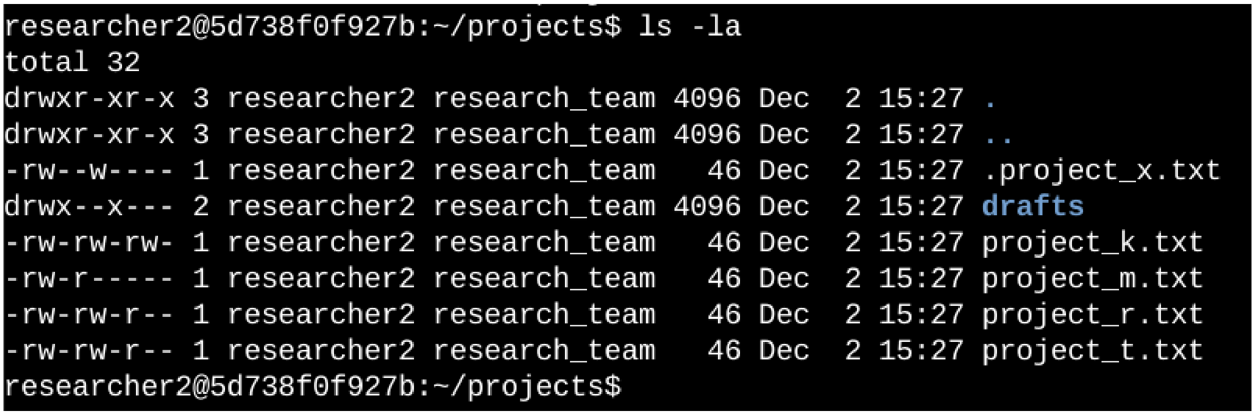
File permissions in Linux

Project description

The `projects` directory contains files and folders that require the file permissions to be updated by the research team at my organization. Right now, the permissions do not correspond to the appropriate level of authority. Their system will remain secure if these permissions are checked and updated. I completed the following tasks in order to finish this task:

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@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 command I typed is shown in the first line of the screenshot, and the output is shown in the remaining lines. The `projects` directory's contents are all listed in the code. I was able to see a comprehensive list of the file contents—which included hidden files—by using the ls program with the `-la` option. My command produced the following output: one secret file called `.project_x.txt`, five other project files, and a subfolder called `drafts`. The permissions assigned on each file or directory are represented by the 10-character string in the first column.

Describe the permissions string

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

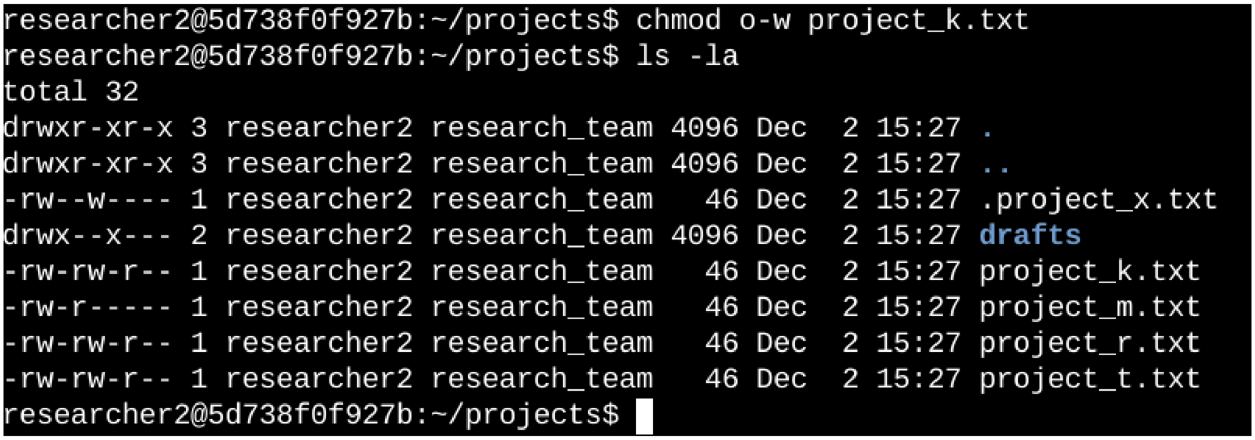
- **1st character**: This character is either a d or hyphen (-) and indicates the file type. If it's a d, it's a directory. If it's a hyphen (-), it's a regular file.
- **2nd-4th characters**: Read (r), write (w), and execute (x) are the characters that represent the user's permissions. This permission is not provided to the user when one of these characters is a hyphen (-).
- **5th-7th characters:** The read (r), write (w), and execute (x) permissions for the group are indicated by these characters. If one of these characters is a hyphen (-) instead, it means that the group is not eligible for this permission.
- **8th-10th characters:** The read (r), write (w), and execute (x) rights for others are indicated by these characters. All other users on the system excluding the user and the group make up this owner type. In the event that one of these characters is a hyphen (-), it signifies that permission for other characters is not given.

`Project_t.txt`, for instance, has file permissions of `-rw-rw-r--`. `Project_t.txt` is a file, not a directory, as shown by the hyphen (-) that appears in the first character. The characters `r` in the second, fifth, and eighth positions signify that read permissions are granted to the user, group, and other. Only the user and group have write rights, as indicated by the third and sixth characters, which are w. For `project_t.txt`, no one has execute permissions.

Change file permissions

The company came to the conclusion that no one else should be able to write to any of their files. I used the file permissions that I had previously returned to comply with this. I found that the write access to `project_k.txt` needs to be deleted for other users.

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



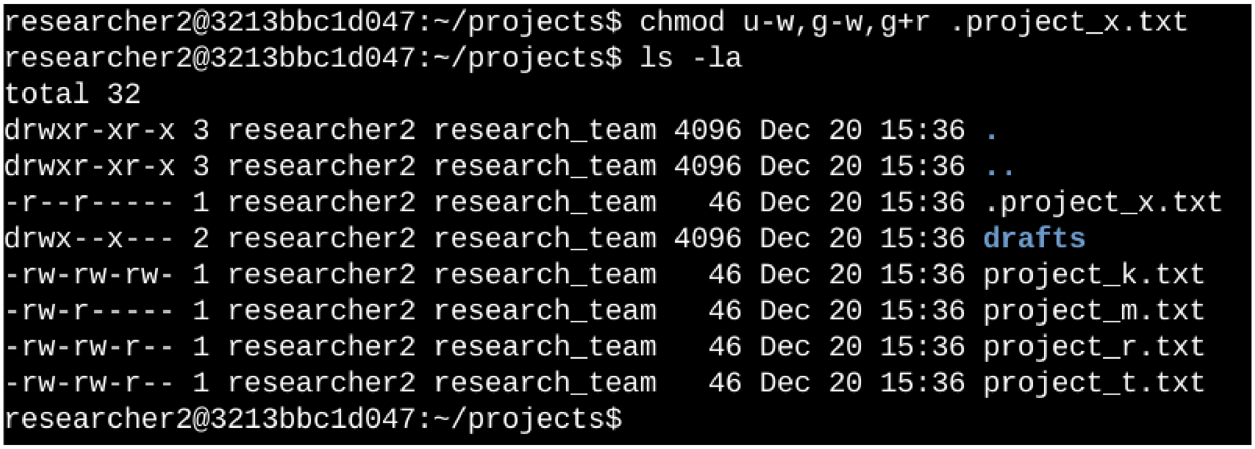
```
researcher2@5d738f0f927b:~/projects$ chmod o-w project_k.txt
researcher2@5d738f0f927b:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Dec  2 15:27 .
drwxr-xr-x 3 researcher2 research_team 4096 Dec  2 15:27 ..
-rw--w---- 1 researcher2 research_team   46 Dec  2 15:27 .project_x.txt
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 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 file permissions on a hidden file

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



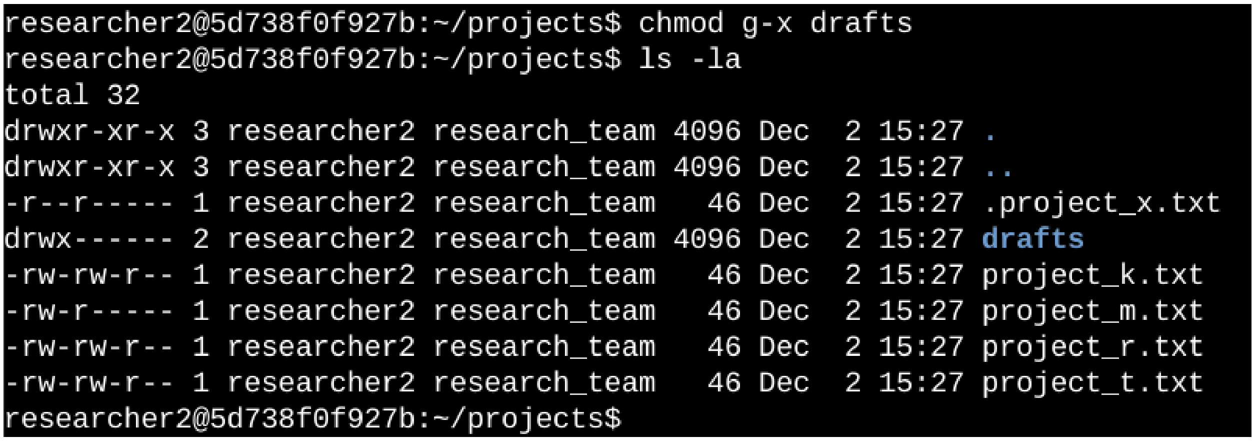
```
researcher2@3213bbc1d047:~/projects$ chmod u-w,g-w,g+r .project_x.txt
researcher2@3213bbc1d047:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Dec 20 15:36 .
drwxr-xr-x 3 researcher2 research_team 4096 Dec 20 15:36 ..
-r--r----- 1 researcher2 research_team   46 Dec 20 15:36 .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 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 am aware.Because `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

The `researcher2` user should only have access to the `drafts` directory and its contents, per our organization's policy. This indicates that execute permissions should only be granted to `researcher2`.

The following code demonstrates 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 permission listing for a number of files and folders is shown in this output. The parent directory (home) is shown in line 2, while the current directory (projects) is shown in line 1. A regular file called ".`project_x.txt`" is indicated on line 3. The directory with restricted permissions, "`drafts`," is located on line 4. As you can see, only researcher 2 is authorized to execute commands. I used the `chmod` command to delete the group's execute permissions since it had previously been established that they existed. It was not necessary to add execute rights because the `researcher2` user already has them.

Summary

I made numerous changes to permissions in order to provide files and directories in the projects directory the amount of access that my business required. Checking the directory's permissions with `ls -la` was the initial step in this process. This helped me make decisions about what to do next. I then changed the permissions of files and directories several times using the `chmod` command.