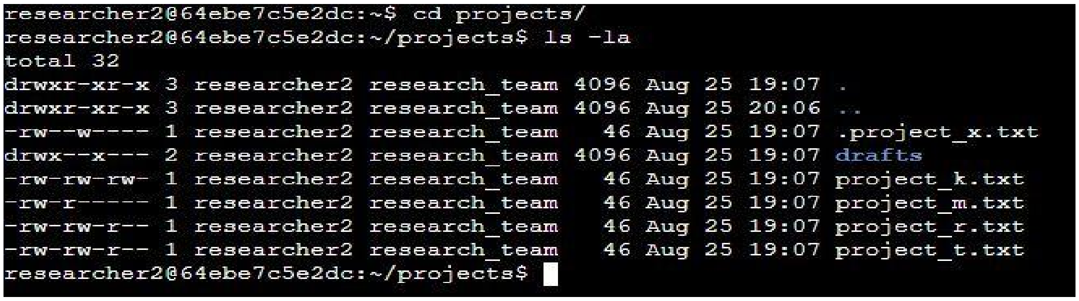
File permissions in Linux

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, I performed the following tasks:

Check file and directory details

I used the command: `cd projects/` to navigate into the projects directory, I then used the command: `ls -la` to display all permissions for the files and directories inside, including hidden files and directories (prefixed with a .) inside the projects directory.



```
researcher2@64ebe7c5e2dc:~$ cd projects/
researcher2@64ebe7c5e2dc:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Aug 25 19:07 .
drwxr-xr-x 3 researcher2 research_team 4096 Aug 25 20:06 ..
-rw--w---- 1 researcher2 research_team   46 Aug 25 19:07 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Aug 25 19:07 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Aug 25 19:07 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Aug 25 19:07 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Aug 25 19:07 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Aug 25 19:07 project_t.txt
researcher2@64ebe7c5e2dc:~/projects$ 
```

The output of my command shows 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 represents the permissions for each file or directory.

The current permissions show that:

- Owner type user has read & write permissions for all files and directories in this directory as indicated by the 2nd and 3rd characters of each of the permissions strings containing read (`r`) & write (`w`) respectively.

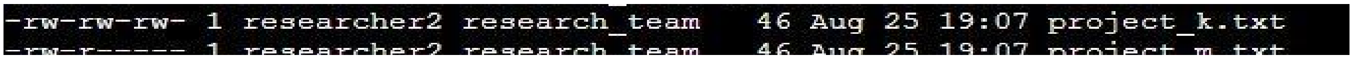
- Owner type group has read permissions for all non hidden files as indicated by the 5th character of each of the permissions strings containing read (`r`). group also has write (`w`) permissions for the files `project_k.txt`, `project_r.txt`, and `project_t.txt` as well as the hidden file `.project_x.txt` as indicated by the 6th character of the permissions strings containing write (`w`). Lastly, group has executable permissions for the drafts directory as indicated by the 7th character of the permissions string containing executable (x).

- Owner type other has read permissions for the files project_k.txt, project_r.txt and project_t.txt as indicated by the 8th character in the permissions strings containing read (r). other also has write permissions for the file project_k.txt as indicated by the 9th character of the permissions string containing write (w).

Describe the permissions string

The permissions string is a 10 character string that displays the permissions for each owner group, the characters and what they represent are as follows:

- **1st** character indicates if it is a file or directory, directories are indicated by the first character being d and a file is indicated by the first character being a hyphen (-). The remaining 9 characters show the read, write and executable permissions for each owner type.
- **2nd, 3rd and 4th** characters show read (r), write (w) and executable (x) permissions for the owner type user. When one of these characters is a hyphen (-) instead, it indicates that this permission is not granted to the user.
- **5th, 6th and 7th** characters show read (r), write (w) and executable (x) permissions for the owner type group. When one of these characters is a hyphen (-) instead, it indicates that this permission is not granted to the group.
- **8th, 9th and 10th** characters of the string show read (r), write (w) and executable (x) permissions for the owner type other, this owner type consists of any other users that are not in the user or group types. When one of these characters is a hyphen (-) instead, it indicates that this permission is not granted for other.



```
-rw-rw-rw- 1 researcher2 research_team    46 Aug 25 19:07 project_k.txt
-rw-r----- 1 researcher2 research_team    46 Aug 25 19:07 project_m.txt
```

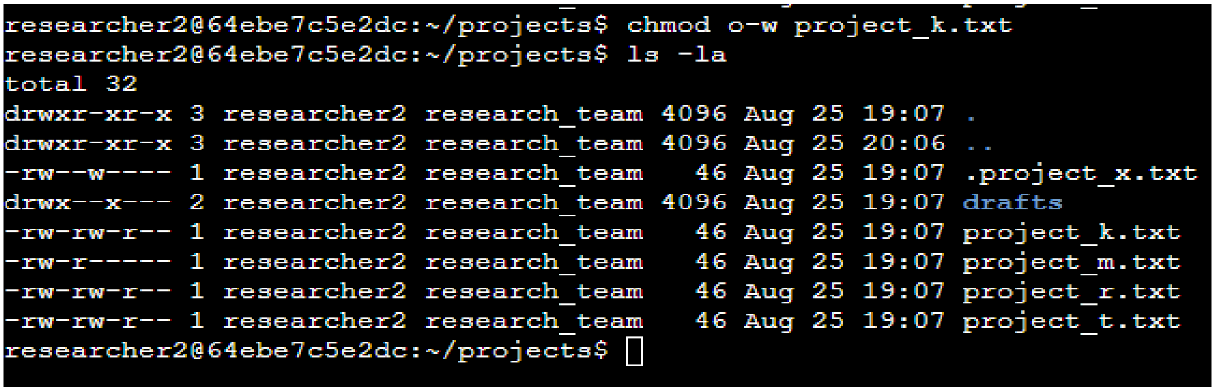
The above screenshot shows that project_k.txt is a file, shown by the hyphen (-) being the first character instead of a d, the next 3 characters show what permissions the owner type user has. The 2nd and 3rd characters show read (r) & write (w) respectively, this indicates that user has read and write permissions for this file, but does not have executable permissions as indicated by the 4th character containing a hyphen (-), instead of executable (x) which would indicate they did have executable permissions also.

The 5th and 6th characters show that owner type group also have read (r) and write (w) permissions because they contain the characters r & w respectively, group does not have executable permissions as the 7th character of the permissions string shows a hyphen (-) instead of an `x`.

Finally, the owner type other also has read (`r`) and write (`w`) permissions. This is indicated by the 8th and 9th characters of the permissions string being `r` & `w` respectively, other does not have executable permissions as shown by a hyphen (-) as the 10th and last character of the permissions string instead of an `x`.

Change file permissions

The organization determined that other should not have write access to any files, to comply with this I referred to the previous file permissions I returned and determined that currently other has write permissions for the file project_k.txt. To remove the permissions from this file, I used the command: `chmod o-w project_k.txt`. This command removes the write permissions for the type other from this file as shown in the screenshot below.

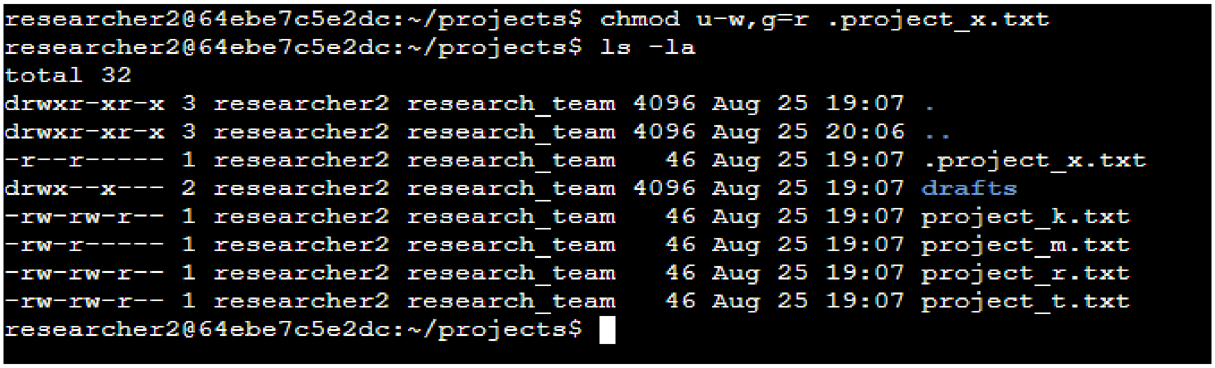


```
researcher2@64ebe7c5e2dc:~/projects$ chmod o-w project_k.txt
researcher2@64ebe7c5e2dc:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Aug 25 19:07 .
drwxr-xr-x 3 researcher2 research_team 4096 Aug 25 20:06 ..
-rw--w---- 1 researcher2 research_team   46 Aug 25 19:07 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Aug 25 19:07 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Aug 25 19:07 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Aug 25 19:07 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Aug 25 19:07 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Aug 25 19:07 project_t.txt
researcher2@64ebe7c5e2dc:~/projects$ 
```

The chmod command changes the permissions on files and directories, the first argument given to the command specifies what permissions should be changed and the second argument specifies the file or directory. After entering the command and viewing the most recent snapshot of the files you can see other no longer has write permissions to any of the files or folders in this directory. This is indicated by the 8th character of each permissions string containing a hyphen (-) instead of an `x`.

Change file permissions on a hidden file

The research team has archived `.project_x.txt`, which is why it's now a hidden file. This file should not have write permissions for anyone, but the user and group should have read access.



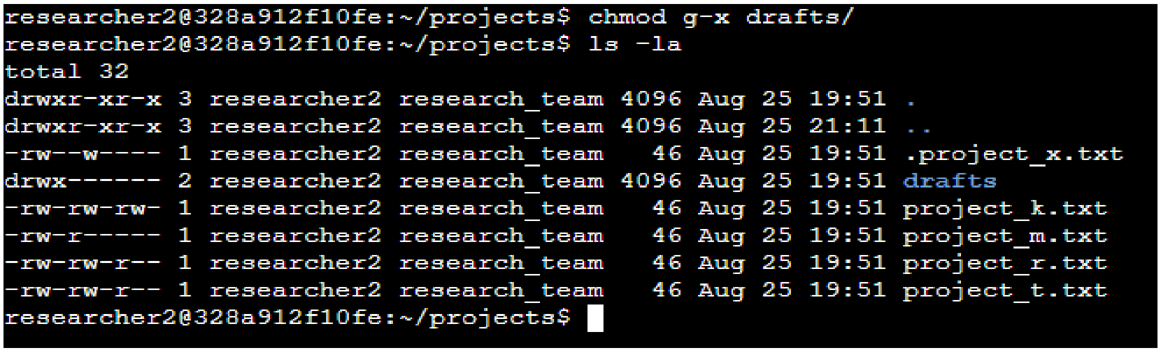
```
researcher2@64ebe7c5e2dc:~/projects$ chmod u-w,g=r .project_x.txt
researcher2@64ebe7c5e2dc:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Aug 25 19:07 .
drwxr-xr-x 3 researcher2 research_team 4096 Aug 25 20:06 ..
-r--r----- 1 researcher2 research_team   46 Aug 25 19:07 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Aug 25 19:07 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Aug 25 19:07 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Aug 25 19:07 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Aug 25 19:07 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Aug 25 19:07 project_t.txt
researcher2@64ebe7c5e2dc:~/projects$ 
```

To accomplish this I used the command: `chmod u-w,g=r .project_x.txt` to assign the correct permissions. In the command, user has its write permissions removed with the `u-w` portion of the command, leaving its read permissions the same. Group has its permissions overridden only allow read permissions, removing its write permissions. This is accomplished with the `g=r` part of the command. Rather than removing its write permissions then assigning the read permissions with 2 commands. The equals (=) operator overwrites its permissions with the specified argument to only allow read permissions, in one simple command. As shown above both user and group only have read permissions on the `.project_x.txt` file shown by the `r` in both the 2nd and 5th characters of the permissions string.

Change directory permissions

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

To accomplish this I used the Linux command: `chmod g-x drafts/` to remove the executable permissions of group from the drafts directory. As seen below, once this command has been executed, entering the command `ls -la` shows that only user has executable permissions for the drafts directory now, this is shown by the `x` in the 4th character of the permissions string with no further `x` characters present in the permissions string. The `researcher2` user already had executable permissions so no action was necessary.



```
researcher2@328a912f10fe:~/projects$ chmod g-x drafts/
researcher2@328a912f10fe:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Aug 25 19:51 .
drwxr-xr-x 3 researcher2 research_team 4096 Aug 25 21:11 ..
-rw--w---- 1 researcher2 research_team   46 Aug 25 19:51 .project_x.txt
drwx------ 2 researcher2 research_team 4096 Aug 25 19:51 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Aug 25 19:51 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Aug 25 19:51 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Aug 25 19:51 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Aug 25 19:51 project_t.txt
researcher2@328a912f10fe:~/projects$
```

Summary

I changed multiple permissions to match the level of authorization my organization wanted for files and directories in the `projects` directory. To accomplish this task I used the command `ls -la` to check the permissions for the directory. I then used the `chmod` command throughout to change the permissions on files and directories.