

Task 4: Setup and Use a Firewall on Kali Linux with UFW

✓ Step-by-Step Process

1. Install and Enable UFW

- `$sudo apt update`
- `$sudo apt install ufw -y` # Install UFW if not already installed

```
Home x Kali Linux x
1 2 3 4 k
Session Actions Edit View Help
(kalix@KaliX)-[~]
$ sudo apt install ufw -y
[sudo] password for kalix:
Installing:
  ufw

Suggested packages:
  rsyslog

Summary:
  Upgrading: 0, Installing: 1, Removing: 0, Not Upgrading: 25
  Download size: 169 kB
  Space needed: 880 kB / 21.8 GB available

Get:1 http://kali.download/kali kali-rolling/main amd64 ufw all 0.36.2-9 [169 kB]
Fetched 169 kB in 1s (150 kB/s)
Preconfiguring packages ...
Selecting previously unselected package ufw.
(Reading database ... 421967 files and directories currently installed.)
Preparing to unpack .../archives/ufw_0.36.2-9_all.deb ...
Unpacking ufw (0.36.2-9) ...
Setting up ufw (0.36.2-9) ...
Creating config file /etc/ufw/before.rules with new version
Creating config file /etc/ufw/before6.rules with new version
Creating config file /etc/ufw/after.rules with new version
Creating config file /etc/ufw/after6.rules with new version
update-rc.d: We have no instructions for the ufw init script.
update-rc.d: It looks like a non-network service, we enable it.
Created symlink '/etc/systemd/system/multi-user.target.wants/ufw.service' -> '/usr/lib/systemd/system/ufw.service'.
Processing triggers for kali-menu (2025.4.0) ...
Processing triggers for man-db (2.13.1-1) ...
```

- `$sudo ufw enable` # Enable firewall

```
(kalix@KaliX)-[~]
$ sudo ufw enable
Firewall is active and enabled on system startup

(kalix@KaliX)-[~]
$
```

- `$sudo ufw status verbose` # Show current firewall status and rules

```
(kalix@KaliX)-[~]
$ sudo ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), deny (routed)
New profiles: skip

(kalix@KaliX)-[~]
$
```

2. Check Default Policies

Use if Default Policies not set

- `$sudo ufw default deny incoming`
- `$sudo ufw default allow outgoing`

3. List current firewall rules

- `$sudo ufw status numbered`

```
(kalix@KaliX)-[~]  
$ sudo ufw status numbered  
[sudo] password for kalix:  
Status: active  
  
(kalix@KaliX)-[~]  
$
```

- Output shows all rules in effect (numbered for easy deletion later).

4. Block Inbound Traffic on Port 23 (Telnet)

- `$sudo ufw deny 23/tcp`
- `$sudo ufw status numbered`

```
(kalix@KaliX)-[~]  
$ sudo ufw deny 23/tcp  
Rule added  
Rule added (v6)  
  
(kalix@KaliX)-[~]  
$ sudo ufw status numbered  
Status: active  
  
      To Action From  
      --  
[ 1] 23/tcp DENY IN Anywhere  
[ 2] 23/tcp (v6) DENY IN Anywhere (v6)
```

- Now port **23/TCP** is blocked.

5. Test the Rule

- `$nc -vz localhost 23`

```
(kalix@KaliX)-[~]  
$ nc -vz localhost 23  
localhost [127.0.0.1] 23 (telnet) : Connection refused  
  
(kalix@KaliX)-[~]  
$
```

- Result: **connection refused or blocked.**

6. Allow SSH (Port 22)

To avoid locking yourself out:

- `$sudo ufw allow 22/tcp`
- `$sudo ufw status numbered`

```
(kalix@KaliX)-[~]
$ sudo ufw allow 22/tcp
[sudo] password for kalix:
Rule added
Rule added (v6)

(kalix@KaliX)-[~]
$ sudo ufw status numbered
Status: active

      To Action From
      --
[ 1] 23/tcp DENY IN Anywhere
[ 2] 22/tcp ALLOW IN Anywhere
[ 3] 23/tcp (v6) DENY IN Anywhere (v6)
[ 4] 22/tcp (v6) ALLOW IN Anywhere (v6)

(kalix@KaliX)-[~]
$
```

- This ensures remote SSH is still accessible.

7. Remove the Test Rule

When finished, remove the Telnet block rule:

- `$sudo ufw delete deny 23/tcp`

```
(kalix@KaliX)-[~]
$ sudo ufw delete deny 23/tcp
Rule deleted
Rule deleted (v6)

(kalix@KaliX)-[~]
$ sudo ufw status numbered
Status: active

      To Action From
      --
[ 1] 22/tcp ALLOW IN Anywhere
[ 2] 22/tcp (v6) ALLOW IN Anywhere (v6)

(kalix@KaliX)-[~]
$
```

Or delete by rule number:

- `$sudo ufw status numbered`