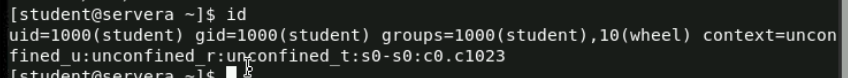
Name-Surname:Onur Çetin

ID:20200808050

1- It provides details about a user or group, including unique identifiers such as User ID (UID) and Group ID (GID), supplementary group IDs, and the associated username or group name.



2-For user information : /etc/passwd file.(username,UID,GID,home directory…)

metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

Açıklama otomatik olarak oluşturuldu

-For password information: /etc/shadow file. I put a screenshot on the third question.

3-Password field stores three pieces of information: the hashing algorithm in use, the *salt*, and the cryptographical hash.

Username:$hashing\_algorithm$salt$hash with this sequence.

metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

Açıklama otomatik olarak oluşturuldu

4- Both **PrimaryGroup** and **SupplementaryGroup** classes inherit from the **Group** class. This inheritance relationship allows them to inherit common attributes from the base class.

One-to-one relationship between a user and their primary group.

One-to-many relationship between a user and supplementary groups.

metin, ekran görüntüsü, sayı, numara, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

5-

Su and su – similarities:If you don't specify a username, the **su** or **su -** command will attempt to switch to the root user by default.

Su and su – difference: su - configures the shell environment as if it is a fresh login as the specified user, whereas su starts a shell as that user but retains the original user's environment settings.

Su and sudo – similarities: Environments are same (current user)

Su and sudo – difference:When using **su**, you essentially initiate a new shell with the environment of the target user. On the other hand, **sudo** allows you to execute a single command with elevated privileges without altering the entire shell environment.

Su and sudo –i similarities:Both **su** and **sudo -i** are used to gain elevated privileges, typically to become the superuser (root) or another user with increased permissions.

Su and sudo – i difference: Environment Variables,home directory and password requation are different.

Su- and sudo similarities:Both **su -** and **sudo** are used to gain elevated privileges and initiate a new shell with an environment similar to the target user's environment.

Su- and sudo difference:**su -** is used to switch to another user or the root user, requiring the target user's password, while **sudo** is used to run a command with elevated privileges, typically requiring the user's own password.

Su- and sudo-i similarities:Both **su -** and **sudo -i** are used to initiate a new shell with the environment of the target user.

Su- and sudo-i difference:**su -** is used to switch to another user or the root user with the environment of the target user, while **sudo -i** is used to run a command with elevated privileges and initialize the environment as if the target user had logged in directly.

Sudo and sudo-i similarities: Authorization: Both commands require the user to be authorized in the sudoers file to execute commands with elevated privileges.

Sudo and sudo-i difference:**sudo** allows executing a single command with elevated privileges without altering the entire shell environment, while **sudo -i** initializes the environment as if the root user had logged in directly.

6- sudo useradd miles

sudo passwd miles

sudo groupadd netflix

sudo usermod -aG netflix miles

sudo usermod -aG netflix co (my username Cetin Onur)

metin, ekran görüntüsü, yazı tipi, sayı, numara içeren bir resim

Açıklama otomatik olarak oluşturuldu

7- sudo userdel -r uncleaaron

8- sudo passwd -l miguel

9- Permissions in a file system dictate the actions that users and groups can perform on a file or directory. There are three types of permissions: read(r), write(w), and execute(x).

These permissions apply to three categories of users: owner,group, and others.

For example, -rw-r--r--. This means that the owner can read and write the file, but not execute it.The group and others can also only read the file.

Although the user is not the owner, they share the group ownership. If the group permission allows reading, this user will be able to view the file, even if the owner cannot, demonstrating how permissions are applied based on user and group settings.(Like ---rw-r--)

10- Miguel owns this file but don’t have a read permission for this(-wx). So that just villains group can read this.-->Spot

11- Owner of the file and spiderman group can write this file.-->gwen,miles

12- chmod g+x evilPlans🡪The command adds permission for the group to execute the file.(result --wxrwx-w-)

13- We need read and execute (r and x) permission to use ls -l command.

drwxr-xr-x As it can be seen, everyone has these permissions 🡪miles,spiderman,spiderverse

14- If we want to make a deletion, we need the write permission(w).

-Everyone has w permission. -wx rw- -w- 🡪 miguel villains evilPlans

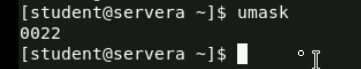
15- umask

* The first digit represents the owner's permissions.
* The second digit represents the group's permissions.
* The third digit represents others' (non-owner, non-group) permissions.

- The default permissions for a new file or directory would then be rw-r--r-- (read and write for the owner, read-only for the group and others).

-The meaning the following exapmle, it means the default permissions are set to deny write permission for the group and others (2 corresponds to write permission).

-The umask value of 0022 indicates that write permissions (2) for the group and others are masked. As a result, the default permissions for a new file would be 644 (666 - 022), and for a new directory, it would be 755 (777 - 022).



16- As you see , default permissions are totally different:

drwxr-xr-x (umask 0022)

-rw-r----- (umask 0027)

