GESTIÓN DE USUARIOS Y GRUPOS

Laboratorio de Sistemas Operativos II

- ★ *GNU/Linux es un sistema operativo **multiusuario** en red que provee servicio y procesamiento a múltiples usuarios que comparten los mismos recursos simultáneamente.*
- ★ *Es necesaria la gestión de usuarios y grupos de usuarios para garantizar que los recursos del sistema sean utilizados de forma correcta, segura y rastreable*

Gestión de Usuarios y Grupos

- Es definir un perfil de privilegios para quienes operan con un sistema multiusuario. Incluye crear, modificar privilegios, asignar permisos y eliminar usuarios.
- Debe acompañar las políticas de seguridad que se establecen para garantizar la integridad y funcionalidad del sistema.

Importancia de la Gestión de Usuarios y Grupos

- **Seguridad del Sistema**

  Controlar el acceso y las operaciones de cada usuario a fin de prevenir accesos no autorizados, escaladas de privilegios y brechas de datos. Los permisos deben ser los estrictamente necesarios para su función.

- **Auditoría y Trazabilidad**

  Rastrear las acciones realizadas por cada usuario para detectar comportamientos sospechosos, diagnosticar problemas y cumplir con normativas de seguridad. En /var/log/auth.log se documentan las autenticaciones.

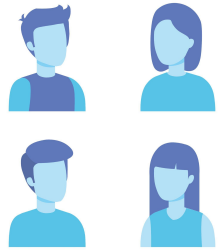
- **Organización y Permisos**

  Asignar de manera eficiente los recursos del sistema. Los grupos de usuarios facilitan la administración colectiva de permisos.

- **Cumplimiento Normativo**

  Cumplir con políticas de seguridad organizacionales. Es fundamental para demostrar cumplimiento ante auditorías internas y externas.

Usuarios



- Pueden corresponder a personas o programas que hacen uso del sistema.
- Cada usuario del sistema tiene una cuenta asignada.
- Las cuentas se usan para identificar en forma única a los usuarios del sistema.

Cuenta de usuario

- Conjunto de información que indica al S.O. el tipo de usuario, los accesos permitidos, los privilegios y permisos en general.
- Se identifica con el nombre de usuario y contraseña
- Los usuarios se comunican con el sistema a través de una “interfaz de usuario”
- Proceso de conexión al sistema o **login**
- Proceso de desconexión al sistema o **logout**

Usuarios y grupos en GNU/Linux

- Un **usuario** en GNU/Linux siempre está asociado a un grupo de usuarios.
- Los **grupos** son expresiones lógicas que unen usuarios bajo un propósito común.
- Un usuario puede pertenecer a varios grupos, además de su principal.
- Grupo principal: Es el asociado a la cuenta por defecto, y al que se le asignan los archivos y directorios creados por el usuario. Suele ser el mismo nombre de usuario.
- Grupos secundarios: los grupos suplementarios son aquellos grupos adicionales a los que el usuario pertenezca.

Tipos de usuarios en GNU/Linux

root

Superusuario o administrador del sistema operativo

Especiales o de servicio

Corresponden a cuentas del sistema operativo

Normales

Corresponden a personas físicas

root

- UID = 0
- Tiene privilegios sobre todo el sistema.
- Acceso total a todos los archivos y directorios
- Gestiona cuentas de usuarios
- Ejecuta tareas de mantenimiento
- Puede detener el sistema
- Puede instalar software
- Puede modificar el kernel

Usuarios especiales

- UID entre 1 y 999 (en /etc/login.defs)
- Ejemplos: bin, daemon, adm, lp, sync, shutdown, mail, squid, apache, etc.
- Asumen distintos privilegios de root, no todos para proteger.
- No tienen contraseñas, no están diseñadas para iniciar sesiones (nologin).
- Se crean durante la instalación del sistema operativo o de aplicaciones.

Usuarios normales

- UID desde 1000 (en /etc/login.defs)
- Se usan para usuarios individuales.
- Cada usuario dispone de un directorio de trabajo, ubicado generalmente en /home.
- Cada usuario puede personalizar su entorno de trabajo.
- Tienen solo privilegios completos en su directorio de trabajo o HOME.

```
usuario@server:~$
```

Archivos de configuración

- /etc/passwd
- /etc/group
- /etc/shadow
- /etc/login.defs
- /etc/skel

/etc/passwd

- **Todos** los usuarios del sistema operativo tienen su cuenta definida en este archivo.
- Es de texto tipo ASCII y se crea durante la instalación (root, cuentas especiales y usuario normal indicado).
- Contiene una línea para cada usuario y se actualiza a medida que se crean, modifican o eliminan usuarios.
- Formato: 7 campos separados por el símbolo **:**

```
root:x:0:0:root:/root:/bin/bash
sergio:x:501:500:Sergio González:/home/sergio:/bin/bash
```

Campos de /etc/passwd

Campo 1	Es el nombre del usuario, identificador de inicio de sesión (login). Único.
Campo 2	La 'x' indica la contraseña encriptada del usuario, además también indica que se está haciendo uso del archivo /etc/shadow, si no se hace uso de este archivo, este campo se vería algo así como: 'ghy675gjuXCc12r5gt78uuu6R'.
Campo 3	Número de identificación del usuario (UID). único.
Campo 4	Numeración de identificación del grupo (GID). Puede repetirse.
Campo 5	Comentarios o el nombre completo del usuario.
Campo 6	Directorio de trabajo (Home) donde se sitúa al usuario después del login.
Campo 7	Shell que va a utilizar el usuario de forma predeterminada.

/etc/group

- Es un archivo de texto ASCII que guarda información de los grupos a los que pertenecen los usuarios del sistema.
- Contiene una línea para cada usuario con tres o cuatro campos por usuario:

```
nombre_grupo:contraseña:GID:lista_usuarios
```

Campos de /etc/group

Campo 1	Nombre del grupo
Campo 2	Contraseña: del grupo (encriptada). Si este campo está vacío, no se utiliza ninguna contraseña.
Campo 3	Número de identificación del grupo (GID). único.
Campo 4	Lista de nombres de usuarios de todos los miembros del grupo, separados por comas, excepto aquellos para quienes este es el grupo primario.

/etc/shadow

- Antes las contraseñas cifradas se almacenaban en el mismo /etc/passwd. El problema es que 'passwd' es un archivo que puede ser leído por cualquier usuario del sistema, aunque solo puede ser modificado por root.
- El archivo 'shadow' fue creado para resolver el problema, ya que solo puede ser leído y modificado por root.
- Almacena la contraseña encriptada de cada usuario y tiene otros campos de control de contraseñas.

Campos de /etc/shadow

```
root:$6$RSX9ggySG4XhWqS35kkjDQF7p.HhN0:17399:0:99999:7:::
prueba:$6$3ZXePzr2U9z7KGWgMwCD7OZMSDHzb1:18385:0:15:7::18503:
```

Campo 1	Nombre de la cuenta del usuario.
Campo 2	Contraseña cifrada o encriptada, un '*' indica cuenta de 'nologin'. Si el valor es !, la cuenta está bloqueada.
Campo 3	La fecha del último cambio de contraseña, como número de días desde 01/01/1970. Un valor de 0 significa que el usuario debe cambiar la contraseña en el siguiente acceso.
Campo 4	Número mínimo de días que deben transcurrir hasta que la contraseña se pueda volver a cambiar.
Campo 5	Número máximo de días tras los cuales hay que cambiar la contraseña. (-1 significa nunca). A partir de este dato se obtiene la fecha de expiración de la contraseña.
Campo 6	Número de días antes de la expiración de la contraseña en que se le avisará al usuario al inicio de la sesión.
Campo 7	Días después de la expiración en que la contraseña se inhabilitará, si es que no se cambió.
Campo 8	La fecha, como número de días desde el 01/01/1970 en que se deshabilitará la cuenta de usuario. Un campo vacío significa que la cuenta de usuario nunca caducará.
Campo 9	Reservado para uso futuro

/etc/login.defs

- Contiene los valores de las variables que controlan los aspectos de la creación de usuarios y de los campos de shadow usados por defecto.
  - PASS_MAX_DAYS: Número máximo de días que una contraseña es válida
  - PASS_MIN_DAYS: Número mínimo de días que una contraseña es válida
  - PASS_MIN_LEN: Número mínimo de caracteres en la contraseña
  - UMASK: Determina la máscara de permisos. Por defecto es 007
  - CREATE_HOME: Indica si se crea el directorio personal por defecto
- Lo definido se usará al momento de crear o modificar usuarios.

/etc/skel

- El directorio /etc/skel (skeleton o esqueleto) es un directorio que se utiliza para definir los archivos de configuración **predeterminados que se copian automáticamente al directorio de trabajo de un nuevo usuario al ser creado**. Sirve como plantilla para asegurar que todos los usuarios tengan configuraciones iniciales idénticas.
- Al ejecutar comandos como useradd o adduser, el sistema copia los archivos ocultos (como .bashrc, .bash_profile, .profile) desde /etc/skel a la carpeta del nuevo usuario.
- El administrador puede agregar directorios o archivos en /etc/skel para que cada nuevo usuario lo tenga al iniciar sesión.
- Su uso garantiza políticas de seguridad o configuraciones de entorno consistentes para todos los usuarios porque actúa como una plantilla automática al momento de la creación de usuarios (todos los usuarios nuevos comienzan con la misma configuración inicial).

USUARIOS Y GRUPOS

Herramientas para gestionarlos

Comando adduser (ver man)

El comando adduser es el método recomendado en Debian para crear usuarios de forma guiada. Al ejecutarlo, el sistema solicita paso a paso los datos necesarios: contraseña, nombre completo, número de teléfono y otros campos opcionales. También crea automáticamente el directorio de trabajo (/home/usuario) y copia los archivos de configuración desde /etc/skel.

★ Crear un usuario nuevo	`#adduser nuevo_usuario`
★ Agregar un usuario existente a un grupo existente	`#adduser usuario grupo`
★ Agregar usuario nuevo con UID = 1050	`#adduser nuevo_usuario --uid nro_id`
★ Otorgar privilegios de root a un usuario existente	`#adduser usuario sudo`

Comando usermod

★ Cambiar el directorio personal de un usuario y mover su contenido	`#usermod -d nuevodir -m usuario`
★ Cambiar grupo principal de un usuario	`#usermod -g nuevo_grupo usuario`
★ Cambiar nombre de usuario/login	`#usermod -l nuevo_nombre usuario`
★ Bloquear la cuenta de un usuario	`#usermod -L usuario`

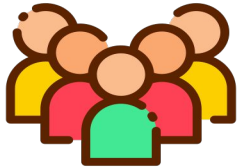
Otras operaciones con usuarios

★ Ver información de usuarios	`#id usuario`
	`#lslogins usuario`
	`#finger usuario` (instalar finger)
★ Eliminar un usuario	`#userdel usuario`
★ Eliminar un usuario y sus archivos asociados	`#userdel -rf usuario`

Gestión de claves de usuario

★ Modificar/asignar clave a un usuario (¡cuidado!)	`#passwd usuario`
★ Ver información de claves de un usuario	`#chage -l usuario`
★ Cambiar fecha de expiración de clave de un usuario	`#chage -E AAAA-MM-DD usuario`
★ Obligar a cambiar la clave en el próximo login	`#chage -d 0 usuario`

Gestión de grupos

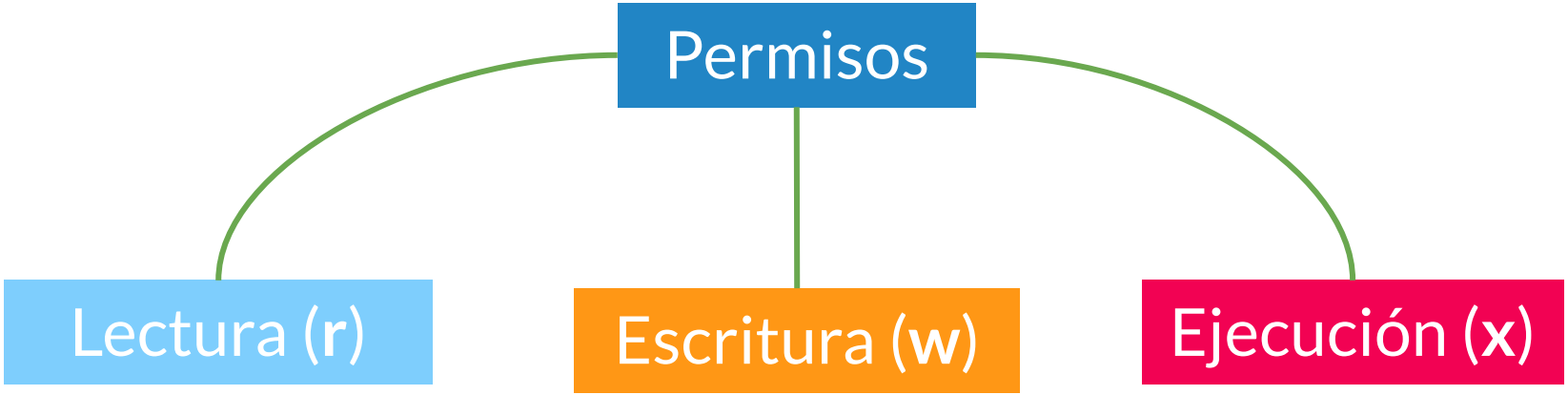


★ Crear un grupo nuevo	`#addgroup nuevo_grupo`
★ Eliminar un grupo	`#groupdel grupo`
★ Modificar atributos de grupos, por ejemplo cambiar su nombre	`#groupmod -n nuevo_grupo grupo`

PERMISOS

Herramientas para gestionarlos

Permisos en GNU/Linux



- **Archivo:** leer contenido
- **Directorio:** listar contenido

- **Archivo:** escribir y modificar
- **Directorio:** crear, borrar archivos

- **Archivos:** ejecutar como programa
- **Directorio:** ubicarse en directorio

VER PERMISOS: #ls -l

```
root@nuna:~# ls -l /etc/apt
drwxr-xr-x 2 root root  4096 abr 14  2019 apt.conf.d
-rw-r--r-- 1 root root   104 jun 26  2017 listchanges.conf
drwxr-xr-x 2 root root  4096 jun  1  2017 preferences.d
```

Tipo de archivo
d: directorio
- : artchivo
l : enlace

Permisos de usuario o dueño. Un guión es la ausencia del permiso.

Permisos de grupo. Un guión es la ausencia del permiso.

Permisos de otros usuarios. No incluye dueño ni grupo. Un guión es la ausencia del permiso.

Gestión de Permisos con chmod

- Sintaxis

  #chmod a-quien operación permisos archivos

- Ejemplos:
  - Añade a usuario dueño y grupo el permiso de lectura sobre el archivo carta

    ```
    #chmod ug+r /home/alumno/carta
    ```

  - Añade permiso de ejecución a todos los usuarios del sistema (a= all)

    ```
    #chmod a+x /opt/programa.sh
    ```

Chmod en formato octal

- Valor de permisos

  `Lectura = 4  // Escritura = 2  // Ejecución = 1`

- Ejemplos:
  - Asigna a **u**suario dueño y **g**rupo el permiso de lectura y elimina todos los permisos de los **o**tros usuarios sobre el archivo carta

    ```
    #chmod 440 /home/alumno/carta
    ```

  - Asigna todos los permisos al **u**suario dueño y al **g**rupo, y el permiso de lectura y ejecución a los **o**tros usuarios.

    ```
    #chmod 775 /opt/programa.sh
    ```

Comando chown
Cambiar propietario a un archivo

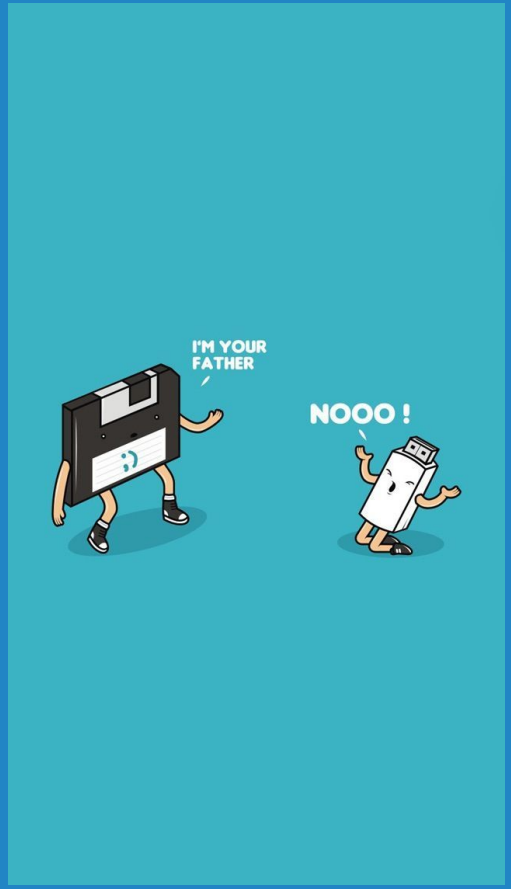
- Sintaxis: `#chown [-R] nuevo_propietario archivos`
- Opción -R para aplicación recursiva
- Ejemplos:

```
#chown mgarcia /home/glopez/carta.txt

#chown -R root:root /home/jperez/informes
```

¡Gracias!

¿Preguntas?



Referencias:

https://www.linuxtotal.com.mx/index.php?cont=info_admon_008

https://rm-rf.es/