William Castro

File permissions in Linux

Project description

The research team at my organization needs to update the file permissions for certain files and directories within the `projects` directory. The permissions do not currently reflect the level of authorization that should be given. Checking and updating these permissions will help keep their system secure. To complete this task, I performed the following tasks:

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@f7c712596e7d:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Jun 11 18:54 .
drwxr-xr-x 3 researcher2 research_team 4096 Jun 11 19:28 ..
-rw--w---- 1 researcher2 research_team   46 Jun 11 18:54 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Jun 11 18:54 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Jun 11 18:54 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Jun 11 18:54 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jun 11 18:54 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jun 11 18:54 project_t.txt
researcher2@f7c712596e7d:~/projects$
```

The code shows everything that is inside the projects folder. I used a special command called “`ls` “with an option called “`-la` “to get a detailed list of all the files, including the ones that are hidden. The result of my command shows that there is a folder called "`drafts`," a hidden file called “. `project_x.txt,` “and five other project files. The string of 10 characters in the first column shows the permissions assigned to each file or folder. The permissions string consists of ten characters, organized into three groups.

Describe the permissions string

The first character in the string represents the type of file or directory. If it is a regular file, it is represented by a hyphen ("-"). If it is a directory, it is represented by the letter "d".

The next three characters represent the permissions for the owner of the file or directory. The first character indicates whether the owner has the permission to read the file ("r"), the second character

indicates the permission to write to the file ("w"), and the third character indicates the permission to execute the file ("x").

The following three characters represent the permissions for the group associated with the file or directory. Similarly, the first character represents the read permission, the second character represents the write permission, and the third character represents the execute permission for the group.

The final three characters represent the permissions for other users who are neither the owner nor a part of the group. Again, the first character represents the read permission, the second character represents the write permission, and the third character represents the execute permission for others.

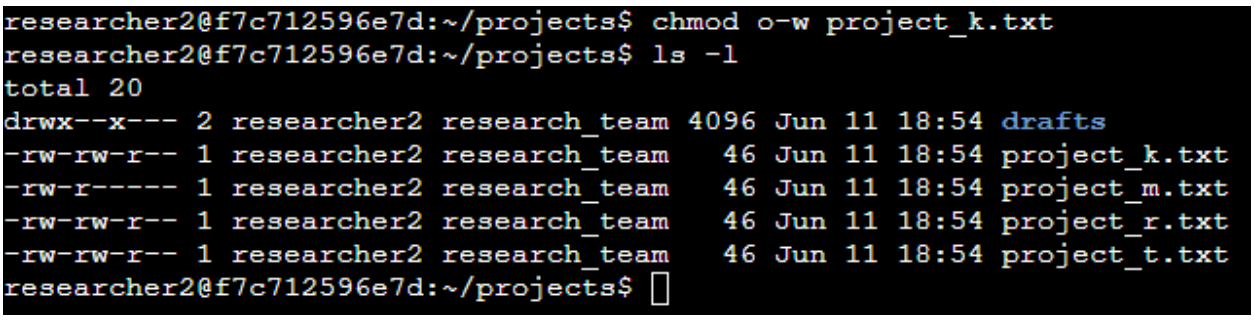
If a specific permission is granted, it is represented by the corresponding character (e.g., "r" for read, "w" for write, "x" for execute). If the permission is not granted, it is represented by a hyphen ("-").

By interpreting the permissions string, users can determine what actions are allowed on a file or directory for the owner, the group, and others in the system.

Change file permissions

The organization determined that others shouldn't have write access to any of their files. To comply with this, I referred to the file permissions that I previously returned. I determined `project_k.txt` must have the write access removed for other.

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



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 argument indicates what permissions should be changed, and the second argument specifies the file or directory. 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.

William Castro

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@f7c712596e7d:~/projects$ chmod u-w,g-w,g+r .project_x.txt
researcher2@f7c712596e7d:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Jun 11 18:54 .
drwxr-xr-x 3 researcher2 research_team 4096 Jun 11 19:28 ..
-r--r----- 1 researcher2 research_team   46 Jun 11 18:54 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Jun 11 18:54 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Jun 11 18:54 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Jun 11 18:54 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jun 11 18:54 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jun 11 18:54 project_t.txt
researcher2@f7c712596e7d:~/projects$
```

The first two lines of the screenshot display the commands I entered, and the other lines display the output of the second command. I know `.project_x.txt` is a hidden file because it starts with a period (`.`). In this example, I removed write permissions from the user and group, and added read permissions to the group. I removed write permissions from the user with `u-w`. Then, I removed write permissions from the group with `g-w`, and added read permissions to the group with `g+r`.

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:

```
researcher2@f7c712596e7d:~/projects$ chmod g-x drafts
researcher2@f7c712596e7d:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Jun 11 18:54 .
drwxr-xr-x 3 researcher2 research_team 4096 Jun 11 19:28 ..
-r--r----- 1 researcher2 research_team   46 Jun 11 18:54 .project_x.txt
drwx------ 2 researcher2 research_team 4096 Jun 11 18:54 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Jun 11 18:54 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Jun 11 18:54 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jun 11 18:54 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Jun 11 18:54 project_t.txt
researcher2@f7c712596e7d:~/projects$
```

The first two lines of the screenshot display the commands I entered, and the other lines display the output of the second command. I previously determined that the group had execute permissions, so I used the `chmod` command to remove them. The `researcher2` user already had execute permissions, so they did not need to be added.

Summary

I changed multiple permissions to match the level of authorization my organization wanted for files and directories in the `projects` directory. The first step in this was using `ls -la` to check the permissions for the directory. This informed my decisions in the following steps. I then used the `chmod` command multiple times to change the permissions on files and directories.