# File permissions in Linux

## Project description

The research team at my organization has provided the permission guidelines as follows:

Others should not have write access to any files.

The hidden file, .project\_x.text, should not have write access by anyone.

Only users and groups are allowed to read .project\_x.text.

User, researcher2, should be the only one to access the subdirectory named drafts.

I have been tasked to examine and modify users’ permissions to ensure that our system data remains secure.

## Check file and directory details

To check file permissions, an analyst must enter the following text into the text editor:

ls -la checks the permissions of all files and directories, including hidden files.

Below are the permissions for the files located in the /home/researcher2/projects directory.

**project\_k.txt**

User = read, write,

Group = read, write

Other = read, write

**project\_m.txt**

User = read, write

Group = read

Other = none

**project\_r.txt**

User= read, write

Group = read, write

Other = read

**project\_t.txt**

User = read, write

Group = read, write

Other = read

**.project\_x.txt**

User = read, write

Group = write

Other = none

There is also one subdirectory inside the projects directory named **drafts**. The permissions

on drafts are:

User = read, write, execute

Group = execute

Other = none

## Describe the permissions string

In the file .project\_x.txt, the permissions string is -rw-w—--. This 10-character string represents what permissions each group has. The first character in the string represents the file type. A dash is present because it is not a directory, it is a file. Each permission group (user, group, and other) is split into 3-character segments. The first set is the user’s permissions. Permissions go in the order of read (r), write (w), and execute (x). A dash represents that no permissions are granted for that area. This permissions string outlines that the user only has access to read and write, the group can only write, and the other category has no permissions.

## Change file permissions

The organization explicitly stated that the others cannot have write permissions on any files. Due to this, the file project\_k.txt needs to be changed. To change these permissions, a security analyst would type the following command into the text editor:

chmod o-w project\_k.txt

The o represents the Others and the - means to remove the permission, in this case, the writing permissions.

## Change file permissions on a hidden file

Given that file .project\_x.txt is a hidden file, the user and group should only have access to read the file. To change these permissions, a security analyst would type the following command into the text editor:

chmod u=r .project\_x.txt

Chmod g=r .project\_x.txt

The = sign displays that the person only has a specific permission, in this case, only reading permissions for both the user and group.

## Change directory permissions

The drafts directory belongs to the user. The user is the only one who should have access this directory and its contents. To change this, a security analyst would type the following command into the text editor:

chmod g-x drafts

The g in this command represents the group. In the current directory permissions, the group has execution permissions. By using the “g-x” in this command, the analyst is removing the execution permission.

## Summary

Linux commands provide security analysts with information on who has permissions to which data to protect the privacy and security of files and directories. As a security analyst, I have to examine and modify permissions of data (files and directories) to ensure that each user has the appropriate authorization based on their titles. I have been provided guidelines on which users are authorized to read, write, and execute data in our systems. I have modified the group permissions of drafts, the user and group permissions .project\_x.txt, and the write permissions for others for project\_k.txt