**Task 4 : Setup and Use a Firewall on Windows/Linux.**

Step 1 :-

Check if UFW is installed.

```
Session  Actions  Edit  View  Help

┌──(root㉿kali)-[/home/kali/Downloads]
└─# ufw status
Command 'ufw' not found, but can be installed with:
apt install ufw
Do you want to install it? (N/y)
```

```
┌──(root㉿kali)-[/home/kali/Downloads]
└─# apt install ufw -y
Installing:
  ufw

Suggested packages:
  rsyslog

Summary:
  Upgrading: 0, Installing: 1, Removing: 0, Not Upgrading: 873
  Download size: 169 kB
  Space needed: 880 kB / 54.3 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 (134 kB/s)
Preconfiguring packages ...
Selecting previously unselected package ufw.
(Reading database ... 417410 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 (2025.3.2) ...
Processing triggers for man-db (2.13.1-1) ...
```

Step 2 :-

Active the firewall.

```
┌──(root㉿kali)-[/home/kali/Downloads]
└─# ufw enable
Firewall is active and enabled on system startup
```

Step 3 :-

Let’s see the current firewall rules.

```
┌──(root㉿kali)-[/home/kali/Downloads]
└─# ufw status numbered

Status: active

     To                         Action      From
     --                         ------      ----
[ 1] 23/tcp                     ALLOW IN    Anywhere
[ 2] 80/tcp                     ALLOW IN    Anywhere
[ 3] 23/tcp (v6)                ALLOW IN    Anywhere (v6)
[ 4] 80/tcp (v6)                ALLOW IN    Anywhere (v6)
```

Step 4 :-

Add new rule to deny or block.

```
┌──(root㉿kali)-[/home/kali/Downloads]
└─# ufw status numbered
Status: active

     To                         Action      From
     --                         ------      ----
[ 1] 23/tcp                     DENY IN     Anywhere
[ 2] 80/tcp                     ALLOW IN    Anywhere
[ 3] 23/tcp (v6)                DENY IN     Anywhere (v6)
[ 4] 80/tcp (v6)                ALLOW IN    Anywhere (v6)
```

Step 5 :-

Let's try to connect to the blocked port locally.

```
┌──(root㉿kali)-[/home/kali/Downloads]
└─# telnet localhost 23
Trying ::1...
Connection failed: Connection refused
Trying 127.0.0.1...
telnet: Unable to connect to remote host: Connection refused

┌──(root㉿kali)-[/home/kali/Downloads]
└─# nc -zv localhost 23
localhost [127.0.0.1] 23 (telnet) : Connection refused
```

Step 6 :-

Let's allow the SSH port – 22 as new rule.

```
┌──(root㉿kali)-[/home/kali/Downloads]
└─# ufw allow 22/tcp
Rule added
Rule added (v6)

┌──(root㉿kali)-[/home/kali/Downloads]
└─# ufw status numbered
Status: active

     To                         Action      From
     --                         ------      ----
[ 1] 23/tcp                     DENY IN     Anywhere
[ 2] 80/tcp                     ALLOW IN    Anywhere
[ 3] 22/tcp                     ALLOW IN    Anywhere
[ 4] 23/tcp (v6)                DENY IN     Anywhere (v6)
[ 5] 80/tcp (v6)                ALLOW IN    Anywhere (v6)
[ 6] 22/tcp (v6)                ALLOW IN    Anywhere (v6)
```

Step 7 :-

Remove the test block rule to restore original state.

```
┌──(root㉿kali)-[/home/kali/Downloads]
└─# ufw status numbered
Status: active

     To                         Action      From
     --                         ------      ----
[ 1] 23/tcp                     DENY IN     Anywhere
[ 2] 80/tcp                     ALLOW IN    Anywhere
[ 3] 22/tcp                     ALLOW IN    Anywhere
[ 4] 23/tcp (v6)                DENY IN     Anywhere (v6)
[ 5] 80/tcp (v6)                ALLOW IN    Anywhere (v6)
[ 6] 22/tcp (v6)                ALLOW IN    Anywhere (v6)


┌──(root㉿kali)-[/home/kali/Downloads]
└─# ufw delete 1
Deleting:
 deny 23/tcp
Proceed with operation (y|n)? Y
Rule deleted

┌──(root㉿kali)-[/home/kali/Downloads]
└─# ufw status numbered
Status: active

     To                         Action      From
     --                         ------      ----
[ 1] 80/tcp                     ALLOW IN    Anywhere
[ 2] 22/tcp                     ALLOW IN    Anywhere
[ 3] 23/tcp (v6)                DENY IN     Anywhere (v6)
[ 4] 80/tcp (v6)                ALLOW IN    Anywhere (v6)
[ 5] 22/tcp (v6)                ALLOW IN    Anywhere (v6)


┌──(root㉿kali)-[/home/kali/Downloads]
└─# ufw delete 3
Deleting:
 deny 23/tcp
Proceed with operation (y|n)? y
Rule deleted (v6)

┌──(root㉿kali)-[/home/kali/Downloads]
└─# ufw status numbered
Status: active

     To                         Action      From
     --                         ------      ----
[ 1] 80/tcp                     ALLOW IN    Anywhere
[ 2] 22/tcp                     ALLOW IN    Anywhere
[ 3] 80/tcp (v6)                ALLOW IN    Anywhere (v6)
[ 4] 22/tcp (v6)                ALLOW IN    Anywhere (v6)
```

Step 8 :-

Document the used commands to configure firewall.

Note :- All this commands is runnable in root access only.

| Step | Description | Command |
|---|---|---|
| 1 | Enable UFW | ufw enable |
| 2 | List rules | ufw status numbered |
| 3 | Block Telnet (port 23) | ufw deny 23/tcp |
| 4 | Test rule | telnet localhost 23 or nc -zv localhost 23 |
| 5 | Allow SSH | ufw allow 22/tcp |
| 6 | Remove block rule | ufw delete deny 23/tcp |
| 7 | Verify changes | ufw status |

Step 9 :-

Summarize How Firewall Filters Traffic.

A firewall acts as a traffic filter between system (computer) and the network.
It checks all incoming and outgoing packet against a set of rules.

- Inbound traffic: traffic coming into your system from the network.
- Outbound traffic: traffic leaving your system.

Each rule defines:

- Port number (e.g., 22 for SSH, 80 for HTTP)
- Protocol (TCP or UDP)
- Action (ALLOW or DENY)

When a connection attempt occurs:

- If a packet matches an ALLOW rule → it's accepted.
- If it matches a DENY rule → it's dropped or rejected.
- If it matches nothing → the default policy applies (usually "deny incoming, allow outgoing").