# File permissions in Linux

## Project description.

In this project, I used Linux commands to examine and manage file permissions, ensuring that only authorized users have access to specific files and directories while restricting unauthorized access.

## Check file and directory details.

the command you can use to check permissions in the "Check file and directory details" section of the template using ls -la command.

**Example:**

* Command to Check Permissions: To check file and directory permissions, I used the 'ls -la' command. This command displays a detailed list of files and directories, including their permissions, owners, groups, and more.

## Describe the permissions string.

The 10-character string focusing on u, g, and o permissions using r, w, and x.

**Example:**

* Permissions String Description: The 10-character string represents file permissions, with 'u' indicating user permissions, 'g' for group permissions, and 'o' for others (everyone else) permissions. Within each group, 'r' stands for read permission, 'w' for write permission, and 'x' for execute permission. If a permission is granted, the corresponding letter is present; otherwise, it's replaced with a hyphen '-.'

## Change file permissions.

Based on the permissions established in Step 3, identify which file needs to have its permissions modified. Use a Linux command to modify these permissions using chmod

**Example:**

* Command to Change File Permissions: To modify file permissions, I used the 'chmod' command. For example, to remove write access from others ('-w') on a file named 'example.txt,' I used the following command:
* chmod o-w example.txt
* The output showed the successful modification of permissions.

## Change file permissions on a hidden file.

Use the chmod command to assign the appropriate permissions to the hidden file '.project\_x.txt' with 'u', 'g', and 'o' permissions as well as 'r' and 'w'.

**Example:**

* Command to Change Hidden File Permissions: To assign the appropriate permissions to the hidden file '.project\_x.txt,' I used the following command:
* chmod u=rw, g=rw, o= .project\_x.txt
* This command grants read and write permissions to the user and group while removing all permissions from others. The output confirmed the successful permission change.

## Change directory permissions.

Use the chmod command to modify directory permissions for the 'drafts' directory within the 'projects' directory using 'u', 'g', and 'o' permissions.

Command to Change Directory Permissions: To modify directory permissions for the 'drafts' directory within the 'projects' directory, I used the 'chmod' command.

The following command was used to restrict access to only the 'researcher2' user:

* chmod u=rwx, g=, o= drafts

In this command, 'u=rwx' grants the user ('researcher2') read, write, and execute permissions, while 'g=' and 'o=' remove all permissions from the group and others. The output confirmed the successful permission change.

## Summary.

In this project, I effectively examined and managed file permissions in a Linux environment to ensure the security of the organization's research team. By using the 'chmod' command with 'u', 'g', and 'o' permissions and 'r', 'w', and 'x' attributes, I verified and adjusted file and directory permissions to align with the organization's security policies. This proactive approach helps maintain data integrity and confidentiality while preventing unauthorized access to critical files and directories.