Day 3 – Phase 3: User, Group, and Permissions Management

Boss’s Request: Secure the project and restrict access to authorized users only.

Tasks:

• Create a new group iot\_team and add your user to it.

A screenshot of a computer program

AI-generated content may be incorrect.

• Create a new developer user, add it to the group.



• Change ownership of iot\_logger to the developer + group.

A computer screen shot of a computer code

AI-generated content may be incorrect.

• Set permissions: group can read/write logs, others blocked.

A computer screen with blue and green text

AI-generated content may be incorrect.

• Test access as new user, then remove test user.

A purple background with white text

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

Open-Ended Questions:

• How do Linux file permissions (r, w, x) work for files vs directories? Give an example using ls -l.

**For Files :**

**r (read)** : you can view the contents (e.g., cat file.txt).

**w (write)** : you can modify the file (edit, truncate, remove content).

**x (execute)** : you can run the file as a program/script.

**For Directories :**

**r (read) :** you can list the files inside (ls).

**w (write) :** you can create, delete, or rename files inside (but not necessarily edit them).

**x (execute) :** you can enter the directory (cd) and access files *if you know their names*.

**A blue and black background

AI-generated content may be incorrect.**

• Explain octal notation for permissions and what the umask command does. Give one calculation example.

U stands for users

G stands for group

O stands for other

A number and numbers on a white background

AI-generated content may be incorrect.

**The umask** command in Linux is used to set default permissions for files or directories the user creates.

* **File ->**The full permission set for a file is 666 (read/write permission for all)
* **Directory ->** The full permission set for a directory is 777 (read/write/execute)
* When we make a new directory, the permissions will be calculated as (full permissions for directory) - (umask value) i.e. 777 - 543 = 234
* When we make a new file, the permission will be given out similarly but with a slight change as follows: (full permissions for file) - (umask value) i.e. 666-543 = 123

• What is the difference between the root user and a normal user? Why is root considered dangerous?

**Root User**

* The **root** account is the **superuser** in Linux.
* It has **unrestricted access** to the entire system:
  + Can read, write, and execute any file (regardless of permissions).
  + Can install/remove software.
  + Can add/remove users or groups.
  + Can change ownership and permissions of any file.
  + Can shut down, reboot, or modify the kernel.

**Normal User**

* A normal (non-root) user has **limited privileges**:
  + Can only access files they own (or have permission for).
  + Cannot change system-critical files in directories like /etc, /bin, /usr.
  + Cannot install software system-wide (unless using sudo).
  + Can only manage processes they started.

The root user is dangerous because it can change and delete important files that can break the system