LAB- 02 System Hardening

By :- Faraz Ahmed

- We use “sudo apt update” to update all of the existing packages to latest available version from its official sources as shown in figure 1.

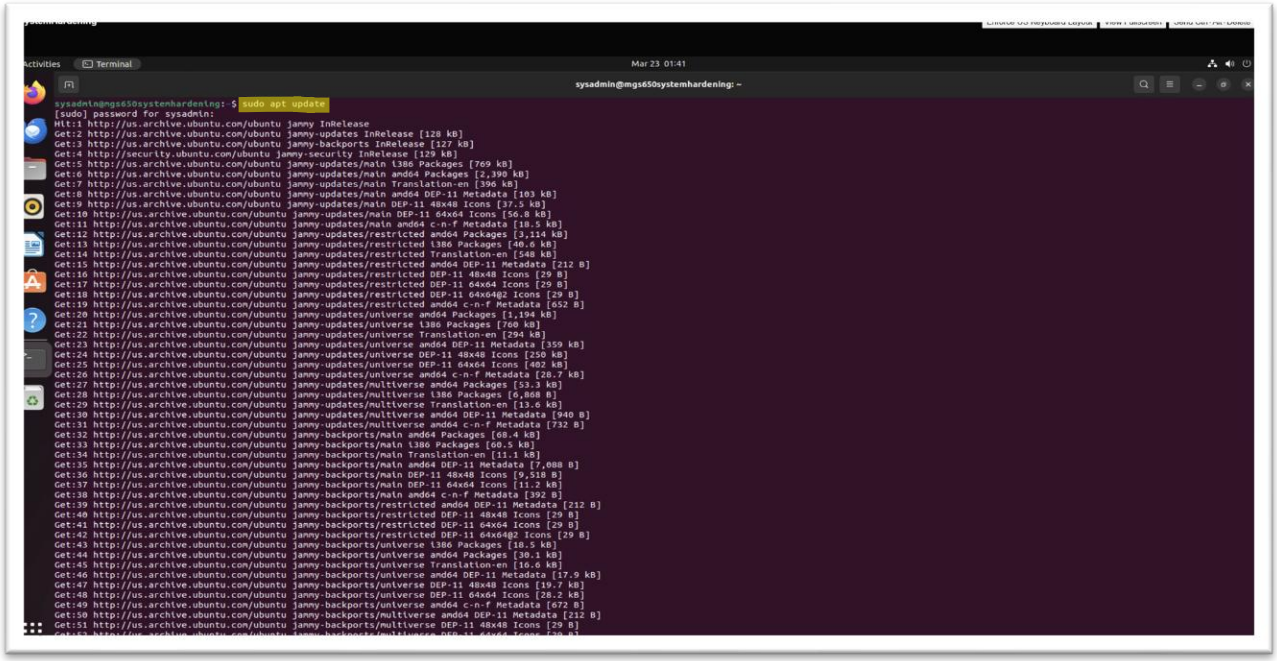

Figure 1: Screenshot of executing “sudo apt update” command to update all packages.

- This “sudo apt upgrade” command is similar to command used in figure 2 which means it updates installed packages to their latest available versions while keeping existing system configurations without any changes.

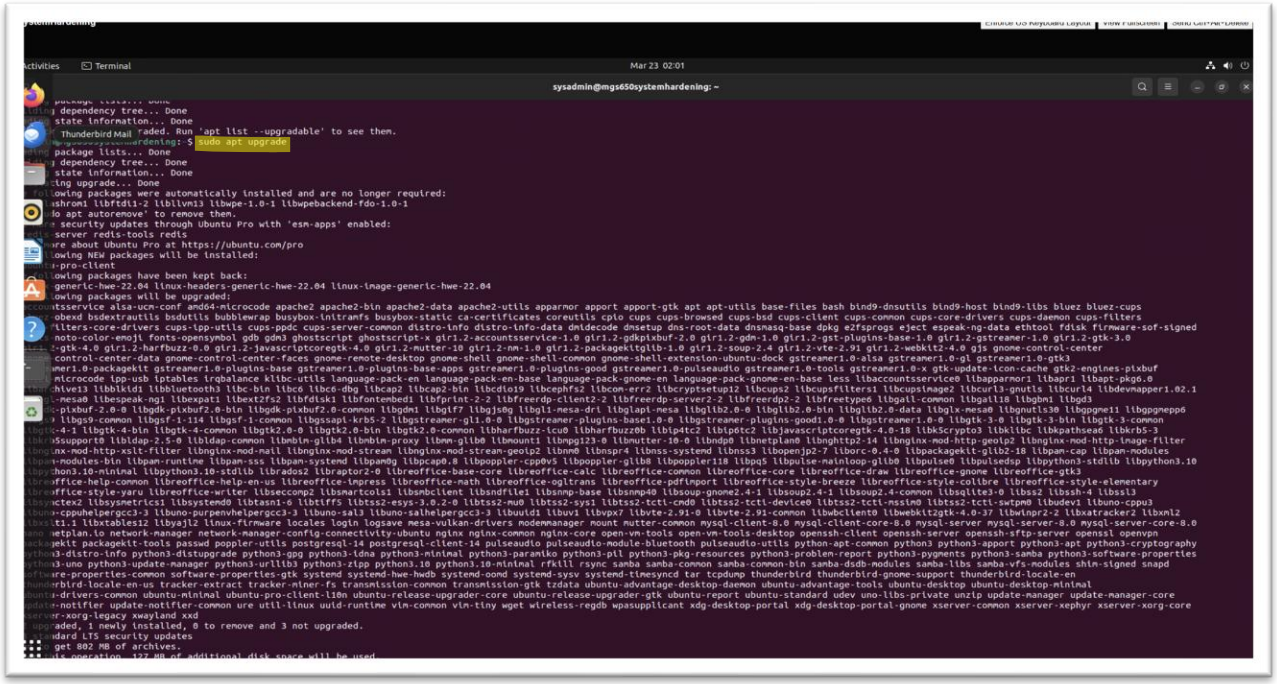

Figure 2: Screenshot of “sudo apt upgrade” to update installed packages to latest versions.

- We use “systemctl status ufw” command to check whether ufw service are up and running or not as highlighted in figure 3.

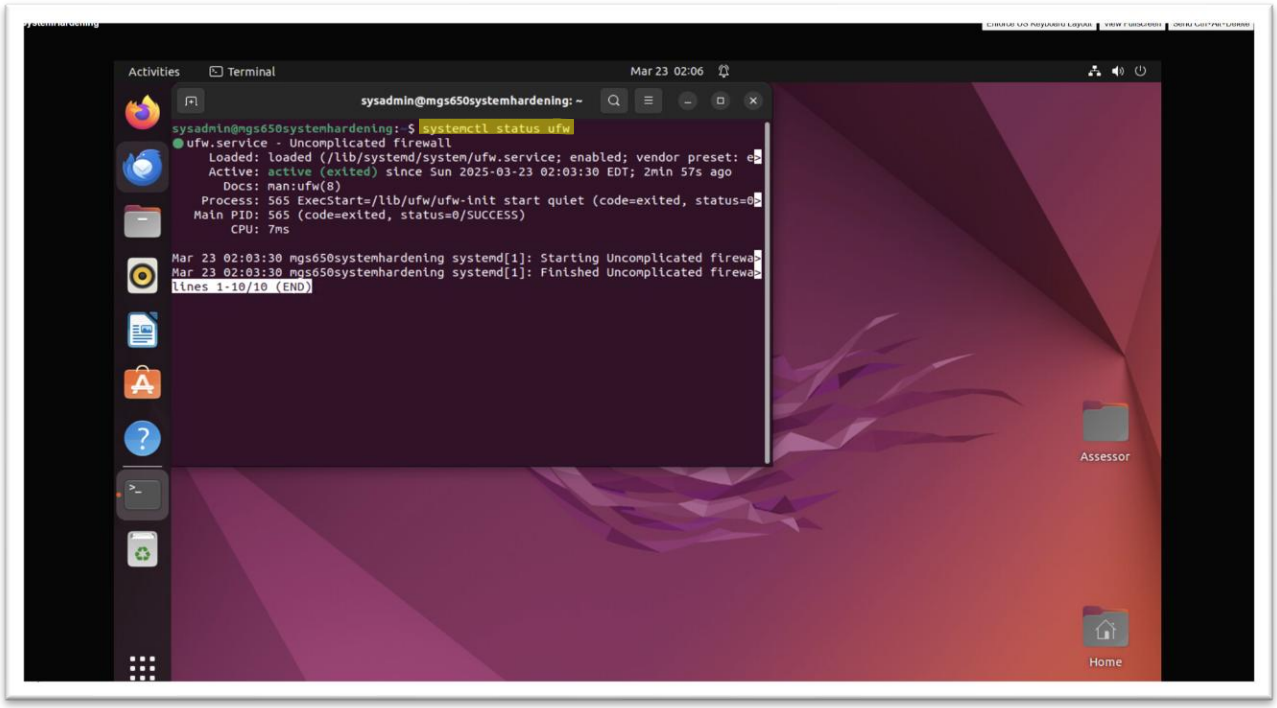

Figure 3: Screenshot of “systemctl status ufw” to check status of ufw service if its active or not.

- We use “sudo ufw status” to check if ufw firewall is running or not. So, it’s not necessary that if ufw service is active, ufw firewall is active too with it.

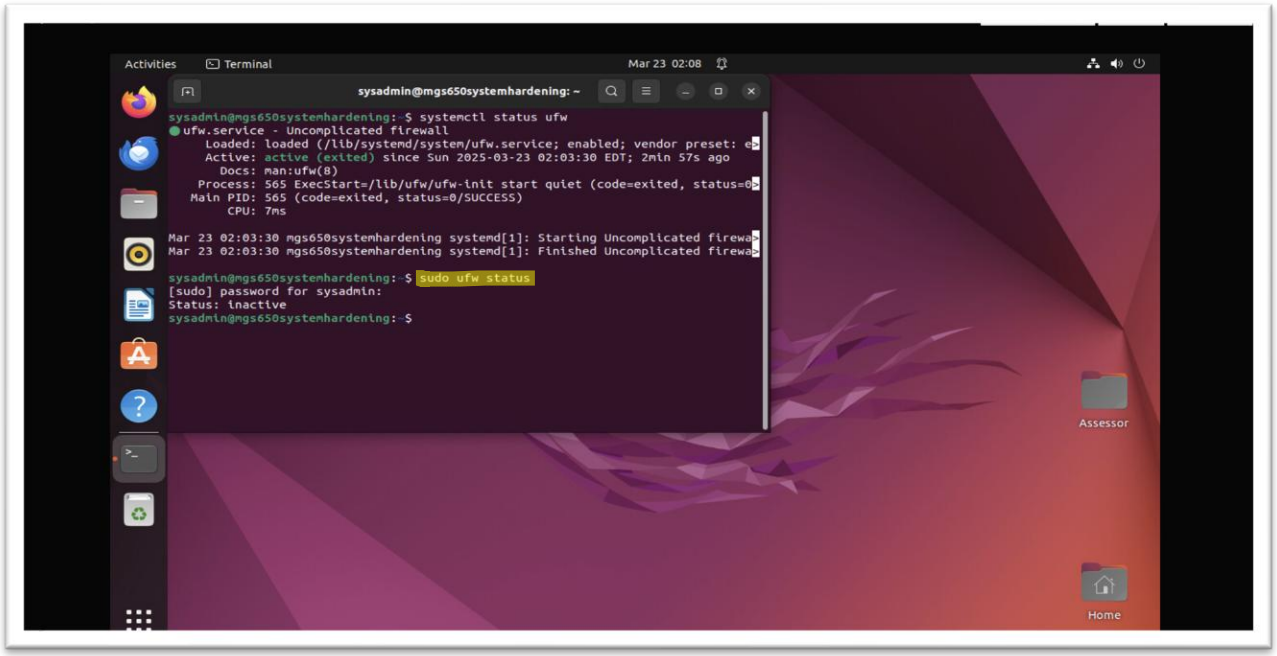

Figure 4: Screenshot of “sudo ufw status” to check status of ufw firewall.

- So, we enter “sudo ufw enable” to enable the firewall and then check the status (which is active) as highlighted in figure 5.

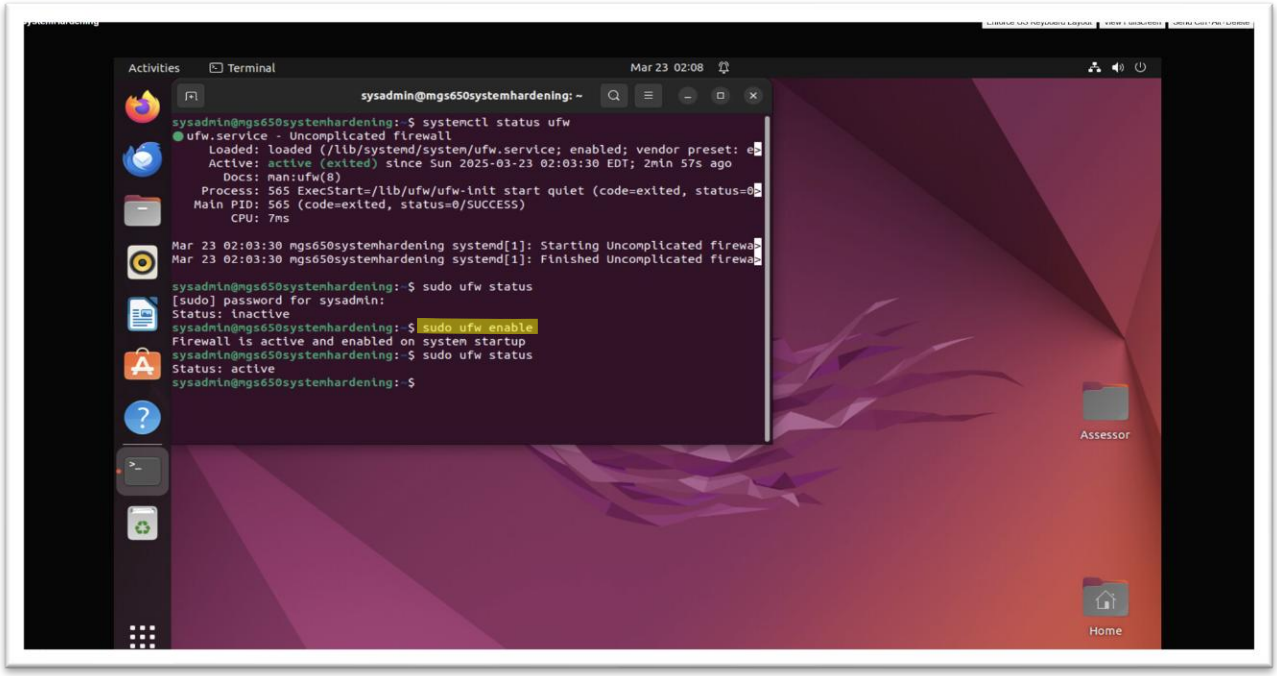

Figure 5: Screenshot of “sudo ufw enable” to enable firewall in the system.

- Now we enter “sudo ufw status verbose” to check the additional information of verbose status such as the default policies (allow incoming and outgoing and disable routing) and any rules that were added (there are no rules in our case).

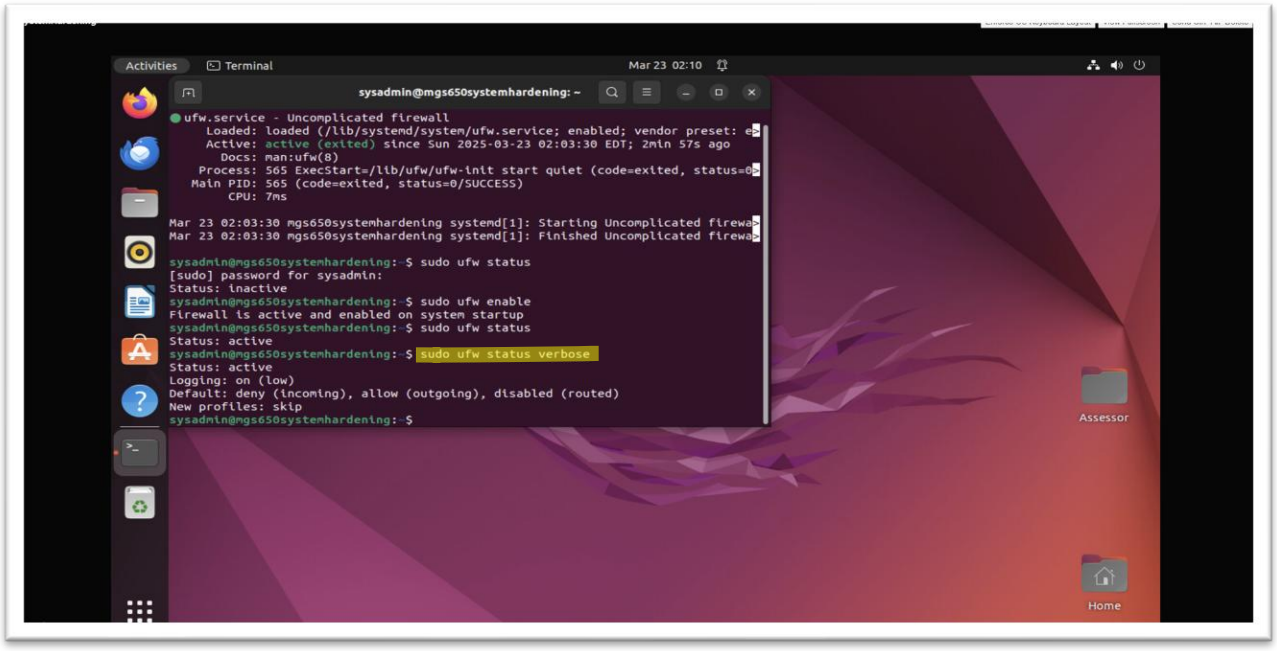

Figure 6: Screenshot of “sudo ufw status verbose” to check verbose status with additional information.

- We input “sudo lsof -i” to find out what commands are using the network, along with the process ID, user, ports, and more as shown in figure 7.

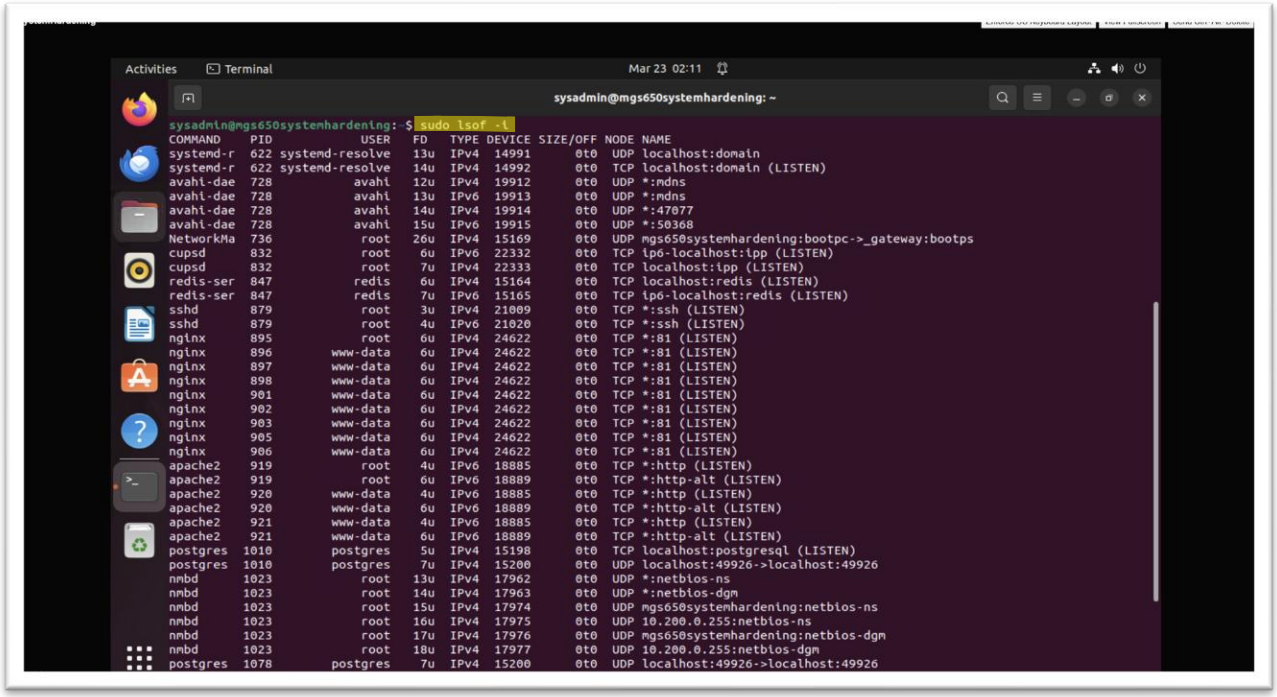

Figure 7: Screenshot of “sudo lsof -I” to find out all commands running on that network."

- Now, in order to connect to the server we need to allow inbound SSH traffic using “sudo ufw allow 22” (SSH port is 22) as shown in figure 8.

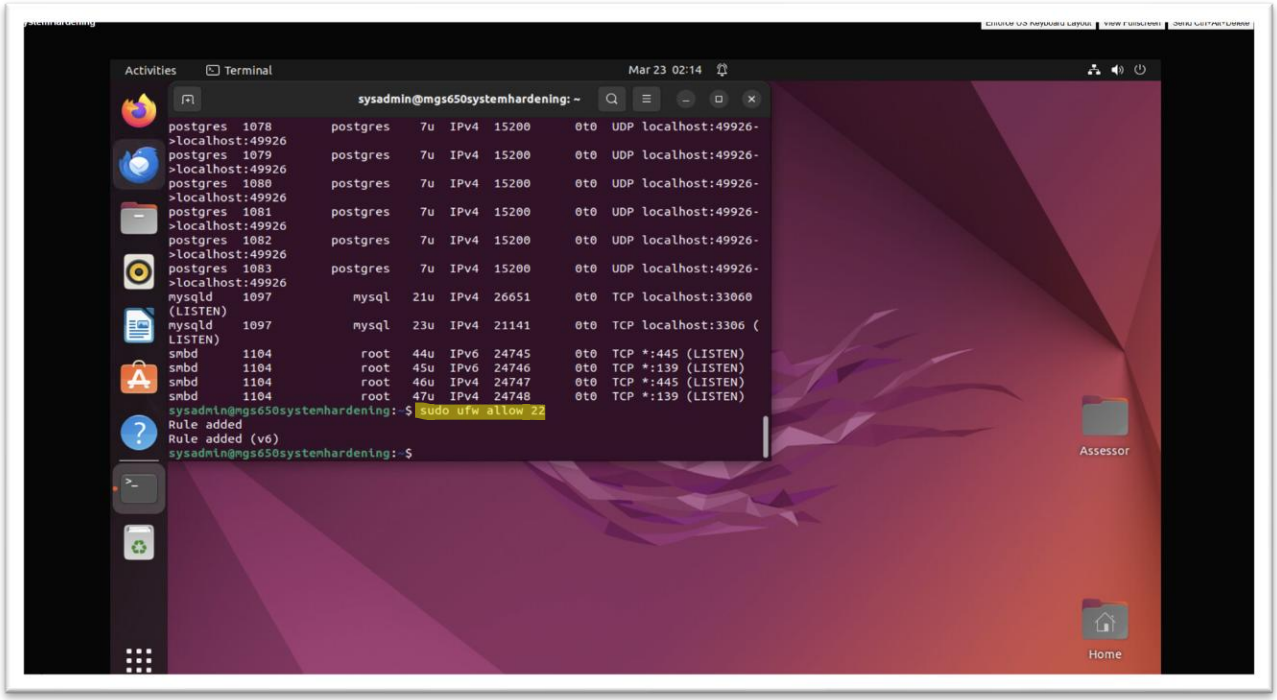

Figure 8: Screenshot of “sudo ufw allow 22” to allow inbound SSH traffic.

- After that, to allow all of the trafiic used by Apache server which is already installed in the machine, we can use “sudo lsof -I -P | grep apache” to first figure out what type of traffic is available for apache server and then enter firewall rules “sudo ufw allow 443” which is highlighted in figure 9, to allow all traffic for apache server.

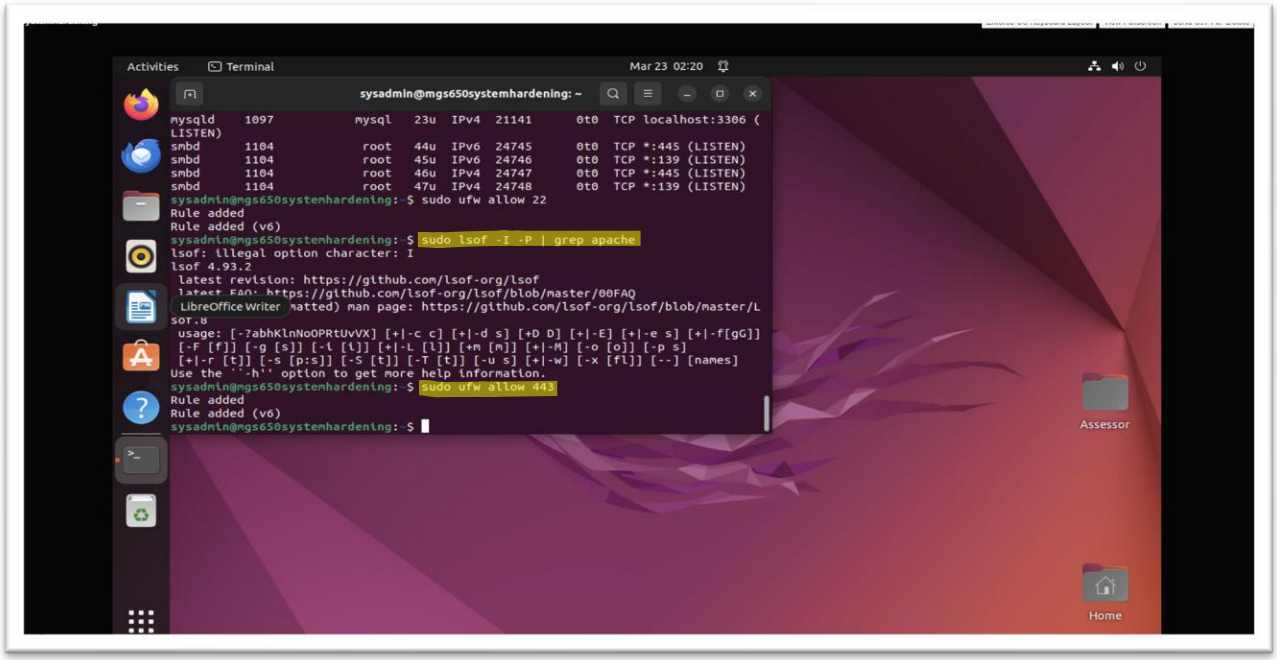

Figure 9: Screenshot of “sudo ufw allow 443” to allow all traffic for apache server to run.

- So, for additional security, we can block all of the available open ports running by changing default policy to deny using “sudo ufw default deny incoming” as shown in figure 10.

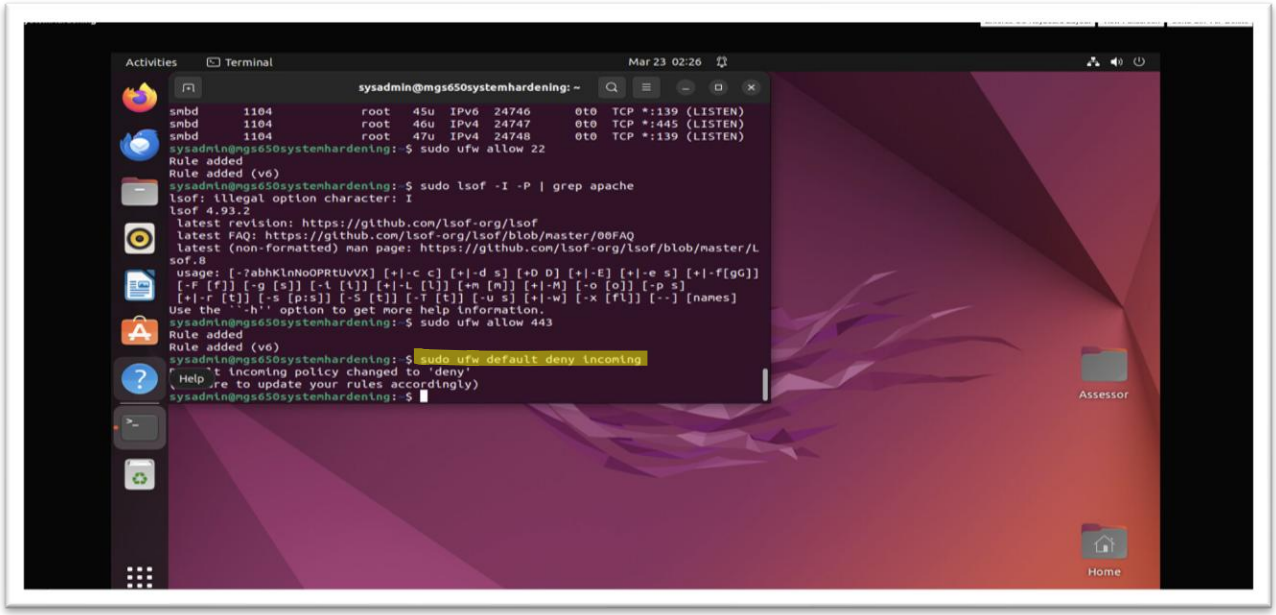

Figure 10: Screenshot of “sudo ufw default deny incoming” to deny any open ports.

- To allow outgoing rules to work, we need to enter every outgoing ports separately such as- for dns, we need to enter “sudo ufw allow out 53/udp” and “sudo ufw allow out 53/tcp” (dns sometimes uses tcp for larger traffic), for http enter “sudo ufw allow out 80/tcp” and for https enter “sudo ufw allow out 443/tcp” (all rules are highlighted in figure 11).

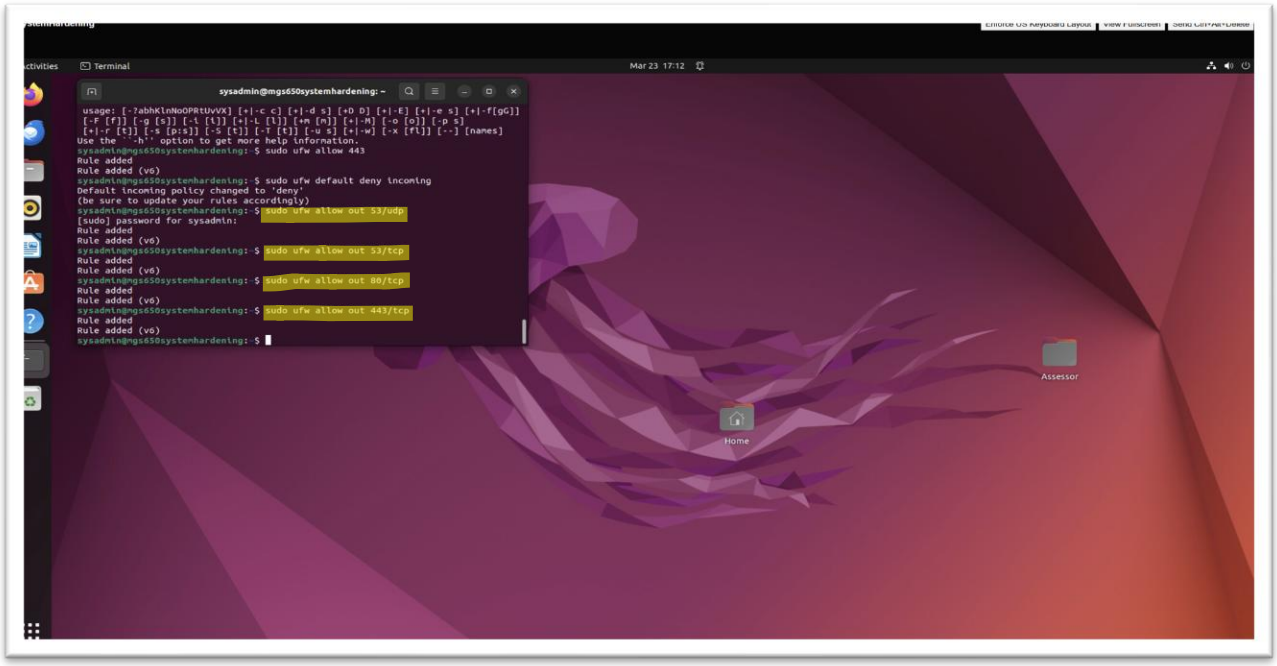

Figure 11: Screenshot of all different outgoing rules applied in ufw services.

- Now, we can again reject every outgoing traffic by default using “sudo ufw default reject outgoing” as shown in figure 12.

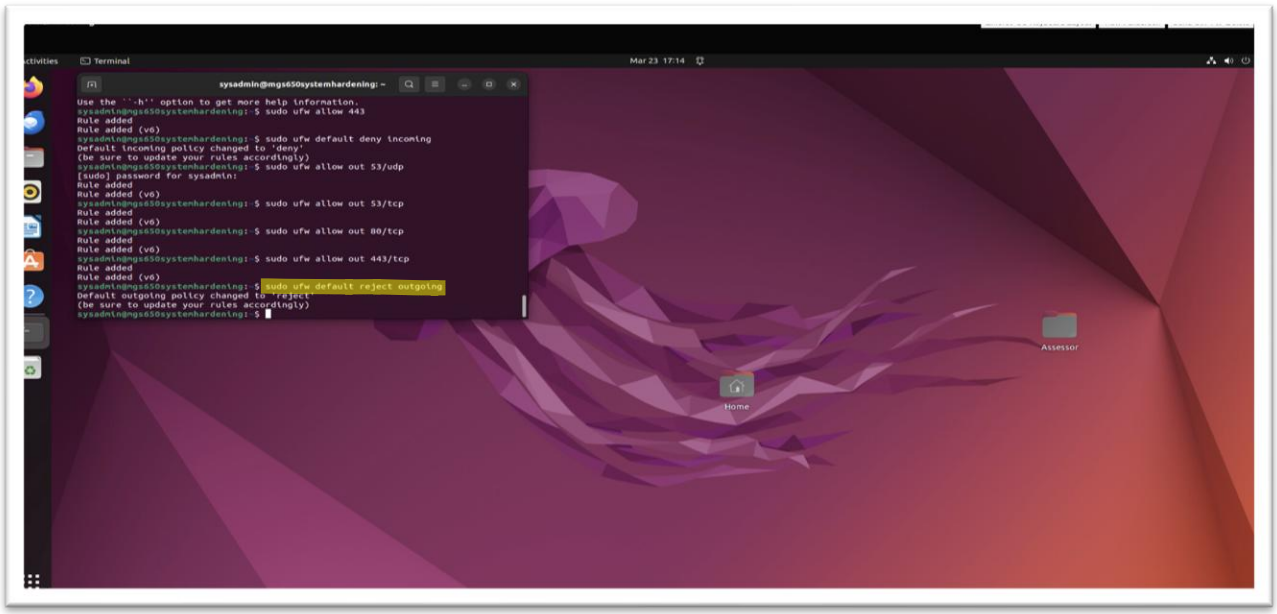

Figure 12: Screenshot of default reject policy using “sudo ufw default reject outgoing”.

- Now finally we can use “sudo ufw status verbose” as higlhighted in figure 13 to check all firewall rules which were newly apllied to the ufw services.

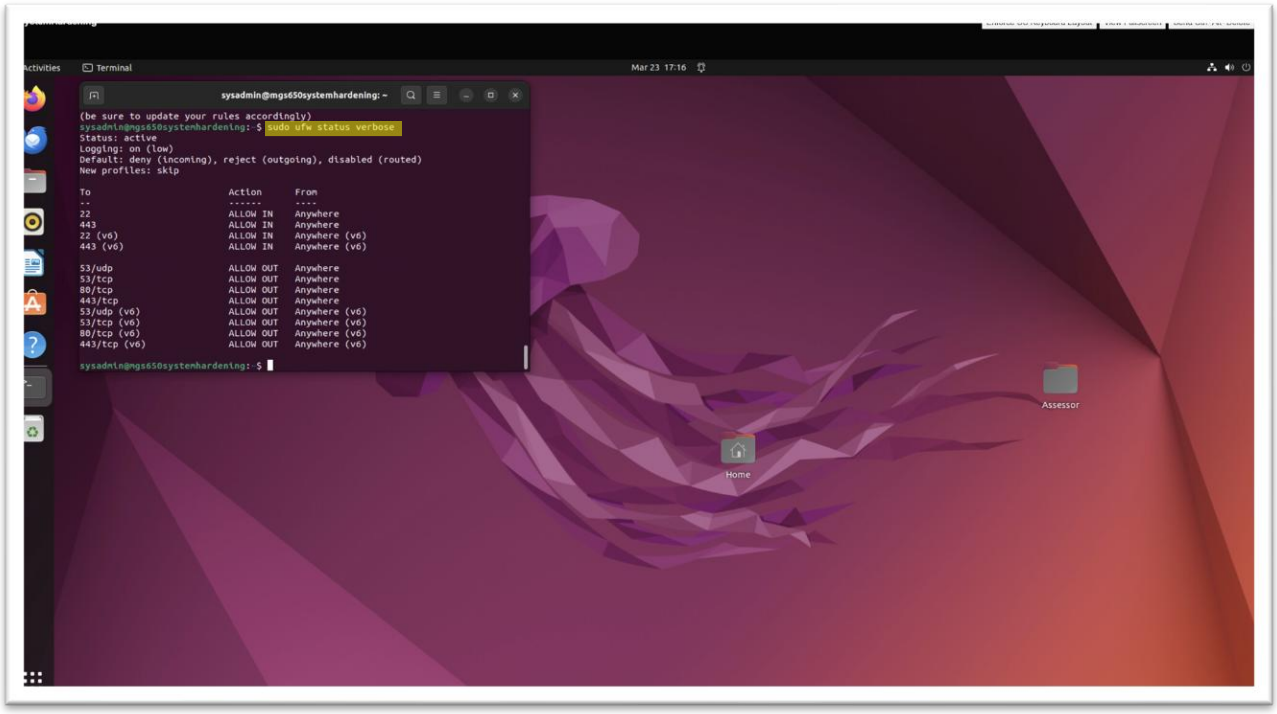

Figure 13: Screenshot of “sudo ufw status verbose” to check all newly applied firewall rules.

- We can input “ps aux” to view all of the running or active processes in one single list in shown in figure 14.

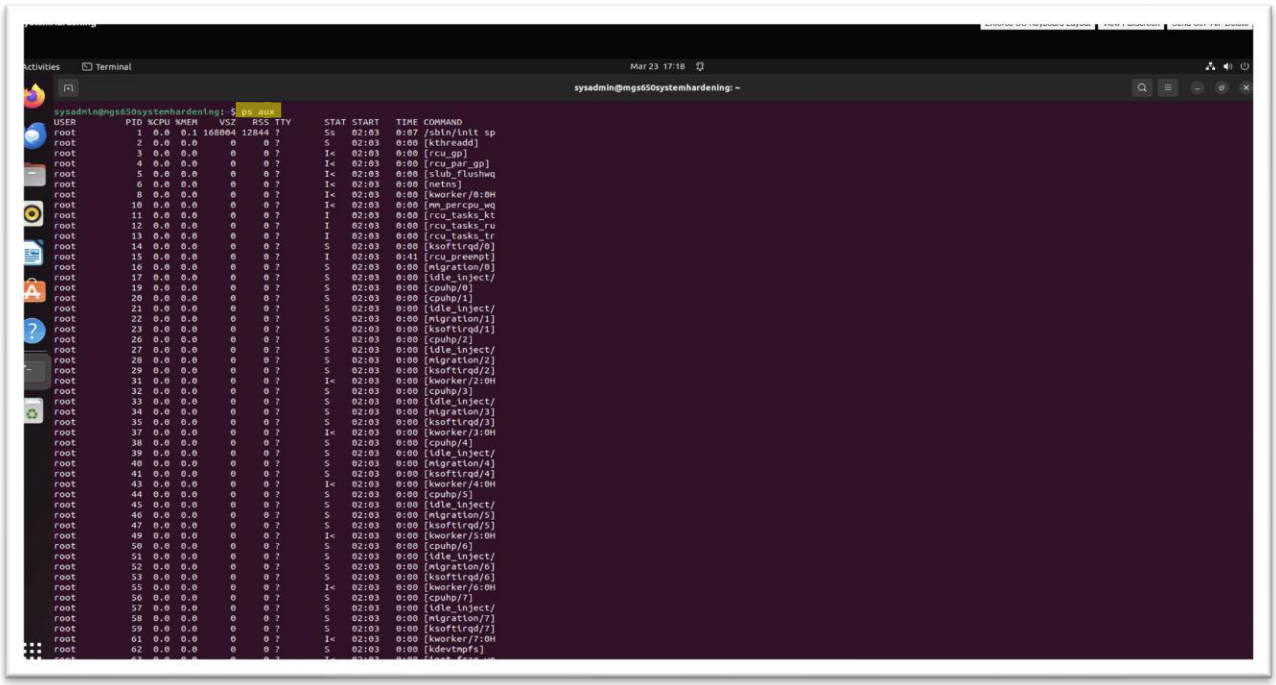

Figure 14: Screenshot of “ps aux” to check all running or active processes.

- We can also use “systemctl status | grep service” to check for all services which are managed by the Systemd init system and filter it out as shown in figure 15.

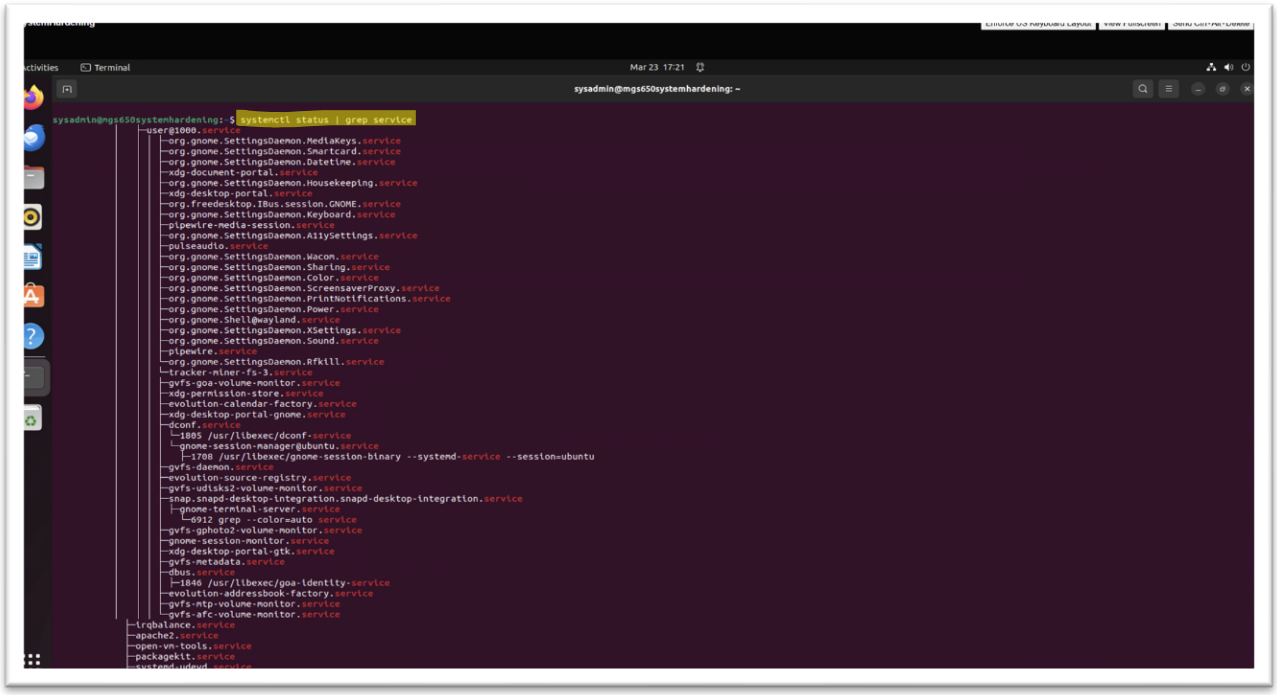

Figure 15: Screenshot of “systemctl status | grep service” to check for all services in system init.

- Now we can also observe smbd as a running services which is not being used by our current web servers so we can uninstall it using “sudo apt purge samba” as highlighted in figure 16.

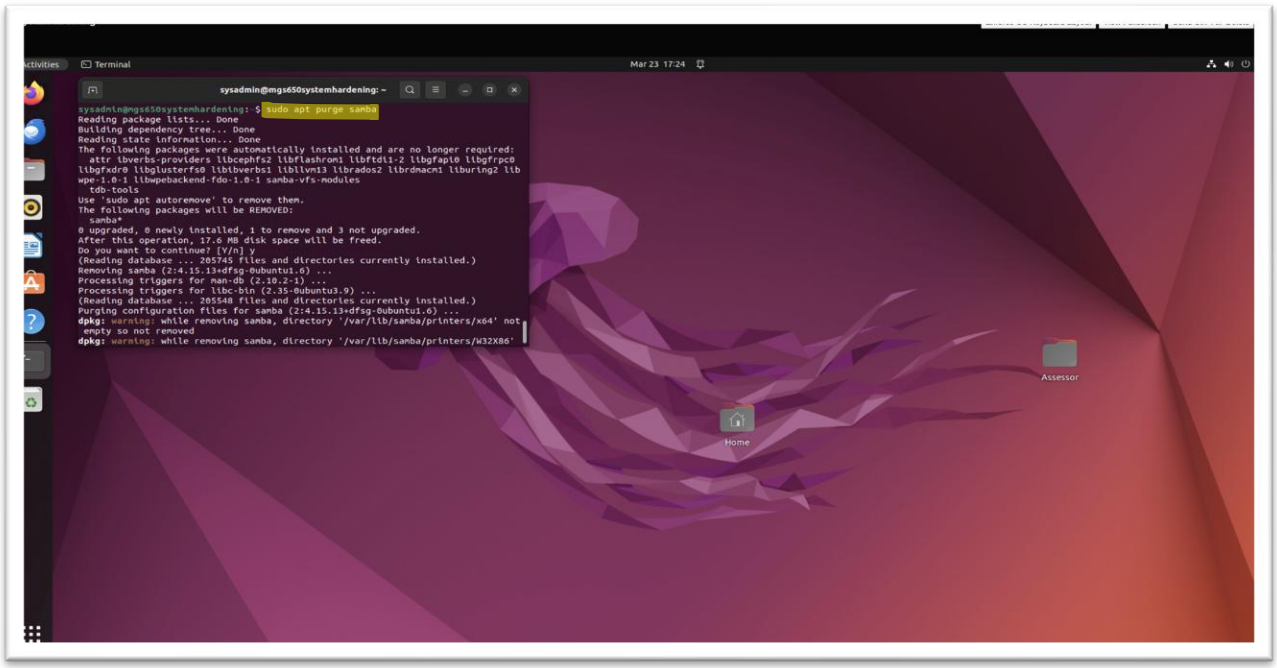

Figure 16: Screenshot of “sudo apt purge samba” to uninstall Samba’s running services.

- We enter “systemctl status nginx” to check if nginx services is up and running in the system as shown in figure 17.

Figure 17: Screenshot of “systemctl status nginx” to check if nginx service is active or not.

- We use “sudo systemctl stop nginx” to stop its services. But then if we reboot the system, it will automatically restart so to stop it, we can use “sudo systemctl disable nginx” to disable the services properly as highlighted in figure 18.

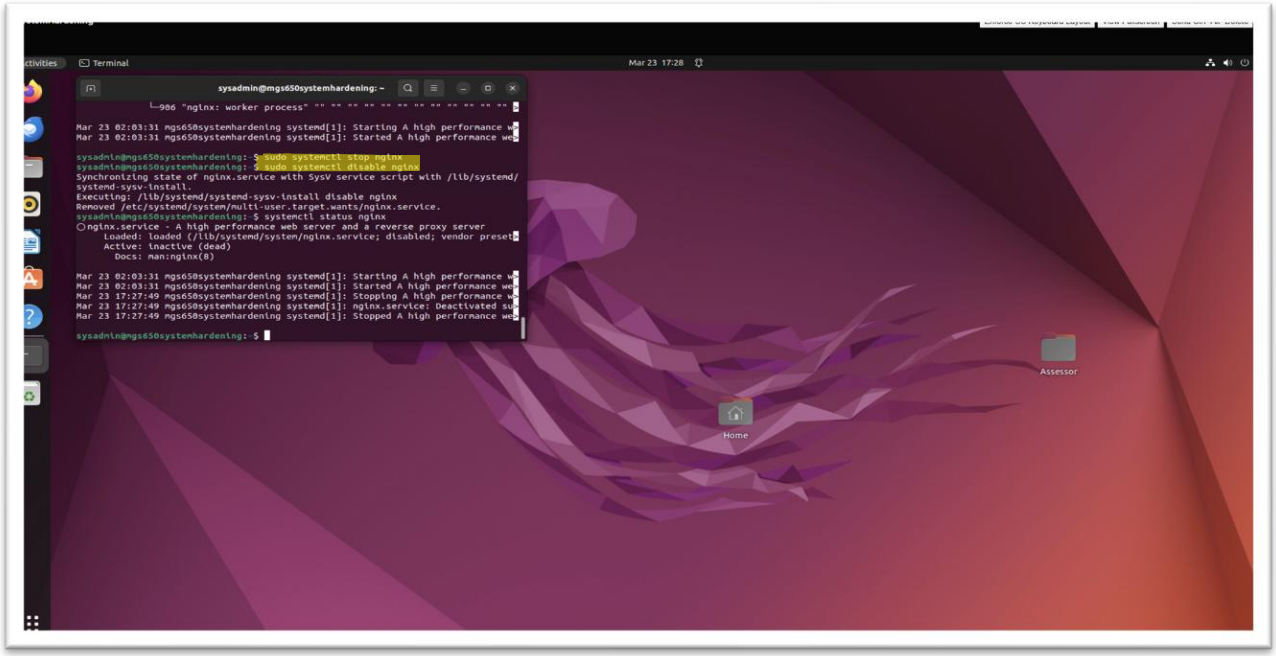

Figure 18: Screenshot of “sudo systemctl disable nginx” to disable nginx services.

- We can delete a user- amaright from system using “sudo deluser amaright” and by entering “sudo rm -rf /home/amaright”, we can delete her home folder too as highlighted in figure 19.

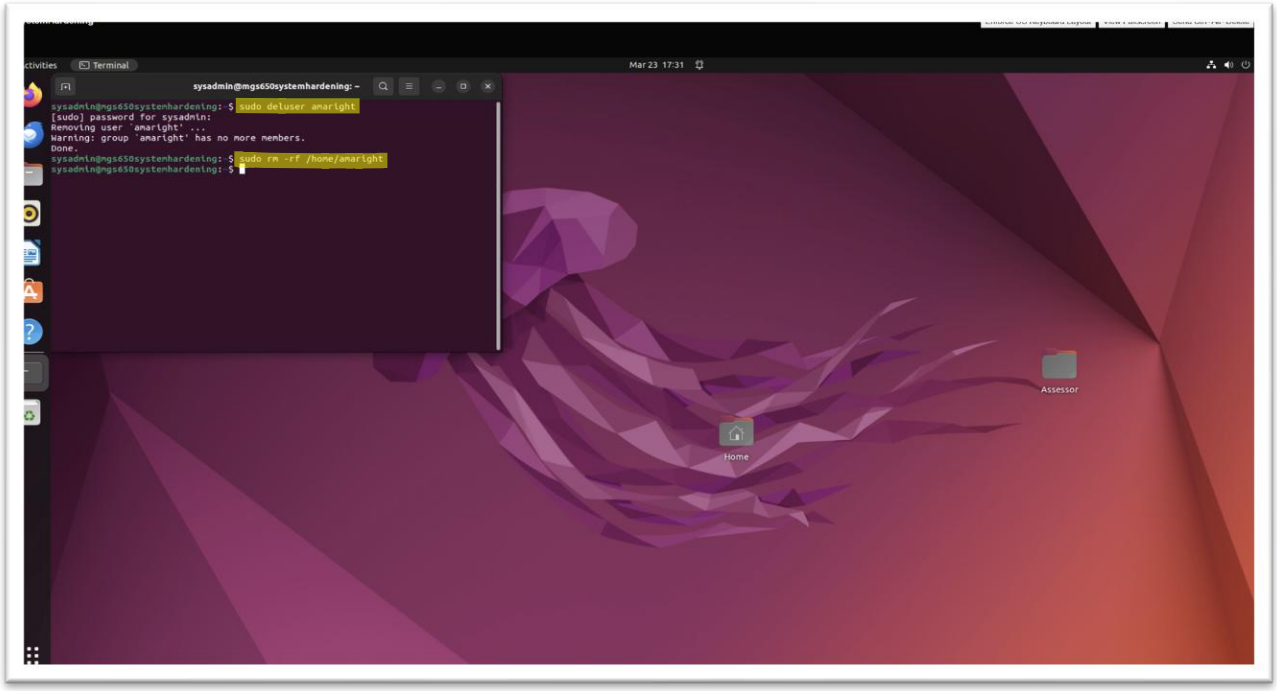

Figure 19: Screenshot of “sudo deluser amaright” to delete that user from the system.

- We can also lock a user’s account temporarily (in this case, user is jsweeney) by inputting “sudo passwd -l jsweeney”. This will put an exclamation mark in front of her password hash till we unlock it.

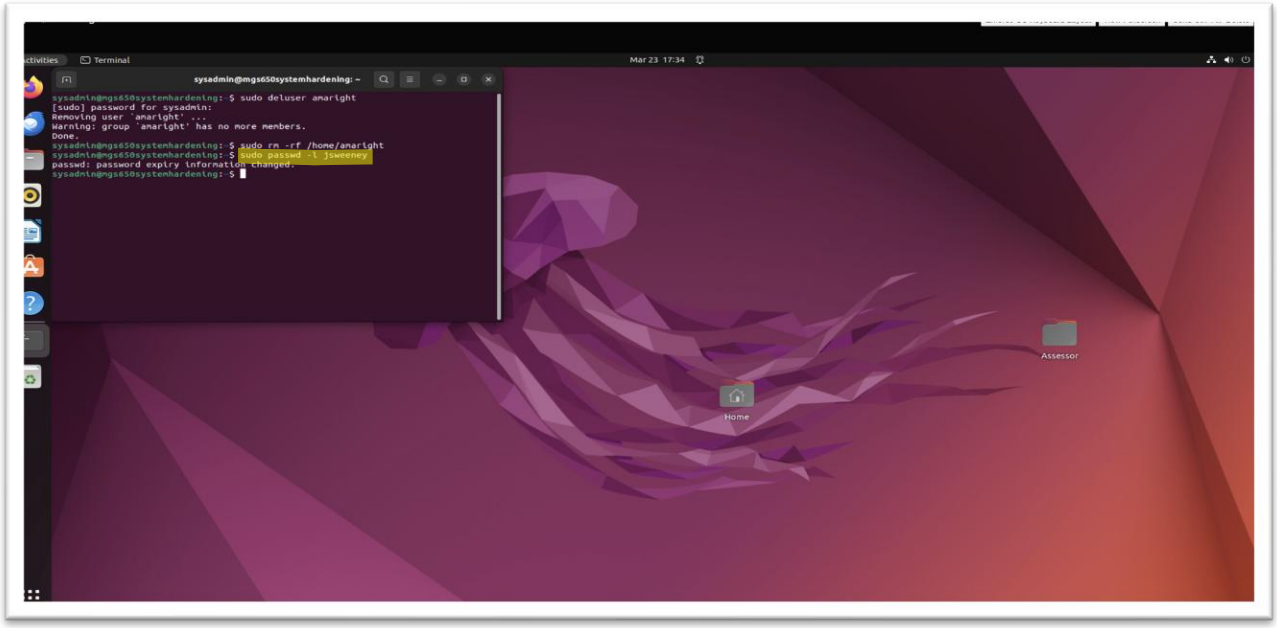

Figure 20: Screenshot of “sudo passwd -l jsweeney” to lock her account temporarily.

- We can view the shadow file using “sudo cat /etc/shadow” as shown in figure 21.

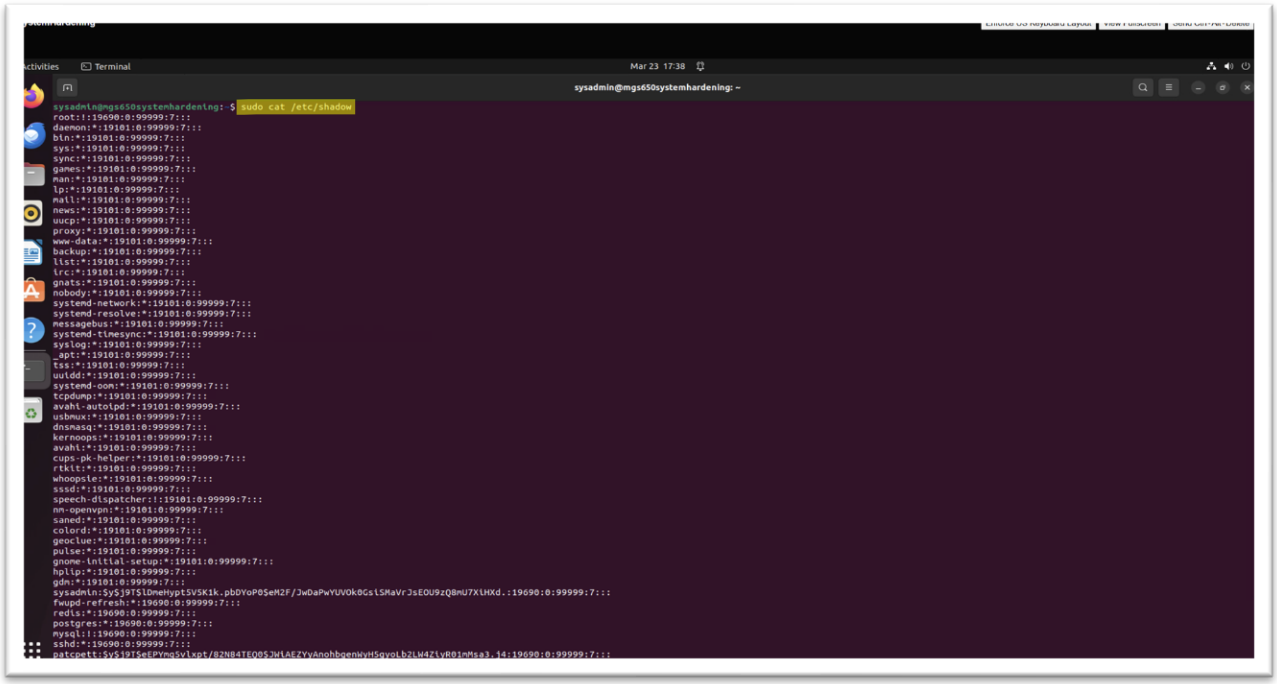

Figure 21: Screenshot of “sudo cat /etc/shadow” to view shadow file.

- So, user- dunnxter doesn’t have any password so to set it, we can enter “sudo passwd dunnxter” as highlighted in figure 22 and it will update the hash password by putting “$” in the blank as highlighted in figure 23.

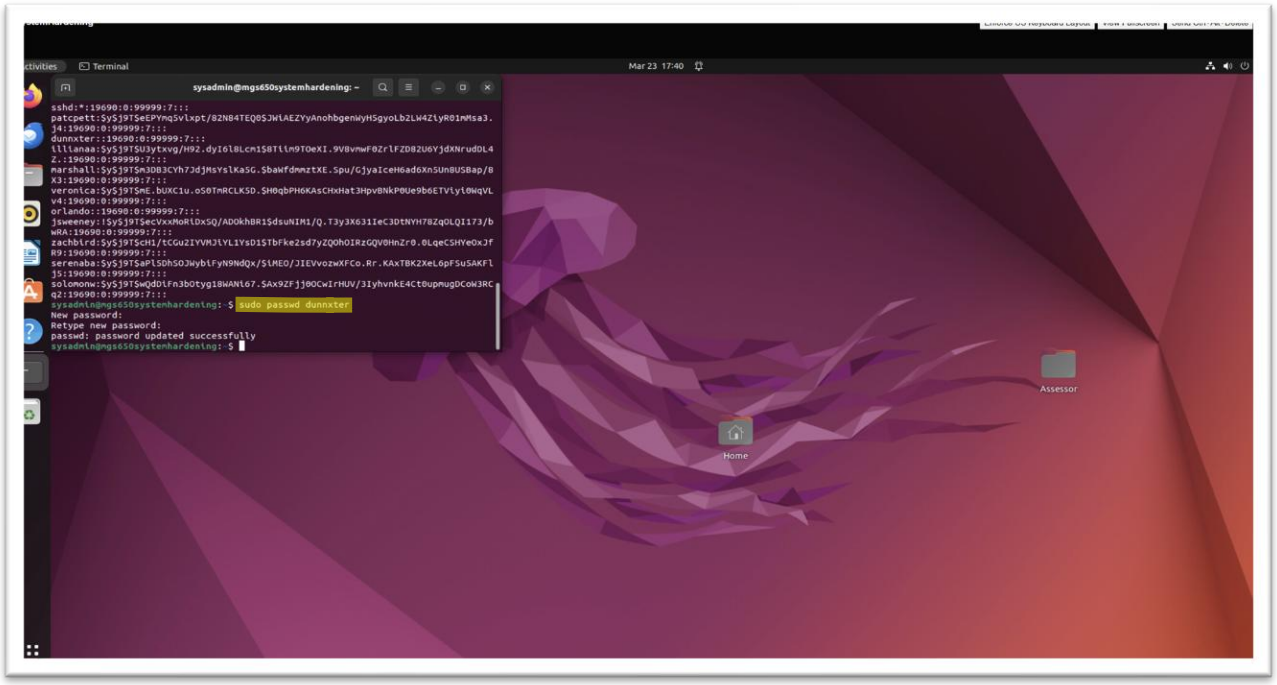

Figure 22: Screenshot of “sudo passwd dunnxter” to set new password for that user.

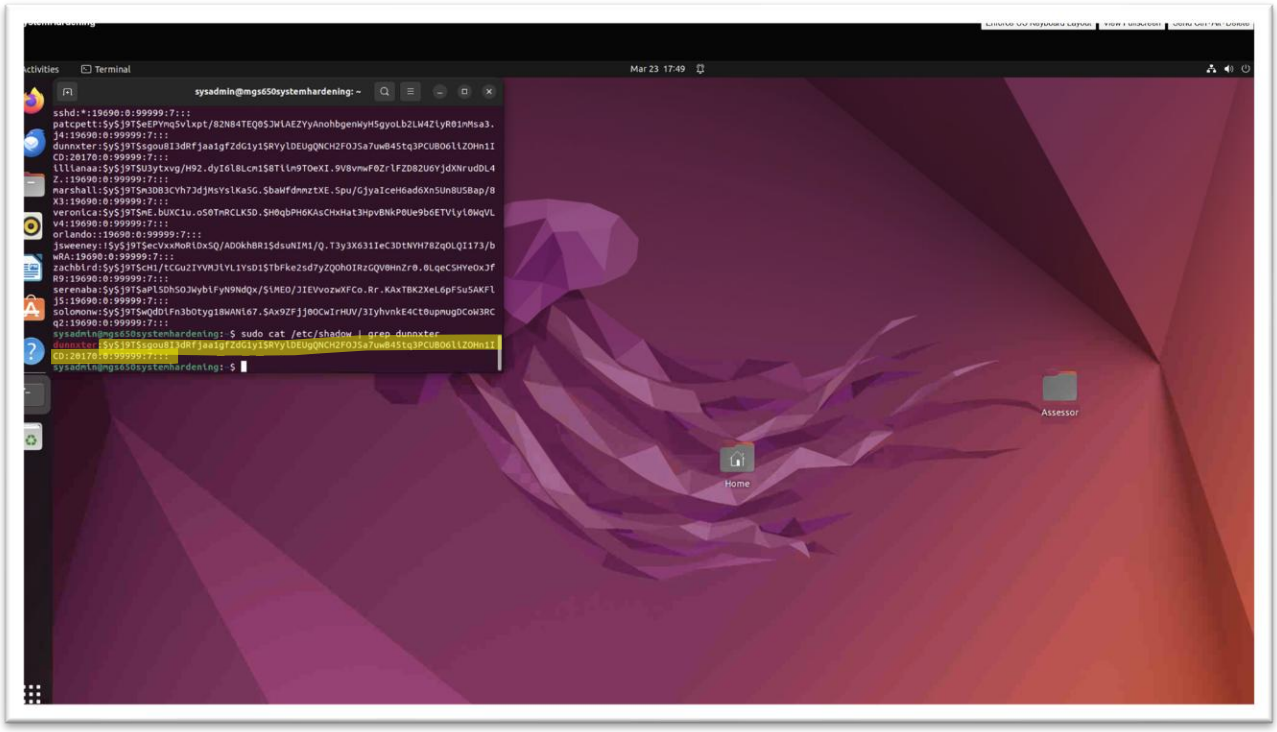

Figure 23: Screenshot of hash password of user "dunnxter".

- We can also enter "date" to get the current data with time and also put 'echo "This is fahmed29."' to get that same reply back as shown below.

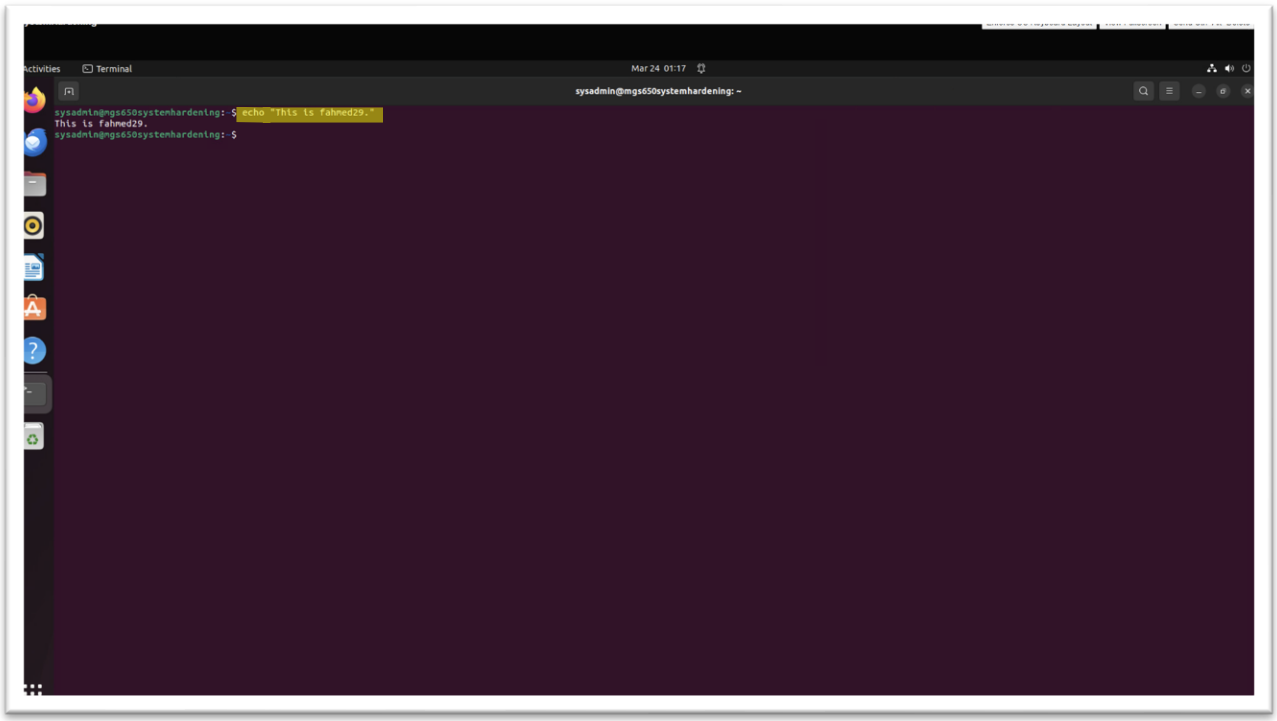

Figure 24: Screenshot of 'echo "This is fahmed29."' To get same reply back.

- We can use “sudo visudo” to access it to find which users to use root command via sudo as shown below.

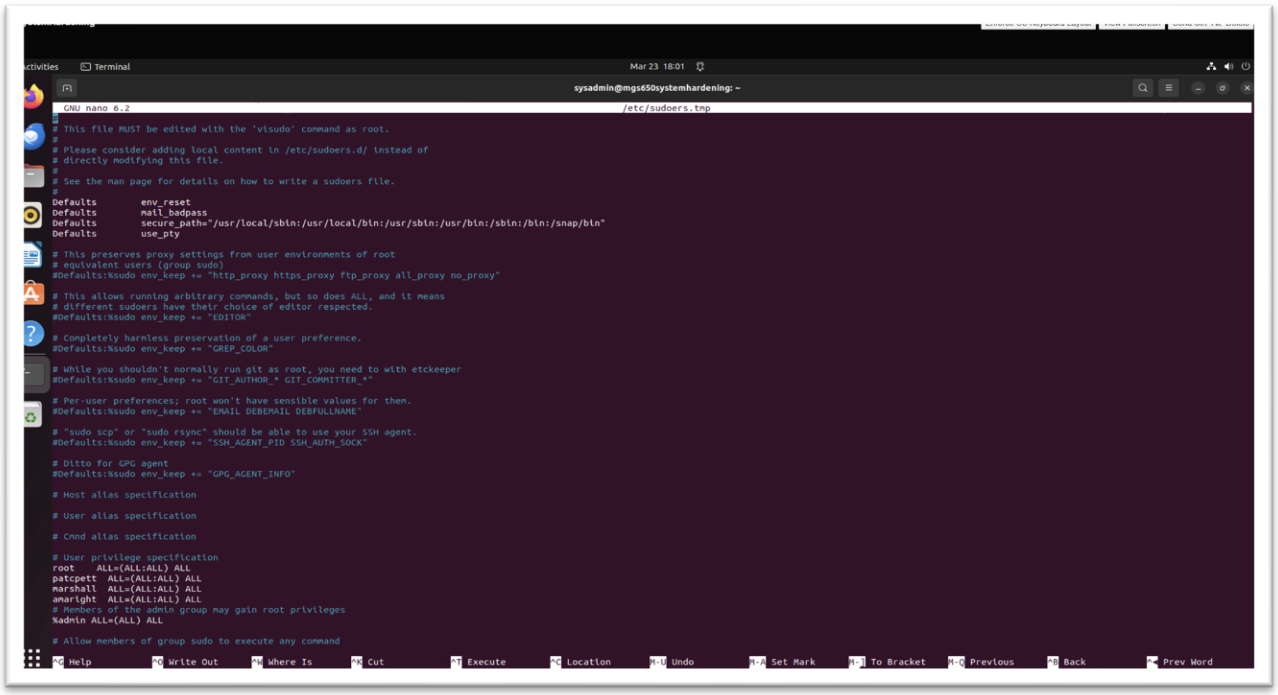

Figure 25: Screenshot of “sudo visudo” to view all sudo users.

- We can view all of the sudo users specialfically using “sudo cat /etc/group | grep sudo” as shown in figure 26.

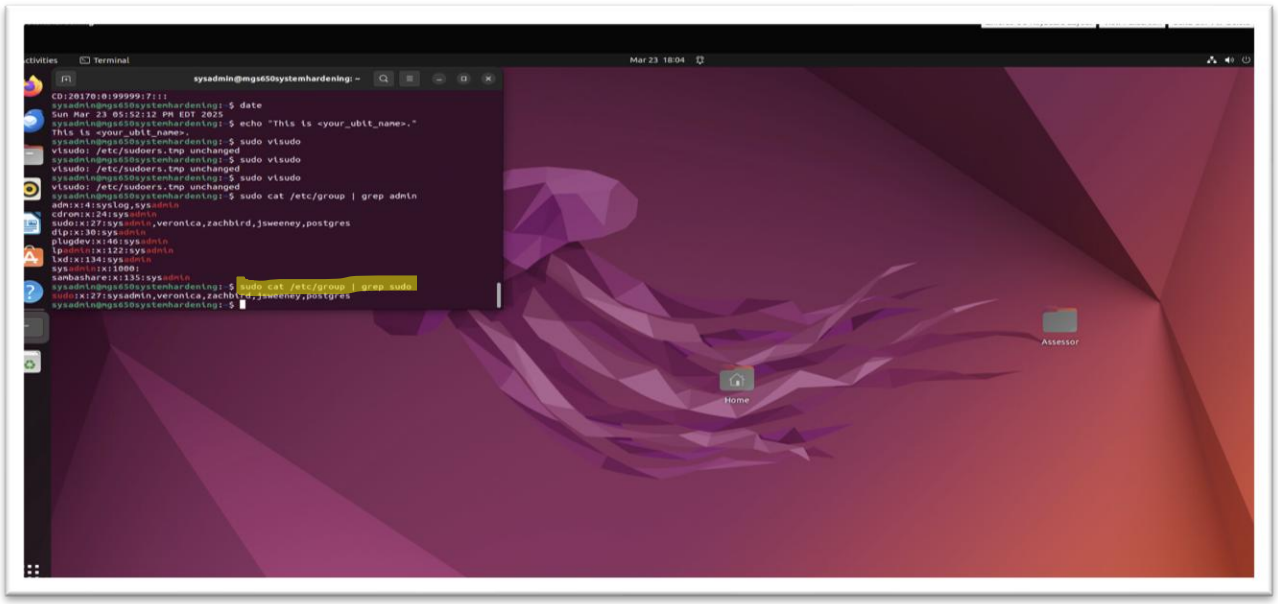

Figure 26: Screenshot of “sudo cat /etc/group | grep sudo” to get all users with sudo access.

- We have to remove “amaright ALL=(ALL:ALL) ALL” to clearing it as highligted in figure 27 and removed in figure 28.

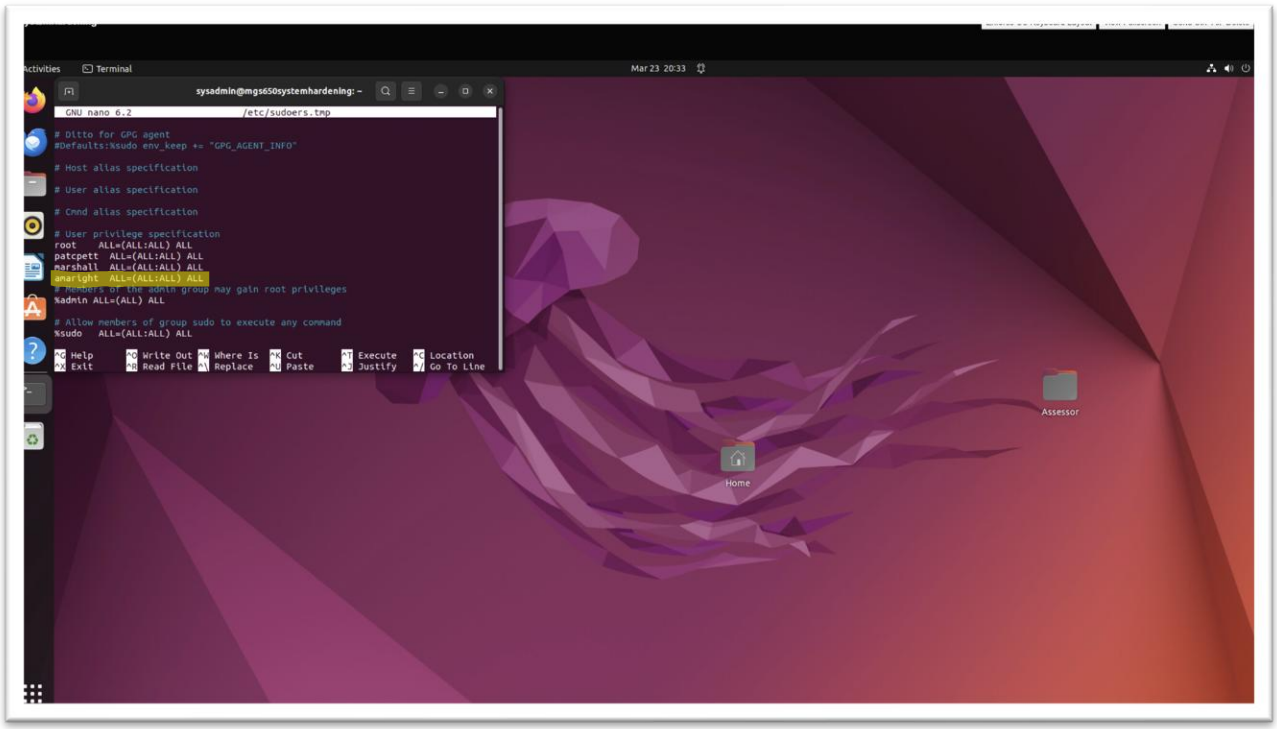

Figure 27: Screenshot of all users with sudo privileges.

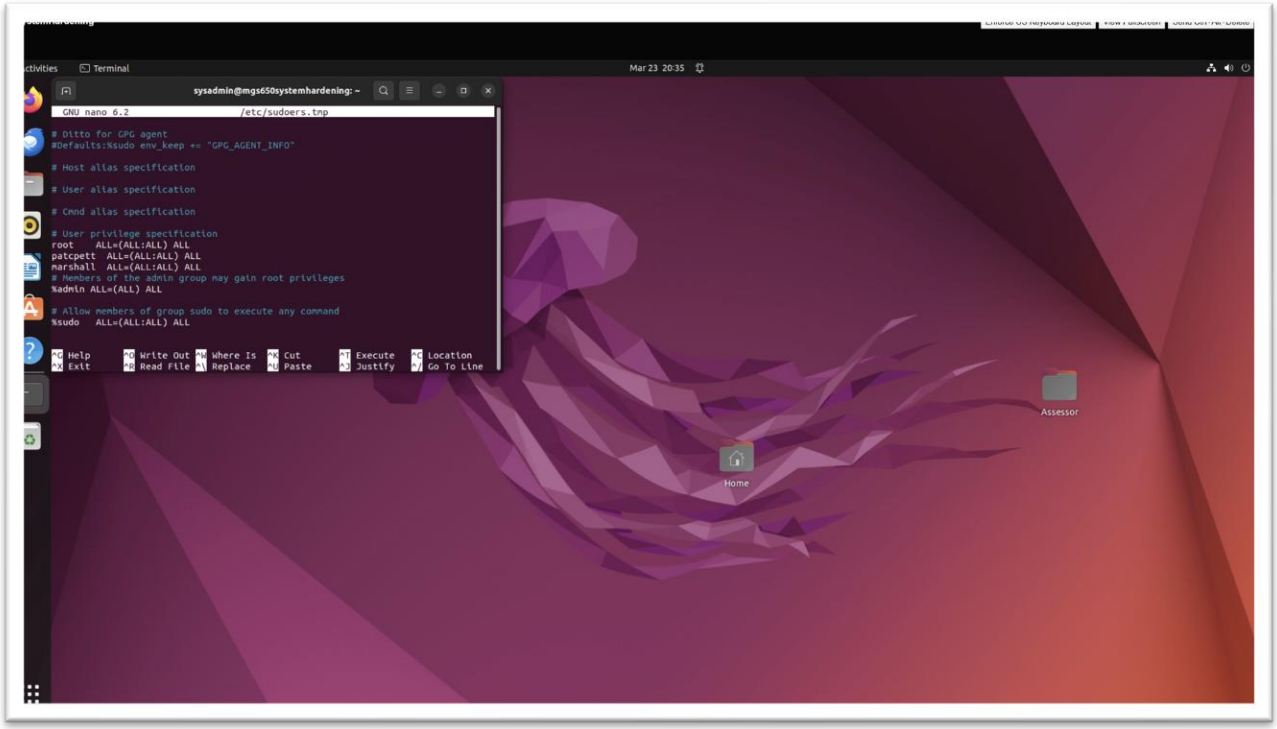

Figure 28: Screenshot of deleting amaright user in sudo privileges.

- Now, we can remove a user “postgres” using “sudo deluser postgres sudo” as highlighted in figure 29 and then enter “sudo cat /etc/group | grep sudo” to check if postgres user is removed or not from sudo privileges as highlighted in figure 30.

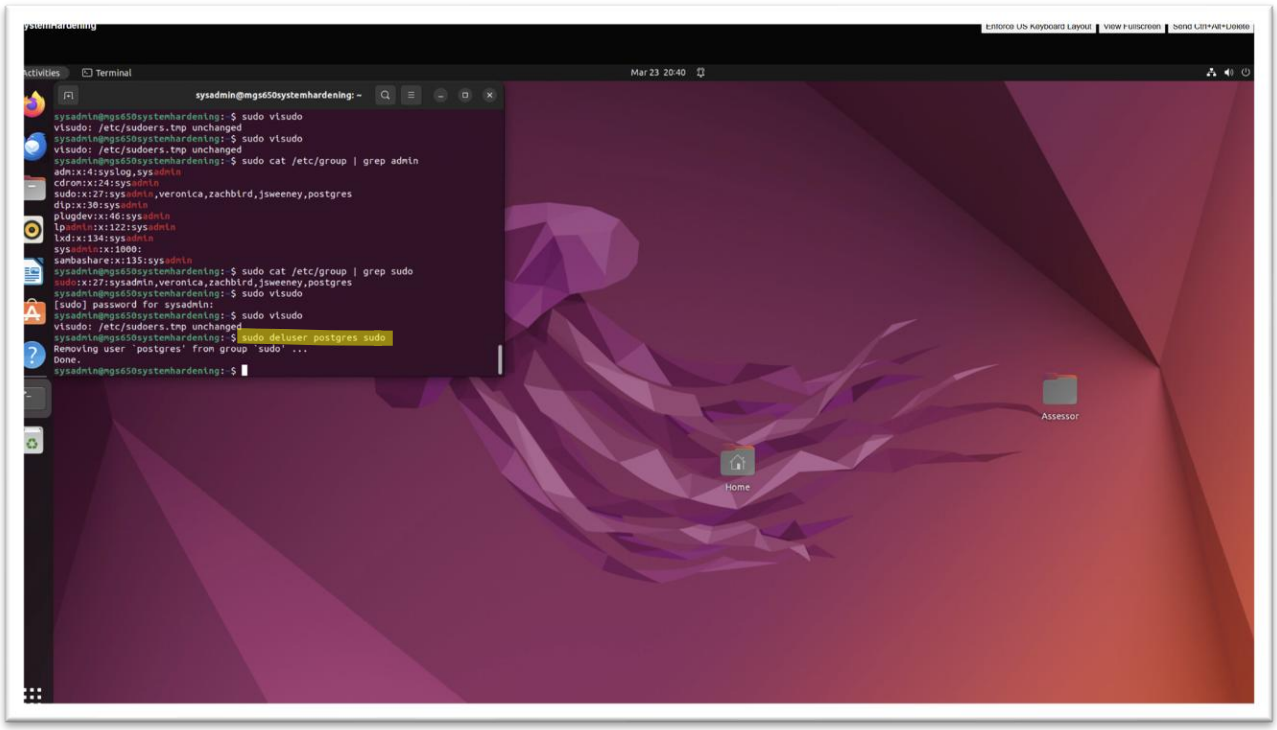

Figure 29: Screenshot of “sudo deluser postgres sudo” to remove user postgres from sudo.

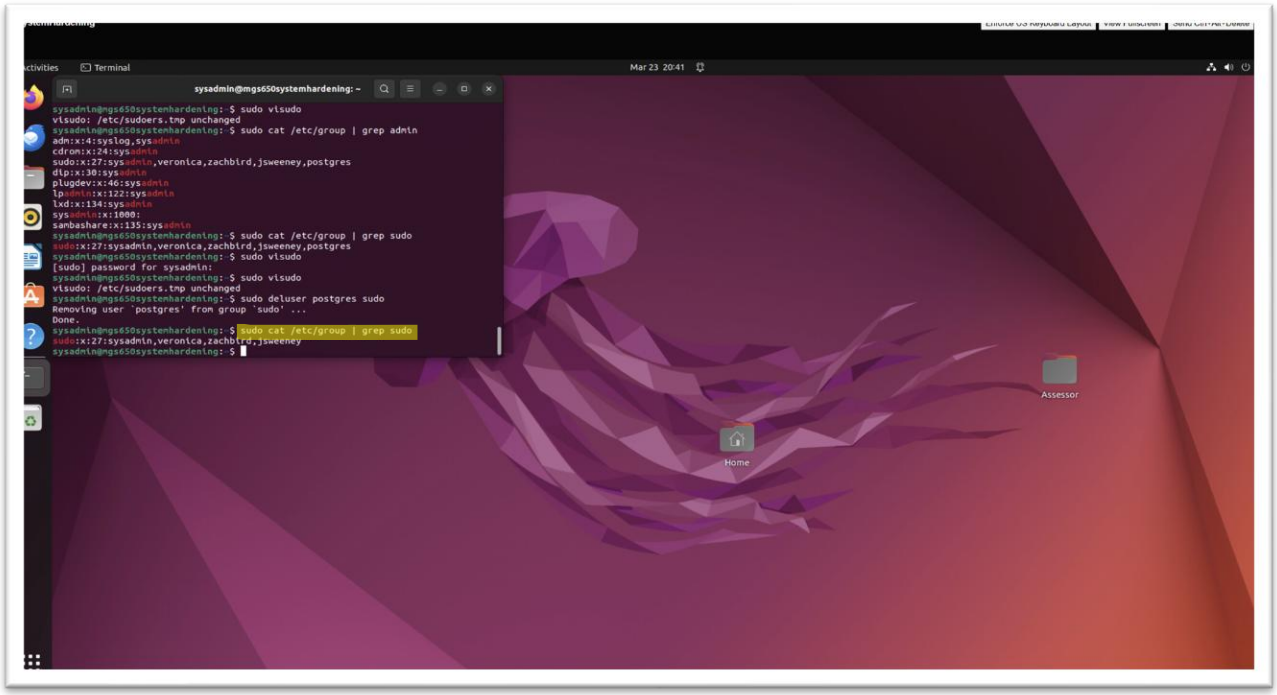

Figure 30: Screenshot of checking for user postgres in sudo privileges.