Linux

Usuarios, Grupos y Permisos

Gustavo C. Distel
gd@cs.uns.edu.ar
D.C.I.C. – U.N.S.

Usuarios

- Antes de utilizar un sistema Linux nos debemos identificar con un *nombre de usuario*.
- El *nombre de usuario* nos representa dentro del sistema.
- Está asociado con:
  - **Lo que hacemos**: Cada proceso tiene asociado un nombre de usuario.
  - **Lo que creamos**: Cada archivo en el sistema es propiedad de un usuario.
  - **Lo que usamos**: Espacio en disco, uso del procesador, etc.

Usuarios

- Cada usuario en el sistema no sólo tiene un *nombre de usuario* único, sino también un *userid*, a menudo abreviado como *uid*.
  - **Nota**: En la distribución Fedora el mínimo uid es 1000, o dicho de otra manera, la numeración automática al crear los usuarios comienza en 1000 y está definido en el archivo /etc/login.defs
- El sistema mantiene una base de datos que asigna los *nombres de usuarios* a los *userids*. Esta base de datos se almacena en el archivo de configuración */etc/passwd*.
- Además, se utiliza el archivo /etc/shadow para almacenar el password y otra información relacionada al estado del mismo.

Usuario root

- En sistemas operativos del tipo Unix como Linux, root (o superusuario) es el nombre convencional de la cuenta de usuario que posee todos los derechos en todos los modos (mono o multi usuario).
  - El uid de root es 0.
  - **Nota**: En los sistemas Windows también existe, es el usuario administrador.
- El usuario root puede hacer muchas cosas que un usuario común no puede, tales como cambiar el dueño o permisos de los archivos.

- **Advertencia**: No es recomendable utilizar el usuario root para una simple sesión de uso cotidiano, ya que pone en riesgo al sistema al garantizar acceso privilegiado a cada programa en ejecución. Es preferible utilizar una cuenta de usuario normal.

Grupos

- Los usuarios, los procesos que estos operan y los archivos que poseen pertenecen a una colección de *grupos*.
- Las membresías de grupo le permiten a los administradores de sistemas manejar eficientemente las *colecciones de los usuarios que tienen objetivos similares*.
- Para el kernel de Linux, un grupo es nombrado usando un id de grupo (*GID*) (ídem usuarios).
- El archivo */etc/group* asocia nombres de grupo con GIDs y define los usuarios que pertenecen a cada grupo.
- Además, se utiliza el archivo /etc/gshadow

Comandos

- whoami: Who am I? - *Quién soy, Imprime el nombre de usuario asociado con el ID de usuario efectivo actual.*
- logname: Login Name - *Muestra el nombre de inicio de sesión del usuario.*
- id: Identification - *Identificación*.
  - Imprime UIDs (Número de identificación de usuario) y GIDs (Número de identificación de grupo) reales y efectivos. Ej.: id, id <usuario>
- useradd: User Add - *Agrega Usuario*.
  - **Nota**: Añade una entrada en los archivos: /etc/passwd, /etc/group y /etc/shadow.
- passwd: Password - *Cambia el password del usuario*.
- userdel: User Delete - *Elimina usuario.*
  - userdel -rf <usuario>
- usermod: User Modification - *Modifica cuenta de usuario*.
  - usermod –c '[Programador Web]' <usuario>
- chage: Change - *Cambia información de caducidad de la contraseña de usuario.*

Comandos

- groupadd: *Agrega Grupo.*
  - Cuando se añade un nuevo grupo, una entrada correspondiente se crea en los archivos: /etc/group y /etc/gshadow.
  - Ej.: groupadd developers, luego agregamos el usuario: usermod -a -G developers <usuario>
- groupdel: *Elimina Grupo.*
- groups: *Muestra los grupos de un usuario.*
  - groups <usuario>
- chwon: change owner - *cambia propietario de usuario o grupo*.
  - Nota: utilizando un signo de dos puntos (:) o bien un punto (.), sin espacios entre ellos, se cambia el usuario y grupo al que pertenece cada fichero.
  - **chown admin /etc/hosts** → el nuevo propietario del archivo sería admin.
  - **chown gdistel:gdistel /etc/hosts** → el nuevo propietario será gdistel perteneciente al grupo gdistel.
- su: *Ejecuta una shell como otro usuario.*
- sudo: *Ejecuta un comando como otro usuario.*

Permisos

- Existen tres clases de accesos:
  - Usuario (u: user) - Los archivos son propiedad de un usuario.
  - Grupo (g: group) - Los archivos son asignados a un grupo.
  - Otros (o: others) - Usuarios que no pertenecen a las dos clases anteriores.

- Existen tres tipos de permisos:
  - Lectura (r: read) - Ver el archivo.
  - Escritura (w: write) - Modificar el archivo.
  - Ejecución (x: execute) - Usar el archivo como comando, ejecutarlo.

- Entonces, cada archivo tiene permisos de lectura(r), escritura(w), y ejecución (X) para las tres clases diferentes de acceso de archivo: usuario(u), grupo(g) y otros(o).

Acceso

- Cuando un usuario trata de acceder a un archivo, Linux hace las siguientes preguntas en este orden:

    1. ¿El usuario es propietario del archivo? Si lo es, entonces utiliza los permisos de usuario.
    2. ¿El usuario es miembro del grupo dueño del archivo? Si lo es, entonces se utilizan los permisos de grupo.
    3. De lo contrario, se utilizan los otros permisos.

- Como veremos más adelante, los bits de permisos definen la forma de cómo las tres clases diferentes de usuarios pueden usar el archivo.

Notación simbólica

- Esta representación aparece al ejecutar el comando ls -l, ej.:
  - `-rw-rw-r--. 1 gdistel gdistel 2 Aug 25 17:55 prueba1.txt`
- El primer carácter indica el tipo de archivo y no está relacionado con los permisos, ej.:
  - '-': archivo ordinario.
  - 'd': directorio.
  - 'l': link simbólico.
- Los restantes nueve caracteres se dividen en tres conjuntos. Cada uno representa una clase de permisos con tres caracteres. El primer conjunto representa la clase usuario, el segundo grupo representa la clase grupo y el tercer conjunto representa la clase otros. Para el ejemplo anterior:
  - **rw-** **rw-** **r--**

- Una letra indica que el permiso correspondiente se ha activado, mientras que si aparece un '-'(guión) significa que no se tiene permiso.

Notación numérica

- Al ejecutar el comando stat de la siguiente manera obtenemos los permisos en octal:
  - stat -c %a prueba1.txt
- Para cada clase de usuario (dueño, grupo, otros) tenemos:
  - Tres permisos con dos posibilidades cada uno de ellos.
    - **(r|-) (w|-) (x|-)**; está o no concedido el permiso.
    - obtenemos $2^3 = 8$ permisos en total que pueden ser asignados o denegados.
    - Por lo anterior se utiliza el sistema numérico en base 8 (octal), que utiliza los dígitos del 0 a 7.

Notación numérica

Número	Binario	Lectura (*r*)	Escritura (*w*)	Ejecución (*x*)
0	000	✗	✗	✗
1	001	✗	✗	✓
2	010	✗	✓	✗
3	011	✗	✓	✓
4	100	✓	✗	✗
5	101	✓	✗	✓
6	110	✓	✓	✗
7	111	✓	✓	✓

Notación numérica

- Luego existe un dígito octal por cada combinación de permisos. Es decir, un dígito octal para el usuario propietario, otro para el grupo y otro para otros.

- Así, las posibles combinaciones se resumen en números octales de tres dígitos del 000 al 777, cada uno de los cuales permite establecer un tipo de permiso distinto a cada clase de usuario.

- El primer dígito establece el tipo de permiso otorgado al dueño, el segundo al grupo y el tercero al resto de los usuarios.

comando chmod

- chmod: change mode - *cambiar modo, Permite cambiar los permisos de acceso de un archivo o directorio.*
  - Existen 2 formas o modos de asignar los permisos a los usuarios:
    - Modo carácter.
    - Modo octal.
- chmod: Modo carácter
  - Posee 3 modificadores que permiten realizar la tarea:
    - '+' añade un modo.
    - '–' elimina un modo.
    - '=' especifica un modo (sobrescribiendo el modo anterior).
  - Clases de accesos que se pueden utilizar:
    - 'u' dueño del archivo o directorio.
    - 'g' grupo al que pertenece el archivo.
    - 'o' todos los demás usuarios que no son el dueño ni del grupo, otros.
    - 'a' incluye al dueño, al grupo y a otros (todos).

chmod: Modo carácter, ejemplos

- **`chmod +r prueba.txt`**: agrega permisos de lectura a todos los usuarios.
- **`chmod u+w prueba.txt`**: agrega permisos de escritura al dueño.
- **`chmod –x prueba.txt`**: elimina el permiso de ejecución a todos los usuarios.
- **`chmod u=rw,go= prueba.txt`**: establece los permisos de lectura y escritura al dueño y elimina todos los permisos a los demás usuarios.
- **`chmod o-r prueba2`**: suprime el permiso de lectura a otros.
- **`chmod g-w prueba2`**: suprime el permiso de escritura para el grupo.
- **`chmod ug+x prueba2`**: agrega permiso de ejecución al usuario y al grupo.
- **`chmod o+w prueba2`**: agrega permiso de escritura a otros.
- **`chmod a-w prueba2`**: suprime el permiso de escritura a todos.

chmod: Modo octal, ejemplos

- **`chmod 766 file.txt`**: brinda acceso total al dueño y lectura y escritura a los demás.
- **`chmod 770 file.txt`**: brinda acceso total al dueño y al grupo y elimina todos los permisos a los demás usuarios.
- **`chmod 635 file.txt`**: permite lectura y escritura al dueño, escritura y ejecución al grupo y lectura y ejecución al resto

Permisos adicionales

- Permiso set user ID, setuid o SUID: Cuando a un archivo ejecutable se le ha dado el atributo setuid, los usuarios normales en el sistema que tienen permiso para ejecutar el archivo obtienen los privilegios del usuario que posee el archivo (normalmente root) al momento de crear el proceso.
- El ejemplo típico es el cambio de una clave de usuario: ningún usuario debería poder modificar /etc/passwd directamente.
- La única forma de poder modificarlo debería ser a través del comando correspondiente, que necesariamente tendrá que tener asignado el setuid. Es decir, el comando /usr/bin/passwd ejecutado por un usuario se ejecutará como si lo hubiese invocado el superusuario, de manera de poder modificar /etc/passwd.

Permisos adicionales

- Permiso set group ID, setgid o SGID: cuando un archivo que tiene este permiso asignado se ejecuta, el proceso resultante asumirá la ID de grupo efectiva dada a la clase de grupo.

- Cuando el setgid le es asignado a un directorio, archivos nuevos y directorios creados debajo de ese directorio heredarán el grupo de ese directorio. Esto se diferencia del comportamiento por defecto, que usa el grupo primario del usuario efectivo al asignar el grupo de archivos nuevos y directorios.

Permisos adicionales

- Permiso de sticky bit (o menos común, bit pegadizo): el comportamiento típico del sticky bit en archivos ejecutables fuerza al kernel a retener la imagen del proceso resultante luego de su terminación.

- Originalmente esta era una característica para ahorrar memoria; pero hoy en día los precios de las memorias han disminuido y existen mejores técnicas para manejarlo, así que no se lo suele utilizar para optimizaciones en archivos.

- En un directorio, por el contrario, el sticky bit previene que los usuarios renombren, muevan o borren los archivos que allí se encuentran, incluso si tienen permiso de escritura en el directorio. Solo el propietario del directorio y el superusuario quedan exentos de esto.

Ejercicios adicionales y opcionales

1. Crear los siguiente usuarios desde una consola Linux: john, paul, george, ringo, mick, keith, ronnie, charlie.
2. Asignar un password a cada usuario.
3. Crear los siguiente grupos: beatles y rolling.
4. Asignar los usuarios john, paul, george, ringo al grupo beatles.
5. Asignar los usuarios mick, keith, ronnie, charlie al grupo rolling.
6. Crear una carpeta en /home que se llame songs y que la puedan leer y escribir todos los usuarios del sistema.
7. Dentro de la carpeta songs crear con el usuario mick un archivo que se llame satisfaction, que lo puedan leer y escribir mick y los usuarios del grupo rolling.
8. Dentro de la carpeta songs crear un archivo con el usuario keith que se llame wildhorses, que lo pueda leer y escribir keith y nadie más.
9. Dentro de la carpeta songs crear un archivo con el usuario por defecto que instaló el sistema, que se llame route66 y que lo puedan leer y escribir todos.
10. Dentro de la carpeta songs crear un archivo con el usuario john que se llame help, que lo pueda leer y escribir john y lo puedan leer los usuarios del grupo beatles y nadie más.
11. En el directorio / utilice el comando find para encontrar los archivos anteriormente creados.

Referencias

- Manual de Linux desde la consola de CentOS y Fedora.
- www.centos.org.
- www.fedoraproject.org.
- www.linuxfoundation.org.
- www.redhat.com.
- www.wikipedia.org.
- www.oreilly.com.
- http://en.wikipedia.org/wiki/Filesystem_permissions