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.

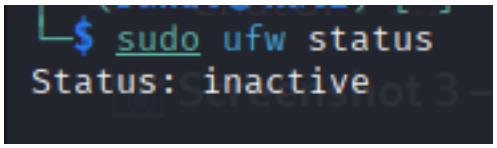
Firewall Configuration Steps

1. Checking Firewall Status

- Command used:

- Bash:

  i. sudo ufw status

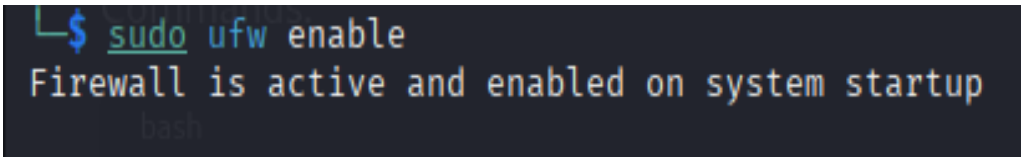


Initially, the firewall was found to be **inactive**.

2 Enabling Firewall

- Command:
- Bash:

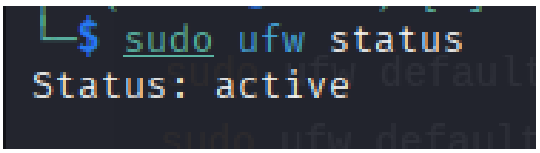
  i. sudo ufw enable



After enabling, the firewall status was verified.

- Bash:

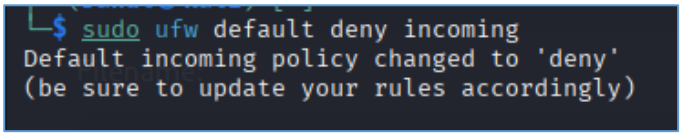
  I. sudo ufw status



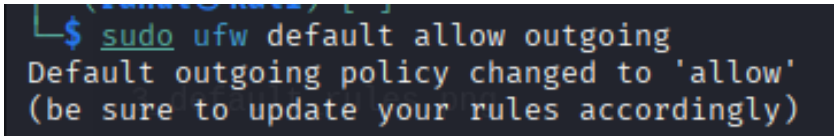
3 Setting Default Firewall Rules

- Commands:
- Bash:

I. sudo ufw default deny incoming



II. sudo ufw default allow outgoing



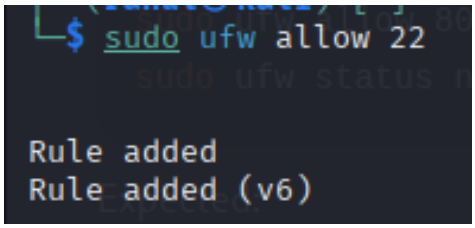
This configuration ensures:

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

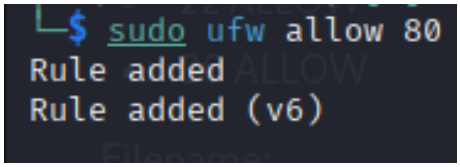
4 Allowing Required Ports

- Commands:
- Bash:

I. sudo ufw allow 22



II. sudo ufw allow 80

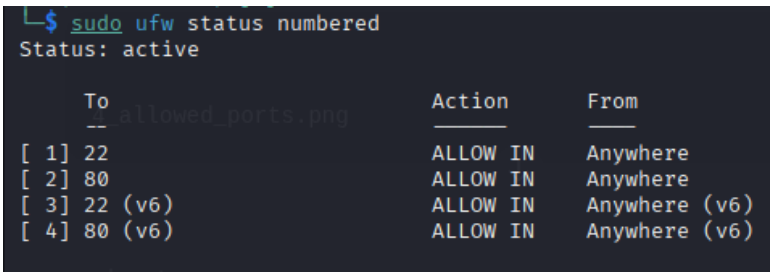


- Port **22 (SSH)** was allowed for remote access
- Port **80 (HTTP)** was allowed for web traffic

Firewall rules were verified using:

- Bash:

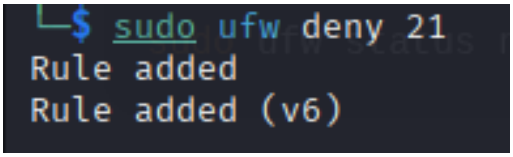
I. sudo ufw status numbered



5 Blocking a Port

- Command:
- Bash:

I. sudo ufw deny 21



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

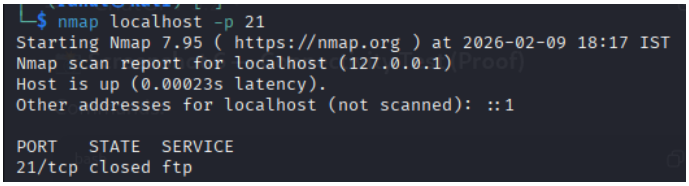
---

Firewall Testing Using Nmap

1 Testing Blocked Port

- Command:
- Bash:

I. nmap localhost -p 21



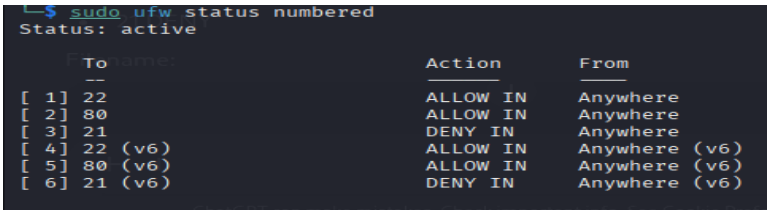
Result:

- 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)



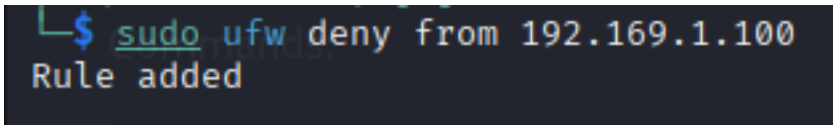
Result:

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

Blocking a Malicious IP Address

- Command:
- Bash:

I. sudo ufw deny from 192.168.1.100



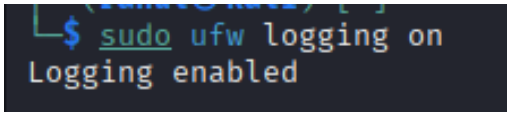
This rule blocks all traffic from the specified IP address.

Firewall Logging

1 Enabling Firewall Logs

- Command:
- Bash:

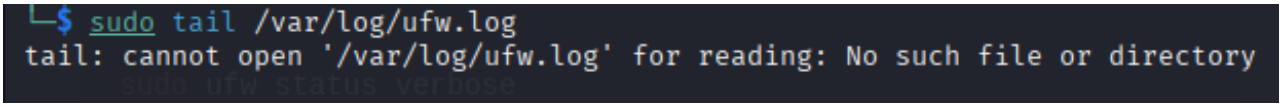
I. sudo ufw logging on



2 Viewing Firewall Logs

- Command:
- Bash:

I. sudo tail /var/log/ufw.log

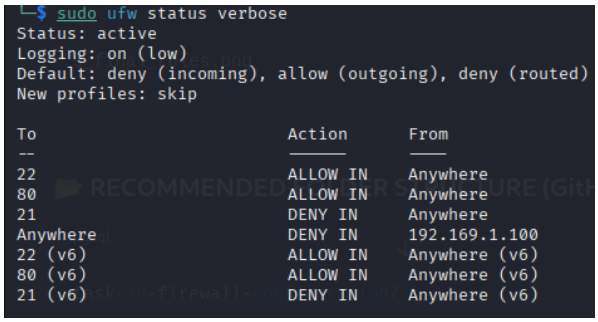


The logs showed blocked connection attempts, confirming firewall activity.

Final Firewall Rules Verification

- Command:
- Bash:

I. sudo ufw status verbose



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.