PRÁCTICA 4.7

Recursos Locales. Gestión de permisos

*Para hacer esta práctica partimos de la práctica en la que hemos creado los usuarios **usu1** y **usu2** que pertenecen al grupo **programadores** y **usu4** y **usu5** que pertenecen al grupo **testers** y el usuario **admin0** que tiene privilegios administrativos.*

1. Inicia sesión con el usuario **admin0** y crea el directorio **/home/Proyectos**. Indica que permisos tiene por defecto ese directorio.

2. Cambia el propietario y el grupo del directorio **Proyectos** para que pertenezca al usuario **admin0** y al grupo **programadores**.

3. Crea con **admin0** dentro de **Proyectos** la siguiente estructura de directorios sin usar sudo:

   /home/Proyectos/

   Proyecto1/

   Proyecto2/

   Mensajes/

4. Haz las acciones necesarias para que con permisos **UGO** en el directorio **Proyecto1** el usuario **usu1** tenga control total (lectura, escritura y ejecución), y usu4 y usu5 solo puedan listar y leer el contenido del directorio, el resto de usuarios del sistema no podrán entrar. Modifica los permisos en **octal**. Prueba a entrar con **usu1** y cea un fichero dentro llamado **tareas.txt**. ¿Que permisos tiene el nuevo fichero creado?

5. Haz las acciones necesarias para que en el directorio **Proyecto2,** con permisos **UGO,** los usuarios **usu2** y **usu5** tengan acceso total (lectura, escritura y ejecución) y el resto de usuarios no podrán acceder.

6. Configura el directorio **Mensajes** para que todos los usuarios puedan acceder con control total, pero no podrán borrar lo que no les pertenezca. Con **admin0** crea dentro de **Mensajes** un fichero llamado **avisos_importantes.txt** en el que todos podrán leer y escribir. Entra con **usu1** y guarda un mensaje dentro del fichero, luego entra como **usu5**, lee el contenido e intenta borrar el fichero.

7. Con **admin0** crea otro fichero dentro de **Mensajes** llamado **avisos_privados.txt.** Modifica sus permisos **UGO** para que todo el mundo pueda escribir pero no leer su contenido. Solo podrá leer el contenido **admin0** y los miembros del grupo **testers.** Prueba escribir algo dentro del fichero con **usu2** con "echo 'El usu1 no documenta sus programas' >> avisos_privados.txt". Prueba a leer el contenido con **usu4** y con **usu1**.

8. Entra al sistema con **usu1** y cambia la mascara por defecto para que los nuevos ficheros creados en esta sesión tengan permisos **r w _ r _ _ _ _ _.** Prueba que funciona creando un fichero llamado **tareas2.txt** con el comando **touch** dentro de **Proyecto1.**

9. Con **admin0** crea un fichero dentro de **Proyectos** llamado **debug.sh**, modifica los permisos usando **ACLs** para que **admin0** tenga control total, el grupo **testers** podrá leer y ejecutar, menos **usu5** que solo lo podrá leer, el grupo **programadores** podrá leer y escribir (no ejecutar). El resto de usuarios del sistema no podrán hacer nada con este fichero. Realiza una captura de pantalla donde aparezca la ACL del fichero **debug.sh.**

10. ¿Como aparecen y que significan ahora los permisos clásicos del fichero **debug.sh**?

11. Con **admin0** elimina de la ACL del fichero **debug.sh** al usuario **usu5,** ¿qué podrá hacer ahora el usuario usu5?.

12. Cambia la máscara de la acl del fichero **debug.sh** como solo lectura, lista la acl e indica como se muestra el permiso final efectivo. ¿Para qué sirve la máscara en las acls?

13. Instala **eiciel** en el sistema y modifica con **eiciel** la ACL del fichero "debug.sh" estableciendo la máscara con rwx y añade al usuario **usu2** con permiso solo de escritura. Inicia sesión como usu2 ¿Puede leer el contenido del fichero por pertenecer al grupo programadores? Realiza una captura de pantalla de la ACL.

Nota: *La práctica se entregará en el aula virtual en formato pdf indicando en el fichero el número de práctica y tu nombre con el siguiente formato: prácticaX.X_nombre.pdf*