NAME: Kadiyala Sri Pavan

UNIVERSITY: Aditya University

EMAIL: pavancc1478@gmail.com

Task-10

Firewall Configuration & Testing

Objective:

To understand basic firewall concepts and gain practical experience in configuring firewall rules using UFW, including allowing and denying ports, testing connectivity, observing logs, blocking malicious IP addresses, and improving overall system security.

1. Learn firewall concepts.

A firewall is a network security system, available as hardware or software, that monitors and controls incoming and outgoing traffic based on predefined rules. It acts like a security guard, filtering data packets to either:

- **Accept:** Allow the traffic.
- **Reject:** Block with an error response.
- **Drop:** Block silently without response.

Importance of Firewalls

A firewall is the first line of defense in cybersecurity, acting as a security barrier between internal systems and external networks. It forces all traffic through a single checkpoint, where data packets are monitored, filtered, and either allowed or blocked based on predefined rules. Firewalls are essential because they:

- **Prevent Unauthorized Access:** Like a locked door with a guard, only trusted users and traffic are allowed through.
- **Block Malicious Traffic:** Harmful data such as viruses, phishing attempts, or denial-of-service (DoS) attacks are stopped before reaching the system.
- **Protect Sensitive Information:** Safeguards personal and business data from theft or accidental leaks.
- **Control Network Usage:** Enforces policies such as parental controls, workplace restrictions, or government filtering.
- **Mitigate Insider Risks:** Detects suspicious applications or data exfiltration attempts from within the network.

By combining prevention, monitoring, and control, firewalls provide customizable and comprehensive protection against both external and internal threats.

Types of Firewall

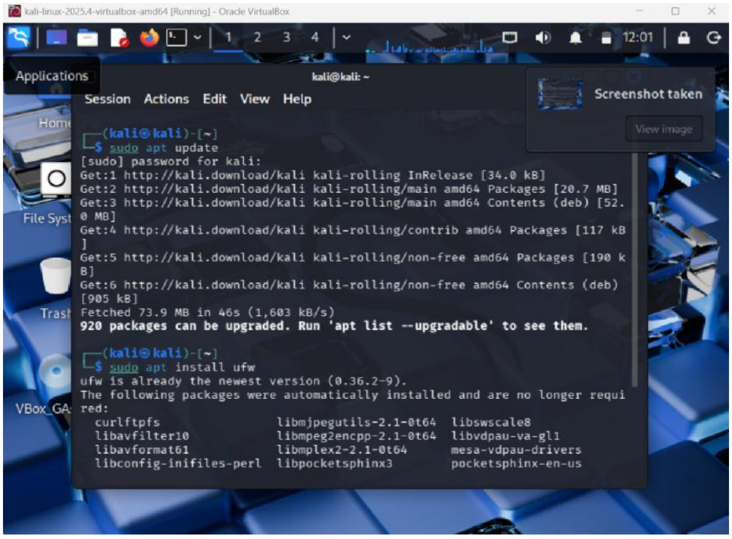
- **Host-Based Firewall**
  A firewall installed on an individual computer.
  Example: UFW, Windows Firewall.
- **Network-Based Firewall**
  A firewall placed between network and internet.
  Usually runs on router or hardware device.
- **Packet Filtering Firewall**
  Checks packets based on IP, port, and protocol.
  Allows or blocks traffic using simple rules.
- **Stateful Inspection Firewall**
  Tracks active connections and allows related traffic.
  More secure than packet filtering firewall.
- **Application Firewall**

- Filters traffic based on application data.
  Protects web applications from attacks.
- **Next-Generation Firewall (NGFW)**
  Advanced firewall with IDS/IPS and deep packet inspection.
  Used in enterprise networks.

2. Check & Install UFW.

sudo apt update

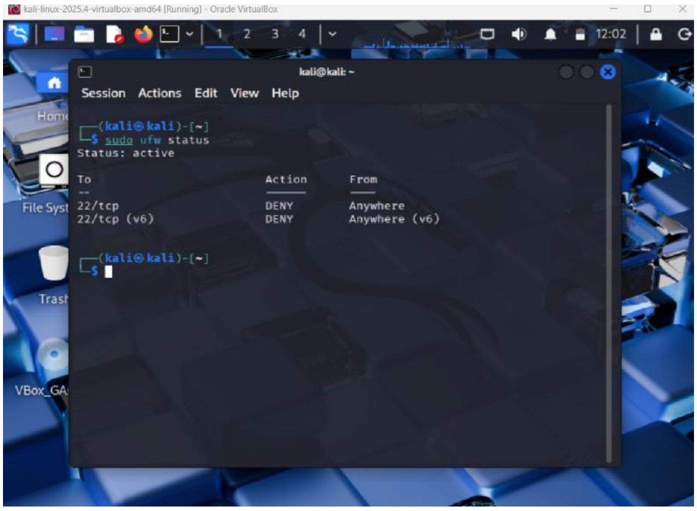
sudo apt install ufw


kali-linux-2025.4-virtualbox-amd64 [Running] - Oracle VirtualBox
1 2 3 4
12:01
Applications
kali@kali: ~
Session Actions Edit View Help
Screenshot taken
View image
Home
File Syst
Trash
VBox_GA

```
┌──(kali㉿kali)-[~]
└─$ sudo apt update
[sudo] password for kali:
Get:1 http://kali.download/kali kali-rolling InRelease [34.0 kB]
Get:2 http://kali.download/kali kali-rolling/main amd64 Packages [20.7 MB]
Get:3 http://kali.download/kali kali-rolling/main amd64 Contents (deb) [52.
0 MB]
Get:4 http://kali.download/kali kali-rolling/contrib amd64 Packages [117 kB
]
Get:5 http://kali.download/kali kali-rolling/non-free amd64 Packages [190 k
B]
Get:6 http://kali.download/kali kali-rolling/non-free amd64 Contents (deb)
[905 kB]
Fetched 73.9 MB in 46s (1,603 kB/s)
920 packages can be upgraded. Run 'apt list --upgradable' to see them.

┌──(kali㉿kali)-[~]
└─$ sudo apt install ufw
ufw is already the newest version (0.36.2-9).
The following packages were automatically installed and are no longer requi
red:
  curlftpfs              libmjpegutils-2.1-0t64  libswscale8
  libavfilter10          libmpeg2encpp-2.1-0t64  libvdpau-va-gl1
  libavformat61          libmplex2-2.1-0t64      mesa-vdpau-drivers
  libconfig-inifiles-perl libpocketsphinx3       pocketsphinx-en-us
```

Then check the status:

sudo ufw status

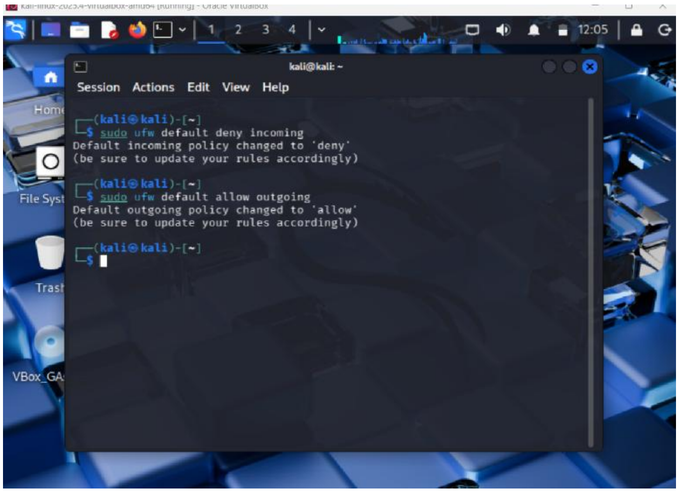

kali-linux-2025.4-virtualbox-amd64 [Running] - Oracle VirtualBox
1 2 3 4
12:02
kali@kali: ~
Session Actions Edit View Help
Home
File Syst
Trash
VBox_GA

```
┌──(kali㉿kali)-[~]
└─$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
22/tcp                     DENY        Anywhere
22/tcp (v6)                DENY        Anywhere (v6)


┌──(kali㉿kali)-[~]
└─$ 
```

3. Set Default Policies

sudo ufw default deny incoming

sudo ufw default allow outgoing



```
┌──(kali㉿kali)-[~]
└─$ sudo ufw default deny incoming
Default incoming policy changed to 'deny'
(be sure to update your rules accordingly)

┌──(kali㉿kali)-[~]
└─$ sudo ufw default allow outgoing
Default outgoing policy changed to 'allow'
(be sure to update your rules accordingly)

┌──(kali㉿kali)-[~]
└─$ 
```

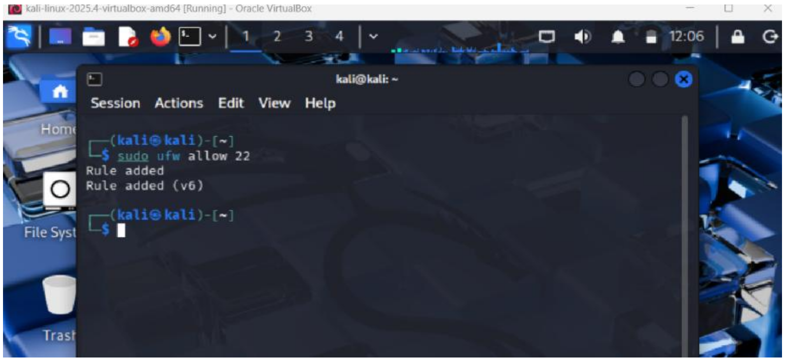
Meaning:

- Block all incoming
- Allow all outgoing

4. Allow Important Ports

Allow SSH:

sudo ufw allow 22

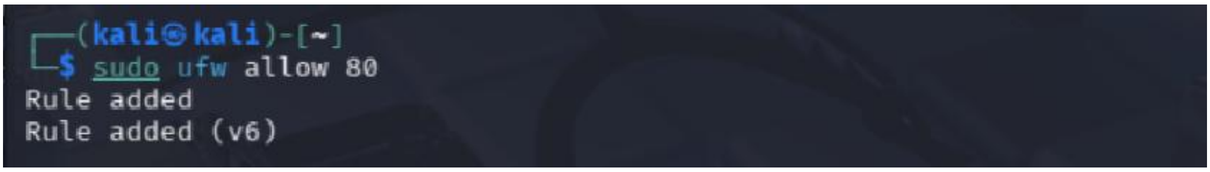


kali-linux-2025.4-virtualbox-amd64 [Running] - Oracle VirtualBox
12:06
kali@kali: ~
Session
Actions
Edit
View
Help
Home
File Syst
Trash

```
┌──(kali㉿kali)-[~]
└─$ sudo ufw allow 22
Rule added
Rule added (v6)

┌──(kali㉿kali)-[~]
└─$ 
```

Allow HTTP:

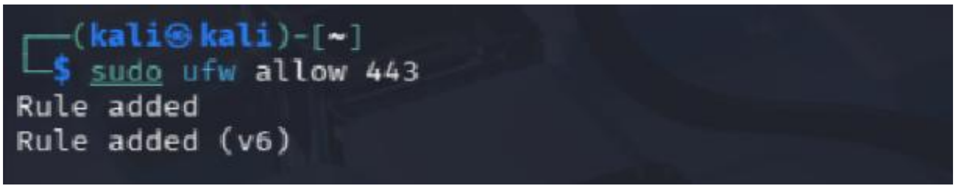
Sudo ufw allow 80



```
┌──(kali㉿kali)-[~]
└─$ sudo ufw allow 80
Rule added
Rule added (v6)
```

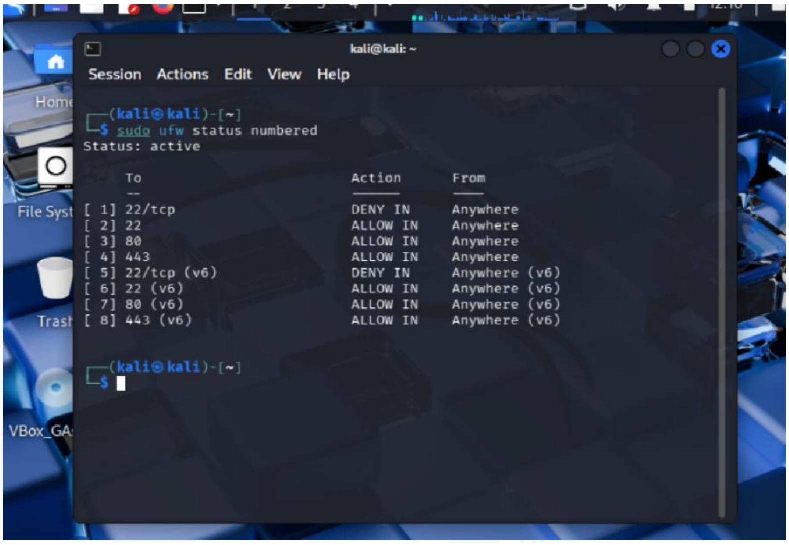
Allow HTTPS:

Sudo ufw allow 443



```
┌──(kali㉿kali)-[~]
└─$ sudo ufw allow 443
Rule added
Rule added (v6)
```

Now we check rules:

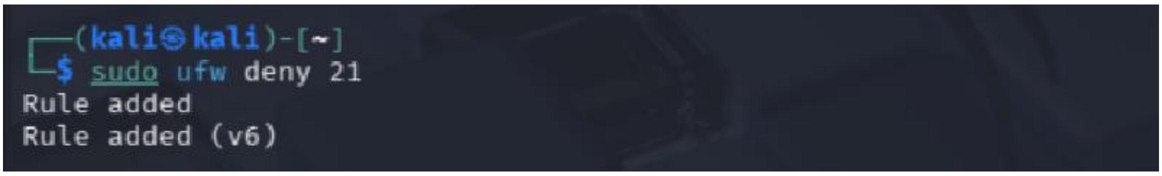


```
┌──(kali㉿kali)-[~]
└─$ sudo ufw status numbered
Status: active

     To                         Action      From
     --                         ------      ----
[ 1] 22/tcp                     DENY IN     Anywhere
[ 2] 22                         ALLOW IN    Anywhere
[ 3] 80                         ALLOW IN    Anywhere
[ 4] 443                        ALLOW IN    Anywhere
[ 5] 22/tcp (v6)                DENY IN     Anywhere (v6)
[ 6] 22 (v6)                    ALLOW IN    Anywhere (v6)
[ 7] 80 (v6)                    ALLOW IN    Anywhere (v6)
[ 8] 443 (v6)                   ALLOW IN    Anywhere (v6)


┌──(kali㉿kali)-[~]
└─$ 
```

5. Deny a Port

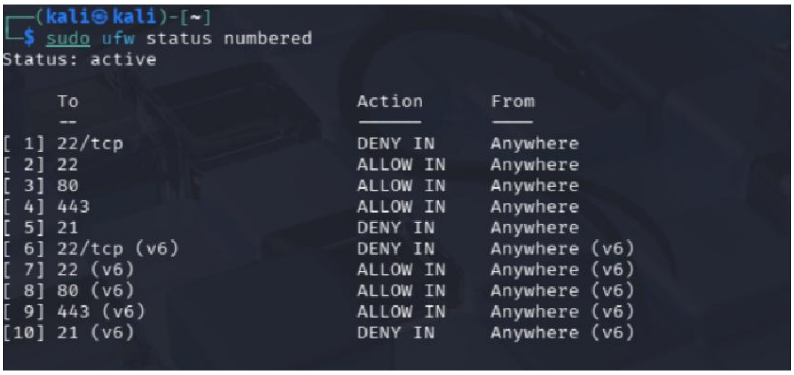
Block FTP (port 21):

sudo ufw deny 21



```
┌──(kali㉿kali)-[~]
└─$ sudo ufw deny 21
Rule added
Rule added (v6)
```

Again check rules:

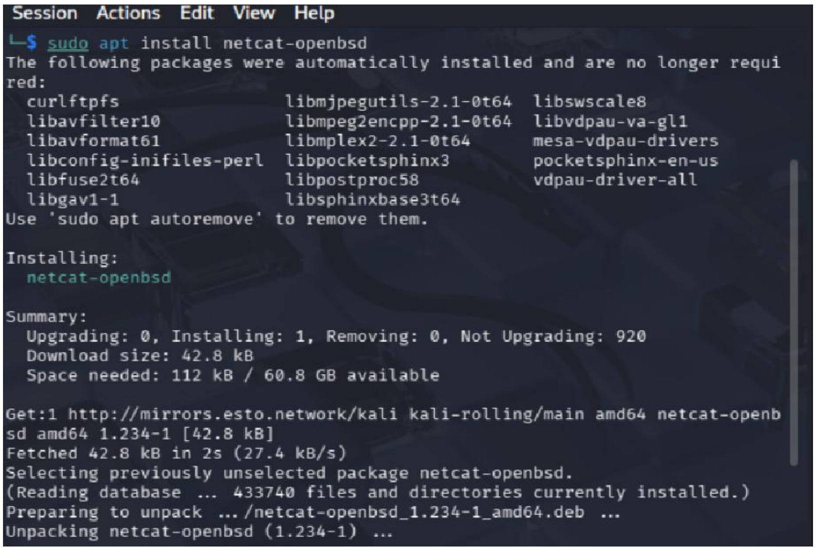


```
┌──(kali㉿kali)-[~]
└─$ sudo ufw status numbered
Status: active

     To                         Action      From
     --                         ------      ----
[ 1] 22/tcp                     DENY IN     Anywhere
[ 2] 22                         ALLOW IN    Anywhere
[ 3] 80                         ALLOW IN    Anywhere
[ 4] 443                        ALLOW IN    Anywhere
[ 5] 21                         DENY IN     Anywhere
[ 6] 22/tcp (v6)                DENY IN     Anywhere (v6)
[ 7] 22 (v6)                    ALLOW IN    Anywhere (v6)
[ 8] 80 (v6)                    ALLOW IN    Anywhere (v6)
[ 9] 443 (v6)                   ALLOW IN    Anywhere (v6)
[10] 21 (v6)                    DENY IN     Anywhere (v6)
```

6. Test Connectivity:

Install netcat:

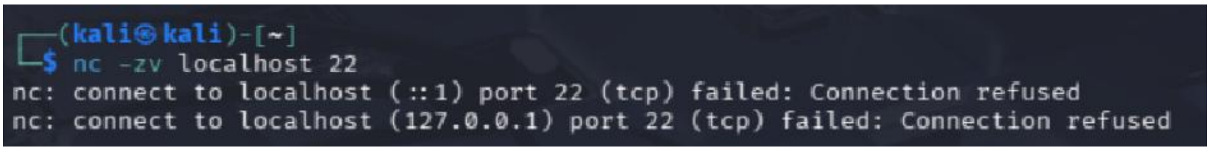
sudo apt install netcat-openbsd



```
└─$ sudo apt install netcat-openbsd
The following packages were automatically installed and are no longer requi
red:
  curlftpfs                libmjpegutils-2.1-0t64  libswscale8
  libavfilter10            libmpeg2encpp-2.1-0t64  libvdpau-va-gl1
  libavformat61            libmplex2-2.1-0t64      mesa-vdpau-drivers
  libconfig-inifiles-perl  libpocketsphinx3        pocketsphinx-en-us
  libfuse2t64              libpostproc58           vdpau-driver-all
  libgav1-1                libsphinxbase3t64
Use 'sudo apt autoremove' to remove them.

Installing:
  netcat-openbsd

Summary:
  Upgrading: 0, Installing: 1, Removing: 0, Not Upgrading: 920
  Download size: 42.8 kB
  Space needed: 112 kB / 60.8 GB available

Get:1 http://mirrors.esto.network/kali kali-rolling/main amd64 netcat-openb
sd amd64 1.234-1 [42.8 kB]
Fetched 42.8 kB in 2s (27.4 kB/s)
Selecting previously unselected package netcat-openbsd.
(Reading database ... 433740 files and directories currently installed.)
Preparing to unpack .../netcat-openbsd_1.234-1_amd64.deb ...
Unpacking netcat-openbsd (1.234-1) ...
```

Test allowed port:

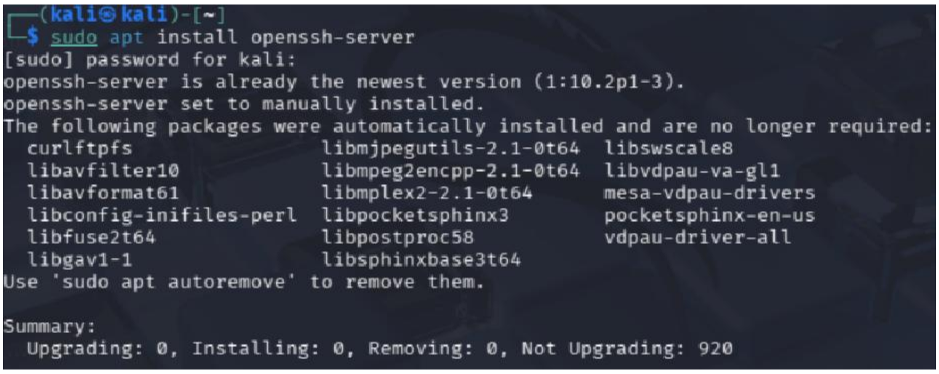
nc -zv localhost 22



```
┌──(kali㉿kali)-[~]
└─$ nc -zv localhost 22
nc: connect to localhost (::1) port 22 (tcp) failed: Connection refused
nc: connect to localhost (127.0.0.1) port 22 (tcp) failed: Connection refused
```

Here, SSH service is missing them we have to install it.

sudo apt install openssh-server

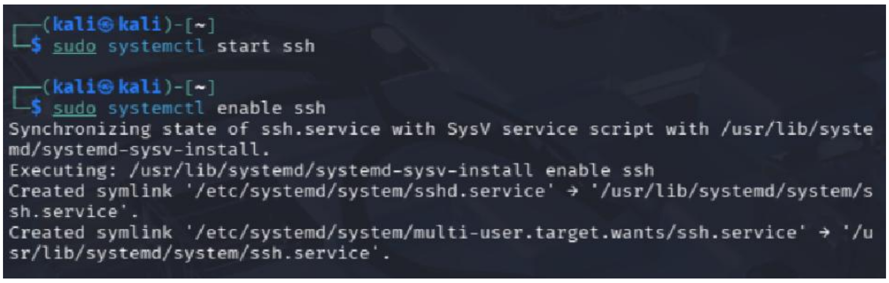


```
┌──(kali㉿kali)-[~]
└─$ sudo apt install openssh-server
[sudo] password for kali:
openssh-server is already the newest version (1:10.2p1-3).
openssh-server set to manually installed.
The following packages were automatically installed and are no longer required:
  curlftpfs              libmjpegutils-2.1-0t64  libswscale8
  libavfilter10          libmpeg2encpp-2.1-0t64  libvdpau-va-gl1
  libavformat61          libmplex2-2.1-0t64      mesa-vdpau-drivers
  libconfig-inifiles-perl libpocketsphinx3       pocketsphinx-en-us
  libfuse2t64            libpostproc58           vdpau-driver-all
  libgav1-1              libsphinxbase3t64
Use 'sudo apt autoremove' to remove them.

Summary:
  Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 920
```

- Start SSH Service and Enable SSH at Boot

sudo systemctl start ssh

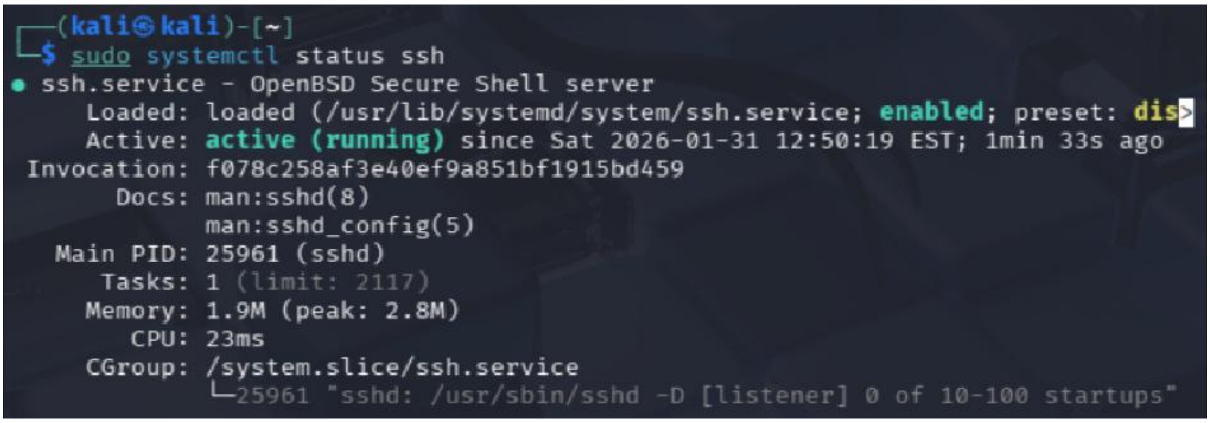
sudo systemctl enable ssh



```
┌──(kali㉿kali)-[~]
└─$ sudo systemctl start ssh

┌──(kali㉿kali)-[~]
└─$ sudo systemctl enable ssh
Synchronizing state of ssh.service with SysV service script with /usr/lib/syste
md/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable ssh
Created symlink '/etc/systemd/system/sshd.service' → '/usr/lib/systemd/system/s
sh.service'.
Created symlink '/etc/systemd/system/multi-user.target.wants/ssh.service' → '/u
sr/lib/systemd/system/ssh.service'.
```

Check Status:

sudo systemctl status ssh



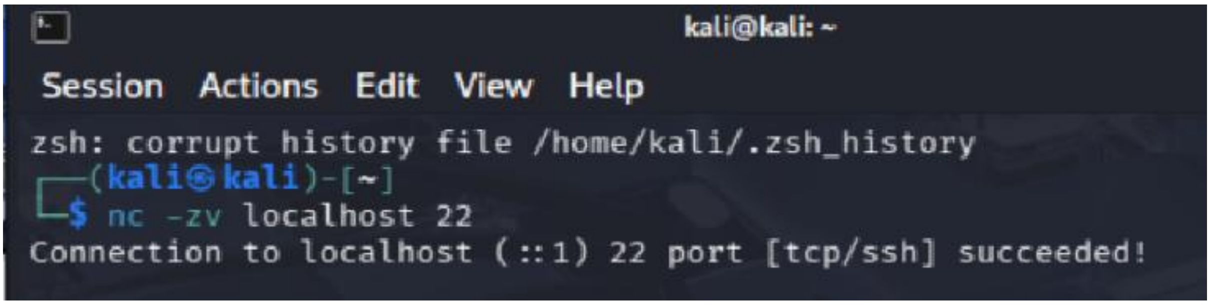
```
┌──(kali㉿kali)-[~]
└─$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled; preset: dis>
     Active: active (running) since Sat 2026-01-31 12:50:19 EST; 1min 33s ago
 Invocation: f078c258af3e40ef9a851bf1915bd459
       Docs: man:sshd(8)
             man:sshd_config(5)
   Main PID: 25961 (sshd)
      Tasks: 1 (limit: 2117)
     Memory: 1.9M (peak: 2.8M)
        CPU: 23ms
     CGroup: /system.slice/ssh.service
             └─25961 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"
```

You should see:

active (running)

Now, again Test Allowed Port

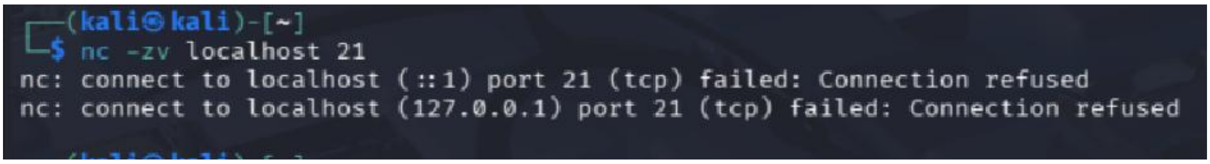
nc -zv localhost 22



```
zsh: corrupt history file /home/kali/.zsh_history
┌──(kali㉿kali)-[~]
└─$ nc -zv localhost 22
Connection to localhost (::1) 22 port [tcp/ssh] succeeded!
```

STEP 6: Test Blocked Port:

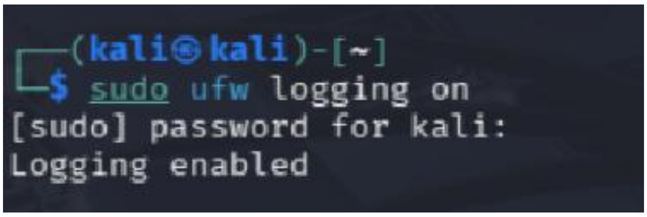
nc -zv localhost 21



```
┌──(kali㉿kali)-[~]
└─$ nc -zv localhost 21
nc: connect to localhost (::1) port 21 (tcp) failed: Connection refused
nc: connect to localhost (127.0.0.1) port 21 (tcp) failed: Connection refused
```

STEP 7: Enable Firewall Logging

sudo ufw logging on



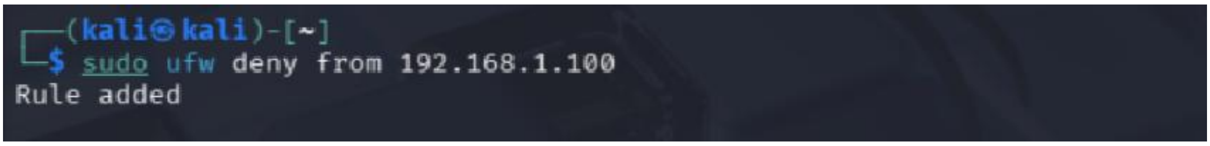
```
┌──(kali㉿kali)-[~]
└─$ sudo ufw logging on
[sudo] password for kali:
Logging enabled
```

STEP 8: View Logs

sudo tail -f /var/log/ufw.log

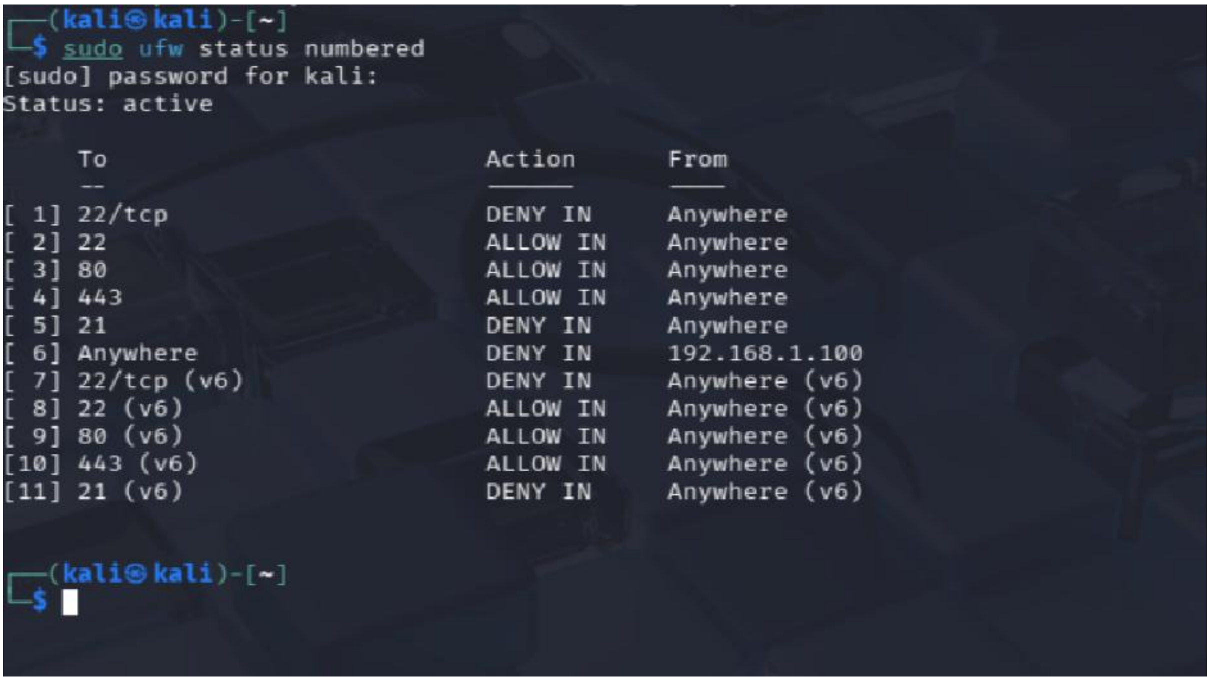
STEP 9: Block a Malicious IP

sudo ufw deny from 192.168.1.100



```
┌──(kali㉿kali)-[~]
└─$ sudo ufw deny from 192.168.1.100
Rule added
```

STEP 10: Save Documentation



```
┌──(kali㉿kali)-[~]
└─$ sudo ufw status numbered
[sudo] password for kali:
Status: active

     To                         Action      From
     --                         ------      ----
[ 1] 22/tcp                     DENY IN     Anywhere
[ 2] 22                         ALLOW IN    Anywhere
[ 3] 80                         ALLOW IN    Anywhere
[ 4] 443                        ALLOW IN    Anywhere
[ 5] 21                         DENY IN     Anywhere
[ 6] Anywhere                   DENY IN     192.168.1.100
[ 7] 22/tcp (v6)                DENY IN     Anywhere (v6)
[ 8] 22 (v6)                    ALLOW IN    Anywhere (v6)
[ 9] 80 (v6)                    ALLOW IN    Anywhere (v6)
[10] 443 (v6)                   ALLOW IN    Anywhere (v6)
[11] 21 (v6)                    DENY IN     Anywhere (v6)

┌──(kali㉿kali)-[~]
└─$ 
```

Conclusion

The firewall was successfully configured using UFW. Security rules were applied to allow essential services while blocking insecure ports and malicious sources. Connectivity testing confirmed correct rule behavior, and logging enabled effective monitoring.