

# File permissions in Linux

## Project description

In this project, I updated file permissions for certain files and directories within the projects directory. The permissions do not currently reflect the level of authorization that should be given. Checking and updating the permissions, this will help keep the organization's system secure. To increase the system's security I performed the following tasks:

```
researcher2@a735462327e2:~/projects$ ls -l
total 20
drwx--x--- 2 researcher2 research_team 4096 Oct 15 05:27 drafts
-rw-rw-rw- 1 researcher2 research_team  46 Oct 15 05:27 project_k.txt
-rw-r----- 1 researcher2 research_team  46 Oct 15 05:27 project_m.txt
-rw-rw-r-- 1 researcher2 research_team  46 Oct 15 05:27 project_r.txt
-rw-rw-r-- 1 researcher2 research_team  46 Oct 15 05:27 project_t.txt
researcher2@a735462327e2:~/projects$
```

## Check file and directory details

The following code demonstrates how I used Linux commands to determine the existing permissions set for a specific directory in the file system.

```
researcher2@a735462327e2:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Oct 15 05:27 .
drwxr-xr-x 3 researcher2 research_team 4096 Oct 15 05:54 ..
-rw--w---- 1 researcher2 research_team  46 Oct 15 05:27 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Oct 15 05:27 drafts
-rw-rw-rw- 1 researcher2 research_team  46 Oct 15 05:27 project_k.txt
-rw-r----- 1 researcher2 research_team  46 Oct 15 05:27 project_m.txt
-rw-rw-r-- 1 researcher2 research_team  46 Oct 15 05:27 project_r.txt
-rw-rw-r-- 1 researcher2 research_team  46 Oct 15 05:27 project_t.txt
researcher2@a735462327e2:~/projects$
```

The first line of the screenshot displays the command I entered, and the other lines display the output. The code lists all contents within the projects directory we are working with. I used the “ls” command with the “-la” option to display a detailed listing of the file contents which also

returns hidden files. The output of the command indicates that there is one other directory named drafts, one hidden file named “.project\_x.txt”, and five other project files. The 10 character string in the first column represents the permissions set on each file or directory.

## Describe the permissions string

The 10-character string can be deconstructed to determine who has authorized access to the file and their specific permission. The character and what they represent are:

- 1st character: is either a “d” or hyphen “-” and indicates the file type. If the first character is a “d”, it is a directory. Otherwise, it is a regular file.
- 2nd -4th character: These characters indicate the read(r), write(w), and execute(x) permissions for the user. When one of these characters is a hyphen(-), that indicates that this permission is not authorized for the user.
- 5th -7th character: These characters indicate the read(r), write(w), and execute(x) permissions for the group. When one of these characters is a hyphen(-), it indicates that this permission is not authorized for the group.
- 8th -10th characters: These characters indicate the read(r), write(w), and execute(x) permissions for other. When one of these characters is a hyphen(-), it indicates that this permission is not authorized for others..

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

I worked to change the permissions so that other shouldn't have write access to any of their files. To comply with this, I referred to the file permissions that I previously returned. I determined `project_k.txt` must have the write access removed for other.

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

```
researcher2@a735462327e2:~/projects$ chmod o-w project_k.txt
researcher2@a735462327e2:~/projects$ ls -l
total 20
drwx--x--- 2 researcher2 research_team 4096 Oct 15 05:27 drafts
-rw-rw-r-- 1 researcher2 research_team  46 Oct 15 05:27 project_k.txt
-rw-r----- 1 researcher2 research_team  46 Oct 15 05:27 project_m.txt
-rw-rw-r-- 1 researcher2 research_team  46 Oct 15 05:27 project_r.txt
-rw-rw-r-- 1 researcher2 research_team  46 Oct 15 05:27 project_t.txt
```

The first two lines of the screenshot display the commands I entered, and the other lines display the output of the second command. The `chmod` command changes the permissions on files and directories. The first argument indicates what permissions should be changed, and the second argument specifies the file or directory. In this example, I removed write permissions from other for the `project_k.txt` file. After this, I used `ls -l` to review the updates I made.

## Change file permissions on a hidden file

Reviewing `project_x.txt`, I adjusted file permissions on a hidden file so that no one 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:

```

researcher2@a735462327e2:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Oct 15 05:27 .
drwxr-xr-x 3 researcher2 research_team 4096 Oct 15 05:54 ..
-rw--w---- 1 researcher2 research_team  46 Oct 15 05:27 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Oct 15 05:27 drafts
-rw-rw-r-- 1 researcher2 research_team  46 Oct 15 05:27 project_k.txt
-rw----- 1 researcher2 research_team  46 Oct 15 05:27 project_m.txt
-rw-rw-r-- 1 researcher2 research_team  46 Oct 15 05:27 project_r.txt
-rw-rw-r-- 1 researcher2 research_team  46 Oct 15 05:27 project_t.txt
researcher2@a735462327e2:~/projects$ chmod u-w, g-w, g+r .project_x.txt
chmod: invalid mode: 'u-w,'
Try 'chmod --help' for more information.
researcher2@a735462327e2:~/projects$ chmod u-w,g-w,g+r .project_x.txt
researcher2@a735462327e2:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Oct 15 05:27 .
drwxr-xr-x 3 researcher2 research_team 4096 Oct 15 05:54 ..
-r--r----- 1 researcher2 research_team  46 Oct 15 05:27 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Oct 15 05:27 drafts
-rw-rw-r-- 1 researcher2 research_team  46 Oct 15 05:27 project_k.txt
-rw----- 1 researcher2 research_team  46 Oct 15 05:27 project_m.txt
-rw-rw-r-- 1 researcher2 research_team  46 Oct 15 05:27 project_r.txt
-rw-rw-r-- 1 researcher2 research_team  46 Oct 15 05:27 project_t.txt
researcher2@a735462327e2:~/projects$

```

The first two lines of the screenshot display the commands I entered, and the other lines display the output of the second command. I know `.project_x.txt` is a hidden file because it starts with a period (`.`). 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`.

## Change directory permissions

I only wanted the `researcher2` user to have access to the `drafts` directory and its contents. This means that no one other than `researcher2` should have execute permissions.

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

```
researcher2@a735462327e2:~/projects$ ls -l
total 20
drwx--x--- 2 researcher2 research_team 4096 Oct 15 05:27 drafts
-rw-rw-r-- 1 researcher2 research_team  46 Oct 15 05:27 project_k.txt
-rw----- 1 researcher2 research_team  46 Oct 15 05:27 project_m.txt
-rw-rw-r-- 1 researcher2 research_team  46 Oct 15 05:27 project_r.txt
-rw-rw-r-- 1 researcher2 research_team  46 Oct 15 05:27 project_t.txt
researcher2@a735462327e2:~/projects$ chmod g-x drafts
researcher2@a735462327e2:~/projects$ ls -l
total 20
drwx----- 2 researcher2 research_team 4096 Oct 15 05:27 drafts
-rw-rw-r-- 1 researcher2 research_team  46 Oct 15 05:27 project_k.txt
-rw----- 1 researcher2 research_team  46 Oct 15 05:27 project_m.txt
-rw-rw-r-- 1 researcher2 research_team  46 Oct 15 05:27 project_r.txt
-rw-rw-r-- 1 researcher2 research_team  46 Oct 15 05:27 project_t.txt
researcher2@a735462327e2:~/projects$
```

The first two lines of the screenshot display the commands I entered, and the other lines display the output of the second command. I previously determined that the group had execute permissions, so I used the `chmod` command to remove them. 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 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.