File Permissions in Linux

Project description

Research team needs to update file permissions for certain files and directories within projects directory. However, after examining existing permissions on the file system, it was found that current permissions do not currently reflect the level of authorization that should be given. Checking and updating these permissions will ensure their system is secure. To change these file permissions, the following command lines were run in Linux:

Check file and directory details

The following code demonstrates how I used Linux commands to resolve these issues.

```
researcher2@37d539313370:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Aug  5 01:03 .
drwxr-xr-x 3 researcher2 research_team 4096 Aug  5 02:14 ..
-rw--w---- 1 researcher2 research_team   46 Aug  5 01:03 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Aug  5 01:03 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Aug  5 01:03 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Aug  5 01:03 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Aug  5 01:03 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Aug  5 01:03 project_t.txt
researcher2@37d539313370:~/projects$
```

The first line denoted with `ls -la` (ls command with -la option) allows all content of the projects directory to be seen, including hidden files. The output of my command indicates one directory named `drafts`, a hidden file named `.project_x.tx`, and five other project files, where the first 10-character string in first column represents permissions set on each file or directory for the user, group, and other users, respectively.

Describe the permissions string

The 10-character string can be broken down to determine who is authorized to access the file and their specific permissions. The 1st character is denoted by either `d` or `-` and indicates file type. If `d` , it's a directory. If it's a `-` hyphen , it's a regular file. Characters 2-4 indicate read `r`, write `w`, and execute `x` permissions for the user. If there is a hyphen `-`, it indicates that permission is not granted for the group. Following a similar pattern, Characters 5-7 indicate read `r`, write `w`, execute `x`, or no permission `-` for group, and characters 8-10 indicate read `r`, write `w`, execute `x`, or no permission `-` for other.

Change file permissions

As the organization has asked that other should not have any access to any files, therefore, `project_k.txt` must have write access removed for other. The following Linux commands performs this:

```
researcher2@37d539313370:~/projects$ chmod o-w project_k.txt
researcher2@37d539313370:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Aug  5 01:03 .
drwxr-xr-x 3 researcher2 research_team 4096 Aug  5 02:14 ..
-rw--w---- 1 researcher2 research_team   46 Aug  5 01:03 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Aug  5 01:03 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Aug  5 01:03 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Aug  5 01:03 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Aug  5 01:03 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Aug  5 01:03 project_t.txt
researcher2@37d539313370:~/projects$
```

The command line `chmod o-w project_k.txt` removes the write permissions from other for the `project_k.txt` file. `ls -la` was used to review updates in permissions.

Change file permissions on a hidden file

```
researcher2@37d539313370:~/projects$ chmod u-w,g+r,g-w .project_x.txt
researcher2@37d539313370:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Aug  5 01:03 .
drwxr-xr-x 3 researcher2 research_team 4096 Aug  5 02:14 ..
-r--r----- 1 researcher2 research_team   46 Aug  5 01:03 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Aug  5 01:03 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Aug  5 01:03 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Aug  5 01:03 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Aug  5 01:03 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Aug  5 01:03 project_t.txt
researcher2@37d539313370:~/projects$
```

The screenshot above shows commands that were entered to ensure that the hidden file `.project_x.txt` was only able to be read by both user and group, removing write permissions from user `u-w`, adding read permission for group `g+r` , and removing write permission for group `g-r`.

Change directory permissions

Finally, my organization wants `researcher2` user to have access to `drafts` directory and its contents, meaning no one other than `researcher2` should have execute permissions.

```
researcher2@37d539313370:~/projects$ chmod g-x drafts
researcher2@37d539313370:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Aug  5 01:03 .
drwxr-xr-x 3 researcher2 research_team 4096 Aug  5 02:14 ..
-r--r----- 1 researcher2 research_team   46 Aug  5 01:03 .project_x.txt
drwx------ 2 researcher2 research_team 4096 Aug  5 01:03 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Aug  5 01:03 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Aug  5 01:03 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Aug  5 01:03 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Aug  5 01:03 project_t.txt
researcher2@37d539313370:~/projects$
```

The `chmod g-x drafts` command resolves the previous permission issues by removing execute permissions from the group from the file `drafts`.

Summary

I had successfully changed multiple permissions to match the level of authorization my organization had requested, using the command `ls -la` to check permissions for each directory change after using the `chmod` command which changed the permissions on files and directories.