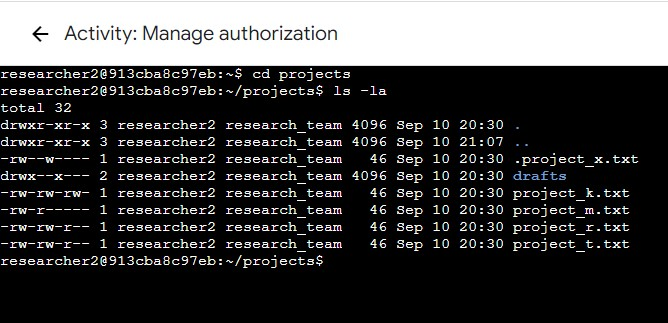
# File permissions in Linux

## Project description

The research team at my organization was tasked to examine the current existing permissions on the file system to ensure the system is secure. The research team and I will check and update certain files and directories within the *PROJECTS* directory. By doing this it will help the organization’s system to be more secure. I performed the following tasks:

## Check file and directory details

The following code demonstrates how I used the Linux commands to check the current permissions in the *PROJECTS* directory in the file system.



The first line of the screenshot displays the command I entered which is *cd* to change to a specific directory and the argument *projects* for me to change into the *PROJECTS* directory to conduct our tasks.

The second line of the screenshot I used the *ls* command with the *-la* option to reveal a 10-character string in the first column represents the current permissions set on each file or directory, detailed lists of file contents which is a, 5 project files,a directory named *drafts* and a hidden file content *.project\_x.txt.*

## Describe the permissions string

To determine who is authorized to access the file and their specific permissions, we can interpret the 10-character string which is *drwxrwxrwx* to have a better understanding in this directory and its specific permissions.

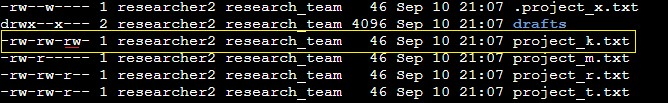
1. The 1st Character of ***d****rwxrwxrwx*- this character is either a d which means it is a directory, if it is a hyphen (-) means it is a regular file.
2. The 2nd,3rd and 4th Characters of *d****rwx****rwxrwx* - The *r* means read, *w* means write, and *x* means execute. The 2nd,3rd and 4th *rwx* in the 10-character string means it is for the type of owner which is the ***user***. If hyphen (-) is in the place instead of character it means that this permission is not granted to the type of owner which is the ***user****.*
3. The 5th,6th, and 7th Characters of *drwx****rwx****rwx -* The *r* means read, *w* means write, and *x* means execute. The 5th,6th, and 7th *rwx* in the 10-character string means it is for the type of owner which is the ***group***. If hyphen (-) is in the place instead of character it means that this permission is not granted to the type of owner which is the ***group****.*
4. The 8th, 9th, and 10th Characters of drwxrwx**rwx** - The *r* means read, *w* means write, and *x* means execute. The 8th,9th, and 10th *rwx* in the 10-character string means it is for the type of owner which is the ***other***. If hyphen (-) is in the place instead of character it means that this permission is not granted to the type of owner which is the ***other****.*

For example, 

The file permissions for project\_t.txt are -rw-rw-r--. Since the first character is a hyphen (-), this indicates that project\_t.txt is a file, not a directory. The second, fifth, and eighth characters are all r, which indicates that user, group, and other all have read permissions. The third and sixth characters are w, which indicates that only the user and group have write permissions. No one has execute permissions for project\_t.txt.

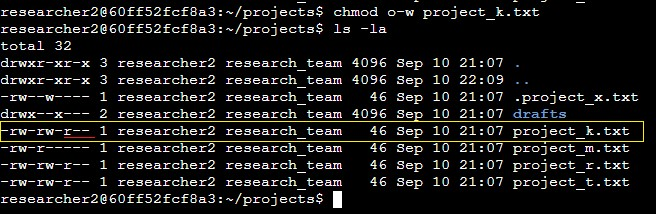
## Change file permissions

The organization wants to block the *write* permission of *other* (the type of owner of the 8th,9th and 10th character of the 10-character string) to access any of their files.



To comply with this, all of the *project.txt* files does not have write permissions aside from the *project\_k.txt* and must have the write access removed for the *other.*

The following code demonstrates how I use the Linux commands to block the write permission of *other* in the file *project\_k.txt* :



The first line of the screenshot displays the commands I entered, *chmod o-w project\_k.txt*. The *chmod* command changes the permission on the files and directories which in this case is a file. The first argument *o-w* indicates what permissions should be changed which in this case *o* means *other* the type of owner, (-) this symbol means taking away a permission, and *w* means write permission. *o-w* means we took away the write permission of the *other.* The second argument *project\_k.txt* means on what file or directory we want to change permissions. In this case is a file.

The second line of the screenshot displays the command I entered, *ls -la* to review the updates I made.

## 

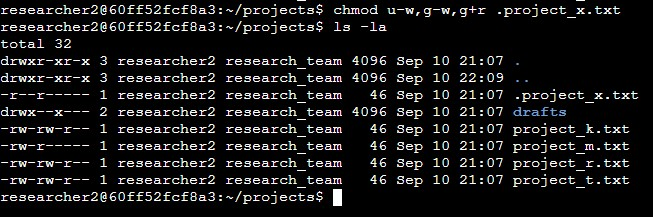
## Change file permissions on a hidden file

The research team at my organization recently archived the hidden file *.project\_x.txt*.



The organization wants to block write permission access to this hidden file, but the *user* and *group* should be able to read the file.

The following code demonstrates how I use the Linux commands to block the write permission of all the type of owner, grant read permissions to *user* and *group*, leave as it is for the *other*:



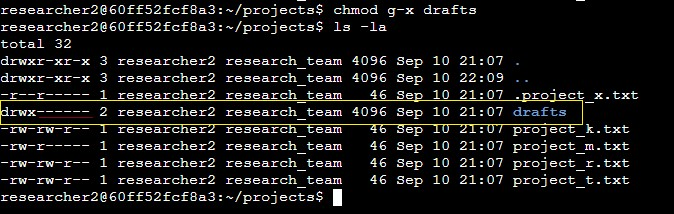
The first line of the screenshot displays the command I entered, *chmod u-w,g-w,g+r .project\_x.txt*. In this example, I removed write permissions from the user and group, and added read permissions to the group. I removed write permissions from the user with ***u-w***. Then, I removed write permissions from the group with ***g-w***, and added read permissions to the group with ***g+r***. Since it is an archived hidden file, we looked for a dot (.) after the file for us to know it is a hidden file. Which this case the *.project\_x.txt* file,because other files does not have dot (.).

The second line of the screenshot displays the command I entered, *ls -la* to review the updates I made.

## Change directory permissions

My organization only wants the *researcher2* user to have access to the *drafts* directory and its contents. This means the *user* only has access and block access to *group* and *other.*

The following code demonstrates how I use the Linux commands to block execute access of *group* and *other.*

**

The first line of the screenshot display the commands I entered, *chmod g-x drafts*. I used again *chmod* command to remove the execute permissions of *group*. The *researcher2* *user* already had all the permissions granted and leave it as it is.

The second line of the screenshot displays the command I entered, *ls -la* to review the updates I made.

## 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 in this was using *ls -la* to

check the permissions for the directory. This informed my decisions in the following steps. I

then used the *chmod* command multiple times to change the permissions on files and

directories.