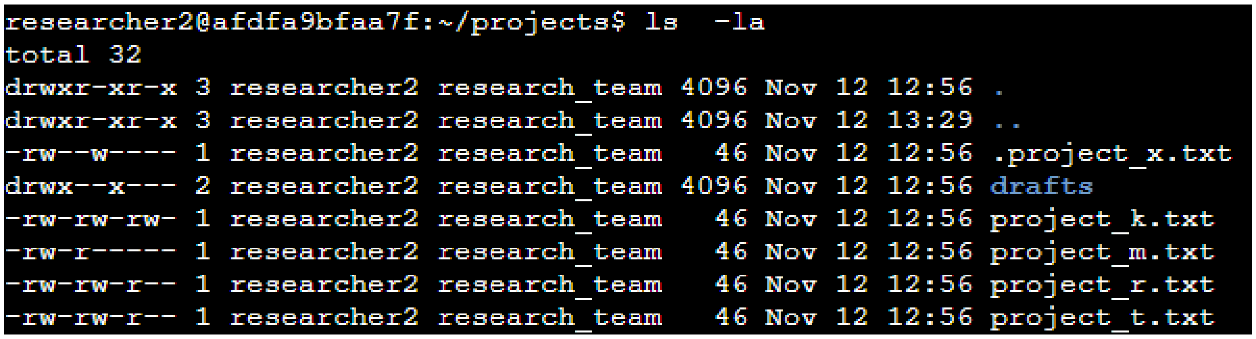
# File permissions in Linux

## Project description

The research team at our organization needed to fix file permissions for certain files and directories in the projects directory to improve system security. To complete this task, I did the following:

## 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@afdfa9bfaa7f:~/projects$ ls  -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Nov 12 12:56 .
drwxr-xr-x 3 researcher2 research_team 4096 Nov 12 13:29 ..
-rw--w---- 1 researcher2 research_team   46 Nov 12 12:56 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Nov 12 12:56 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Nov 12 12:56 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Nov 12 12:56 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Nov 12 12:56 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Nov 12 12:56 project_t.txt
```

I used the command `ls -la` to check existing permissions for the projects directory. This showed one directory called drafts, a hidden file `.project_x.txt`, and five other project files. The 10-character string in the first column represented the permissions for each file or directory.

***The 10-character string breakdown:***

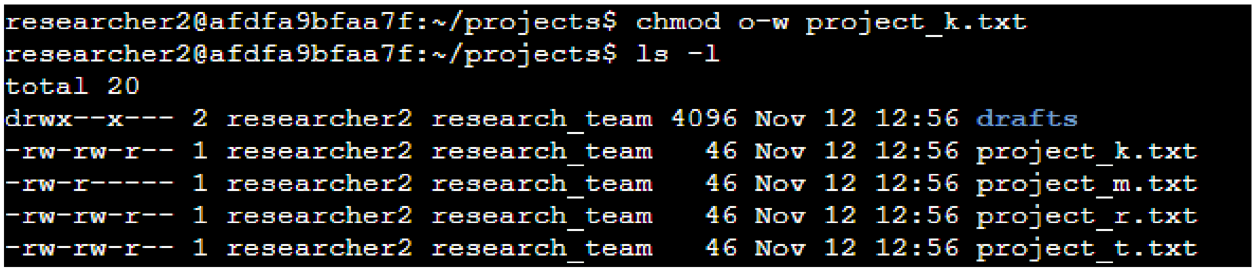
1st character: Indicates file type (`d` for directory, `–` for a regular file).
2nd-4th characters: `U`ser's read (`r`), write (`w`), and execute (`x`) permissions.
5th-7th characters: `G`roup's read, write, and execute permissions.
8th-10th characters: `O`thers' read, write, and execute permissions.

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 - Removing Other's Write Access

Following the organization's decision, I used the `chmod` command to remove write access for others on the file `project_k.txt`.
The following code demonstrates how I used Linux commands to do this:



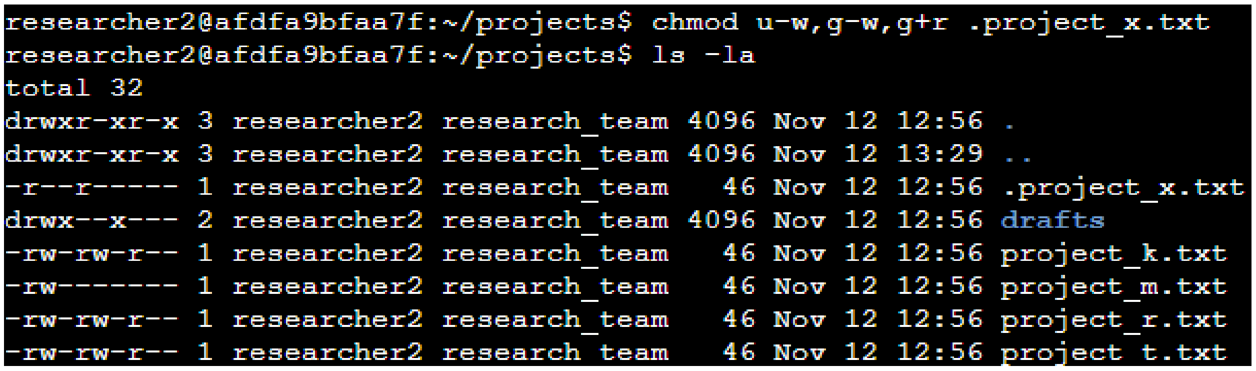
```
researcher2@afdfa9bfaa7f:~/projects$ chmod o-w project_k.txt
researcher2@afdfa9bfaa7f:~/projects$ ls -l
total 20
drwx--x--- 2 researcher2 research_team 4096 Nov 12 12:56 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Nov 12 12:56 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Nov 12 12:56 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Nov 12 12:56 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Nov 12 12:56 project_t.txt
```

The commands I typed are shown in the first two lines of the screenshot, and the result of the second command is shown in the remaining lines. The permissions of files and directories can be modified with the `chmod` command. The file or directory is specified by the second argument, while the first argument defines which permissions need to be adjusted. For the `project_k.txt` file in this example, I deleted the write permissions from other. I then used `ls -la` to examine the changes I had made.

## Change Hidden File Permissions - Restricting Write Access

`Project_x.txt` was recently archived by my organization's research team. The user and group should have `read` access to this project; they do not want anyone to have `write` access.

The code that follows shows how I modified the permissions using Linux commands:



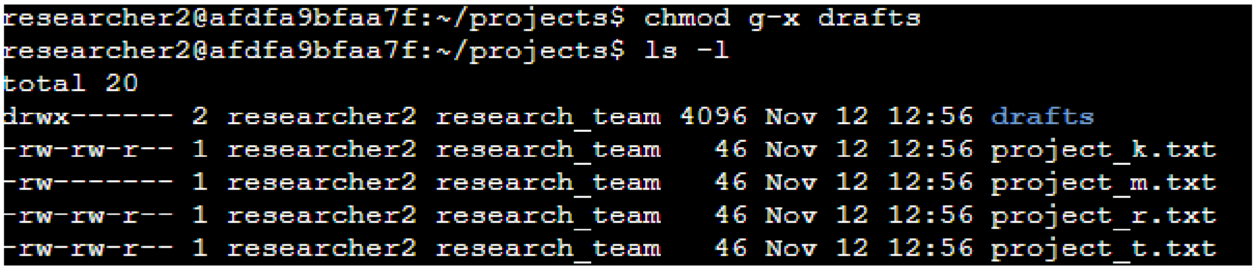
```
researcher2@afdfa9bfaa7f:~/projects$ chmod u-w,g-w,g+r .project_x.txt
researcher2@afdfa9bfaa7f:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Nov 12 12:56 .
drwxr-xr-x 3 researcher2 research_team 4096 Nov 12 13:29 ..
-r--r----- 1 researcher2 research_team   46 Nov 12 12:56 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Nov 12 12:56 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Nov 12 12:56 project_k.txt
-rw------- 1 researcher2 research_team   46 Nov 12 12:56 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Nov 12 12:56 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Nov 12 12:56 project_t.txt
```

The commands I typed are shown in the first two lines of the screenshot, and the result of the second command is shown in the remaining lines. Here `project_x.txt` begins with a period (`.`), it is a hidden file. In this example, I gave the group read permissions and took away the user's and group's write capabilities. I took away the user's `u-w` write permissions. Next, I gave the group with `g+r` read capabilities and withdrew write permissions from it.

## Change directory permissions

To ensure only the `researcher2` user had execute permissions for the drafts directory, I removed execute permissions for the group.

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



```
researcher2@afdfa9bfaa7f:~/projects$ chmod g-x drafts
researcher2@afdfa9bfaa7f:~/projects$ ls -l
total 20
drwx------ 2 researcher2 research_team 4096 Nov 12 12:56 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Nov 12 12:56 project_k.txt
-rw------- 1 researcher2 research_team   46 Nov 12 12:56 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Nov 12 12:56 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Nov 12 12:56 project_t.txt
```

The commands I typed are shown in the first two lines of the screenshot, and the result of the second command is shown in the remaining lines. I used the `chmod` command to remove the group's execute permissions after discovering earlier that they were there. It was not necessary to add execute rights because the `researcher2` user already has them.

## Summary

I adjusted multiple permissions in the projects directory using `ls -la` to check and `chmod` to modify permissions. These changes aligned with the desired authorization levels.

Alexander Lungu