LAB NO: 7 **Date:**

IPTABLES

Objectives:

In this lab, student will be able to:

- Identify the role & working of iptables.
- install iptables, configure, and use iptables in Linux
- defining a set of rules by which we can monitor, allow or block incoming or outgoing network packets.

Description:

In linux operating system, the firewalling is taken care of using netfilter. Which is a kernel module that decides what packets are allowed to come in or to go outside.iptables are just the interface to netfilter. The two might often be thought of as the same thing. A better perspective would be to think of it as a back end and a front end.The fundamentals, firewalling is the idea of deciding which packets are allowed to go in/out of the system.The packets in the internet (or any other network for that matter) are transferred using ports.We also have ports that are used by the user itself. For example when you have written a web application that runs on port 8000.To decide which port is allowed to communicate to the outside world (or even on the localhost) is the firewall's responsibility. You would command it to either accept, reject or drop a packet. Other things can also happen to a packet but let's keep it simple

IPTABLES ARCHITECTURE

iptables consists of different components which are discussed below:

- **chains**: There are 5 chains in iptables, and each is responsible for a specific task. These chains are: PREROUTING, input, forward, output & post routing. As their name suggests, they're responsible for packets either as soon as they arrive, if they are destined for local socket or just before routing to the outer world. We'll discuss these below.
- **tables**: Again, different tables are responsible for different tasks. The list contains filter, nat, mangle, raw & security. The first two are the most used. Filter is responsible for filtering and restricting the packets to/from our computer. Nat is responsible for Network Address Translation. We'll discuss these terms below as well.
- **targets**: Targets specify where a packet should go. This is decided using either iptables' own targets: ACCEPT, DROP, or RETURN, or it's extensions' target which are 39 at the moment and the most popular ones are DNAT, LOG, MASQUERADE, REJECT, SNAT, TRACE and TTL. Targets are divided into terminating and non-terminating. Which is just what the name suggests. Terminating targets ends rule traversal and the packets will be stopped there, but non-terminating ones touch a packet in some way and the rule traversal will continue afterward.

IPTABLES CHAINS

As mentioned above, each chain is responsible for a specific task.

- **Prerouting**: this chain decides what happens to a packet as soon as it arrives at the network interface. We have different options such as altering the packet (for NAT probably), dropping a packet, or doing nothing at all and letting it slip and be handled elsewhere along the way.
- **Input**: This is one of the popular chains as it almost always contains strict rules to avoid some evil doers on the internet harming our computer. If you want to open/block a port, this is where you'd do it.
- **Forward**: This chain is responsible for packet forwarding. Which is what the name suggests. We may want to treat a computer as a router and this is where some rules might apply to do the job.
- **Output**: This chain is the one responsible for all your web browsing among many others. You can't send a single packet without this chain allowing it. You have a lot of options whether you want to allow a port to communicate or not. It's the best place to limit your outbound traffic if you're not sure what port each application is communicating through. (A small hint: use the command ss -tulpen).
- **Postrouting**: This chain is where packets leave their trace last, before leaving our computer. This is used for routing among many other tasks just to make sure the packets are treated the way we want them to.

IPTABLES TABLES

- **Filter**: This is the table most used on a daily basis. Which is why it's the default table. In this table you would decide whether a packet is allowed in/out your computer. If you want to block a port to stop receiving anything, this is your stop.

- **Nat**: This table is the second most popular table and is responsible for creating new connection. Which is shorthand for Network Address Translation.

- **Mangle**: For specialized packets only. This table is for changing something inside the packet either before coming in or leaving out.

- **Raw**: This table is dealing with the raw packet as the name suggests. Mainly this is for tracking the connection state.

LAB EXERCISES

1.Identify the current iptables ruleset

Ans: iptables -S and sudo iptables -L.

2.Allowing Loopback Connections

(The loopback interface, also referred to as lo, is what a computer uses to forward network

connections to itself. For example, if you run ping localhost or ping 127.0.0.1)

sudo iptables -A INPUT -i lo -j ACCEPT

sudo iptables -A OUTPUT -o lo -j ACCEPT

3. o block network connections that originate from a specific IP address, 203.0.113.51 for example, run this command:

sudo iptables -A INPUT -s 203.0.113.51 -j DROP

4. To block connections from a specific IP address, e.g. 203.0.113.51, to a specific network interface, e.g. eth0, use this command:

iptables -A INPUT -i eth0 -s 203.0.113.51 -j DROP

5. Deleting Rules by Chain and Number

The other way to delete iptables rules is by its chain and line number. To determine a rule's line number, list the rules in the table format and add the --line-numbers option:

sudo iptables -L --line-numbers

6. delete all of the rules in the INPUT chain, run this command:

sudo iptables -F INPUT

7.Flushing All Chains

To flush all chains, which will delete all of the firewall rules, you may use the -F, or the equivalent --flush, option by itself:

sudo iptables -F

8.Reject all tcp packets with (specific ip,port numbers,mac address,destination port etc)

9. Filtering Packets Based on Source

10. Dropping all Other Traffic

11. Allow Traffic on Specific Ports

12. Dropping Unwanted Traffic