File permissions in Linux

Project description

The research team at my organization needs to revise the file permissions for specific files and directories within the projects folder. The current permissions do not align with the required authorization levels. Reviewing and adjusting these permissions will enhance the security of their system. To accomplish this, I follow multiple steps.

Check file and directory details

The code below shows how to determine the permission for directory in the file system. The command ls - la:

```
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-rw- 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
```

The first line of the screenshot shows the command I executed, while the subsequent lines display the resulting output. The command lists all items within the projects directory. I used the ls command with the -la option to generate a detailed listing of the files, including hidden ones. The output reveals a directory named "drafts," a hidden file called .project_x.txt, and five additional project files. The 10-character string in the first column represents the permissions assigned to each file or directory.

Describe the permissions string

Basically the 10-character string can be broken down to determine who has access to the file and what permissions they have. The characters and their meanings are as follows:

1st character: This character is either a "d" or a hyphen (-) and represents the file type. If it's a "d," the item is a directory; if it's a hyphen (-), it's a regular file.

2nd-4th characters: These characters represent the read (r), write (w), and execute (x) permissions for the user. If a character is a hyphen (-), it means that particular permission is not granted to the user.
5th-7th characters: These characters show the read (r), write (w), and execute (x) permissions for the group. A hyphen (-) in place of any of these characters means that the permission is not granted to the group.
8th-10th characters: These characters represent the read (r), write (w), and execute (x) permissions for others, meaning all users on the system who are neither the user nor the group. A hyphen (-) indicates that the permission is not granted to others.

For example, the file permissions for project_t.txt are -rw-rw-r--. Since the first character is a hyphen (-), it indicates that project_t.txt is a file, not a directory. The second, fifth, and eighth characters are "r," meaning the user, group, and others all have read permissions. The third and sixth characters are "w," showing that only the user and group have write permissions. No one has executed permissions for project_t.txt.

drwxrwxrwx

Change file permissions

The organization decided that others should not have write access to any of their files. To adhere to this policy, I referred to the file permissions I had previously retrieved. I concluded that write access for "other" needed to be removed from project_k.txt.

The first two lines of the screenshot show the commands I entered, while the remaining lines display the output from the second command. The chmod command modifies file and directory permissions. The first argument specifies which permissions should be altered, and the second argument indicates the file or directory. In this case, I removed write permissions for "other" from the project_k.txt file. Afterward, I used ls -la to verify the changes I had made.

```
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
```

Change file permissions on a hidden file

The research team at my organization recently archived project_x.txt and wants to ensure that no one has write access to it, but both the user and group should retain read access.

The first two lines of the screenshot show the commands I entered, while the remaining lines display the output from the second command. I identified .project_x.txt as a hidden file because it begins with a period (.). In this case, I removed write permissions for the user and group while ensuring the group had read access. I removed write permissions from the user using u-w, then removed write permissions from the group with g-w and added read permissions to the group with g+r.

```
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
```

Change directory permissions

My organization requires that only the user "researcher2" has access to the drafts directory and its contents, meaning no one else should have execute permissions.

The output here shows the permission listing for several files and directories. Line 1 represents the current directory (projects), and line 2 represents the parent directory (home). Line 3 shows a regular file named .project_x.txt. Line 4 displays the directory (drafts) with restricted permissions. As seen, only the user "researcher2" has execute permissions. Since the group previously had execute permissions, I used the chmod command to remove them. The user "researcher2" already had execute permissions, so there was no need to add them.

```
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
```

Summary

In summary of the specific laboratory task. I was able to change multiple permissions based on user, group or other. The command ls -la is to check the permission and hidden file. Chmod command is to change permissions on files and directories