

# Setup and Use a Firewall on Windows/Linux

**Objective:** Configure and test basic firewall rules to allow or block traffic.

**Tools:** Windows Firewall / UFW (Uncomplicated Firewall) on Linux.

**Deliverables:** Screenshot/configuration file showing firewall rules applied.

**STEP 01-** Open firewall configuration tool ( Terminal for UFW).

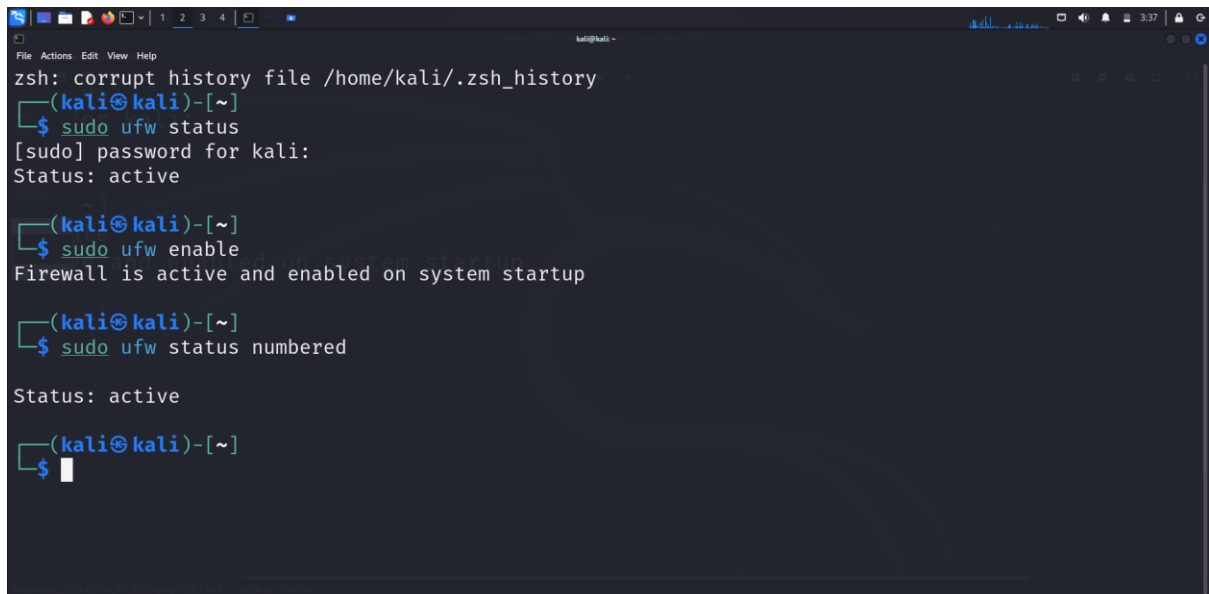
A terminal window on a Kali Linux system. The terminal shows the following commands and output: 1. A message: "zsh: corrupt history file /home/kali/.zsh\_history". 2. Prompt: "(kali㉿kali)-[~]". 3. Command: "\$ sudo ufw status". 4. Output: "[sudo] password for kali:" followed by "Status: active". 5. Prompt: "(kali㉿kali)-[~]". 6. Command: "\$ sudo ufw enable". 7. Output: "Firewall is active and enabled on system startup". 8. Prompt: "(kali㉿kali)-[~]". 9. Command: "\$" followed by a cursor. The terminal has a dark background with a faint Kali Linux dragon logo. The window title is "kali@kali: ~". The top bar shows standard Linux window controls and system status (3:35).

```
File Actions Edit View Help
kali@kali: ~
zsh: corrupt history file /home/kali/.zsh_history
(kali㉿kali)-[~]
$ sudo ufw status
[sudo] password for kali:
Status: active

(kali㉿kali)-[~]
$ sudo ufw enable
Firewall is active and enabled on system startup

(kali㉿kali)-[~]
$
```

## STEP 02 - List current firewall rules

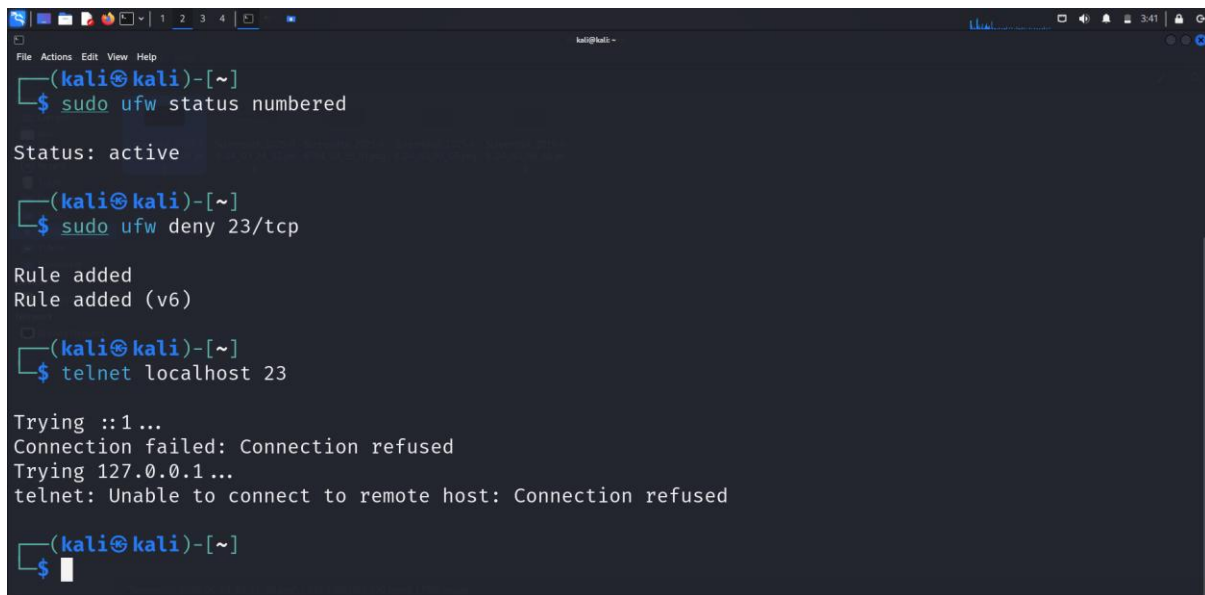
A terminal window on a Kali Linux system. The user has corrupted the zsh history file and is now in a shell. They run 'sudo ufw status', which shows the firewall is active. Then they run 'sudo ufw enable', which shows the firewall is active and enabled on system startup. Finally, they run 'sudo ufw status numbered', which shows the firewall is active. The prompt is now '\$'.

**ACTIVE**

## STEP 03 - Add a rule to block inbound traffic on a specific port (e.g., 23 for Telnet).

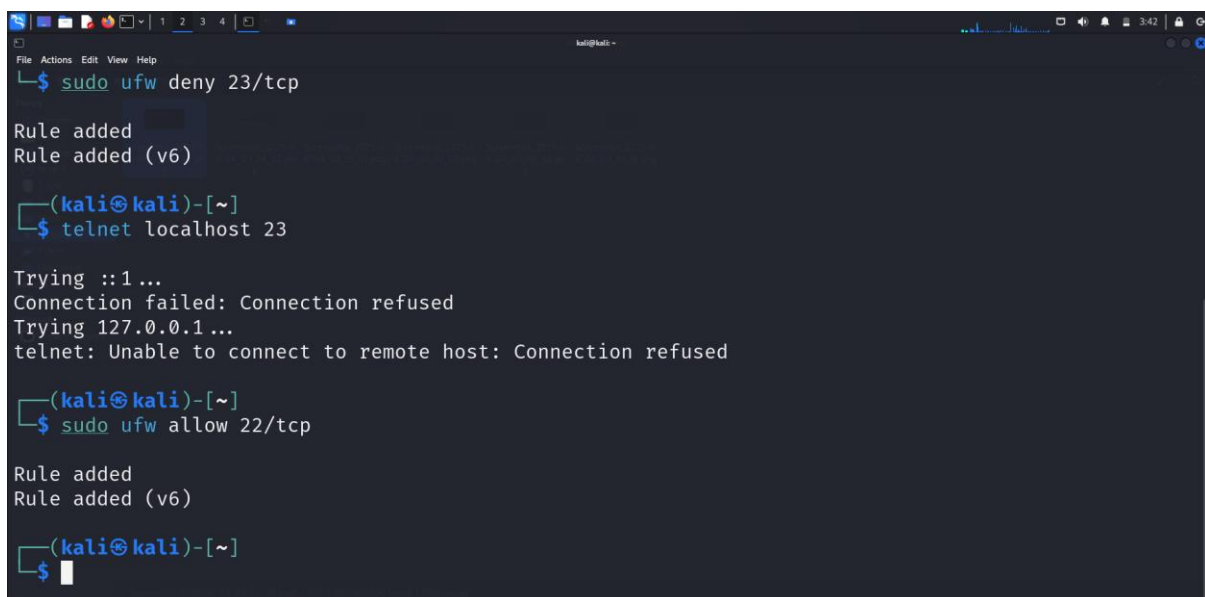
A terminal window on a Kali Linux system. The user runs 'sudo ufw status', which shows the firewall is active. Then they run 'sudo ufw enable', which shows the firewall is active and enabled on system startup. Then they run 'sudo ufw status numbered', which shows the firewall is active. Finally, they run 'sudo ufw deny 23/tcp', which shows the rule added. The prompt is now '\$'.

**STEP 04** - Test the rule by attempting to connect to that port locally or remotely.

A terminal window on a Kali Linux system. The user runs 'sudo ufw status numbered', which shows 'Status: active'. Then they run 'sudo ufw deny 23/tcp', which shows 'Rule added' and 'Rule added (v6)'. Finally, they run 'telnet localhost 23', which shows 'Trying ::1...', 'Connection failed: Connection refused', 'Trying 127.0.0.1...', and 'telnet: Unable to connect to remote host: Connection refused'.

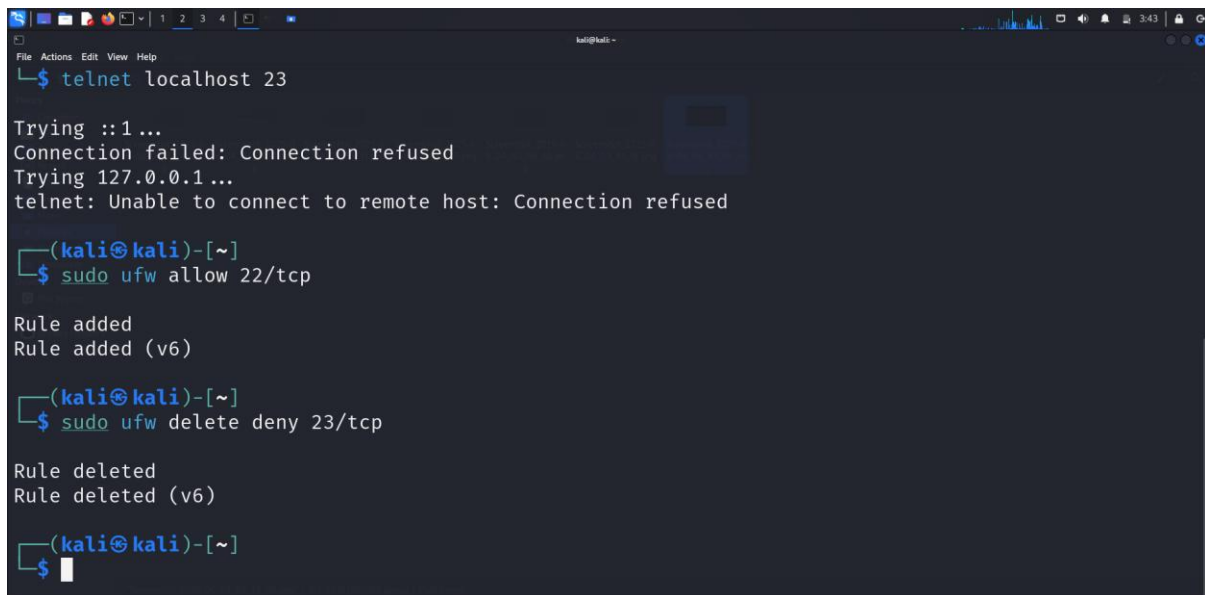
```
(kali㉿kali)-[~]  
$ sudo ufw status numbered  
  
Status: active  
  
(kali㉿kali)-[~]  
$ sudo ufw deny 23/tcp  
  
Rule added  
Rule added (v6)  
  
(kali㉿kali)-[~]  
$ telnet localhost 23  
  
Trying ::1...  
Connection failed: Connection refused  
Trying 127.0.0.1...  
telnet: Unable to connect to remote host: Connection refused  
  
(kali㉿kali)-[~]  
$
```

**STEP 05** - Add rule to allow SSH (port 22) if on Linux

A terminal window on a Kali Linux system. The user runs 'sudo ufw deny 23/tcp', which shows 'Rule added' and 'Rule added (v6)'. Then they run 'telnet localhost 23', which shows 'Trying ::1...', 'Connection failed: Connection refused', 'Trying 127.0.0.1...', and 'telnet: Unable to connect to remote host: Connection refused'. Finally, they run 'sudo ufw allow 22/tcp', which shows 'Rule added' and 'Rule added (v6)'.

```
(kali㉿kali)-[~]  
$ sudo ufw deny 23/tcp  
  
Rule added  
Rule added (v6)  
  
(kali㉿kali)-[~]  
$ telnet localhost 23  
  
Trying ::1...  
Connection failed: Connection refused  
Trying 127.0.0.1...  
telnet: Unable to connect to remote host: Connection refused  
  
(kali㉿kali)-[~]  
$ sudo ufw allow 22/tcp  
  
Rule added  
Rule added (v6)  
  
(kali㉿kali)-[~]  
$
```

## STEP 06 – Remove the test block rule to restore original state.

A terminal window on a Kali Linux system. The user runs 'telnet localhost 23', which fails with 'Connection refused'. Then they run 'sudo ufw allow 22/tcp', which adds a rule. Finally, they run 'sudo ufw delete deny 23/tcp', which deletes a rule. The terminal output is as follows:

```
(kali㉿kali)-[~]  
$ telnet localhost 23  
  
Trying ::1...  
Connection failed: Connection refused  
Trying 127.0.0.1...  
telnet: Unable to connect to remote host: Connection refused  
  
(kali㉿kali)-[~]  
$ sudo ufw allow 22/tcp  
  
Rule added  
Rule added (v6)  
  
(kali㉿kali)-[~]  
$ sudo ufw delete deny 23/tcp  
  
Rule deleted  
Rule deleted (v6)  
  
(kali㉿kali)-[~]  
$
```

## STEP 07 - Document commands or GUI steps used.

- **sudo ufw status numbered**

### Explanation:

This command shows the current list of firewall rules in an ordered format (numbered).

This is useful when you want to delete a specific rule by its number later.

- **sudo ufw deny 23/tcp**

### Explanation:

This rule blocks all TCP traffic on port 23, which is used for Telnet .Telnet is insecure and often blocked in secure environments.

**Why it's done:**

You're simulating a security measure — preventing remote access via an outdated protocol.

- **telnet localhost 23**

**Explanation:**

This command tries to connect to the Telnet service on your own machine (port 23)

- **sudo ufw allow 22/tcp**

**Explanation:**

Allows inbound TCP connections on port 22, which is used for SSH (Secure Shell).

This ensures you can still remotely access your system securely.

**Why it's done:**

To prevent accidentally locking yourself out of the system (especially on remote servers).

- **sudo ufw delete deny 23/tcp**

**Explanation:**

This command removes the previously added rule that blocked Telnet (port 23).

## **STEP 08** - Summarize how firewall filters traffic.

A firewall acts as a barrier between a trusted network (like your computer or internal network) and untrusted networks (like the internet). It filters incoming and outgoing traffic based on a set of rules.

### **Key Functions of a Firewall:**

#### **1. Packet Inspection:**

It examines each data packet's source, destination IP, port number, and protocol.

#### **2. Rule-Based Filtering:**

Based on configured rules (like allow/deny on specific ports), it either:

- **Allows** (accepts) the packet, or
- **Blocks** (drops/denies) it.

#### **3. Port Control:**

Only open (allowed) ports can receive traffic. For example:

- Allow SSH on port 22
- Block Telnet on port 23

#### **4. Direction Control:**

Rules can apply to:

- **Inbound traffic** (from outside to your system)
- **Outbound traffic** (from your system to outside)

