

File permissions in Linux

Project description

The research team asked that many files and folders in the /home/researcher2/projects directory have their permissions updated. Some things have greater access than needed, therefore reducing permissions mitigates possible security issues. To do this assignment, I looked over and assessed the present permissions, then corrected access levels using Linux commands.

Check file and directory details

First, I used the following command to view all files, including hidden files:

```
ls -la /home/researcher2/projects
```

Describe the permissions string

The 10-character string can be deconstructed to determine who is authorized to access the file and their specific permissions. The characters and what they represent are as follows:

- **1st character:** This character is either a `d` or hyphen (`-`) and indicates the file type. If it's a `d`, it's a directory. If it's a hyphen (`-`), it's a regular file.
- **2nd-4th characters:** These characters indicate the read (`r`), write (`w`), and execute (`x`) permissions for the user. When one of these characters is a hyphen (`-`) instead, it indicates that this permission is not granted to the user.
- **5th-7th characters:** These characters indicate the read (`r`), write (`w`), and execute (`x`) permissions for the group. When one of these characters is a hyphen (`-`) instead, it indicates that this permission is not granted for the group.
- **8th-10th characters:** These characters indicate the read (`r`), write (`w`), and execute (`x`) permissions for other. This owner type consists of all other users on the system apart from the user and the group. When one of these characters is a hyphen (`-`) instead, that indicates that this permission is not granted for other.

For example, the file permissions for `project_t.txt` are `-rw-rw-r--`. Since the first character is a hyphen (`-`), this indicates that `project_t.txt` is a file, not a directory. The second, fifth, and eighth characters are all `r`, which indicates that user, group, and other all have read permissions. The third and sixth characters are `w`, which

indicates that only the user and group have write permissions. No one has execute permissions for `project_t.txt`.

Change file permissions

I removed write access for “other” using:

```
chmod o-w project_k.txt  
ls -la /home/researcher2/projects
```

Updated permissions for `project_k.txt` became:

```
rw-rw-r--
```

Change file permissions on a hidden file

To correct this, I ran:

```
chmod g-w .project_x.txt  
chmod u-w .project_x.txt  
ls -la /home/researcher2/projects
```

The new permissions became:

```
r--r----
```

Change directory permissions

```
chmod g-x drafts  
ls -la /home/researcher2/projects
```

Summary

After reviewing the `/home/researcher2/projects` directory, I changed the permissions to increase security. Using `ls -la`, I found extremely permissive settings and used `chmod` to change individual files and the `drafts` directory. These improvements guarantee that important files are only available to authorized users and prevent unauthorized alterations.

