## **File Permissions in Linux**

### **Project Description**

In this project, I managed file permissions in the /home/researcher2/projects directory to ensure compliance with the organization's security policies. This involved examining current permissions, updating them as needed to remove unauthorized access, and ensuring that files and directories had appropriate access rights. The objective was to secure the system by properly configuring file permissions, including those for hidden files and directories.

### **Check File and Directory Details**

To check the permissions of files and directories, including hidden ones, use the ls -la command. This command lists all files and directories with detailed information, including permissions.

**Command used:**

bash

ls -la /home/researcher2/projects

**Example output:**

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

-rw-rw-r-- 1 researcher2 research\_team 46 Oct 14 18:40 .project\_x.txt

**Explanation:**

* ls -la lists all files and directories, including hidden ones (those starting with a dot), and displays detailed information such as permissions, owner, group, size, and modification date.

### **Describe the Permissions String**

For the file -rw-r----- 1 researcher2 research\_team 46 Jul 31 01:58 project\_m.txt, the permissions string -rw-r----- is described as follows:

* The first character - indicates it is a regular file (not a directory).
* The next three characters rw- denote the permissions for the file owner: read (r) and write (w), but not execute (-).
* The following three characters r-- denote the permissions for the group: read (r) only, and no write (-) or execute (-) permissions.
* The last three characters --- denote the permissions for others: no read, write, or execute permissions.

### **Change File Permissions**

To ensure that no file allows write access for unauthorized users, the permissions for project\_k.txt were modified. The following command was used to remove write permissions for others:

**Command used:**

bash

Copiar código

1. chmod o-w /home/researcher2/projects/project\_k.txt

**Result:** The file project\_k.txt was updated to remove write permissions for "others". The updated permissions are -rw-rw-r--.

**Explanation:**

* chmod o-w removes write permissions for others. This ensures that the file is not writable by unauthorized users, aligning with security policies.

### **Change File Permissions on a Hidden File**

The hidden file .project\_x.txt should have read permissions for both the user and the group but no write permissions. The following command was used to update the permissions:

**Command used:**

bash

Copiar código

1. chmod 664 /home/researcher2/projects/.project\_x.txt

**Result:** The file .project\_x.txt now has permissions -rw-rw-r--, allowing the user and group to read and write, but "others" cannot write.

**Explanation:**

* chmod 664 sets permissions so that the user and group have read and write access, while others have read-only access. This configuration prevents unauthorized write access.

### **Change Directory Permissions**

To ensure that only researcher2 can access the drafts directory and its contents, the following command was used:

**Command used:**

bash

Copiar código

1. chmod 700 /home/researcher2/projects/drafts

**Result:** The drafts directory now has permissions drwx------, meaning only the user researcher2 has full access (read, write, and execute) to the directory.

**Explanation:**

* chmod 700 grants read, write, and execute permissions only to the user and removes all permissions for the group and others. This secures the directory from unauthorized access.

### **Summary**

In this project, I reviewed and adjusted file and directory permissions in the /home/researcher2/projects directory to meet security requirements. This included removing unauthorized write access, updating permissions on hidden files, and securing directories. These changes enhance the security of the file system by ensuring proper access control.

### **Conclusion**

Effective management of file permissions is essential for maintaining system security. By carefully reviewing and modifying permissions, we can protect sensitive information and prevent unauthorized access. This project highlights the importance of applying the principle of least privilege and ensuring that permissions are correctly configured to align with organizational policies.

### **Recommendations**

1. **Regular Audits:** Perform regular audits of file and directory permissions to ensure compliance with security policies and detect any unauthorized changes.
2. **Automation:** Use scripts or tools to automate permission management tasks, especially in large environments with numerous files and users.
3. **User Education:** Educate users about the significance of file permissions and the potential risks associated with incorrect configurations.
4. **Least Privilege Principle:** Always apply the principle of least privilege by granting only the permissions necessary for users to perform their tasks.
5. **Monitoring:** Implement monitoring and logging to track permission changes and access attempts, enabling quick response to potential security incidents.

## **Permisos de archivos en Linux (spanish)**

### **Descripción del Proyecto**

En este proyecto, he utilizado comandos de Linux para administrar permisos de archivos en el directorio /home/researcher2/projects. El objetivo principal fue asegurar que los permisos en los archivos y subdirectorios coincidan con las políticas de seguridad de la organización, garantizando que solo los usuarios autorizados tengan acceso adecuado a los recursos. Esto incluyó la revisión y modificación de permisos para asegurar que ningún archivo permitiera escritura por parte de usuarios no autorizados, y que los archivos ocultos y subdirectorios tuvieran los permisos adecuados para el usuario y el grupo.

### **Comprobar detalles de archivos y directorios**

Para verificar los permisos establecidos para los archivos y subdirectorios en el directorio projects, se usó el comando ls -l en la terminal. Este comando muestra una lista detallada de archivos y directorios, incluyendo sus permisos, propietarios y grupos.

**Comando utilizado:**

bash

Copiar código

ls -la /home/researcher2/projects

**Salida esperada:**

plaintext

Copiar código

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

-rw-rw-r-- 1 researcher2 research\_team 46 Oct 14 18:40 .project\_x.txt

### **Describe la cadena de permisos**

Para la entrada de ejemplo: -rw-r----- 1 researcher2 research\_team 46 Oct 14 18:40 project\_m.txt

La cadena de permisos -rw-r----- se desglosa de la siguiente manera:

* El primer carácter (-) indica que es un archivo regular (no un directorio).
* Los siguientes tres caracteres (rw-) representan los permisos para el usuario propietario: lectura (r) y escritura (w), pero no ejecución (-).
* Los siguientes tres caracteres (r--) representan los permisos para el grupo: solo lectura (r) y no escritura (-) ni ejecución (-).
* Los últimos tres caracteres (---) representan los permisos para otros usuarios: sin permisos de lectura, escritura ni ejecución.

### **Cambiar permisos de archivos**

Para cumplir con la política de seguridad que requiere que ningún archivo permita escritura por parte de usuarios no autorizados, se identificó que el archivo project\_k.txt tenía permisos incorrectos. Se usó el siguiente comando para modificar los permisos de este archivo:

**Comando utilizado:**

bash

Copiar código

chmod o-w /home/researcher2/projects/project\_k.txt

**Resultado:** Después de ejecutar el comando, el archivo project\_k.txt fue modificado para que no permitiera permisos de escritura para "otros" usuarios. Ahora los permisos se actualizan a -rw-rw-r--.

### **Cambiar los permisos de un archivo oculto**

El archivo oculto .project\_x.txt no debe tener permisos de escritura para nadie, pero tanto el usuario como el grupo deben poder leerlo. Se utilizó el siguiente comando para cambiar los permisos:

**Comando utilizado:**

bash

Copiar código

chmod 664 /home/researcher2/projects/.project\_x.txt

**Resultado:** El archivo .project\_x.txt se actualizó para que el usuario y el grupo pudieran leer y escribir, pero "otros" no tuvieron permisos de escritura. Los permisos resultantes son -rw-rw-r--.

### **Cambiar permisos de directorio**

El directorio drafts dentro de projects debe permitir acceso solo al usuario researcher2, sin permisos para el grupo ni otros usuarios. Se utilizó el siguiente comando para modificar los permisos:

**Comando utilizado:**

bash

Copiar código

chmod 700 /home/researcher2/projects/drafts

**Resultado:** El directorio drafts ahora tiene permisos drwx------, lo que significa que solo el usuario researcher2 tiene acceso completo (lectura, escritura y ejecución) al directorio.

### **Resumen**

En este proyecto, se revisaron y modificaron los permisos de archivos y directorios en el directorio /home/researcher2/projects para garantizar que se cumplieran las políticas de seguridad de la organización. Se aseguraron de que los archivos no permitieran escritura por parte de usuarios no autorizados, se ajustaron los permisos de un archivo oculto para cumplir con los requisitos de acceso, y se restringió el acceso a un directorio a un solo usuario. Estos cambios ayudan a proteger la integridad y la confidencialidad de los datos en el sistema de archivos.

### **Conclusión**

El proceso de administración de permisos de archivos en Linux es esencial para mantener la seguridad y la integridad de los datos en un sistema. Al revisar y ajustar los permisos, se asegura que solo los usuarios autorizados tengan acceso adecuado, y se previene el acceso no autorizado que podría comprometer la seguridad del sistema. Este ejercicio demuestra la importancia de una gestión adecuada de permisos para proteger la información y garantizar un entorno de trabajo seguro.

### **Recommendations**

1. **Revisión Periódica:** Realiza revisiones periódicas de los permisos de archivos y directorios para asegurarte de que continúen cumpliendo con las políticas de seguridad de la organización.
2. **Automatización de Permisos:** Considera la implementación de scripts o herramientas de automatización para gestionar permisos y auditar cambios de manera eficiente, especialmente en entornos con muchos archivos y usuarios.
3. **Capacitación:** Proporciona capacitación a los usuarios sobre la importancia de los permisos de archivos y cómo pueden afectar la seguridad del sistema, para que estén al tanto de las mejores prácticas.
4. **Uso de Permisos Mínimos:** Aplica el principio de mínimo privilegio, otorgando solo los permisos necesarios para que los usuarios realicen sus tareas. Esto reduce el riesgo de acceso no autorizado o accidental a datos sensibles.
5. **Monitoreo y Auditoría:** Implementa sistemas de monitoreo y auditoría para detectar cambios en los permisos y posibles intentos de acceso no autorizado, permitiendo una respuesta rápida a incidentes de seguridad.