File permissions in Linux

Project description

This project demonstrates how to secure a directory by setting proper file and directory permissions in Linux. Here's a breakdown of the commands and techniques:

Check file and directory details

Here's how I used Linux commands to find out who has access to a directory:

```
jehuty-v1@Analyst-Machine:~/Desktop/researcher2/Projects$ ls -la
total 12
drwxr-xr-x 3 root root 4096 Nov  1 14:05 .
drwxr-xr-x 3 root root 4096 Nov  1 00:04 ..
drwx--x--- 2 root root 4096 Nov  1 14:05 drafts
-rw-rw-rw- 1 root root    0 Nov  1 00:05 project_k.txt
-rw-r----- 1 root root    0 Nov  1 00:05 project_m.txt
-rw-rw-r-- 1 root root    0 Nov  1 00:37 project_r.txt
-rw-rw-r-- 1 root root    0 Nov  1 00:37 project_t.txt
-rw--w---- 1 root root    0 Nov  1 00:28 .project_x.txt
jehuty-v1@Analyst-Machine:~/Desktop/researcher2/Projects$
```

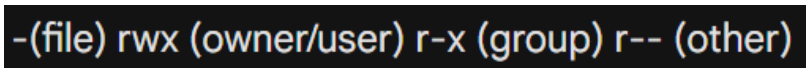
First, I listed the contents of the selected directory using `ls -la`. This command shows all files, even hidden ones (those with a period . at the beginning of their name), and their permissions. You can see there's a directory called `drafts`, a hidden file (`.project_x.txt`), and four other files. The permissions are shown in the first column—that's the 10-character string.

Describe the permissions string

Let's break down those 10 characters that show permissions:

- **1st character:** This character tells you if it's a directory (`d`) or a regular file (`-`).
- **2nd-4th characters:** These characters show the **owner**'s permissions: read (`r`), write (`w`), and execute (`x`). A `-` means that permission is not granted.
- **5th-7th characters:** These characters are the same, but for the **group**.
- **8th-10th characters:** These characters are for everyone else (**'others'**).

For example, `-rwxr-xr--` means that: (1) it's a file (because of the `-` at the beginning), (2) the **owner** can read, write, and execute; (3) the **group** can read and execute; and (4) **others** can only read. Here's a quick visual to help explain it in a different way:



Change file permissions on a hidden file

In this task, I was asked to remove the 'write' permission for “others” on the `project_k.txt` file. I used the following command:

```
jehuty-v1@Analyst-Machine:~/Desktop/researcher2/Projects$ sudo chmod u-w,g-w,g+r .project_x.txt
[sudo] password for jehuty-v1:
jehuty-v1@Analyst-Machine:~/Desktop/researcher2/Projects$ ls -la
total 12
drwxr-xr-x 3 root root 4096 Nov  1 14:05 .
drwxr-xr-x 3 root root 4096 Nov  1 00:04 ..
drwx--x--- 2 root root 4096 Nov  1 14:05 drafts
-rw-rw-r-- 1 root root    0 Nov  1 00:05 project_k.txt
-rw-r----- 1 root root    0 Nov  1 00:05 project_m.txt
-rw-rw-r-- 1 root root    0 Nov  1 00:37 project_r.txt
-rw-rw-r-- 1 root root    0 Nov  1 00:37 project_t.txt
-r--r----- 1 root root    0 Nov  1 00:28 .project_x.txt
jehuty-v1@Analyst-Machine:~/Desktop/researcher2/Projects$
```

I used the `chmod` command with the following arguments:

- `u-w`: Removes write permission for the owner.
- `g-w`: Removes write permission for the group.
- `g+r`: Adds read permission for the group.

I used `sudo` because I needed administrator privileges. After running the command, I used `ls -la` to double-check that the permissions were updated correctly

Change directory permissions

In the final task, I was asked to make the `drafts` directory accessible only to the “owner”, only. I used this command:

```
jehuty-v1@Analyst-Machine:~/Desktop/researcher2/Projects$ sudo chmod g-x drafts
[sudo] password for jehuty-v1:
jehuty-v1@Analyst-Machine:~/Desktop/researcher2/Projects$ ls -la
total 12
drwxr-xr-x 3 root root 4096 Nov  1 14:05 .
drwxr-xr-x 3 root root 4096 Nov  1 00:04 ..
drwx------ 2 root root 4096 Nov  1 14:05 drafts
-rw-rw-r-- 1 root root    0 Nov  1 00:05 project_k.txt
-rw-r----- 1 root root    0 Nov  1 00:05 project_m.txt
-rw-rw-r-- 1 root root    0 Nov  1 00:37 project_r.txt
-rw-rw-r-- 1 root root    0 Nov  1 00:37 project_t.txt
-r--r----- 1 root root    0 Nov  1 00:28 .project_x.txt
jehuty-v1@Analyst-Machine:~/Desktop/researcher2/Projects$
```

This command uses symbolic notation (`g-x`) to remove the execute permission (`-x`) for the group (`g`). Once again I used `sudo` (personal preference instead of using superuser)because I needed administrator privileges to change the permissions of this directory.

Summary

In this project, I used Linux commands to secure a directory by setting the correct permissions on files and folders. I started by checking the current permissions with `ls -la`, then used `chmod` to make the necessary changes.