

Laboratorio 02: Gestión de Usuarios



Asignatura: Administración de Redes

Nombre: Carlos Alberto Espinoza Mansilla

INDICE

ACTIVIDAD 7.....3

✓ P1.....3

✓ P2.....4

✓ P3.....5

✓ P4.....5

✓ P5.....6

✓ P6.....7

ACTIVIDAD 8.....7

✓ P1.....7

✓ P2.....8

✓ P3.....9

ACTIVIDAD 9.....10

✓ P1.....10

✓ P2.....10

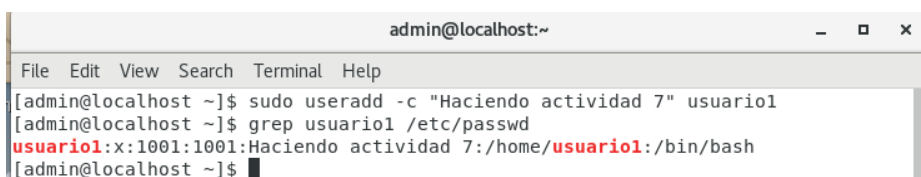
✓ P3.....11

Actividad 7

1.El comando *useradd* crea una cuenta y añade las entradas necesarias en *passwd*, *shadow* y *group*, además del directorio de usuario. Opciones importantes para definir la cuenta (ver *man useradd*):

- c comentario (sección información)
- e fecha de expiración
- f días para que se bloquee la cuenta después de que caduque la contraseña
- g grupo principal (por defecto creará uno, ver *USERGROUPS_ENAB* en *login.defs*)
- G grupos adicionales
- m crea el directorio *home* del usuario
- s *shell*

Crear varias cuentas de usuario con diferentes opciones. Comprobar el contenido de *passwd*, *groups* y *shadow*. Explique la salida de cada una de ellas.



```
admin@localhost:~  
File Edit View Search Terminal Help  
[admin@localhost ~]$ sudo useradd -c "Haciendo actividad 7" usuario1  
[admin@localhost ~]$ grep usuario1 /etc/passwd  
usuario1:x:1001:1001:Haciendo actividad 7:/home/usuario1:/bin/bash  
[admin@localhost ~]$
```

Aquí usamos *useradd -c usuario1* para crear el usuario “*usuario1*” y agregar el comentario “*Haciendo actividad 7*”, luego vemos que la información de ese comentario esta guardada en */etc/passwd* por lo que lo buscamos dentro usando *grep*.

```
[admin@localhost ~]$ sudo useradd -e 2019-06-06 usuario2
[sudo] password for admin:
[admin@localhost ~]$ sudo chage -l usuario2
Last password change                : Apr 07, 2019
Password expires                    : never
Password inactive                   : never
Account expires                     : Jun 06, 2019
Minimum number of days between password change : 0
Maximum number of days between password change : 99999
Number of days of warning before password expires : 7
[admin@localhost ~]$
```

Luego creamos un “*usuario2*” usando la opción `-e` para agregarle una fecha de expiración la cual será *2019-06-06* con el formato AÑO-DÍA-MES luego con el comando *chage* y la opción de listado `-l` verificamos su fecha de expiración.

```
[admin@localhost ~]$ sudo useradd -g usuario1 usuario3
[sudo] password for admin:
[admin@localhost ~]$ id usuario3
uid=1003(usuario3) gid=1001(usuario1) groups=1001(usuario1)
[admin@localhost ~]$ ~
```

Ahora usando `-g` a *useradd* podemos especificar que al crear un nuevo usuario este se agregue a un grupo ya existente, en este caso como por default cuando se crea un usuario se crea un grupo del mismo nombre agregaremos al nuevo “*usuario3*” al grupo “*usuario1*”, finalmente chequeamos con el comando *id* a que grupo pertenece nuestro nuevo usuario.

2.Las contraseñas se pueden asignar con el comando *passwd*. Un usuario puede cambiar su propia contraseña:

- cambiar la contraseña de un usuario con ese mismo usuario (*passwd*, sin opciones).
- poner una contraseña a las cuentas creadas en el ejercicio anterior.
Comprobar los cambios en el fichero *shadow*.

```
[admin@localhost ~]$ sudo passwd usuario1
[sudo] password for admin:
Changing password for user usuario1.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[admin@localhost ~]$ sudo passwd usuario2
Changing password for user usuario2.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[admin@localhost ~]$ sudo passwd usuario3
Changing password for user usuario3.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[admin@localhost ~]$
```

Cambiamos las contraseñas para usuario 1, 2 y 3, nos pide ingresar la contraseña 2 veces y se graba.

```
[admin@localhost ~]$ sudo grep usuario /etc/shadow
[sudo] password for admin:
usuario1:$6$Lar7n0rz$rsld43YH06IVWd.93dLHgqZB3Qmv1U6TcqVLhbvAJk8aIcrhdF7qEkjhL2Zoby.KZCseQhvaiBIwmf4gdukML1:17993:0:99999:7:::
usuario2:$6$9QYq3WF1$HNApKnd2fQcfRJlRyZAxV4E8EgDzAakFSGEET.AAdagbAvn.pLCbIJN24clVkAxpEjWKL323gb.jE/lR0KT380:17993:0:99999:7::18053:
usuario3:$6$cIPLnAAq$LK7By8nwdT7uiUj.p9/q5qjNAXisJsnYPLZ9zqdB481j.4J4hd6xKT7y556cB07PWKRfdfKQVIBPBHPnfRlLT0:17993:0:99999:7:::
[admin@localhost ~]$
```

Buscamos dentro del fichero shadow en la dirección /etc/shadow usando grep y observamos las contraseñas de las cuentas que hemos creado pero cifradas además de otros datos de interés en el formato: <nombre><password cifrado><1><2><3><4><5><6>

- 1-Días transcurridos desde 1-1-1970 donde el password fue cambiado por última vez.
- 2-El mínimo número de días entre cambios de contraseña.
- 3-Días máximos de validez de la cuenta.
- 4-Días que avisa antes de caducar la contraseña.
- 5- Días después de que un password caduque para deshabilitar la cuenta
- 6- Fecha de caducidad. días desde 1-1-1970, donde la cuenta es deshabilitada y el usuario no podrá iniciar sesión.

Información sacada de la web: <http://www.nexolinux.com/ficheros-de-usuarios-etcpasswd-y-etcshadow/>

3. El comando groupadd crea nuevos grupos. Crear un par de grupos uno de ellos con el GID 60002.

```
[admin@localhost ~]$ sudo groupadd -g 60002 grupo1
[sudo] password for admin:
[admin@localhost ~]$ sudo groupadd -g 60003 grupo2
[admin@localhost ~]$ grep grupo /etc/group
grupo1:x:60002:
grupo2:x:60003:
[admin@localhost ~]$
```

Creamos 2 grupos llamados “grupo1” y “grupo2” con los GID 60002 y 60003, lo verificamos viendo el archivo /etc/group.

4. Para modificar una cuenta de usuario se usa el comando usermod:

- Deshabilitar una de las cuentas creada cambiando su shell.
- Añadir una de las cuentas creadas a uno de los nuevos grupos (notar la diferencia entre -g y -G y la opción -a).

De la misma forma se puede modificar un grupo con groupmod (consultar su página de manual).

```
[admin@localhost ~]$ sudo usermod -s /sbin/nologin usuario1
[sudo] password for admin:
[admin@localhost ~]$ su usuario1
Password:
This account is currently not available.
[admin@localhost ~]$
```

Para deshabilitar un usuario modificando su Shell debemos cambiar el nombre de su Shell de inicio de sesión por nologin que está en /sbin/nologin eso lo hacemos con el comando: *usermod -s /sbin/nologin <nombredeusuario>*. Luego verificamos que no se puede ingresar a dicha cuenta.

```
[admin@localhost ~]$ sudo usermod -g grupo1 usuario2
[sudo] password for admin:
[admin@localhost ~]$ id usuario2
uid=1002(usuario2) gid=60002(grupo1) groups=60002(grupo1)
[admin@localhost ~]$ sudo usermod -G grupo1,grupo2 usuario2
[admin@localhost ~]$ id usuario2
uid=1002(usuario2) gid=60002(grupo1) groups=60002(grupo1),60003(grupo2)
```

La opción *-g* define un grupo de inicio de logeo para el usuario, *-G* establece a que grupos suplementarios pertenece el usuario borrando cualquier otro grupo que no se mencione durante el comando y la opción *-a* se usa en combinación con *-G* para anexar más grupos suplementarios sin la necesidad de escribir aquellos a los que ya pertenece el usuario.

5. El comando `last` nos permite ver qué usuarios están activos, o no, y desde cuándo. Verifique este comando con los usuarios anteriores.

```
[admin@localhost ~]$ last
admin pts/1 :0 Sun Apr 7 15:39 still logged in
admin pts/0 :0 Sun Apr 7 13:15 still logged in
admin :0 :0 Sun Apr 7 12:20 still logged in
reboot system boot 3.10.0-957.10.1. Sun Apr 7 12:20 - 17:40 (05:20)
admin pts/0 :0 Sun Mar 31 21:46 - 22:33 (00:47)
admin pts/0 :0 Sun Mar 31 20:05 - 21:46 (01:40)
admin pts/0 :0 Sun Mar 31 17:36 - 20:05 (02:29)
admin pts/0 :0 Sun Mar 31 13:27 - 17:36 (04:09)
admin pts/0 :0 Sun Mar 31 13:21 - 13:27 (00:05)
admin pts/1 :0 Sun Mar 31 13:15 - 13:21 (00:05)
admin pts/0 :0 Sun Mar 31 12:00 - 13:21 (01:20)
admin pts/0 :0 Sun Mar 31 11:34 - 11:59 (00:25)
admin :0 :0 Sun Mar 31 11:33 - crash (7+00:46)
reboot system boot 3.10.0-957.10.1. Sun Mar 31 11:28 - 17:40 (7+06:11)
admin :0 :0 Sat Mar 30 18:59 - crash (16:29)
reboot system boot 3.10.0-957.10.1. Sat Mar 30 18:58 - 17:40 (7+22:41)
admin pts/2 :0 Mon Mar 18 18:57 - 19:56 (00:59)
admin pts/1 :0 Mon Mar 18 18:52 - 19:56 (01:03)
admin pts/0 :0 Mon Mar 18 18:31 - 19:56 (01:24)
admin :0 :0 Mon Mar 18 18:24 - down (01:32)
reboot system boot 3.10.0-957.el7.x Mon Mar 18 18:22 - 19:56 (01:33)

wtmp begins Mon Mar 18 18:22:50 2019
[admin@localhost ~]$
```

6. Se pueden borrar las cuentas con `userdel` y `groupdel`, consultar las opciones (especialmente `-r` para `userdel`). Probar estos comandos con algunos de los usuarios inactivos vistos en el punto anterior.

```
[admin@localhost ~]$ ls /home
admin prueba usuario1 usuario2 usuario3
[admin@localhost ~]$ sudo userdel -r usuario1
[admin@localhost ~]$ ls /home
admin prueba usuario2 usuario3
[admin@localhost ~]$
```

La opción `-r` te permite remover la carpeta `/home/nombredeusuario`.

Actividad 8

1. Puede ser necesario permitir el acceso a root al sistema, aunque se puede restringir los terminales desde los que se puede hacer login. El fichero `/etc/securetty` especifica que terminales son seguros para root:

- Hacer una copia del fichero.
- Dejar solo tty3 y probar su comportamiento.

```
[admin@localhost ~]$ sudo cp /etc/securetty /home/admin/
[admin@localhost ~]$ ls /home/admin
Desktop      man-db-2.6.3-11.el7.x86_64.rpm      Music      securetty
Documents    mariadb-5.5.60-1.el7_5.x86_64.rpm    Pictures   Templates
Downloads    mariadb-libs-5.5.60-1.el7_5.i686.rpm Public     Videos
```

El archivo /etc/securetty tiene todos los dispositivos que puede conectarse a root si solo se deja a tty3 entonces solo él podría conectarse a root

En tty2(Ctrl+F2):

```
CentOS Linux 7 (Core)
Kernel 3.10.0-957.10.1.el7.x86_64 on an x86_64

localhost login: root
Password:
Login incorrect

localhost login:
```

En tty3(Ctrl+F3):

```
CentOS Linux 7 (Core)
Kernel 3.10.0-957.10.1.el7.x86_64 on an x86_64

localhost login: root
Password:
Last failed login: Sun Apr  7 18:15:26 EDT 2019 on tty2
There was 1 failed login attempt since the last successful login.
Last login: Sun Apr  7 18:14:49 on tty3
[root@localhost ~]#
```

Información sacada de la web: <https://maslinux.es/como-restringir-el-acceso-de-los-usuarios-en-una-maquina-gnulinux/>

2. Además de /etc/securetty para root, está el fichero /etc/security/access.conf que configura que usuarios y en que terminales pueden entrar al sistema. Cada entrada determina (+/-) habilita/deshabilita el acceso de un grupo o conjunto de usuarios al sistema desde una terminal o host (-:ALL EXCEPT root:tty1). Observar el contenido del fichero y explicar algún otro aspecto importante que observe.


```
#####
#
# Disallow non-root logins on tty1
#
#-:ALL EXCEPT root:tty1
#
# Disallow console logins to all but a few accounts.
#
#-:ALL EXCEPT wheel shutdown sync:LOCAL
#
# Same, but make sure that really the group wheel and not the user
# wheel is used (use nodegroup argument, too):
#
#-:ALL EXCEPT (wheel) shutdown sync:LOCAL
#
# Disallow non-local logins to privileged accounts (group wheel).
#
#-:wheel:ALL EXCEPT LOCAL .win.true.nl
#
# Some accounts are not allowed to login from anywhere:
#
#-:wsbscaro wsbscr wsbspac wsbsym wscosor wstaiwde:ALL
#
# All other accounts are allowed to login from anywhere.
#
#####
```

-: ALL EXCEPT Wheel shutdown sync: LOCAL

No permite inicios de sesión de consola para nadie excepto por shutdown, sync y Wheel.

-: ALL EXCEPT (Wheel) shutdown sync: LOCAL

Lo mismo pero esta vez permite que aquellos dentro del grupo wheel y no solo el usuario wheel tengan acceso.

-: wheel:ALL EXCEPT LOCAL .win.true.nl

No permite que ninguna cuenta con privilegios del grupo wheel accede a menos que sea LOCAL.

-:wsbscaro wsbscr wsbspac wsbsym wscosor wstaiwde: ALL

No permite que ninguna de estas cuentas accede sin importar nada.

3. El comando su, permite cambiar de usuario y requiere conocer la contraseña de la cuenta destino. Normalmente se usa la orden sudo, que permite acceder a los usuarios a comandos de administración con su propia password:

- El fichero de configuración es /etc/sudoers y se edita con visudo
- Observar el fichero y la sintaxis empleada (usuario máquina=comandos). ¿Qué significan las entrada:
 - ✓ root ALL=(ALL) ALL
 - ✓ %sys ALL = NETWORKING, SOFTWARE
 - ✓ %wheel ALL=(ALL) NOPASSWD: ALL
- Dar permisos al usuario para ejecutar cualquier comando sin contraseña.
- Comprobar el comportamiento, reiniciando el servicio sshd mediante sudo.
- Como su usuario cambiar al usuario root usando sudo y la opción -i. Una vez que podemos cambiar a root con su usuario, deshabilitar el acceso con contraseña a root.

Así mismo, tenemos otros ficheros importantes en esta carpeta como “time.conf”, “limits.conf” y “group.conf” y “capability.conf”. Explíquelos y exponga un ejemplo de configuración con su salida obtenida.

root ALL=(ALL) ALL: Permite a root ejecutar cualquier comando desde cualquier lugar.

%sys ALL = NETWORKING, SOFTWARE: Permite a los usuarios del grupo “sys” correr los programas networking, software.

%wheel ALL=(ALL) NOPASSWD: ALL: Permite a los usuarios del grupo “wheel” correr todos los comandos sin necesitar contraseña.

```
##
##      user      MACHINE=COMMANDS
##
## The COMMANDS section may have other options added to it.
##
## Allow root to run any commands anywhere
root    ALL=(ALL)      ALL
admin   ALL=(ALL)      NOPASSWD:ALL
```

Agregando admin para ejecutar cualquier comando sin contraseña.

```
[admin@localhost ~]$ sudo service sshd restart
Redirecting to /bin/systemctl restart sshd.service
[admin@localhost ~]$ █
```

Reiniciando sshd.

```
# To disable tunneled clear text passwords, change to no here!
#PasswordAuthentication yes
#PermitEmptyPasswords no
PasswordAuthentication no
```

Deshabilitar acceso por contraseña

Actividad 9

1. Consultar el contenido del directorio /etc/skel, que contiene los archivos que se copian cuando se crea una cuenta de usuario (.bashrc, .bash_profile, .bash_logout...) y explicar su contenido.

Bashrc : Cada vez que se crea un nuevo usuario /etc/skel/.bashrc se copia a su carpeta respectiva en home/username y es usado cada vez que dicho usuario abre la Shell, además puedes escribir scripts en /home/skel/.bashrc para que cada vez que se ejecute la Shell con dicho usuario se ejecute el script.

Bash_profile: cuando un usuario entra a la Shell se lee este archivo puede establecer variables de entorno al entrar al iniciar la Shell.

Bash_logout: se lee siempre que el usuario correspondiente sale de la Shell.

2. El fichero /etc/bashrc contiene definiciones y configuraciones globales, se carga desde la configuración de usuario (.bashrc); estudiar su comportamiento.

El fichero `/etc/bashrc` al iniciar la Shell verifica que sea un interprete interactivo y luego llama al archivo de conf. `Home/username/.bashrc`. A través de este archivo se puede configurar globalmente el un script que quieras que se active al iniciar la Shell puedes por ejemplo poner alias a ciertos comandos o configurar permisos por default para crear directorios, etc.

3. Finalmente `/etc/profile` y `/etc/profile.d` contienen la configuración global del entorno. Observar el contenido del fichero `profile` (`PATH,USER, HOSTNAME...`) y el contenido de algunos de los ficheros en `/etc/profile.d` (e.g. `colorls.sh`).

`/etc/profile` se lee al iniciar la Shell para asegurarse de que el entorno sea correcto para el interprete bash y cargar las configuraciones que estén por defecto en `/home/username/.bash_profile` dentro de `/etc/profile.d` están archivos de configuración de la Shell por ejemplo `colorls.sh` controla los colores específicos designados para las variables dentro del Shell.

Información sacada de la página web: <http://www.escomposlinux.org/lfs-es/blfs-es-5.0/postlfs/profile.html>