File permissions in Linux

Project description

In this activity, I will create a new portfolio document to demonstrate my experience using Linux commands to manage file permissions.

Check file and directory details

ls -a: Displays hidden files. Hidden files start with a period (**.**) at the beginning.

ls -l: Displays permissions to files and directories. Also displays other additional information, including owner name, group, file size, and the time of last modification.

ls -la: Displays permissions to files and directories, including hidden files. This is a combination of the

other two options.

Describe the permissions string

1st

drwxrwxrwx

file type

d for directory

- for a regular file

2nd

d**r**wxrwxrwx

read permissions for the user

r if the user has read permissions

- if the user lacks read permissions

3rd

dr**w**xrwxrwx

write permissions for the user

w if the user has write permissions

- if the user lacks write permissions

4th

drw**x**rwxrwx

execute permissions for the user

x if the user has execute permissions

- if the user lacks execute permissions

5th

drwx**r**wxrwx

read permissions for the group

r if the group has read permissions

- if the group lacks read permissions

6th

drwxr**w**xrwx

write permissions for the group

w if the group has write permissions

- if the group lacks write permissions

7th

drwxrw**x**rwx

execute permissions for the group

x if the group has execute permissions

- if the group lacks execute permissions

8th

drwxrwx**r**wx

read permissions for other

r if the other owner type has read permissions

- if the other owner type lacks read permissions

9th

drwxrwxr**w**x

write permissions for other

w if the other owner type has write permissions

- if the other owner type lacks write permissions

10th

drwxrwxrw**x**

execute permissions for other

x if the other owner type has execute permissions

- if the other owner type lacks execute permissions

Change file permissions

Changing permissions

The **principle of least privilege** is the concept of granting only the minimal access and authorization required to complete a task or function. In other words, users should not have privileges that are beyond what is necessary. Not following the principle of least privilege can create security risks.

The `chmod` command can help you manage this authorization. The `chmod` command changes permissions on files and directories.

Using chmod

The `chmod` command requires two arguments. The first argument indicates how to change permissions, and the second argument indicates the file or directory that you want to change permissions for. For example, the following command would add all permissions to `login_sessions.txt`:

```
chmod u+rwx,g+rwx,o+rwx login_sessions.txt
```

If you wanted to take all the permissions away, you could use

```
chmod u-rwx,g-rwx,o-rwx login_sessions.txt
```

Another way to assign these permissions is to use the equals sign (=) in this first argument. Using = with `chmod` sets, or assigns, the permissions exactly as specified. For example, the following command would set read permissions for `login_sessions.txt` for user, group, and other:

```
chmod u=r,g=r,o=r login_sessions.txt
```

This command overwrites existing permissions. For instance, if the user previously had write permissions, these write permissions are removed after you specify only read permissions with =.

The following table reviews how each character is used within the first argument of `chmod`:

Character	Description
`u`	indicates changes will be made to user permissions
`g`	indicates changes will be made to group permissions
`o`	indicates changes will be made to other permissions
`+`	adds permissions to the user, group, or other
`-`	removes permissions from the user, group, or other
`=`	assigns permissions for the user, group, or other

Change file permissions on a hidden file

`ls -a`: Displays hidden files. Hidden files start with a period (`.`) at the beginning.

`ls -l`: Displays permissions to files and directories. Also displays other additional

information, including owner name, group, file size, and the time of last modification.

`ls -la`: Displays permissions to files and directories, including hidden files. This is a

combination of the other two options.

Change directory permissions

In the /home/researcher2/projects directory, there are five files with the following names and permissions:

- project_k.txt
  - User = read, write,
  - Group = read, write
  - Other = read, write
- project_m.txt
  - User = read, write

○ Group = read
○ Other = none
● project_r.txt
○ User= read, write
○ Group = read, write
○ Other = read
● project_t.txt
○ User = read, write
○ Group = read, write
○ Other = read
● .project_x.txt
○ User = read, write
○ Group = write
○ Other = none

There is also one subdirectory inside the projects directory named drafts. The permissions on drafts are:
● User = read, write, execute
● Group = execute
● Other = none

Summary

Authorization is the concept of granting access to specific resources in a system. It's important because without authorization any user could access and modify all files belonging to other users or system files. This would certainly be a security risk.

In Linux, file and directory permissions are used to specify who has access to specific files and directories. You'll explore file and directory permissions and change the ownership of a file and a directory to limit who can access them.