Start Lab 01:00:00

Exemplar: Manage authorization

Lab | Three | No cost | Introductory

★★★★★

This lab may incorporate AI tools to support your learning.

Activity overview

In this lab activity, you'll use Linux commands to configure authorization.

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 in 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.

Scenario

In this scenario, you must examine and manage the permissions on the files in the `/home/researchers/projects` directory for the `research_team` user.

The `research_team` is part of the `research_team` group.

You must check the permissions for all files in the directory, including any hidden files, to make sure that permissions align with the authorization that should be given. When it doesn't, you must change the permissions.

Here's how you'll do this task: **First**, you'll check the user and group permissions for all files in the `projects` directory. **Next**, you'll block another any files have incorrect permissions and change the permissions as needed. **Finally**, you'll check the permissions of the `/home/researchers/projects/drafts` directory and modify these permissions to remove any unauthorized access.

Note: *The lab starts with your user account, called `researcher2`, already logged in to the Linux shell. This means you can start with the tasks as soon as you click the* **Start Lab** *button.*

Disclaimer: For optimal performance and compatibility, it is recommended to use either **Google Chrome** or **Mozilla Firefox** browsers while accessing the labs.

Start your lab

You'll need to start the lab before you can access the materials. To do this, click the green "Start Lab" button at the top of the screen.

Start Lab

After you click the **Start Lab** button, you will see a shell, where you will be performing further steps in the lab. You should have a shell like this:

```
researcher2@ip-10-0-1-9:~$
```

When you have completed all the tasks, refer to the End your Lab section that follows the tasks for information on how to end your lab.

Task 1. Check file and directory details

In this task, 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 `research_team`.

1. Navigate to the `projects` directory.

The command to complete this step:

```
cd projects
```

2. List the contents and permissions of the `projects` directory.

The command to complete this step:

```
ls -l
```

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

```
total 28
drwx--x--- 2 research_team research_team 4096 Oct 15 14:40 drafts
-rw-rw---- 1 research_team research_team   46 Oct 15 14:40 project_k.txt
-rw-r----- 1 research_team research_team   46 Oct 15 14:40 project_m.txt
-rw-rw-r-- 1 research_team research_team   46 Oct 15 14:40 project_r.txt
-rw-rw--- 1 research_team research_team   46 Oct 15 14:40 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 shown 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 to 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.

The second block of text in the expanded directory listing is the user who owns the file. The third block of text is the group owner of the file.

What is the name of the group that owns the files in the projects directory?

- research_team
- other_users
- researcher2
- security_team

Submit

Answer: The `research_team` owns the files in the projects directory.

3. Check whether any hidden files exist in the `projects` directory.

The command to complete this step:

```
ls -la
```

Which of these files is hidden in the projects directory?

- project_a.txt
- There are no hidden files
- project_r.txt
- project_m.txt

Submit

Answer: The `project_n.txt` file is hidden.

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

Check file and directory details

Check my progress

Task 2. Change file permissions

In this task, 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.

The command to complete this step:

```
ls -l
```

Which file grants other users write permissions?

- project_k.txt
- project_r.txt
- project_m.txt

Submit

Answer: The `project_t.txt` file has write permissions for other users.

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

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

The command to complete this step:

```
ls -l
```

What are the group permissions on the project_m.txt file?

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

Submit

Answer: The group permissions of the `project_m.txt` file is 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.

The command to complete this step:

```
chmod g-r project_m.txt
```

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

Change file permissions

Check my progress

Task 3. Change file permissions on a hidden file

In this task, 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_n.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_n.txt` and answer the question that follows.

The command to complete this step:

```
ls -la
```

Which owner type has the incorrect write permissions?

- Just the group
- Just the user
- The user and the group

Submit

Answer: The user and group owner types have incorrect write permissions.

2. Change the permissions of the file `project_n.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 (`.`).*

The command to complete this step:

```
chmod u-w,g-w,g+r .project_n.txt
```

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

Change file permissions on a hidden file

Check my progress

Task 4. Change directory permissions

In this task, you must change the permissions of a directory. First, you'll check the group permissions of the `/home/researchers/projects/drafts` directory and then modify the permissions as required. *This should be in the `projects` directory while managing the permissions of its subdirectory `drafts`.*

Only the `research_team` 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.

The command to complete this step:

```
ls -l
```

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

- Yes
- No

Submit

Answer: Yes, the group has execute permissions and therefore has access to the `drafts` directory.

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

The command to complete this step:

```
chmod g-x drafts
```

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

Change directory permissions

Check my progress

Conclusion

Great work!

You now have practical experience in using basic Linux Bash shell commands to:

- examine file and directory permissions,
- change permissions on files, and
- change permissions on directories.

This is an important milestone on your journey toward managing authorization in Linux!

End your lab

Before you end the lab, make sure you're satisfied that you've completed all the tasks, and follow these steps:

1. Click **End Lab**. A pop-up box will appear. Click **Submit** to confirm that you're done. Ending the lab will remove your access to the Bash shell. You won't be able to access the work you've completed in it again.
2. Another pop-up box will ask you to rate the lab and provide feedback comments. You can complete these if you choose to.
3. Close the browser tabs containing the lab instructions to return to your course.
4. Refresh the browser tab for the course to mark the lab as complete.