File permissions in Linux

Project description

The research team at my organization needs to update file permissions for specific files and directories. Linux commands allow you to update permissions for file access based on the principle of least privilege. The intent of this project is to use Linux to check current permissions and change permissions as required. Below are the steps taken to fulfill this objective.

Check file and directory details

The first command used provides a list of permissions for each non-hidden file in a directory. See below:

```
ls -l
```

The output from this command generates:

```
drwx--x--- 2 researcher2 research_team 4096 Nov 15 01:48 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Nov 15 01:48 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Nov 15 01:48 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Nov 15 01:48 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Nov 15 01:48 project_t.txt
```

The ls command with -l option displays the permission strings for all files and directories within the relevant `projects` directory. This command shows there is one directory and four files within this directory.

Describe the permissions string

Permission strings contain 10-digits. The 1st digit refers to the file type. “d” is for directory whereas “-” indicates this is a file permission. The following 9 digits can be described as three sets of 3 digit strings. Each set is for a specific owner type: the first is for user, the second is for group, and the third is for other. Look at the example `project_k.txt` (above) with a permission string of `-rw-rw-rw-`. For this file, the first (-) designates this as a file versus a directory. The next three digits correspond with the user permissions, which are read and write. Execution permissions are not granted, as indicated by the (-). The next three digits correspond to the group permissions. These are identical to the user’s. The third group corresponds to the other user permissions, which are also set the same in this example.

Change file permissions

The organization doesn't permit "other" to have write access to any files. The only file with write access for "other" is `project_k.txt`. Using the below command, one can edit the permissions for this file.

```
chmod o-w project_k.txt
```

The "chmod" stands for change mode. The first argument indicates what permissions should be changed. In this case, "o-w" states to change the permissions of the other group (o) by removing (-) write permissions (w). This can likewise be done for user (u) and group (g). However, in this example, this was not necessary. The second argument is the file or directory whose permissions are changing. The output from this command input is below. This verifies that no files are set to have other with write access.

```
drwx--x--- 2 researcher2 research_team 4096 Nov 15 01:48 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Nov 15 01:48 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Nov 15 01:48 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Nov 15 01:48 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Nov 15 01:48 project_t.txt
```

Change file permissions on a hidden file

I wanted to check if there were hidden files to manage. To do so, I included the `-la` option to the `ls` command as below:

```
ls -la
```

This included hidden files in the permissions look up. As seen below there are two hidden directories and one hidden file.

```
drwxr-xr-x 3 researcher2 research_team 4096 Nov 15 01:48 .
drwxr-xr-x 3 researcher2 research_team 4096 Nov 15 02:10 ..
-rw--w---- 1 researcher2 research_team   46 Nov 15 01:48 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Nov 15 01:48 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Nov 15 01:48 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Nov 15 01:48 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Nov 15 01:48 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Nov 15 01:48 project_t.txt
```

The research team in my organization recently archived `project_x.txt`. They do not want anyone to have write access to this project but the "user" and "group" should have read access. To address this, I used the below command:

```
chmod u=r,g=r .project_x.txt
```

This set "user" and "group" permissions to read only. As "other" had no permissions, I did not update this owner type. The result of this change is shown below:

```
drwxr-xr-x 3 researcher2 research_team 4096 Nov 15 01:48 .
drwxr-xr-x 3 researcher2 research_team 4096 Nov 15 02:10 ..
-r--r----- 1 researcher2 research_team   46 Nov 15 01:48 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Nov 15 01:48 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Nov 15 01:48 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Nov 15 01:48 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Nov 15 01:48 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Nov 15 01:48 project_t.txt
```

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. As seen above, group users have execute permissions. To address this, I used the below command.

```
chmod g=- drafts
```

This set group permissions to nothing as indicated by the (-). The result is shown below.

```
drwxr-xr-x 3 researcher2 research_team 4096 Nov 15 01:48 .
drwxr-xr-x 3 researcher2 research_team 4096 Nov 15 02:10 ..
-r--r----- 1 researcher2 research_team   46 Nov 15 01:48 .project_x.txt
drwx------ 2 researcher2 research_team 4096 Nov 15 01:48 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Nov 15 01:48 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Nov 15 01:48 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Nov 15 01:48 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Nov 15 01:48 project_t.txt
```

Summary

I changed multiple permissions to match the permissions required by my organization. The first step in this was using the `ls` command with correct options to look up permissions on files in the relevant directory. I then used the chmod command to update permissions on files and directories to match the organization's authorization policies.