🛡 Firewall Configuration and Testing Report

1. Objective

The objective of this task was to configure a firewall to manage incoming and outgoing network traffic, allow necessary services, block insecure ports, test connectivity, and monitor firewall activity. The purpose was to understand how firewalls operate in real-world environments to improve system security.

2. Environment

- Operating System: Kali Linux
- Firewall Tool: UFW (Uncomplicated Firewall)
- Testing Tools: Netcat (nc), Python temporary web server, journalctl

3. Firewall Configuration

The firewall was enabled and configured with secure default rules:

- Incoming traffic: Denied by default
- Outgoing traffic: Allowed by default

Commands used:

- ➢ sudo ufw enable
- ➢ sudo ufw default deny incoming
- ➢ sudo ufw default allow outgoing

4. Firewall Rules Implemented

Rule No.	Action	Port	Protocol	Purpose
1	Allow	80	TCP	Allow web service traffic
2	Deny	21	TCP	Block insecure FTP service

Commands used:

- sudo ufw allow 80
- sudo ufw deny 21

5. Connectivity Testing

To properly test the open port, a temporary web server was started using Python to simulate a real web service on port 80.

Command used:

- sudo python3 -m http.server 80

Netcat was then used to test firewall behavior.

- **Test for allowed port:**
- nc -zv localhost 80

Result: Connection successful (port open)

- **Test for blocked port:**
- nc -zv localhost 21
  Result: Connection refused (port blocked)

These tests confirmed that the firewall rules were functioning as expected.

6. Firewall Logging and Monitoring

Firewall logging was enabled and logs were reviewed using:

- sudo ufw logging on
- sudo journalctl | grep UFW

Multiple [UFW BLOCK] entries were observed, showing blocked traffic with source IP addresses, destination ports, and protocols. This demonstrated active firewall monitoring.

7. Observations

- The firewall allowed only necessary services while blocking insecure ports.
- The temporary web server successfully simulated a real service for testing purposes.
- Unauthorized incoming traffic was blocked and logged.
- The system security posture was improved by minimizing exposed ports.

8. Conclusion

The firewall was successfully configured using UFW on Kali Linux. A temporary Python web server was used to simulate a web service on port 80 for realistic testing. Insecure FTP access on port 21 was blocked. Connectivity tests and log monitoring confirmed the effectiveness of the firewall configuration. This task provided practical experience in implementing and managing firewall security in real-world scenarios.