File permissions in Linux

Project description

In this project, I demonstrate my knowledge of various Linux commands used to modify file permissions. In this scenario I am a security professional at a large organization. I mainly work with their research team. Part of my job is to ensure users on this team are authorized with the appropriate permissions. This helps keep the system secure.

My task is to examine existing permissions on the file system. I'll need to determine if the permissions match the authorization that should be given. If they do not match, I'll need to modify the permissions to authorize the appropriate users and remove any unauthorized access.

Check file and directory details

To check permissions, you can use the command:

```
ls -la
```

This command uses **ls** to to display the names of files and directories in the current working directory, while the **-la** option modifies the behavior of the command to display permissions to files and directories, including hidden files. In the shell, permissions are as follows:

```
drwxr-xr-x 3 researcher2 research_team 4096 Apr  1 22:37 .
drwxr-xr-x 3 researcher2 research_team 4096 Apr  1 23:11 ..
-rw--w---- 1 researcher2 research_team   46 Apr  1 22:37 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Apr  1 22:37 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Apr  1 22:37 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Apr  1 22:37 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Apr  1 22:37 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Apr  1 22:37 project_t.txt
```

Describe the permissions string

In front of each file and directory, there is a 10-character string that is used to describe the permissions for each file. The first character represents the file type, with **d** being used for directories, and **-** being used for files.

The next 9 characters describe the permissions for each group, with each group having 3 permissions. The first group of 3 characters is used to describe the permissions for the user, the next 3 describe the permissions of the group, and the final 3 describe the permissions of others.

The first character is used to represent read permissions, represented by either an **r** if they have the permission granted, or the a **-** if they do NOT have the permission granted.

The second character represents the permission to write, as represented by either a **w** or a **-**, depending on if they have the permission granted or not.

Finally, the third character in each group represents the ability to execute, as represented by either an **x** or a **-** to show whether the permission has been granted or not.

For example, if we take a look at the file **project_k.txt**, we can see that their 10 characters read as follows:

-rw-rw-rw-

As stated before the first character, highlighted in yellow, shows the file type. In this case there is a **-**, indicating that this is a file and not a directory. Next, the group of characters that represent user permissions, highlighted in red, read **rw-**, indicating that the user has permissions to read and write, but not to execute. The permissions are the same for the group, highlighted in green, and for other, highlighted in blue.

Change file permissions

The organization does not allow others to have write access to any files. To make this change, I use the **chmod** command to change the permissions on each file. Following the **chmod** command, I added the options to remove write permissions for each file. These options are represented as **u-w**, **g-w**, and **o-w**. In this case, the first letter represents the owner whose permission is being changed. The center symbol represents whether a permission is being granted or revoked with **-** being used to revoke and **+** used to grant. Finally, to end the command, I type the file/directory being modified. Let's use **project_k.txt** as an example. The current permissions are as follows:

```
-rw-rw-rw-
```

To remove permissions i would input these commands:

```
chmod u-w project_k.txt
chmod g-w project_k.txt
chmod o-w project_k.txt
```

Alternatively, I could reduce the number of commands by using a single command, separating each option with a comma like so:

```
chmod u-w,g-w,o-w project_k.txt
```

This would produce the same result as the 3 commands I used before.

Change file permissions on a hidden file

The research team has archived **.project_x.txt**, which is why it's a hidden file. This file should not have write permissions for anyone, but the user and group should be able to read the file. Because I already removed the write permissions, I would only need to add the read permissions for the group for this file. To do this, first I would need to make sure I display the hidden files and its permissions using the **ls -la** command. Next I would use the following command:

```
chmod g+r .project_x.txt
```

It is very important that the **.** is included at the beginning of the file name, as it indicates that it is a hidden file.

Change directory permissions

The files and directories in the projects directory belong to the **researcher2** user. Only **researcher2** should be allowed to access the **drafts** directory and its contents. This requires a change in the execute permissions. As **researcher2** already has execute permissions, we would only need to remove execute permissions from anyone else who has it, in this case, that would be the group. To do this we would simply use the command:

```
chmod g-x drafts
```

This command removes execute permissions from the group for the **drafts** directory.

Summary

During the course of this project, I have changed the permissions on files and directories for all types of owners. I have also demonstrated my knowledge of the Linux command line and bash shell through the explanations of various commands such as **ls** and **chmod**. I have also demonstrated my ability to read permissions through my explanation of the 10-character string that demonstrates permissions. Finally, I have included my ability to also read and modify permissions for hidden files and directories.