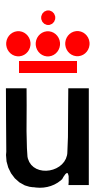
Gestión de usuarios y permisos

Laboratorio de Sistemas

Enrique Soriano, Gorka Guardiola

GSYC

4 de marzo de 2021



Universidad Rey Juan Carlos

Usuarios y grupos de Unix

- En un sistema de tipo Unix tenemos **usuarios** y **grupos**.
- Los usuarios pertenecen a grupos.

```
 paurea@alpha01:~$ whoami
paurea
 paurea@alpha01:~$ id
uid=5012(paurea) gid=600(profes) groups=600(profes),20(dialout),201(android),202(kvm),205(docker)
```

Usuarios de Unix

- Los usuarios sirven para dar permisos a los procesos.
- El proceso de autenticar a un usuario se llama *login*.
- El *login* lo hace el programa `login` en consolas de texto o los programas gráficos como `gdm` para el sistema de ventanas.
- El usuario escribe una contraseña para que el sistema le autentique y comenzar la sesión.

Pseudo-usuarios

- No todos los usuarios pueden hacer *login*, algunos ejecutan procesos del sistema (*daemons*) y ejecutan por algún tipo de escalado de privilegios (por ejemplo *sudo*).
- Los usuarios que no pueden hacer login se llaman *pseudo-usuarios*.

uids y gids

- En el kernel, los usuarios y grupos de Unix son números (id, usuarios: uid, grupos gid).
- En espacio de usuario se trabaja con nombres, (*usernames* o *login names*).
- Los programas de espacio de usuario traducen entre números y nombres usando ficheros.

El fichero /etc/passwd

- Es un fichero de texto con la traducción entre nombres y uids de usuarios
- 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
...
```

El fichero /etc/passwd

Tiene, separados por dos puntos, una línea por usuario:

- Nombre de Usuario
- Hash segura del password (o mejor, una KDF). Ya no suele estar, está en /etc/shadow, más adelante.
- Número identificador de usuario uid

```
root:x:0:0:root:/root:/bin/bash
paurea:x:1000:1001:Gorka Guardiola:/home/paurea:/bin/bash
```

El fichero /etc/passwd

- Número identificador de grupo: gid, grupo primario al que pertenece el usuario.
- Nombre real y apellidos.
- Directorio casa del usuario.
- Shell, programa cuando se ejecuta una sesión interactiva (nologin o false para demonios que no permiten login).

```
root:x:0:0:root:/root:/bin/bash
paurea:x:1000:1001:Gorka Guardiola:/home/paurea:/bin/bash
```

El fichero `/etc/shadow`

- Para evitar que cualquiera tenga acceso a la hash del password.
- Se mete en otro fichero sin permisos de lectura para todos.
- `/etc/shadow`.
- Este fichero también está separado por dos puntos.
- El segundo campo es la hash de la password, y puede estar en distintos formatos.
- Contiene fechas de cambio de passwords, cuando expiran...
- Una cuenta desactivada tiene como contraseña una admiración (!).
- Ver **shadow(5)**.

Root

- El UID 0 y GID 0 es especial.
- Es el superusuario, que se llama root.
- Tiene permisos para todo.

Grupos

- Sirven también para permisos, son parecidos a los usuarios.
- Tienen nombre (*group name*) e identificador gid.
- Están definidos en `/etc/group`

El fichero /etc/group

Separados por dos puntos

- Nombre del grupo.
- Password del grupo (no se usa).
- Número que identifica al grupo, gid.
- Lista de nombres de usuarios separados por comas que pertenecen al grupo (aparte de los que lo tienen como grupo primario 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
...
```

Grupos

- Para dar permisos a usuarios a usar servicios.
- Se les suele añadir al grupo (por ejemplo, audio).

PAM

- Lo que hemos visto hasta ahora, no suficientemente flexible.
- Por ejemplo, varias máquinas comparten cuentas (LDAP) o queremos usar tarjetas para autenticación.
- Utiliza un sistema con módulos cargables (librerías dinámicas) sofisticado.
- `man 5 pam`.
- Para ver módulos, buscar con `man -k pam_` (el subrayado al final).
- Ver `/etc/pam.d` ficheros de configuración.

Comandos

Casi nunca se editan los fichero a mano.

- Para cambiar el password de un usuario: `passwd username`.
- Para añadir un usuario (o borrarlo) `adduser username` (`deluser username`).
- El comando `adduser` sirve tambien para añadir o eliminar un usuario de un grupo. Para añadir un usuario `deluser username groupname` y para borrarlo `deluser username groupname`
- Para crear un nuevo grupo (o borrarlo) `addgroup groupname` (`delgroup groupname`).

Credenciales

Un proceso tiene *credenciales*[1]:

- `PID`: identificador de proceso.
- `UID`: identificador de usuario.
- `GID`: identificador de grupo.
- `EUID`: identificador de usuario efectivo, el que se usa para comprobar privilegios.
- `EGID`: identificador de grupo efectivo, el que se usa para comprobar privilegios.
- ...

[1] Dependen del "sabor" de Unix

Control de credenciales

- `/bin/id`: Permite ver tu UID y GID.
- `/bin/su`: Permite ejecutar un shell con otro UID. Por omisión, intenta ejecutar un shell con UID 0.
- `/usr/bin/sudo`: Permite ejecutar un comando con otro UID, proporcionando tu propia contraseña. El fichero `/etc/sudoers` especifica quién puede convertirse en quién, y para qué. Ese fichero se puede editar con el comando `visudo`, nunca hay que hacerlo con otro editor. **Mucho cuidado**.

Control de credenciales

Ejemplo de sudoers (I):

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

- jose puede, en cualquier máquina
- adquirir el UID de root y bin
- adquirir el GID de operator y system
- para ejecutar cualquier comando

Control de credenciales

Ejemplo de sudoers (II):

```
ramon mono = NOPASSWD: /bin/kill, PASSWD: /bin/ls,
/usr/bin/lprm
```

- ramon puede, en la máquina mono
- adquirir el UID de root
- para ejecutar kill sin proporcionar contraseña
- para ejecutar ls y lprm proporcionando contraseña

Permisos POSIX

Se establecen esos permisos para:

- dueño.
- grupo.
- resto de usuarios.

```
rwx rwx rwx
```

Tipo de acceso:

- **r**: permiso de lectura. En directorio: se pueden leer las entradas de directorio.
- **w**: permiso de escritura. En directorio: se pueden escribir las entradas del directorio (borrar, renombrar, añadir ficheros).
- **x**: permiso de ejecución. En directorios: se puede entrar o atravesar el directorio cuando se evalúa una ruta. Es necesario para acceder a un fichero del directorio (datos y metadatos).

Unix: Permisos POSIX

- Los permisos se representan normalmente en octal:
  P. ej: 0664 es 110 para el dueño, 110 para el grupo, 100 para el resto.
- Hay otros permisos
  - **sticky bit** (+t): es para directorios: no puedes borrar una entrada si no eres el dueño del directorio, del fichero/directorio que representa la entrada, o root (se usa en `/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).
- Podemos ver los permisos con el comando `ls -l`. Si los queremos ver del directorio y no de las entradas: `-d`.

Unix: 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.

OJO: en Linux los permisos POSIX conviven con otros permisos más potentes llamados ACLs. `man 5 acl`.

Cambiar permisos

- Dos maneras de usar `chmod`
- En octal: `chmod 0777`, `rwx` para todos.
- Con letras `chmod g+x` da permiso de ejecución al grupo.
- Con letras `chmod u+x` da permiso de ejecución al usuario.
- Puede ser `u` user, `g` grupo, `o` el resto o nada (para todos, igual que `a`).
- Y recursivamente (todos los ficheros y directorios del árbol) `chmod -R g-rwx directorio`.

Cambiar dueño

- Para cambiar simultáneamente usuario y grupo
- `chown paurea:audio ruido.mp3`