File permissions in Linux

Project description

The purpose of this project is to demonstrate how to manage file permissions in Linux. We will use commands that allow us to check permissions for files and assign the correct permissions to files based on the principle of least privilege.

Check file and directory details

To check file and directory permissions, you can use the ls -al command:

```
researcher2@e15f8d2006fa:~/projects$ ls -al
```

These are the results from the command:

```
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Sep 17 06:12 .
drwxr-xr-x 3 researcher2 research_team 4096 Sep 17 07:03 ..
-rw--w---- 1 researcher2 research_team   46 Sep 17 06:12 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Sep 17 06:12 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Sep 17 06:12 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Sep 17 06:12 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Sep 17 06:12 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Sep 17 06:12 project_t.txt
```

Describe the permissions string

The first piece of information for each file is the permission string. This 10 character string indicates the file type and access permissions across the organization. For example, the project_r.txt file starts with a “-” meaning it's a regular file type. The next 3 characters represent user access permissions, and it can be seen that the user has r (read) permissions meaning they can read the document and w (write) permissions meaning they can edit the document, but they have a “-” in the third position meaning they don’t have execute permissions. The next 3 characters represent permissions for the group the user is a part of and in this example they are the same (rw-). The last 3 characters represent other users in the organization. In this case they only have the r permission therefore they can only read the file.

Change file permissions

To change file permissions you must use the chmod command:

```
researcher2@e15f8d2006fa:~/projects$ chmod o-w project_k.txt
```

To use this command you type chmod followed by the permission changes you want to make and then the file that you are changing. In this example, we wanted to remove write permissions from the other group.

Change file permissions on a hidden file

Changing hidden file permissions uses the same command (chmod) as normal files as seen here:

```
researcher2@0a35f216763e:~/projects$ chmod u=r,g=r .project_x.txt
```

In this case we wanted to only have read permissions for the user and the group. The user had read and write permissions and the group had write permissions. By entering the arguement u=r and g=r, all previous permissions are overwritten to now only include those specified in the current command which are read permissions for the user and group.

Change directory permissions

```
researcher2@0a35f216763e:~/projects$ chmod g-x drafts
```

Changing directory permissions again uses the chmod command. In this case on the user needed access to the drafts directory so the group access needed to be removed. By entering the argument g-x, we remove the group's ability to access the directory.

Summary

Through this scenario we were tasked to give and take permissions as required by the organization. We accomplished this using the chmod command and adding arguments to correct improper permission statuses of certain documents. We also worked with permissions of a hidden file and a directory.