File permissions in Linux

Project description

Using Linux commands to configure file permissions for specific files and directories on a virtual machine. Checking and updating permissions will ensure the security of systems.

Check file and directory details

Using the following command `ls -la` in the `projects` directory reveals existing permissions set for each file and directory.

```
researcher2@5c6fc775fe3c:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Oct 30 20:58 .
drwxr-xr-x 3 researcher2 research_team 4096 Oct 30 21:23 ..
-rw--w---- 1 researcher2 research_team   46 Oct 30 20:58 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Oct 30 20:58 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Oct 30 20:58 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Oct 30 20:58 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct 30 20:58 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct 30 20:58 project_t.txt
researcher2@5c6fc775fe3c:~/projects$
```

Describe the permissions string

The first column shown as output in the screenshot represents the permissions on the file or directory. This can be broken down to reveal who can access the content. The character representations are as follows:

- 1st character: The first character in this sequence may be either a "`d`" or a hyphen (`-`) and serves as an indicator of the file's type. When it's a "`d`," the file is a directory, and if it's a hyphen (`-`), it designates a regular file.

- 2nd-4th characters: These characters represent the permissions for the user and are used to signify "read (`r`)," "write (`w`)," and "execute (`x`)." In the absence of any of these characters, replaced by a hyphen (`-`), it signifies the absence of that specific permission for the user.

- 5th-7th characters: These characters represent the permissions for the group, denoting "read (r)," "write (w)," and "execute (x)." A hyphen (-) in any of these positions indicates the lack of that specific permission for the group.

- 8th-10th characters: These characters illustrate the permissions for others, encompassing all users on the system except for the owner and the group. They signify "read (r)," "write (w)," and "execute (x)." If any of these characters is replaced by a hyphen (-), it indicates the absence of that particular permission for others.

Change file permissions

It is determined that others shouldn't have write access to any files. The following code in Linux shows how I addressed the request:

```
researcher2@5d738f0f927b:~/projects$ chmod o-w project_k.txt
researcher2@5d738f0f927b:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Dec  2 15:27 .
drwxr-xr-x 3 researcher2 research_team 4096 Dec  2 15:27 ..
-rw--w---- 1 researcher2 research_team   46 Dec  2 15:27 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Dec  2 15:27 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Dec  2 15:27 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Dec  2 15:27 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Dec  2 15:27 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Dec  2 15:27 project_t.txt
researcher2@5d738f0f927b:~/projects$
```

I used the `chmod` command to change the permissions of the file `project_k.txt`. `chmod o-w` removed write permissions from other.

Change file permissions on a hidden file

The task is to remove all write access permissions to the hidden file `.project_x.txt`.

```
researcher2@3213bbc1d047:~/projects$ chmod u-w,g-w,g+r .project_x.txt
researcher2@3213bbc1d047:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Dec 20 15:36 .
drwxr-xr-x 3 researcher2 research_team 4096 Dec 20 15:36 ..
-r--r----- 1 researcher2 research_team   46 Dec 20 15:36 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Dec 20 15:36 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Dec 20 15:36 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Dec 20 15:36 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Dec 20 15:36 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Dec 20 15:36 project_t.txt
researcher2@3213bbc1d047:~/projects$
```

Using `chmod u-w, g-w, g+r` on the hidden file first removes the write permissions to user and group and adds read permission to the group.

Change directory permissions

The task is to allow `researcher2` user to gain access permissions to the drafts `directory`.

```
researcher2@5d738f0f927b:~/projects$ chmod g-x drafts
researcher2@5d738f0f927b:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Dec  2 15:27 .
drwxr-xr-x 3 researcher2 research_team 4096 Dec  2 15:27 ..
-r--r----- 1 researcher2 research_team   46 Dec  2 15:27 .project_x.txt
drwx------ 2 researcher2 research_team 4096 Dec  2 15:27 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Dec  2 15:27 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Dec  2 15:27 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Dec  2 15:27 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Dec  2 15:27 project_t.txt
researcher2@5d738f0f927b:~/projects$
```

The initial group permissions allowed execution of directory and its contents. Using `chmod g-x drafts` resulted in removing execute permissions for the group. As `researcher2` user already has execute permissions, further action is not required.

Summary

I adjusted several permissions to align them with the desired authorization level established for the files and directories within the `projects` directory. The initial phase involved utilizing the "`ls -la`" command to assess the existing permissions for the directory, which provided the basis for my subsequent actions. Subsequently, I employed the "chmod" command multiple times to modify the permissions for both files and directories as needed.