Ex. No.: 11 **Date:**

INSTALL AND CONFIGURE IPTABLES FIREWALL

Aim:

To install iptables and configure it for a variety of options.

Common Configurations & outputs:

1. **Start/stop/restart firewalls**

```
[root@localhost ~]# systemctl start firewalld
[root@localhost ~]# systemctl restart firewalld
 [root@localhost ~]# systemctl stop firewalld
[root@localhost ~]#
```

2. **Check all exitsting IPtables Firewall Rules**

```
[root@localhost ~]# iptables -L -n -v
[root@localhost ~]#
```

3. **Block specific IP Address(eg. 172.16.8.10) in IPtables Firewall**

```
[root@localhost ~]# iptables -A INPUT -s 172.16.8.10 -j DROP
[root@localhost ~]#
```

4. Block specifig port on IPtables Firewall

```
[root@localhost ~]# iptables -A OUTPUT -p tcp --dport xxx -j DROP
[root@localhost ~]#
```

5. Allow specific network range on particular port on iptables

```
[root@localhost ~]# iptables -A OUTPUT -p tcp -d 172.16.8.0/24 --dport xxx -j ACCEPT
 [root@localhost ~]#
```

6. **Block Facebook on IPTables**

```
[root@localhost ~]# host facebook.com
facebook.com has address 157.240.24.35
facebook.com has IPv6 address 2a03:2880:f10c:283:face:b00c:0:25de
facebook.com mail is handled by 10 smtpin.vvv.facebook.com.
```

7. Whois

```
[root@localhost ~]# whois 157.240.24.35 | grep CIDR CIDR: 157.240.0.0/16
[root@localhost ~]#
```

```
[root@localhost ~]# whois 157.240.24.35 [Querying whois.arin.net] [whois.arin.net]

#
# ARIN WHOIS data and services are subject to the Terms
of Use # available at:
https://www.arin.net/resources/registry/whois/tou/ #
# If you see inaccuracies in the results, please report at
#
https://www.arin.net/resources/registry/whois/inaccuracy_repo
rting/ #
# Copyright 1997-2019, American Registry for Internet
Numbers, Ltd. #


NetRange:          157.240.0.0 - 157.240.255.255 CIDR:       157.240.0.0/16
NetName:      THEFA-3 NetHandle:       NET-157-240-0-0-1
Parent:       NET157 (NET-157-0-0-0-0)
NetType:       Direct Assignment OriginAS:
Organization: Facebook, Inc. (THEFA-3) RegDate:        2015-05-14
Updated:       2015-05-14
Ref:          https://rdap.arin.net/registry/ip/157.240.0.0
OrgName:     Facebook, Inc. OrgId: THEFA-3
Address:       1601
Willow Rd. City:     Menlo Park StateProv:                CA
PostalCode:    94025
Country:       US
RegDate:       2004-08-11
Updated:       2012-04-17
Ref:          https://rdap.arin.net/registry/entity/THEFA-3
OrgTechHandle: OPERA82-ARIN
OrgTechName: Operations
OrgTechPhone: +1-650-543-4800
OrgTechEmail: domain@facebook.com
OrgTechRef:    https://rdap.arin.net/registry/entity/OPERA82-ARIN
OrgAbuseHandle: OPERA82-ARIN
OrgAbuseName: Operations
OrgAbusePhone: +1-650-543-4800
OrgAbuseEmail: domain@facebook.com
OrgAbuseRef:    https://rdap.arin.net/registry/entity/OPERA82-ARIN

#
# ARIN WHOIS data and services are subject to the Terms of Use
 # available at: https://www.arin.net/resources/registry/whois/tou/#
# If you see inaccuracies in the results, please report at
# https://www.arin.net/resources/registry/whois/inaccuracy_reporting/ #
# Copyright 1997-2019, American Registry for Internet  Numbers, Ltd. #
```

```
[root@localhost ~]# iptables -A OUTPUT -p tcp -d 157.240.0.0/16 -j DROP
```

Open browser and check whether http://facebook.com is accessible

To allow facebook use -D instead of -A option

```
[root@localhost ~]# iptables -D OUTPUT -p tcp -d 157.240.0.0/16 -j DROP
 [root@localhost ~]#
```

8. Block Access to your system from specific MAC Address(say 0F:22:1E:00:02:30)

```
    [root@localhost ~]# iptables -A INPUT -m mac --mac-source 0F:22:1E:00:02:30 -j DROP
   [root@localhost ~]#
```

9. Save IPtables rules to a file

```
[root@localhost ~]# iptables-save > ~/iptables.rules
 [root@localhost ~]# vi iptables.rules
[root@localhost ~]#
```

10. Restrict number of concurrent connections to a Server(Here restrict to 3 connections only)

```
[root@localhost ~]# iptables -A INPUT -p tcp --syn --dport 22 -m connlimit --connlimit-above 3 -j REJECT
```

11. Disable outgoing mails through IPtables

```
[root@localhost ~]# iptables -A OUTPUT -p tcp --dport 25 -j REJECT
[root@localhost ~]#
```

12. Flush IPtables Firewall chains or rules

```
[root@localhost ~]# iptables -F
[root@localhost ~]#
```

Result:

Installing and Configuring iptables

Objective:

To install iptables and configure it for a variety of options, gaining practical experience with firewall management.

Introduction:

iptables is a powerful command-line firewall utility for Linux systems. It allows you to define rules for network traffic, controlling which packets are allowed to pass through your system. This lab will guide you through installing iptables and configuring it for common scenarios.

Prerequisites:

- A Linux system (virtual machine or physical machine) with root or sudo privileges.
- Basic understanding of Linux command line.

Materials:

- A Linux system with network connectivity.

Procedure:

1. Installing iptables:

Most Linux distributions come with iptables pre-installed. However, if it's not, you can install it using your distribution's package manager.

- **Debian/Ubuntu-based systems:**

```
sudo apt update

sudo apt install iptables
```

- **Red Hat/CentOS-based systems:**

```
sudo yum update

sudo yum install iptables
```

- **Verify Installation:**

```
iptables -V
```

This command will display the iptables version, confirming successful installation.

2. Understanding iptables Basics:

iptables uses tables to organize rules. The most commonly used tables are:

- **filter:** The default table, used for general packet filtering (allowing or blocking traffic).
- **nat:** Used for Network Address Translation (NAT), which is often used to share a single public IP address among multiple devices on a local network.
- **mangle:** Used for specialized packet alteration.

Within each table, rules are organized into chains. Common chains in the `filter` table are:

- **INPUT:** Handles incoming traffic to the system.
- **OUTPUT:** Handles outgoing traffic from the system.
- **FORWARD:** Handles traffic passing through the system (e.g., routing between networks).

3. Basic iptables Commands:

- **Listing Rules:**

```
sudo iptables -L  # Lists rules in the filter table
sudo iptables -t nat -L # Lists rules in the nat table
sudo iptables -L -v # Lists rules with more details (verbose)
sudo iptables -L --line-numbers # Lists rules with line numbers (useful for deleting)
```

- **Appending a Rule (Adding a rule to the end of a chain):**

```
sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT  # Allow HTTP traffic
```

- `-A`: Append
- `INPUT`: Chain
- `-p tcp`: Protocol (tcp, udp, icmp)
- `--dport 80`: Destination port (for incoming traffic)
- `-j ACCEPT`: Action (ACCEPT, DROP, REJECT)
- **Inserting a Rule (Adding a rule at a specific position):**

  ```
  sudo iptables -I INPUT 2 -p udp --dport 53 -j ACCEPT # Insert rule at line 2
  ```

- -I: Insert

- 2: Line number
- **Deleting a Rule:**

  ```
  sudo iptables -D INPUT 2 # Delete rule at line 2
  ```
- -D: Delete

- **Flushing all Rules (Clearing all rules in a table):**
  ```
  sudo iptables -F # Flush the filter table
  ```

-
  ```
  sudo iptables -t nat -F # Flush the nat table
  ```
-

  -F: Flush

- **Saving Rules (M**

```
 sudo iptables-save > /etc/iptables/rules.v4 # Save IPv4 rules (Debian/Ubuntu)

sudo iptables-save > /etc/sysconfig/iptables # Save IPv4 rules        (Red Hat/CentOS)
```

- **Restoring Rules (Load saved rules):**

```
sudo iptables-restore < /etc/iptables/rules.v4 # Restore IPv4 rules (Debian/Ubuntu)
sudo iptables-restore < /etc/sysconfig/iptables # Restore IPv4 rules (Red Hat/CentOS)
```

4. Configuring iptables for Various Options:

- **Allowing SSH traffic:**

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

Blocking all incoming traffic (except SSH):

```
sudo iptables -P INPUT DROP  # Set default policy for INPUT chain to DROP
```

- **Allowing outgoing traffic:**

```
sudo iptables -P OUTPUT ACCEPT # Set default policy for OUTPUT chain to ACCEPT
```

- **Allowing specific IP address:**

  ```
  sudo iptables -A INPUT -s 192.168.1.10 -j ACCEPT
  ```

- **Blocking a specific IP address:**

  ```
  sudo iptables -A INPUT -s 192.168.1.20 -j DROP
  ```

Forwarding traffic (for routing):

```
sudo iptables -t nat -A POSTROUTING -j MASQUERADE # Enable NAT masquerading
sudo iptables -A FORWARD -i eth0 -o eth1 -j ACCEPT # Allow forwarding between interfaces
```

5. Saving and Restoring Rules:

After configuring iptables, save the rules to make them persistent across reboots. Use the commands mentioned in section 3.

Lab Exercises:

1. Configure iptables to allow HTTP and HTTPS traffic.
2. Block all ICMP (ping) traffic.
3. Allow SSH access only from a specific IP address.
4. Implement NAT for a local network.

Conclusion:

This lab provided a basic understanding of iptables installation and configuration. By experimenting with different rules and options, you can gain practical skills in managing network security using iptables.