

# File permissions in Linux

## Project description

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.

## Check file and directory details

To check the file and directory details we would normally just use the command **ls**. However, since we are looking for permissions within the files and directory we would use **ls -l**. Which can be further modified by also checking for hidden files and their permissions with **ls -la**.

```
researcher2@06f6627cac89:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Jun 29 19:31 .
drwxr-xr-x 3 researcher2 research_team 4096 Jun 29 20:04 ..
-rw-rw---- 1 researcher2 research_team  46 Jun 29 19:31 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Jun 29 19:31 drafts
-rw-rw-rw- 1 researcher2 research_team  46 Jun 29 19:31 project_k.txt
-rw-r----- 1 researcher2 research_team  46 Jun 29 19:31 project_m.txt
-rw-rw-r-- 1 researcher2 research_team  46 Jun 29 19:31 project_r.txt
-rw-rw-r-- 1 researcher2 research_team  46 Jun 29 19:31 project_t.txt
```

## Describe the permissions string

To describe the permissions string we'll use the `project_k.txt` permissions. Here the first character is a `-` signalling that this is a file and not a directory. It is followed by `rw-`. This three character string means `r` for read, `w` for write, and `-` for lacking the permission to execute which would be represented by `x`. Furthermore, these three together reference the permissions the user has. The two following repeating strings of `rw-` reference group permissions and the others permissions. So, in this example all three types of users can read and write within the file.

```
researcher2@06f6627cac89:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Jun 29 19:31 .
drwxr-xr-x 3 researcher2 research_team 4096 Jun 29 20:04 ..
-rw-rw---- 1 researcher2 research_team  46 Jun 29 19:31 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Jun 29 19:31 drafts
-rw-rw-rw- 1 researcher2 research_team  46 Jun 29 19:31 project_k.txt
-rw-r----- 1 researcher2 research_team  46 Jun 29 19:31 project_m.txt
-rw-rw-r-- 1 researcher2 research_team  46 Jun 29 19:31 project_r.txt
-rw-rw-r-- 1 researcher2 research_team  46 Jun 29 19:31 project_t.txt
```

## Change file permissions

The organization does not allow others to have write access to any files. As such, we need to change the file's permissions to remove the previous permission. By utilizing the command: **chmod o-w project\_k.txt**. We do just that. Chmod is the command to manage access to files and directories, o is for others, - means the removal of permission, w is for write, and finally project\_k.txt is the file being modified.

```
researcher2@06f6627cac89:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Jun 29 19:31 .
drwxr-xr-x 3 researcher2 research_team 4096 Jun 29 20:04 ..
-rw--w---- 1 researcher2 research_team  46 Jun 29 19:31 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Jun 29 19:31 drafts
-rw-rw-rw- 1 researcher2 research_team  46 Jun 29 19:31 project_k.txt
-rw-r----- 1 researcher2 research_team  46 Jun 29 19:31 project_m.txt
-rw-rw-r-- 1 researcher2 research_team  46 Jun 29 19:31 project_r.txt
-rw-rw-r-- 1 researcher2 research_team  46 Jun 29 19:31 project_t.txt
researcher2@06f6627cac89:~/projects$ chmod o-w project_k.txt
researcher2@06f6627cac89:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Jun 29 19:31 .
drwxr-xr-x 3 researcher2 research_team 4096 Jun 29 20:04 ..
-rw--w---- 1 researcher2 research_team  46 Jun 29 19:31 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Jun 29 19:31 drafts
-rw-rw-r-- 1 researcher2 research_team  46 Jun 29 19:31 project_k.txt
-rw-r----- 1 researcher2 research_team  46 Jun 29 19:31 project_m.txt
-rw-rw-r-- 1 researcher2 research_team  46 Jun 29 19:31 project_r.txt
-rw-rw-r-- 1 researcher2 research_team  46 Jun 29 19:31 project_t.txt
researcher2@06f6627cac89:~/projects$
```

## 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. So, we use this command to do just that: **chmod u-w,g=r .project\_x.txt**. Chmod is used once again, user has its write access removed, and now with group since we need to both add read permission and remove the write permissions we can use = to overwrite all permissions to whatever follows, and finally the file itself to finish the command. Note that a . in front of a file indicates that it is hidden.

```
researcher2@06f6627cac89:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Jun 29 19:31 .
drwxr-xr-x 3 researcher2 research_team 4096 Jun 29 20:04 ..
-rw--w---- 1 researcher2 research_team  46 Jun 29 19:31 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Jun 29 19:31 drafts
-rw-rw-r-- 1 researcher2 research_team  46 Jun 29 19:31 project_k.txt
-rw-r----- 1 researcher2 research_team  46 Jun 29 19:31 project_m.txt
-rw-rw-r-- 1 researcher2 research_team  46 Jun 29 19:31 project_r.txt
-rw-rw-r-- 1 researcher2 research_team  46 Jun 29 19:31 project_t.txt
researcher2@06f6627cac89:~/projects$ chmod u-w,g=r .project_x.txt
researcher2@06f6627cac89:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Jun 29 19:31 .
drwxr-xr-x 3 researcher2 research_team 4096 Jun 29 20:04 ..
-r--r----- 1 researcher2 research_team  46 Jun 29 19:31 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Jun 29 19:31 drafts
-rw-rw-r-- 1 researcher2 research_team  46 Jun 29 19:31 project_k.txt
-rw-r----- 1 researcher2 research_team  46 Jun 29 19:31 project_m.txt
-rw-rw-r-- 1 researcher2 research_team  46 Jun 29 19:31 project_r.txt
-rw-rw-r-- 1 researcher2 research_team  46 Jun 29 19:31 project_t.txt
researcher2@06f6627cac89:~/projects$
```

## 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. We need to remove the execute access the group currently has access to. We do so with: **chmod g-x drafts**. Chmod as utilized previously, g-x is the removal of the execute access from the group, and lastly drafts is the directory.

```
researcher2@06f6627cac89:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Jun 29 19:31 .
drwxr-xr-x 3 researcher2 research_team 4096 Jun 29 20:04 ..
-r--r----- 1 researcher2 research_team  46 Jun 29 19:31 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Jun 29 19:31 drafts
-rw-rw-r-- 1 researcher2 research_team  46 Jun 29 19:31 project_k.txt
-rw-r----- 1 researcher2 research_team  46 Jun 29 19:31 project_m.txt
-rw-rw-r-- 1 researcher2 research_team  46 Jun 29 19:31 project_r.txt
-rw-rw-r-- 1 researcher2 research_team  46 Jun 29 19:31 project_t.txt
researcher2@06f6627cac89:~/projects$ ^C
researcher2@06f6627cac89:~/projects$ chmod g-x drafts
researcher2@06f6627cac89:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Jun 29 19:31 .
drwxr-xr-x 3 researcher2 research_team 4096 Jun 29 20:04 ..
-r--r----- 1 researcher2 research_team  46 Jun 29 19:31 .project_x.txt
drwx----- 2 researcher2 research_team 4096 Jun 29 19:31 drafts
-rw-rw-r-- 1 researcher2 research_team  46 Jun 29 19:31 project_k.txt
-rw-r----- 1 researcher2 research_team  46 Jun 29 19:31 project_m.txt
GX -rw-rw-r-- 1 researcher2 research_team  46 Jun 29 19:31 project_r.txt
... -rw-rw-r-- 1 researcher2 research_team  46 Jun 29 19:31 project_t.txt
researcher2@06f6627cac89:~/projects$
```

## Summary

With the concept of least privilege in mind, access management is a huge component of ensuring solid cyber security. Linux allows us to put this concept to practice and allows us to ensure users only have access to what they need for them to complete their job.