Manejo de Grupo, Usuarios y Permisos

Grupos:

Fichero: /etc/group

1. Nombre de grupo
2. Contraseña
   HASH tipo MD5
3. GID.
4. Nombre de los usuarios miembros del grupo (separados por comas).

Comandos para manejar grupos:

groupadd
: Permite dar de alta un grupo.

groupmod
: Permite modificar parámetros de un grupo.

groupdel
: Borra un grupo.

Fichero: /etc/gshadow
: Contiene parámetros avanzados de los grupos

Ejemplos:

```
groupadd COMPRAS
groupadd VENTAS
groupadd CLIENTES

groupmod -g 2000 VENTAS

groupdel CLIENTES
```

Usuarios:

Fichero: /etc/passwd

1. Nombre de usuario
2. Contraseña
   - HASH tipo MD5
   - x la clave esta en el /etc/shadow
3. UID
4. GID
5. Información real del usuario (separados por comas)
   1. Nombre real del usuario
   2. Número de despacho del usuario
   3. Número de teléfono de la oficina
   4. Número de teléfono particular
6. Directorio de entrada
7. Shell (Interprete de comandos)

Fichero: /etc/shadow

1. Nombre del usuario
2. Contraseña
   - HASH tipo MD5
   - * Indica que el usuario no se puede logear
3. Fecha del último cambio de contraseña
4. Número mínimo de días que debe permanecer la contraseña sin cambiarse
5. Número máximo de días que puede permanecer la contraseña sin cambiarse. A partir de este dato se obtiene la fecha de expiración de la contraseña
6. Fecha del primer aviso al usuario de que la contraseña está a punto de expirar, expresada en días previos a la fecha de expiración. A partir del primer aviso, el usuario no recibirá una advertencia cada vez que entre al sistema.
7. Días que pasaran entre la expiración de la contraseña y la in habilitaciones (bloqueo) de la cuenta. Un valor (-1) significa que no existirá relación entre ambas fechas.
8. Fecha de caducidad de la cuenta. Se expresa en días transcurridos desde el 1 de enero de 1970 hasta la fecha en que se bloqueara la cuenta.
9. Reservado.

Comandos para manejar usuarios:

useradd
: Permite dar de alta un usuario.

userdel
: Borra a un usuario.

usermod
: Permite modificar algún parámetro de un usuario.

passwd
: Permite crear y actualizar la clave de un usuario.

Ejemplos:

useradd pedro
useradd maria
useradd -g mail juan

usermod -c “Pedro MARTINEZ” pedro

userdel maria
userdel -rf juan

Comandos de administración y control de usuarios	
adduser	Ver useradd
chage	Permite cambiar o establecer parámetros de las fechas de control de la contraseña.
chpasswd	Actualiza o establece contraseñas en modo batch, múltiples usuarios a la vez. (se usa junto con newusers)
id	Muestra la identidad del usuario (UID) y los grupos a los que pertence.
gpasswd	Administra las contraseñas de grupos (/etc/group y /etc/gshadow).
groupadd	Añade grupos al sistema (/etc/group).
groupdel	Elimina grupos del sistema.
groupmod	Modifica grupos del sistema.
groups	Muestra los grupos a los que pertence el usuario.
newusers	Actualiza o crea usuarios en modo batch, múltiples usuarios a la vez. (se usa junto chpasswd)
pwconv	Establece la protección shadow (/etc/shadow) al archivo /etc/passwd.
pwunconv	Elimina la protección shadow (/etc/shadow) al archivo /etc/passwd.
useradd	Añade usuarios al sistema (/etc/passwd).
userdel	Elimina usuarios del sistema.
usermod	Modifica usuarios.

Archivos de administración y control de usuarios	
.bash_logout	Se ejecuta cuando el usuario abandona la sesión.
.bash_profile	Se ejecuta cuando el usuario inicia la sesión.
.bashrc	Se ejecuta cuando el usuario inicia la sesión.
/etc/login.defs	Contraseñas encriptadas de los grupos.
	Variables que controlan los aspectos de la creación de usuarios. • Número máximo de días que una contraseña es válida PASS_MAX_DAYS • El número mínimo de caracteres en la contraseña PASS_MIN_LEN • Valor mínimo para usuarios normales cuando se usa useradd UID_MIN • El valor umask por defecto UMASK • Si el comando useradd debe crear el directorio home por defecto CREATE_HOME

Permisos:

Los permisos o atributos de los archivos Linux son de tres tipos y se aplican en tres niveles de prioridad. Estos permisos se pueden ver con la opción (ls -l). Como podemos ver, vienen agrupados en tres partes, -rrwxrwxrwx-, una por cada tipo de prioridad. Los tres primeros permisos (los que vemos más a la izquierda), son los del propietario, los tres siguientes son los permisos aplicados a los miembros del grupo al que pertenece el propietario del archivo y los tres últimos son los aplicados al resto de usuarios del sistema; El ultimo es un permiso especial denominado s, que se utiliza para que el fichero tome los derechos de la persona que lo ejecuta.

El primer carácter que se muestra de los atributos nos informa el tipo de archivo y es codificado por 4 bits internamente:

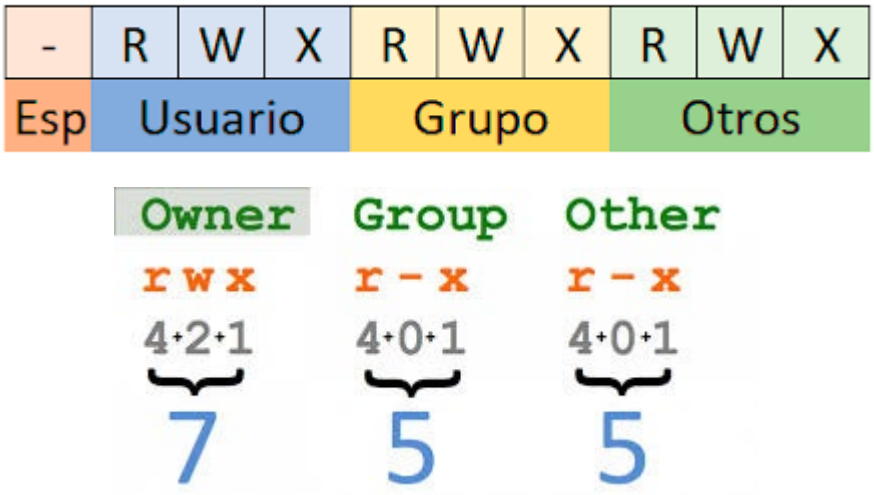
Solo el propietario del archivo puede cambiar los permisos del mismo y esto lo puede hacer mediante algunos de los siguientes comandos:

chown

Permite cambiar el dueño y grupo de un archivo o carpeta
chown juan:compras archivo

chmod

Permite cambiar los permisos de un archivo o carpeta



Ej:
chmod 775 archivo
chmod u+x,. g+r archivo

chgrp

Permite cambiar el grupo de un archivo o carpeta
Ej:
chgrp ventas archivo

Permisos Avanzados

SUID, SGID y StickyBit

Estos permisos se configuran mediante el comando chmod

SUID: Permiso de super usuario, si se tiene este permiso cuando se ejecuta una aplicación, dicha aplicación se ejecuta con los permisos del dueño del archivo no con el dueño que lo ejecuta.

SGID: Permiso de super grupo, si se tiene este permiso cuando se ejecuta una aplicación, dicha aplicación se ejecuta con los permisos del grupo del archivo no con el grupo que lo ejecuta.

StitckBit: Este permiso tiene efecto sobre las carpetas e indica que solo los archivos pueden ser eliminados por su usuario dueño.

Permisos Extendidos

POSIX (Access Control List)

Permite gestionar más de una entrada de permisos como permisos por defecto, se debe habilitar dichos permisos en la partición deseada.

Comandos:

getfacl, permite ver las entradas de permisos.
setfacl, permite configurar las entradas de permisos.

SELinux

Sistema más avanzado de seguridad, agrega auditoria y el control mediante etiquetas sobre archivos, carpetas, socket, puertos, etc…

Comandos varios:

w
: Muestra los usuarios conectados y que es los que están haciendo.

who
: Muestra la lista de los usuarios que están conectados al equipo.

last

lastb
: Muestra el listado de los últimos usuarios conectados.

id
: Muestra información del usuario actual.

¿Que es sudo?

El programa sudo (del inglés **s***uper* **u***ser* **do**) es una utilidad de los sistemas operativos tipo Unix, como Linux, BSD, o Mac OS X y permite a los usuarios ejecutar programas con los privilegios de seguridad de otro usuario (normalmente el usuario root) de manera segura. El usuario de este modo se convierte así temporalmente en superusuario.

Uso y configuración

El archivo principal de configuración de sudo es /etc/sudoers , y no se recomienda editarlo manualmente. Para modificar el comportamiento y configuración del programa se recomienda utilizar la utilidad visudo.

Ejemplo: /etc/sudoers

```
Defaults env_reset
Defaults mail_badpass
Defaults secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"

root ALL=(ALL:ALL) ALL
%sudo ALL=(ALL:ALL) ALL
```

Las últimas dos líneas configuran que todos los comandos pueden solo ser ejecutados por el usuario root (root…) y por los que pertenecen al grupo sudo (%sudo…).

Ejemplos de configuración

A través de diferentes casos de uso se mostraran diferentes ejemplos de cómo transferir privilegios de root a diferentes usuarios para que puedan o no ejecutar ciertos comandos con privilegios de «superuser» y de este modo no compartir la contraseña de root con muchos administradores.

Caso de uso 1

El siguiente ejemplo transfiere al usuario admin1 los privilegios del usuario root . Con esto admin1 ejecutara cualquiera de los comandos con privilegios de superusuario.

> Se recomienda el uso de visudo para realizar las modificaciones de funcionamiento y configuración del comportamiento de sudo.

Se ejecuta visudo del siguiente modo:

```
visudo
```

Según sea el editor por defecto del sistema, abre el archivo /etc/sudoers para su edición. Agregue al final del archivo la linea siguiente:

```
admin1 ALL=(ALL:ALL) ALL
```

Se guardan los cambios, y se cierra el archivo.
Los cambios son tomados instantáneamente. Hemos agregado que el usuario admin1 puede ejecutar cualquier comando con privilegios de root mediante el uso del comando.

La siguiente captura muestra la secuencia de comandos ejecutados por admin1 para mostrar el contenido del archivo /etc/sudoers y que es propiedad del usuario root . Observar que solo funciona cuando se ejecuta mediante sudo

Caso de uso 2

Transferir y permitir privilegios de root a un usuario admin2, pero deshabilitar algunos comandos, en particular no debe poder apagar, ni reiniciar el sistema mediante comandos.

La configuración se realiza a continuación, considerando que el usuario admin2 ya está creado en el sistema. Salvo indicación expresa los comandos son realizados por el usuario root y son las siguientes.

Con el comando visudo se agrega el siguiente contenido a la configuración de sudoers:

```
# Especificación de Cmnd_alias
Cmnd_Alias SHUTDOWN = /usr/sbin/halt, /usr/sbin/shutdown, \
 /usr/sbin/poweroff, /usr/sbin/reboot, /usr/sbin/init, /usr/bin/systemctl

# admin2 no puede ejecutar los comandos especificados en Cmnd_Alias
admin2 ALL=(ALL:ALL) ALL, !SHUTDOWN
```

sudoers

Archivo de configuración de `sudo`, generalmente ubicado bajo /etc y se modifica a través del uso de `visudo`. En este archivo se establece quien (usuarios) puede ejecutar que (comandos) y de que modo (opciones), generando efectivamente una lista de control de acceso que puede ser tan detallada como se desee.

Es más fácil entender `sudo` si dividimos en tres partes su posible configuración, estás son:

- Alias
- Opciones (Defaults)
- Reglas de acceso

Por extraño que parezca ninguna de las secciones es obligatoria, o tienen que estar en algún orden específico, pero la que al menos debe de existir es la tercera, que es la definición de los controles o reglas de acceso. Se detallará cada uno de estos en un momento. Para los que les gusta saber más la cuestión técnica es interesante saber que la construcción de un archivo *sudoers* está basado en la forma BNF (Backus-Naur Form), concretamente en versión extendida (EBNF), si estudiaste algún curso de informática universitario seguramente sabes de lo que hablo. EBNF describe de una forma precisa y exacta la gramática de un lenguaje, esta se va creando a través de reglas de producción que a la vez son la base para ser referenciadas por otras reglas. Afortunadamente no necesitas saber nada de esto, solo entender cómo se aplican estas reglas.

Alias

Un alias se refiere a un usuario, un comando o a un equipo. El alias engloba bajo un solo nombre (nombre del alias) una serie de elementos que después en la parte de definición de reglas serán referidos aplicados bajos cierto criterio. Es decir, regresando a EBNF estamos creando las reglas de

producción inicial. La forma para crear un alias es la siguiente:

tipo_alias NOMBRE_DEL_ALIAS = elemento1, elemento2, elemento3, ... elementoN

tipo_alias NOMBRE1 = elemento1, elemento2 : NOMBRE2 = elemento1, elemento2

En el segundo caso, separado por ":" es posible indicar más de un alias en una misma defínción.

El tipo_alias define los elementos, es decir, dependiendo del tipo de alias serán sus elementos. Los tipos de alias son cuatro y son los siguientes:

- Cmnd_Alias - define alias de comandos.
- User_Alias - define alias de usuarios normales.
- Runas_Alias - define alias de usuarios administradores o con privilegios.
- Host_Alias - define alias de hosts o equipos.

El NOMBRE_DEL_ALIAS puede llevar letras, números o guión bajo (_) y DEBE de comenzar con una letra mayúscula, se acostumbra a usarlos siempre en mayúsculas.

Los elementos del alias varían dependiendo del tipo de alias, asi que veámoslos por partes asi como varios ejemplos para que comience a quedar claro todo esto.

Cmnd_Alias

Definen uno o más comandos y otros alias de comandos que podrán ser utilizados después en alias de usuarios. Ejemplos:

Cmnd_Alias WEB = /usr/sbin/apachectl, /usr/sbin/httpd, sudoedit /etc/httpd/

Indica que a quien se le aplique el alias WEB podrá ejecutar los comandos apachectl, httpd y editar todo lo que este debajo del directorio /etc/httpd/, nótese que debe de terminar con '/' cuando se indican directorios. También, la ruta completa a los comandos debe ser indicada.

Cmnd_Alias APAGAR = /usr/bin/shutdown -h 23\:00

Al usuario que se le asigne el alias APAGAR podrá hacer uso del comando 'shutdown' exactamente con los parámetros como están indicados, es decir apagar *-h* (halt) el equipo a las 23:00 horas. Nótese que es necesario escapar el signo ':', asi como los símbolos ' : , = \

Cmnd_Alias NET_ADMIN = /sbin/ifconfig, /sbin/iptables, WEB

NET_ADMIN es un alias con los comandos de configuración de interfaces de red ifconfig y de firewall iptables, pero además le agregamos un alias *previamente* definido que es WEB, así que a quien se le asigne este alias podrá hacer uso de los comandos del alias WEB.

Cmnd_Alias TODO_BIN = /usr/bin/, !/usr/bin/rpm

A quien se le asigne este alias podrá ejecutar todos los comandos que estén dentro del directorio /usr/bin/ menos el comando 'rpm' ubicado en el mismo directorio. *NOTA IMPORTANTE: este tipo de alias con un permiso muy amplios menos '!' algo, generalmente no son una buena idea, ya que comandos nuevos que se añadan después a ese directorio también podrán ser ejecutados, es mejor siempre definir específicamente lo que se requiera.*

User_Alias

Definen a uno o más usuarios, grupos del sistema (indicados con %), grupos de red (netgroups indicados con +) u otros alias de usuarios. Ejemplos:

User_Alias MYSQL_USERS = andy, marce, juan, %mysql

Indica que al alias MYSQL_USERS pertenecen los usuarios indicados individualmente más los usuarios que formen parte del grupo 'mysql'.

User_Alias ADMIN = sergio, ana

'sergio' y 'ana' pertenecen al alias ADMIN.

User_Alias TODOS = ALL, !samuel, !david

Aquí encontramos algo nuevo, definimos el alias de usuario TODOS que al poner como elemento la palabra reservada 'ALL' abarcaría a todos los usuarios del sistema, pero no deseamos a dos de ellos, así que negamos con '!', que serían los usuarios 'samuel' y 'david'. Es decir, todos los usuarios menos esos dos. *NOTA IMPORTANTE: este tipo de alias con un permiso muy amplios menos '!' algo, generalmente no son una buena idea, ya que usuarios nuevos que se añadan después al sistema también serán considerados como ALL, es mejor siempre definir específicamente a los usuarios que se requieran. ALL es válido en todos los tipos de alias.*

User_Alias OPERADORES = ADMIN, alejandra

Los del alias ADMIN más el usuario 'alejandra'.

Runas_Alias

Funciona exactamente igual que User_Alias, la única diferencia es que es posible usar el ID del usario UID con el caracter '#'.

Runas_Alias OPERADORES = #501, fabian

Al alias OPERADORES pertenecen el usuario con UID 501 y el usuario 'fabian'

Host_Alias

Definen uno o más equipos u otros alias de host. Los equipos pueden indicarse por su nombre (si se encuentra en /etc/hosts) por nombre de dominio, si existe un resolvedor de dominios, por dirección IP, por dirección IP con máscara de red. Ejemplos:

Host_Alias LANS = 192.168.0.0/24, 192.168.0.1/255.255.255.0

El alias LANS define todos los equipos de las redes locales.

Host_Alias WEBSERVERS = 172.16.0.21, web1 : DBSERVERS = 192.168.100.10, dataserver

Se define dos alias en el mismo renglón: WEBSERVERS y DBSERVERS con sus respectivas listas de elementos, el separador ':' es válido en cualquier definición de tipo de alias.

Las opciones o defaults permiten definir ciertas características de comportamiento para los alias previamente creados, para usuarios, usuarios privilegiados, para equipos o de manera global para todos. No es necesario definir opciones o defaults, `sudo` ya tiene establecidas el valor de cada uno, y es posible conocerlas a través de `sudo -V` (ver en la sección sudo de este tutorial).

Sin embargo, la potencia de `sudo` está en su alta granularidad de configuración, así que es importante conocer como establecer opciones específicas.

Las opciones o defaults es posible establecerlos en cuatro niveles de uso:

- De manera global, afecta a todos
- Por usuario
- Por usuario privilegiado
- Por equipo (host)

Se usa la palabra reservada 'Defaults' para establecer las opciones y dependiendo del nivel que deseamos afectar su sintaxis es la siguiente:

- Global: Defaults opcion1, opcion2 ...
- Usuario: Defaults:usuario opcion1, opcion2 ...
- Usuario Privilegiado: Defaults>usuario opcion1, opcion2 ...
- Equipo: Defaults@equipo opcion1, opcion2 ...

La lista de opciones es algo extensa, pueden consultarse en las páginas del manual (`man sudoers`). En este tutorial de LinuxTotal.com.mx me concretaré a ejemplificar varios ejemplos del uso de establecer opciones.

Los defaults los divide el manual (man sudoers) en cuatro: flags o booleanos, enteros, cadenas y listas. Veamos entonces algunos ejemplos de uso para cada uno de ellos:

flags o booleanos

Generalemente se usan de manera global, simplemente se indica la opción y se establece a 'on' para desactivarla 'off' se antepone el símbolo '!' a la opción. Es necesario consultar el manual para saber el

valor por defecto 'on' o 'off' para saber si realmente necesitamos invocarla o no.

Defaults mail_always

Establece a 'on' la opción 'mail_always' que enviara un correo avisando cada vez que un usuario utiliza `sudo`, a la vez, este opción requiere que 'mailto_user' este establecida.

Defaults !authenticate, log_host

Desactiva 'off' el default 'authenticate' que por defecto esta activado 'on' e indica que todos los usuarios que usen `sudo` deben identificarse con su contraseña, obviamente esto es un ejemplo y sería una pésima idea usarlo realmente, ya que ningún usuario necesitaria autenticarse, esto es porque estamos usando Defaults de manera global. La segunda opción 'log_host' que por defecto está en 'off' la activamos y bitacoriza el nombre del host cuando se usa un archivo (en vez de syslog) como bitácora de `sudo`.

Defaults:ana !authenticate

Aqui se aprecia algo más lógico, usamos opciones por usuario en vez de global, indicando que el usuario 'ana' no requerirá autenticarse. Pero todos los demás si.

Defaults>ADMIN rootpw

Opciones para usuarios privilegiados, en vez de usar una lista de usuarios, usamos un alias 'ADMIN' que se supone fue previamente definido, y establecemos en 'on' la opción 'rootpw' que indica a `sudo` que los usuarios en el alias 'ADMIN' deberán usar la contraseña de 'root' en vez de la propia.

Enteros

Tal como su nombre lo indica, manejan valores de números enteros en sus opciones, que deben entonces usarse como *opción = valor*.

Defaults:fernanda, regina passwd_tries = 1, passwd_timeout = 1

Ejemplo donde se aprecia el uso de opciones con valores enteros. En este caso se establecen opciones para los usuarios 'fernanda' y 'regina' solamente, que solo tendrán una oportunidad de ingresar la contraseña correcta 'passwd_tries' el valor por defecto es de 3 y tendrán un minuto para ingresarla 'passwd_timeout' el valor por defecto son 5 minutos.

La mayoría de las opciones de tiempo o de intentos, al establecerlas con un valor igual a cero entonces queda ilimitado la opción.

Defaults@webserver umask = 011

Se establecen opciones solo para los usuarios que se conectan al servidor 'webserver' y el valor 'umask' indica que si mediante la ejecución del comando que se invoque por `sudo` es necesario crear archivos

o diectorios, a estos se les aplicará la máscara de permisos indicada en el valor de la opción.

Cadenas

Son valores de opciones que indican mensajes, rutas de archivos, etc. Si hubiera espacios en el valor es necesario encerrar el valor entre comillas dobles (" ").

Defaults badpass_message = "Intenta de nuevo: "

Para todos los usuarios, cuando se equivoquen al ingresar la contraseña, es el mensaje que saldría. En este caso la opción por defecto es "Sorry: try again".

Listas

Permite establecer/eliminar variables de entorno propias de `sudo`. Los 'Defaults' para variables es de los menos usados en las configuraciones de `sudo` y ciertamente de los más confusos. Para entender como se aplican es más fácil si primero ejecutas como 'root' el comando `sudo -V`, y al final del listado encontrarás en mayúsculas las posibles variables de entorno que se pueden establecer o quitar y que vienen del shell.

Solo existen tres opciones de listas: *env_check, env_delete y env_keep*, las listas pueden ser remplazadas con '=', añadidas con '+=', eliminadas con '-=' o deshabilitadas con '!'. Con un par de ejemplos quedará más claro.

Defaults env_delete -= HOSTNAME

Elimina la variable de entorno 'HOSTNAME', (pero preserva todas las demás que hubiera) y comandos que se ejecuten bajo `sudo` y que requieran de esta variable no la tendrían disponible.

Defaults env_reset

Defaults env_check += DISPLAY, PS1

La primera opción 'env_reset' reinicializa las variables de entorno que `sudo` utilizará o tendrá disponibles, y solo quedan disponibles LOGNAME, SHELL, USER y USERNAME. La siguiente línea indica que agregue (+=) a lo anterior, también la variable de entorno DISPLAY a su valor establecido antes del reset.

Aunque no es obligatorio declarar alias, ni opciones (defaults), y de hecho tampoco reglas de acceso, pues el archivo /etc/sudoers no tendría ninguna razón de ser si no se crean reglas de acceso. De hecho podríamos concretarnos a crear solamente reglas de acceso, sin opciones ni alias y podría funcionar todo muy bien.

Las reglas de acceso definen que usuarios ejecutan que comandos bajo que usuario y en que equipos. La mejor y (según yo, única manera) de entender y aprender a configurar sudoers es con ejemplos, asi que directo al grano:

usuario host = comando1, comando2, ... comandoN

Sintaxis básica, 'usuario' puede ser un usuario, un alias de usuario o un grupo (indicado por %), 'host' puede ser ALL cualquier equipo, un solo equipo, un alias de equipo, una dirección IP o una definición de red IP/máscara, 'comandox' es cualquier comando indicado con su ruta completa. Si se termina en '/' como en /etc/http/ entonces indica todos los archivos dentro de ese directorio.

daniela ALL = /sbin/iptables

Usuario 'daniela' en cualquier host o equipo puede utilizar iptables.

ADMIN ALL = ALL

Los usuarios definidos en el alias 'ADMIN' desde cualquier host pueden ejecutar cualquier comando.

%gerentes dbserver = (director) /usr/facturacion, (root) /var/log/*

Un ejemplo más detallado. Los usuarios que pertenezcan al grupo del sistema llamado 'gerentes' pueden en el equipo llamado 'dbserver' ejecutar como si fueran el usuario 'director' la aplicación llamada 'facturacion', además como usuarios 'root' pueden ver el contenido de los archivos que contenga el directorio /var/log.

Lo anterior introduce algo nuevo, que en la lista de comandos es posible indicar bajo que usuario se debe ejecutar el permiso. Por defecto es el usuario 'root', pero no siempre tener que así. Además la lista 'hereda' la primera definición de usuario que se indica entre paréntesis ( ), por eso si se tiene más de alguno hay que cambiar de usuario en el comando conveniente, el ejemplo anterior también sería válido de la siguiente manera:

%gerentes dbserver = /var/log/*, (director) /usr/facturacion

No es necesario indicar (root) ya que es el usuario bajo el cual se ejecutan los comandos por defecto. También es válido usar (ALL) para indicar bajo cualquier usuario. El ejemplo siguiente da permisos absolutos.

sergio ALL = (ALL) ALL

Se establece permiso para el usuario 'sergio' en cualquier host, ejecutar cualquier comando de cualquier usuario, por supuesto incluyendo los de root.

SUPERVISORES PRODUCCION = OPERACION

Una regala formada solo por alias. En el alias de usuario 'SUPERVISORES' los usuarios que esten indicados en ese alias, tendrán permiso en los equipos definidos en el alias de host 'PRODUCCION', de ejecutar los comandos definidos o listados en el alias de comandos 'OPERACION'.

En este último ejemplo se aprecia lo últil que pueden ser los alias, ya que una vez definida la regla, solo

debemos agregar o eliminar elementos de las listas de alias definidos previamente. Es decir, se agrega un equipo más a la red, se añade al alias 'PRODUCCION', un usuario renuncia a la empresa, alteramos el alias 'SUPERVISORES' eliminándolo de la lista, etc.

checo ALL = /usr/bin/passwd *, !/usr/bin/passwd root

Este es un ejemplo muy interesante de la potencia y flexibilidad de `sudo`. Al usuario 'checo', desde cualquier equipo, tiene permiso de cambiar la contraseña de cualquier usuario (usando el comando 'passwd'), excepto '!' la contraseña del usuario 'root'. Lo anterior se logra mediante el uso de argumentos en los comandos. En el primer ejemplo '/usr/bin/passwd *' el asterisco indica una expansión de comodin (wildcard) que indica cualquier argumento, es decir, cualquier usuario. En el segundo caso '!/usr/bin/passwd root', si indica un argumento específico 'root', y la '!' como ya se sabe indica negación, negando entonces el permiso a cambiar la contraseña de root.

Cuando se indica el comando sin argumentos: /sbin/iptables `sudo` lo interpreta como 'puede usar iptables con cualquiera de sus argumentos'.

mariajose ALL = "/sbin/lsmod"

Al estar entre comillas dobles un comando, entonces `sudo` lo interpreta como 'puede hacer uso del comando lsmod pero sin argumentos'. En este caso el usuario 'mariajose' podrá ver la lista de módulos del kernel, pero solo eso.

Tags (etiquetas de comandos)

Cuando se definen reglas, en la lista de comandos, estos pueden tener cero (como en los ejemplos anteriores) o más tags. Existen 6 de estas etiquetas o tags,

NOPASSWD Y PASSWD

Por defecto `sudo` requiere que cualquier usuario se identifique o auténtifique con su contraseña. Aprendimos en la sección de 'Opciones' o 'Defaults' que es posible indicar que un usuario o alias de usuario no requiera de autentificación. Pero el control granular propio de `sudo`, permite ir aun más lejos al indicar a nivel de comandos, cuáles requieren contraseña para su uso y cuáles no.

gerardo webserver = NOPASSWD: /bin/kill, /usr/bin/lprm, /etc/httpd/conf/

Usuario 'gerardo' en el equipo 'webserver' no requerira contraseña para los comandos listados. El tag se hereda, es decir no solo el primer elemento de la lista de comandos, sino los subsiguientes. Suponiendo que el último '/etc/httpd/conf/' elemento, que permite modificar cualquier archivo contenido en el directorio, si deseamos que use contraseña, lo siguiente lo conseguirá:

gerardo webserver = NOPASSWD: /bin/kill, /usr/bin/lprm, PASSWD: /etc/httpd/conf/

Aunque ya que solicitar contraseña es el default o defecto preestablecido, lo anterior también funcionará de la siguiente manera:

gerardo webserver = /etc/httpd/conf/, NOPASSWD: /bin/kill, /usr/bin/lprm,

NOEXEC Y EXEC

Este es un tag muy importante a considerar cuando sobre se otorgan permisos sobre programas que permiten escapes a shell (shell escape), como en el editor 'vi' que mediante el uso de '!' es posible ejecutar un comando en el shell sin salir de 'vi'. Con el tag NOEXEC se logra que esto no suceda, aunque no hay que tomarlo como un hecho, ya que siempre existe la posibilidad de vulnerabilidades no conocidas en los múltiples programas que utilizan escapes a shell. Al igual que los tags anteriores, el tag se hereda y se deshabilita con su tag contrario (EXEC), en caso de que en la lista de comandos hubiera varios comandos.

valeria ALL = NOEXEC: /usr/bin/vi

SETENV Y NOSETENV

Una de las múltiples opciones que pueden establecerse en la sección 'Defaults' u 'opciones' es la opción booleana o de flag 'setenv' que por defecto y para todos los usuarios esta establecida en 'off'. Esta opción si se activa por usuario (Defaults:sergio setenv) permitirá al usuario indicado cambiar el entorno de variables del usuario del cual tiene permisos de ejecutar comandos, y como generalmente este es 'root' pues es obvio que resulta bastante peligrosa esta opción. A nivel de lista de comandos, es posible entonces especificar el tag 'SETENV' a un solo comando o a una pequeña lista de estos y solo cuando se ejecuten estos se podrán alterar su entorno de variables. Es decir, en vez de establecerlo por usuario, sería más conveniente establecerlo por comando a ejecutarse solamente.

ADMIN ALL = SETENV: /bin/date, NOSETENV ALL

A los usuarios definidos en el alias de usuario 'ADMIN' en cualquier host, pueden alterar las variables de entorno cuando ejecuten el comando 'date' (que puede ser útil por ejemplo para cambiar variables del tipo LOCALE), y cualquier otro comando, no tendrá esta opción al habilitar el tag contrario 'NOSETENV'. Y ya que este es el default, también sería válido de la siguiente manera y harían lo mismo:

ADMIN ALL = ALL, SETENV: /bin/date

ARCHIVO /ETC/SUDOERS DE EJEMPLO

Para concluir este manual, veamos un pequeño ejemplo de un archivo /etc/sudoers:

```
# **********************
# LinuxTotal.com.mx, ejemplo de un archivo sudoers
# sergio.gonzalez.duran@gmail.com
# **********************

# **********************
# DEFINCION DE ALIAS
# **********************

# administradores con todos los privilegios
User_Alias ADMINS = sergio, ana

# administradores de red - network operators
User_Alias NETOPS = marcela, andrea

# webmasters -
User_Alias WEBMAS = cristina, juan
```

```
# supervisores de producción (todos los del grupo de sistema supervisores)
User_Alias SUPPRO = samuel, %supervisores

# usuarios que pueden conectarse desde Internet
User_Alias INETUS = NETOPS, ADMINS, samuel

# servidores web
Host_Alias WEBSERVERS = 10.0.1.100, 10.0.1.101

# servidores de aplicaciones
Host_Alias APLICACIONES = WEBSERVERS, 10.0.1.102, 10.0.1.103, mailserver

# comandos de red permitidos
Cmnd_Alias REDCMDS = /sbin/ifconfig, /sbin/iptables

# comandos de apache
Cmnd_Alias APACHECMDS = /usr/sbin/apachectl, /sbin/service httpd *

# ***********************
# DEFINCION DE OPCIONES
# ***********************

# Los usuarios administradores, requieren autentificarse con la contraseña de 'root'
Defaults>ADMINS rootpw

# Para todos los usuarios, tienen hasta dos intentos para ingresar su contraseña y 3 minuto
para que esta expire
Defaults passwd_tries = 4, passwd_timeout = 1

# Los usuarios que se conectan desde Internet, solo tienen una oportunidad y cero timeout lo
que implica
# que cada comando que usen a través de sudo requerira siempre de autentificación.
Defaults:INETUS passwd_tries = 1, passwd_timeout = 0

# Máscara de directorios y archivos por default, para los que ejecuten sudo en los
servidores web
Defaults@WEBSERVERS umask = 022

# ***********************
# DEFINCION DE REGLAS
# ***********************

# administradores todo se les permite en cualquier equipo (¡¡¡¡¡cuidado con esto en la vida
real!!!!!
ADMINS ALL = (ALL) ALL

# administradores de red, en todos los equipos, los comandos de red
NETOPS ALL = REDCMDS

# webmasters, en los servidores web con los comandos indicados en apachecmds y además sin
necesidad
# de contraseña acceder a las bítacoras de apache y reiniciar los servidores.
WEBMAS WEBSERVERS = APACHECMDS, NOPASSWD: /var/log/apache/, /sbin/reboot

# supervisores, pueden ejecutar los comandos indicados en los equipos indicados en el alias
# aplicaciones y además son ejecutados bajo el usuario apps.
SUPPRO APLICACIONES = NOEXEC: (apps) /usr/local/facturacion.exe, /usr/local/ventas.exe,
/usr/local/nomina.exe

# no definidos por alias previos, sino directamente

# regina es de recursos humanos y puede cambiar contraseñas de cualquier usuario menos de
root
regina ALL = /usr/bin/passwd *, !/usr/bin/passwd root
```

```
# david, puede apagar los equipos de aplicaciones
david APLICACIONES = /sbin/shutdown, /sbin/halt

#  El equipo firewall de la red puede ser reiniciado (no apagado) por fernanda que es
asistente de redes
fernanda firewall = /sbin/shutdown -r now
```

Referencias

Como siempre, la referencia más a la mano la tienes en las páginas de manual:

- man sudo
- man visudo
- man sudoers

Y en los siguientes sitios encuentras información que complementa esta manual.

- http://www.sudo.ws/ - sitio oficial de sudo

Sistema de Seguridad “SELinux”

Security-Enhanced Linux (**SELinux**) es un módulo de seguridad para el kernel Linux que proporciona el mecanismo para soportar políticas de seguridad para el control de acceso, incluyendo controles de acceso obligatorios como los del Departamento de Defensa de Estados Unidos. Se trata de un conjunto de modificaciones del núcleo y herramientas de usuario que pueden ser agregadas a diversas distribuciones Linux. Su arquitectura se enfoca en separar las decisiones de las aplicaciones de seguridad de las políticas de seguridad mismas y racionalizar la cantidad de software encargado de las aplicaciones de seguridad. Los conceptos clave que soportan SELinux pueden ser trazados a diversos proyectos previos de la Agencia de Seguridad Nacional de Estados Unidos.
SELinux ha sido integrado a la rama principal del núcleo Linux desde la versión 2.6, el 8 de agosto del 2003.

Links de información:

https://www.nsa.gov/research/selinux/docs.shtml
http://es.wikipedia.org/wiki/SELinux

Ver el estado del SELinux

sestatus

Desabilitar el SELinux

/etc/selinux/config

(buscar y modificar la siguiente variable)
SELINUX=disabled