Justin Goncalves

8/4/24

Linux File Permissions Management Activity

Activity Overview:

You will review a scenario and follow a series of steps. This scenario is connected to the lab 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 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.

Follow the instructions and fill in the sections to complete the activity.

> View the Current File Permissions for the Scenario:
>
> https://docs.google.com/document/d/1AGfcPzLeJcLXchAgEn5sG0mAZS0tmJLCN8GsOWfXKm0/edit#heading=h.dooa9fyvnog2

File permissions in Linux

Project description

In this project, I will ensure that file and directory permissions in the `/home/researcher2/projects` directory are appropriately set to match the authorization requirements for the research team. This involves examining the current permissions, understanding their implications, and modifying them as necessary to secure the system by authorizing the appropriate users and removing any unauthorized access.

Check file and directory details

To check the file and directory details, I will use the `ls -la` command in Linux. This command lists all files and directories in the current directory, including hidden ones, along with their details such as permissions, number of links, owner, group, size, and modification date.

```
ls -la /home/researcher2/projects
```

Explanation: The `ls -la` command displays a detailed list of all files and directories, including hidden files (those starting with a dot .), in the specified directory. The output includes the permissions, owner, group, size, and other attributes.

Example Output:

```
drwxr-x---  2 researcher2 researcher 4096 Jul 23 14:33 drafts
-rw-rw-rw-  1 researcher2 researcher  123 Jul 23 14:33 project_k.txt
-rw-r--r--  1 researcher2 researcher  123 Jul 23 14:33 project_m.txt
-rw-rw-r--  1 researcher2 researcher  123 Jul 23 14:33 project_r.txt
-rw-rw-r--  1 researcher2 researcher  123 Jul 23 14:33 project_t.txt
-rw-rw----  1 researcher2 researcher  123 Jul 23 14:33 .project_x.txt
```

Describe the permissions string

The permissions string in Linux is a 10-character string that represents the access permissions for a file or directory. The string is divided into four parts:

1. **File Type:** The first character indicates the file type. It can be:
    - `-` for a regular file
    - `d` for a directory
    - `l` for a symbolic link
    - `c` for a character device
    - `b` for a block device
2. **Owner Permissions:** The next three characters represent the permissions for the file owner:
    - `r` (read)
    - `w` (write)
    - `x` (execute)

3. **Group Permissions:** The following three characters represent the permissions for the group:
    - `r` (read)
    - `w` (write)
    - `x` (execute)
4. **Others Permissions:** The final three characters represent the permissions for others:
    - `r` (read)
    - `w` (write)
    - `x` (execute)

Example: The permissions string `drwxr-x---` indicates a directory (`d`) with the following permissions:

- Owner: read, write, execute (`rwx`)
- Group: read, execute (`r-x`)
- Others: no permissions (`---`)

Change file permissions

To change file permissions in Linux, the `chmod` command is used. This command allows you to modify the read, write, and execute permissions for the owner, group, and others.

Explanations:

- `chmod 660 /home/researcher2/projects/project_k.txt`: Sets read and write permissions for the owner and group, and no permissions for others.
- `chmod 640 /home/researcher2/projects/project_m.txt`: Sets read and write permissions for the owner, read permissions for the group, and no permissions for others.
- `chmod 664 /home/researcher2/projects/project_r.txt`: Sets read and write permissions for the owner and group, and read permissions for others.
- `chmod 664 /home/researcher2/projects/project_t.txt`: Sets read and write permissions for the owner and group, and read permissions for others.
- `chmod 620 /home/researcher2/projects/.project_x.txt`: Sets read and write permissions for the owner, write permissions for the group, and no permissions for others.

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 the following command to assign `.project_x.txt` the appropriate authorization.

```
chmod 640 /home/researcher2/projects/.project_x.txt
```

Explanation:

- `chmod 640 /home/researcher2/projects/.project_x.txt`: Sets read and write permissions for the owner, read permissions for the group, and no permissions for others.

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. To modify the permissions accordingly, use the following command:

Command: `chmod 700 /home/researcher2/projects/drafts`

Explanation**:** The `chmod 700 /home/researcher2/projects/drafts` command sets the permissions of the `drafts` directory so that only the owner (user `researcher2`) has read, write, and execute permissions. The group and others have no permissions. This ensures that only `researcher2` can access and modify the contents of the `drafts` directory.

Summary

In this project, I examined and modified file and directory permissions in the `/home/researcher2/projects` directory to ensure appropriate authorization for the research team. Using commands like `ls -la` to check current permissions and `chmod` to modify them, I set the necessary access levels for files and directories, ensuring that users have the required access while preventing unauthorized access. This process helps maintain the security and integrity of the organization's file system.
