

Day 4 - Firewall Configuration and Traffic Filtering Report

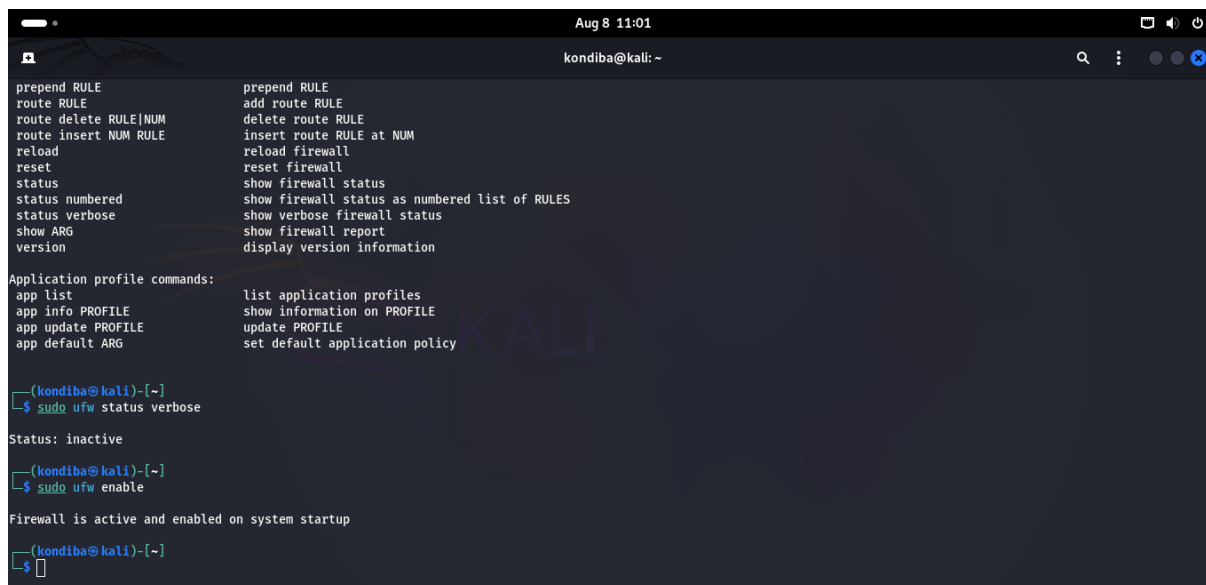
Introduction

This report documents the process of basic firewall management and network traffic filtering using UFW (Uncomplicated Firewall) on Kali Linux.

The steps cover enabling the firewall, adding and removing rules, and verifying their effects using Nmap scans.

Initial Firewall Status:

The firewall (UFW) was initially inactive, meaning no filtering rules were applied to incoming or outgoing network traffic. This could leave the system exposed to potential unauthorized access attempts.

A screenshot of a Kali Linux terminal window. The window title is "Aug 8 11:01" and the user is "kondiba@kali: ~". The terminal shows a list of UFW commands on the left and their descriptions on the right. Below this, it shows application profile commands. The user then runs "sudo ufw status verbose", which returns "Status: inactive". Next, the user runs "sudo ufw enable", which returns "Firewall is active and enabled on system startup".

```
prepend RULE      prepend RULE
route RULE        add route RULE
route delete RULE|NUM  delete route RULE
route insert NUM RULE  insert route RULE at NUM
reload            reload firewall
reset            reset firewall
status            show firewall status
status numbered      show firewall status as numbered list of RULES
status verbose        show verbose firewall status
show ARG            show firewall report
version            display version information

Application profile commands:
app list           list application profiles
app info PROFILE   show information on PROFILE
app update PROFILE update PROFILE
app default ARG     set default application policy

(kondiba@kali)-[~]
└─$ sudo ufw status verbose

Status: inactive

(kondiba@kali)-[~]
└─$ sudo ufw enable

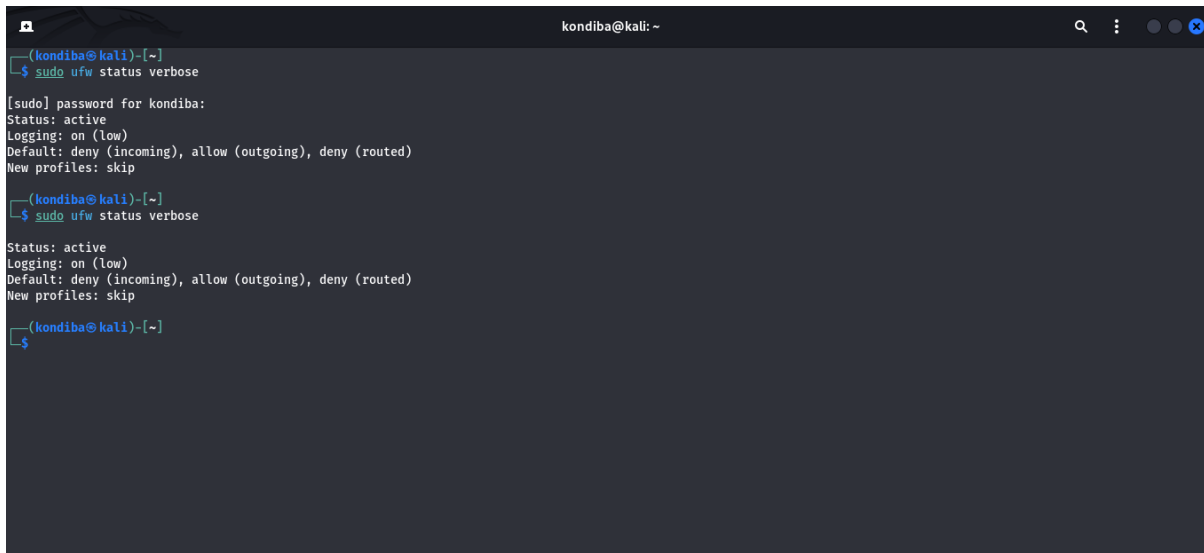
Firewall is active and enabled on system startup

(kondiba@kali)-[~]
└─$
```

Firewall Activation:

The UFW firewall was enabled using 'sudo ufw enable'. This action starts the firewall service and ensures it runs on system startup, applying default and user-defined rules.

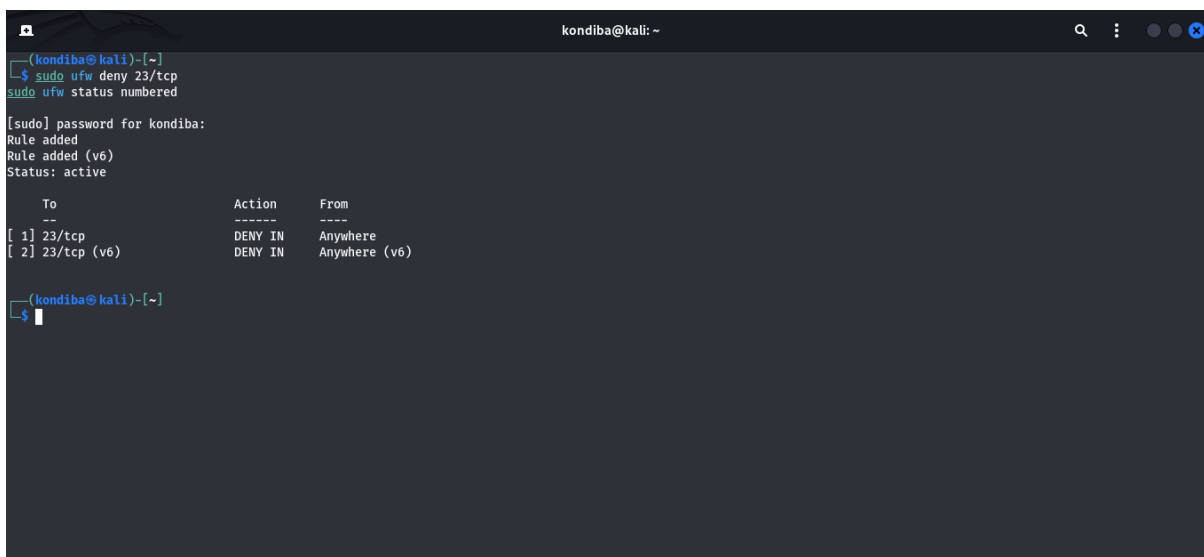
Day 4 - Firewall Configuration and Traffic Filtering Report

A terminal window titled 'kondiba@kali: ~' showing the command 'sudo ufw status verbose' being executed twice. The output for both executions is identical: 'Status: active', 'Logging: on (low)', 'Default: deny (incoming), allow (outgoing), deny (routed)', and 'New profiles: skip'.

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

Blocking Telnet (Port 23):

A firewall rule was added to deny incoming traffic on port 23 (Telnet). Telnet is insecure as it transmits data in plaintext, including credentials, making it vulnerable to interception.

A terminal window titled 'kondiba@kali: ~' showing the command 'sudo ufw deny 23/tcp' being executed, followed by 'sudo ufw status numbered'. The output shows the rule being added successfully. Below the status output, a table lists the active rules.

```
(kondiba@kali)-[~]  
$ sudo ufw deny 23/tcp  
[sudo] password for kondiba:  
Rule added  
Rule added (v6)  
Status: active  
  
To Action From  
--  
[ 1] 23/tcp DENY IN Anywhere  
[ 2] 23/tcp (v6) DENY IN Anywhere (v6)  
  
(kondiba@kali)-[~]  
$
```

Port 23 Reachability Test:

Using Nmap, port 23 was confirmed as closed. This indicates that the firewall successfully blocked access to the Telnet service.

Day 4 - Firewall Configuration and Traffic Filtering Report

```
(kondiba@kali)-[~]  
$ nmap -p 23 127.0.0.1  
  
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-08 11:09 IST  
Nmap scan report for localhost (127.0.0.1)  
Host is up (0.00019s latency).  
  
PORT      STATE  SERVICE  
23/tcp    closed telnet  
  
Nmap done: 1 IP address (1 host up) scanned in 0.29 seconds
```

Testing SSH Port (Port 22) Before Allow Rule:

Port 22 was initially closed because no active SSH service was running. This demonstrates that port status also depends on service availability.

```
(kondiba@kali)-[~]  
$ nmap -p 22 127.0.0.1  
  
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-08 11:13 IST  
Nmap scan report for localhost (127.0.0.1)  
Host is up (0.00011s latency).  
  
PORT      STATE  SERVICE  
22/tcp    closed ssh  
  
Nmap done: 1 IP address (1 host up) scanned in 0.14 seconds
```

Allowing SSH (Port 22):

A firewall rule was added to allow incoming traffic on port 22 for SSH. This is crucial for secure remote administration of the system.

Day 4 - Firewall Configuration and Traffic Filtering Report

```
(kondiba@kali)-[~]
$ sudo ufw allow 22/tcp
sudo ufw status numbered

Rule added
Rule added (v6)
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)

(kondiba@kali)-[~]
$
```

Why Port 22 Showed Closed Initially:

Even after adding the allow rule for port 22, it remained closed until the SSH service was started. This highlights that firewall rules alone cannot open a port without an active service.

```
kondiba@kali: ~

(kondiba@kali)-[~]
$ sudo systemctl start ssh
sudo systemctl enable ssh

Synchronizing state of ssh.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable ssh
Created symlink '/etc/systemd/system/ssh.service' → '/usr/lib/systemd/system/ssh.service'.
Created symlink '/etc/systemd/system/multi-user.target.wants/ssh.service' → '/usr/lib/systemd/system/ssh.service'.

(kondiba@kali)-[~]
$ nmap -p 22 127.0.0.1

Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-08 11:20 IST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00013s latency).

PORT      STATE SERVICE
22/tcp    open  ssh

Nmap done: 1 IP address (1 host up) scanned in 0.14 seconds

(kondiba@kali)-[~]
$
```

Removing Firewall Rules:

The allow rule for port 22 and the deny rule for port 23 were deleted, restoring the firewall to its default state for these ports.

Day 4 - Firewall Configuration and Traffic Filtering Report

```
Aug 8 11:31
kondiba@kali: ~
(kondiba@kali)~$ sudo ufw delete 1
sudo ufw delete 2
Deleting:
deny 23/tcp
Proceed with operation (y/n)? y
Rule deleted
Deleting:
deny 23/tcp
Proceed with operation (y/n)? y
Rule deleted (v6)

(kondiba@kali)~$ sudo ufw status numbered
Status: active

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

(kondiba@kali)~$ sudo ufw delete 1
sudo ufw delete 2
Deleting:
allow 22/tcp
Proceed with operation (y/n)? y
Rule deleted
ERROR: Could not find rule '2'

(kondiba@kali)~$ y
y: command not found
```

Summary

Through this task, we learned:

1. How to check firewall status and enable it.
2. Adding rules to block insecure services like Telnet (port 23).
3. Allowing secure services like SSH (port 22) for remote access.
4. The importance of service availability in port scanning results.
5. How to remove rules to revert to the original firewall state.

Outcome: Developed basic firewall management skills and understood the role of firewalls in controlling network traffic.