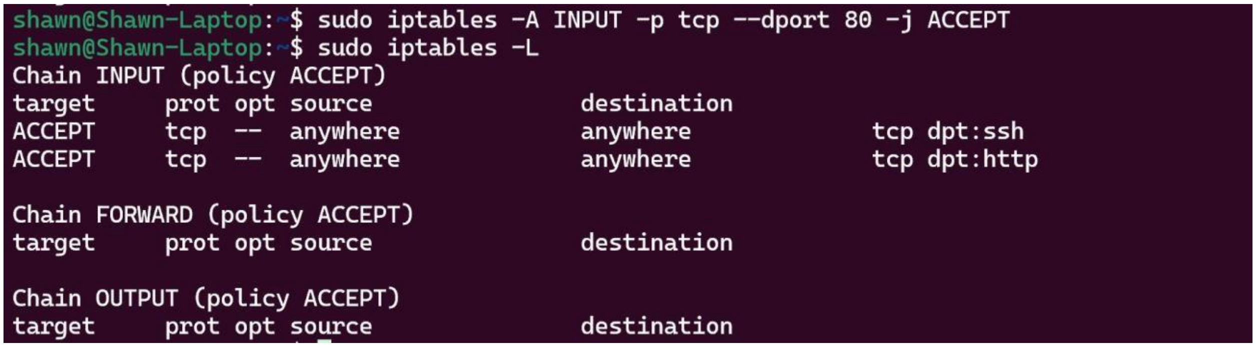


```
shawn@Shawn-Laptop:~$ sudo iptables -L
[sudo] password for shawn:
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

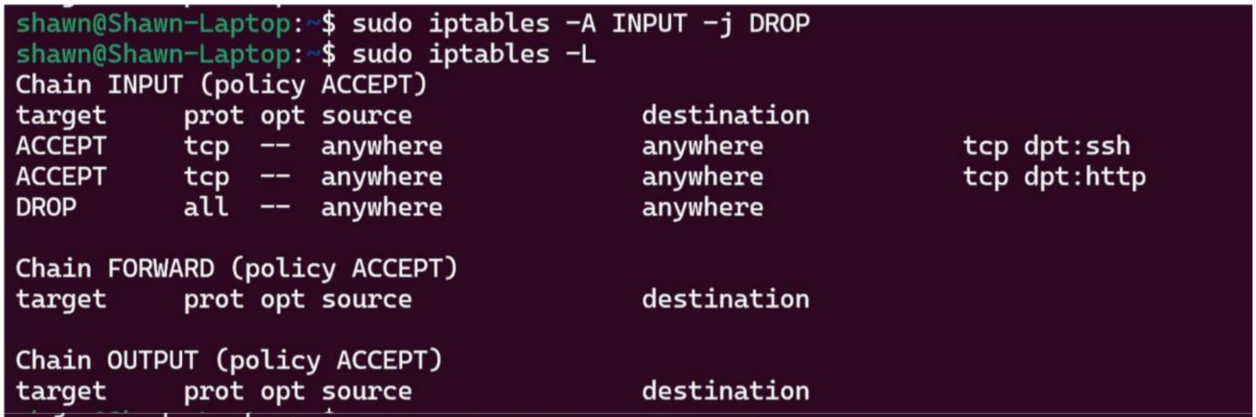
Output:



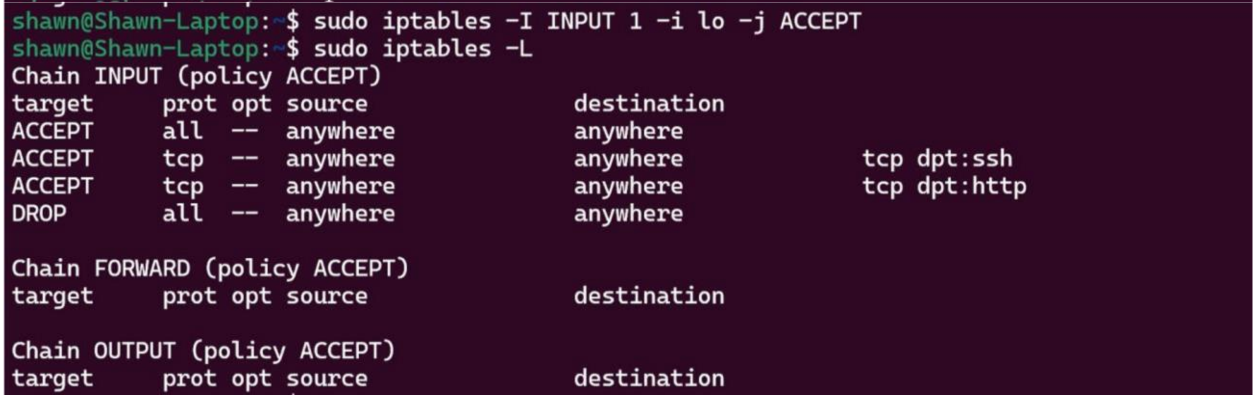
```
shawn@Shawn-Laptop:~$ sudo iptables -A INPUT -p tcp --dport ssh -j ACCEPT
shawn@Shawn-Laptop:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```



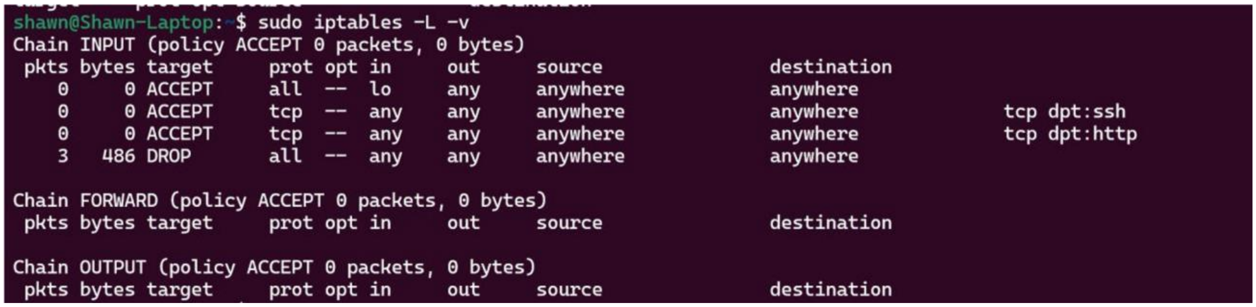
```
shawn@Shawn-Laptop:~$ sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT
shawn@Shawn-Laptop:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:http

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```



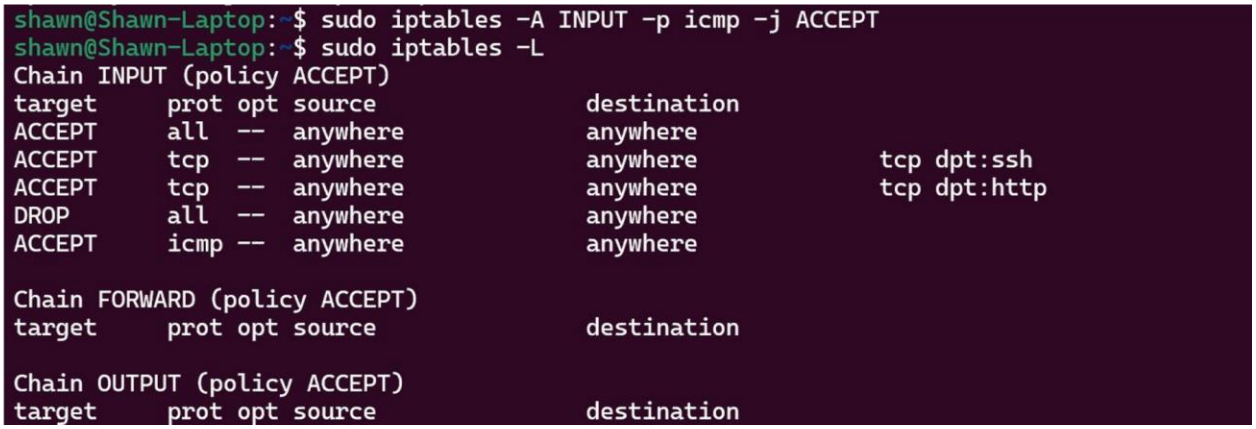
```
shawn@Shawn-Laptop:~$ sudo iptables -A INPUT -j DROP
shawn@Shawn-Laptop:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:http
DROP       all  --  anywhere             anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```



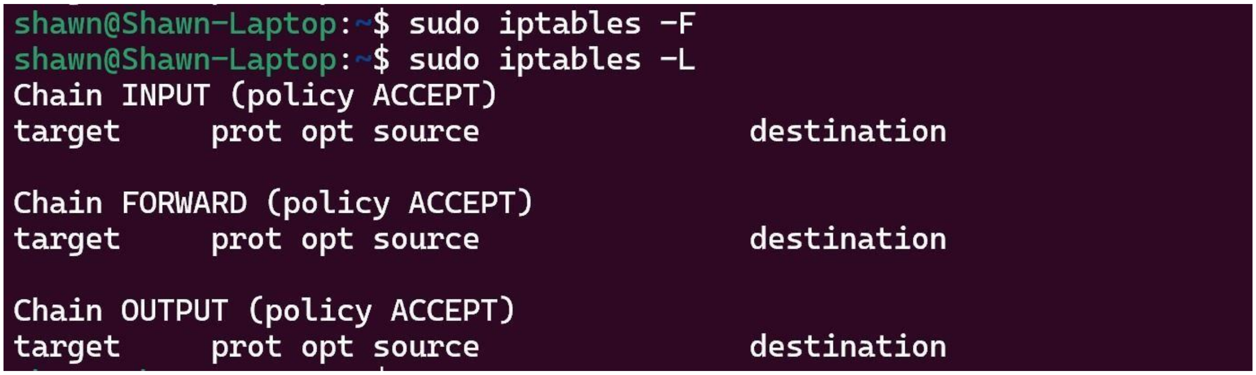
```
shawn@Shawn-Laptop:~$ sudo iptables -I INPUT 1 -i lo -j ACCEPT
shawn@Shawn-Laptop:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:http
DROP       all  --  anywhere             anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```



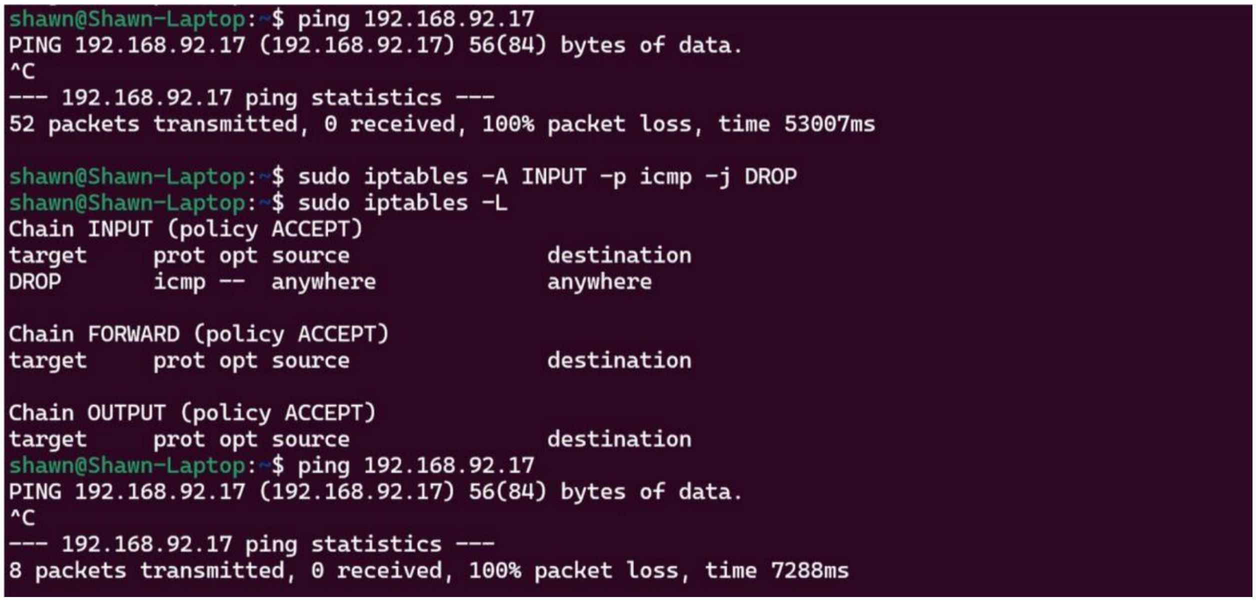
```
shawn@Shawn-Laptop:~$ sudo iptables -L -v
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     all  --  lo     any     anywhere             anywhere
    0     0 ACCEPT     tcp  --  any    any     anywhere             anywhere             tcp dpt:ssh
    0     0 ACCEPT     tcp  --  any    any     anywhere             anywhere             tcp dpt:http
    3   486 DROP       all  --  any    any     anywhere             anywhere

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
```



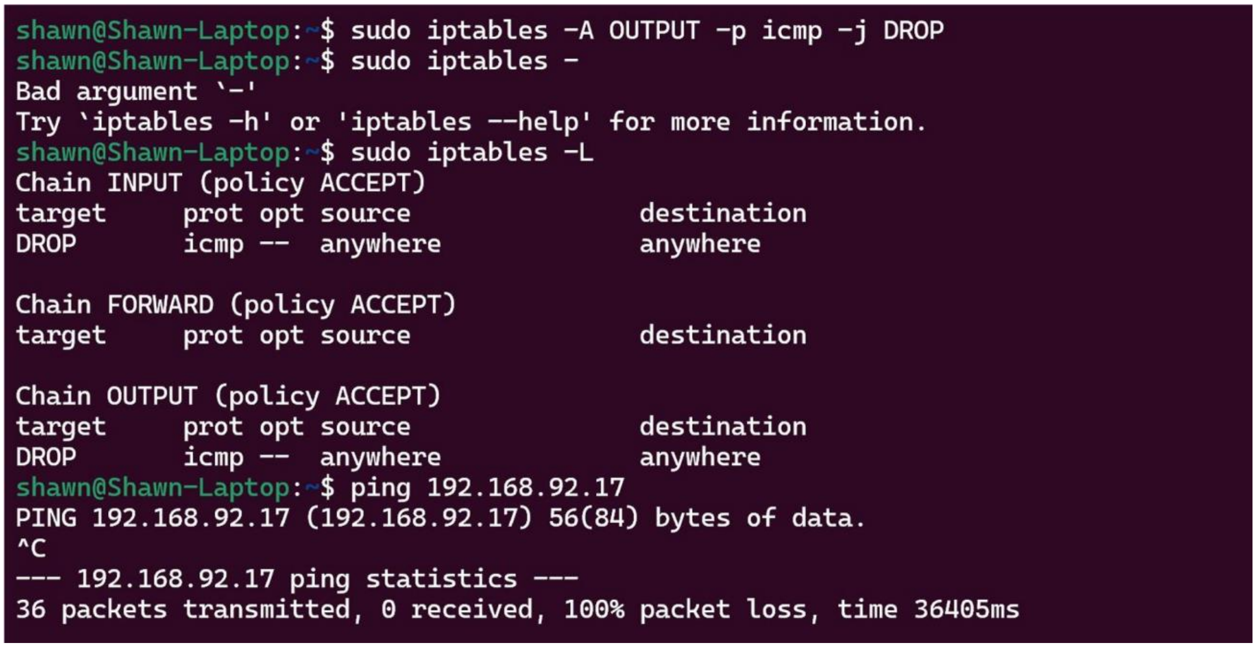
```
shawn@Shawn-Laptop:~$ sudo iptables -A INPUT -p icmp -j ACCEPT
shawn@Shawn-Laptop:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:http
DROP       all  --  anywhere             anywhere
ACCEPT     icmp --  anywhere             anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```



```
shawn@Shawn-Laptop:~$ sudo iptables -F
shawn@Shawn-Laptop:~$ sudo iptables -L
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
```



```
shawn@Shawn-Laptop:~$ ping 192.168.92.17
PING 192.168.92.17 (192.168.92.17) 56(84) bytes of data.
^C
--- 192.168.92.17 ping statistics ---
52 packets transmitted, 0 received, 100% packet loss, time 53007ms

shawn@Shawn-Laptop:~$ sudo iptables -A INPUT -p icmp -j DROP
shawn@Shawn-Laptop:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       icmp --  anywhere             anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
shawn@Shawn-Laptop:~$ ping 192.168.92.17
PING 192.168.92.17 (192.168.92.17) 56(84) bytes of data.
^C
--- 192.168.92.17 ping statistics ---
8 packets transmitted, 0 received, 100% packet loss, time 7288ms
```



```
shawn@Shawn-Laptop:~$ sudo iptables -A OUTPUT -p icmp -j DROP
shawn@Shawn-Laptop:~$ sudo iptables -
Bad argument `-'
Try `iptables -h' or 'iptables --help' for more information.
shawn@Shawn-Laptop:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       icmp --  anywhere             anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
DROP       icmp --  anywhere             anywhere
shawn@Shawn-Laptop:~$ ping 192.168.92.17
PING 192.168.92.17 (192.168.92.17) 56(84) bytes of data.
^C
--- 192.168.92.17 ping statistics ---
36 packets transmitted, 0 received, 100% packet loss, time 36405ms
```

Types of iptables:

I. IPTABLES TABLES and CHAINS

IPTables has the following 4 built-in tables.

1. Filter Table

Filter is default table for iptables. So, if you don’t define you own table, you’ll be using filter table. Iptables’s filter table has the following built-in chains.

- INPUT chain – Incoming to firewall. For packets coming to the local server.
- OUTPUT chain – Outgoing from firewall. For packets generated locally and going out of the local server.
- FORWARD chain – Packet for another NIC on the local server. For packets routed through the local server.

Type the following command and see the result sudo iptables -t filter -L

2. NAT table

Iptable’s NAT table has the following built-in chains.

- PREROUTING chain – Alters packets before routing. i.e Packet translation happens immediately after the packet comes to the system (and before routing). This helps to translate the destination ip address of the packets to something that matches the routing on the local server. This is used for DNAT (destination NAT).

- POSTROUTING chain – Alters packets after routing. i.e Packet translation happens when the packets are leaving the system. This helps to translate the source ip address of the packets to something that might match the routing on the desintation server.

This is used for SNAT (source NAT).

- OUTPUT chain – NAT for locally generated packets on the firewall.

Type the following command and see the result sudo iptables -t nat -L

3. Mangle table

Iptables’s Mangle table is for specialized packet alteration. This alters QOS bits in the TCP header. Mangle table has the following built-in chains.

- PREROUTING chain

- OUTPUT chain

- FORWARD chain

- INPUT chain

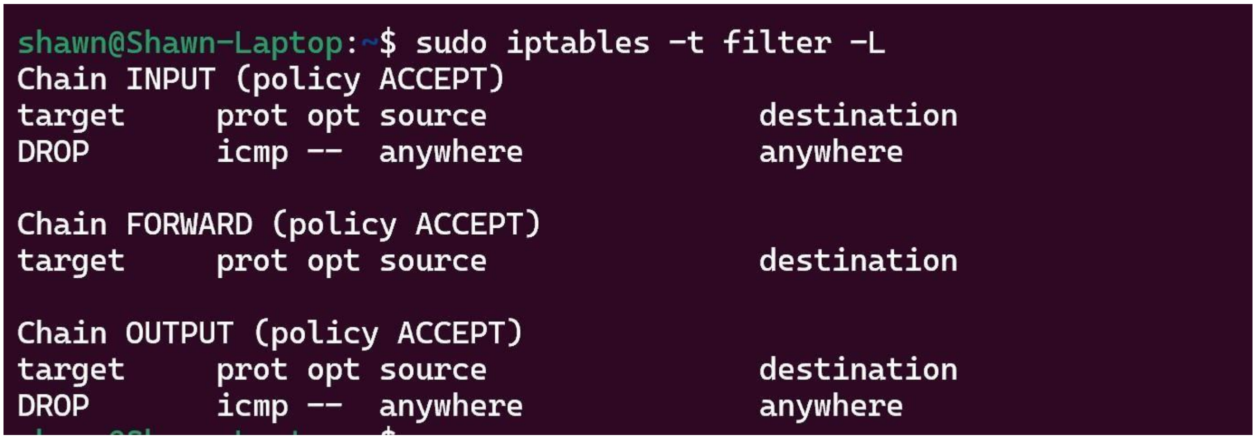
- POSTROUTING chain

Type the following command and see the result sudo iptables -t nat -L **4. Raw table**

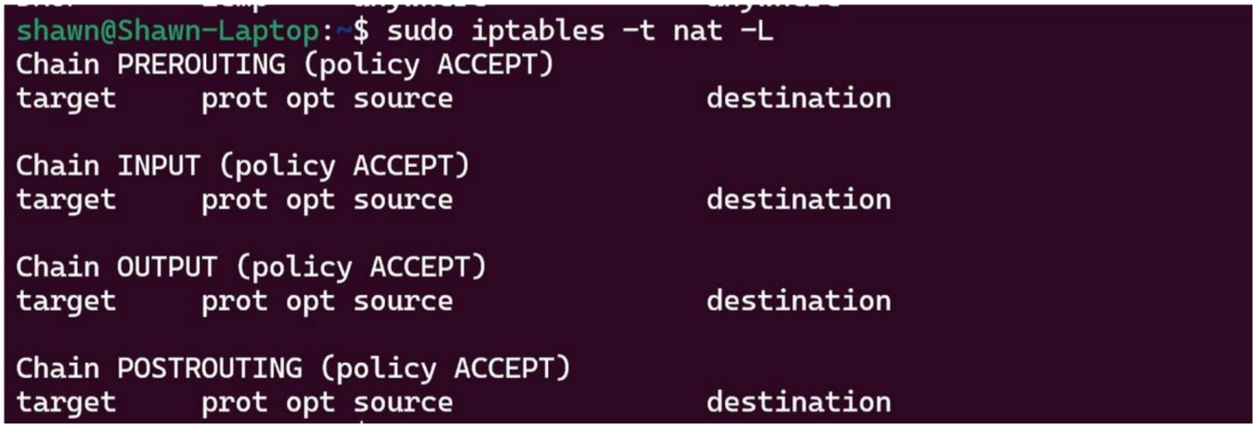
Iptable’s Raw table is for configuration excemptions. Raw table has the following built-in chains.

- PREROUTING chain

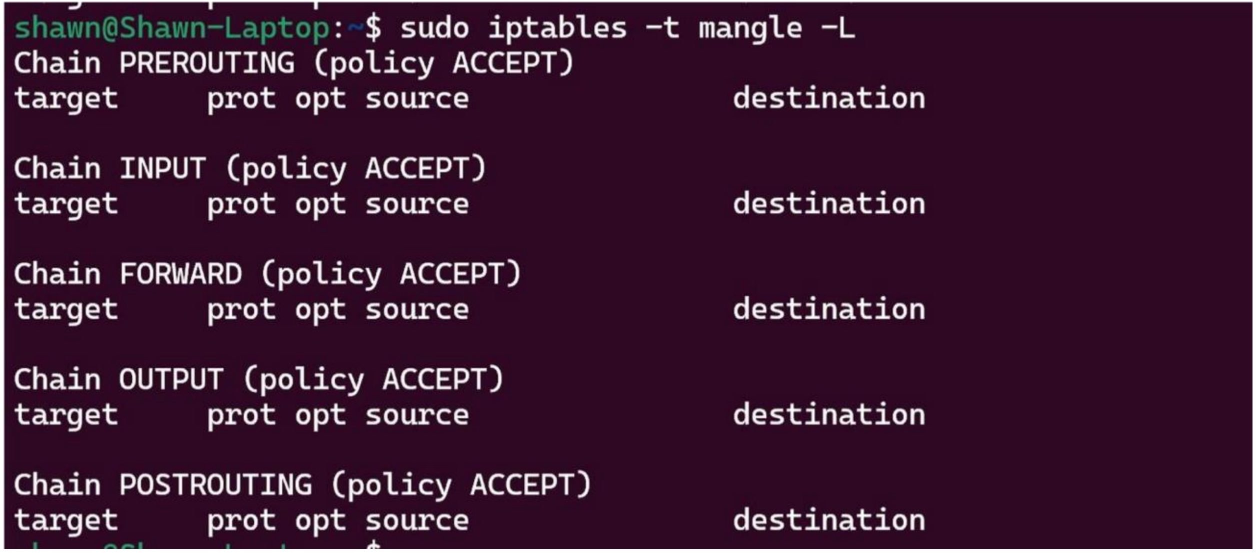
- OUTPUT chain



```
shawn@Shawn-Laptop:~$ sudo iptables -t filter -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       icmp --  anywhere             anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
DROP       icmp --  anywhere             anywhere
```



```
shawn@Shawn-Laptop:~$ sudo iptables -t nat -L
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination

Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination
```



```
shawn@Shawn-Laptop:~$ sudo iptables -t mangle -L
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination

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

Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination
```

Conclusion: Demonstrated the network security system using open source tools (LO6 is achieved).