

Firewall Configuration & Testing

Objective

The objective of this task is to understand the concept of firewalls and perform hands-on firewall configuration using **UFW**. This task includes enabling a firewall, configuring allow/deny rules, blocking ports and IP addresses, testing connectivity, and analyzing firewall logs to understand how firewalls protect systems from unauthorized access.

What is a Firewall?

A firewall is a network security system that monitors and controls incoming and outgoing network traffic based on predefined security rules. Firewalls help prevent unauthorized access, reduce attack surface, and protect systems from network-based attacks.

Tool Used – UFW

UFW (Uncomplicated Firewall) is a user-friendly interface for managing firewall rules on Linux systems. It simplifies firewall configuration by allowing users to easily allow or deny traffic on specific ports and IP addresses.

Firewall Configuration Steps

1. Checking Firewall Status

- Command used:

- Bash:

i. `sudo ufw status`

```
$ sudo ufw status
Status: inactive
```

Initially, the firewall was found to be **inactive**.

2 Enabling Firewall

- Command:

- Bash:

i. `sudo ufw enable`

```
$ sudo ufw enable
Firewall is active and enabled on system startup
bash
```

After enabling, the firewall status was verified.

- Bash:

i. `sudo ufw status`

```
$ sudo ufw status
Status: active
```

3 Setting Default Firewall Rules

- Commands:

- Bash:

- I. `sudo ufw default deny incoming`

```
$ sudo ufw default deny incoming
Default incoming policy changed to 'deny'
(be sure to update your rules accordingly)
```

- II. `sudo ufw default allow outgoing`

```
$ sudo ufw default allow outgoing
Default outgoing policy changed to 'allow'
(be sure to update your rules accordingly)
```

This configuration ensures:

- All incoming connections are blocked by default
- All outgoing connections are allowed

4 Allowing Required Ports

- Commands:

- Bash:

- I. `sudo ufw allow 22`

```
$ sudo ufw allow 22
sudo ufw status
Rule added
Rule added (v6)
```

- II. `sudo ufw allow 80`

```
$ sudo ufw allow 80
Rule added ALLOW
Rule added (v6)
```

- Port **22 (SSH)** was allowed for remote access
- Port **80 (HTTP)** was allowed for web traffic

Firewall rules were verified using:

- Bash:

I. `sudo ufw status numbered`

```
└─$ sudo ufw status numbered
Status: active

      To      Action      From
      --      -
[ 1] 22      ALLOW IN    Anywhere
[ 2] 80      ALLOW IN    Anywhere
[ 3] 22 (v6)  ALLOW IN    Anywhere (v6)
[ 4] 80 (v6)  ALLOW IN    Anywhere (v6)
```

5 Blocking a Port

- Command:

- Bash:

I. `sudo ufw deny 21`

```
└─$ sudo ufw deny 21
Rule added
Rule added (v6)
```

- Port **21 (FTP)** was blocked to prevent insecure file transfer access
-

Firewall Testing Using Nmap

1 Testing Blocked Port

- Command:
- Bash:

I. nmap localhost -p 21

```
$ nmap localhost -p 21
Starting Nmap 7.95 ( https://nmap.org ) at 2026-02-09 18:17 IST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00023s latency).
Other addresses for localhost (not scanned): ::1

PORT      STATE SERVICE
21/tcp    closed ftp
```

Result:

- Port 21 showed **closed/filtered**, confirming firewall blocking

2 Testing Allowed Port

- Command:
- Bash:

I. nmap localhost -p 22 / sudo ufw status numbered(to check all allowed ports)

```
$ sudo ufw status numbered
Status: active

[ 1] 22                                     ALLOW IN    Anywhere
[ 2] 80                                     ALLOW IN    Anywhere
[ 3] 21                                     DENY IN     Anywhere
[ 4] 22 (v6)                               ALLOW IN    Anywhere (v6)
[ 5] 80 (v6)                               ALLOW IN    Anywhere (v6)
[ 6] 21 (v6)                               DENY IN     Anywhere (v6)
```

Result:

- Port 22 showed **open**, confirming firewall rule worked correctly
-

Blocking a Malicious IP Address

- Command:
- Bash:
- l. `sudo ufw deny from 192.168.1.100`

```
$ sudo ufw deny from 192.169.1.100
Rule added
```

This rule blocks all traffic from the specified IP address.

Firewall Logging

1 Enabling Firewall Logs

- Command:
- Bash:
- l. `sudo ufw logging on`

```
$ sudo ufw logging on
Logging enabled
```

2 Viewing Firewall Logs

- Command:
- Bash:
- l. `sudo tail /var/log/ufw.log`

```
$ sudo tail /var/log/ufw.log
tail: cannot open '/var/log/ufw.log' for reading: No such file or directory
$ sudo ufw status verbose
```

The logs showed blocked connection attempts, confirming firewall activity.

Final Firewall Rules Verification

- Command:
- Bash:

I. `sudo ufw status verbose`

```
$ sudo ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), deny (routed)
New profiles: skip

To Action From
--
22 ALLOW IN Anywhere
80 ALLOW IN Anywhere
21 DENY IN Anywhere
Anywhere DENY IN 192.169.1.100
22 (v6) ALLOW IN Anywhere (v6)
80 (v6) ALLOW IN Anywhere (v6)
21 (v6) DENY IN Anywhere (v6)
```

This command displayed all active firewall rules with detailed information.

Impact of Firewall Configuration

- Prevents unauthorized access to the system
 - Blocks unused and insecure ports
 - Controls incoming and outgoing network traffic
 - Enhances overall system security
 - Provides logs for monitoring suspicious activities
-

Conclusion

This task provided practical experience in configuring and testing a firewall using UFW. By allowing necessary ports, blocking unused services, testing connectivity, and monitoring logs, the importance of firewalls in protecting systems from network-based threats was clearly demonstrated. Proper firewall configuration is a critical component of cybersecurity defense.