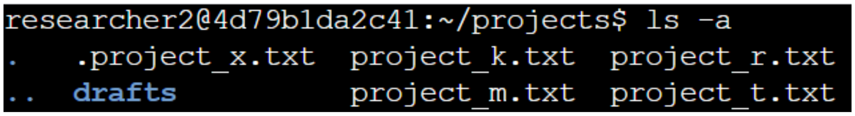
File permissions in Linux

Project description

As a security professional at my company, I mainly work with the research team. Part of my job involves ensuring users on this team are authorized with the appropriate permissions. This helps keep the system secure. For this project, I will examine existing permissions through bash. I will determine if the permissions match the authorization that should be given. If they do not match, I will be required to modify the permissions to authorize the appropriate users and remove any unauthorized access. This document displays the file structure of the `/home/researcher2/projects` directory and the permissions of the files and subdirectory it contains.

Check file and directory details

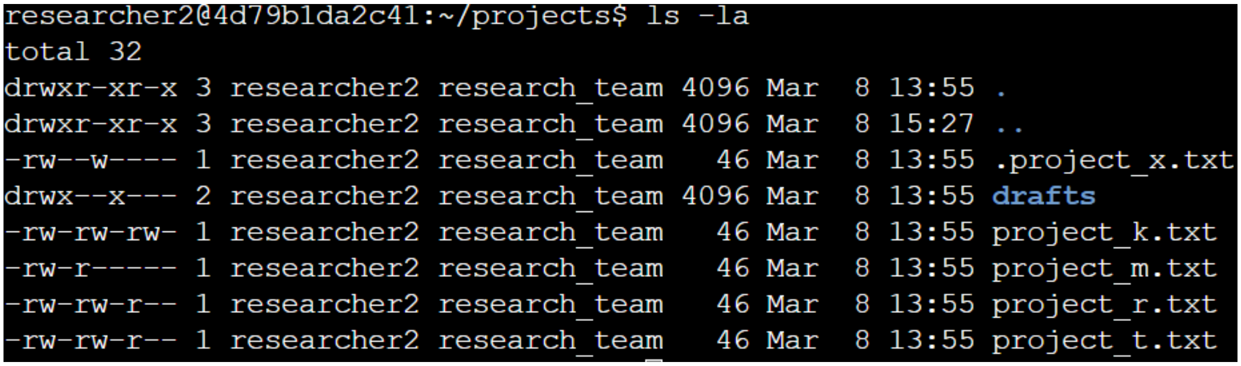
Since I am already within the `projects` directory, I will key in `ls -a` to list down all file and directory within `projects` including any hidden file.



```
researcher2@4d79b1da2c41:~/projects$ ls -a
.   .project_x.txt  project_k.txt  project_r.txt
..  drafts          project_m.txt  project_t.txt
```

From this command, it gives a result of 1 subdirectory `drafts` and 5 files (`project_k.txt, project_m.txt, project_r.txt, project_t.txt, .project_x.txt)`
As shown, `.project_x.txt` is a hidden file.

Next, I would need to check the permissions of the directory and files. To do this, I will key in `ls -la` to display permissions to all files and directories within `projects` including the hidden file.



```
researcher2@4d79b1da2c41:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Mar  8 13:55 .
drwxr-xr-x 3 researcher2 research_team 4096 Mar  8 15:27 ..
-rw--w---- 1 researcher2 research_team   46 Mar  8 13:55 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Mar  8 13:55 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Mar  8 13:55 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Mar  8 13:55 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Mar  8 13:55 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Mar  8 13:55 project_t.txt
```

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

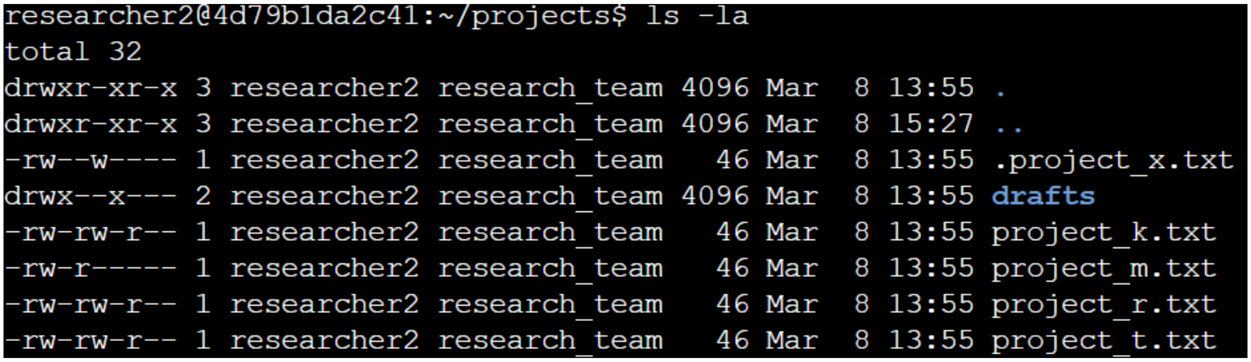
From the results, we can view the permissions for the individual directory and files. Let's take one as an example for us to understand the command better. For the file `project_k.txt` ,the permission is displayed as `-rw-rw-rw-`
The first character indicates the file type. `-` means that it is a file, not a directory.
The 2nd-4th character, is for the user's permission. `rw` means the user has read and write permission. A hyphen `-` means that the executable permission is not granted to the user.
The 5th-7th character, is for the group's permission, which has the same permission as user, which is only read and write permission.
The 8th-10th character, is for the other's permission, which has the same permission as user and group, which is only read and write permission.

Change file permissions

My organization does not allow others to have write access to any files. Based on the permissions established in previous step, we know that the file `project_k.txt` needs to have its permissions modified, since it has write permission for other. To remove the write permission, we use the Linux command `chmod o-w project_k.txt`

The command `chmod` will change permissions on files and directories.
The following `o-w` means that for other, we would like to remove its write permission. Next we will need to indicate in which file that we would like to make those changes to. In this case, it is `project_k.txt`

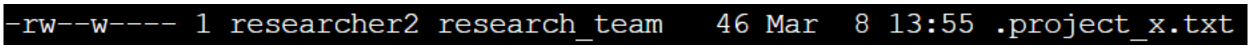
Once we have given the command to modify the permission, we can check the latest file and directory permissions by giving the command `ls -la.`



```
researcher2@4d79b1da2c41:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Mar  8 13:55 .
drwxr-xr-x 3 researcher2 research_team 4096 Mar  8 15:27 ..
-rw--w---- 1 researcher2 research_team   46 Mar  8 13:55 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Mar  8 13:55 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Mar  8 13:55 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Mar  8 13:55 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Mar  8 13:55 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Mar  8 13:55 project_t.txt
```

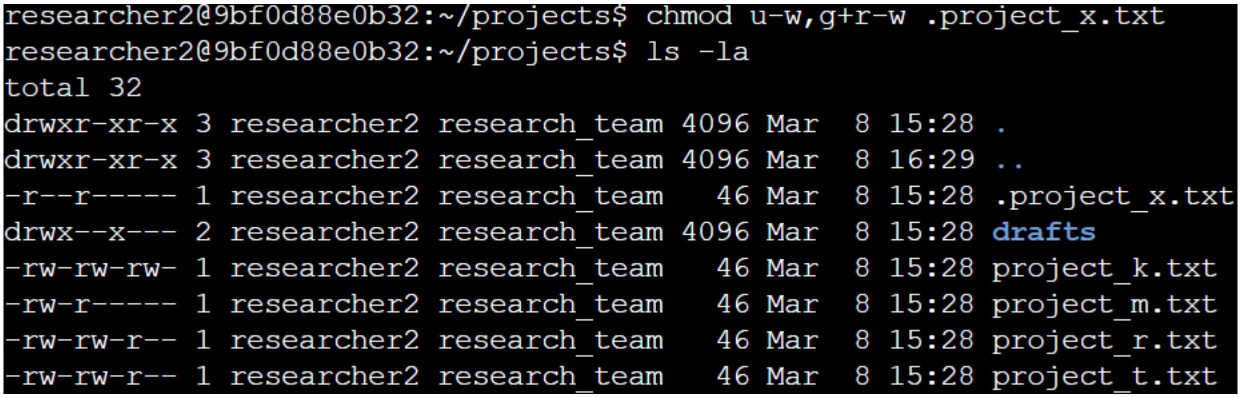
Change file permissions on a hidden file

The research team has archived `.project_x.txt`, hence why it is kept hidden. This file should not have write permissions for anyone, but the user and group should be able to read the file. Currently, we can see within the file, that the user has read and write permission, and group has only write permission.



```
-rw--w---- 1 researcher2 research_team   46 Mar  8 13:55 .project_x.txt
```

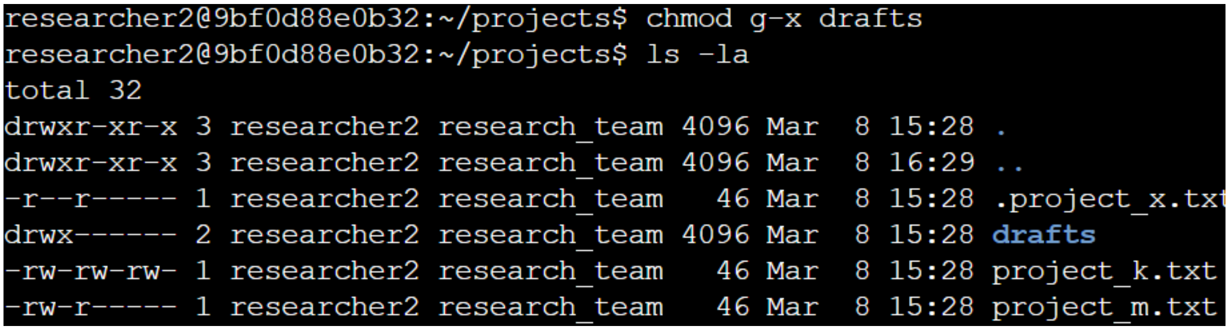
To modify the permissions, we can still use the `chmod` command as follows: `chmod u-w, g+r-w .project_x.txt.` Once we have given the command to modify the permission, we can check the latest file and directory permissions by giving the command `ls -la.`



```
researcher2@9bf0d88e0b32:~/projects$ chmod u-w,g+r-w .project_x.txt
researcher2@9bf0d88e0b32:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Mar  8 15:28 .
drwxr-xr-x 3 researcher2 research_team 4096 Mar  8 16:29 ..
-r--r----- 1 researcher2 research_team   46 Mar  8 15:28 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Mar  8 15:28 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Mar  8 15:28 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Mar  8 15:28 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Mar  8 15:28 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Mar  8 15:28 project_t.txt
```

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. For this purpose, we use the `chmod` command as follows: `chmod g-x drafts`



```
researcher2@9bf0d88e0b32:~/projects$ chmod g-x drafts
researcher2@9bf0d88e0b32:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Mar  8 15:28 .
drwxr-xr-x 3 researcher2 research_team 4096 Mar  8 16:29 ..
-r--r----- 1 researcher2 research_team   46 Mar  8 15:28 .project_x.txt
drwx------ 2 researcher2 research_team 4096 Mar  8 15:28 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Mar  8 15:28 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Mar  8 15:28 project_m.txt
```

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 in this was using `ls -la` to check the permissions for the directory. This helps to further confirm that I have made the necessary changes. I then used the `chmod` command multiple times to change the permissions on files and directories, either to remove or to add.