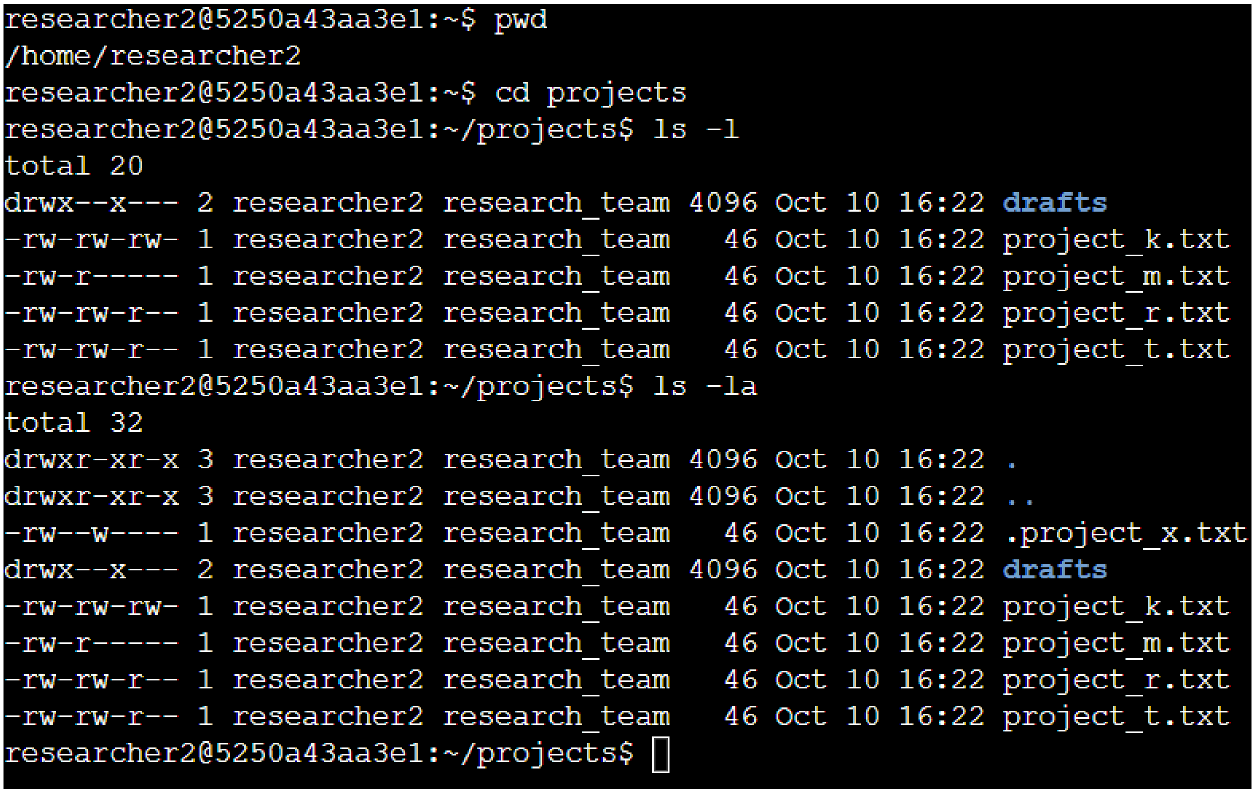
File permissions in Linux

Project description

The research team at my organization needs to update the file permissions for 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 their system secure. To complete this task, I performed the following tasks:

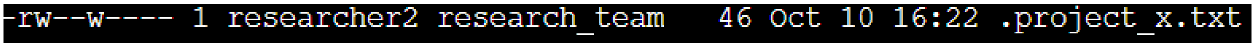
Check file and directory details



```
researcher2@5250a43aa3e1:~$ pwd
/home/researcher2
researcher2@5250a43aa3e1:~$ cd projects
researcher2@5250a43aa3e1:~/projects$ ls -l
total 20
drwx--x--- 2 researcher2 research_team 4096 Oct 10 16:22 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Oct 10 16:22 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Oct 10 16:22 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct 10 16:22 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct 10 16:22 project_t.txt
researcher2@5250a43aa3e1:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Oct 10 16:22 .
drwxr-xr-x 3 researcher2 research_team 4096 Oct 10 16:22 ..
-rw--w---- 1 researcher2 research_team   46 Oct 10 16:22 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Oct 10 16:22 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Oct 10 16:22 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Oct 10 16:22 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct 10 16:22 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct 10 16:22 project_t.txt
researcher2@5250a43aa3e1:~/projects$
```

You can use the `ls -l command` to display permissions to files and directories. Additionally, you can use the `ls -la command` to display permissions to files and directories, including hidden files.

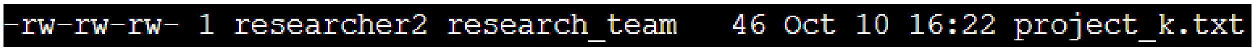
Describe the permissions string



```
-rw--w---- 1 researcher2 research_team   46 Oct 10 16:22 .project_x.txt
```

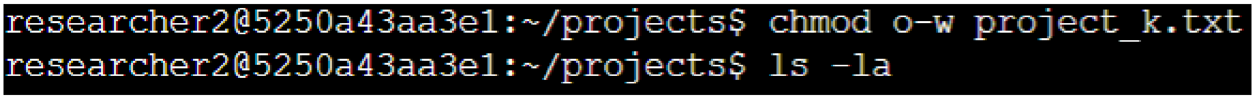
A 10-character string begins each entry and indicates how the permissions on the file are set. This example is a regular file, as the first character is a hyphen. The second and third characters show that the user has read and write permissions. Moving on to the group owner type, the second character in this group indicates the user has write permissions. Lastly, the 8th-10th characters indicate permissions for the owner type of other. In this case, these last three characters have hyphens highlighting that the permissions of read, write, and execute are not granted for the users of the owner type of other.

Change file permissions

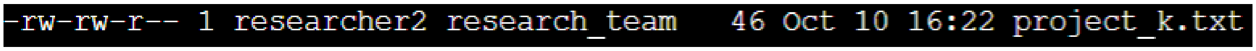


```
-rw-rw-rw- 1 researcher2 research_team   46 Oct 10 16:22 project_k.txt
```

My organization does not allow the other owner type to have write access to any files.



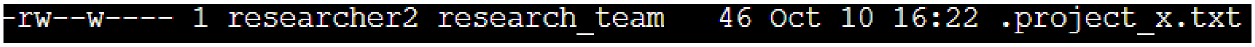
```
researcher2@5250a43aa3e1:~/projects$ chmod o-w project_k.txt
researcher2@5250a43aa3e1:~/projects$ ls -la
```



```
-rw-rw-r-- 1 researcher2 research_team   46 Oct 10 16:22 project_k.txt
```

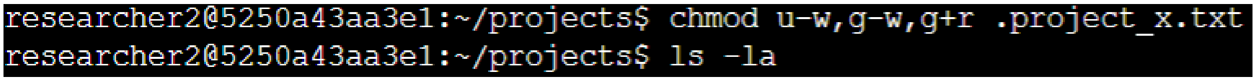
I used the `chmod` command as it is used to change permissions on files. I also used the `ls -la` command to check that the `project_k.txt` file no longer had write permissions in the other owner type section.

Change file permissions on a hidden file

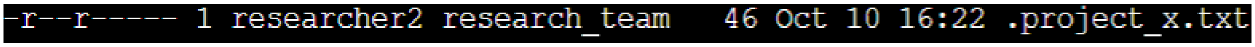


```
-rw--w---- 1 researcher2 research_team   46 Oct 10 16:22 .project_x.txt
```

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.



```
researcher2@5250a43aa3e1:~/projects$ chmod u-w,g-w,g+r .project_x.txt
researcher2@5250a43aa3e1:~/projects$ ls -la
```



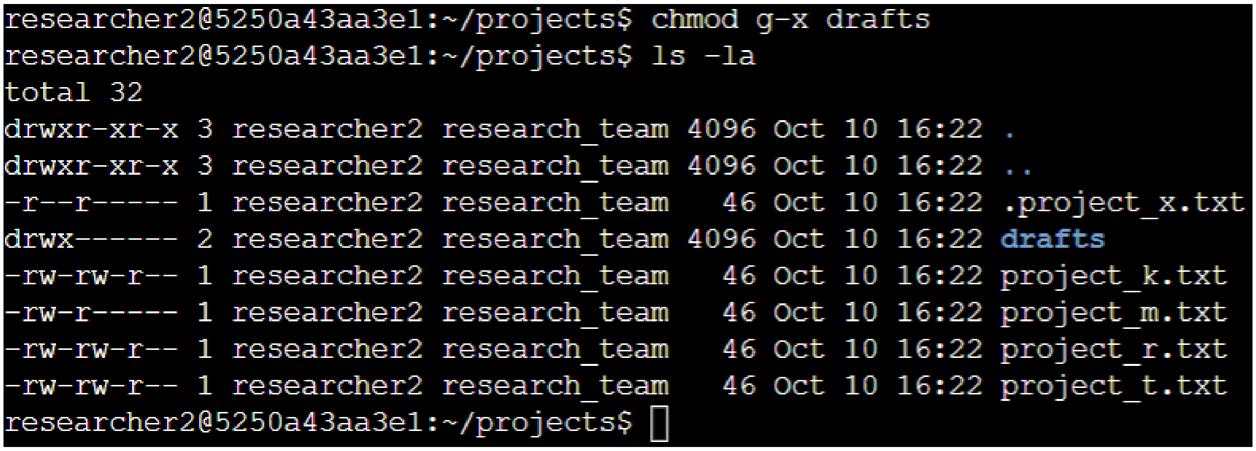
```
-r--r----- 1 researcher2 research_team   46 Oct 10 16:22 .project_x.txt
```

I used the `chmod` command to change the permissions for the `.project_x.txt` hidden file to remove the write permissions for each owner type as well as ensuring the user and group owner types had read permissions. I checked to see if the changes were properly made using the `ls -la` command.

Change directory permissions

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

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



```
researcher2@5250a43aa3e1:~/projects$ chmod g-x drafts
researcher2@5250a43aa3e1:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Oct 10 16:22 .
drwxr-xr-x 3 researcher2 research_team 4096 Oct 10 16:22 ..
-r--r----- 1 researcher2 research_team   46 Oct 10 16:22 .project_x.txt
drwx------ 2 researcher2 research_team 4096 Oct 10 16:22 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Oct 10 16:22 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Oct 10 16:22 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct 10 16:22 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct 10 16:22 project_t.txt
researcher2@5250a43aa3e1:~/projects$ 
```

The first two lines of the screenshot display the commands I entered, and the other lines display the output of the second command. I previously determined that the group had execute permissions, so I used the `chmod` command to remove them. The `researcher2` user already had 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 in this was using `ls -la` to check the permissions for the directory. This informed my decisions in the following steps. I then used the `chmod` command multiple times to change the permissions on files and directories.