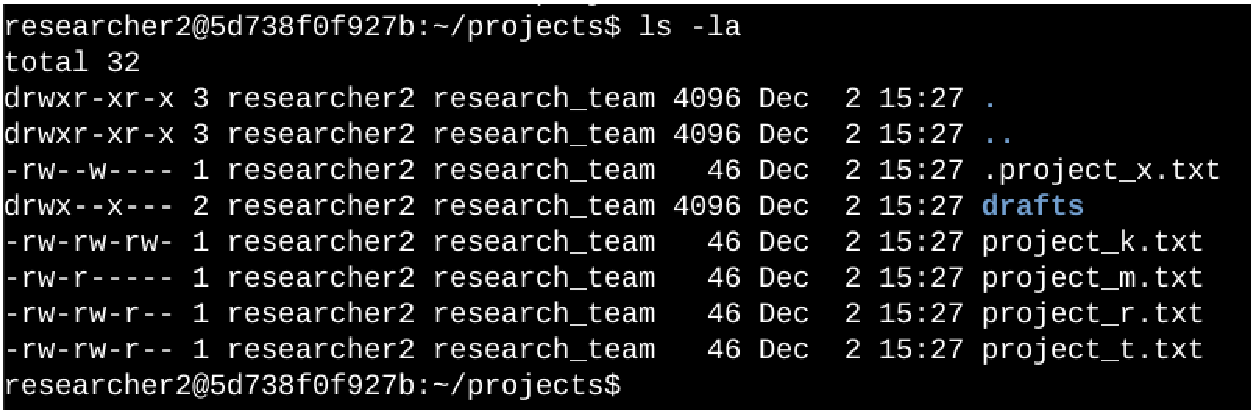
File permissions in Linux

Project description

The research team at my organization needs to update the permissions of 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 these permissions will help keep the system secure. To complete the task, I performed the following actions:

Check file and directory details

The Screenshot below shows which Linux command I used to list the existing permissions for specific directory in the file system.



```
researcher2@5d738f0f927b:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Dec  2 15:27 .
drwxr-xr-x 3 researcher2 research_team 4096 Dec  2 15:27 ..
-rw--w---- 1 researcher2 research_team   46 Dec  2 15:27 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Dec  2 15:27 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Dec  2 15:27 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Dec  2 15:27 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Dec  2 15:27 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Dec  2 15:27 project_t.txt
researcher2@5d738f0f927b:~/projects$
```

The first line of the screenshot displays the Linux command I entered. After running the command, it lists all contents with permissions of the `projects` directory. I used the `ls` command with the `-la` option to display a detailed listing of the file contents which also shows hidden files. The output indicates that there is one directory named `drafts`, one hidden file named `.project_x.txt`, and five other project files. The 10-character string in the first column shows the permissions of each file or directory.

Describe the permissions string

The 10-character string is used to determine who is authorized to access the file and their specific permissions. The characters and what they represent are as follows:

- **1st character**: This character is either a `d` or hyphen (`-`) and indicates the file type. If it's a `d`, it's a directory. If it's a hyphen (`-`), it's a regular file.

- **2nd-4th characters**: These characters indicate the read (`r`), write (`w`), and execute (`x`) permissions for the user. When one of these characters is a hyphen (`-`), it indicates that this permission is not granted to the user.

- **5th-7th characters:** 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 granted for the group.

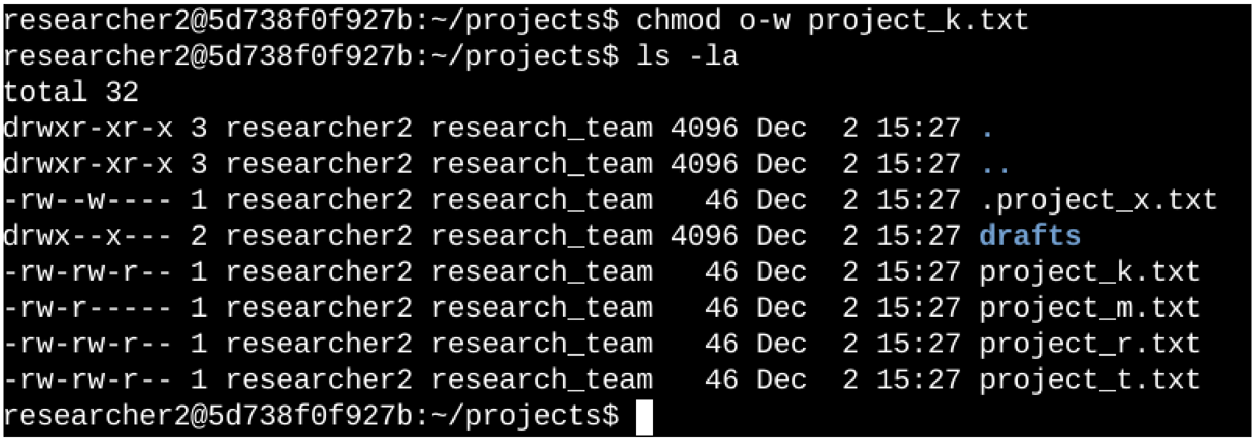
- **8th-10th characters:** These characters indicate the read (`r`), write (`w`), and execute (`x`) permissions for other. This owner type consists of all other users on the system apart from the user and the group. When one of these characters is a hyphen (`-`), it indicates that this permission is not granted for other.

For example, the file permissions for `project_r.txt` are `-rw-rw-r--`. Since the first character is a hyphen (`-`), it indicates that `project_r.txt` is a file. 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_r.txt`.

Change file permissions

The organization determined that other shouldn't have write access to any of their files. To comply with this, I referred to the file permissions. I determined that other must not have write access permissions for `project_k.txt.`

The following screenshot shows how I used Linux commands to do remove the permissions:



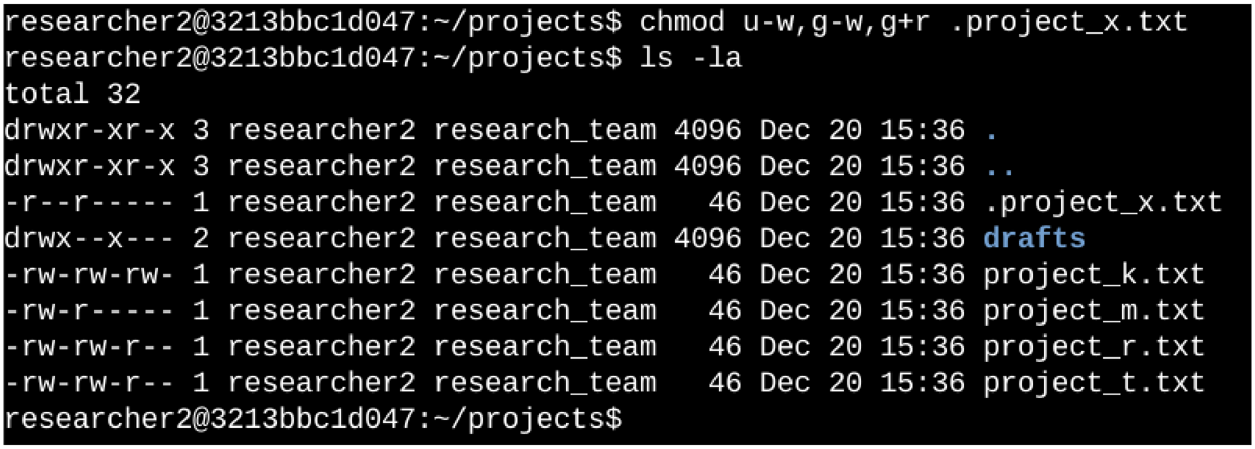
```
researcher2@5d738f0f927b:~/projects$ chmod o-w project_k.txt
researcher2@5d738f0f927b:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Dec  2 15:27 .
drwxr-xr-x 3 researcher2 research_team 4096 Dec  2 15:27 ..
-rw--w---- 1 researcher2 research_team   46 Dec  2 15:27 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Dec  2 15:27 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Dec  2 15:27 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Dec  2 15:27 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Dec  2 15:27 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Dec  2 15:27 project_t.txt
researcher2@5d738f0f927b:~/projects$ 
```

The first two lines of the screenshot display the commands I entered, and the other lines display the output of the second command, as the first line command doesn’t return any output. The `chmod` command changes the permissions of files and directories. The first argument indicates what permissions should be changed, and the second argument specifies which file or directory permissions to be changed. In this example, I removed write permissions from other for the `project_k.txt` file. After this, I used `ls -la` command to review the changes i made.

Change file permissions on a hidden file

The research team at my organization recently archived `project_x.txt`. They do not want anyone to have write access to this project file, but the user and group should have read access.

The following screenshot shows how I used Linux commands to change the permissions:



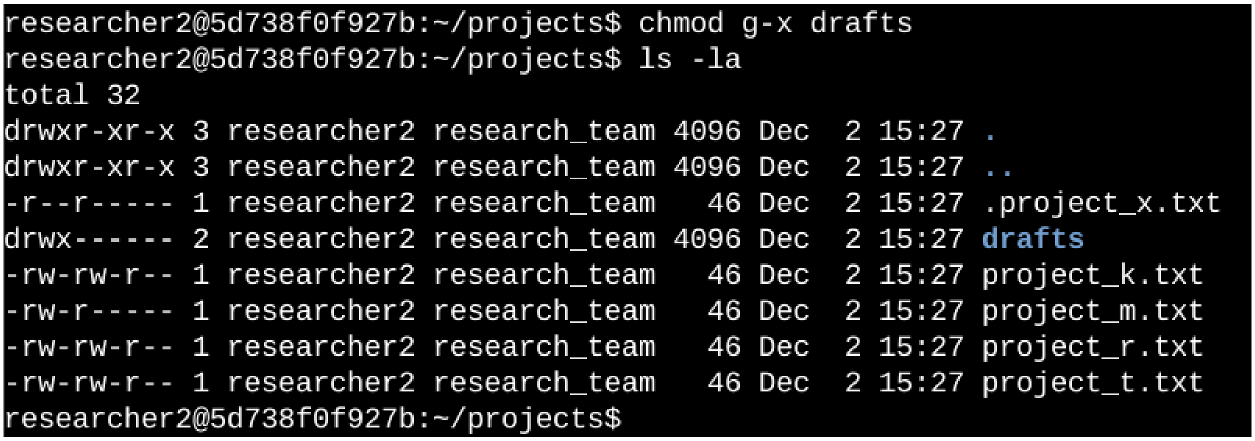
```
researcher2@3213bbc1d047:~/projects$ chmod u-w,g-w,g+r .project_x.txt
researcher2@3213bbc1d047:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Dec 20 15:36 .
drwxr-xr-x 3 researcher2 research_team 4096 Dec 20 15:36 ..
-r--r----- 1 researcher2 research_team   46 Dec 20 15:36 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Dec 20 15:36 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Dec 20 15:36 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Dec 20 15:36 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Dec 20 15:36 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Dec 20 15:36 project_t.txt
researcher2@3213bbc1d047:~/projects$
```

The first two lines of the screenshot display the commands I entered, and the other lines display the output of the second command, as the first line command doesn't return any output. I know `.project_x.txt` is a hidden file because it starts with a period (`.`). I removed write permissions from the user with argument `(u-w)` and group with argument (`g-w)`, and added read permissions to the group with argument (`g+r) .`

Change directory permissions

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

The following screenshot shows how I used Linux commands to change the permissions:



```
researcher2@5d738f0f927b:~/projects$ chmod g-x drafts
researcher2@5d738f0f927b:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Dec  2 15:27 .
drwxr-xr-x 3 researcher2 research_team 4096 Dec  2 15:27 ..
-r--r----- 1 researcher2 research_team   46 Dec  2 15:27 .project_x.txt
drwx------ 2 researcher2 research_team 4096 Dec  2 15:27 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Dec  2 15:27 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Dec  2 15:27 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Dec  2 15:27 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Dec  2 15:27 project_t.txt
researcher2@5d738f0f927b:~/projects$
```

The first two lines of the screenshot display the commands I entered, and the other lines display the output of the second command, as the first line command doesn't return any output. I used `chmod` command to remove execute permissions from the group, as other doesn’t have any permissions to be changed or removed and the `researcher2` user has already execute permissions, so they did not need to be added.

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 was using `ls -la` to check the permissions for the files and directory. Then I used `chmod` command multiple times to change the permissions on files and directories to match the appropriate authorization level.