

# Linux Security Hardening Project – By Priant Singh

This is a small security project I did on my Ubuntu virtual machine to make it safer. I used simple Linux commands to update the system, create a safer user, secure SSH, set up a firewall, install Fail2Ban, and check open ports. Here is everything I did and what each command means.

## **1. Updated and Upgraded the System**

### **Command:**

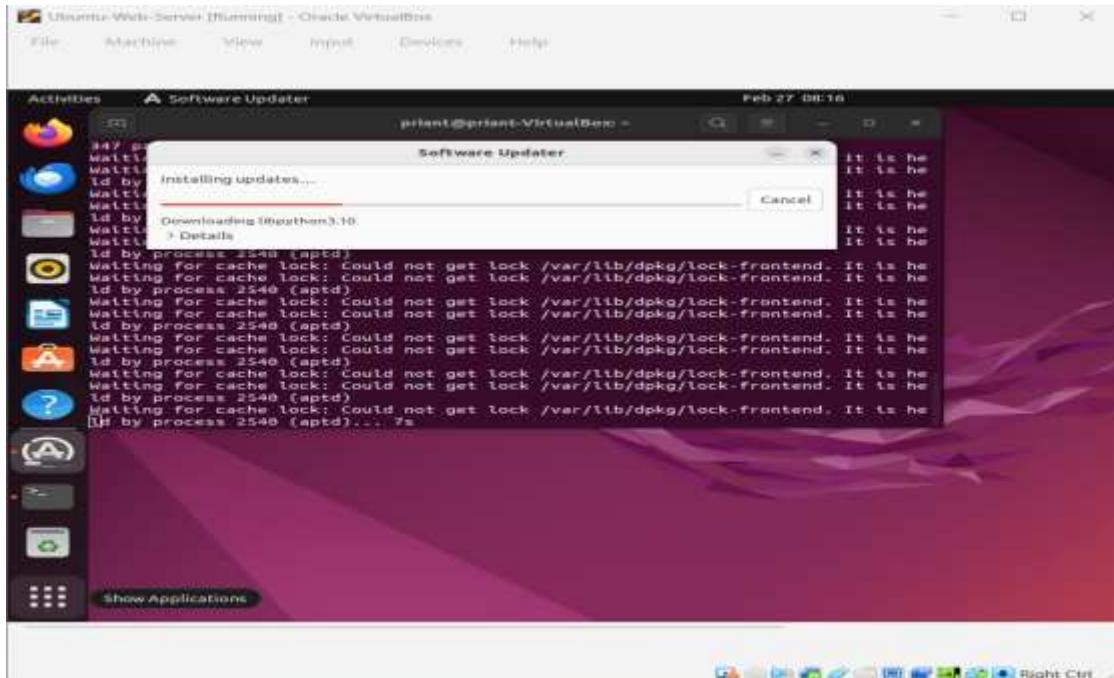
```
sudo apt update && sudo apt upgrade -y
```

### **What it does (simple explanation):**

- **apt update** checks for the latest updates available.
- **apt upgrade** installs the updates.
- **sudo** runs the command as admin.
- **-y** automatically answers “yes”.

### **Why I did this:**

Keeping the system updated is one of the easiest and most important security steps because it fixes known vulnerabilities.



## 2. Created a New Non-Root User (Least Privilege)

### Commands:

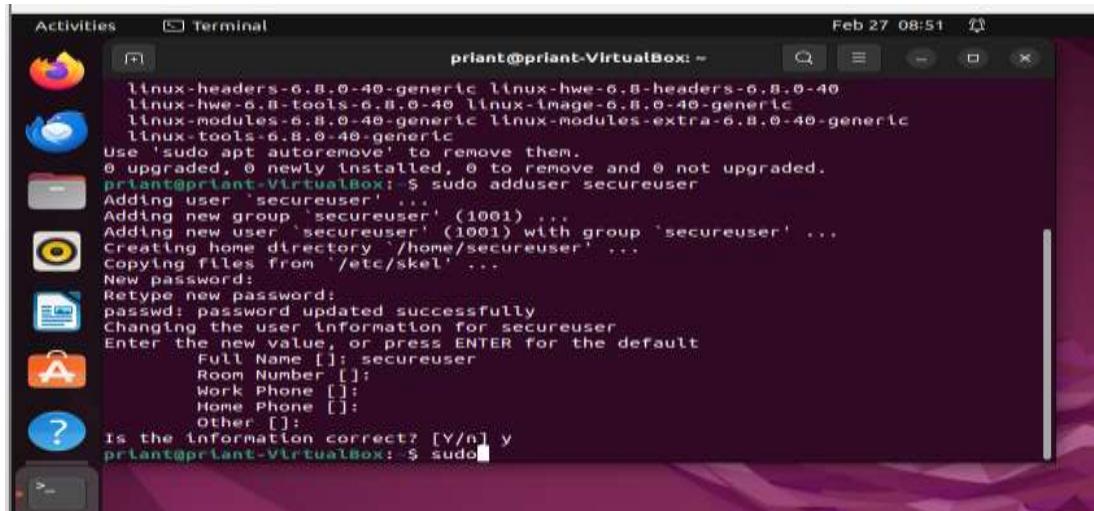
```
sudo adduser secureuser  
sudo usermod -aG sudo secureuser
```

### What they mean:

- **adduser secureuser** creates a new user named “secureuser”.
- **usermod -aG sudo secureuser** adds that user to the sudo (admin) group.

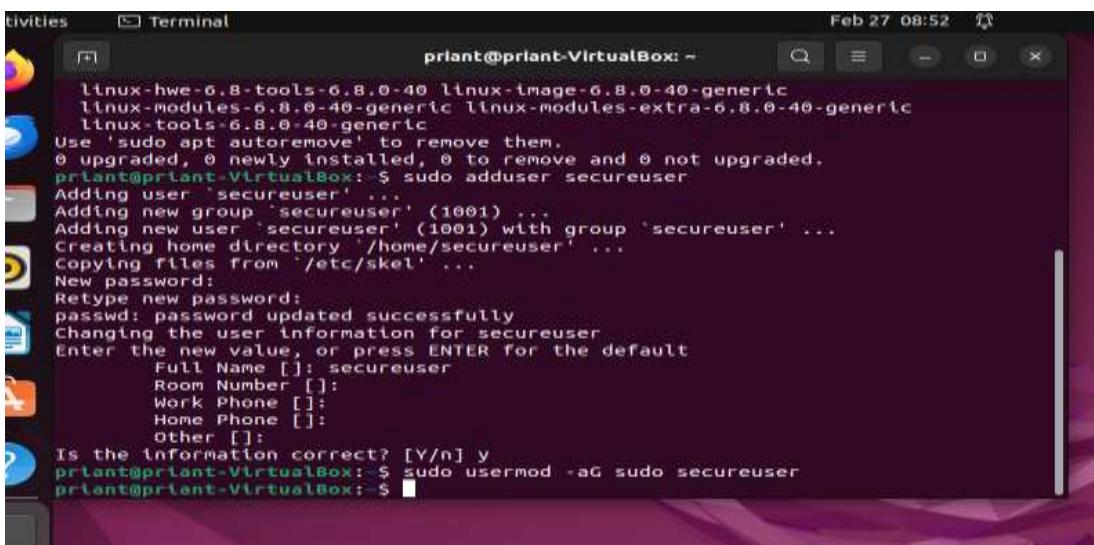
### Why I did this:

It is unsafe to use the root account for everything. Creating a normal user and only using sudo when needed reduces risk.



A screenshot of a Linux desktop environment showing a terminal window. The terminal window title is "Activities Terminal". The command entered is "sudo adduser secureuser". The output shows the user being added to the "secureuser" group and creating a home directory. It also asks for a new password and confirms it was updated successfully. Finally, it asks if the information is correct, and the user types "y". The timestamp in the top right corner is "Feb 27 08:51".

```
Activities Terminal Feb 27 08:51  
priant@priant-VirtualBox: ~  
priant@priant-VirtualBox: ~$ sudo adduser secureuser  
Adding user 'secureuser' ...  
Adding new group 'secureuser' (1001) ...  
Adding new user 'secureuser' (1001) with group 'secureuser' ...  
Creating home directory '/home/secureuser' ...  
Copying files from '/etc/skel' ...  
New password:  
Retype new password:  
passwd: password updated successfully  
Changing the user information for secureuser  
Enter the new value, or press ENTER for the default  
    Full Name []: secureuser  
    Room Number []:  
    Work Phone []:  
    Home Phone []:  
    Other []:  
Is the information correct? [Y/n] y  
priant@priant-VirtualBox: ~$
```



A screenshot of a Linux desktop environment showing a terminal window. The terminal window title is "Activities Terminal". The command entered is "sudo usermod -aG sudo secureuser". The output shows the user being added to the "sudo" group. The timestamp in the top right corner is "Feb 27 08:52".

```
Activities Terminal Feb 27 08:52  
priant@priant-VirtualBox: ~  
priant@priant-VirtualBox: ~$ sudo usermod -aG sudo secureuser  
priant@priant-VirtualBox: ~$
```

### 3. Installed OpenSSH Server

#### Command:

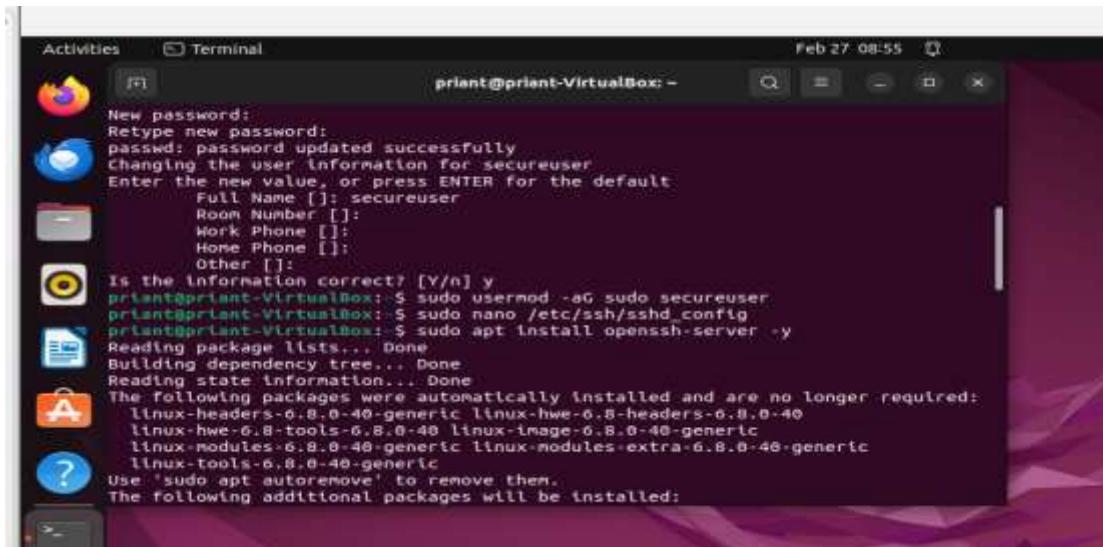
```
sudo apt install openssh-server -y
```

#### What it does:

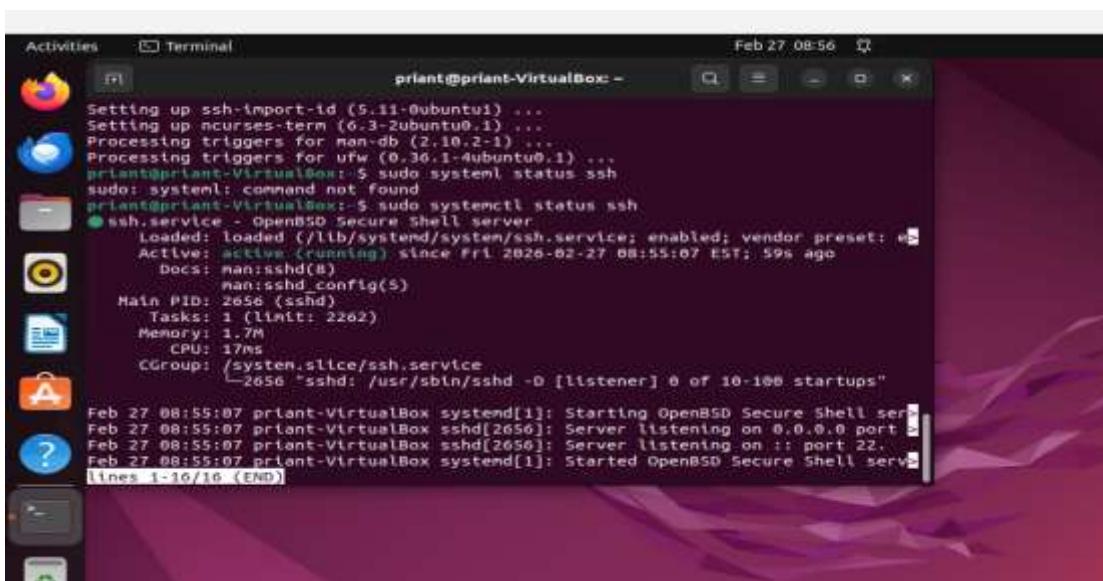
- Installs the SSH service so I can secure it.
- Creates the /etc/ssh/sshd\_config file.

#### Why I did this:

SSH allows remote access, so it must be installed before securing it.



```
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for secureuser
Enter the new value, or press ENTER for the default
  Full Name []:
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []:
Is the information correct? [Y/n] y
priant@priant-VirtualBox: ~$ sudo usermod -aG sudo secureuser
priant@priant-VirtualBox: ~$ sudo nano /etc/ssh/sshd_config
priant@priant-VirtualBox: ~$ sudo apt install openssh-server -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  linux-headers-6.8.0-40-generic linux-hwe-6.8-headers-6.8.0-40
  linux-hwe-6.8-tools-6.8.0-40 linux-image-6.8.0-40-generic
  linux-modules-6.8.0-40-generic linux-modules-extra-6.8.0-40-generic
  linux-tools-6.8.0-40-generic
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
```



```
Setting up ssh-import-id (5.11-0ubuntu1) ...
Setting up ncurses-term (6.3-2ubuntu0.1) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for ufw (0.30.1-4ubuntu0.1) ...
priant@priant-VirtualBox: ~$ sudo systemctl status ssh
sudo: systemctl: command not found
priant@priant-VirtualBox: ~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2026-02-27 08:55:07 EST; 59s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
     Main PID: 2656 (sshd)
        Tasks: 1 (limit: 2262)
       Memory: 1.7M
          CPU: 17ms
         CGroup: /system.slice/ssh.service
                   └─2656 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Feb 27 08:55:07 priant-VirtualBox systemd[1]: Starting OpenBSD Secure Shell server...
Feb 27 08:55:07 priant-VirtualBox sshd[2656]: Server listening on 0.0.0.0 port 22.
Feb 27 08:55:07 priant-VirtualBox sshd[2656]: Server listening on :: port 22.
Feb 27 08:55:07 priant-VirtualBox systemd[1]: Started OpenBSD Secure Shell server.
lines 1-16/16 (END)
```

## 4. Secured SSH by Disabling Root Login

**Open SSH config:**

```
sudo nano /etc/ssh/sshd_config
```

**Edited this line:**

Original:

```
#PermitRootLogin prohibit-password
```

Changed to:

```
PermitRootLogin no
```

**What this means:**

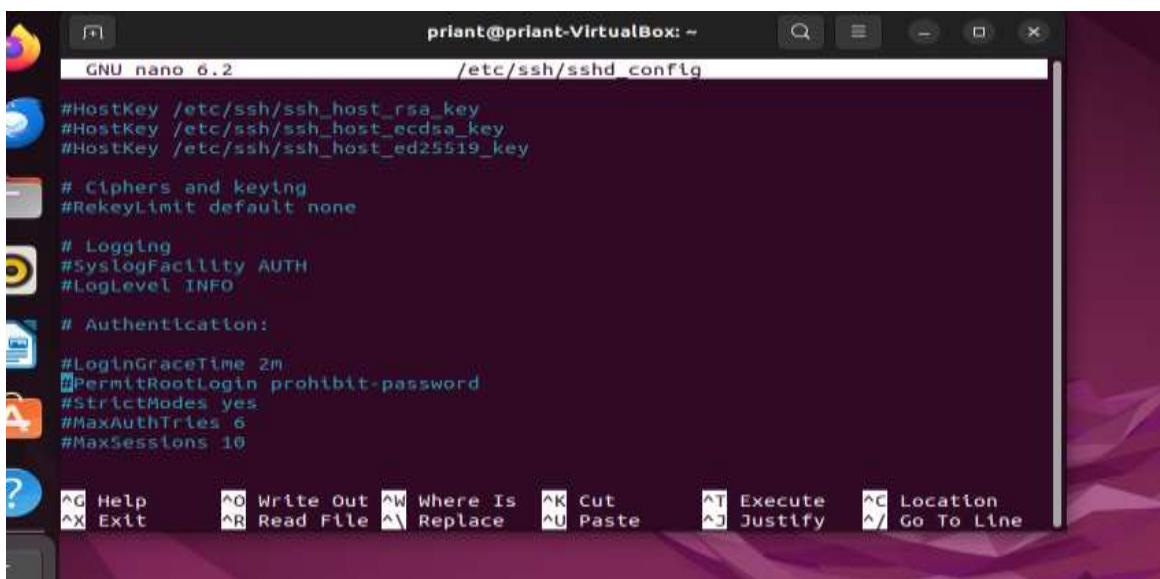
- **nano** opens the file for editing.
- Changing **PermitRootLogin no** stops anyone from logging in as root through SSH.

**Why I did this:**

Disabling root login is a big security improvement because attackers always try the “root” user first.

**Restarted SSH:**

```
sudo systemctl restart ssh
```



```
priant@priant-VirtualBox: ~
GNU nano 6.2          /etc/ssh/sshd_config

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
#PermitRootLogin prohibit-password
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

^C Help      ^O Write Out  ^W Where Is   ^K Cut      ^T Execute   ^C Location
^X Exit      ^R Read File  ^\ Replace   ^U Paste    ^J Justify   ^/ Go To Line
```

The screenshot shows two terminal windows side-by-side. The top window displays the contents of the file `/etc/ssh/sshd_config` using the `GNU nano 6.2` editor. The configuration includes host key settings, cipher and keying limits, logging levels, authentication methods (including password and public key), and session limits. The bottom window shows the output of several commands: `systemctl status ssh.service` (which shows the service is active and running with PID 2656), `sudo nano /etc/ssh/sshd_config`, `sudo systemctl restart ssh`, `sudo ufw status` (showing the firewall is inactive), and `sudo ufw enable` (enabling the firewall). The terminal interface includes standard Linux navigation keys like `Alt+F1` through `Alt+F6`.

```
GNU nano 6.2 /etc/ssh/sshd_config
#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keyeng
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
#PermitRootLogin no
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes

priant@priant-VirtualBox: ~
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2026-02-27 08:55:07 EST; 59s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
     Main PID: 2656 (sshd)
        Tasks: 1 (limit: 2262)
       Memory: 1.7M
          CPU: 17ms
        CGroup: /system.slice/ssh.service
                └─2656 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Feb 27 08:55:07 priant-VirtualBox systemd[1]: Starting OpenBSD Secure Shell server...
Feb 27 08:55:07 priant-VirtualBox sshd[2656]: Server listening on 0.0.0.0 port >
Feb 27 08:55:07 priant-VirtualBox sshd[2656]: Server listening on :: port 22.
Feb 27 08:55:07 priant-VirtualBox systemd[1]: Started OpenBSD Secure Shell server.

priant@priant-VirtualBox: $ sudo nano /etc/ssh/sshd_config
priant@priant-VirtualBox: $ sudo nano /etc/ssh/sshd_config
priant@priant-VirtualBox: $ sudo systemctl restart ssh
priant@priant-VirtualBox: $ sudo ufw status
Status: inactive
priant@priant-VirtualBox: $ sudo ufw enable
Firewall is active and enabled on system startup
priant@priant-VirtualBox: $
```

## 5. Enabled and Configured the Firewall (UFW)

### Commands:

```
sudo ufw enable
sudo ufw allow OpenSSH
sudo ufw allow 80
sudo ufw status
```

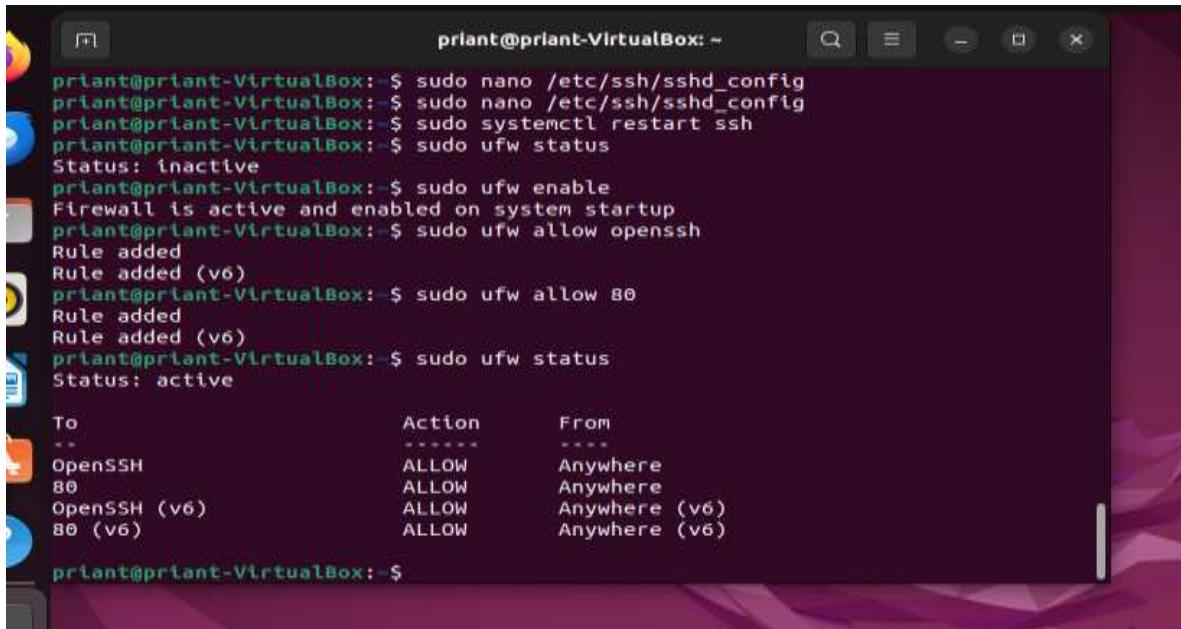
### What they mean:

- **ufw enable** turns on the firewall.

- **ufw allow OpenSSH** allows SSH so I don't lock myself out (port 22).
- **ufw allow 80** allows web traffic if I ever run a website.
- **ufw status** shows which ports are allowed.

### Why I did this:

A firewall blocks unwanted network traffic and only allows the ports I want open.



```

priant@priant-VirtualBox:~$ sudo nano /etc/ssh/sshd_config
priant@priant-VirtualBox:~$ sudo nano /etc/ssh/sshd_config
priant@priant-VirtualBox:~$ sudo systemctl restart ssh
priant@priant-VirtualBox:~$ sudo ufw status
Status: inactive
priant@priant-VirtualBox:~$ sudo ufw enable
Firewall is active and enabled on system startup
priant@priant-VirtualBox:~$ sudo ufw allow openssh
Rule added
Rule added (v6)
priant@priant-VirtualBox:~$ sudo ufw allow 80
Rule added
Rule added (v6)
priant@priant-VirtualBox:~$ sudo ufw status
Status: active

To                         Action      From
--                         --          --
OpenSSH                    ALLOW      Anywhere
80                         ALLOW      Anywhere
OpenSSH (v6)                ALLOW      Anywhere (v6)
80 (v6)                    ALLOW      Anywhere (v6)

priant@priant-VirtualBox:~$
```

## 6. Installed and Enabled Fail2Ban (Brute-Force Protection)

### Commands:

```

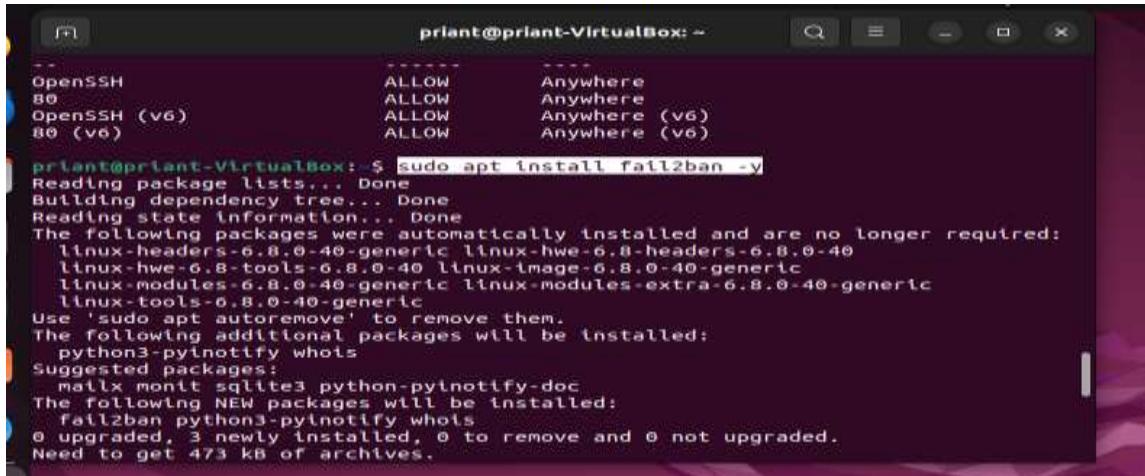
sudo apt install fail2ban -y
sudo systemctl start fail2ban
sudo systemctl enable fail2ban
sudo cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local
```

### What they mean:

- **fail2ban** watches for repeated failed login attempts.
- **start fail2ban** runs it immediately.
- **enable fail2ban** makes it start automatically on boot.
- **copying jail.conf** creates a safe editable config file.

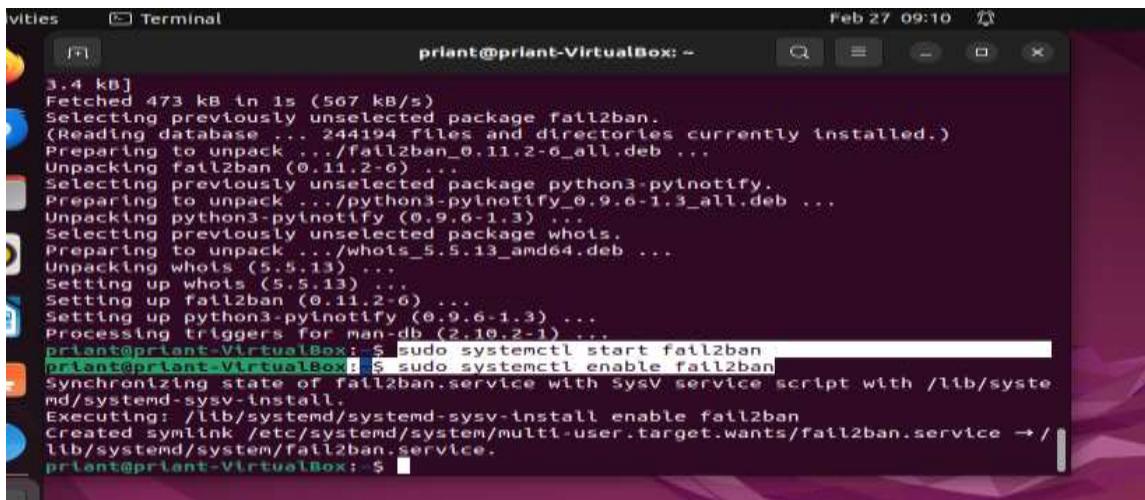
### Why I did this:

Fail2Ban automatically blocks IP addresses that try to brute-force passwords, adding an extra security layer.

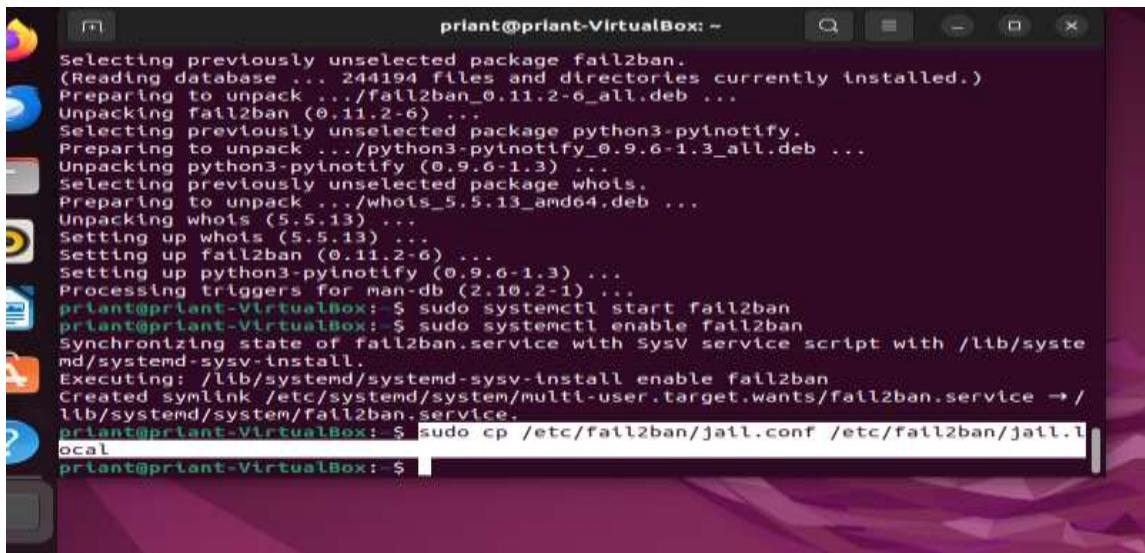


```
priant@priant-VirtualBox: ~
OpenSSH          ALLOW      Anywhere
80               ALLOW      Anywhere
OpenSSH (v6)     ALLOW      Anywhere (v6)
80 (v6)          ALLOW      Anywhere (v6)

priant@priant-VirtualBox: $ sudo apt install fail2ban -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  linux-headers-6.8.0-40-generic linux-hwe-6.8-headers-6.8.0-40
  linux-hwe-6.8-tools-6.8.0-40 linux-image-6.8.0-40-generic
  linux-modules-6.8.0-40-generic linux-modules-extra-6.8.0-40-generic
  linux-tools-6.8.0-40-generic
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  python3-pynotify whois
Suggested packages:
  mailx monit sqlite3 python-pynotify-doc
The following NEW packages will be installed:
  fail2ban python3-pynotify whois
0 upgraded, 3 newly installed, 0 to remove and 0 not upgraded.
Need to get 473 kB of archives.
```



```
3.4 kB]
Fetched 473 kB in 1s (567 kB/s)
Selecting previously unselected package fail2ban.
(Reading database ... 244194 files and directories currently installed.)
Preparing to unpack .../fail2ban_0.11.2-6_all.deb ...
Unpacking fail2ban (0.11.2-6) ...
Selecting previously unselected package python3-pynotify.
Preparing to unpack .../python3-pynotify_0.9.6-1.3_all.deb ...
Unpacking python3-pynotify (0.9.6-1.3) ...
Selecting previously unselected package whois.
Preparing to unpack .../whois_5.5.13_amd64.deb ...
Unpacking whois (5.5.13) ...
Setting up whois (5.5.13) ...
Setting up fail2ban (0.11.2-6) ...
Setting up python3-pynotify (0.9.6-1.3) ...
Processing triggers for man-db (2.10.2-1) ...
priant@priant-VirtualBox: $ sudo systemctl start fail2ban
priant@priant-VirtualBox: $ sudo systemctl enable fail2ban
Synchronizing state of fail2ban.service with SysV service script with /lib/systemd/system-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable fail2ban
Created symlink /etc/systemd/system/multi-user.target.wants/fail2ban.service → /lib/systemd/system/fail2ban.service.
priant@priant-VirtualBox: $
```



```
Selecting previously unselected package fail2ban.
(Reading database ... 244194 files and directories currently installed.)
Preparing to unpack .../fail2ban_0.11.2-6_all.deb ...
Unpacking fail2ban (0.11.2-6) ...
Selecting previously unselected package python3-pynotify.
Preparing to unpack .../python3-pynotify_0.9.6-1.3_all.deb ...
Unpacking python3-pynotify (0.9.6-1.3) ...
Selecting previously unselected package whois.
Preparing to unpack .../whois_5.5.13_amd64.deb ...
Unpacking whois (5.5.13) ...
Setting up whois (5.5.13) ...
Setting up fail2ban (0.11.2-6) ...
Setting up python3-pynotify (0.9.6-1.3) ...
Processing triggers for man-db (2.10.2-1) ...
priant@priant-VirtualBox: $ sudo systemctl start fail2ban
priant@priant-VirtualBox: $ sudo systemctl enable fail2ban
Synchronizing state of fail2ban.service with SysV service script with /lib/systemd/system-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable fail2ban
Created symlink /etc/systemd/system/multi-user.target.wants/fail2ban.service → /lib/systemd/system/fail2ban.service.
priant@priant-VirtualBox: $ sudo cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local
priant@priant-VirtualBox: $
```

## 7. Checked Open Ports (Network Exposure Check)

**Command:**

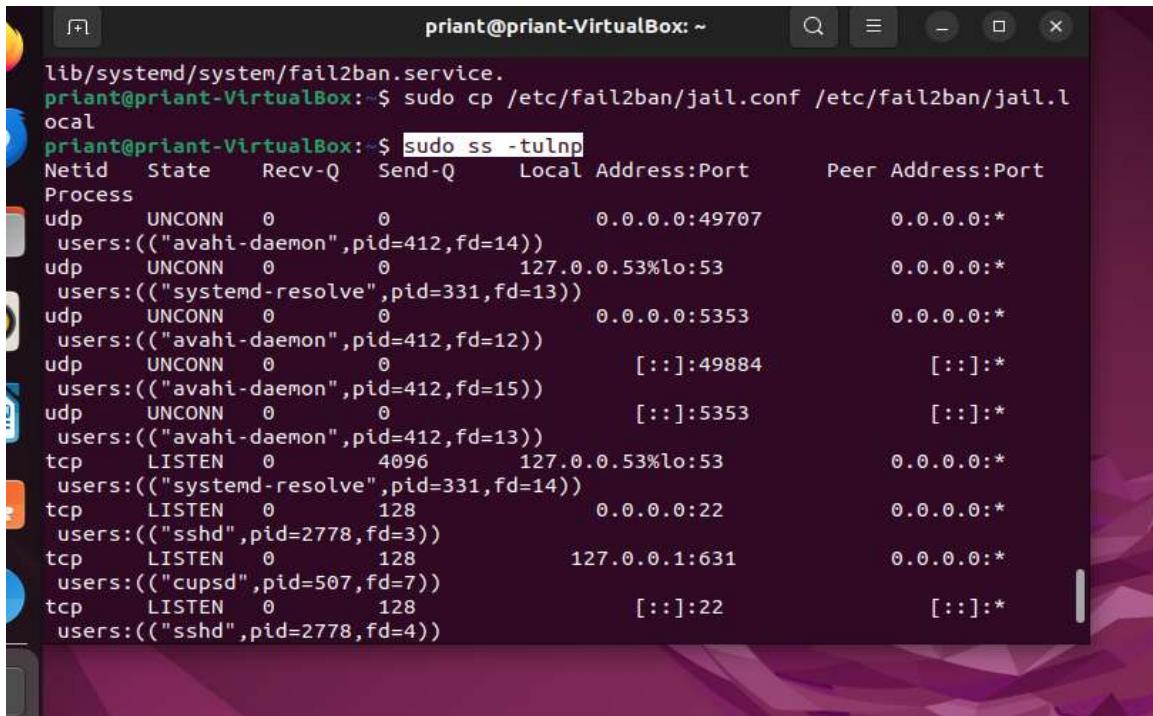
```
sudo ss -tulnp
```

**What it means:**

- **ss** shows open ports and network connections.
- **-t** shows TCP ports.
- **-u** shows UDP ports.
- **-l** shows listening ports.
- **-n** shows numeric output.
- **-p** shows which program is using the port.

**Why I did this:**

It helps me see exactly which services are visible to the network and if anything unnecessary is open.



```
priant@priant-VirtualBox:~$ sudo cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local
priant@priant-VirtualBox:~$ sudo ss -tulnp
Netid      State     Recv-Q    Send-Q      Local Address:Port      Peer Address:Port
Process
udp      UNCONN      0          0      0.0.0.0:49707      0.0.0.0:*
  users:(("avahi-daemon",pid=412,fd=14))
udp      UNCONN      0          0    127.0.0.53%lo:53      0.0.0.0:*
  users:(("systemd-resolve",pid=331,fd=13))
udp      UNCONN      0          0      0.0.0.0:5353      0.0.0.0:*
  users:(("avahi-daemon",pid=412,fd=12))
udp      UNCONN      0          0          [::]:49884      [::]:*
  users:(("avahi-daemon",pid=412,fd=15))
udp      UNCONN      0          0          [::]:5353      [::]:*
  users:(("avahi-daemon",pid=412,fd=13))
tcp      LISTEN      0        4096    127.0.0.53%lo:53      0.0.0.0:*
  users:(("systemd-resolve",pid=331,fd=14))
tcp      LISTEN      0         128      0.0.0.0:22      0.0.0.0:*
  users:(("sshd",pid=2778,fd=3))
tcp      LISTEN      0         128    127.0.0.1:631      0.0.0.0:*
  users:(("cupsd",pid=507,fd=7))
tcp      LISTEN      0         128          [::]:22      [::]:*
  users:(("sshd",pid=2778,fd=4))
```

## **Final Summary**

I completed a Linux Security Hardening mini project where I updated the system, created a safer user, disabled root SSH login, set up the firewall, installed Fail2Ban, and checked open ports. This helped me understand how to reduce attack surface, protect SSH access, and secure a Linux machine in a simple and hands-on way.