# File permissions in Linux

## Project description

The research team at my organization needs to revise the file permissions for specific files and directories within the projects directory. Currently, the permissions do not match the required level of authorization. Reviewing and updating these permissions will enhance system security. To complete this task, I took the following steps:

## Check file and directory details

Here’s an example of using Linux commands to check the current permissions for a specific directory in the file system.

A screen shot of a computer program

Description automatically generated

I first entered commands to find the projects directory. The code outputs the entire contents of the projects directory. By using the ls command with the -la option, I generated a detailed listing that included hidden files. The result shows a directory named "drafts," a hidden file called ".project\_x.txt," and five other project files. The 10-character string in the first column displays the permissions set for each file or directory.

## Describe the permissions string

The 10-character string breaks down to show who can access the file and their specific permissions:

* 1st character: A d means it's a directory, while a - means it's a regular file.
* 2nd-4th characters: Indicate the read (r), write (w), and execute (x) permissions for the user. A - means the permission is not granted.
* 5th-7th characters: Show the read (r), write (w), and execute (x) permissions for the group. A - means the permission is not granted.
* 8th-10th characters: Represent the read (r), write (w), and execute (x) permissions for others (all users apart from the user and the group). A - means the permission is not granted.

For instance, the permissions for project\_t.txt are -rw-rw-r--. The - indicates it's a file, not a directory. The r in the second, fifth, and eighth positions shows that user, group, and others have read permissions. The w in the third and sixth positions indicates that only the user and group have write permissions. No one has execute permissions for project\_t.txt.

## Change file permissions

The organization decided that no files should have write access for others. To comply, I checked the file permissions I previously obtained and identified that project\_k.txt needs to have write access removed for others.

The following code demonstrates how I used Linux commands to do this:

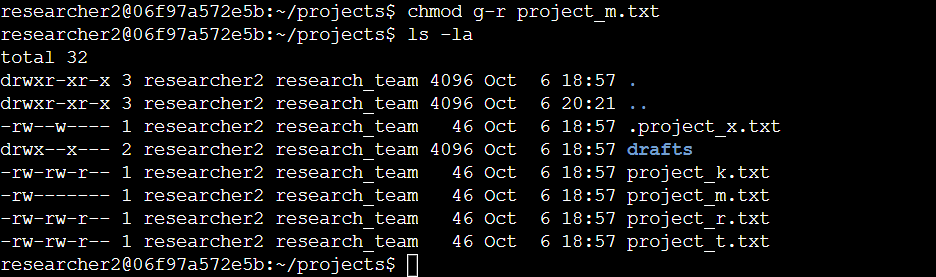
A screenshot of a computer program

Description automatically generated

The first two lines of the screenshot show the commands I typed, while the remaining lines show the output from the second command. The chmod command is used to change file and directory permissions. The first argument specifies the changes, and the second argument identifies the file or directory. In this case, I removed write permissions for others on the project\_k.txt file. Then, I ran ls -la to verify the changes.

I also removed the read permission for groups on file “project\_m.txt”.

The following code demonstrates how I used Linux commands to do this:



## Change file permissions on a hidden file

The research team at my organization has recently archived project\_x.txt. They want to ensure that no one has write access to this file, but both the user and group should retain read access.

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

A screen shot of a computer

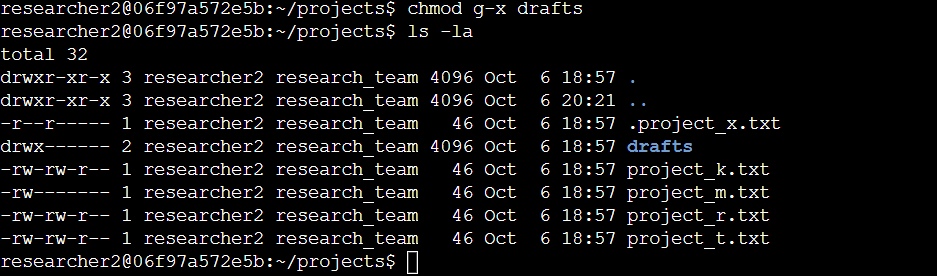
Description automatically generated

The first two lines of the screenshot show the commands I entered, while the rest display the output of the second command. The file “.project\_x.txt” is hidden because its name starts with a period. In this example, I added read permissions for the group and removed write permissions for the user and group. I added read permissions for the group with g+r, removed write permissions for the group with g-w, and removed the write permission with u-w.

## Change directory permissions

The organization requires exclusive access to the drafts directory and its contents for the researcher2 user. As a result, execute permissions should be granted solely to researcher2, while all other users are denied this access.

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



The first two lines of the screenshot show the commands I typed, while the following lines display the output from the second command. Since I had previously noted that the group had execute permissions, I used the chmod command to remove them. The researcher2 user already had execute permissions, so no changes were needed for them.

## Summary

I updated several permissions to align with the authorization levels my organization required for files and directories within the projects directory. First, I used ls -la to review the directory permissions, which guided my next steps. I then used the chmod command multiple times to adjust the permissions accordingly.