Protección y Permisos en Sistemas Unix

20 de febrero de 2025

Objetivos de la Presentación

- Comprender los atributos de protección de procesos y ficheros en Unix.
- Analizar el significado de los bits de permiso y su aplicación en ficheros y directorios.
- Conocer las reglas de protección básicas que determinan el acceso a recursos.
- Explicar el funcionamiento de los bits especiales: SETUID, SETGID y sticky bit.
- Revisar la influencia de la máscara de creación de ficheros (umask) en los permisos.
- Conocer las herramientas para modificar atributos: `chown`, `chgrp` y `chmod`.

Atributos de Protección de Procesos Unix

- Cada proceso posee dos identificadores de usuario:
  - **rUID:** Usuario real (quien creó el proceso).
  - **eUID:** Usuario efectivo (utilizado para la verificación de permisos).
- De forma similar, existen dos identificadores de grupo:
  - **rGID:** Grupo real (grupo primario del creador).
  - **eGID:** Grupo efectivo (utilizado para la comprobación de permisos).
- Además, el proceso hereda una lista de grupos suplementarios del usuario.
- El comando `id` permite visualizar todos estos atributos.

Atributos de Protección de Ficheros Unix

- Cada fichero tiene asociados:
  - **ownerUID:** Identificador del usuario propietario.
  - **ownerGID:** Identificador del grupo propietario.
  - **Bits de permiso:** 12 bits que incluyen 9 bits de permisos básicos (propietario, grupo, otros) y 3 bits especiales (SETUID, SETGID y sticky).
- Estos atributos se asignan en el momento de creación (almacenados en el inodo) y, en ciertos casos, pueden heredarse (por ejemplo, el bit SETGID en directorios).

Significado de los Bits de Permiso

- Para ficheros regulares:
  - **r:** Permite leer el contenido.
  - **w:** Permite modificar el fichero.
  - **x:** Permite ejecutar el fichero.
- Para directorios:
  - **r:** Permite listar el contenido.
  - **w:** Permite crear, eliminar o renombrar ficheros dentro del directorio.
  - **x:** Permite entrar en el directorio (acceder a sus ficheros mediante un camino).
- El permiso para borrar un fichero se controla por los permisos del directorio que lo contiene.

Reglas de Protección Básicas Unix

- El sistema verifica los permisos en el siguiente orden:
  1. Si el proceso tiene **eUID = 0** (root), se concede el permiso sin restricciones.
  2. Si el **eUID** coincide con el **ownerUID** del fichero, se utilizan los permisos asignados al propietario.
  3. Si el **eGID** o alguno de los grupos suplementarios coincide con el **ownerGID**, se aplican los permisos del grupo.
  4. En caso contrario, se utilizan los permisos asignados a otros (others).
- Sólo se aplica la primera condición que se cumpla.

SETUID y SETGID en Ejecutables

- **SETUID:** Permite que un programa se ejecute con los privilegios del propietario del fichero en lugar de los del usuario que lo lanza.
  - Se representa con una `s` en lugar de la `x` en la categoría del propietario (por ejemplo, `rws`).
- **SETGID:** Permite que un programa se ejecute con los privilegios del grupo propietario.
  - En directorios, asegura que los ficheros creados hereden el grupo del directorio.

Sticky Bit en Directorios

- El **sticky bit** se utiliza en directorios para que solo el propietario de un fichero (o root) pueda eliminarlo, aun cuando otros tengan permisos de escritura en el directorio.
- Se muestra como una `t` en la categoría de "others".
- **Ejemplo típico:** el directorio `/tmp` suele tener permisos `drwxrwxrwt`.

La Máscara de Creación de Ficheros (umask)

- Al crear un fichero, el sistema asigna permisos máximos por defecto (por ejemplo, `rw-rw-rw-` para ficheros y `rwxrwxrwx` para directorios).
- La **umask** desactiva (quita) ciertos permisos de estos valores por defecto.
- **Ejemplo:** Una umask de 022 desactiva los permisos de escritura para grupo y otros:

  ```
  umask 022
  ```

- Resultado típico: ficheros con permisos `rw-r-r-` y directorios con `rwxr-xr-x`.

Resumen de los Bits Especiales

- **SETUID:** Ejecuta programas con los privilegios del propietario.
- **SETGID:** Ejecuta programas con los privilegios del grupo; en directorios, los nuevos ficheros heredan el grupo.
- **Sticky Bit:** Restringe la eliminación de ficheros en un directorio, permitiendo que sólo el propietario o root pueda borrarlos.

Herramientas para Modificar Atributos

- **chown:** Cambia el propietario de un fichero.

  `chown usuario archivo`

- **chgrp:** Cambia el grupo propietario de un fichero.

  `chgrp grupo archivo`

- **chmod:** Modifica los bits de permiso.

  `chmod 755 archivo`

Ejemplos Prácticos de Permisos Especiales

- **SETUID en un ejecutable:** Hacer que `/usr/bin/passwd` se ejecute con los privilegios del propietario (root):

  ```
  chmod 4755 /usr/bin/passwd
  ```

- **SETGID en un directorio:** Asegurar que los ficheros creados en el directorio `proyectos` hereden el grupo del directorio:

  ```
  chmod 2755 proyectos
  ```

- **Sticky Bit en un directorio:** Configurar `/tmp` para que solo el propietario de un fichero pueda borrarlo:

  ```
  chmod 1777 /tmp
  ```