Administración de Usuarios

Introdución

Dous tipos de usuarios:

- **Locais**: Son os que veremos neste tema
- **Dun Dominio**: Non se almacenan neste equipo, almacénanse no servidor de dominio e para consultar a súa información emprégase o protocolo **OpenLDAP**.

Comandos Básicos

- **passwd:** Un usuario pode cambiar o seu contrasinal. O root pode cambiar la contrasinal de calquera usuario
- **whoami:** Indica o usuario actual

su (substitute user)

Permite abrir unha sesión como root ou como outro usuario se coñecemos a súa contrasinal.

`su`	Convértenos en root
`su usuario`	Iniciamos un shell como outro usuario, pero mantendo o noso entorno
`su - usuario`	Iniciamos un shell como outro usuario, pero co entorno do novo usuario.
`su -c comando`	Executamos ese comando como se fósemos root `su -c ifconfig`

Todas as sesións iniciadas como root quedan rexistradas no arquivo `/var/log/auth.log`.

O arquivo de contrasinais

Existe unha liña no arquivo **`/etc/passwd`** por cada conta de usuario existente no sistema. Cada entrada ten o seguinte formato:

```
nombreDeUsuario:contraseña:uid:gid:gecos:directorio:shell
```

Cada campo significa:

- **nombreDeUsuario**: En minúsculas e sen espacios.
- **Contraseña**: Unha representación encriptada da contrasinal do usuario.
- **uid** (**u**ser **id**entifier): É un número enteiro que o sistema emprega para identificar unha conta de usuario. O 0 identifica ao usuario root.
- **gid:** O identificador de grupo é un enteiro que caracteriza ao grupo principal do usuario. Atópase no arquivo `/etc/group`
- **gecos:** Información diversa sobre o usuario: como o nome real, dirección ou número de teléfono.
- **directorio:** É o directorio de traballo do usuario. Cando inicia unha sesión, o sistema o conduce automaticamente a este directorio.
- **shell:** Por defecto /bin/bash.

```
root:x:0:0:root:/root:/bin/bash
bin:x:2:2:bin:/bin:/usr/sbin/nologin
usuario:x:1000:1000:Usuario Martínez,Cangas,,:/home/usuario:/bin/bash
```

Se en el lugar da contrasinal aparece unha x, é que está empregando contrasinais de sombra ou **shadow** que veremos máis adiante.

Temos dous tipos de usuarios:

- **Usuarios reais:** Poden iniciar sesión. Teñen UID's a partir de 1000.

```
usuario:x:1000:1000:usuario,,,:/home/usuario:/bin/bash
```

- **Usuarios software:** Representa programas e non poden iniciar sesión

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

Temos varios comandos para xestionar ós usuarios que nos evitan ter que modificar o arquivo `/etc/passwd` a man.

Contrasinais de sombra ou shadow

Se observamos os permisos do arquivo `/etc/passwd` comprobamos que calquera usuario pode ler o seu contido. Isto podería ser un risco de seguridade, xa que existen programas especializados en obter contrasinais, que funcionan probando posibles palabras ata que a codificación de algunha coincide coa escrita no arquivo (Ataque por fuerza bruta).

Para evitar este tipo de cosas empréganse as contrasinais sombra ou shadow que se almacenan en `/etc/shadow`.

```
root@debianVM:~# ls -l /etc/passwd
-rw-r--r-- 1 root root 1832 Mar 18 11:52 /etc/passwd
root@debianVM:~# ls -l /etc/shadow
-rw-r----- 1 root shadow 1332 Mar 18 11:52 /etc/shadow
```

O empregar unha sombra, o campo correspondente á contrasinal contén unha x ou un asterisco. A versión encriptada da contrasinal almacénase en **`/etc/shadow`** que é propiedade do root, e ninguén máis pode lelo.

```
root@debianVM:~# cat /etc/passwd | grep usuario
usuario:x:1000:1000:usuario,,,:/home/usuario:/bin/bash
root@debianVM:~# cat /etc/shadow | grep usuario
usuario:
$6$FyZbQkJW$U9YAW61wr2G97cFUpi2hYZeg7aYPXxL3aY9xD2dPWWUMpbnrK0ssX69xB5
sFinhqycnan0KrND66mx7k6.KsI1:17630:0:99999:7:::
```

Por defecto as contrasinais de sombra están activadas. Podemos cambiar a súa configuración con:

- `shadowconfig off` → Desactiva as contrasinais shadow e migra as contrasinais encriptadas a `/etc/passwd`
- `shadowconfig on` → Activa as contrasinais shadow e crea o arquivo `/etc/shadow`

Xestión de Contas de usuario

adduser ou useradd

Crea contas de usuario e grupos. Só a pode empregar o root.

useradd é menos completo, é un estándar, mentres que adduser é un script propio de cada distribución.

Exemplo

`adduser pepe`	Engade unha liña a `/etc/passwd` Crea a carpeta `/home/pepe` Pregunta contrasinal e a súa información de gecos
`adduser -group alumnos`	Engade o grupo alumnos

Cando creamos un usuario fanse as seguintes tarefas:

- El sistema busca o seguinte uid dispoñible para asignarllo,
- Crea un grupo co nome do usuario, se non lle proporcionamos un.
- Crea o directorio de traballo dese usuario e copia nel uns arquivos de configuración comúns a partir de `/etc/skel`
- Pídenos a contrasinal
- Pídenos información relativa ao usuario (opcional) que pode ser empregada por outros comandos como **`finger`**.

Podemos engadir parámetros para modificar o seu funcionamento por defecto

`--uid número`	Uid do usuario
`--gid número`	Gid do grupo primario do usuario
`--home directorio`	Directorio de traballo
`--add_extra_groups grupo1, grupo2 ..`	Grupos secundarios
`--shell directorio del shell`	Para especificar outro shell que non sexa o que está por defecto
`--gecos comentario`	Para incluír un comentario acerca de ese usuario

deluser ou userdel

Por defecto elimina unha conta de usuario pero conserva o seu directorio persoal e os seus arquivos correspondentes.

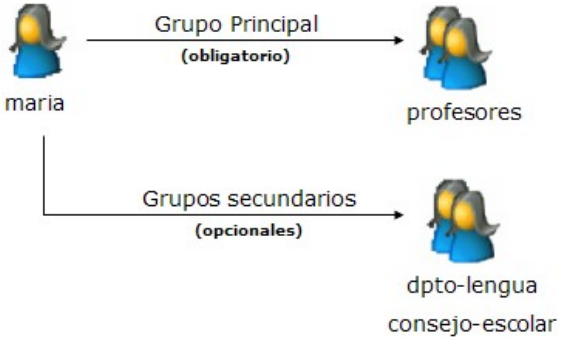
`--remove-home`	Borra o directorio de traballo
`--remove-all-files`	Borra o directorio de traballo e todos os arquivos dese usuario.

Se queremos tan só desactivar unha conta temporalmente, podemos engadir un símbolo de finalización de admiración (!) como primeiro carácter do campo contrasinal no arquivo `/etc/passwd`.

Administración de grupos

Os grupos son un mecanismo para administrar dun xeito uniforme a varios usuarios para, entre outras cousas, que podan compartir os seus arquivos entre eles. Cada arquivo do sistema está asociado a un usuario e un grupo.

- Cada **usuario pertence ao menos a un grupo**, o que se indica co campo gid no archivo `/etc/passwd`, é o **seu grupo principal**,
- En caso de que pertenza a máis grupos serán os seus **grupos secundarios**.



O arquivo **`/etc/group`** ten unha liña por cada grupo existente. O seu formato é moi similar ao arquivo `/etc/passwd`.

```
Grupo:contrasinal:gid:membros
plugdev:x:46:usuario1,usuario2
```

- **Grupo** é o nome do grupo
- O **contrasinal** é opcional e emprégase para que entren no grupo de xeito momentáneo, usuarios que non pertencen a el. Isto faise con **newgrp**.
  Os contrasinais dos grupos almacénanse de xeito similar ás dos usuarios, así que se empregamos contrasinais sombra, están almacenadas en **/etc/gshadow.** Hoxe en día non se soen empregar.
- **gid** é o número de identificación do grupo
- **membros** é a lista de nomes dos usuarios do grupo que teñen como grupo principal a outro, separados por comas e sen espazos entre eles.

addgroup o groupadd

Crea un novo grupo
Sintaxe: addgroup [parámetros] nombreGrupo
`addgroup grupo1` → Crea unha nova entrada en `/etc/group`
`grep grupo1 /etc/group` → Amosa a entrada creada

delgroup o groupdel

Elimina grupos. Só elimina un grupo se non existen usuarios que teñan ese grupo como principal
Sintaxe: delgroup nombreGrupo
`delgroup grupo1` → Elimina a entrada de grupo1 en `/etc/group`

groupmod

Modifica os atributos dun grupo

`groupmod –g gid nombreGrupo`	Cambia o gid
`groupmod –n nuevoNombre nombreGrupo`	Cambia o nome do grupo

Exemplo:
`groupmod –n grupo5 grupo2` → Cambia o nome de grupo2 a grupo5

usermod

Sirve para modificar as características da conta de algún usuario e para engadir usuarios a grupos.

Podemos cambiar –u(uid),-g(gid), -d(home), -s(shell), -c(gecos) y	
`–l nuevo_nombre`	Nome de usuario
`–p contraseña`	Contrasinal
`–G grupo1, grupo2`	Asignar grupos secundarios

Ejemplos
`usermod –c 'Esto es un comentario' pepe` → Cambia su gecos
`usermod –G Grupo1 pepe` → Cambia su Grupo secundario
`usermod –g Grupo1 pepe` → Cambia su Grupo principal

groups

Amosa a que grupos pertence o usuario que teclee a orden.
`groups usuario` → Amosa os grupos ós que pertence ese usuario.

ATENCIÓN: Os cambios de usuarios a grupos *NON* son efectivos ata o seguinte **login** (se o usuario xa ten sesións activas)

Permisos en Linux

- Se un usuario é o propietario dun arquivo aplícaselle o grupo owner
- Se un usuario ten como grupo principal ou secundario, o grupo do arquivo, se lle aplican os permisos do grupo
- Senón aplícanselle os permisos outros

gpasswd

Dun grupo se almacénanse tres cousas: membros, contrasinais e administradores. O administrador do grupo pode entre outras cousas, engadir e eliminar usuarios del grupo.

gpasswd –A usuario grupo	Especifica administrador
gpasswd –M usuario grupo	Quita administrador
gpasswd grupo	Permite establecer contrainal a un grupo
gpasswd –r grupo	Quita o contrasinal
gpasswd –a usuario grupo	Añade miembros al grupo
gpasswd –d usuario grupo	Elimina miembros del grup

newgrp

Permite iniciar un novo shell cambiando o identificador do grupo (gid) do usuario actual. Só poderán facelo os usuarios que pertenzan ó grupo, ou en caso de que este teña unha contrasinal, aqueles que a coñezan.

Pode ser útil para iniciar sesión momentaneamente dun grupo o que non pertenzamos.

Sintaxis: newgrp [grupo]

Grupos especiais

En Debian existen uns grupos predefinidos que controlan o acceso a dispositivos e directorios do sistema.

Para engadir a un usuario a un grupo:

- `adduser usuario grupo`
- `gpasswd -a usuario grupo`

Cando creamos un usuario durante a instalación do sistema engádense por defecto os seguintes grupos:

```
root@debian:~# groups usuario
usuario : usuario cdrom floppy audio dip video plugdev netdev
bluetooth
```

`dialout`	Conexión a Internet mediante un modem
`dip`	
`cdrom`	Acceso ao cdrom
`floppy`	Acceso rw a disquetera
`audio`	Acceso rw al sonido (mezclador, etc)
`video`	Acceso a cdrom
`plugdev`	Acceso a las unidades removibles
`netdev`	Administrar interfaces de rede

Como en Windows existen moitos grupos predefinidos (built-in) que conceden diversos privilexios no sistema. Para saber máis:

Exemplos de permisos de dispositivos:

- `brw-rw---- 1 root **cdrom** 11, 0  Mar 11 23:19 /dev/sr0`
- `crw------- 1 root **tty**   4, 1  Mar 11 23:27 /dev/tty1`

Executando tarefas como superusuario

En Linux **só existe un usuario Administrador.** Isto reforza a seguridade e estabilidade do sistema, pero a veces poderíanos interesar que un usuario normal poda realizar algunhas tarefas que por defecto só pode realizar o root, como por exemplo poder instalar programas. Para solucionar isto temos o comando **sudo (SuperUser Do)**.

En moitos sistemas, **incluído Ubuntu**, a conta de root está deshabilitada e o único xeito de traballar como administrador é empregando o comando sudo. Isto faise así, entre outras cosas, para:

1. Minimizar o risco de ataques por forza bruta.
2. Permite realizar tarefas de administración sen que ninguén teña que coñecer o contrasinal de root.

```
apt-get install sudo
```

O comando lee a configuración do arquivo **/etc/sudoers**. Observa os seus permisos. Hai algo raro neles?

```
user@debian:~$ ls -ls /etc/sudoers
4 -r--r----- 1 root root 322 may 10 12:25 /etc/sudoers
```

- Para modificalo temos que empregar o comando
```
visudo
```

- Cada liña ten o seguinte formato

```
usuario/grupo host =  comando
```

  Podemos dar permisos a usuarios o a grupos.

- Por exemplo se queremos que usuario poda instalar programas e reiniciar a máquina

```
usuario ALL= /usr/bin/apt-get
```

- Para un grupo a liña sería

```
%grupo ALL= /usr/bin/apt-get
```

Para instalar un programa como usuario, pediranos a nosa contrasinal.

```
sudo apt-get install programa
```

Un comando muy parecido es el comando **super**

Utilizando ACL's en los permisos de Linux

Linux emprega o modelo de permisos **UGO** (**U**suario **G**rupo **O**utros), pero este modelo pode quedarse corto cando intentamos dar diferentes permisos a diferentes grupos ou usuarios sobre o mesmo arquivo. Por exemplo: Supoñamos unha carpeta Apuntes, na que queremos os seguintes permisos:

- os usuarios do grupo Alumnos1 teñen permiso de lectura e escritura
- os usuarios do grupo Alumnos2 teñen só permiso de lectura
- xiana que é do Alumnos1 non ten acceso
- O resto de usuarios do sistema non teñen acceso

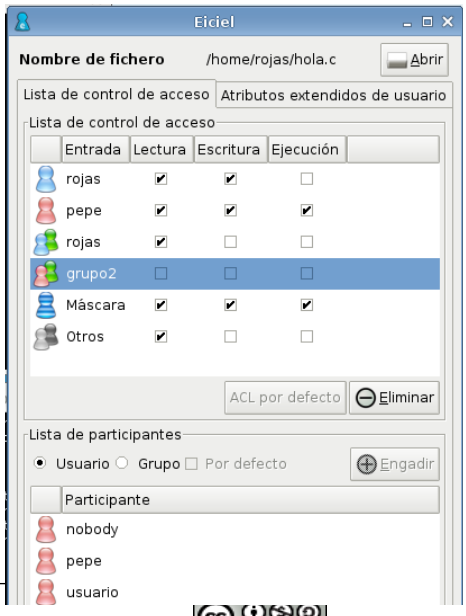
Este caso, co modelo de permisos tradicional de ext4, non o podemos implantar, necesitamos empregar **ACL's** (**L**istas de **C**ontrol de **A**cceso) do mesmo xeito que con NTFS, para asignar diferentes permisos a distintos usuarios ou grupos sobre o mesmo obxecto.

Para asignar e consultar os permisos empréganse respectivamente **`setfacl`** e **`getfacl`**.

Sintaxe:

`setfacl -m group:grupo2:r-x dir1`	Engade ou modifica os permisos dun grupo sobre o directorio
`setfacl -m user:user1:r-x dir1`	Engade ou modifica os permisos dun usuario sobre o directorio
`setfacl -x user:user1 dir1`	Elimina a entrada dese usuario sobre ese directorio.
`-R`	Cambia os permisos de xeito recursivo
`-b`	Borra todos os permisos adicionais conservando unicamente os UGO.

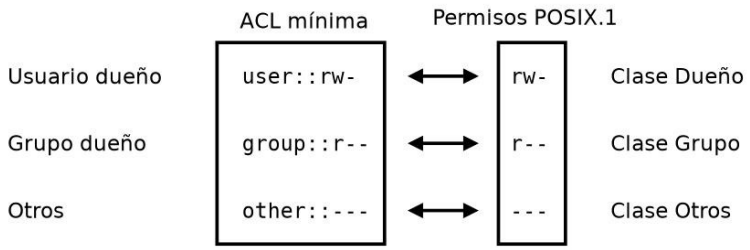
Tamén dispoñemos dunha ferramenta chamada **eiciel** que nos permite asignar e comprobar as acl's de xeito gráfico.



Afondando no entendemento das ACL's

Temos dous tipos de permisos ACL's:

- **Os mínimos:**
  Mapeanse directamente dos permisos UGO



Acl

```
brais@debian:/tmp$ getfacl file1.txt
# file: file1.txt
# owner: brais
# group: brais
user::rw-
group::r--
other::r--
```

Permisos UGO

```
user@debian:/tmp$ ls -l file1.txt
-rw-r--r-- 1 brais brais 0 Out 30 23:39 file1.txt
```

Serían equivalentes os comandos

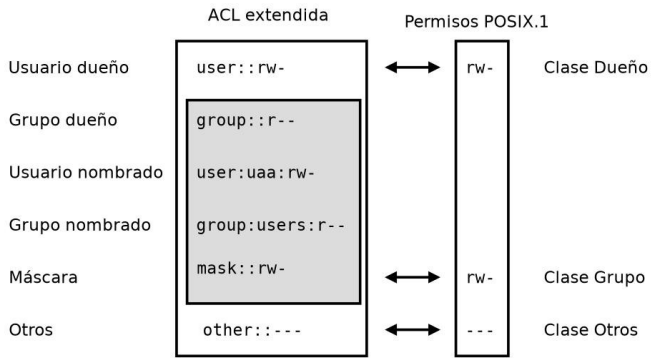
Acl

```
setfacl -m user::rwx file1.txt
setfacl -m group::rwx file1.txt
setfacl -m other::rwx file1.txt
```

Permisos UGO

```
chmod u+rwx file1.txt
chmod g+rwx file1.txt
chmod o+rwx file1.txt
```

- **Os estendidos**
  Cando engadimos entradas acl's a maiores das básicas



  - Listamos os permisos

```
user@debian:/tmp$ ls -l file1.txt
-rw-r--r-- 1 brais brais 0 Out 30 23:39 file1.txt
```

  - Engadimos unha acl para o usuario proxy

```
user@debian:/tmp$ setfacl -m user:proxy:rw- file1.txt
```

  - Listamos os permisos

```
user@debian:/tmp$ ls -l file1.txt
-rw-rw-r--+ 1 brais brais 0 Out 30 23:39 file1.txt
```

    - Engadiuse un +
    - Os permisos de grupo cambiaron

  - Listamos as acls

```
user@debian:/tmp$ getfacl file1.txt
# file: file1.txt
# owner: brais
# group: brais
user::rwx
user:proxy:rw-
group::r--
mask::rw-
other::r--
```

    - Engadiuse a acl para o usuario proxy
    - Engadiuse mask::rw-

A entrada **mask**

Representa os permisos máximos para os grupos e é o que se amosa con un ls.

```
rojas@debianRojas:/tmp/1$ setfacl -m mask::r-- file1.txt
rojas@debianRojas:/tmp/1$ getfacl file1.txt
# file: file1.txt
# owner: rojas
# group: rojas
user::rwx
user:proxy:rw-                  #effective:r--
group::r--
mask::r--
other::r--
```

Se lle quitamos o permiso w á máscara, o usuario proxy terá de permisos efectivos r.

Default

Podemos especificar os permisos por defecto para os futuros arquivos o directorios que se creen dentro dunha carpeta

```
setfacl -dm u:proxy:rw dir1/

user@debian:/tmp/1$ getfacl dir1
# file: dir1
# owner: brais
# group: brais
user::rwx
group::r-x
other::r-x
default:user::rwx
default:user:proxy:rw-
default:group::r-x
default:mask::rwx
default:other::r-x
```

- Creamos unha subcarpeta e herda a entrada do usuario proxy e os permisos por defecto

```
brais@debian:/tmp/1$ mkdir dir1/subdir1
brais@debian:/tmp/1$ getfacl dir1/subdir1/
# file: dir1/subdir1/
# owner: brais
# group: brais
user::rwx
user:proxy:rw-
group::r-x
mask::rwx
other::r-x
default:user::rwx
default:user:proxy:rw-
default:group::r-x
default:mask::rwx
default:other::r-x
```

Para saber máis:

- https://juncotic.com/acl-access-control-lists-y-los-permisos-en-gnu-linux/