

# File Permission in Linux

## Introduction

This presentation covers three key aspects of the project: the scenario, the notes used, and the final result. The scenario outlines the project goals, the notes provide guidance, and the final project demonstrates how these elements were applied.

## Scenario:


### Scenario

---

Review the scenario below. Then, complete the step-by-step instructions.


You are a security professional at a large organization. You mainly work with their research team. Part of your job is to ensure users on this team are authorized with the appropriate permissions. This helps keep the system secure.

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

**Note:** This scenario involves investigating and updating the same file permissions as the ones in the [Manage authorization](#)  lab. You can revisit the lab to get screenshots to include in your portfolio document. If you choose, it's also possible to complete this activity without revisiting the lab by typing your commands in the template.

## Notes:

### ✓ Step 3: Check file and directory details

In the **Manage authorization** lab, check the permissions set for files and subdirectories in the **projects** directory. Make sure you display all permissions, including hidden files. Or, use the content of [Current file permissions](#)  document to determine the current permissions.

Describe the command you can use to check permissions in the **Check file and directory details** section of the **File permissions in Linux** template. From the lab, take a screenshot of the Linux command you used. Or, type this command directly into the template.

Then, use either the output of this command in the lab or the content or the **Current file permissions** document to indicate the current permissions. If using the **Current file permissions** document, write these in the 10-character string that would be part of the command's output.

### ✓ Step 4: Describe the permissions string

Choose one example from the output in the previous step. In the **Describe the permissions string** section of the **File permissions in Linux** template, write a short description that explains the 10-character string in the example. You should describe what the 10-character string is for and what each character represents.

### ▼ Step 5: Change file permissions

The organization does not allow other to have write access to any files. Based on the permissions established in Step 3, identify which file needs to have its permissions modified. Use a Linux command to modify these permissions.

Describe the command you used and its output in the **Change file permissions** section of the **File permissions in Linux** template. In the **Manage authorization** lab, take a screenshot of the Linux command you used. Or, type this command directly into the template.

### ▼ Step 6: 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. Use a Linux command to assign `.project_x.txt` the appropriate authorization.

Describe the command you used and its output in the **Change file permissions on a hidden file** section of the **File permissions in Linux** template. In the **Manage authorization** lab, take a screenshot of the Linux command you used. Or, type this command directly into the template.

### ▼ Step 7: 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. Use a Linux command to modify the permissions accordingly.

Describe the command you used and its output in the **Change directory permissions** section of the **File permissions in Linux** template. In the **Manage authorization** lab, take a screenshot of the Linux command you used. Or, type this command directly into the template.

### ▼ Step 8: Finalize your document

To finalize the document and make its purpose clear to potential employers, be sure to complete the **Project description** and **Summary** sections of the **File permissions in Linux** template.

In the Project description section, give a general overview of the scenario and what you accomplish through Linux. Write two to four sentences.

In the Summary section, provide a short summary of the previous tasks and connect them to the scenario. Write approximately two to four sentences.

## Final Project:

# File permissions in Linux

## Project description

The research team at my organization must revise the file permissions for specific files and directories within the projects directory. The current permissions do not align with the required level of authorization. Verifying and adjusting these permissions is crucial for maintaining the security of their system. To accomplish this task, I undertook the following steps:

## Check file and directory details

The code below demonstrates how I utilized Linux commands to discover the current permissions, including those of hidden files, for a specific folder in the file system.

```
researcher2@bd2cd4ce8276:~$ pwd
/home/researcher2
researcher2@bd2cd4ce8276:~$ cd projects
researcher2@bd2cd4ce8276:~/projects$ pwd
/home/researcher2/projects
researcher2@bd2cd4ce8276:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Jan 27 17:27 .
drwxr-xr-x 3 researcher2 research_team 4096 Jan 27 17:43 ..
-rw--w---- 1 researcher2 research_team  46 Jan 27 17:27 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Jan 27 17:27 drafts
-rw-rw-rw- 1 researcher2 research_team  46 Jan 27 17:27 project_k.txt
-rw-r----- 1 researcher2 research_team  46 Jan 27 17:27 project_m.txt
-rw-rw-r-- 1 researcher2 research_team  46 Jan 27 17:27 project_r.txt
-rw-rw-r-- 1 researcher2 research_team  46 Jan 27 17:27 project_t.txt
researcher2@bd2cd4ce8276:~/projects$
```

The initial line in the screenshot presents the command I input, while the subsequent lines showcase the results. The code enumerates all items within the projects directory. I employed the ls command with the -la option to exhibit a comprehensive list of file contents, including hidden files. The results from my command reveal the presence of a directory named drafts, a hidden file named .project\_x.txt, and five additional project files. The 10-character string in the primary column denotes the permissions assigned to each file or directory.

## Describe the permissions string

- 1st character: This symbol is either a d or a hyphen (-), indicating the file type. If it's a d, it signifies a directory, and if it's a hyphen (-), it designates a regular file.
- 2nd-4th characters: These symbols represent the read (r), write (w), and execute (x) permissions for the user. If any of these symbols is a hyphen (-), it denotes that the respective permission is not granted to the user.
- 5th-7th characters: These symbols denote the read (r), write (w), and execute (x) permissions for the group. If any of these symbols is a hyphen (-), it signifies that the corresponding permission is not granted for the group.
- 8th-10th characters: These symbols indicate the read (r), write (w), and execute (x) permissions for others. This owner type includes all other users on the system apart from the user and the group. If any of these symbols is a hyphen (-), it indicates that the corresponding permission is not granted for others.

For instance, the permission string for `project_t.txt` is `-rw-rw-r--`. As the first character is a hyphen (-), it denotes that `project_t.txt` is a file, not a directory. The second, fifth, and eighth characters, all being 'r', signify that the user, group, and others have read permissions. The third and sixth characters, 'w', indicate that only the user and group possess write permissions. No user, group, or others have execute permissions for `project_t.txt`.

## Change file permissions

The organization decided that 'other' should not have write access to any of their files. To adhere to this, I consulted the file permissions I had retrieved earlier. I identified that project\_k.txt requires the removal of write access for 'other'.

The subsequent code illustrates how I employed Linux commands to accomplish this:

```
researcher2@bd2cd4ce8276:~/projects$ chmod o-w project_k.txt
researcher2@bd2cd4ce8276:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Jan 27 17:27 .
drwxr-xr-x 3 researcher2 research_team 4096 Jan 27 17:43 ..
-rw--w---- 1 researcher2 research_team  46 Jan 27 17:27 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Jan 27 17:27 drafts
-rw-rw-r-- 1 researcher2 research_team  46 Jan 27 17:27 project_k.txt
-rw-r----- 1 researcher2 research_team  46 Jan 27 17:27 project_m.txt
-rw-rw-r-- 1 researcher2 research_team  46 Jan 27 17:27 project_r.txt
-rw-rw-r-- 1 researcher2 research_team  46 Jan 27 17:27 project_t.txt
researcher2@bd2cd4ce8276:~/projects$
```

The initial two lines in the screenshot showcase the commands I inputted, while the subsequent lines exhibit the output of the second command. The `chmod` command is employed to modify permissions on files and directories. The first argument signifies which permissions are to be altered (`o-w`), and the second argument specifies the file or directory (`project_k.txt`). In this instance, I eliminated write permissions from 'other' for the `project_k.txt` file. Following this, I utilized `ls -la` to inspect the modifications I implemented.

## Change the permissions on a hidden file

The research team at my organization has recently archived project\_x.txt. They intend to restrict write access to this project, allowing only the user and group to have read access. The subsequent code illustrates how I employed Linux commands to modify the permissions:

```
researcher2@bd2cd4ce8276:~/projects$ chmod u-w,g-w,g+r .project_x.txt
researcher2@bd2cd4ce8276:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Jan 27 17:27 .
drwxr-xr-x 3 researcher2 research_team 4096 Jan 27 17:43 ..
-r--r----- 1 researcher2 research_team  46 Jan 27 17:27 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Jan 27 17:27 drafts
-rw-rw-r-- 1 researcher2 research_team  46 Jan 27 17:27 project_k.txt
-rw-r----- 1 researcher2 research_team  46 Jan 27 17:27 project_m.txt
-rw-rw-r-- 1 researcher2 research_team  46 Jan 27 17:27 project_r.txt
-rw-rw-r-- 1 researcher2 research_team  46 Jan 27 17:27 project_t.txt
researcher2@bd2cd4ce8276:~/projects$
```

The initial two lines in the screenshot exhibit the commands I inputted, while the subsequent lines reveal the output of the second command. I identified that .project\_x.txt is a hidden file due to its initial period (.). In this instance, I revoked write (w) permissions from both the user and group, and granted read (r) permissions to the group. Specifically, I withdrew the user's write permissions using 'u-w'. Subsequently, I removed the group's write permissions with 'g-w', and introduced read permissions to the group with 'g+r'.

## Change directory permissions

My organization exclusively intends to grant access to the drafts directory and its contents to the user 'researcher2'. This implies that only 'researcher2' should possess execute permissions, excluding all others.

The following code demonstrates how I used Linux commands to change the permissions:

```
researcher2@bd2cd4ce8276:~/projects$ chmod g-x drafts
researcher2@bd2cd4ce8276:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Jan 27 17:27 .
drwxr-xr-x 3 researcher2 research_team 4096 Jan 27 17:43 ..
-r--r----- 1 researcher2 research_team  46 Jan 27 17:27 .project_x.txt
drwx----- 2 researcher2 research_team 4096 Jan 27 17:27 drafts
-rw-rw-r-- 1 researcher2 research_team  46 Jan 27 17:27 project_k.txt
-rw-r----- 1 researcher2 research_team  46 Jan 27 17:27 project_m.txt
-rw-rw-r-- 1 researcher2 research_team  46 Jan 27 17:27 project_r.txt
-rw-rw-r-- 1 researcher2 research_team  46 Jan 27 17:27 project_t.txt
researcher2@bd2cd4ce8276:~/projects$
```

The initial two lines in the screenshot showcase the commands I inputted, while the subsequent lines present the output of the second command. I had earlier established that the group held execute permissions, prompting me to utilize the chmod command to revoke them (chmod g-x drafts). As the researcher2 user already possessed execute permissions, there was no need to include them.

## Summary

I adjusted various permissions to align with the desired level of authorization outlined by my organization for files and directories within the projects directory, even the hidden files or directories. The initial phase involved utilizing ls -la to inspect the permissions of the directory, guiding my subsequent actions. Subsequently, I employed the chmod command multiple times to modify the permissions on both files and directories.