Administración Linux

Administración Usuarios

Roberto Gómez Cárdenas
rogomez@itesm.mx
http://homepage.cem.itesm.mx/rogomez

Fecha última modificación: abril 2007

El archivo /etc/passwd

- Archivo ASCII manipulable con un editor
- Debe poder ser leído por todos los usuarios para ciertos comandos
- A cada usuario le corresponde una entrada
- Los programas realizan una búsqueda secuencial de las entradas (no vale la pena ordenar las entradas)
- Los campos de cada entrada están separados por carácter de dos puntos (:)

Campos de las entradas

- El identificador del usuario
- El password del usuario
- En algunas versiones: información sobre fecha del último cambio del password y sobre el periodo para realizar dichos cambios; dicha información separada por comas
- Un valor numérico, de 0 a 6000 que representa el UID del usurario

Lámina 3 Roberto Gómez Cárdenas

- Otro valor numérico, que no pase de 600, que representa el GID
- Un campo de comentarios conocido como GECOS
- El directorio hogar
- El shell de inicio

Lámina 4 Roberto Gómez Cárdenas

Ejemplo archivo /etc/passwd

xetaboada:ypK2awu1hBqGs:1326:41:Eunice Taboada Ibarra:/home/dacs/etaboada:/bin/csh
dgonzale:dU8MloKM7Af8Y:10106:41:John Lucien Gonzalez:/home/dacs/dgonzale:/bin/csh
abermude:Fe5/I/SHg53HM:2404:43:Adriana Diaz B Pagos:/home/prepa/abermude:/bin/csh
sa448020:iqC7X.6SUEASE:1832:215:David Bernal G Di Soporte:/home/sap/sa448020:/bin/csh
rcaballe:j3KODtAuQ8uEQ:8773:41:Ricardo Caballero Valdes:/home/dacs/rcaballe:/bin/csh
csanchez:YYoHlXDeYHanM:1212:43:Concepcion Sanchez:/home/prepa/csanchez:/bin/csh
sduenas:1ube95PeMQZOQ:10140:41:Lic. Sergio F Rodriguez:/home/dacs/sduenas:/bin/csh
rperrin:rKWggQip3DIHQ:10021:44:Rafael Fausto:/home/dae/rperrin:/bin/csh
gperrin:Bj87cqMfSXzmc:10012:44:Graciela Patricia:/home/dae/gperrin:/bin/csh
rvilla:4McraxhY8AVB6:8839:43:Rafael Villa:/home/prepa/rvilla:/bin/csh
lvelio:IifTeZS98v/H.:1248:41:Lucrecia Velio-mejia:/home/dacs/lvelio:/bin/csh
tpacheco:UNbyYZ.dNCY3.:10275:510:Tito Omar:/home/dia/tpacheco:/bin/csh
jorozco:QOdtJnflY.1.s:3656:206:Jorge Orozco S:/home/unicom/jorozco:/bin/csh
bmerced:FzniebygQZSRs:1613:510:Bernando Isidro Merced S Dia :/home/dia/bmerced:/bin/csh
amoreno:1voQAFGgLpxWg:5161:40:Asuncion Moreno 3122:/home/dia/amoreno:/bin/ksh
mahernan:AxBSyYy/tiHM6:1166:203:Magdalena Hernandez S:/home/dsa/mahernan:/bin/csh
aantunan:nhd5kmfXoGVP.:8937:41:Alma L Antunano Arias Dacs:/home/dacs/aantunan:/bin/csh

Lámina 5 Roberto Gómez Cárdenas

Peligro del archivo password

- En algunas ocasiones el archivo es accesible en lectura para todo usuario.
  - existen otros métodos: NIS
- Esto permite a un cracker el obtener una copia de este archivo y dedicarse a descifrarlo en la comodidad de su casa.
  - ataque de diccionario
- Bajo esta premisa, algunos sistemas han propuesto un mecanismo llamado *shadow password file*.

Lámina 6 Roberto Gómez Cárdenas

Archivo shadow

- Los passwords encriptados no se almacenan en el archivo *passwd* sino en un archivo llamado *shadow*
- En el archivo de passwords, se reemplaza la contraseña por un caracter
  - indica al sistema que verifique contra el archivo shadow
- Al no tener acceso a los passwords encriptados, un atacante tendrá mayor oposición para descifrar un password.
- Desafortunadamente, en ocasiones se olvida prevenir que un archivo shadow sea públicamente accesible.

Lámina 7 Roberto Gómez Cárdenas

¿Donde se encuentra el archivo?

System	Shadow	Token
AIX	/etc/security/passwd	!
BSD	/etc/master.passwd	*
DG/UX	/etc/tcb/aa/user/	*
HP-UX	/.secure/etc/passwd	*
IRIX	/etc/shadow	x
Linux	/etc/shadow	*
SCO	/tcb/auth/files/[first letter of username]/[username]	*
SunOS4.1+c2	/etc/security/passwd.adjunct	##username
SunOS 5.x	/etc/shadow [optional NIS+ private secure maps/tables]	##username
System V < 4.2	/etc/shadow	x
System V >= 4.2	/etc/security/* database	x

Lámina 8 Roberto Gómez Cárdenas

Archivo /etc/passwd con shadow

```
root:x:0:0:root:/root:/bin/bash
user1:x:500:500:usuario1:/home/user1:/bin/bash
user2:x:501:501:usuario2:/home/user2:/bin/bash
user3:x:502:502:usuario3:/home/user3:/bin/bash
solovino:x:504:504:Perro:/home/solovino:/bin/bash
sshd:x:505:505:usuario sshd:/home/sshd:/bin/bash
```

Lámina 9 Roberto Gómez Cárdenas

El archivo /etc/shadow correspondiente

```
root:$1$M2dso4qm$2mU2HAhMsKTXBLTnj/KQ5.:12025:0:99999:7:::
user1:$1$TGBAB7Ri$o1KEDjDYKDldG97eBVFdB0:12564:0:99999:7:::
user2:$1$èÃQÉHÇUP$qlc0Q51vkFE1HipprGmE00:11937:0:99999:7:::
user3:$1$sÍÓâkqôe$jwwUR/zokS0pgGWRcPGEo1:11937:0:99999:7:::
solovino:$1$I/Na5oTM$7FDUJfBd79oTO77xNqdiN.:12511::99999::::
sshd:$1$tIm2sbv7$DP9S5R0Y0ZJf5cSluvHK41:12512::99999::::
```

Lámina 10 Roberto Gómez Cárdenas

Envejecimiento de passwords

- Otra mecanismo de protección de passwords, de Shadow Password, es el de envejecimiento de passwords
  - Aging Password
- La idea básica es proteger los passwords de los usuarios dándoles un determinado periodo de vida:
  - sólo va a ser válido durante un cierto tiempo, pasado el cual expirará y el usuario deberá cambiarla.
- Los periodos de expiración de las claves se suelen definir a la hora de crear a los usuarios con las herramientas que cada sistema ofrece para ello.

Lámina 11 Roberto Gómez Cárdenas

Campos /etc/shadow

```
login:password:lastchg:min:max:warn:inactive:expire:flag
```

- Nombre de la cuenta
- password: del usuario
- El numero de días transcurridos, desde 1 ene 1970, que el password fue cambiado
- El número de días que el usuario tiene que esperar antes de cambiar su password
  - no lo puede cambiar antes de ese periodo de tiempo
- El número de días tras los cuales el usuario debe cambiar su password

Lámina 12 Roberto Gómez Cárdenas

Los otros campos /etc/shadow

```
login:password:lastchg:min:max:warn:inactive:expire:flag
```

- Número de días anteriores a que el usuario debe notificarse para que cambie su password
- Días que la cuenta estará habilitada tras la expiración de su password
- El número de días, desde el 1o. de enero de 1970, hasta que la cuenta se deshabilite.
- El último campo esta reservado para otros usos
- Ejemplo:

```
example:$1$7RhT4hhG$K3fEau5Tn..uEJUq8tjEn/:11109:0:
99999:7:-1:-1:1075502268
```

Lámina 13 Roberto Gómez Cárdenas

Encripción de password

- Los passwords se almacenan encriptados mediante un algoritmo llamado crypt().
- Algoritymo crypt() está basado en DES.
- Usa una llave de 56 bits (8 caracteres ASCII).
- Utiliza una variación del DES que usa el password introducido por el usuario como llave para encriptar un bloque de 64 bits incializado en cero.
- El resultado se vuelve a encriptar con el password hasta un total de 25 veces.

Lámina 14 Roberto Gómez Cárdenas

Encriptando passwords

- Resultado final (bloque 64 bits) se empaqueta en un string de 11 caracteres.
- Este string se almacena en el archivo /etc/passwd.
- Cada uno de los caracteres contiene 6 bits del bloque obtenido como resultado del encriptado.
- Los caracteres usados son: “.” “/” 0-9, A-Z, a-z
- *Es una función de un solo sentido.*

Lámina 15 Roberto Gómez Cárdenas

Saltos

- Para hacer más robusto el algoritmo, se le añade un número de 12 bits (entre 0 y 4,095), obtenido del tiempo del sistema.
- Este número se le conoce como salto.
- El salto es convertido en un string de dos caracteres y es almacenado junto con el password en el archivo /etc/passwd ocupando los dos primeros lugares.
- Cuando se teclea el password este es encriptado con el salto, ya que si usa otro, el resultado obtenido no coincidiría con el password almacenado.

Lámina 16 Roberto Gómez Cárdenas

Ejemplo passwords y saltos

Password	Salto	Password Encriptado
My+Self	oZ	**oZ**sV5zgRK6sjw
vaLgLo	Na	**Na**WyhsolA2gTM
ATSw.IM!	Hc	**Hc**LrEM.BYtLwk
Global	Gi	**Gi**RzWzP5IEPM
Global	DY	**DY**meXoTgacmWY
Global	pd	**pd**OTBzon3G2KU

Lámina 17
Roberto Gómez Cárdenas

Detalles de los saltos

- Con el salto cada password puede ser encriptado de 4096 formas distintas.
- Esto provoca que ataque por diccionario sea más lento.
- Si se realiza el ataque sobre un sólo usuario el salto no influye, ya que se conoce de antemano.
- Se puede dar el caso (muy remoto) de que un un password y un salto distinto generen un resultado idéntico.

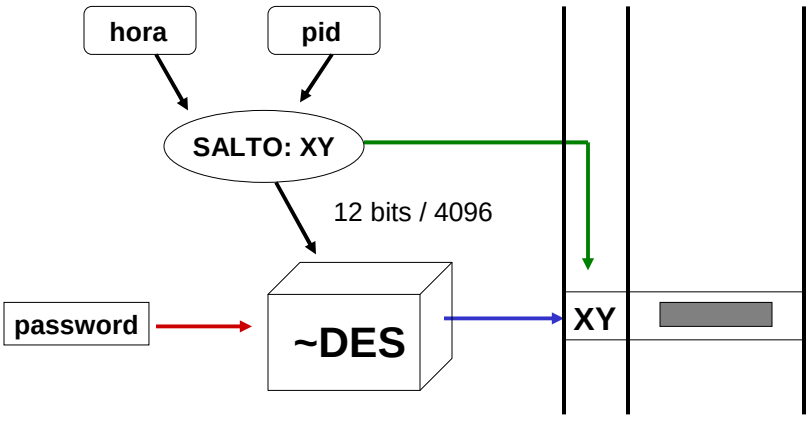
Lámina 18
Roberto Gómez Cárdenas

Ejemplo coincidencia passwords

Password	Salto	Password Encriptado
2NGGMda3 gnB9Gw1j	Hx s8	**Hx**yX8CL2luKyI **s8**yX8CL2luKyI

Lámina 19 Roberto Gómez Cárdenas

Encripción de un password



/etc/passwd

Lámina 20 Roberto Gómez Cárdenas

Verificación de un password

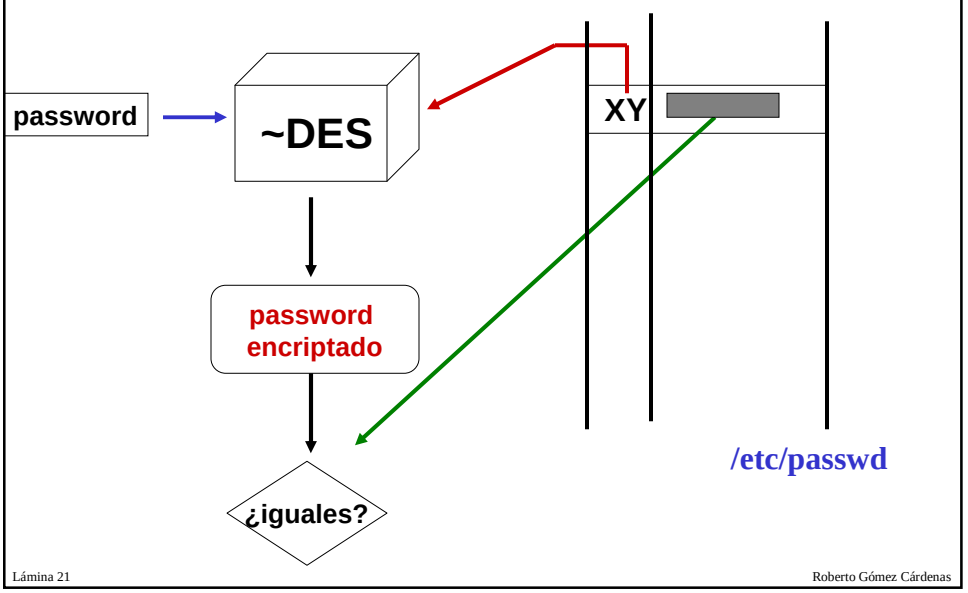


Lámina 21
Roberto Gómez Cárdenas

Alternativas encripción

- Los primeros caracteres denotan el tipo de encripción
  - Encripción DES:

    fp:i6v76dyNQzwjA:1007:1007::0:0:Bogus Name:/home/fp:/bin/ksh93
  - Encripción Blowfish

    fp:$2a$04$.d4.6FZpPIj9GC6DRIRDUuJhPWGP059OmLP2IxSgTQ11LWHVGxxbu:1007:1007::0:0:Bogus
  - Encripción MD5

    fp:1cdTdrg6t$mk4TW.xk15XFoygp1S3UQ1 :1007:1007::0:0:Bogus
- Posible contener el aging sin shadow

  voyager:5fg63fhD3d,M.z8:9406:12:voyager:/home/voyager:/bin/bash
  - carácter 1: número máximo semanas password es válido
  - carácter 2: número mínimo semanas password es válido
  - carácter 3 y 4: última vez password fue cambiado

Lámina 22
Roberto Gómez Cárdenas

Arranque shell: ejemplo bash

Arranque del shell bash

Leer y ejecutar comandos en archivo */etc/profile* → ejecuta scripts en directorio: /etc/profile.d

shell conexión (/bin/login + /etc/passwd) ↔ Busca archivos *$HOME/.bash_profile*

shell no conexión (/usr/X11R6/bin/xterm o /bin/bash) ↔ *$HOME/.bashrc* → ejecuta /etc/bashrc

los busca en ese orden, leyendo y ejecutando dependiendo si se trata de un shell de conexión o no conexión

Lámina 23 Roberto Gómez Cárdenas

Archivos inicializacion del shell

Shell	Archivo	Uso
b-shell	.profile	ejecuta comandos cada vez que un shell tipo bourne (sh, ksh, o bash) es lanzado
c-shell	.cshrc	ejecuta comandos cada vez que un shell tipo C (csh, tcsh) es lanzado
	.login	comandos de inicialización de shells tipo C, que son ejecutados una vez, al inicio de una sesión interactiva
k-shell	.profile	igual que b-shell
tc-shell	.cshrc	igual que c-shell
	.login	igual que c-shell
bash	/etc/profile .bash_profile .bash_login	Uno de estos archivos es ejecutado, solo cuando el usuario se “loguea” (login shell)
	.bashrc	es un non-login-shell, se lee una vez logueado.

Lámina 24 Roberto Gómez Cárdenas

La variable ambiente PATH

- La variable define donde se van a buscar los comandos que el usuario introduzca.
- Muchos usuarios introducen un “.” en la variable, generalmente al principio.
- Esto permite que el usuario desarrolle sus propios programas y los ejecute facilmente, aunque tengan el mismo nombre de un comando.
- Ejemplo variable:

```
$echo $PATH
.:/usr/local:/usr/ucb/:/usr/bin/:/usr/etc/:usr/local/bin
$
```

Lámina 25 Roberto Gómez Cárdenas

Problemas con variable ambiente PATH

- Desafortunadamente lo anterior permite que un intruso cree facilmente caballos de troya
- Alguien puede crear un programa llamado ls y dejarlo en algún directorio donde la víctima trabaja.
- Si la victima tiene un “.” al principio de su variable PATH, cuando teclee ls se ejecutará el programa del intruso en lugar del comando del mismo nombre.
- Solución:
  - no incluir un “.” en la variable
  - incluir el “.” al final de la variable

Lámina 26 Roberto Gómez Cárdenas

Alta y modificación usuarios

comandos y como se hace

Añadiendo usuarios

- Antes de crearla
  - usuario debe firmar, con todo y fecha, una copia de acuerdo de la política de seguridad
- Requerimientos
  - tres pasos requeridos por el sistema
  - dos pasos que establecen un ambiente útil para el nuevo usuario
  - varios pasos extras para facilitar la tarea del administrador

Pasos alta usuario

- Pasos requeridos
  - editar archivo passwd y shadow para definir cuenta usuario
  - asignar un password inicial
  - crear, chown y chmod directorio hogar usuario
- Pasos para el usuario
  - copiar archivos inicialización en directorio hogar usuario
  - establecer directorio correo y alias
- Pasos para el administrador
  - añadir usuario al archivo /etc/group
  - configurar cuotas discos
  - verificar cuenta esta bien configurada
  - añadir información contacto usuario en base datos

Lámina 29 Roberto Gómez Cárdenas

Asignando un password inicial

- Usuario root puede modificar cualquier passwrod con el comando password

```
# passwd toto
Enter existing password          /* el de toto o el de root */
Enter new password
Enter new password again         /* validando */
#
```

- mkpasswd
  - comando parte de la paquete expect de Don Libes
  - generación de password para nuevos usuarios
  - no confundir utilería mkpasswd de expect, con comando mkpasswd
    - este último solo codifica un determinado string como un password

Lámina 30 Roberto Gómez Cárdenas

Creando directorio usuario toto

```
# cd /usr/home
# mkdir toto
# chown toto toto
# chgrp users toto
# chmod 700 toto
# cd toto
```


Ejemplo usuario toto

```
# cp /usr/local/lib/skel/.[a-zA-Z]* ~toto
# chmod 644 ~toto/.[a-zA-Z]*
# chown toto ~toto/.[a-zA-Z]*
# chgrp users ~toto/.[a-zA-Z]*
```

Nota: no se puede usar:

```
# chown toto ~toto/.*
```

ya que no solo sería dueño de sus archivos sino de su directorio padre

Verificando

- Para verificar que todo esta bien: salirse de root y entrar como el nuevo usuario y ejecutar los siguientes comandos:

$**pwd**	-- *verificar directorio home*
$ **ls -lag**	-- *verificar GID y UID de los archivos*
	-- *del nuevo usuario*
$ touch toto	-- *creando un archivo para verificar*
$ ls -l	-- *los permisos de los archivos nuevos*
$ \rm toto	-- *dejando todo como estaba*
$ ls -l ..	-- *verificando permisos directorio usuario*

Lámina 33 Roberto Gómez Cárdenas

Baja usuarios

- Dejar quota usuario en cero, si se están usando.
- Remover usuario de cualquier base de datos local o lista de telefonos, correos
- Matar procesos usuario que se encuentren corriendo
- Borrar trabajos definidos en el archivo crontab y sistema at
- Borrar archivos temporales usuario en archivo /var/tmp o /tmp
- Borrar usuario de archivos passwd, shadow y groups
- Borrar directorio hogar usuario
- Borrar spool de correo

Lámina 34 Roberto Gómez Cárdenas

Utilerías manejo de usuarios

Utilería	Uso
useradd	Añadir un usuario
userdel	Eliminar un usuario
usermod	Modificar los atributos de un usuario
groupadd	Añadir un grupo
groupdel	Eliminar un grupo
groupmod	Modificar los atributos de un grupo
passwd	Cambiar la contraseña de un usuario

Lámina 35 Roberto Gómez Cárdenas

Ejemplo opciones: comando *useradd*

```
useradd  [-u uid [-o]] [-g group] [-G group,...]
         [-d home] [-s shell] [-c comment] [-m [-k template]]
         [-f inactive] [-e expire mm/dd/yy] [-p passwd] [-n] [-r] name
useradd  -D [-g group] [-b base] [-s shell] [-f inactive]
         [-e expire mm/dd/yy]

Crea un usuario. Opciones:

        -c  Comentario sobre el usuario
        -d  Directorio de conexión del usario
        -e  Fecha en la cual la cuenta de usuario se desactiva
        -f  Días que transcurrirán desde la caducidad de la
            contraseña hasta la desactivación de la cuenta
        -g  Nombre o número del grupo primario
        -G  Lista de los grupos suplementarios del usuario
            (separados por ,) (sin espacios)
        -m  Crea el directorio de conexión. Si no se especifica la opción
            -k, copia al directorio de conexión los ficheros del directorio
            /etc/skel
        -k  Copia al directorio de conexión los ficheros del directorio
            template
        -p  Asigna una contraseña cifrada al usuario
        -r  Crea una cuenta del sistema
        -s  Asigna el shell a utilizar por el usuario
        -u  Asigna un UID concreto al usuario
        -o  Permite crear un UID duplicado con la opción -u
        -D  Se asignan valores por defecto para las opciones indicadas.
            (No crea usuario)
```

Lámina 36 Roberto Gómez Cárdenas

Manejo usuarios RedHat

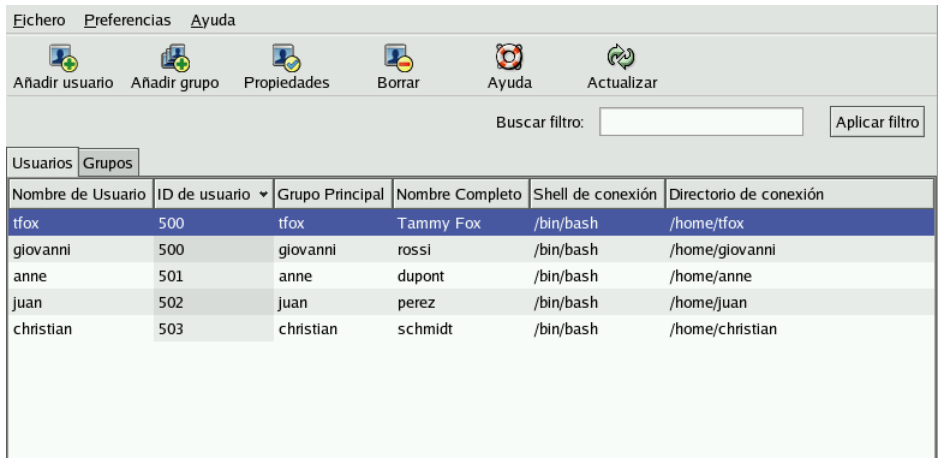
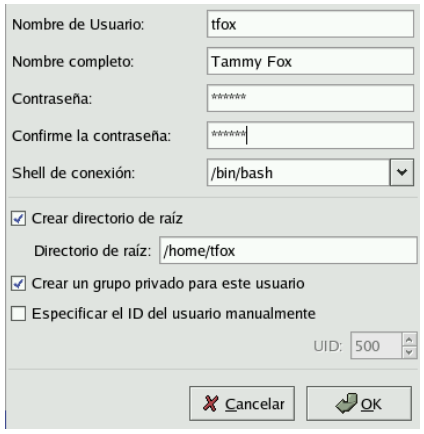


Lámina 37 Roberto Gómez Cárdenas

Alta y actualización usuarios



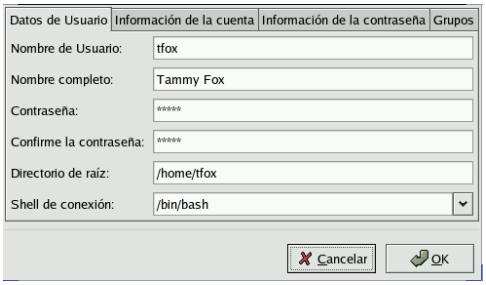


Lámina 38 Roberto Gómez Cárdenas

Manejo de grupos

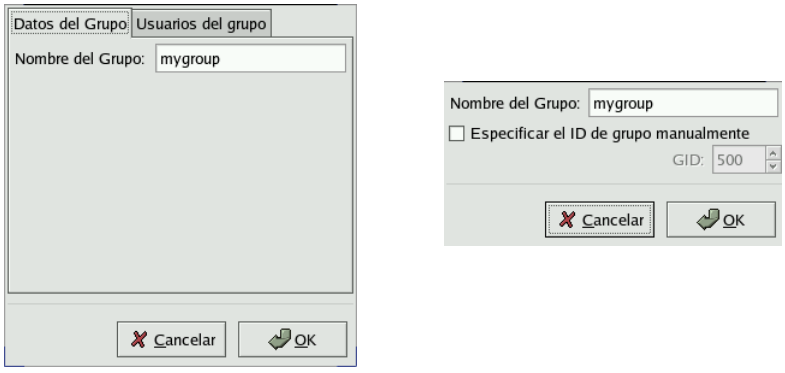


Lámina 39
Roberto Gómez Cárdenas

Archivo /etc/login.defs

- Este archivo se utiliza por algunos sistemas Linux.
- Permite definir algunos valores por defecto para diferentes programas como useradd y expiración de contraseñas.
- Usar este archivo para proporcionar control centralizado sobre ambientes de usuario.
- Asegurarse que el propietario es root.
- Asegurarse que los permisos están puestos a 600.

Lámina 40
Roberto Gómez Cárdenas

Ejemplo archivo

```
# *REQUIRED*
# Directory where mailboxes reside, _or_ name of file, relative to the
# home directory. If you _do_ define both, MAIL_DIR takes precedence.
MAIL_DIR /var/spool/mail
#MAIL_FILE .mail
# Password aging controls:
#
PASS_MAX_DAYS 99999
PASS_MIN_DAYS 0
PASS_MIN_LEN 5
PASS_WARN_AGE 7
#
# Min/max values for automatic uid
# selection in useradd
#
UID_MIN 500
UID_MAX 60000
#
# Min/max values for automatic gid
# selection in groupadd
#
GID_MIN 500
GID_MAX 60000
```

```
# If defined, this command is run when removing a user.
# It should remove any at/cron/print jobs etc. owned by
# the user to be removed (passed as the first argument).
#
#USERDEL_CMD /usr/sbin/userdel_local
#
# If useradd should create home directories for users by default
# On RH systems, we do. This option is ORed with the -m flag on
# useradd command line.
#
CREATE_HOME yes
```

Lámina 41

Administración Linux

Administración Sistemas Archivos

Roberto Gómez Cárdenas
rogomez@itesm.mx
http://webdia.cem.itesm.mx/ac/rogomez

Lámina 42 Roberto Gómez Cárdenas

Permisos por default y umask

- Abreviación de user file creation mode mask
- El número octal de cuatro digitos que Unix usa para determinar los permisos de los nuevos archivos creados.
- Especifica los permisos que *no se quiere* que tengan los nuevos archivos y directorios.
- Comando trabaja haciendo un AND de bits con el complemento de umask
- Por default los archivos se crean con permiso 666 y los directorios con 777

Lámina 43 Roberto Gómez Cárdenas

El comando umask

- Es un comando interno (built-in) de sh, ksh y csh
- Los valores más comunes son 022, 027 y 077
- Un valor de 022 permite que lea y escriba todos los archivos recien creados, y el resto solo puede leerlos.

```
0666  (mode de creación por default)
0022  (valor de umask)
0644  (modo resultante)
```

- Con 077 solo el propietario puede leer y escribir los archivos creados

```
0666  (mode de creación por default)
0077  (valor de umask)
0600  (modo resultante)
```

Lámina 44 Roberto Gómez Cárdenas

El comando umask (cont)

- Una forma simple de calcular valores de umask es recordar que:
  - valor 2: apaga el permiso de escritura
  - valor 7: apaga permisos lectura, escritura y ejecución
- Si se esta usando ksh, se puede asignar el valor de umask de forma simbólica, tal y como se hace con el comando chmod

```
emata@francia:7> umask u=rwx, g=x
emata@francia:8> umask 067
```

Lámina 45 Roberto Gómez Cárdenas

El sticky bit

- Aplicable en archivos ejecutables
- Le indica a Unix que deje el ejecutable en memoria después de que esta haya terminado su ejecución
- Dejando el programa en memoria, reduce el tiempo para otros usuarios (en teoría)
- Fue una interesante idea hace tiempo, pero es obsoleta hoy en día
  - técnicas memoria virtual la hacen innecesaria
  - paginación hace que ya no se use

Lámina 46 Roberto Gómez Cárdenas

Sticky bit y los directorios

- Si un usuario tiene permiso escritura en un directorio puede renombrar o borrar archivos en él (aunque no le pertenezcan)
- Varias nuevas versiones de Unix tiene una forma de impedir lo anterior
- El propietario del directorio puede activar el sticky bit
- Los usuarios que pueden renombrar o borrar archivos en dicho subdirectorio son:
  - el propietario del archivo
  - el propietario del directorio
  - el superusuario

Lámina 47 Roberto Gómez Cárdenas

Ejemplo uso sticky bit en directorios

```
egarcia> mkdir proyecto
egarcia> chmod 777 proyecto
egarcia> ls -ld
drwxrwxrwx      2   egarcia  profes        32 Sep 23  19:30 proyecto

/* usuario jvazquez borra un archivo que no le pertenece */

jvazquez> cd /home/usr/egarcia/proyecto
  ls -lg
total 3
-rw-r--r--   1  rogomez    profes    120 Sep 23 19:23 data.rogomez
-rw-r--r--   1  jvazquez   profes 3421 Sep 24 20:03 data.jvazquez
-rw-r--r--   1  egarcia     profes   728 Sep 25 01:34 data.egarcia
-rw-r--r--   1  aortiz      profes   716 Sep 27 12:52 data.aortiz
jvazquez> rm data.aortiz
```

Lámina 48 Roberto Gómez Cárdenas

```
jvazquez> ls -lg
total 2
-rw-r--r--   1  rogomez    profes   120 Sep 23 19:23 data.rogomez
-rw-r--r--   1  jvazquez   profes      3421 Sep 24 20:03 data.jvazquez
-rw-r--r--   1  egarcia     profes   728 Sep 25 01:34 data.egarcia

egarcia> chmod 1777 proyecto
egarcia> ls -ld
drwxrwxrwxt     2   egarcia  profes        32 Sep 23  19:30 proyecto

jvazquez> rm data.rogomez
data.rogomez: 644 mode ? y
rm: data.rogomez not removed
Permission denied
jvazquez>
```

Lámina 49 Roberto Gómez Cárdenas

Los usuarios y los procesos

- Procesos pertenecen a un solo y único usuario
- El propietario es el que lanzó el proceso
  - puede enviarle señales y, en consecuencia, matarlo
- Para lanzarlo debe poseer los permisos de ejecución del archivo que contiene el código binario

Lámina 50 Roberto Gómez Cárdenas

- La “propiedad” del archivo del código no influye en la del proceso
  - usuario toto ejecuta código de un archivo que pertenece a cachafas
  - el proceso pertenece a usuario toto
- Esto es limitativo
  - se desea permitir a un usuario modificar el contenido de un archivo sin darle derecho de escritura en él
  - ejemplo archivo /etc/passwd, un usuario debe poder cambiar su password sin poder modificar el archivo que lo contiene

Lámina 51 Roberto Gómez Cárdenas

El bit Set UID (SUID)

- Derecho complementario de un proceso que condiciona la propiedad del proceso que ejecuta su código
- Retomando el ejemplo anterior:
  - si usuario cachafas activa el bit SUID del archivo
  - el usuario toto es el propietario del archivo, pero el propietario efectivo es cachafas
  - toto adquiere los derechos de cachafas durante el tiempo que dure la ejecución del proceso

Lámina 52 Roberto Gómez Cárdenas

Cuidados del bit SUID

- El bit SUID puede representar un hoyo en la seguridad del sistema
- Es necesario minimizar el número de archivos que pertenezcan al super-usuario y que tengan activado el bit SUID
- Algunas versiones de Unix ignoran el bit SUID y SGID en scripts, solo programas compilados pueden tenerlo activo

Lámina 53 Roberto Gómez Cárdenas

El bit Set Group ID (SGID)

- Mismo principio que SUID pero para grupos
- Ejecutar un archivo con bit SGID activo asigna el ID de grupo del usuario al mismo que el del archivo ejecutado, durante el tiempo que dura la ejecución de este
- Archivos con SGID o SUID activo pierden sus propiedades especiales cuando son copiados

Lámina 54 Roberto Gómez Cárdenas

Ejemplo bits SUID y SGID

```
rogomez@armagnac:3> ls -l /usr/bin/passwd  /usr/bin/login
                                /usr/bin/mailx /etc/passwd
-rw-r--r--   1 root         752 Oct 22  1998 /etc/passwd
-r-sr-xr-x   1 root       29192 Jul 15  1997 /usr/bin/login*
-r-x--s--x   1 bin       127540 Jul 15  1997 /usr/bin/mailx*
-r-sr-sr-x   3 root        96796 Jul 15  1997 /usr/bin/passwd*
rogomez@armagnac:4>
```

Lámina 55 Roberto Gómez Cárdenas

Comando chmod: SGID, SUID, sticky bit

chmod n777 a1

Valor n	Efecto	Ejemplo	Resultado ls -l a1
1	Activar sticky bit	chmod 1777 a1	-rwxrwxrwt
2	Activar SGID	chmod 2777 a1	-rwxrwsrwx
4	Activar SUID	chmod 4777 a1	-rwsrwxrwx
6	Activar SUID y SGID	chmod 6777 a1	-rwsrwxswx
0	Desactivar sticky bit, SUID y SGID	chmod 0777 a1	-rwxrwxrwx

Lámina 56 Roberto Gómez Cárdenas

Las listas de control de acceso (acl)

- Algunos sistemas que cumplen con el libro Naranja, han cambiado el sistema de protección de permisos al de listas de control de acceso.
- Se basa en el concepto de derechos sobreentendidos
- Se trata de afinar la noción de permiso a usuarios o grupos específicos
- Se puede dotar de permisos de rwx a un determinado usuario o grupo de usuarios
- Existen en Unix desde hace más de diez años
- No todas las versiones de Unix lo soportan

Lámina 57 Roberto Gómez Cárdenas

Comandos

- Dos comandos
  - getfacl archivo
    - despliega el ACL del archivo
  - setfacl archivo
    - opción -m: modificación del ACL

      user:<user name>:rwx
      group:<group name>:rwx
      other:rwx

    - opción -x: borrar entradas ACL

      setfacl -x g:staff file
- Necesario montaje apropiado

  mount -o remount,acl /

Lámina 58 Roberto Gómez Cárdenas

Ejemplo uso ACLs

Desplegando permisos de forma normal:

```
root@cachafas:2# ls -l /usr/local/sshd
  - -rwx------   1 root     bin      2616160 Apr 28  1997 /usr/local/sshd
root@cachafas:3#
```

Verificando permisos con comando getfacl:

```
root@cachafas:3# getfacl /usr/local/sshd
# file: /usr/local/sshd
# owner: root
# group: bin
user::rwx
group::---
mask:---
other:---
root@cachafas:4#
```

Lámina 59
Roberto Gómez Cárdenas

Extendiendo los permisos

```
root@cachafas:4# setfacl -m user:toni:r-x /usr/local/sshd
root@cachafas:5# getfacl /usr/local/sshd
# file: /usr/local/sshd
# owner: root
# group: bin
user::rwx
user:toni:r-x
group::---
mask:---
other:---
root@cachafas:6#
```

Lámina 60
Roberto Gómez Cárdenas