

Access Control & File Permissions in Linux

Project description

The research team at my organization needed to update file permissions for specific files and directories in the `projects` directory. The current permissions didn't match the intended authorization levels, posing a security risk. To address this, I performed the following tasks:

Check file and directory details

The following command was used to check the existing permissions in the `projects` directory:

```
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
researcher2@5d738f0f927b:~/projects$
```

The first line of the screenshot shows the command I entered, and the subsequent lines display the output. The command lists all contents of the `projects` directory, including hidden files. The output revealed one directory, `drafts`, one hidden file, `.project_x.txt`, and five other project files. The 10-character string in the first column indicates the permissions set for each file or directory.

Describe the permissions string

The 10-character string defines file type and access permissions. The characters and what they represent are as follows:

- **1st character:** Indicates file type, "d" for directory, "-" for a regular file.
- **2nd-4th characters:** Show read (r), write (w), and execute (x) permissions for the user. A hyphen (-) means the permission is not granted.
- **5th-7th characters:** Represent the same permissions for the group.

- **8th-10th characters:** Represent permissions for others, meaning all users besides the owner and group.

For example, the file permissions for `project_t.txt` are `-rw-rw-r--`.

The first character (-) shows it's a regular file. The r in the user, group, and other sections gives all read access. The w in the user and group sections grants them write access, while no one has execute permissions.

Change file permissions

The organization decided that “others” should not have write access to any files. Based on my earlier review, `project_k.txt` needed its write permission for “others” removed.

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

```
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$
```

The first two lines of the screenshot display the commands I entered, and the other lines display the output of the second command. The `chmod` command changes the permissions on files and directories. The first command removes write permissions for “other”, and the second verifies the change. In this example, I removed write permissions from “other” for the `project_k.txt` file. After this, I used `ls -la` to review the updates I made.

Change file permissions on a hidden file

The team recently archived `project_x.txt` and wanted no one to have write access, while retaining read access for the user and group.

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

```
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$
```

The first two lines of the screenshot display the commands I entered, and the other lines display the output of the second command. Since the filename starts with a period, it's hidden. The command `chmod u-w, g-w, g+r .project_x.txt` removes write access from the user and group and adds read access to the group.

Change directory permissions

Only the researcher2 user should have access to the drafts directory and its contents, meaning only they should retain execute permissions.

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

```
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 output confirms that only researcher2 has execute permissions, as group execute access was removed.

Summary

I modified file and directory permissions to align with organizational access requirements.

Using `ls -la`, I identified the existing permissions, and I then applied `chmod` to adjust them accordingly. These changes strengthened system security by ensuring proper authorization levels for each user type.