# File permissions in Linux

## Project description

The research team at my organisation needs updates to the file and directory permissions within the projects directory. There are permissions granted to individuals who are not authorised to have those permissiosn. Checking and updating these permissions will ensure the correct access is provided and keep their system secure. To improve the security of the system, I carried out the following tasks:

## Check file and directory details

I used Linux to investigate the existing permissions for the projects directory in the system.

A screenshot of a computer

AI-generated content may be incorrect.

I used the ls command to list the contents of the directory and the -la option to view permissions and also include hidden files. The output of the command shows that there is one drafts directory; one hidden file—.project\_x.txt; and four other project files. The permissions of the files are represented by a 10-letter string in the first column.

## Describe the permissions string

The 10-letter string is an alphabetic representation for the permissions granted for the file. There are three types of permissions provided: read (r); write (w); and execute (w). These permissions are granted, or not granted, to three types of owners: user (u); group (g); and other (o). The characters and what they represent are as follows:

* **1st character:** This indicates the file type. This is a d for directory, or a hyphen (-) for a regular file.
* **2nd – 4th characters:** These indicate the permissions granted for a user—respectively, read (r); write (w); and execute (w). If any one of these characters is a hyphen (-), this indicates that permission is not granted for the user.
* **5th – 7th characters:** These indicate permissions granted for the group—respectively, read (r); write (w); and execute (w). If any one of these characters is a hyphen (-), this indicates that permission is not granted for the group.
* **8th – 10th characters:** These indicate permissions granted for other—respectively, read (r); write (w); and execute (w). If any one of these characters is a hyphen (-), this indicates that permission is not granted for other.

## Change file permissions

Based on organisation rules, other should not have write access to any files. However, as visible in the image above, project\_k.txt currently grants write permissions for other.

I used the following Linux command to remove the write permissions for other.

A screen shot of a computer

AI-generated content may be incorrect.

The first two lines of the screenshot display the commands I input, and the remaining lines display the output of the second command.

The chmod command changes the permissions on files and directories.

* The first argument (o-w) indicates what permissions should be changed; and
* The second argument (project\_k.txt) specifies the file or directory where the change should be applied.

In this example, I removed write permissions from other for the project\_k.txt file. After this, I used ls -la to review the updates I made.

## Change file permissions on a hidden file

The research team at my organisation recently archived project\_x.txt. They do not want anyone to have write access to this project, but the user and group should have read access.

The following code demonstrates how I used Linux commands to change the permissions:

A screenshot of a computer program

AI-generated content may be incorrect.

The first two lines of the screenshot display the commands I entered, and the other lines display the output of the second command. The full stop (.) before the filename of .project\_x.txt indicates it is a hidden file. In this example, I overwrote permissions to the user and group, with u=r, and g=r, respectively. These only give read permissions for the user and the group.

## Change directory permissions

The organisaition only wanted researcher2 (the user) to have access to the drafts directory. This means that no one other than researcher2 should have execute permissions. However, the group had execute permissions to the directory that they should not have—this needs to be removed.

The following code demonstrates how I used Linux commands to change the permissions:

A screenshot of a computer program

AI-generated content may be incorrect.

The first two lines of the screenshot display the commands I entered, and the other lines display the output of the second command. I used the chmod command and g-x to remove (-) execute (x) permissions from the group (g). The researcher2 user already had execute permissions, so they did not need to be added.

## Summary

I changed multiple permissions to match the level of authorization my organization wanted for files and directories in the projects directory. The first step to this process was using ls -la to check the permissions for the directory. This informed the next steps, where I used the chmod command multiple times to change the permissions on files and directories.