4. Gestión de cuentas, usuarios y grupos

Elena García-Morato, Felipe Ortega, Enrique Soriano, Gorka Guardiola, Miguel Ortuño

GSyC, ETSIT. URJC.

Laboratorio de Sistemas (LSIS)

9 marzo, 2023

Contenidos

4.1 Usuarios y grupos

La cuenta de usuario en Unix/Linux

- Los sistemas Unix/Linux son multiusuario: varios usuarios pueden usar el mismo sistema, incluso de forma simultánea (conexión remota o local).
- Ejemplo: los propios PCs de laboratorios de la universidad.
- En un sistema Unix/Linux tenemos **usuarios** y grupos.
- Todo usuario pertenece a un grupo. Se puede consultar esta información con los comandos `whoami` e `id`.

```
jfelipe@f-l3103-pc23:~$ whoami
jfelipe
jfelipe@f-l3103-pc23:~$ id
uid=6017(jfelipe) gid=600(profes) grupos=600(profes),20(dialout),29(audio),46(plugdev),108(kvm),...
```

La cuenta de usuario en Unix/Linux

- Los usuarios permiten identificar quién a lanzado un proceso y otorgar permisos a los procesos.
- El proceso de autenticar a un usuario para que acceda a su cuenta es el *login*.
- Este proceso en la terminal lo hace el programa `login` o también gestores en los escritorios (`gdm` con GNOME, SDDM con KDE, etc.).
- El usuario introduce su nombre de usuario y su contraseña para que el sistema le autentique y le permita comenzar la sesión, accediendo a su cuenta.
- El acceso en modo remoto suele implicar más paso (autenticación multifactor usando el móvil u otro dispositivo, certificados criptográficos, etc.).

Pseudo-usuarios

- Algunos usuarios no pueden hacer *login*.
  - Ejecutan procesos de sistema (demonios).
  - Ejecutan mediante algún tipo de escalado de privilegios (por ejemplo, mediante el comando sudo).
- Los usuarios que no pueden hacer *login* se llaman pseudo-usuarios.

El directorio $HOME

- Cada usuario/a tiene un directorio etiquetado como `$HOME`.
- Se traduce automáticamente a la ruta de su propia carpeta de usuario/a.
- Recordemos que el símbolo ∼ también se traduce a la ruta a la carpeta `$HOME` del usuario/a.

Identificador de usuario y de grupo

- En en *kernel* los usuarios y grupos se identifican mediante etiquetas numéricas.
  - Identificador de usuario: `uid`.
  - Identificador de grupo: `gid`.
- En el espacio de usuario trabajamos con nombres (*usernames* o *login names*).
- Los programas de espacio de usuario traducen entre los números (del kernel) y los nombres (espacio de usuario) usando ficheros que mantienen datos de correspondencia entre ambos.

Fichero /etc/passwd

- Fichero de texto con la traducción entre *login names* y `uids` de usuarios/as.
- Contiene más información.

```
paurea@alpha01:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
pulse:x:115:122:PulseAudio daemon,,,:/var/run/pulse:/usr/sbin/nologin
paurea:x:1000:1001:Gorka Guardiola:/home/paurea:/bin/bash
```

Fichero /etc/passwd

- Contiene una línea por cada usuario, en la cual los datos de cada campo están separados por el carácter `:` .
- Nombre de usuario.
- Identificador de usuario (`uid`) y de grupo primario (`gid`).
- Identificador del usuario en formato legible (para humanos).
- Ruta al directorio `$HOME`.
- Ruta al intérprete de línea de comandos para ese usuario.

```
paurea:x:1000:1001:Gorka Guardiola:/home/paurea:/bin/bash
```

Fichero /etc/passwd

- Si el intérprete de comandos es `nologin` o se indica en ese campo el valor `false` entonces no se permite a ese usuario hacer *login* (para demonios).

```
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
```

Fichero /etc/passwd

- Las contraseñas se almacenan en otro fichero, `/etc/shadow`, que no permite que todos lo lean (para evitar que cualquiera tenga acceso).
- Un *hash* seguro (resumen) de la contraseña de cada usuario.
- Los campos también están separados por el carácter `:`.
- Contiene también fechas de cambio de contraseñas, tiempo máximo de validez, etc.
- Las cuentas desactivadas tienen como contraseña el símbolo `!`.
- Más información en `shadow(5)`.

Superusuario root

- El usuario que tiene los máximos privilegios en el sistema es el superusuario `root`.
- Tiene identificadores de usuario y grupo especiales.
  - `UID = 0`.
  - `GID = 0`.
- Posee permisos totales para hacer cualquier acción, por tanto es muy peligroso.
- Normalmente, muchos sistemas deshabilitan el *login* remoto del usuario `root` por razones de seguridad.

4.2 Gestión de usuarios

Creación y eliminación de usuarios

- Casi nunca editamos los ficheros con información de usuarios/grupos a mano. Es mejor usar herramientas que automatizan el proceso.
- El comando `adduser username` permite añadir un nuevo usuario.
- El comando `passwd username` permite cambiar la contraseña de un usuario.
- El comando deluser username permite borrar un usuario existente.

Credenciales y permisos usuarios

- Un **proceso** tiene *credenciales*, es decir, información sobre quién es el propietario del mismo (usuario y grupo).
- `PID`: Identificador de proceso.
- `UID`: Identificador de usuario propietario del proceso.
- `GID`: Identificador de grupo al que pertenece el usuario.
- `EUID`: Id. de usuario efectivo, que se usa realmente para comprobar permisos.
- `EGID`: Identificador de grupo efectivo, que realmente se emplea para comprobar permisos.

Control y gestión de credenciales

- El programa `/bin/id` devuelve tu UID y GID.
- `/bin/su` ejecuta una shell con otro UID. Por omisión intenta ejecutarla como superusuario (UID = 0).
- `/usr/bin/sudo` ejecuta un comando con otro UID, introduciendo la contraseña del usuario actual. El fichero `/etc/sudoers` indica quién puede convertirse en quién y qué acciones puede realizar.
- Ese fichero **solo se debe editar con el comando `visudo`**, nunca con otro editor.

Control y gestión de credenciales

- Ejemplo de entrada en el fichero `/etc/sudoers`:

```
pepe ALL = (root, bin : operator, system) ALL
```

- El usuario pepe puede, en cualquier máquina (`ALL`)
- ... adquirir el UID de `root` y `bin`
- ... adquirir el GID de `operator` y `system`
- ... para ejecutar cualquier comando.

Control y gestión de credenciales

- Otro ejemplo de entrada en `/etc/sudoers`:

```
paco laptop1 = NOPASSWD: /bin/kill, PASSWD: /bin/ls,
/usr/bin/lprm
```

- El usuario paco puede, en la máquina (mono)
- ... adquirir el UID de `root`
- ... para ejecutar `kill` sin meter contraseña
- ... para ejecutar `ls` o `lprm` metiendo contraseña.

Permisos POSIX

- Los ficheros y directorio tienen metadatos que informan sobre sus **permisos**.
  - Qué puede hacer el usuario (propietario/a) del fichero o directorio.
  - Qué pueden hacer el resto de usuarios que pertenecen al mismo grupo del propietario/a.
  - Qué pueden hacer el resto de usuarios del sistema (que no son el propietario ni pertenecen a su mismo grupo).

Permisos POSIX

- El comando `ls -l` muestra la información sobre ficheros, directorios y otras entradas en *formato largo*.
- `ls -ld` muestra sólo directorios.

```
-rw-rw-r--  1 jperez jperez     893 dic  1 20:10  file1
-rw-rw-r--  1 jperez jperez  198879 dic  1 20:21  file2
-rw-r--r--  1 jperez jperez    8980 may 13  2021  file3
drwxrwxr-x 15 jperez jperez    4096 feb 14 23:00  mydir
```

Permisos POSIX

- Para cada entrada del listado aparece:
  - Permisos: Los 10 primeros caracteres (incluido el primero, tipo de elemento).
  - Número de nombres del elemento (*hard links*).
  - Usuario propietario del elemento.
  - Grupo del propietario del elemento.
  - Tamaño en bytes.
  - Fecha y hora de la última modificación.
  - Nombre del elemento.

Permisos POSIX

```
-rw-r-r- 1 jperez jperez 8980 may 13 2021 file3
drwxrwxr-x 15 jperez jperez 4096 feb 14 23:00 mydir
```

- El primer carácter indica tipo de elemento.

-	Regular file - Fichero ordinario
d	Directory - Directorio
l	(Symbolic) Link - Enlace simbólico
p	Named pipe - Pipe con nombre
s	Socket - Socket
c	Character device - Dispositivo orientado a carácter
b	Block device - Dispositivo orientado a bloque

Permisos POSIX

- Los permisos de cada elemento (fichero, directorio, etc.) se codifican con los 9 últimos caracteres que aparecen al comienzo.

```
-rwxrw-r--
```

- El primer carácter es el tipo de elemento (ver diapositiva previa).
- Los caracteres del 2 al 4 indican los permisos del usuario propietario.
- Los caracteres del 5 al 7 indican los permisos del resto de usuarios del mismo grupo que el propietario.
- Los caracteres del 8 al 10 muestran los permisos del resto de usuarios del sistema (que no están en el mismo grupo que el propietario).

Permisos POSIX

- Al conjunto de permisos de un elemento (fichero, directorio, etc.) se le conoce como *modo de acceso* (*access mode*).
- Cada grupo de tres caracteres indica si el permiso está concedido (aparece la letra correspondiente) o no lo está (aparece –).
- **r**: Permiso de lectura. En un directorio permite leer las entradas de ese directorio (listado de elementos que contiene).
- **w**: Permiso de escritura. En un directorio permite escribir las entradas del directorio (crear nuevos elementos, borrarlos, renombrarlos, moverlos, etc.).
- **x**: Permiso de ejecución. En un directorio permite entrar o atravesarlo cuando se evalúa una ruta. Es necesario para acceder al contenido (datos y metadatos) de un fichero contenido en ese directorio.

Permisos POSIX

- Los permisos se representan comúnmente en octal en los comandos para gestionarlos. Por ejemplo: 0664 es `rw-` para el propietario, `rw-` para el grupo del propietario y `r--` para los demás usuarios.

- Otros permisos:
  - **sticky bit** (`+t`): Para directorios. Impide borrar una entrada si no eres dueño del directorio, del fichero/directorio que representa la entrada o `root`. Por ejemplo, se usa en `/tmp`.

```
drwxrwxrwt  21 root root      36864 feb 27 23:51 tmp
```

  - **setuid/setgid bit** (`+s`): El proceso que ejecute el fichero adoptará el UID/GID efectivo del dueño/grupo del fichero (en Linux se ignora para interpretados).

Permisos POSIX

- El comando `chmod` cambia los permisos de un fichero. Solo lo puede hacer el dueño del fichero y `root`.
- Inicialmente, el creador de un fichero es su dueño y grupo.
- El comando `chown` cambia el dueño de un fichero. Hay que tener privilegios especiales para hacer esto.
- El comando `chgrp` cambia el grupo de un fichero. El dueño puede cambiarlo a un grupo al que él pertenezca.
- **Cuidado**: En Linux los permisos POSIX conviven con otros permisos más potentes llamados ACLs. Véase `man 5 acl`.

Cambio de permisos

- Dos maneras de usar chmod.
- En octal: `chmod 0777` otorga `rwx` a todos.
- Con letras: `chmod u+x` da permiso de ejecución al usuario.
- Con letras: `chmod g+x` da permiso de ejecución al grupo.
- Opciones con letras: `u` usuario; `g` grupo; `o` el resto; `a` o no pongo nada indica para todos.
- Recursivamente: Opción `-R`. Aplica a todos los ficheros y directorios del árbol. `chmod -R g-rwx directorio`.

4.3 Gestión de grupos

Identificadores de grupo

- Permiten que varios usuarios compartan una lista de permisos de manera práctica.
- Todos los usuarios que pertenecen al mismo grupo tiene los mismos permisos.
- Se identifican por un *group name* y un `gid`.
- Están definidos en el fichero `/etc/group`.
- Cada línea del fichero describe un grupo. Los campos están separados por el carácter `:`.

Fichero /etc/group

- Nombre del grupo.
- Contraseña del grupo (no se usa).
- Identificador numérico del grupo `gid`.
- Lista de nombres de usuarios, separados por comas, que son miembros del grupo (además de los que lo tienen como grupo primario en el fichero `/etc/passwd`).

```
paurea@alpha01:~$ cat /etc/group
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
sudo:x:27:paurea
audio:x:29:pulse,paurea
```

Fichero /etc/group

- Para dar permisos a usuarios se pueden añadir a un grupo (por ejemplo para acceso a servicios, gestionar recursos de impresión, etc.).
- Se les puede añadir al grupo manualmente en este fichero, pero no es flexible.
- Método más potente y flexible: PAM (*Pluggable Authentication Modules*).
- Autenticación dinámica: es un servicio que permite al administrador del sistema decidir cómo autentican a los usuarios/as las diferentes aplicaciones.
- `man 5 pam`.
- Podemos buscar módulos con `man -k pam_` (ojo al subrayado al final).
- Ficheros de configuración en el directorio `/etc/pam.d`.

Fichero /etc/group

- El comando `adduser username groupname` permite añadir o eliminar un usuario de un grupo.
- El comando `deluser username groupname` permite añadir o eliminar un usuario de un grupo.
- Para crear un nuevo grupo `addgroup groupname`.
- Para borrar un grupo `delgroup groupname`.

Para saber más

- El capítulo 4 de [1] explica los sistemas de control de acceso y permisos en Linux, así como buenas prácticas para su gestión.
- El libro de referencia sobre administración en Unix/Linux [2] incluye mucha información adicional y excelentes explicaciones:
  - Capítulo 4: Control de acceso y privilegios de `root`.
  - Capítulo 7: Gestión de usuarios.
- El capítulo 7 de [3] también incluye apartados relevantes sobre gestión de usuarios y ficheros de configuración para gestión de usuarios y permisos.

Referencias I

[1] M. Hausenblas. *Learning Modern Linux*. O'Reilly Media, abr. de 2022.

[2] E. Nemeth y col. *Unix and Linux System Administration Handbook*. 4ª ed. Pearson, 2010.

[3] B. Ward. *How Linux Works: What Every Superuser Should Know*. 3ª ed. No Starch Press, abr. de 2021.