# Managing Users and Permissions in Linux

In this project, I practiced fundamental Linux user and permission management tasks on a virtual machine. The goal was to gain hands-on experience with core administrative skills used in system security and access control.

I gained experience in:

- Create and confirm new user accounts, with and without home directories.
- Set and manage user passwords and properties (including default shells).
- Assign users to groups to control privileges.
- Lock and unlock user accounts for security enforcement.
- Safely delete users and their associated data.

## Adding a new user

### Command: 'Sudo adduser joker'


- sudo è Runs the command with administrator (superuser) privileges, as this command requires higher permission
- adduser è The program used to create a new user account on the system
- joker è Username of the account being created

**Output:**

```
Adding user `joker' ...
Adding new group `joker' (1000) ...
Adding new user `joker' (1000) with group `joker' ...
Creating home directory `/home/joker' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for joker
Enter the new value, or press ENTER for the default
        Full Name []: Sami Assi
        Room Number []: 3
        Work Phone []: 54325624
        Home Phone []: 54364236
        Other []:
Is the information correct? [Y/n] y
labex:project/ $
```

- The system adds user ‘joker’ and requests a password to then reconfirm the password for the user
- After that, its prompted me for optional information such as **Full Name**, **Room Number**, **Work Phone**, **Home Phone** & **Other**

## Confirming User Creation

### Command: ‘Sudo greg -w ‘joker’ /etc/passwd

```
labex:project/ $ sudo grep -w 'joker' /etc/passwd
joker:x:1000:1000:Sami Assi,3,54325624,54364236:/home/joker:/bin/bash
```

- This command searches the system to confirm user ‘joker’ was created
- grep è A command-line tool used to search text inside files. Looks for patterns or strings
- -w è This tells grep to match the whole word exactly, so it wont match partial words such as ‘joker123’ — it will only find ‘joker’
- joker è The search term which is the name of the user, we’re asking grep to search ‘joker’
- /etc/passwd è This is the file that stores all local user account information.

### Output:

```
joker:x:1000:1000:Sami Assi,3,54325624,54364236:/home/joker:/bin/bash
```

- Username: joker
- Password: x (actual password is stored securely elsewhere)
- User ID: 1000
- Group ID: 500
- Full name: Sami Assi
- Home Directory: /home/joker (hasn’t been made yet)
- Default shell: /bin/bash

## Creating a User with a Home directory

### Command: ‘sudo useradd -m bob’

```
labex:project/ $ sudo useradd -m bob
```

- Creates another user named bob

- -m è The -m option tells the system to create a home directory for the user (/home/bob). A directory is being like a personal folder where the user can store their files and settings.

## Confirming a User with a Home Directory

**Command:**

```
labex:project/ $ sudo ls -ld /home/bob
```

- This command verifies the home directory was created
- Sudo è Runs the command with super (administrator) privileges.
- ls è The command used to list directory contents
- -l è This option provides a detailed (long) listing format, showing permissions, owner, group, size and modification date
- -d è This option tells ls to show information about the directory itself rather than the contents
- The path to the directory you want to examine /home/bob

**Output:**

```
drwxr-x--- 2 bob bob 57 Sep 23 05:58 /home/bob
```

- File type & permissions è drwxr-x---
- First character = d for directory

**Next 3 characters = owners permissions (rwx)**
- r = read (can list directory contents), w = write (can create/delete files in the dir), x = execute (can enter cd into this dir)

**Next 3 characters = group permissions (r-x)**
- r = group members can list files, - = no write permission for group's, x = can enter the directory

**Last 3 characters = other's permissions (- - -)**
- No read, write, no execute è outsiders cannot access

Therefore: Owner (bob) è Full rights (read, write, execute)
Group (bob) è Read + enter, but not modify
Others è no access
bob bob è First bob shows user who owns it (the account bob)
bob bob è Second bob shows the group that owns it (the group named bob)
57 è is the size of the directory /home/bob in bytes

## Setting a User Password for a user

**Command:**

```
labex:project/ $ sudo passwd joker
```

- This command sets a password for the new user ‘joker’

**Output:**

```
New password:
Retype new password:
passwd: password updated successfully
```

- The output is asking for a password for the user, if successful. It will say “password updated successfully”.

## Modifying User Properties

**Changing the home directory of the user ‘joker’**

**Command:**

```
labex:project/ $ sudo usermod -d /home/wayne joker
```

- Usermod è command used to modify user account settings
- -d /home/wayne è specifies the new directory name /home/wayne that we want it to be changed to
- Joker è is the user we’re modifying
- No output will be visible

**Verifying the change in home directory for user ‘joker’**

**Commands:**

```
labex:project/ $ sudo grep -w 'joker' /etc/passwd
```

- This command verifies the change in user home directory

- Same command as used previously, searching for the user ‘joker’ in the /etc/passwd directory which contains user information
- Output should be information of the user and his new home directory name

**Output:**

```
joker:x:5001:5001::/home/wayne:/bin/sh
```

- This output confirms the user joker has a new home directory called /home/wayne.
- Therefore, the change in directory was successful

## Changing User Shell

**By default the sh (Bourne Shell) is the basic shell that’s used on most Unix-like systems, bash (Bourne Again Shell) offers more features and is generally more user friendly. So we’ll be changing to this shell.**

**Command:**

```
labex:project/ $ sudo usermod -s /bin/bash joker
```

- **This command changes the user ‘joker’ from default shell sh to bash**

**Confirming the change in shell**

**Command:**

```
labex:project/ $ sudo grep -w 'joker' /etc/passwd
```

- The output should show the joker user information from the /etc/passwd directory which contains user information. It should now be replaced with /bin/bash

**Output:**

```
joker:x:5001:5001::/home/wayne:/bin/bash
```

This confirms the change of shell to bash from sh by the /bin/bash

## Adding a User to a Group

**We use groups to organise users and manage permissions. An important group is the sudo group. Users in the sudo group have administrative privileges. Sudo users are able to install and update software packages, adjust system configuration, create/modify & delete other user accounts. So in this section we will add a user 'joker' to the sudo group.**

**Command:**

```
labex:project/ $ sudo usermod -aG sudo joker
```

- Usermod è modify user accounts
- -aG è means 'Append to group' (Add to a group without removing from other groups)
- Sudo è the group we are adding the user to
- Joker è the user we're modifying/ adding to sudo group

**Confirming user has been added to the sudo group**

We can confirm this by checking if the sudo group is listed among the user 'joker'

**Command & Output:**

```
labex:project/ $ groups joker
joker : joker sudo
```

- Checking what groups user 'joker' belongs to
- Joker : before the colon is the username we are checking
- : Joker after the colon is the primary group (by default every new user gets their own group with the same name)
- Sudo is the additional group the user belongs to giving them sudo privileges

**We can further confirm this by changing from the current user to the joker user and trying a command that requires sudo privileges**

**Command & Output:**

```
labex:project/ $ su - joker
Password:
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
```

- The command 'su – joker' switches current users to the joker user

- You're prompted to enter joker's password which we assigned before to gain access to his user

**Attempting to access a file which only sudo admins are able to access**

**Command & Output:**

```
joker@68d33ea60f92ec70f112dd28:~$ sudo cat /etc/shadow
[sudo] password for joker:
root:*:19901:0:99999:7:::
daemon:*:19901:0:99999:7:::
bin:*:19901:0:99999:7:::
sys:*:19901:0:99999:7:::
sync:*:19901:0:99999:7:::
```

- This command 'sudo cat /etc/shadow' is trying to read a file which requires root/sudo privileges, specifically the password file. Which contains hashed passwords
- The user was able to read this file therefore the user has sudo privileges

## Locking & Unlocking User Accounts

**Command:**

This Command locks the users' (joker) account

```
joker@68d33ea60f92ec70f112dd28:~$ exit
logout
labex:project/ $ sudo passwd -l joker
passwd: password expiry information changed.
labex:project/ $ su - joker
```

- 'sudo passwd -l joker' locks the password for the joker account, making it locked and unable to use
- The -l option locks the password of the account
- 'Su – joker' switches to the user account to verify

There is an authentication failure as the joker user is locked:

```
labex:project/ $ su - joker
Password:
su: Authentication failure
```

## Unblocking the user account

### Command:

This command unblocks the users' password for his account

```
labex:project/ $ sudo passwd -u joker
passwd: password expiry information changed.
```

- -u command unblocks the password for the joker account
- The account is unblocked, but we should check by trying to swap accounts

### Swapping accounts to verify the user account is unblocked

```
labex:project/ $ su - joker
Password:
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
```

- Su – joker command used to enter the joker account from the current account
- The account successfully allows commands to be run, therefore is unblocked

## Deleting a User

### This command deletes user accounts

```
labex:project/ $ sudo userdel -r bob
```

- Userdel command deletes user accounts
- -r option removes the user's home directory and mail spool

### Verify that the user has been deleted

```
labex:project/ $ sudo grep -w 'bob' /etc/passwd
labex:project/ $ sudo ls -ld /home/bob
```

- Both of these commands should return no results, to verify that the user has been deleted