File permissions in Linux

Project description

I have been tasked at my organization with updating the file permissions for certain files and directories within the `projects` directory. Some of the permissions do not currently have the proper level of authorization, so constantly checking and updating these permissions is crucial in helping keep systems secure. To complete this task, I performed the following steps:

Check file and directory details

I used the `ls -la` command to determine the existing permissions set for a specific directory in the file system, including any hidden files/directories within the directory, as shown here:

```
researcher2@92db2a91eb05:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Oct 25 01:06 .
drwxr-xr-x 3 researcher2 research_team 4096 Oct 25 01:41 ..
-rw--w---- 1 researcher2 research_team   46 Oct 25 01:06 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Oct 25 01:06 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Oct 25 01:06 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Oct 25 01:06 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct 25 01:06 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct 25 01:06 project_t.txt
```

The first line of the screenshots displays the command mentioned above, and the other lines represent the output. In this case, it lists the full content of the `projects` directory including hidden files, thanks to the `-a` flag in `-la`. We can see from this output, that there is one hidden directory called `drafts`, and one hidden file named `.project_x.txt`, and four other project files. The 10-character string on the left represents the file permissions.

Describe the permissions string

The 10-character string can be broken down further into four groups, and they are assigned as follows:

- **1st character:** the first character can either be a hyphen (`-`) or a `d`. If it is a `d` ,it means it is a directory, if it is a hyphen (`-`), it means it is a regular file.
- **2nd-4th characters:** the next 3 characters are assigned for the User who created the file/directory. They indicate the read (`r`), write (`w`), and execute (`x`) permissions for the owner in that order, if a character is a hyphen (`-`) instead, it indicated this permission is not granted to the user.

- **5th-7th characters:** the next 3 characters are assigned to the group from which the user, or owner, assigns the file/directory. They indicate the read (`r`), write (`w`), and execute (`x`) permissions for the owner in that order, if a character is a hyphen (`-`) instead, it indicated this permission is not granted to the group.
- **8th-10th characters:** the next 3 characters are assigned for other, who would be anyone in the system outside of the group and user. They indicate the read (`r`), write (`w`), and execute (`x`) permissions for the owner in that order, if a character is a hyphen (`-`) instead, it indicated this permission is not granted to the other.

Change file permissions

The organization concluded that other should not have any write access whatsoever. After examining the permissions previously referred, I determined that `project_k.txt` needed to have the write access removed.

The following code demonstrates how I accomplished this:

```
researcher2@92db2a91eb05:~/projects$ chmod o-w project_k.txt
researcher2@92db2a91eb05:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Oct 25 01:06 .
drwxr-xr-x 3 researcher2 research_team 4096 Oct 25 01:41 ..
-rw--w---- 1 researcher2 research_team   46 Oct 25 01:06 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Oct 25 01:06 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Oct 25 01:06 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Oct 25 01:06 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct 25 01:06 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct 25 01:06 project_t.txt
```

The first two lines of the screenshot display the commands I entered: the first argument remove the write permissions to other in the `project_k.txt` file; after this, I used `ls -la` to verify that the changes were properly made.

Change file permissions on a hidden file

The research team at my organization recently archived `.project_x.txt`. They emphasized they did not want anyone to have write access to this project; however, the user and group should have read access.

To accomplish this, I sued the following Linux command to change the permissions:

```
researcher2@92db2a91eb05:~/projects$ chmod ug-w,g+r .project_x.txt
researcher2@92db2a91eb05:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Oct 25 01:06 .
drwxr-xr-x 3 researcher2 research_team 4096 Oct 25 01:41 ..
-r--r----- 1 researcher2 research_team   46 Oct 25 01:06 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Oct 25 01:06 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Oct 25 01:06 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Oct 25 01:06 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct 25 01:06 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct 25 01:06 project_t.txt
```

The first two lines show the commands I entered, and the remaining lines show the output from the commands. I know `.project_x.txt`. is a hidden file as it starts with a period. In this example, I removed the user and group write permissions with `ug-w,` and added read permissions the the group with `g+r.` Then I verified the changes with `ls -la.`

Change directory permissions

I have been informed that only `researcher2` should have access to the `drafts` directory and its contents. We can see that `researcher2` is the creator of the folder, which means that the organization wants only the User to have full access to the directory.

To follow through with this, I employed the following commands:

```
researcher2@92db2a91eb05:~/projects$ chmod g-x drafts
researcher2@92db2a91eb05:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Oct 25 01:06 .
drwxr-xr-x 3 researcher2 research_team 4096 Oct 25 01:41 ..
-r--r----- 1 researcher2 research_team   46 Oct 25 01:06 .project_x.txt
drwx------ 2 researcher2 research_team 4096 Oct 25 01:06 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Oct 25 01:06 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Oct 25 01:06 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct 25 01:06 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct 25 01:06 project_t.txt
```

The first two lines show the commands I entered, and the remaining lines show the output from the commands. I previously determined that the group had execute permissions on this folder, so I employed the `chmod` command to remove them. The `researcher2` user already had execute permissions, so there was no need to change that.

Summary

I changed multiple permissions to match the level of authorization that my organization wanted for files and directories in the `projects` directory. In essence, the first step was using `ls -la` to check the permissions of files/folders for the directory. I then used the `chmod` command multiple times to change the permissions on files and directories.