File permissions in Linux

Project description

The research team at my organization needed to update the file and directory permissions within the **projects** directory, as the existing permissions did not accurately reflect the appropriate levels of access. To ensure system security and proper authorization, I reviewed and updated the necessary permissions.

Check file and directory details

The following code demonstrates how I used Linux commands to determine the existing permissions set for a specific directory in the file system.

```
researcher2@8a1444db59f7:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Jun 25 08:21 .
drwxr-xr-x 3 researcher2 research_team 4096 Jun 25 09:28 ..
-rw--w---- 1 researcher2 research_team   46 Jun 25 08:21 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Jun 25 08:21 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Jun 25 08:21 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Jun 25 08:21 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jun 25 08:21 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jun 25 08:21 project_t.txt
researcher2@8a1444db59f7:~/projects$
```

The first line of the screenshot shows the command I entered, while the remaining lines display its output. I used the `ls -la` command to list all contents of the **projects** directory in detail, including hidden files. The output indicates that the directory contains one subdirectory named `drafts`, one hidden file called `.project_x.txt`, and five additional project files. The 10-character string in the first column of the output represents the permissions assigned to each file or directory.

Describe the permissions string

The 10-character string shown at the beginning of each line in the `ls -la` output represents the file type and the permission settings for different user categories. These characters break down as follows:

- **1st character:** Indicates the file type. A `d` means it is a directory, while a hyphen (`-`) signifies a regular file.

- **2nd to 4th characters:** Represent the **user's** permissions — specifically read (`r`), write (`w`), and execute (`x`). A hyphen (`-`) in place of a letter means that particular permission is not granted to the user.

- **5th to 7th characters:** Indicate the **group's** permissions, following the same format of `r`, `w`, and `x`, or hyphens where permissions are not given.

- **8th to 10th characters:** Define the permissions for **others** (all users who are neither the owner nor part of the group). Again, these are shown using `r`, `w`, and `x`, with hyphens indicating the absence of permission.

Example:
For the file `project_t.txt`, the permissions string is `-rw-rw-r--`.

- The first character is a hyphen (`-`), which means it is a regular file.

- The next three characters `rw-` show that the **user** has read and write permissions.

- The next set `rw-` indicates the **group** also has read and write permissions.

- The final set `r--` shows that **others** have read-only access.

- No user has execute (`x`) permission in this case.

This structured permission system helps control file access and maintain system security.

Change file permissions

The organization decided that no files should grant write access to the **other** user category. To comply with this policy, I reviewed the previously returned file permissions and identified that `project_k.txt` had write access enabled for **others**. To correct this, I used the following Linux command to remove that permission: `chmod o-w project_k.txt`

The following code demonstrates how I used Linux commands to do this:

```
researcher2@8a1444db59f7:~/projects$ chmod o-w project_k.txt
researcher2@8a1444db59f7:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Jun 25 08:21 .
drwxr-xr-x 3 researcher2 research_team 4096 Jun 25 09:28 ..
-rw--w---- 1 researcher2 research_team   46 Jun 25 08:21 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Jun 25 08:21 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Jun 25 08:21 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Jun 25 08:21 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jun 25 08:21 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jun 25 08:21 project_t.txt
researcher2@8a1444db59f7:~/projects$
```

The first two lines of the screenshot show the commands I entered, while the remaining lines display the output of the second command. The `chmod` command is used to modify permissions on files and directories. The **first argument** specifies the permission changes to be made, and the **second argument** indicates the file or directory to which the changes apply. In this example, I removed write permissions to ensure compliance with the organization's access policy.

Change file permissions on a hidden file

The research team at 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.

The following code demonstrates how I used Linux commands to change the permissions:

```
researcher2@8a1444db59f7:~/projects$ chmod u-w,g+r,g-w .project_x.txt
researcher2@8a1444db59f7:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Jun 25 08:21 .
drwxr-xr-x 3 researcher2 research_team 4096 Jun 25 09:28 ..
-r--r----- 1 researcher2 research_team   46 Jun 25 08:21 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Jun 25 08:21 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Jun 25 08:21 project_k.txt
-rw------- 1 researcher2 research_team   46 Jun 25 08:21 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jun 25 08:21 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jun 25 08:21 project_t.txt
researcher2@8a1444db59f7:~/projects$
```

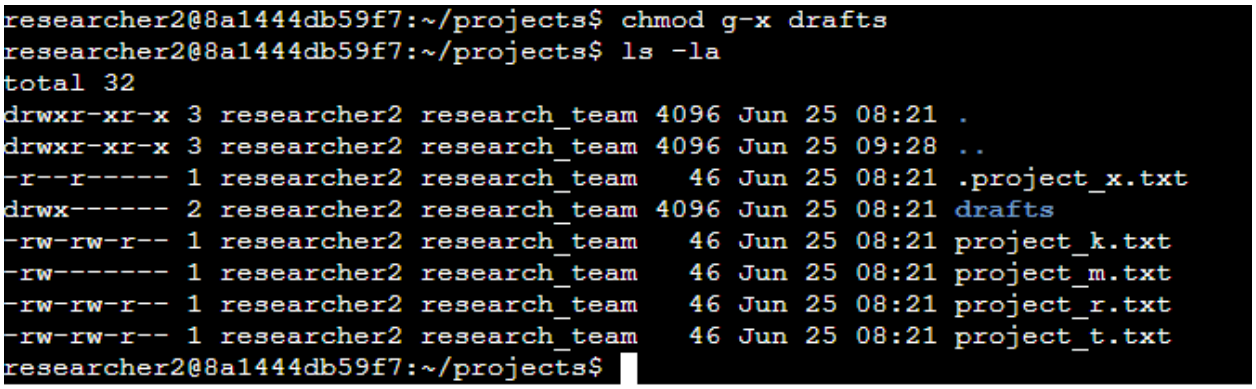
The first two lines of the screenshot show the commands I entered, while the remaining lines display the output of the second command. I identified `.project_x.txt` as a hidden file because its name begins with a period (.). In this example, I modified the file's permissions to enhance security. Specifically, I removed write permissions from both the user and the group, and added read permissions to the group. I achieved this using the following commands:

- `chmod u-w .project_x.txt` — removes write permission from the user
- `chmod g-w .project_x.txt` — removes write permission from the group
- `chmod g+r .project_x.txt` — adds read permission to the group

These adjustments ensure that the file remains readable to the group while preventing unauthorized modifications.

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.
The following code demonstrates how I used Linux commands to change the permissions:
`chmod 700 drafts` or `chmod g-x drafts`



The output shown displays the permission listings for several files and directories.

- **Line 1** represents the current directory (`projects`).
- **Line 2** shows the parent directory (`home`).
- **Line 3** lists a regular file named `.project_x.txt`.
- **Line 4** displays the `drafts` directory, which has restricted permissions.

As shown, only the `researcher2` user has execute permissions on the `drafts` directory. Initially, it was determined that the group also had execute permissions, which was not in line with the access policy. To correct this, I used the `chmod` command to remove execute permissions from the group. Since `researcher2` already had the necessary execute permissions, no additional changes were needed for the user.

Summary

I updated multiple file and directory permissions within the **projects** directory to align with the level of authorization required by my organization. The first step in this process was running the `ls -la` command to review the existing permission settings. This initial assessment guided my decisions in the following steps. Based on the output, I used the `chmod` command multiple times to modify permissions on specific files and directories as needed.