**Activity Overview**



In this activity, you will create a new portfolio document to demonstrate your experience using Linux commands to manage file permissions. You can add this document to your cybersecurity portfolio, which you can share with prospective employers or recruiters. To review the importance of building a professional portfolio and options for creating your portfolio, read [Create a cybersecurity portfolio](https://www.coursera.org/learn/linux-and-sql/resources/wDFrc)

.

To create your portfolio document, you will review a scenario and follow a series of steps. This scenario is connected to [the lab](https://www.coursera.org/learn/linux-and-sql/ungradedLti/483Kl/activity-manage-authorization)

you have just completed about how to examine and manage file permissions. You will explain the commands you used in that lab, and this will help you prepare for future job interviews and other steps in the hiring process.

Be sure to complete this activity and answer the questions that follow before moving on. The next course item will provide you with a completed exemplar to compare to your own work.

**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](https://www.coursera.org/learn/linux-and-sql/ungradedLti/483Kl/activity-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.

**Task 1. Check file and directory details.**

A screenshot of a computer program

Description automatically generated

* 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.

The research team owns the file, so we will either change a group, or user permissions from here on out. Inputing the command ls- -a -l lets us know who has permissions on the regular files and the hidden files, if you want to list hidden files you can type -a and see hidden files.

A screenshot of a computer program

Description automatically generated

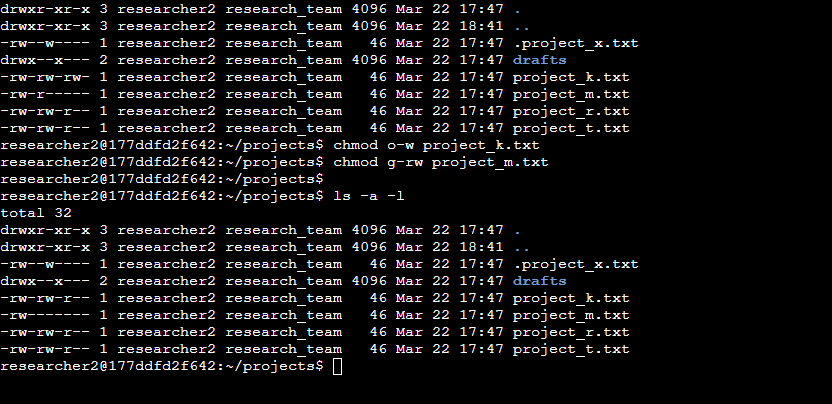
Project\_k.txt gives write permissions to other users.

**Task 2. Change file permissions**

Running this command “chmod o-w project\_k.txt” Will remove write permissions from the “other” users.

A screenshot of a computer

Description automatically generated



Running “chmod g-rw project\_m.txt” will remove Read and Write permissions from the group.

**Task 3. Change file permissions on a hidden file.**

ls -l .project\_x.txt

Running this command will let me check the permissions on the hidden file.

To check the permissions of the hidden file .project\_x.txt, you can use the ls command with the -l option to display a detailed listing of the file, including its permissions. Since the file is hidden, it might not appear in a regular listing unless explicitly specified.

You need to apply permissions to both USER and GROUP

chmod ug+r-w .project\_x.txt

Running that command wil remove Write permissions for both user and groups.

A screenshot of a computer screen

Description automatically generated

**Task 4. Change directory permissions.**

ls -ld drafts

Running this command will show the permissions for the directory and show who has permissions to this directory.

A screen shot of a computer

Description automatically generated

Running “chmod g-x drafts”command will change permissions for the directory.

* chmod: This is the command used to change file or directory permissions.
* g-x: This specifies that we're removing execute (x) permission for the group (g) on the drafts directory.

A screenshot of a computer program

Description automatically generated

Conclusion of Lab:

In this lab, we embarked on a comprehensive exploration of file and directory permissions within a Linux environment. Initially, we delved into understanding the intricacies of file permissions, where each character within the permission string held significance, denoting permissions for the user, group, and others. Through the ls -a -l command, we scrutinized these permissions, distinguishing between regular files and hidden files, and gained insights into who possessed access rights. Moving forward, we engaged in modifying file permissions to align with security requirements. By utilizing commands such as chmod o-w project\_k.txt and chmod g-rw project\_m.txt, we efficiently adjusted permissions to restrict write access for other users and remove read and write permissions for the group, respectively. Moreover, we tackled the task of managing permissions for hidden files, exemplified by the chmod ug+r-w .project\_x.txt command, ensuring both user and group permissions were appropriately adjusted, underscoring our meticulous approach to security measures.

Furthermore, our exploration extended to directory permissions, where we analyzed and modified access rights for directories. Employing commands like ls -ld drafts to inspect permissions and chmod g-x drafts to remove execute permissions for the group on the drafts directory, we exercised precision in tailoring permissions to adhere to organizational security policies. Throughout this lab, we honed our skills in managing file and directory permissions, emphasizing the critical role they play in safeguarding sensitive data and ensuring only authorized access is granted, thus fortifying the overall security posture of the system.