Server System Management - Linux

Lab 04 – User management

# 

# Lab 04: User Management

## Section 1 : Play with yourself

1. Log in on your virtual machine as your regular user (not-root)

ssh!

A screenshot of a phone

Description automatically generated

2. Use grep to find your UID in /etc/passwd. You should remember this files contains a list of all users on your system.

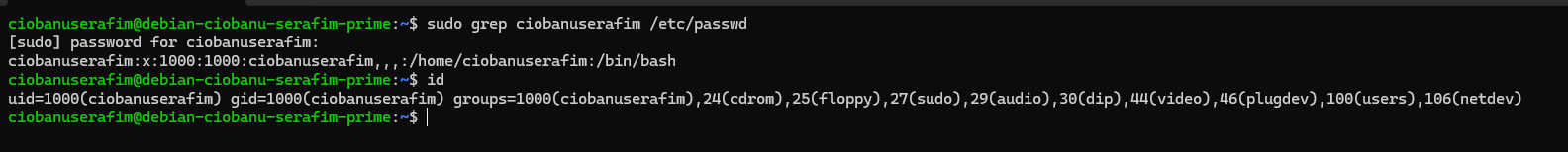
sudo grep ciobanuserafim /etc/passwd

A screenshot of a computer

Description automatically generated

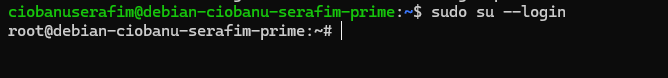
3. Check if the user id you found matches the id reported by the id tool. Just execute id to find out! (QUIZ)

Yes it matches



4. Become root with su –login

sudo su --login



5. Read the man page for the adduser command.

Add your regular user (also called user for the remainder of the document) to the group backup using the adduser command.

usermod -aG backup $USER

A screen shot of a computer program

Description automatically generated

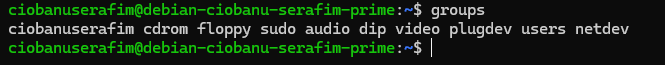
6. Exit your su session. You will be working on behalf of your regular user again now.

exit

7. Type the groups command and study the output; try to guess what it tells you.

Now type man groups to check whether you were right. The answer is in the NAME section at the top.

groups



It will give you output about what the user’s groups are

8. use cat /etc/group and man group

What does the /etc/group file contain? Explain in your own words

A screenshot of a computer

Description automatically generated

It will just present you with groups that are present on the system, some of them can be added to the user, some of them are for daemons

9. Use grep to filter out the lines containing your username user from /etc/group. Now explain in your own words what information you just obtained:

It gives you the groups that the user is parrt of. Preetty much the same as groups, but probably containting more information regarding it.

A screen shot of a computer

Description automatically generated

10. Compare the output of this grep with the output of the groups command in step 7. Do you notice a difference, apart from a different formatting? (QUIZ)

It includes the backup group whereas groups doesn’t

That is because you need to actually relogin for the group to appear (generaly, as simple login to root and back would work)

11. Log out and log in again. Now type the groups command again. This time it does not contradict the grep command: you are now a member of the backup group (see step 5).

A screenshot of a computer program

Description automatically generated

**Note**

In general as long as you are logged in, your group membership is not updated.

That is what I said previously

12. Use su --login to become root again. run apt update && apt install sudo. sudo is not installed by default, although often one might think so because a lot of other packages depend on it and will install it with them.

13. Type nano /etc/group.

Exit the editor again. Now type vigr.

This will have done the same thing, with an important advantage: another administrator cannot simultaneously edit the file and destroy your changes.

A computer screen with white text

Description automatically generated

14. Use vigr to edit the line for group backup: remove your username.

Add your username to the group sudo instead, in exactly the same syntax as you have seen in the backup group line you just edited.

Save your changes and exit the editor.

You have now done the same thing you did in step 5, but by editing the group file directly!

A screen shot of a computer

Description automatically generated

15. Log out of your su session so you are working as regular user again.

exit

16. This time type the id command and use id --help and man id to check if you guessed correctly what it does.

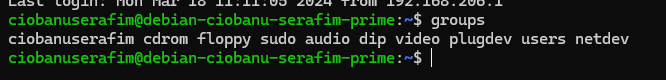
A black screen with white text

Description automatically generated

Are you already a member of the sudo group? If not, how to update your group membership so id and groups tell you you are indeed a member of the sudo group?

relogin

17. Make sure your regular user is indeed a member of sudo



18. Now, as regular user, type a find command to search for all files on your system, under /.

Redirect stderr to /tmp/errors using the 2> construct.

We are going to search for directories that our user cannot access due to permissions.

find / 2>/tmp/errors

19. Use less to read /tmp/errors and pick a directory that find was not permitted to read. Try to ls that file. You should receive an error message.

A screenshot of a computer

Description automatically generated

ls /lost+found/

A black background with white and green text

Description automatically generated

## Section 2 : Fun with sudo

20. This time prefix your command with sudo. This will execute it as root (so you do not have to use su --login). (SCREENSHOT)

You will need to enter **your (own) password**. Because you are now a member of the sudo group (see step 14), sudo entrusts you with the power to run commands as root (without needing to know the actual root password).

This time you should not receive a permission denied error message because root is now executing the command.

**Note**

sudo makes system administration more manageable because it logs actions, unlike commands executed in a root shell.

Even better: the visudo command opens /etc/sudoers myriad of fine-grained administration command granting an entire group (sudo) root access.

And the man page for sudoers offers a lot of customization possibilities.

A screen shot of a computer program

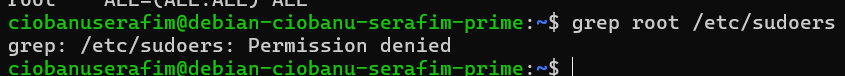
Description automatically generated

21. Use sudo visudo to find the line that permits the entire group sudo to become root using sudo.

Write a grep command that filters out this single line from /etc/sudoers.

Can any user read this file or do you need to use sudo to have permission to read /etc/sudoers? (QUIZ)

No, you need to have root permissions



llustrate by showing the permissions of this file using stat.

sudo grep root /etc/sudoers

A screen shot of a computer program

Description automatically generated

sudo stat /etc/sudoers

A screen shot of a computer

Description automatically generated

22. Now study su --help and man su to see if you can also get it to execute just a single command as root.

su -c ‘COMMAND’

A black screen with green text

Description automatically generated

23. What is the difference, apart from the different command used?

sudo makes use of the groups to understand who can exectute the commandd

24. Now the other way around: try to get sudo to offer you a root shell (the way we used su --login up till now).

If necessary, read the man page, or be creative!

A black background with green and white text

Description automatically generated

sudo -i

It will login you into a root shell, and I guess it is the same as su –login

25. Go back to your regular user if you haven’t already. Use the command man 5 passwd to show you the man page for the /etc/passwd file

**Note**

If you omit the 5, you will get the man page for the passwd program (to update your password) instead.

How did I find the man pages? Simple:

find /usr/share/man -name '\*passwd\*'

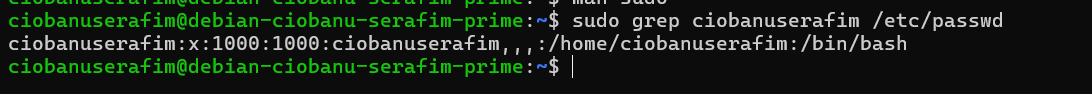
you can also look for any man page containing the passwd keyword by using **man -k passwd**

man man à shows the different chapters (1 for shell commands, 5 for file formats)

You will quickly discover the manual sections passwd has a manual page in. We are fortunate to stumble upon the man page for the passwd tool when just typing man passwd.

26. Use grep to filter out your username from /etc/passwd. Compare this line with the output of the id command.

What is your primary group id, referencing back to the man page ?



The group primary ID is 1000

27. Observe the effects of vipw and nano /etc/passwd. Compare with vigr and nano /etc/group you did before.

They are supposed to do the same thing

28. By now you know a user can be a member of multiple groups in /etc/group, and you also know there is a primary group listed in /etc/passwd.

You probably have also already observed, from the output of the id command, that this primary group has an id that equals your user id.

Third observation: sudo recognizes your membership of the sudo group, since it allows you to execute commands as root. We discovered this when we ran visudo and grepped for the responsible config line.

Fourth observation: you are always working on behalf of one of your groups. id indeed shows you the groups you are member of, but there is only one gid listed.

(This is not a question but a shared observation.)

29. First type id again.

Now use the newgrp command (check its --help or man page), to start working on behalf of one of your other groups: the sudo group again.

newgrp sudo

Confirm your gid has changed to the value you see with grep sudo /etc/group (it's 27).

*Sometimes changes in group membership are NOT propagated to /etc/gshadow.. if you get a “newgrp: failed to crypt password with previous salt” error make sure your user is present there in de sudo group as well. If not, add it manually using nano /etc/gshadow or even simpler, run grpconv as root*

*Green text on a black background

Description automatically generated*

30. Try again to cat a file that is normally off limits to you, a regular user. Think of the /etc/shadow file we mentioned in theory classes: it contains password hashes and normal users cannot read it.

cat /etc/passwd

A screen shot of a computer

Description automatically generated

A black screen with white text

Description automatically generated

A screenshot of a computer

Description automatically generated

**Note**

As you observed, a normal user can read /etc/passwd but this is safe, as it does not contain any password hashes. Tons of programs expect to be able to read this file, and libc does -- even the whoami and ls commands needs to read it so they can translate the uid/gid numbers to the user and group names you invented!

In the old days, the hashes were stored in the same file /etc/passwd.

Later, the shadow suite was introduced, which stored sensitive and extra information as a 1-on-1 mapping between /etc/passwd and /etc/shadow, the latter containing the sensitive info and hashes, with read access only for root.

31. Try again this time using sudo again.

Explain in your own words why, even though your gid is now that of group sudo, you cannot read /etc/shadow, unless you use the sudo command:

Because sudo is executing commands as a root user, and not as a current user.

32. Before proceeding, type Ctrl-D or type exit. You will not be logged out, but still have a shell open. Why?

Because another shell is still open?

33. Get a root login shell (with all root environment’s settings!) again. This time, use sudo. Which command did you use ? (QUIZ)

sudo -i

A close up of a black screen

Description automatically generated

34. Since we were talking about /etc/shadow, we will mention a tool to manipulate its entries.

**Note**

Not passwd. We already used passwd at the start of the lab, but true, it is another tool that modifies the contents of /etc/passwd's dark twin file /etc/shadow

We will study chage.

Open its man page. Search for the option to show account aging information.

chage -l username

35. Now invoke chage with that option, to see the account aging information for your regular user. Study the information, and realize that it is

- maintained in /etc/shadow (consider it a primitive database)

- checked at login by PAM (see man pam, but this is S5 lab material)

- manipulated using chage

No need to modify anything, just remember this tool and know the man page is always just a couple of keystrokes away.

A screen shot of a computer screen

Description automatically generated

## Section 3 : Chmodding the night away

36. Read the man page for chmod to find out how to recursively update the permissions for a folder and everything beneath it.

Now apply this command on /home to set the permissions to this pattern: rw-r---

**Note**

You will need to calculate the octal value. You remember this from last year ?:

rw- : user permissions : rwx = 421 rw- = 4+2=6

r-- : group permissions: rwx = 421 r-- = 4

--- : other's perms : rwx = 421 --- = 0

Hence the octal value expected by tools such as chmod is 644 for this permissions pattern.

**Important**

This question and the next one are perfectly fine!

In this question 36, you indeed take away the possibility to read any files in your own home directory! Try cd for example as your regular user.

However, this is restored in question 37, which restores the executable flag on all folders again using the find command.

A screen shot of a computer

Description automatically generatedchmod -R 640 /home

37. Read the manual page for find. Search for the option to execute a command for every hit. Also search for the option to match just directories (think of file type and search for this word in the man page). Now go and find all folders under /home, and let find execute this command on them:

chmod 750 {} \;

The {} sequence is replaced by find with the currently encountered file. All folders will get rwxr-x--- permissions.

After you’ve done this, use chmod 755 on the /home directory as well or you’ll still be locked out! Can you imagine why ?

sudo find /home -execdir chmod 750 {} \;

A screen shot of a computer program

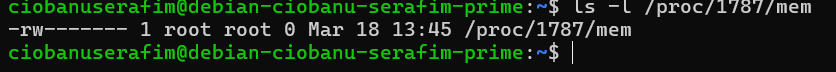
Description automatically generated

38. Read the man page for find to discover what this command does:

find / -perm 0400 2>/dev/null

Also just try to execute it to see if you discover it.

Try ls -l or stat on the files it finds.



It shows the folders that can only be read by the owner I suppose, as you are specifying the permissions

39. Type these commands **as root**: (so not as your regular user)

cp /bin/nano /tmp

chmod u+s /tmp/nano

ls -l /tmp/nano

You just created a copy of nano in /tmp, and root is the owner, and the SETUID bit has been set (u+s).

A black screen with white text

Description automatically generated

40. Work as your regular user again now.

Type nano /etc/shadow. This won’t work (permission denied)

Now try /tmp/nano /etc/shadow. What happens now ?

Now as you look on the strongly locked down contents of the /etc/shadow file, you should realize that the SUID bit grants your user root powers, so this bit must not be bestowed lightly upon binaries.

**Note**

This is a very simple root backdoor, and it can be easily circumvented by setting the nosuid mount option in /etc/fstab so executable files can never execute under another uid than the current user.

A screenshot of a computer

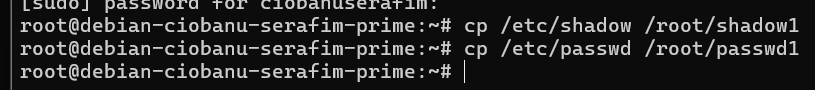
Description automatically generated

## Section 4 : Locking and Unlocking

41. Become root again using su --login.

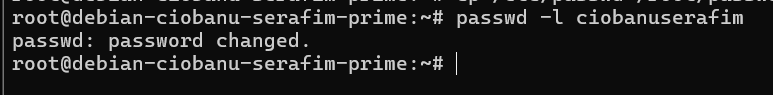
Back to the passwd command. It can lock a user's password or unlock it again. Study how this works as follows.

First use cp to create a backup of /etc/shadow and /etc/passwd to /root/shadow1 and /root/passwd1.



42. Now lock the password of your regular user, still running as root:

passwd -l ciobanuserafim



43. Logout and try to login as your user. What happens when you try to login?

You just locked it

44. Login as root again (you’ll need the console). Take a new copy of /etc/shadow and /etc/passwd so you can study what passwd -l has done exactly.

Now compare them using these commands that will show you the differences:

diff /root/passwd1 /root/passwd2

diff /root/shadow1 /root/shadow2

Which 1 char is different (added) to the version with the locked user? (QUIZ)

Explain the modification that passwd -l user has made to either or both of these files, in your own words:

A screen shot of a computer

Description automatically generated

What is did, it just either cancelled all the possible hash, or it is a somewhat kind of method to indicate invalid characters.

Anyway, it is just clear that it is different and that the user is locked

45. Unlock your regular user again using passwd -u. You see the explanation for the possible options in man passwd or passwd --help.

passwd -u ciobanuserafim

46. Type man find. create a find command, referencing to the man page, to search (as root, using sudo) all files owned by your regular user, on your entire filesystem (root /).

Pipe to less but send the error output to /dev/null.

This gives you an idea where files owned by your user are on your system (you will be surprised where you can find files owned by your user, outside of your homedir).

sudo find / -user ciobanuserafim 2>/dev/null | less

47. Similarly search in /dev for all entries with owning group tty. Redirect all errors to /dev/null. (QUIZ)

sudo find / -group tty 2>/dev/null

sudo find /dev -group tty 2>/dev/null | less (correct answer QUIZ)

## Section 5 : Don’t trust the man behind the umask.

48. Log back in as your regular user account.

Set your umask to 777 using the umask command in your current shell.

umask 777

**Note**

umask is a filter on the file permissions given to new files. You can set the r, w or x bit for user, group and others and those permissions will never be set on newly created files.

49. Now touch testfile.

Now show its permissions using ls -l. What permission bits are set and to which value? The answer should be: none.

You filtered out all possible permission bits from being set on this file, by setting umask first.

A screen shot of a computer

Description automatically generated

Because umask is like getting rid of permissions, and not adding default ones. The default umask I believe is 666

50. Now try the experiment again, but now set umask 0.

Rm testfile and create the testfile again. Again ls -l and compare the permissions for yourself. These are the permissions bash would have when creating testfile, unharmed by a umask filter because it is set to 0.

A screen shot of a computer

Description automatically generated

51. Use chmod to get the same effect as you had in point 48: set zero permissions on testfile.

chmod 000 testfile

A screen shot of a computer program

Description automatically generated

52. Try to cat testfile. Even though the file is owned by user and you are user, you denied yourself any access to the file.

**Note**

Of course since you are the owner, you can again change the permissions using chmod, granting yourself access again. Do not do this.

A screen shot of a computer

Description automatically generated

53. Now try to cat testfile prefixing the command with sudo to execute it as root.

Note

As you see, root bypasses access control. You can cat the file even though it has empty permissions (chmod 0 testfile). Some programs will warn you if you try to do something as root for which you normally have no permission; some will just go on without telling you.

A screen shot of a computer program

Description automatically generated

54. Work as your regular user and edit .bashrc to always set umask 027. This will take effect the next time a bash shell is opened. Explain which limits this imposes on newly created files by your user:

You will get rid of all the permissions on the others part, and get rid of write for the group part

A screen shot of a computer

Description automatically generated

55. Now how would you apply this setting on your existing structure in your home dir using one or more chmod -R commands?

The desired changes are: the permissions to be applied at most when taking question 54 umask setting in account.

sudo chmod -R 750 ~

A screen shot of a computer

Description automatically generated

Note

That way we deviate from the UNIX default of allowing others access to your home directory (but by default not to newly created files).

A university culture is one of openness and trust, but corporate and society culture require us to be closed and not trust each other. That is what we learn in security.

## Section 6 : Adding , Welcoming and removing users

56. Create a message of the day (motd) for all users, greeting them and welcoming them on your system. Read man motd to know how to accomplish this trivial task.

sudo nano /etc/motd

A screen shot of a computer

Description automatically generated

I added welcome to the dungeon

57. Use adduser to create a user testuser. Now use passwd -l to immediately lock it. Try to log in as testuser in vain.

A screenshot of a computer

Description automatically generated

adduser testuser

sudo passwd -l testuser

A screen shot of a computer program

Description automatically generated

58. Unlock your testuser. Now login. You should see your message of the day finally.

A screenshot of a computer program

Description automatically generated

A screen shot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

59. Type:

ls -a /etc/skel

These files are copied to a new user's home folder. So if you want to pre-populate a user’s home folder or set up some default settings to be set up as soon as the user is created, this is the location to remember.

A screenshot of a computer

Description automatically generated

60. Logoff your testuser and remove testuser with the userdel command.

A screen shot of a computer program

Description automatically generated

(Forgot that I had a shell open, oopssie)

This is what userdel will do :

rm -rf the testuser homedir

remove entry from /etc/shadow

remove entry from /etc/passwd

remove entry from /etc/group

remove entry from /etc/gshadow

So you could remove a user manually by doing these things yourself. without using userdel. Not that you should. But you could.

61. Finally, type man usermod

At the bottom, in SEE ALSO, you will find an overview of all the relevant files and commands for user management. It is e.g. possible to add a user to a group using usermod, like we did before using adduser. Find out how to add someuser to somegroup using usermod (QUIZ).

sudo usermod -aG group user

sudo usermod -aG somegroup someuser