Administrar usuarios e grupos en Linux

Linux é un sistema multiusuario que se basea en contas, que son estruturas de datos e procedementos que se utilizan para identificar aos usuarios concretos dun ordenador. Saber administrar estas contas é un coñecemento básico pero importante.

Moitas redes empregan bases de datos de contas de rede. Entre estes sistemas están NIS (*Network Information System*, Sistema de información da rede), unha actualización deste sistema chamada NIS+, LDAP (*Lightweight Directory Access Protocol*, Protocolo lixeiro de acceso a directorios), dominios de Kerberos, dominios de Windows NT 4.0 e dominios de Directorio Activo (AD).

Sumario

Os nomes de usuario de Linux

Os nomes de usuario de Linux consisten en calquera combinación de letras maiúsculas e minúsculas, números e moitos símbolos de puntuación, incluíndo puntos e espazos.

- Non obstante, algúns signos de puntuación, como os espazos, causan problemas en determinadas utilidades de Linux, polo que é preferible evitar o seu uso nos nomes de usuario.
- Ademais, os nomes de usuario deben comezar por unha letra, polo que non será válido un nome de usuario como "69a", pero "a69" é correcto.
- Aínda que os nomes de usuario poden costar de ata 32 caracteres, moitas utilidades truncan os nomes de usuario de mais de 8 caracteres ou similares ao mostralos, polo que é interesante limitar a lonxitude dos nomes a 8 caracteres.
- Linux distingue entre maiúsculas e minúsculas nos nomes de usuario.

Os comandos *w* e *whoami*

Temos estes dous comandos que nos indican:

- O comando w mostra que usuarios están conectados ou con sesión aberta. Se se especifica usuario, só mostra as conexións dese usuario.
- O comando whoami nos indica que usuario somos agora mesmo.

Os grupos en Linux

É interesante agrupar aos usuarios mediante grupos para mellorar a productividade.

- Cada ficheiro dun sistema Linux está asociado a un grupo.
- Cada grupo pode ter dende ningún usuario ata tantos como teña o ordenador. Se un usuario pertence a un grupo ou a outro, agás o seu grupo por defecto ou primario, contrólase mediante o ficheiro ***/etc/group***. Vexamos un exemplo:

```
adm:x:4:patricia,juser
cdrom:x:24:adri,patricia,juser
lpadmin:x:106:patricia,juser
admin:x:112:patricia,juser
patricia:x:1000:
```

Cada liña do ficheiro ten a seguinte estrutura:
grupo:contrasinal:GID:usuario1,usuario2,...,usuarioN

- **grupo**: nome do grupo
- **contrasinal**: campo de contrasinal (neste exemplo cunha x indicando que as auténticas contrasinais están gardadas de forma encriptada no ficheiro /**etc**/**gshadow**)
- **GID**: identificador numérico do grupo
- **listaUsuarios**: lista separada por comas dos usuario que teñen a ese grupo como grupo secundario.

- Ademais do descrito en ***/etc/group***, cada usuario posúe un grupo por defecto ou primario. O grupo primario do usuario defínese na configuración do usuario de ***/etc/passwd*** (o ficheiro que define as contas).

```
javier:x:1000:1000::/home/javier:/bin/bash
```

Como pode verse no exemplo anterior, para cada usuario almacénase a seguinte información:
login:x:UID:GID:descrición:directorioPersoal:Shell
Como no caso de /**etc**/**group** o **x** indica que as auténticas contrasinais están gardadas no ficheiro /**etc**/**shadow**.

```
usuario:$6$kbzYVOsJ$84OVN7IApJLC/jw60jXZP/ierMeCqgK2X9iFYOXTNZokkzxC5i.z7MbBJP7QgT4mQV/Qg9z89CCorJ5Ef4/au1:15596:0:99999:7:::
sshd:*:15604:0:99999:7:::
```

Cada entrada no ficheiro /**etc**/**shadow** ten a seguinte estrutura:
nomeUsuario:contrasinal:ultimoCambio:min:max:aviso:inactividade:exp:reservado

- **nomeUsuario**: nome do usuario
- **contrasinal**: contrasinal cifrada do usuario
- **ultimoCambio**: data do último cambio do contrasinal (días dende o 1 de Xaneiro de 1970)
- **min**: mínimo número de días que deben pasar para poder cambiar o contrasinal
- **max**: número máximo de días que pode estar o usuario con ese contrasinal. Utilízase para obrigar ao usuario a que cambie o contrasinal
- **aviso**: número de días que se avisará con antelación ao usuario de que o contrasinal vai caducar
- **inactividade**: número de días antes de que se bloquee a conta se non cambiou o contrasinal
- **exp**: data na que caduca a conta. Cóntase en días a partir do 1/1/1970. Se o campo está en branco, a conta non caduca nunca. Se caducou o superusuario ou administrador do sistema deberá activala
- **reservado**: reservado para usos futuros

Contrasinal:

O contrasinal almacénase encriptado, polo que non garda parecido evidente có contrasinal real. Un asterisco (*) ou un signo de admiración (!) denota unha conta sen contrasinal (é dicir, unha conta que non permite o acceso, pois está bloqueada). Isto é habitual en contas que utiliza o propio sistema.

- Un usuario pode acceder a ficheiros que pertencen a outros grupos sempre que o usuario pertenza a dito grupo e que o acceso estea permitido polos permisos de acceso do grupo. Sen embargo, para executar programas ou crear ficheiros cun grupo distinto do primario, o usuario deberá executar o comando ***newgrp*** para cambiar a pertenencia ao grupo actual. Nese momento é cando o sistema lle pedirá ao usuario o contrasinal do grupo para poder pertencer a el.

Comando groups:

Con este comando podemos ver os grupos aos que pertence un usuario, tanto o principal como os secundarios:

$ groups usuario.

Asociar UID e GID a usuarios e grupos

Linux define aos usuarios e grupos por números, aos que se refire como ID de usuario (UID) e ID de grupos (GID), respectivamente.

- Linux segue o rastro dos usuarios e grupos por estes números, non polo seu nome. Linux emprega ***/etc/passwd*** ou ***/etc/group***.

Comando *id*

O comando id mostra a identidade do usuario (UID) e os grupos aos que pertence.

- As distribucións de Linux reservan o primeiro centenar de ID de usuario e grupos (0-99) para o uso do sistema. O mais importante destes é o **0**, que corresponde ao ***root*** (tanto o usuario como o grupo).
- Mais alá do 100, as ID de usuario e grupo atópanse dispoñibles para os usuarios e grupos normais.
- Cando se crean contas de usuario e de grupo o sistema localizará o seguinte número sen utilizar.
- Cando se elimine unha conta, o número de ID de dita conta poderase reutilizar.
- Normalmente o GID 100 é ***users***, o grupo por defecto para algunhas distribucións.
- Poderíanse crear varios nomes de usuario que empreguen a mesma UID ou varios nomes de grupo que empreguen a mesma GID. Trataríase, en certo modo, de contas ou grupos distintos; teñen entradas diferentes en */etc/passwd* ou */etc/group*, polo uqe poden ter distintos directorios *home*, diferentes contrasinais, etc. Pero, como teñen o mesmo ID que outros usuarios ou grupos, serán tratados de idéntico xeito no referente aos permisos dos ficheiros.

Configurar contas de usuario

Como administrador de Linux é necesario saber cómo engadir, modificar ou eliminar contas de usuario.

Engadir usuarios

Os usuarios pódense engadir mediante as utilidades useradd e adduser.

- **useradd** - É a utilidade por defecto que ten Debian para engadir usuarios. A súa sintaxe básica é a seguinte:

```
useradd [-c comentario] [-d dir-home] [-e data-expiración] [-f días-inactivo]
 [-g grupo-defecto] [-G grupo[,...]] [-m [-k dir-esqueleto] | -M]
 [-p contrasinal] [-s shell] [-u UID [-o]] [-r] [-n] nome-usuario
```

- **adduser** - Engade usuarios e grupos ao sistema de acordo ás opcións da liña de ordes seleccinadas e á configuración existente no arquivo **/etc/adduser.conf**. Ofrece unha interfaz máis sinxela que **useradd**, seleccionando un valor axeitado para o identificador de usuario (UID), crea un directorio personal (/**home**/**USUARI@**) coa configuración predeterminada, executa un script personalizado e outras funcionalidades.

  - Na súa forma mais simple, pódese escribir só ***useradd nome-usuario***. O resto dos parámetros utilízanse para modificar os valores por defecto do sistema, que se almacenan no ficheiro ***/etc/login.defs***.
  - **Vexamos un exemplo:** "Temos un sistema no que se engadeu un disco duro montado como */home2*. Deséxase crear unha conta para unha usuaria chamada **lara** e colocar o seu directorio *home* nese disco. Tamén se desexa que **lara** sexa membro dos

grupos *proxecto1* e *proxecto4*, tendo *proxecto4* como grupo por defecto. A usuaria tamén quere que *bash* sexa a súa consola por defecto". O comando que crea a usuaria **lara** con eses parámetros é o seguinte:

```
# Se non están creados os grupos:
$ groupadd proxecto4
$ groupadd proxecto1
# Agora créase o usuario:
$ useradd -d /home2/lara -m -g proxecto4 -G proxecto1,proxecto4 -s /bin/bash lara
# Por último, configurámoslle un contrasinal por defecto:
$ passwd lara
```

- Para automatizar o cambio de contrasinais é moito máis útil o comando **chpasswd**:

```
# Có comando mkpasswd ciframos o contrasinal a empregar
$ mkpasswd
# Password:blah
# BVR2Pnr3ro5B2
##
# Agora executamos o comando chpasswd indicando que o contrasinal ven cifrado.
# Pasámoslle o binomio usuario:contrasinal
$ echo "lara:$(mkpasswd abc123..)" | chpasswd -e
```

Se *mkpasswd* non está instalado:

```
$ apt-get install whois
```

Skeleton

É moi interesante o directorio "esqueleto", que por defecto é */etc/skel*, aínda que o podemos configurar onde nos interese coa opción *-k*. Este directorio contén arquivos e directorios que serán copiados no directorio *home* dos usuarios creados coa ferramenta *useradd*. Está claro que isto será así cando sexan empregadas as opcións *-m* (ou *--create-home*).

Modificar contas de usuario

Moitas veces teremos que modificar certos parámetros dalgunha conta de usuario, poderémolo facer de varias maneiras: editando directamente ficheiros críticos como */etc/passwd* ou */etc/shadow* onde se almacenan de forma encriptada os contrasinais dos usuarios, modificar ficheiros de configuración específicos do usuario ou empregar utilidades do sistema como as que se empregan para crear contas.

- **Definir un contrasinal:**

  - Ainda que ***useradd*** ofrece o parámetro **-p** para definir un contrasinal, esta ferramenta require un contrasinal pre-encriptado.
  - Por conseguinte, é mais fácil crear unha conta de forma desactivada (sen empregar **-p**) e definir o contrasinal tras crear a conta.
  - Isto podémolo facer có comando passwd ou, como vimos antes, moito mellor có comando chpasswd.
  - Os usuarios normais poden utilizar ***passwd*** para cambiar os seus contrasinais, pero moitos dos seus parámetros só os pode utilizar *root*. En concreto, **-l**, **-u**,**-i** e**-d** quedan fora do alcance dos usuarios normais.
  - Os contrasinais de Linux poden conter letras, números e signos de puntuación. Linux distingue entre maiúsculas e minúsculas nos contrasinais.

- **Empregar a utilidade *usermod*:**

  - O programa usermod é moi similar a *useradd* nas súas funcionalidades e parámetros, ainda que esta utilidade cambia unha conta existente en lugar de crear unha nova.
  - ***usermod*** permite a incorporación dun parámetro **-m** cando se utiliza con **-d**. O parámetro **-d**, por si só, cambia o directorio *home* do usuario, pero non move ningún ficheiro. Se se agrega **-m**, *usermod* moverá os ficheiros do usuario ao novo sitio.
  - Se cambiamos a UID dunha conta, esta acción non cambiará a UID almacenada cós ficheiros do usuario. Como consecuencia, o usuario pode deixar de ter acceso a estes ficheiros. Pódese actualizar manualmente as UID de todos os ficheiros empregando o comando chown. En concreto, un comando como o seguinte, executado tras cambiar a UID da conta *lara*, restaura o propietario correcto dos ficheiros do directorio *home* de *lara*:

```
# Configuramos o usuario "lara" có UID "2000"
```

```
$ usermod -u 2000 lara
# Cambiamos os permisos de acceso para os seus documentos
$ chown -R lara /home2/lara
```

- **Empregar a utilidade *chage*:**

    - O comando chage permítenos modificar os parámetros das contas relacionados coa expiración desta.
    - As contas de Linux se poden configurar para que expiren automáticamente se se cumple cualquera destas condicións:
        · O contrasinal non cambiou nun periodo de tempo específico.
        · A data do sistema sobrepasou o tempo predeterminado.
    - Estes parámetros contrólanse coa utilidade ***chage***, que posúe a seguinte sintaxe:

```
chage [-l] [-m mindías] [-M maxdías] [-d últimodía] [-I díasinactivo] [-E fechaexpiración ] [-W díasaviso] nomeusuario
```

    - A opción **-l** fai que *chage* mostre a información do periodo de expiración da conta e do contrasinal dun usuario concreto.
    - No seguinte exemplo o *password* do usuario lara configúrase para que expire en 10 días dende o último cambio de contrasinal:

```
$ chage -M 10 lara
```

Eliminar contas

Sobre o papel, borrar contas de usuario é fácil. Pode utilizar o comando userdel para realizar a función de eliminar as entradas do usuario de ***/etc/passwd*** e, se o sistema emprega contrasinais ocultas, ***/etc/shadow***. O comando ***userdel*** só recibe tres parámetros cós que conseguiremos:

- **Borrar ficheiros de usuario:** O parámetro **-r** fai que o sistema elimine todos os ficheiros da cola de correo do usuario e do seu directorio *home*.
- **Forzar o borrado:** Pódese forzar o borrado da conta mentres un usuario está dentro do sistema empregando a opción **-f** xunto con **-r**.
- **Obter axuda:** A opción **-h** resume as opcións de ***userdel***.

Vexamos un exemplo no que se elimina a conta *lara*:

```
$ userdel -r lara
```

Có comando anterior tamén se eliminan todos os arquivos do usuario **lara**. Pero, se o usuario ten arquivos gardados por algún sitio do árbol de directorios podemos empregar o comando ***find*** có seu parámetro **-uid** (ou **-user**, se empregamos ***find*** antes de borrar a conta). Por exemplo, se *lara* ten o UID 2000, pódese empregar o seguinte comando para localizar todos os seus ficheiros:

```
$ find / -uid 2000
```

Configurar grupos

Linux ofrece ferramentas de configuración de grupos moi semellantes ás das contas de usuario en moitos aspectos, pero sen esquecer que "un grupo non é unha conta de usuario".

Engadir grupos

Linux proporciona o comando groupadd para engadir novos grupos. Esta utilidade é similar a ***useradd*** pero ten menos opcións.

A sintaxe de ***groupadd*** é a seguinte:

```
groupadd [-g GID [-o]] [-r] [-f] nomegrupo
```

Na maioría dos casos, os grupos créanse sen especificar parámetros, excepto o propio nome do grupo:

```
$ groupadd proxecto3
```

Modificar a información do grupo

A información dos grupos, como a das contas de usuario, pódese modificar empregando utilidades ou ben directamente, editando o ficheiro de configuración ***/etc/group***.

- **Uso de *groupmod* e *usermod***

- O comando groupmod modifica os parámetros dun grupo existente. A súa sintaxe é a seguinte:

```
groupmod [-g GID [-o]] [-n nomenovogrupo] nomeantigogrupo
```

Có parámetro **-o** podemos configurarlle a un grupo un GID que xa pertence a outro grupo.

- Para engadir un usuario a un grupo empregaremos o comando usermod có seu parámetro **-G**.

Por exemplo, o seguinte comando define a *lara* como membro dos grupos *users*, *proxecto1* e *proxecto4* e a elimina de todos os demais grupos:

```
$ usermod -G users,proxecto1,proxecto4 lara
```

Pero, para cambiar o grupo por defecto dun usuario, empregamos o parámetro **-g**:

```
$ usermod -g proxecto2 lara
```

- Pódese saber a que grupos pertence actualmente un usuario có comando groups:

```
$ groups lara
```

- Para evitar a omisión accidental dun grupo, moitos administradores prefiren modificar o ficheiro ***/etc/group*** ou empregar o comando gpasswd. De calquera deses xeitos poderanse engadir usuarios a un grupo sen especificar a que grupos existentes pertence xa.

- **Uso de *gpasswd***

- O comando gpasswd é o equivalente a passwd para os grupos.
- O comando ***gpasswd*** tamén lle permite modificar outras características dos grupos e asignar administradores de grupos: usuarios que poden realizar nestes algunhas funcións administrativas propias dos grupos.
- A sintaxe básica deste comando é a seguinte:

```
gpasswd [-a usuario] [-d usuario] [-R] [-r] [-A usuario[,...]] [-M usuario[,...]] grupo
```

- Engadimos administradores ao grupo có parámetro **-A usuario**.
- Engadimos usuarios ao grupo có parámetro **-a usuario**.
- Se executamos o comando sen mais parámetros que o nome do grupo, ***gpasswd*** cambia o contrasinal do grupo. Os contrasinais de grupo permiten controlar quen é membro do grupo, os membros dun grupo poden empregar newgrp para cambiar o grupo ao que pertencen actualmente, poderán así cambiar de grupo temporalmente e en calquera momento introducindo o contrasinal configurado. Así e todo, ***newgrp*** non está correctamente implementado en todas as distribución.

- **Modificar directamente os ficheiros de configuración dos grupos:**

- A información dos grupos almacénase fundamentalmente no ficheiro ***/etc/group***.
- Nel teremos unha liña por grupo. Como xa vimos antes, unha liña típica deste ficheiro ten este aspecto:

```
# Nome do grupo:Contrasinal oculta:GID:Lista de usuarios
project1:x:501:lara,rafa,andrea,daniel
```

- Os usuarios tamén poden ser membros dun grupo en base á especificación de grupo primario do seu ficheiro ***/etc/passwd***.
- Os sistemas con contrasinais ocultas utilizan outro ficheiro, ***/etc/gshadow***, para almacenar a información dos contrasinais ocultos dos grupos.

Eliminar grupos

- Os grupos se eliminan có comando groupdel, que recibe un único parámetro: o nome do grupo.

Vexamos un exemplo no que eliminamos o grupo ***proxecto3***:

```
$ groupdel proxecto3
```

- Tamén se pode eliminar un grupo editando o ficheiro ***/etc/group*** (e ***/etc/gshadow***).

Como ocorre ao borrar usuarios, o borrado de grupos pode deixar ficheiros orfos no ordenador có comando ***find***. Vexamos un exemplo:

```
$ find / -gid 503
```

Despois de atopar algún arquivo que pertenza ao grupo borrado o normal é borrar os ficheiros ou asignarlles outro grupo empregando os comandos chown ou chgrp.

Acceso dos usuarios con privilexios de *root*

Ademais de entrar no *login* do sistema como *root*, hai dúas formas de ampliar os privilexios dun usuario e adquirir os de *root*.
Para facer isto empregaremos os comandos su e sudo.

O comando *su*

O comando su fai que un usuario que se identificou coa súa propia conta poida cambiar o seu **uid** ao de *root*.
Por suposto debe saber o *password* do *root*.

O comando *sudo*

O comando sudo permítenos executar un comando como outro usuario. Así, neste caso non é necesario coñecer o contrasinal de *root* para executar certos comandos de administración.
A relación de usuarios e comandos permitidos para eles deben de estar no ficheiro ***/etc/sudoers*** (Axuda de sudoers).
Por exemplo, para que o usuario **lara** poida facer *shutdown* do sistema debe existir unha entrada no ficheiro ***sudoers*** como:

```
...
# Cmnd alias specification
Cmnd_Alias SHUTDOWN_CMDS = /sbin/shutdown, /sbin/halt, /sbin/reboot
...
%admin ALL = (ALL) ALL
lara ALL=(ALL) NOPASSWD: SHUTDOWN_CMDS
```

Logo "lara" poderá apagar o equipo empregando o comando:

```
$ sudo shutdown -h now
```

Ou reinicialo empregando o comando:

```
$ sudo shutdown -r now
```

Axuda de *visudo*

Para modificar o arquivo ***sudoers*** hai que empregar a utilidade visudo. Simplemente con escribir ***visudo*** na liña de comandos o arquivo ***sudoers*** abrirase para editalo e configuralo.

En ubuntu, por defecto, todos os usuarios que pertencen ao grupo ***admin*** teñen permisos de *root*.
A liña que indica iso no arquivo ***sudoers*** é a seguinte:

```
# Members of de admin group may gain root privileges
%admin ALL=(ALL) ALL
```

Neste caso:

- %admin: grupo que obtén os privilexios. Neste caso o grupo *admin*.
- ALL: en que equipos. Neste caso en todos.
- (ALL): de que usuarios gaña os privilexios: Neste caso de todos. Este campo é opcional.
- ALL: que comandos pode executar. Neste caso todos.

Outras accións posibles có comando sudo son as seguintes:

```
# Descubrir que comandos podemos executar empregando sudo:
```

```
$ sudo -l
# Descubrir pode executar un usuario calquera,
# sempre e cando o fagamos dende root ou un usuario con ALL privilexios:
$ sudo -l -U usuario
# Para eliminar as nosas credenciales en sudo:
$ sudo -k
# A seguinte vez que empreguemos sudo nos pedirá outra vez o contrasinal.
# Para executar a Shell como o usuario proba:
$ sudo -u proba
# Para executar a Shell como o usuario proba e ademais configurar este
# tal e como o vería ese usuario (directorio, aparencia...):
$ sudo -u proba -i
# Para executar a Shell como o usuario root:
$ sudo su
# Para executar a Shell como o usuario root e ademais configurar este
# tal e como o vería root (directorio, aparencia...):
$ sudo su -
```

Sudo gráfico

Non se debe usar **sudo** para iniciar aplicacións gráficas como root. No seu lugar débese usar **gksudo** (**kdesudo** en Kubuntu). O comando **gksudo** establece o directorio HOME a /root. Isto evita que se creen ficheiros nos directorios dos usuarios onde o dono é o root.