File permissions in Linux

Project description

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.

As a security analyst, setting appropriate access permissions is critical to protecting sensitive information and maintaining the overall security of a system.

Check file and directory details

you must explore the permissions of the `projects` directory and the files it contains. The lab starts with `/home/researcher2` as the current working directory. This is because you're changing permissions for files and directories belonging to the `researcher2` user.

1. Navigate to the `projects` directory.
2. List the contents and permissions of the `projects` directory.

The permissions of the files in the `projects` directory are as follows:

```
total 20
```

```
drwx--x--- 2 researcher2 research_team 4096 Oct 14 18:40 drafts
-rw-rw-rw- 1 researcher2 research_team   46 Oct 14 18:40 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Oct 14 18:40 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct 14 18:40 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Oct 14 18:40 project_t.txt
```

Note: *The date and time information returned is the same as the date and time when you ran the command. Therefore, it is different from the date and time in the example.*

As you may recall from the video lesson, a 10-character string begins each entry and indicates how the permissions on the file are set. For instance, a directory with full permissions for all owner types would be `drwxrwxrwx`:

- The 1st character indicates the file type. The `d` indicates it's a directory. When this character is a hyphen (`-`), it's a regular file.
- The 2nd-4th 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.
- The 5th-7th 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.
- The 8th-10th characters indicate the read (`r`), write (`w`), and execute (`x`) permissions for the owner type of 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.

This lab exercise requires you to examine the file permissions within the `projects` directory. The starting point for this task is the `/home/researcher2` directory, as the goal is to modify permissions for files and folders owned by the `researcher2` user.

Steps:

1. Change your current working directory to `projects`.
2. Display the contents and associated permissions of the `projects` directory.

Observed Permissions in `projects`:

The permissions and contents of the `projects` directory are listed below:
total 20
drwx--x--- 2 researcher2 research_team 4096 Oct 14 18:40 drafts
-rw-rw-rw- 1 researcher2 research_team 46 Oct 14 18:40 project_k.txt
-rw-r----- 1 researcher2 research_team 46 Oct 14 18:40 project_m.txt
-rw-rw-r-- 1 researcher2 research_team 46 Oct 14 18:40 project_r.txt
-rw-rw-r-- 1 researcher2 research_team 46 Oct 14 18:40 project_t.txt
Note: *The displayed date and time reflect when the command was executed and will differ from the example.*

Understanding the Permission String:

As covered in the video lesson, a 10-character string precedes each file or directory entry, detailing its permissions. For instance, a directory granting full permissions to all owner types would appear as `drwxrwxrwx`. This string is interpreted as follows:

Character Position	Meaning	Details
1st Character	**File Type**	`d` indicates a directory. A hyphen (`-`) indicates a regular file.
2nd - 4th	**User (Owner) Permissions**	Specifies **r**ead, **w**rite, and **x**ecute permissions for the file owner. A hyphen (`-`) indicates the absence of that permission.
5th - 7th	**Group Permissions**	Specifies **r**ead, **w**rite, and **x**ecute permissions for the group associated with the file. A hyphen (`-`) indicates the absence of that permission.
8th - 10th	**Other Permissions**	Specifies **r**ead, **w**rite, and **x**ecute permissions for all users on the system who are not the owner or part of the

		file's group (the 'other' owner type). A hyphen (`-`) indicates the absence of that permission.

Describe the permissions string

- The 1st character indicates the file type. The `d` indicates it's a directory. When this character is a hyphen (`-`), it's a regular file.
- The 2nd-4th 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.
- The 5th-7th 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.
- The 8th-10th characters indicate the read (`r`), write (`w`), and execute (`x`) permissions for the owner type of 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.

File permissions are displayed by 10 characters:

- The 1st character indicates the file type ('d' for directory, '-' for regular file).
- Characters 2-4 show the user's read (r), write (w), and execute (x) permissions. A hyphen (-) means the permission is denied.
- Characters 5-7 show the group's read (r), write (w), and execute (x) permissions. A hyphen (-) means the permission is denied.
- Characters 8-10 show the 'other' users' (all users except the owner and group) read (r), write (w), and execute (x) permissions. A hyphen (-) means the permission is denied.

Change file permissions

you must determine whether any files have incorrect permissions and then change the permissions as needed. This action will remove unauthorized access and strengthen security on the system.

None of the files should allow the other users to write to files.

1. Check whether any files in the `projects` directory have write permissions for the owner type of other.

Which file grants other users write permissions?
project_t.txt
project_m.txt
project_k.txt

2. Change the permissions of the file identified in the previous step so that the owner type of other doesn't have write permissions.

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

Copied!

Note: *Permissions are granted for three different types of owners, namely user, group, and other.*
In the chmod command u sets the permissions for the user who owns the file,g sets the permissions for the group that owns the file, and o sets the permissions for others.

3. The file `project_m.txt` is a restricted file and should not be readable or writable by the group or other; only the user should have these permissions on this file. List the contents and permissions of the current directory and check if the group has read or write permissions.

What are the group permissions on the project_m.txt file?
Read, write, and execute

```
Read and write
Read only
```

4. Use the `chmod` command to change permissions of the `project_m.txt` file so that the group doesn't have read or write permissions.

Click **Check my progress** to verify that you have completed this task correctly.

To enhance system security and prevent unauthorized access, you must first identify any files with incorrect permissions and then adjust those permissions as necessary. A key security requirement is that **no files should allow 'other' users to write to them.**Steps for Permission Management

1. Identify Files with 'Other' Write Permissions:

Check the files within the `projects` directory for any that grant write permissions to the 'other' owner type.

- Which file currently grants write permissions to 'other' users?
  - `project_t.txt`
  - `project_m.txt`
  - **`project_k.txt`**

2. Remove 'Other' Write Permissions:

Modify the permissions of the file identified in the previous step to remove the write permission for the 'other' owner type.

- **Command:** `chmod o-w project_k.txt`

Note on `chmod` owners: *Permissions are categorized for three owner types:* ***user*** *(u),* ***group*** *(g), and* ***other*** *(o).*

3. Verify Permissions for a Restricted File:

The file `project_m.txt` is highly restricted; only the file's owner (user) should have read and write permissions. Neither the group nor 'other' should have read or write access.

List the current directory contents and permissions to check the group's permissions on `project_m.txt`.

- What permissions does the group currently have on `project_m.txt`?

- Read, write, and execute
- **Read and write**
- Read only

4. Restrict Group Permissions on `project_m.txt`:

Use the `chmod` command to remove both read and write permissions for the group on the `project_m.txt` file.

[Action: Implement the appropriate `chmod` command for step 4]

Click **Check my progress** to confirm the correct completion of these steps.

Change file permissions on a hidden file

you must determine if a hidden file has incorrect permissions and then change the permissions as needed. This action will further remove unauthorized access and strengthen security on the system.

The file `.project_x.txt` is a hidden file that has been archived and should not be written to by anyone. (The user and group should still be able to read this file.)

1. Check the permissions of the hidden file `.project_x.txt` and answer the question that follows.

Which owner type has the incorrect write permissions?
Just the group
The user and the group
Just the user

2. Change the permissions of the file `.project_x.txt` so that both the user and the group can read, but not write to, the file.

Note: *Be sure to start the name of a hidden file with a period (`.`).*
Click **Check my progress** to verify that you have completed this task correctly.

To enhance system security and prevent unauthorized access, you need to address the permissions of a specific hidden file.

The archived hidden file, `.project_x.txt`, should be strictly read-only for all users. Currently, both the owner and the group should retain read access, but neither should have write permission.

Steps:

1. **Permission Check:** Examine the current permissions of `.project_x.txt`.
    - **Question:** Which owner type (User, Group, or Both) currently possesses incorrect write permissions?
        - Just the group
        - The user and the group
        - Just the user
2. **Permission Change:** Modify the permissions for `.project_x.txt` to ensure that both the user and the group can read the file, but neither can write to it.

***Note:** Remember to prefix the file name with a period (.) since it is a hidden file.*

Once completed, use the **Check my progress** button to verify the changes.

Change directory permissions

you must change the permissions of a directory. First, you'll check the group permissions of the `/home/researcher2/projects/drafts` directory and then modify the permissions as required. (You should be in the `projects` directory while managing the permissions of its subdirectory `drafts`.)

Only the `researcher2` user should be allowed to access the `drafts` directory and its contents. (This means that only `researcher2` should have execute privileges.)

1. Check the permissions of the `drafts` directory and answer the following question.

Does the group have permissions set to access the drafts directory and its contents?

```
No
Yes
```

2. Remove the execute permission for the group from the `drafts` directory.

Click **Check my progress** to verify that you have completed this task correctly.

You must change the permissions of a directory. First, you'll check the group permissions of the `/home/researcher2/projects/drafts` directory and then modify the permissions as required. (You should be in the `projects` directory while managing the permissions of its subdirectory `drafts`.)

Only the `researcher2` user should be allowed to access the `drafts` directory and its contents. (This means that only `researcher2` should have execute privileges.)

1. Check the permissions of the `drafts` directory.

The group has permissions set to access the `drafts` directory and its contents: No

1. Remove the execute permission for the group from the `drafts` directory.

   Command: `chmod g-x drafts`

Click **Check my progress** to verify that you have completed this task correctly.

Summary

This Linux File Permissions lab focused on a security analyst's task to audit and correct file and directory permissions for the `researcher2` user's files to enforce a least-privilege security model. Key actions, using the `chmod` command with symbolic notation, included:

1. **Removing 'other' write access** from `project_k.txt` (`chmod o-w`).
2. **Removing group read/write access** from `project_m.txt` (`chmod g-rw`).
3. **Making the hidden file `.project_x.txt` read-only** for both user and group (`chmod ug-w`).
4. **Removing group execute (access) permission** from the `drafts` directory (`chmod g-x`).

The exercise confirmed proficiency in using `chmod` to restrict unauthorized access and strengthen system security.