

# 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.

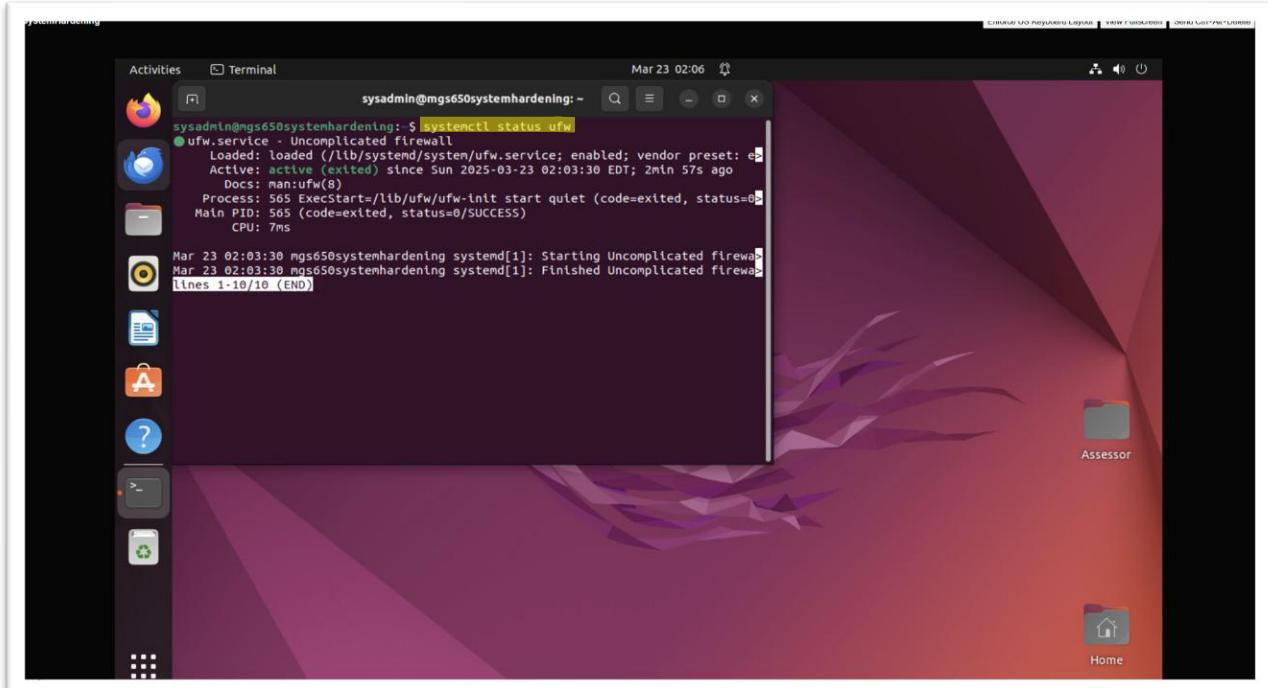
```
[sudo] password for sysadmin
sysadmin@mg650systemhardening:~$ sudo apt update
[sudo] password for sysadmin:
Hit:1 http://us.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu jammy-backports InRelease [127 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu jammy-security InRelease [127 kB]
Get:5 https://us.archive.ubuntu.com/ubuntu jammy-updates/natty l386 Packages [769 kB]
Get:6 https://us.archive.ubuntu.com/ubuntu jammy-updates/natty amd64 Packages [2,390 kB]
Get:7 https://us.archive.ubuntu.com/ubuntu jammy-updates/natty Translation-en [103 kB]
Get:8 https://us.archive.ubuntu.com/ubuntu jammy-updates/main DEP-11 Metadata [103 kB]
Get:9 https://us.archive.ubuntu.com/ubuntu jammy-updates/main DEP-11 48x48 Icons [37.5 kB]
Get:10 http://us.archive.ubuntu.com/ubuntu jammy-updates/main DEP-11 64x64 Icons [56.8 kB]
Get:11 http://us.archive.ubuntu.com/ubuntu jammy-updates/main DEP-11 128x128 Icons [64.8 kB]
Get:12 http://us.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [3,114 kB]
Get:13 http://us.archive.ubuntu.com/ubuntu jammy-updates/restricted l386 Packages [40.0 kB]
Get:14 http://us.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [103 kB]
Get:15 https://us.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en-1 Metadata [212 B]
Get:16 http://us.archive.ubuntu.com/ubuntu jammy-updates/restricted DEP-11 48x48 Icons [29 B]
Get:17 http://us.archive.ubuntu.com/ubuntu jammy-updates/restricted DEP-11 64x64 Icons [29 B]
Get:18 http://us.archive.ubuntu.com/ubuntu jammy-updates/restricted DEP-11 128x128 Icons [29 B]
Get:19 https://us.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 c-n-f Metadata [652 B]
Get:20 https://us.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1,194 kB]
Get:21 https://us.archive.ubuntu.com/ubuntu jammy-updates/universe DEP-11 Metadata [103 kB]
Get:22 https://us.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [294 kB]
Get:23 https://us.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 DEP-11 Metadata [359 kB]
Get:24 http://us.archive.ubuntu.com/ubuntu jammy-updates/universe DEP-11 48x48 Icons [26.8 kB]
Get:25 http://us.archive.ubuntu.com/ubuntu jammy-updates/universe DEP-11 64x64 Icons [40.8 kB]
Get:26 http://us.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 c-n-f Metadata [20.8 kB]
Get:27 https://us.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [53.3 kB]
Get:28 https://us.archive.ubuntu.com/ubuntu jammy-updates/multiverse DEP-11 Metadata [13.6 kB]
Get:29 https://us.archive.ubuntu.com/ubuntu jammy-updates/multiverse Translation-en [13.6 kB]
Get:30 https://us.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 DEP-11 Metadata [940 B]
Get:31 https://us.archive.ubuntu.com/ubuntu jammy-updates/multiverse DEP-11 48x48 Icons [732 B]
Get:32 https://us.archive.ubuntu.com/ubuntu jammy-backports/main amd64 Packages [68.4 kB]
Get:33 https://us.archive.ubuntu.com/ubuntu jammy-backports/main l386 Packages [60.5 kB]
Get:34 https://us.archive.ubuntu.com/ubuntu jammy-backports/main Translation-en [11.1 kB]
Get:35 https://us.archive.ubuntu.com/ubuntu jammy-backports/main DEP-11 Metadata [17,088 B]
Get:36 https://us.archive.ubuntu.com/ubuntu jammy-backports/main DEP-11 48x48 Icons [9,518 B]
Get:37 https://us.archive.ubuntu.com/ubuntu jammy-backports/main DEP-11 64x64 Icons [11.2 kB]
Get:38 https://us.archive.ubuntu.com/ubuntu jammy-backports/main DEP-11 128x128 Icons [11.2 kB]
Get:39 https://us.archive.ubuntu.com/ubuntu jammy-backports/restricted amd64 DEP-11 Metadata [212 B]
Get:40 https://us.archive.ubuntu.com/ubuntu jammy-backports/restricted DEP-11 48x48 Icons [29 B]
Get:41 https://us.archive.ubuntu.com/ubuntu jammy-backports/restricted DEP-11 64x64 Icons [29 B]
Get:42 https://us.archive.ubuntu.com/ubuntu jammy-backports/restricted DEP-11 128x128 Icons [29 B]
Get:43 https://us.archive.ubuntu.com/ubuntu jammy-backports/universe l386 Packages [10.5 kB]
Get:44 https://us.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [30.1 kB]
Get:45 https://us.archive.ubuntu.com/ubuntu jammy-backports/universe DEP-11 Metadata [103 kB]
Get:46 https://us.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 DEP-11 Metadata [17.9 kB]
Get:47 https://us.archive.ubuntu.com/ubuntu jammy-backports/universe DEP-11 48x48 Icons [10.1 kB]
Get:48 https://us.archive.ubuntu.com/ubuntu jammy-backports/universe DEP-11 64x64 Icons [20.4 kB]
Get:49 https://us.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 c-n-f Metadata [672 B]
Get:50 https://us.archive.ubuntu.com/ubuntu jammy-backports/multiverse amd64 DEP-11 Metadata [212 B]
Get:51 https://us.archive.ubuntu.com/ubuntu jammy-backports/multiverse DEP-11 48x48 Icons [29 B]
Get:52 https://us.archive.ubuntu.com/ubuntu jammy-backports/multiverse DEP-11 64x64 Icons [31.8 kB]
```

**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.

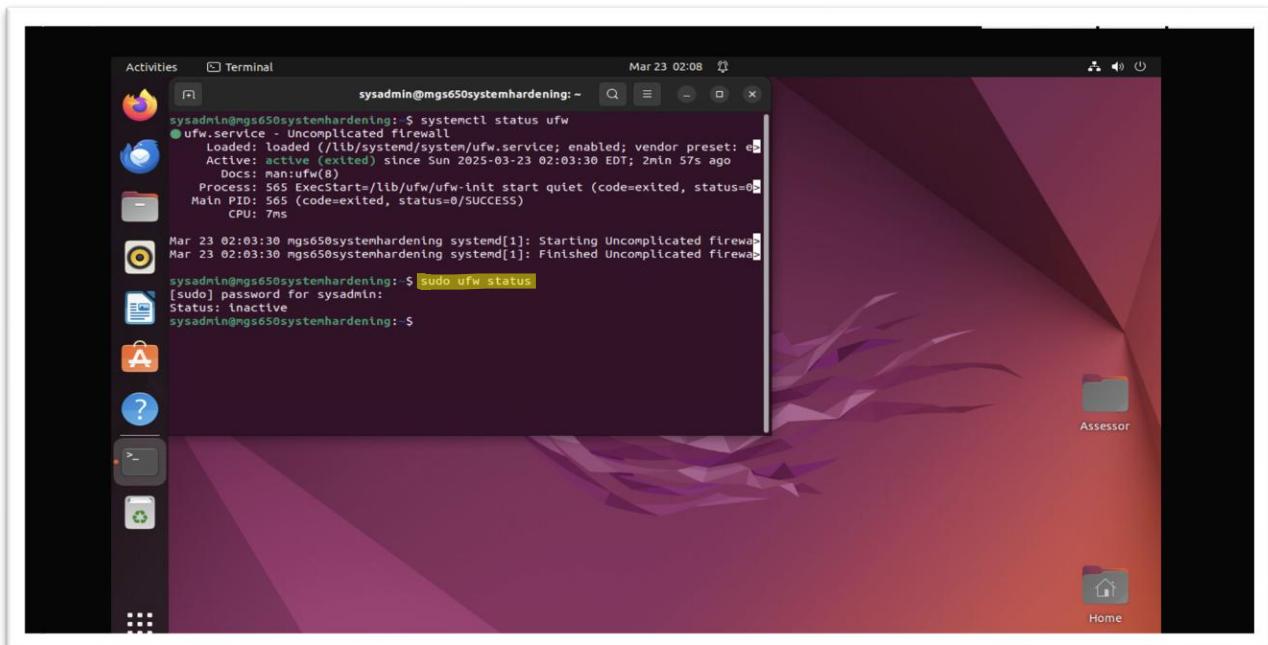
**Figure 2: Screenshot of “sudo apt upgrade” to update installed packages to latest versions.**

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



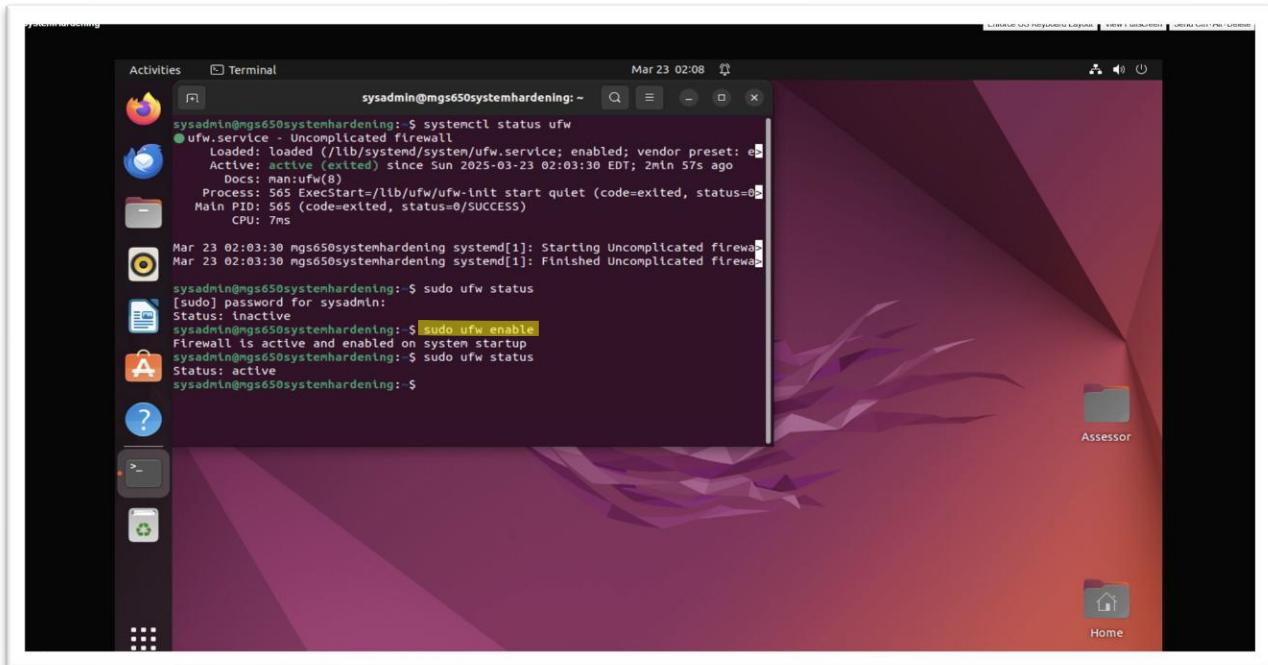
**Figure 3: Screenshot of “`systemctl status ufw`” to check status of ufw service if its active or not.**

- 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.



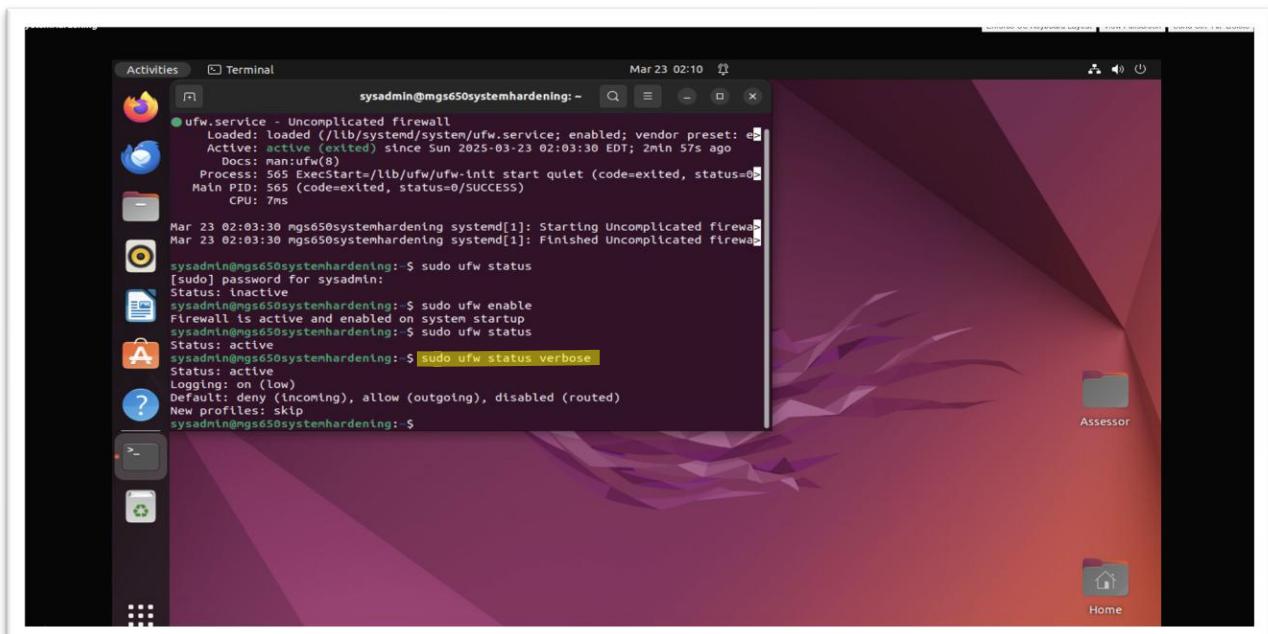
**Figure 4: Screenshot of “`sudo ufw status`” to check status of ufw firewall.**

- 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.

```
Activities Terminal Mar 23 02:11 sysadmin@mgs650systemhardening:~$ sudo lsof -i
COMMAND PID USER FD TYPE DEVICE SIZE/OFF NODE NAME
systemd-r 622 systemd-resolve 13u IPv4 14991 0t0 UDP localhost:domain
systemd-r 622 systemd-resolve 14u IPv4 14992 0t0 TCP localhost:domain (LISTEN)
avahi-dae 728 avahi 12u IPv4 19912 0t0 UDP *:mdns
avahi-dae 728 avahi 13u IPv6 19913 0t0 UDP *:mdns
avahi-dae 728 avahi 14u IPv4 19914 0t0 UDP *:47077
avahi-dae 728 avahi 15u IPv6 19915 0t0 UDP *:50368
NetworkMa 736 root 26u IPv4 15169 0t0 UDP mgs650systemhardening:bootpc->_gateway:bootps
cupsd 832 root 6u IPv6 22332 0t0 TCP ip6-localhost:ipp (LISTEN)
cupsd 832 root 7u IPv4 22333 0t0 TCP localhost:ipp (LISTEN)
redis-ser 847 redis 6u IPv4 15164 0t0 TCP localhost:redis (LISTEN)
redis-ser 847 redis 7u IPv6 15165 0t0 TCP ip6-localhost:redis (LISTEN)
sshd 879 root 3u IPv4 21009 0t0 TCP *:ssh (LISTEN)
sshd 879 root 4u IPv6 21020 0t0 TCP *:ssh (LISTEN)
nginx 895 www-data 6u IPv4 24622 0t0 TCP *:81 (LISTEN)
nginx 896 www-data 6u IPv4 24622 0t0 TCP *:81 (LISTEN)
nginx 897 www-data 6u IPv4 24622 0t0 TCP *:81 (LISTEN)
nginx 898 www-data 6u IPv4 24622 0t0 TCP *:81 (LISTEN)
nginx 901 www-data 6u IPv4 24622 0t0 TCP *:81 (LISTEN)
nginx 902 www-data 6u IPv4 24622 0t0 TCP *:81 (LISTEN)
nginx 903 www-data 6u IPv4 24622 0t0 TCP *:81 (LISTEN)
nginx 905 www-data 6u IPv4 24622 0t0 TCP *:81 (LISTEN)
nginx 906 www-data 6u IPv4 24622 0t0 TCP *:81 (LISTEN)
apache2 919 root 4u IPv6 18885 0t0 TCP *:http (LISTEN)
apache2 919 root 6u IPv6 18889 0t0 TCP *:http-alt (LISTEN)
apache2 920 www-data 4u IPv6 18885 0t0 TCP *:http (LISTEN)
apache2 920 www-data 6u IPv6 18889 0t0 TCP *:http-alt (LISTEN)
apache2 921 www-data 4u IPv6 18885 0t0 TCP *:http (LISTEN)
apache2 921 www-data 6u IPv6 18889 0t0 TCP *:http-alt (LISTEN)
postgres 1010 postgres 5u IPv4 15198 0t0 TCP localhost:postgresql (LISTEN)
postgres 1018 postgres 7u IPv4 15200 0t0 UDP localhost:49926->localhost:49926
nmbd 1023 root 13u IPv4 17962 0t0 UDP *:netbios-ns
nmbd 1023 root 14u IPv4 17963 0t0 UDP *:netbios-dgm
nmbd 1023 root 15u IPv4 17974 0t0 UDP mgs650systemhardening:netbios-ns
nmbd 1023 root 16u IPv4 17975 0t0 UDP 10.200.0.255:netbios-ns
nmbd 1023 root 17u IPv4 17976 0t0 UDP mgs650systemhardening:netbios-dgm
nmbd 1023 root 18u IPv4 17977 0t0 UDP 10.200.0.255:netbios-dgm
postgres 1078 postgres 7u IPv6 15200 0t0 UDP localhost:49926->localhost:49926
```

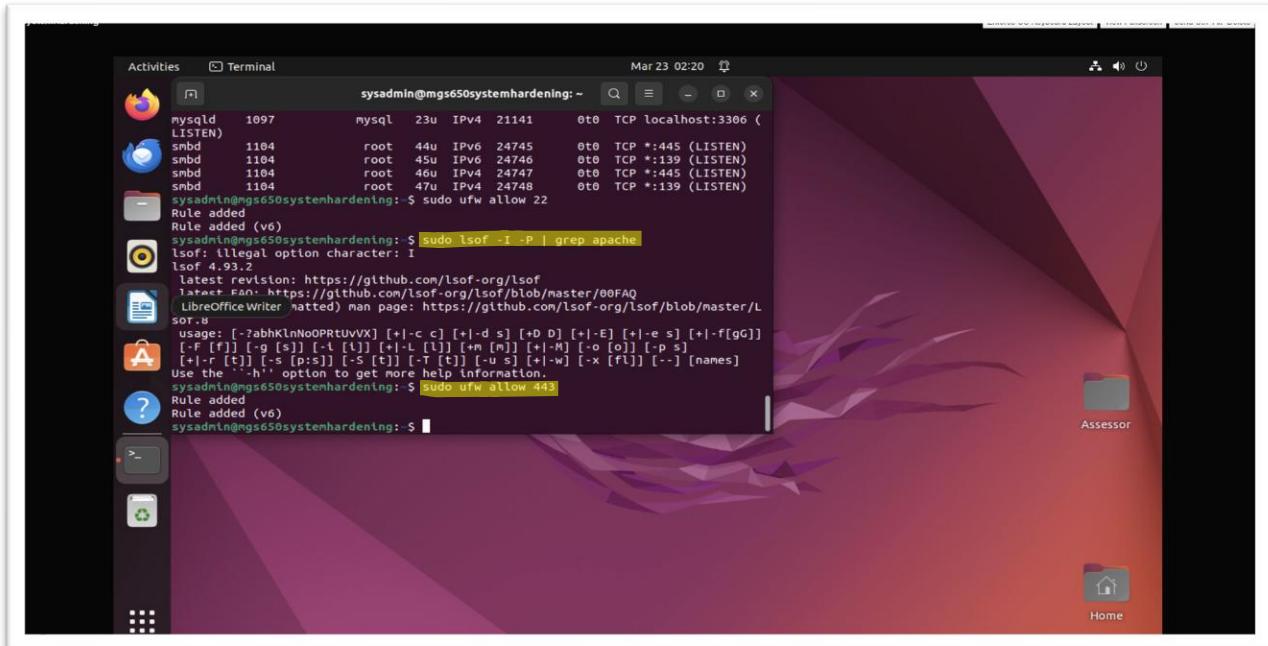
**Figure 7: Screenshot of “sudo lsof -i” to find out all commands running on that network.”**

- 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.

```
Activities Terminal Mar 23 02:14 sysadmin@mgs650systemhardening:~$ sudo ufw allow 22
Rule added
Rule added (v6)
sysadmin@mgs650systemhardening:~$
```

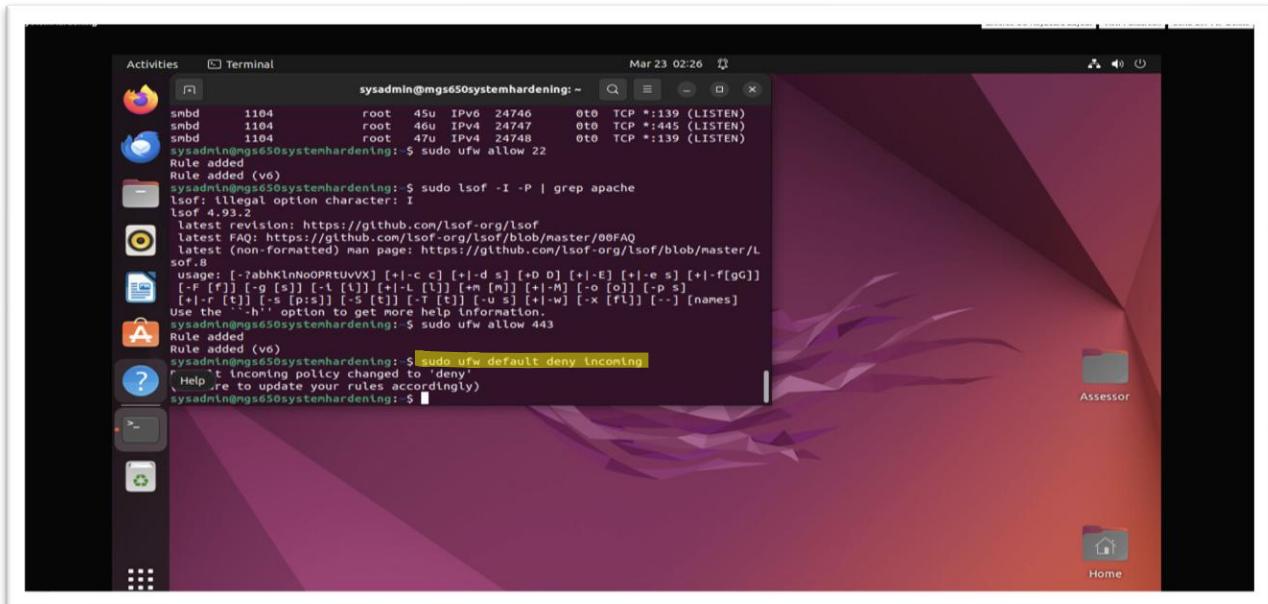
**Figure 8: Screenshot of “sudo ufw allow 22” to allow inbound SSH traffic.**

- After that, to allow all of the traffic 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.



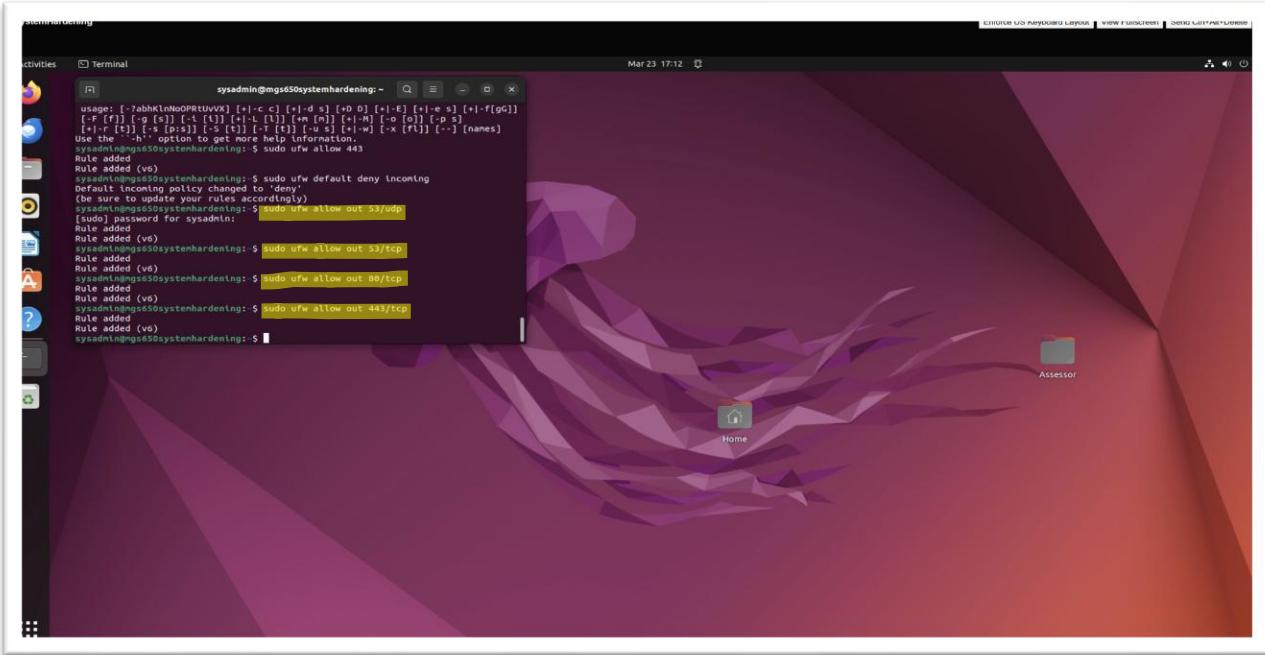
**Figure 9: Screenshot of “sudo ufw allow 443” to allow all traffic for apache server to run.**

- 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.



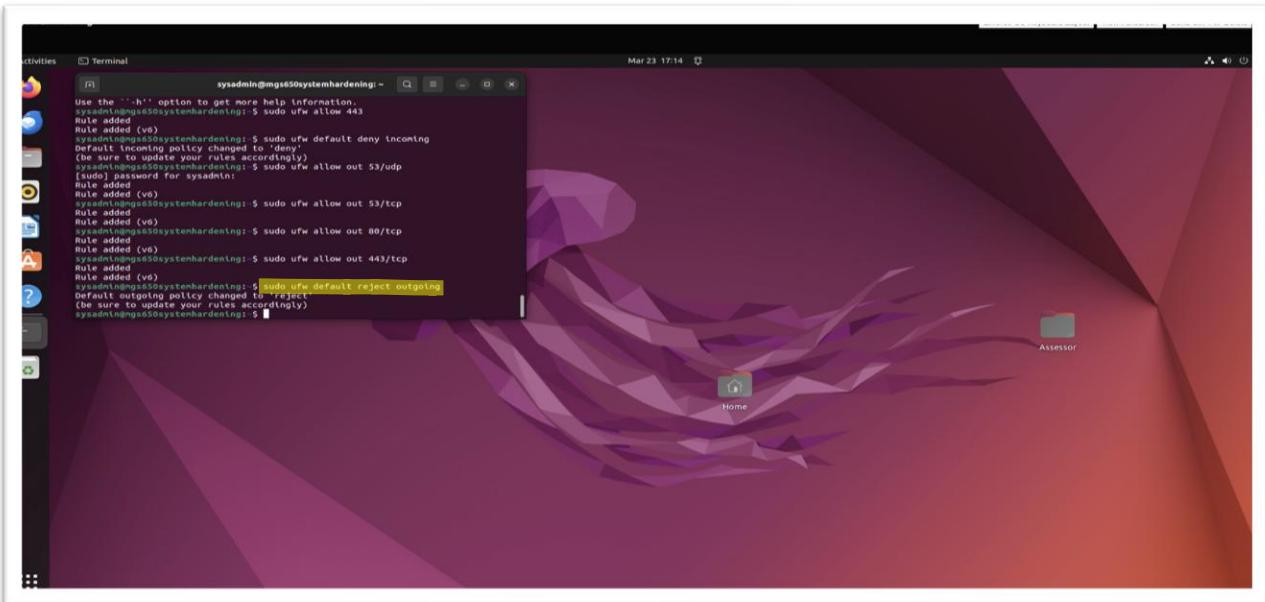
**Figure 10: Screenshot of “sudo ufw default deny incoming” to deny any open ports.**

- 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).



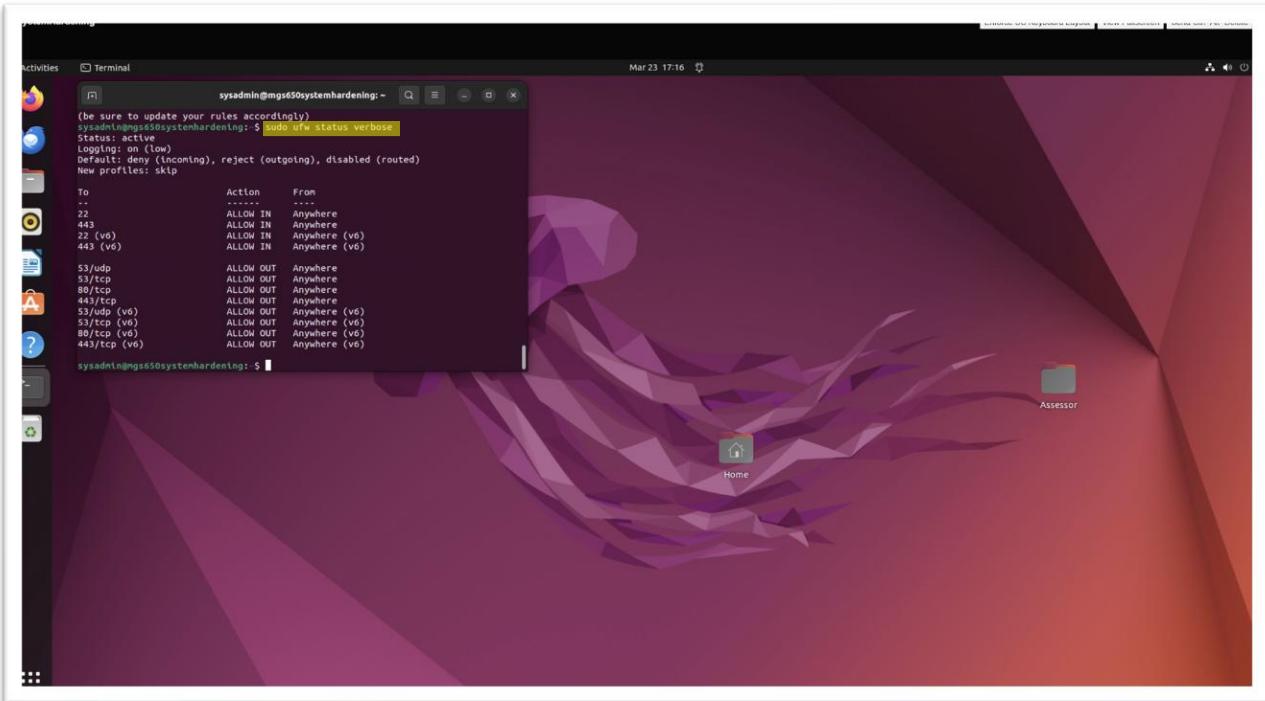
**Figure 11: Screenshot of all different outgoing rules applied in ufw services.**

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



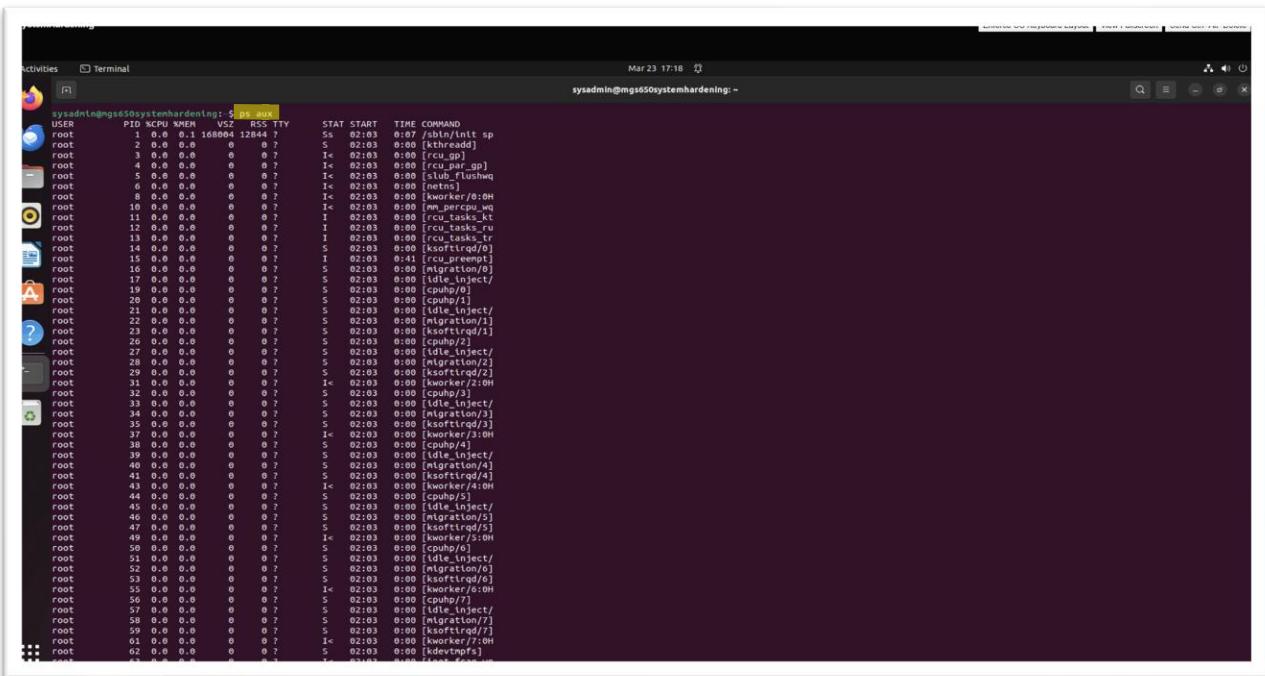
**Figure 12: Screenshot of default reject policy using “sudo ufw default reject outgoing”.**

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

```
activities Terminal Mar 23 17:21 sysadmin@mg1650systemhardening: ~
sysadmin@mg1650systemhardening: $ systemctl status | grep service
● user@mg1650.service
  └─ 0 gnome.SettingsDaemon.MediateKeys.service
      ├─ 1 org.gnome.SettingsDaemon.SmartCard.service
      ├─ 2 org.gnome.SettingsDaemon.Datetime.service
      ├─ 3 org.gnome.SettingsDaemon.Housekeeping.service
      ├─ 4 org.desktop.portal.service
      ├─ 5 org.freedesktop.IBus.session.GNOME.service
      ├─ 6 org.gnome.SettingsDaemon.Keyboard.service
      ├─ 7 pipewire-media-session.service
      ├─ 8 org.gnome.SettingsDaemon.A11ySettings.service
      ├─ 9 pulseaudio.service
      ├─ 10 org.gnome.SettingsDaemon.Wacom.service
      ├─ 11 org.gnome.SettingsDaemon.Sharing.service
      ├─ 12 org.gnome.SettingsDaemon.Color.service
      ├─ 13 org.gnome.SettingsDaemon.ScreenSaverProxy.service
      ├─ 14 org.gnome.SettingsDaemon.ScreenSaverifications.service
      ├─ 15 org.gnome.SettingsDaemon.Power.service
      ├─ 16 org.gnome.Shell@wayland.service
      ├─ 17 org.gnome.SettingsDaemon.XSettings.service
      ├─ 18 org.gnome.SettingsDaemon.BluetoothSound.service
      ├─ 19 pipewire.service
      ├─ 20 org.gnome.SettingsDaemon.akkill.service
      └─ 21 trashed-fs-3.service
      ├─ 22 gpts-volume-monitor.service
      ├─ 23 xdg-permission-store.service
      ├─ 24 evolution-calendar-factory.service
      ├─ 25 xdg-desktop-portal-gnome.service
      └─ 26 icon-theme
      ↳ 1885 /usr/libexec/dconf-service
          ├─ 1886 gnome-session-manager@ubuntu.service
          ├─ 1887 /usr/libexec/gnome-session-binary --systemd-service --session=ubuntu
          ├─ 1888 gvfs-daemon.service
          ├─ 1889 evolution-source-registry.service
          ├─ 1890 gvfs-udisks2-volume-monitor.service
          snapd
          ├─ 1891 gnome-terminal-server.service
          ├─ 1892 grep --color=auto.service
          ├─ 1893 gvfs-ghetto2-volume-monitor.service
          ├─ 1894 gvfs-passion-volume-monitor.service
          ├─ 1895 xdg-desktop-portal-gtk.service
          ├─ 1896 gvfs-metadata.service
          └─ 1897 gvfs
          ↳ 1846 /usr/libexec/goa-identity.service
          ├─ 1847 evolution-addressbook-factory.service
          ├─ 1848 gvfs-ntp-volume-monitor.service
          ├─ 1849 gvfs-selinux-volume-monitor.service
          └─ 1850 org.gnome.Terminal.service
          ├─ 1851 apache2.service
          ├─ 1852 open-vn-tools.service
          ├─ 1853 packagekit.service
          └─ 1854 autoread-indicator.service
```

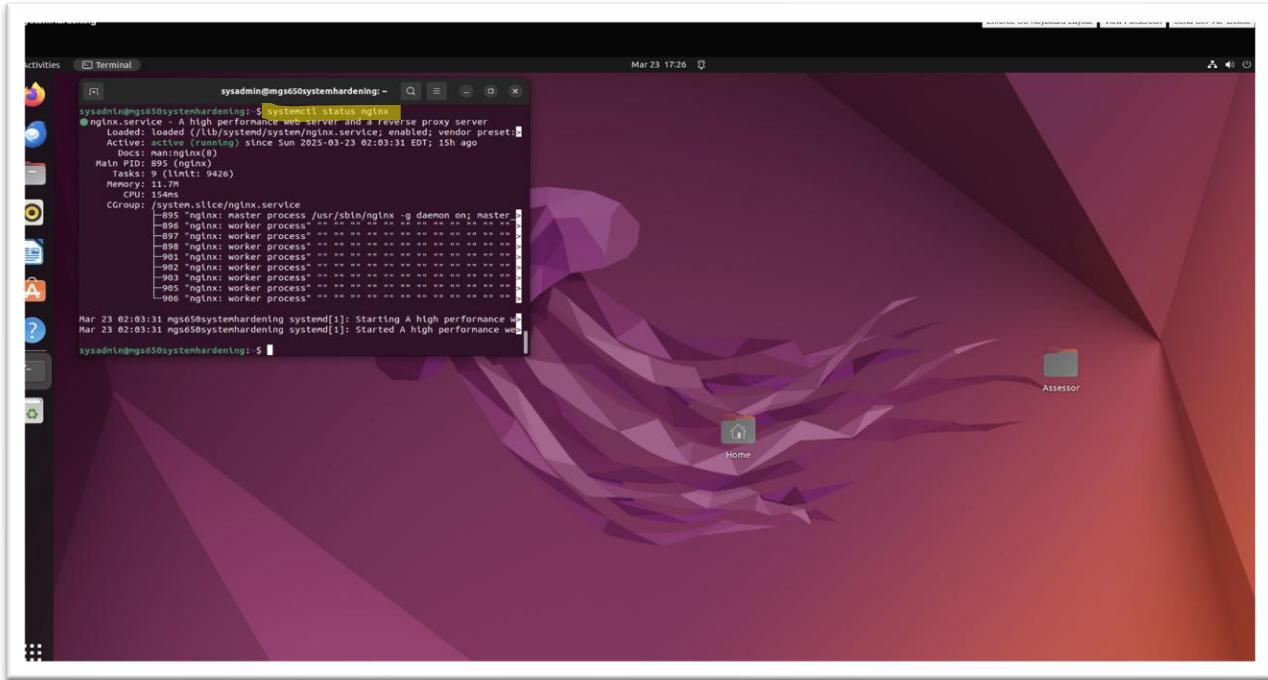
**Figure 15: Screenshot of “`systemctl status | grep service`” to check for all services in system init.**

- 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.

A screenshot of a Linux desktop environment, likely Ubuntu, featuring a purple and orange low-poly wallpaper. A terminal window titled "Terminal" is open in the top-left corner, showing the command "sudo apt purge samba" being run. The terminal output indicates that several packages are being removed, including attr, libcephfs2, libflashrom, libftdi1-2, libgphoto, libgRPC, libgxfs, libglusterfs, libgudev1, liblwtn13, librados2, librdmacm1, liburing2, libwrap0, libxepackend-fde1-1.0.1, and samba-vfis-modules. It also shows that tdb-tools are being upgraded. A warning message from dpkg states that the directory '/var/lib/samba/printers/x64' is not empty and thus cannot be removed. The desktop environment includes a dock with icons for Home, Dash, Activities, Terminal, and a file manager, as well as a system tray icon for "Assessor".

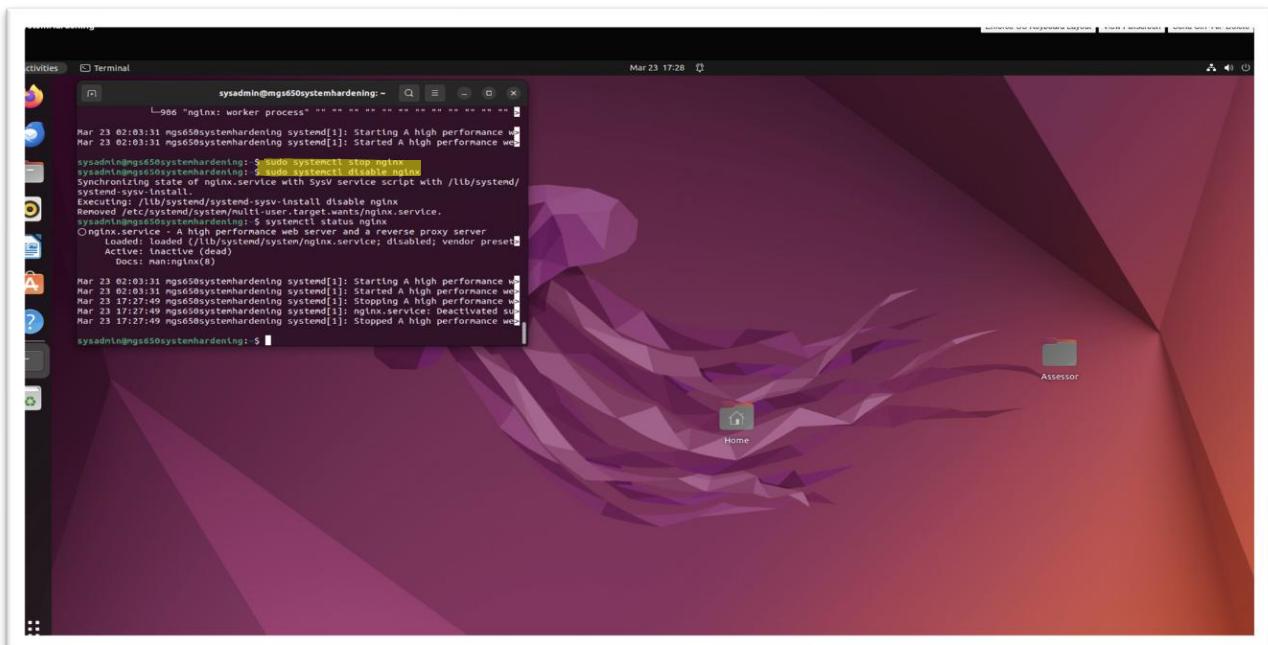
**Figure 16:** Screenshot of “`sudo apt purge samba`” to uninstall Samba’s running services.

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



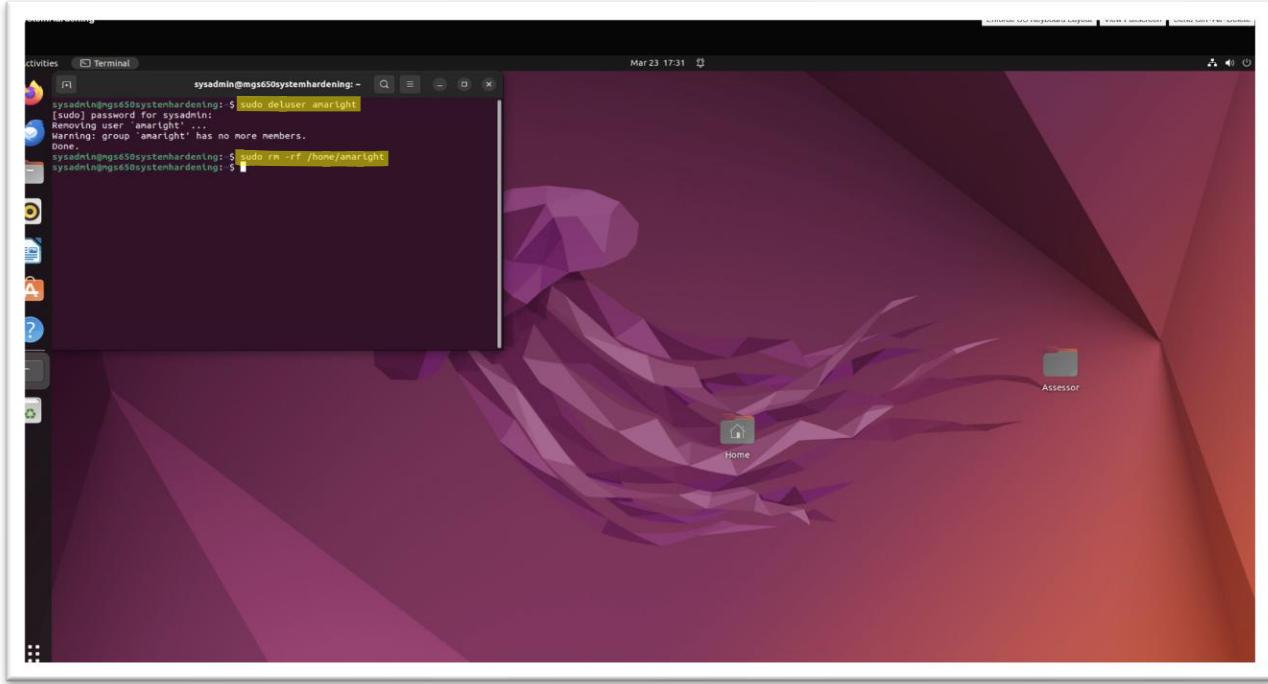
**Figure 17: Screenshot of “systemctl status nginx” to check if nginx service is active or not.**

- 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.



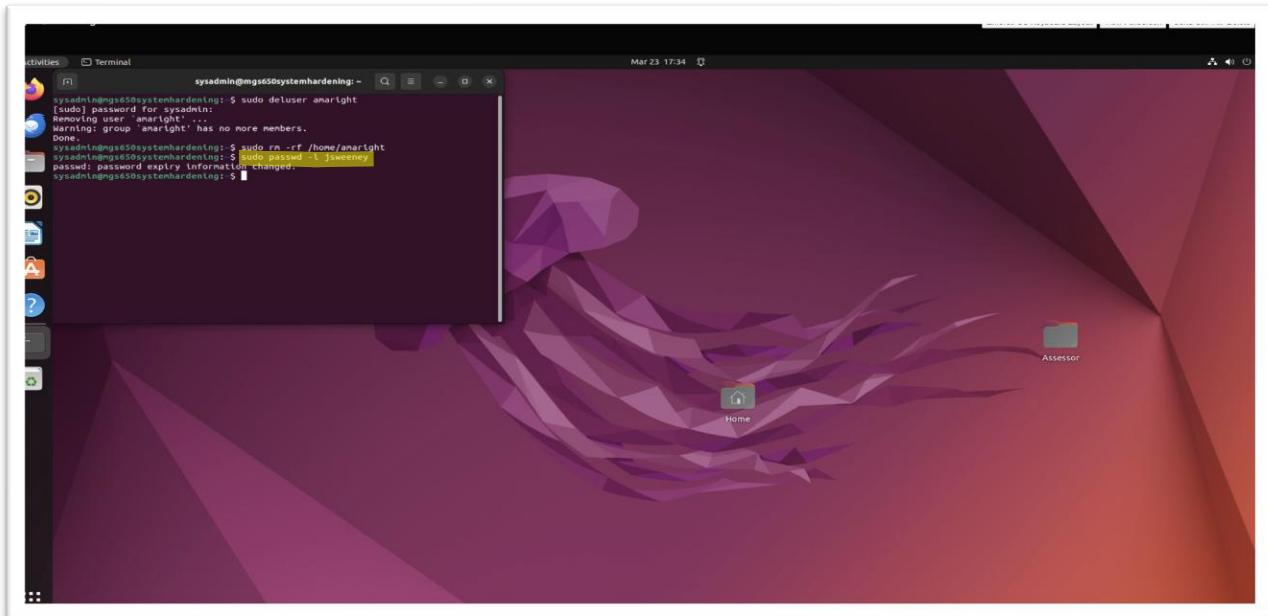
**Figure 18: Screenshot of “sudo systemctl disable nginx” to disable nginx services.**

- 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.

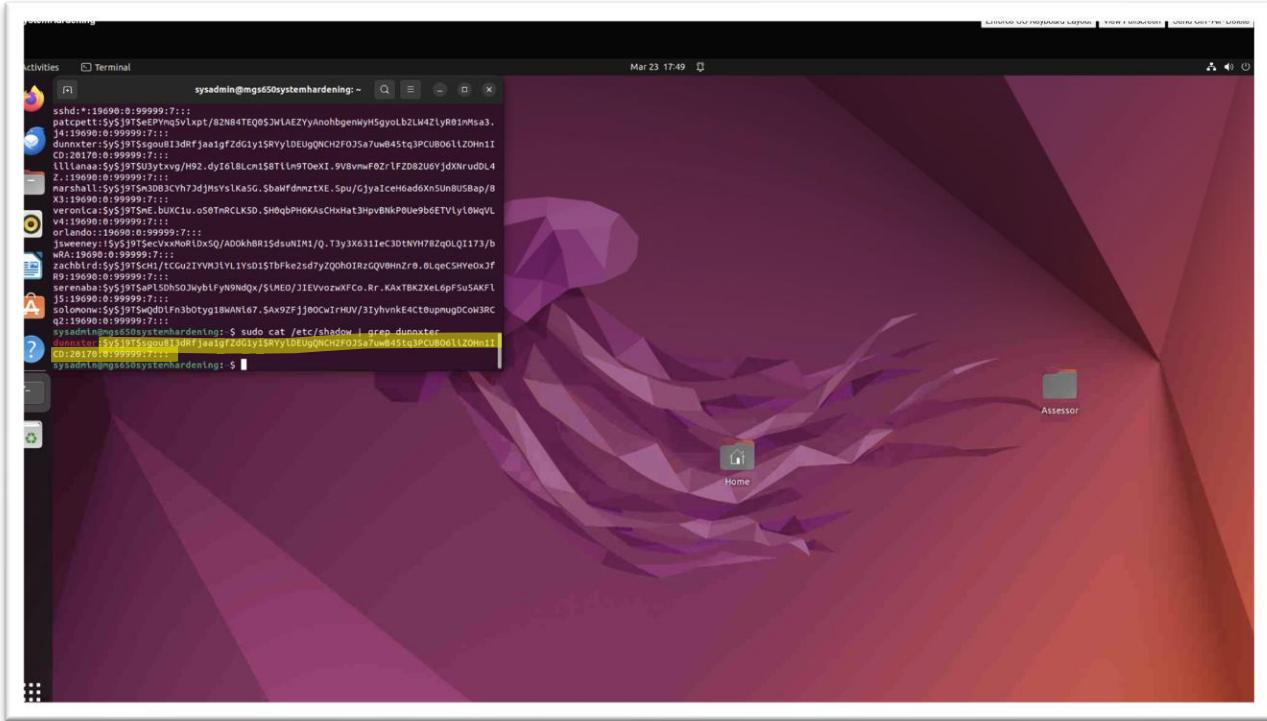
```
sysadmin@mgs650systemhardening:~$ sudo cat /etc/shadow
root::191010:0:99999:7:::
dunnxter::191010:0:99999:7:::
blntr::191010:0:99999:7:::
sys::191010:0:99999:7:::
sync::191010:0:99999:7:::
man::191010:0:99999:7:::
lp::191010:0:99999:7:::
news::191010:0:99999:7:::
uuclipr::191010:0:99999:7:::
proxy::191010:0:99999:7:::
www::191010:0:99999:7:::
backup::191010:0:99999:7:::
llst::191010:0:99999:7:::
print::191010:0:99999:7:::
nobody::191010:0:99999:7:::
systemd-network::191010:0:99999:7:::
systemd-resolve::191010:0:99999:7:::
systemd-timesync::191010:0:99999:7:::
syslog::191010:0:99999:7:::
kern::191010:0:99999:7:::
tskit::191010:0:99999:7:::
uuid::191010:0:99999:7:::
systemd-oom::191010:0:99999:7:::
cups::191010:0:99999:7:::
avahi-autoid::191010:0:99999:7:::
usbmux::191010:0:99999:7:::
dnsmasq::191010:0:99999:7:::
kernel::191010:0:99999:7:::
avahi::191010:0:99999:7:::
cups-pk-helper::191010:0:99999:7:::
rfkill::191010:0:99999:7:::
whopple::191010:0:99999:7:::
ssdd::191010:0:99999:7:::
speech-dispatcher::191010:0:99999:7:::
nn-outputs::191010:0:99999:7:::
sound::191010:0:99999:7:::
colord::191010:0:99999:7:::
geoclue::191010:0:99999:7:::
polkit::191010:0:99999:7:::
gnome-initial-setup::191010:0:99999:7:::
hplip::191010:0:99999:7:::
gdm::191010:0:99999:7:::
sysfs-sysfs-mountpointSVS1kLpbDYoPSeM2F/JwOaPwUvOkGslSMaVrJzEOU9zQ8mJ7XlHxD.:19690:0:99999:7:::
fwupd-refresh::19690:0:0:99999:7:::
redis::19690:0:0:99999:7:::
postgresql::19690:0:0:99999:7:::
mysql::19690:0:0:99999:7:::
sshd::19690:0:0:99999:7:::
patcett:sys19TSeEPYng5v1xpt:/8ZN84TE0D52HlAEZYeyanhbgemiyH5gyolbzLW4ZlyR01msa3:14:19690:0:0:99999:7:::
```

**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.

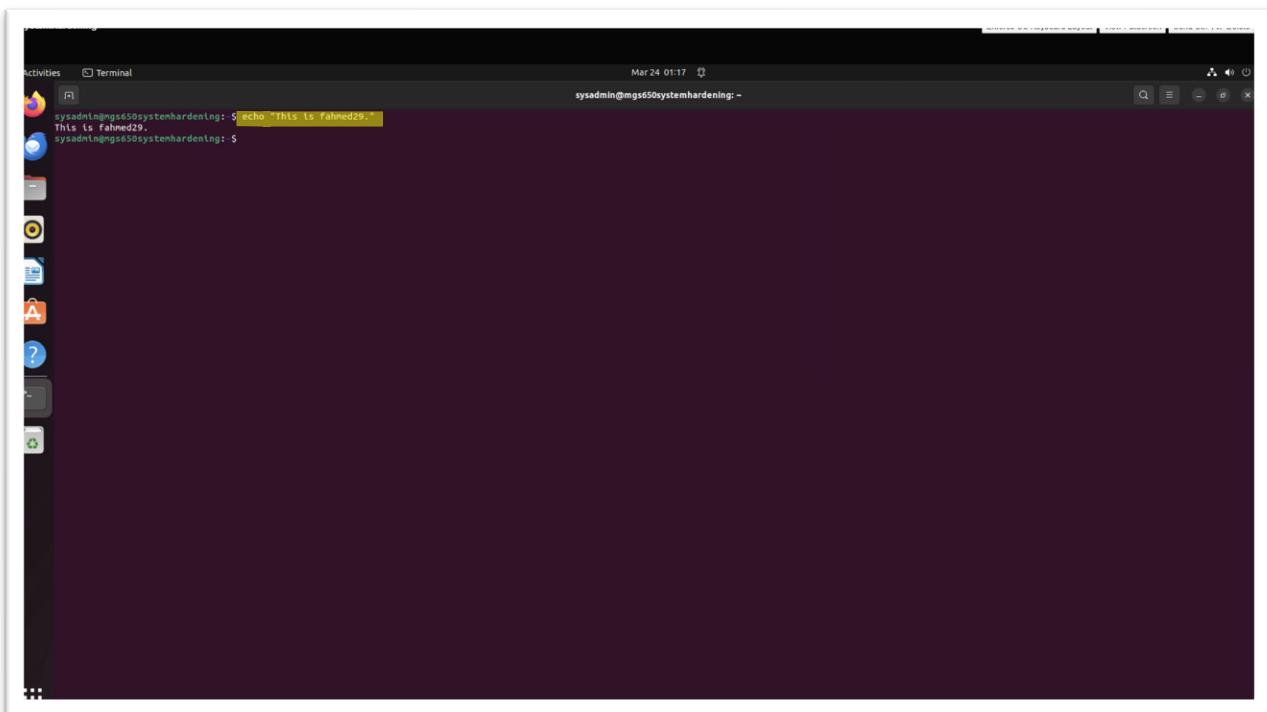
```
sysadmin@mgs650systemhardening:~$ sudo passwd dunnxter
New password:
Retype new password:
passwd: password updated successfully
sysadmin@mgs650systemhardening:~$
```

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



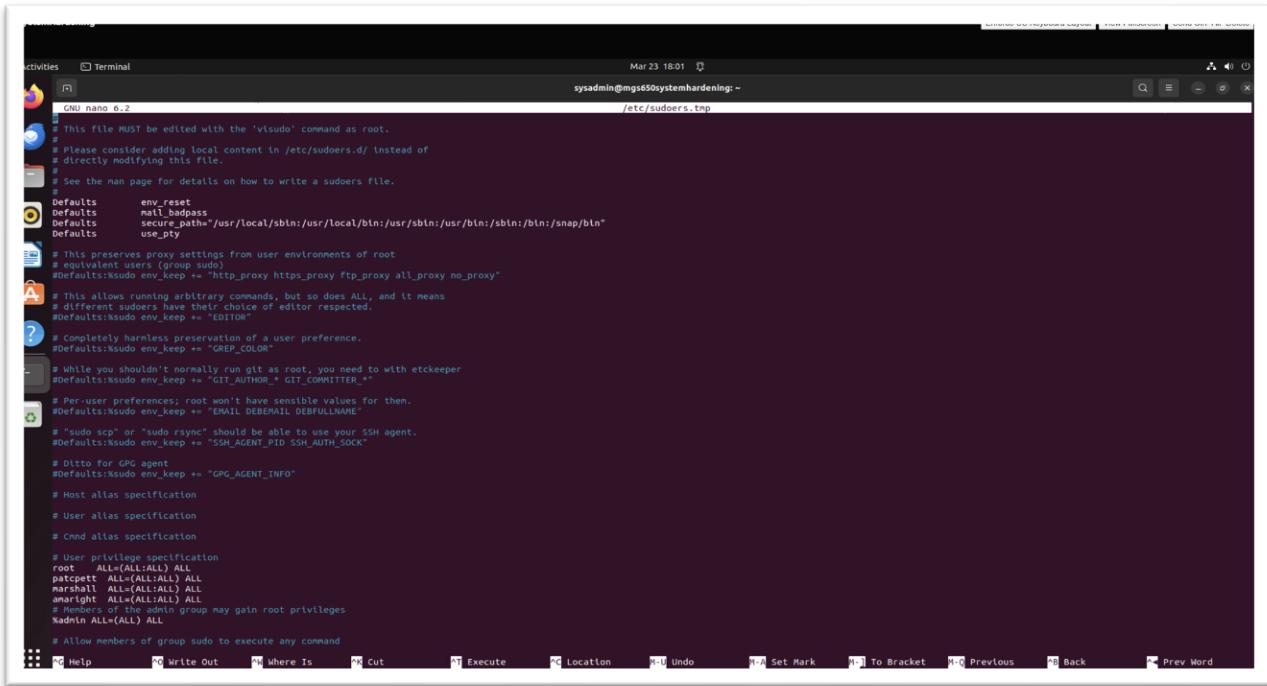
**Figure 23: Screenshot of hash password of user “dunnxter”.**

- 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.



```

activities Terminal Mar 23 18:01
sysadmin@mg1650systemhardening: ~ /etc/sudoers.tmp

GNU nano 6.2
# This file MUST be edited with the 'visudo' command as root.
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
# See the man page for details on how to write a sudoers file.

Defaults env_reset
Defaults mail_badpass
Defaults secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin"
Defaults use_pty

# This preserves proxy settings from user environments of root
# equivalent users (group sudo)
#Defaults:sudo env_keep += "http_proxy https_proxy ftp_proxy all_proxy no_proxy"

# This allows running arbitrary commands, but so does ALL, and it means
# different sudoers have their choice of editor respected.
#Defaults:sudo env_keep += "EDITOR"

# Completely harmless preservation of a user preference.
#Defaults:sudo env_keep += "GREP_COLOR"

# While you shouldn't normally run git as root, you need to with etckeeper
#Defaults:sudo env_keep += "GIT_AUTHOR GIT_COMMITTER"

# Per-user preferences; root won't have sensible values for them.
#Defaults:sudo env_keep += "EMAIL DEBEMAIL DEBFULLNAME"

# "sudo scp" or "sudo rsync" should be able to use your SSH agent.
#Defaults:sudo env_keep += "$SSH_AGENT_PID SSH_AUTH_SOCK"

# Ditto for GPG agent
#Defaults:sudo env_keep += "GPG_AGENT_INFO"

# Host alias specification

# User alias specification

# Cmnd alias specification

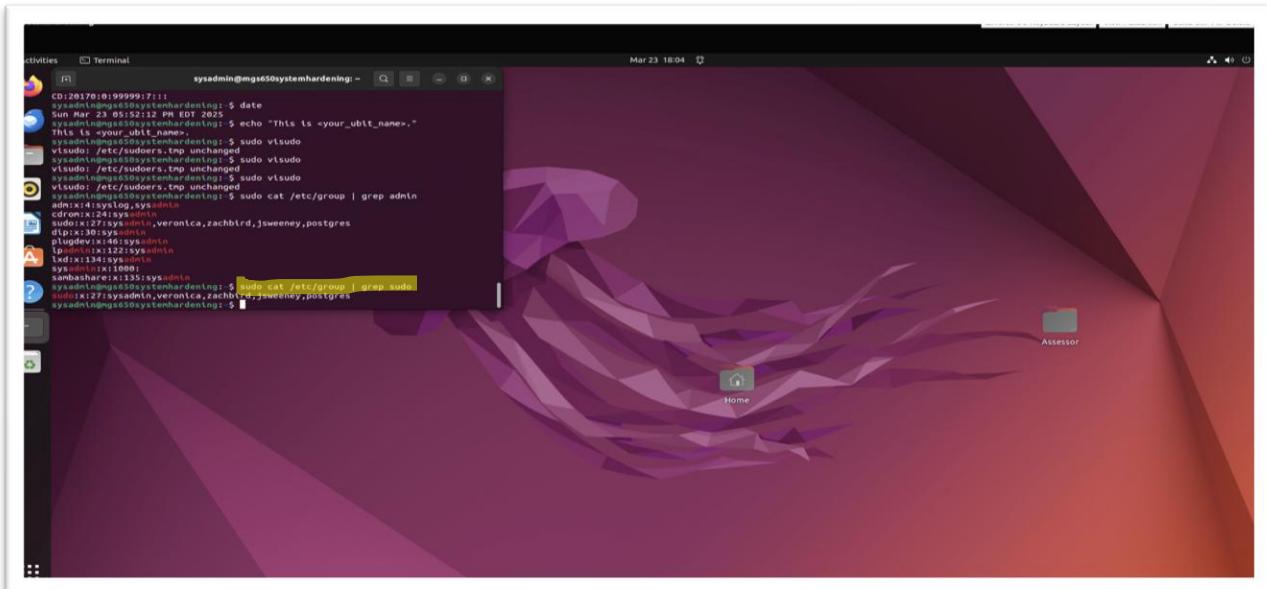
# User privilege specification
root    ALL=(ALL) ALL
patrick ALL=(ALL) ALL
marshall ALL=(ALL) ALL
anaright ALL=(ALL:ALL) ALL
%wheel   ALL=(ALL) ALL
# Members of the wheel group may gain root privileges
%admin  ALL=(ALL) ALL

# Allow members of group sudo to execute any command

```

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

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



```

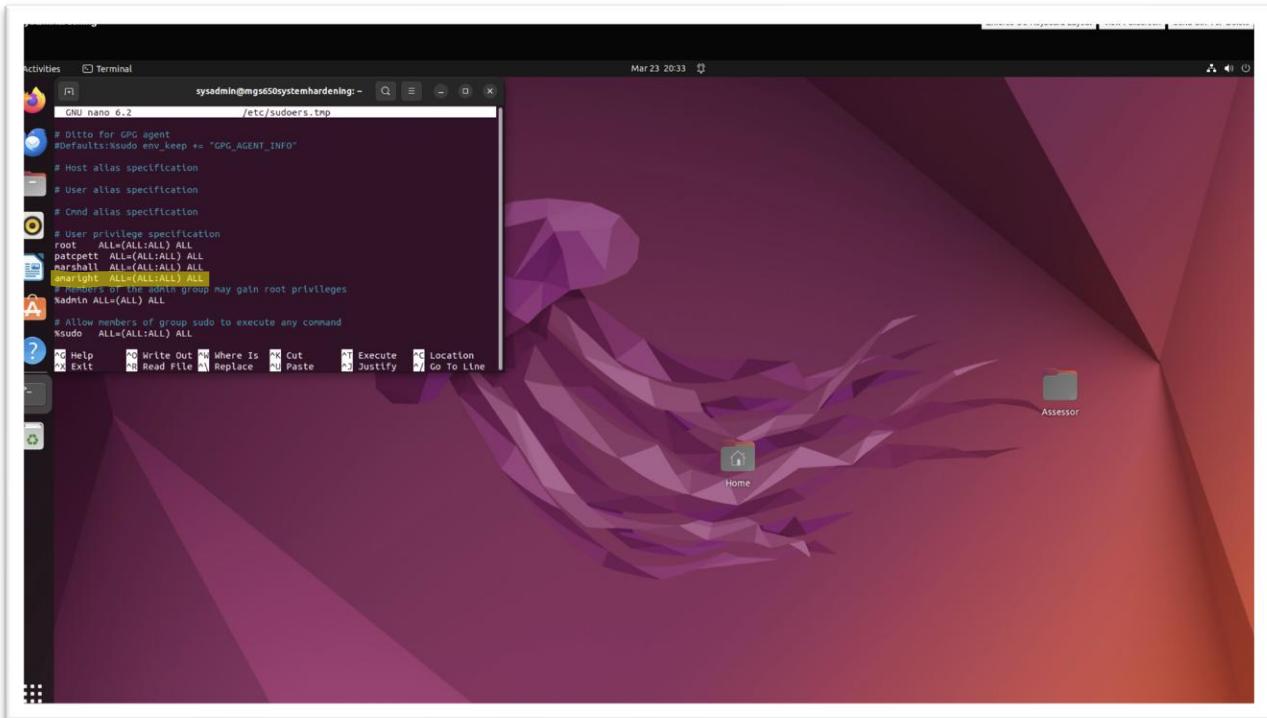
activities Terminal Mar 23 18:04
sysadmin@mg1650systemhardening: ~

CD:20170108:099999:17:::
sysadmin@mg1650systemhardening: ~ date
Sun Mar 23 18:04:00 UTC 2014
sysadmin@mg1650systemhardening: ~ $ echo "This is <your_ubit_name>."
This is <your_ubit_name>.
sysadmin@mg1650systemhardening: ~ $ sudo visudo
visudo: /etc/sudoers.tmp unchanged
sysadmin@mg1650systemhardening: ~ $ sudo visudo
visudo: /etc/sudoers.tmp unchanged
sysadmin@mg1650systemhardening: ~ $ sudo visudo
visudo: /etc/sudoers.tmp unchanged
adm:x:4:syslog,sysadmin
cdrom:x:23:cdrom
dialout:x:24:dialout
sudo:x:27:sysadmin,veronica,zachbird,jsweeney,postgres
dipix:30:sysadmin
plugdev:x:112:sysadmin
lxdi:x:134:sysadmin
sanbshare:x:135:sysadmin
sysadmin@mg1650systemhardening: ~ $ sudo cat /etc/group | grep sudo
sysadmin:x:27:sysadmin,veronica,zachbird,jsweeney,postgres
sysadmin@mg1650systemhardening: ~ $ 

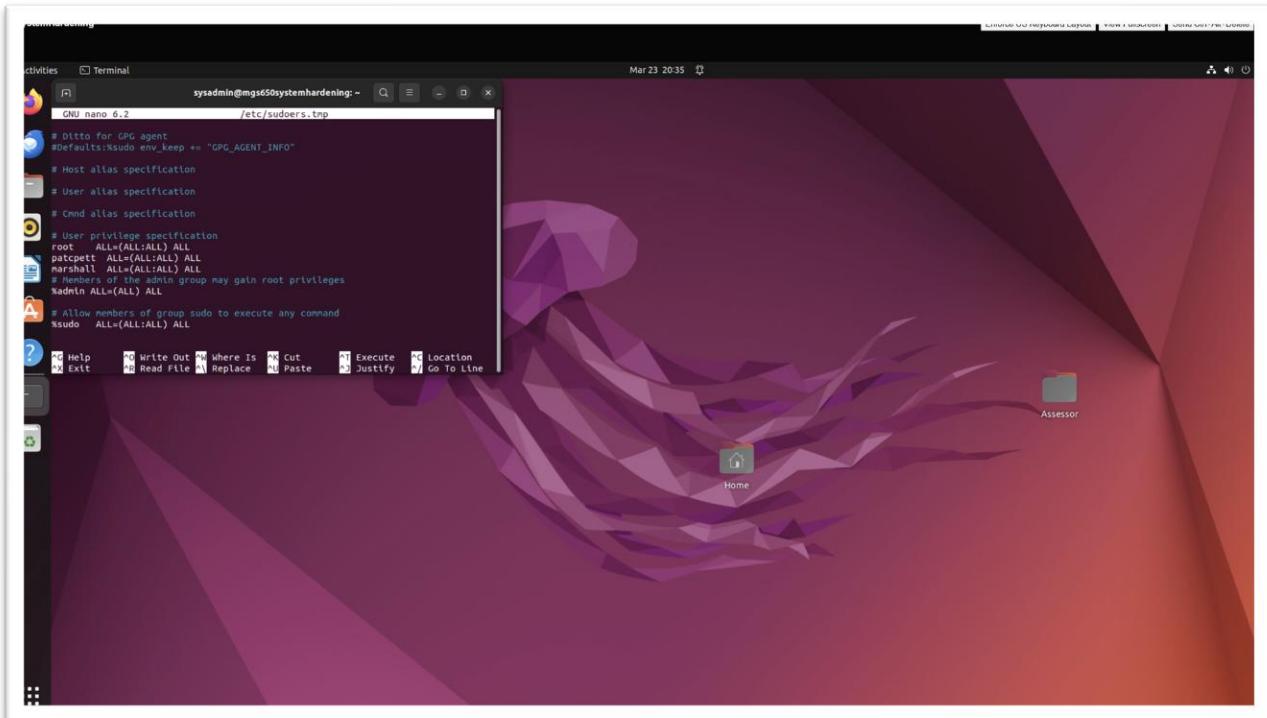
```

**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 highlighted in figure 27 and removed in figure 28.

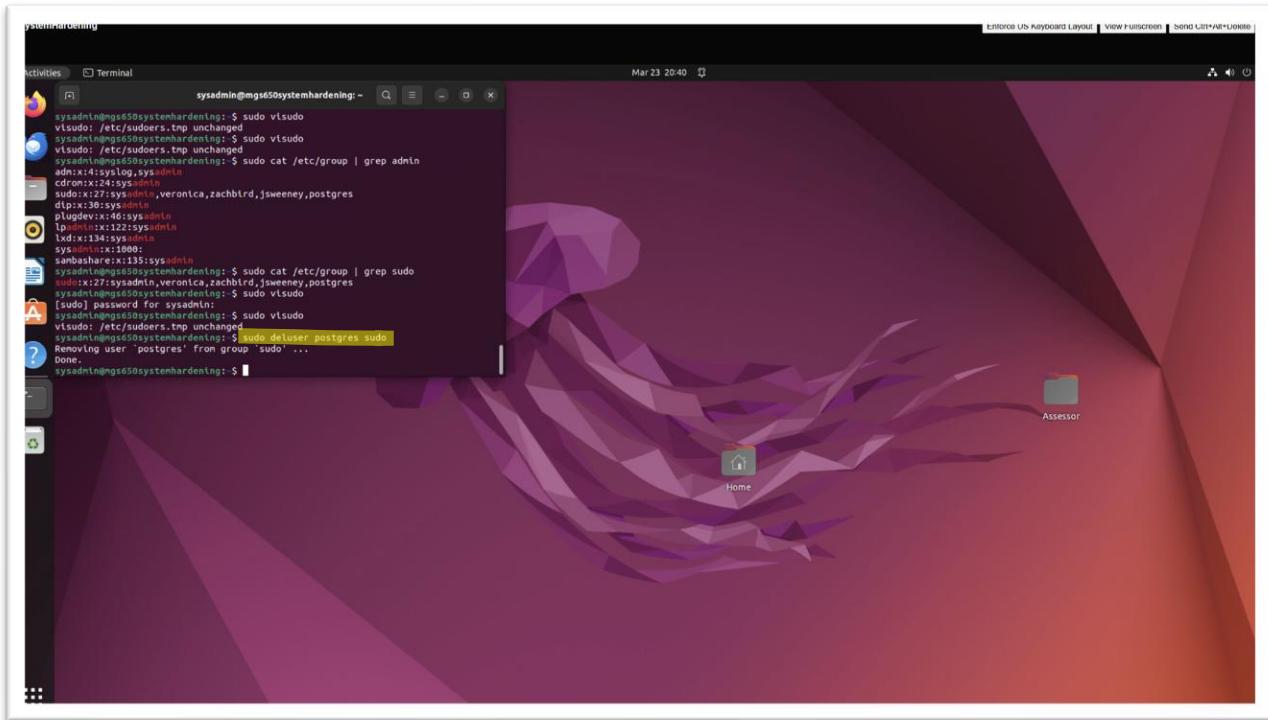


**Figure 27: Screenshot of all users with sudo privileges.**

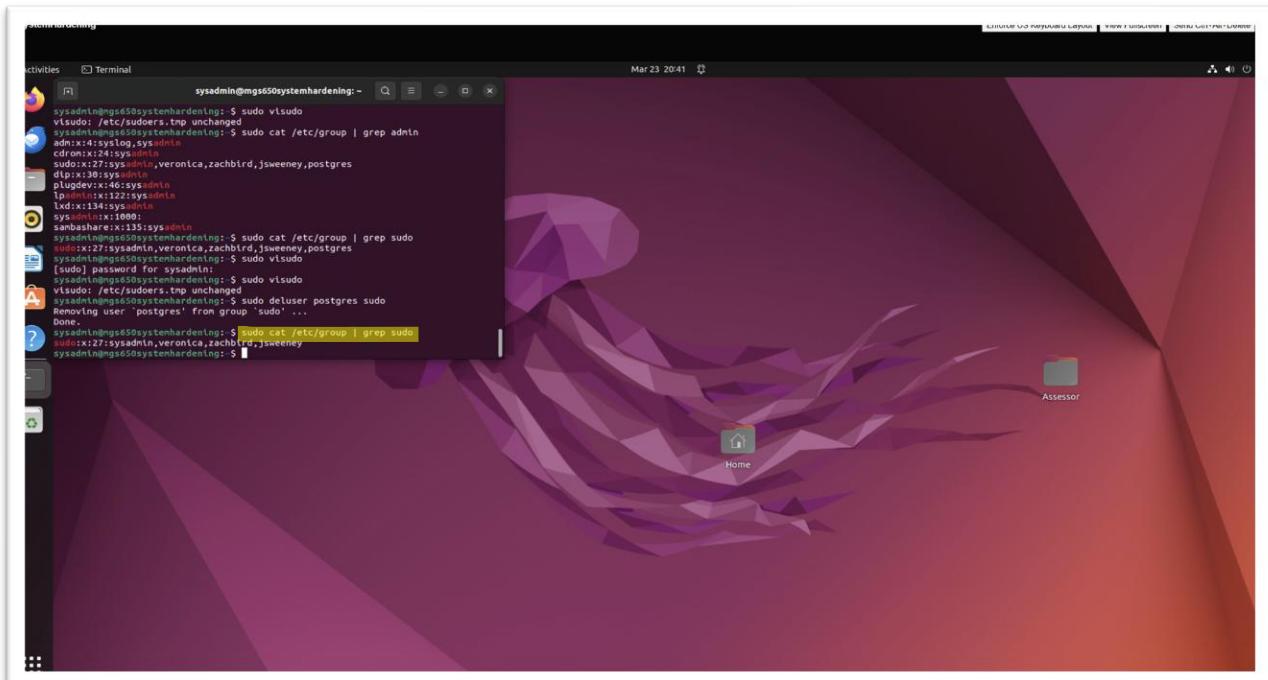


**Figure 28: Screenshot of deleting amaright user in sudo privileges.**

- 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.**