# Use Linux commands to manage file permissions

# Scenario:

This scenario is fictional

You are a security professional at a large organization. You mainly work with their research team. Part of your job is to ensure users on this team are authorized with the appropriate permissions. This helps keep the system secure.

Your task is to examine existing permissions on the file system. You’ll need to determine if the permissions match the authorization that should be given. If they do not match, you’ll need to modify the permissions to authorize the appropriate users and remove any unauthorized access.

# File permissions in Linux

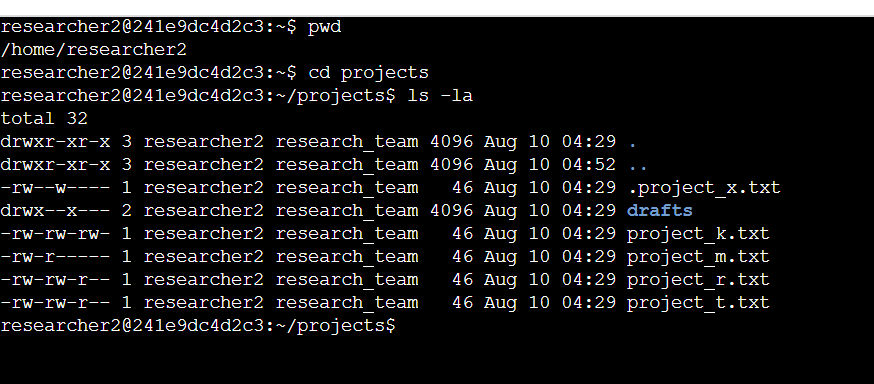
## Project description

The research team in my organisation need to check the permission sets of files and directory in the project directory and want to make sure that users on the team are authorised with appropriate permissions. Checking and ensuring the permissions will keep the system secure. In order to complete this task, the following tasks were done:

## Check file and directory details

For checking the file and directory details, the following commands were used :

## 



## Describe the permissions string

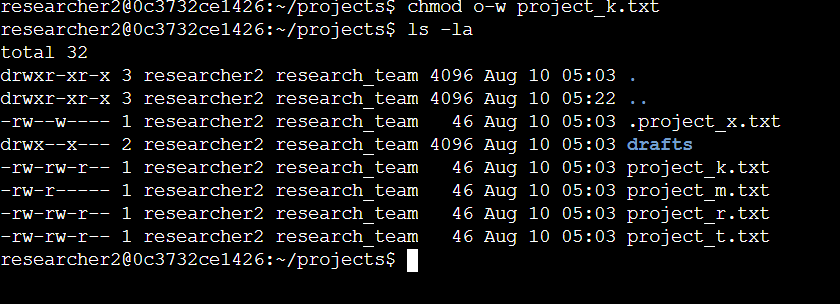
The 10 character string demonstrates who has access to files and certain permissions. drwxrwxrwx is the full 10 character string permission where the ‘d’ represents a directory file and if it is ‘--’ hyphen, it's a regular file. The first ‘rwx’ represents the ‘read’, ‘write’ and ‘execute’ permissions to the user; if any of these is shown ‘--’, that permission is denied to the user. The second set of ‘rwx’ represents permissions to the group; if any of these is shown ‘--’, that permission is denied to the group. The third set of ‘rwx’ represents permissions to other; if any of these is shown ‘--’, that permission is denied to other.

For example, file Permission to projects\_k.txt is -rw-rw-rw-.

Here, the User has read, write permissions, group has read, write permissions and other has read, write permissions.

## Change file permissions

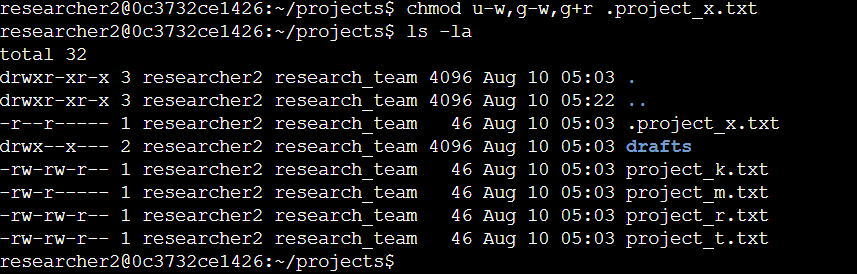
The organization does not allow others to have write access to any files. To which file needs to have its permissions modified, I used the following command:



The first two lines in the screenshot are the commands I used to change file permission of file project\_m.txt. In the first line, the command chmod was used to remove the ‘write’ permission from ‘others’ (o-w). After that, I used ls -la command to review the changes.

## Change file permissions on a hidden file

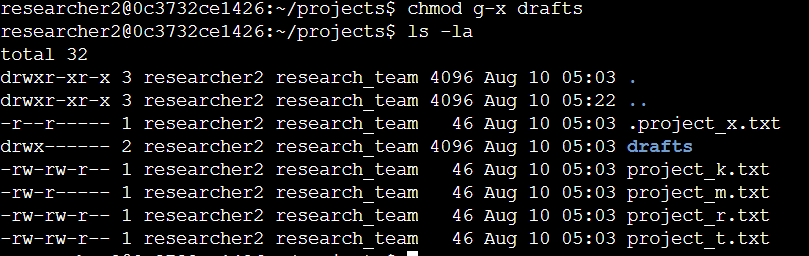
The research team has archived .project\_x.txt, which is why it’s a hidden file. This file should not have write permissions for anyone, but the user and group should be able to read the file. For this , i used the following commands:



For changing the file permissions for a hidden file, the chmod command was used to remove ‘write’ permissions from ‘user’ and ‘group’ (u-w, g-w) and to add read permission to group (g+r)

## Change directory permissions

The files and directories in the projects directory belong to the researcher2 user. Only researcher2 should be allowed to access the drafts directory and its contents. For this, I used the following commands:



The chmod command was used to remove the ‘execute’ permission (g-x) from the group since only the user researcher2 was allowed to access the drafts directory.

## Summary

The organisation wanted to change multiple permissions for the projects directory to match the level of authorisation. For executing the particular task, I used ls -la commands to identify the permissions to the files and directories including hidden files. Then I used the chmod command to change the access and permissions according to the organisations need.