UFW Firewall Configuration and Testing Guide

Objective: Configure and test basic firewall rules to allow or block traffic using UFW on Linux
Tool: UFW (Uncomplicated Firewall)
Deliverables: Configuration commands and screenshot of applied rules
Date: June 27, 2025

1 Configuration Steps

1.1 Open UFW

Open a terminal on a Linux system with UFW installed. If UFW is not installed, install it using:

```
sudo apt install ufw
```

1.2 List Current Firewall Rules

Check the current UFW status and rules:

```
sudo ufw status
```

This displays whether UFW is active and lists all configured rules.

1.3 Add Rule to Block Inbound Traffic on Port 23 (Telnet)

Block all inbound TCP traffic on port 23 (Telnet) for IP address 192.168.1.100:

```
sudo ufw deny from any to 192.168.1.100 port 23 proto tcp
```

1.4 Reload UFW

Apply the changes:

```
sudo ufw reload
```

1.5 Test the Block Rule

Test the rule by attempting a Telnet connection to 192.168.1.100:

```
telnet 192.168.1.100 23
```

Expected Output: “Connection refused” or timeout, confirming the port is blocked.
Note: Ensure Telnet client is installed (`sudo apt install telnet`).

1.6 Add Rule to Allow SSH (Port 22)

Allow inbound TCP traffic on port 22 (SSH) for IP address 192.168.1.100:

```
sudo ufw allow from any to 192.168.1.100 port 22 proto tcp
```

1.7 Reload UFW

Apply the changes:

```
sudo ufw reload
```

1.8 Remove the Test Block Rule

Remove the Telnet block rule to restore the original state:

```
sudo ufw delete deny from any to 192.168.1.100 port 23 proto tcp
```

1.9 Reload UFW

Apply the changes:

```
sudo ufw reload
```

1.10 Verify Final Rules

Confirm the final configuration:

```
sudo ufw status
```

2 Configuration Commands

```
# Check UFW status
sudo ufw status

# Block inbound Telnet (port 23) for IP 192.168.1.100
sudo ufw deny from any to 192.168.1.100 port 23 proto tcp

# Reload UFW
sudo ufw reload

# Test Telnet connection
telnet 192.168.1.100 23

# Allow SSH (port 22) for IP 192.168.1.100
sudo ufw allow from any to 192.168.1.100 port 22 proto tcp

# Reload UFW
sudo ufw reload

# Remove Telnet block rule
sudo ufw delete deny from any to 192.168.1.100 port 23 proto tcp

# Reload UFW
sudo ufw reload
```

```
# Verify rules
sudo ufw status
```

3 Deliverables

- **Configuration File**: The commands above serve as the configuration file.
- **Screenshot**: Capture the output of `sudo ufw status` using a screenshot tool (e.g., `gnome-screenshot` or `scrot`). Save as `ufw_rules.png`.

  ```
  sudo ufw status > ufw_rules.txt
  gnome-screenshot -f ufw_rules.png
  ```

4 Detailed Explanation: How Firewalls Filter Traffic

Firewalls are critical network security tools that monitor and control incoming and outgoing network traffic based on predefined rules. They act as a barrier between a trusted internal network and untrusted external networks, ensuring only authorized traffic is allowed. Below is a detailed breakdown of how firewalls, such as UFW, filter network traffic:

4.1 Packet Inspection

Firewalls operate by inspecting network packets, which are small units of data transmitted over a network. Each packet contains:

- **Header**: Includes metadata like source IP, destination IP, source port, destination port, and protocol (e.g., TCP, UDP, ICMP).
- **Payload**: The actual data being transmitted.

Firewalls analyze packet headers to determine whether to allow, deny, or drop the packet based on configured rules.

4.2 Rule-Based Filtering

Firewalls use rules to evaluate packets. Each rule specifies criteria and an action:

- **Criteria**:
  - **Source IP Address**: The IP address of the sending device (e.g., 192.168.1.10).
  - **Destination IP Address**: The IP address of the receiving device (e.g., 192.168.1.100).
  - **Source/Destination Port**: The port number associated with a service (e.g., 23 for Telnet, 22 for SSH).
  - **Protocol**: The type of traffic (e.g., TCP, UDP, ICMP).
  - **Direction**: Inbound (traffic entering the system) or outbound (traffic leaving the system).
  - **Interface**: The network interface (e.g., eth0, wlan0) handling the traffic.

  - **Connection State**: For stateful firewalls, the state of the connection (e.g., NEW, ESTABLISHED, RELATED).

- **Actions**:
  - **Allow**: Permits the packet to pass.
  - **Deny/Drop**: Blocks the packet (deny may send a rejection message; drop silently discards it).
  - **Reject**: Explicitly rejects the packet with an error message (e.g., ICMP port unreachable).

4.3 Rule Processing Order

Firewalls process rules sequentially, typically in the order they are defined:

- The firewall evaluates each packet against the rules until a match is found.
- Once a matching rule is found, the specified action (allow, deny, etc.) is applied, and no further rules are checked.
- If no rule matches, the firewall applies its default policy (e.g., UFWs default is to deny all inbound traffic unless explicitly allowed).

4.4 Types of Firewalls

- **Stateless Firewalls**: Evaluate each packet independently based on header information, ignoring connection history. UFW operates primarily as a stateless firewall for simple rules.
- **Stateful Firewalls**: Track the state of active connections (e.g., NEW, ESTABLISHED, RELATED) using a state table, allowing return traffic for outbound requests.
- **Application-Layer Firewalls**: Inspect packet payloads to filter based on application-specific data (UFW focuses on network-layer filtering).

4.5 UFW-Specific Mechanisms

UFW, a simplified frontend for `iptables`, translates user-friendly commands into `iptables` rules. For example:

- `sudo ufw allow from any to 192.168.1.100 port 22 proto tcp` creates an `iptables` rule to accept TCP packets destined for port 22 on 192.168.1.100.
- UFW maintains chains (e.g., INPUT, OUTPUT, FORWARD) to organize rules for different traffic types.
- UFWs default policies ensure a secure baseline, requiring explicit rules for allowed traffic.

4.6 Advanced Filtering Features

- **Rate Limiting**: UFW supports rate-limiting to mitigate brute-force attacks (e.g., `sudo ufw limit 22/tcp`).

- **Logging**: UFW can log filtered packets for auditing (`sudo ufw logging on`).
- **Network Address Translation (NAT)**: UFW integrates with `iptables` for NAT or port forwarding.

4.7 Security Implications

- Firewalls reduce the attack surface by limiting open ports and services.
- Misconfigured rules can expose vulnerabilities.
- Regular rule reviews and logging detect unauthorized access attempts.

4.8 Performance Considerations

- Firewalls introduce minimal latency by processing packets at the kernel level.
- Complex rule sets or high traffic volumes may require optimization.

By defining precise rules, UFW ensures only authorized traffic is allowed, while unauthorized traffic is blocked, enhancing system security.

5 Notes

- **Permissions**: Root privileges (`sudo`) are required for UFW commands.
- **Testing**: Install Telnet client for testing (`sudo apt install telnet`).
- **Persistence**: Enable UFW for persistent rules (`sudo ufw enable`).
- **Documentation**: Save `ufw`$_r ules.txtand$`ufw`$_r ules.pngforsubmission.$