

File permissions in Linux

Project description

In the Linux command line we can verify, and if necessary, make changes to the access permissions and use of information as required by the organization. Linux recognizes three types of "owners" of a file: the user, the group to which it belongs, and the rest. Below is a report detailing an activity related to this function.

Check file and directory details

The command "ls -l" displays the current permissions of files and directories.

```
researcher2@62cb60bc8b39:~/projects$ ls -l
total 20
drwx--x--- 2 researcher2 research_team 4096 Dec 16 02:58 drafts
-rw-rw-rw- 1 researcher2 research_team  46 Dec 16 02:58 project_k.txt
-rw-r----- 1 researcher2 research_team  46 Dec 16 02:58 project_m.txt
-rw-rw-r-- 1 researcher2 research_team  46 Dec 16 02:58 project_r.txt
-rw-rw-r-- 1 researcher2 research_team  46 Dec 16 02:58 project_t.txt
```

Describe the permissions string

We will look at the permissions of "**project_r.txt**" to describe the permission string at the very left of the output shown in the screenshot above. The permission string consists of 10 characters. The first character indicates whether we are looking at a directory (**d**) or a file (**-**). The following 9 characters are split into three groups of three characters each. The first three characters show the permissions for the **user**, the next three characters show the permissions for the **group** the owner belongs to, and the last three characters show the permissions of **other** people. The letter "**r**" indicates the owner or group has **read** permission, the letter "**w**" indicates **write** permission and the letter "**x**" indicates **execute** permission. If there is a "**-**" instead of a letter, that means **lack of permission**.

In "**project_r.txt**", all owners have read permission. User and group have write permission, but not other people outside of the group. Lastly, no owner has execute permission which is consistent with the fact that this particular file is not executable.

Change file permissions

The command “**chmod**” is used to change the permissions of a file or directory. Since the organization doesn’t allow other to have write permissions, one change had to be made:

```
researcher2@62cb60bc8b39:~/projects$ chmod o-w project_k.txt
```

Change file permissions on a hidden file

The option “-a” displays hidden files of a directory. If we combine this option with “-l”, we end up with “-la” which as expected displays the permissions of all files and subdirectories, including hidden files. In Linux, hidden files start with a “.”

Knowing this, we use this command to explore the permissions of an archived file, and adjust its permissions according to the organization’s standard of removing write permissions, but allowing the user and group to read said file.

```
researcher2@62cb60bc8b39:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Dec 16 02:58 .
drwxr-xr-x 3 researcher2 research_team 4096 Dec 16 03:41 ..
-rw--w---- 1 researcher2 research_team   46 Dec 16 02:58 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Dec 16 02:58 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Dec 16 02:58 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Dec 16 02:58 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Dec 16 02:58 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Dec 16 02:58 project_t.txt
```

The file “.project_x.txt” needs to have the write permissions removed for user and group. Also, a read permission needs to be added for the group:

```
researcher2@62cb60bc8b39:~/projects$ chmod u-w,g+r-w .project_x.txt
```

And now we confirm our changes:

```
researcher2@62cb60bc8b39:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Dec 16 02:58 .
drwxr-xr-x 3 researcher2 research_team 4096 Dec 16 03:41 ..
-r--r----- 1 researcher2 research_team  46 Dec 16 02:58 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Dec 16 02:58 drafts
-rw-rw-r-- 1 researcher2 research_team  46 Dec 16 02:58 project_k.txt
-rw-r----- 1 researcher2 research_team  46 Dec 16 02:58 project_m.txt
-rw-rw-r-- 1 researcher2 research_team  46 Dec 16 02:58 project_r.txt
-rw-rw-r-- 1 researcher2 research_team  46 Dec 16 02:58 project_t.txt
```

Change directory permissions

The command “chmod” can also be used to change permissions on directories. The syntax is the same, with the exception of the last argument of the command, this being a directory name, without an extension (since it’s not a file).

The “drafts” directory should only be accessible to the user, so we make the change:

```
researcher2@62cb60bc8b39:~/projects$ chmod g-x drafts
```

And we confirm:

```
researcher2@62cb60bc8b39:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Dec 16 02:58 .
drwxr-xr-x 3 researcher2 research_team 4096 Dec 16 03:41 ..
-r--r----- 1 researcher2 research_team  46 Dec 16 02:58 .project_x.txt
drwx----- 2 researcher2 research_team 4096 Dec 16 02:58 drafts
-rw-rw-r-- 1 researcher2 research_team  46 Dec 16 02:58 project_k.txt
-rw-r----- 1 researcher2 research_team  46 Dec 16 02:58 project_m.txt
-rw-rw-r-- 1 researcher2 research_team  46 Dec 16 02:58 project_r.txt
-rw-rw-r-- 1 researcher2 research_team  46 Dec 16 02:58 project_t.txt
```

Summary

The task of verifying the permissions that files and directories should have according to the organization's security specifications and standards was completed, and the pertinent changes were made to reflect those policies in practice.