¿Requieres de una instalación o configuración de Linux o sus servicios?
¿Un desarrollo WEB empresarial a la medida?
¿Un curso o capacitación a la medida?
Revisa el sitio de SERVICIOS (index.php?cont=servicios) de LinuxTotal
LINUXTOTAL.COM.MX - Información y servicios en Linux y Open Source
URL: http://www.linuxtotal.com.mx/index.php?cont=info_admon_014

Manual de sudo, visudo y sudoers

Copyright © 2005-2019 LinuxTotal.com.mx
Se concede permiso para copiar, distribuir y/o modificar este documento siempre y cuando se cite al autor y la fuente de linuxtotal.com.mx y según los términos de la GNU Free Documentation License (http://www.gnu.org/licenses/translations.html), Versión 1.2 o cualquiera posterior publicada por la Free Software Foundation.

Autor: Sergio González D. **(sergio.gonzalez.duran@gmail.com)**

En ambientes donde varios usuarios usan uno o más sistemas GNU/Linux, es necesario otorgar distintos permisos o privilegios para que estos puedan hacer uso de comandos propios del usuario administrador 'root'. Totalmente fuera de lugar e impensable es 'entregar' la contraseña de root para que los usuarios puedan hacer uso de los programas propios de sus funciones pero que son propiedad de 'root'. Por otro lado, hacer uso del comando **su** tampoco es práctico porque es lo mismo, necesitan la contraseña de root, asi que la mejor alternativa es hacer uso de **sudo**.

¿Exáctamente que es y que hace **sudo**?. sudo permite implementar un control de acceso altamente granulado de que usuarios ejecutan que comandos. Si un usuario normal desea ejecutar un comando de root (o de cualquier otro usuario), **sudo** verifica en su lista de permisos y si está permitido la ejecución de ese comando para ese usuario, entonces **sudo** se encarga de ejecutarlo. Es decir, **sudo** es un programa que basado en una lista de control (/etc/sudoers) permite (o no) la ejecución al usuario que lo invocó sobre un determinado programa propiedad de otro usuario, generalmente del administrador del sistema 'root'.

sudo, para fines prácticos se puede dividir en tres partes:

- sudo, el comando con permisos de SUID, que los usuarios usan para ejecutar otros comandos a los que se les permite usar.
- visudo, el comando que permite al administrador modificar /etc/sudoers.
- /etc/sudoers, el archivo de permisos que le indica a **sudo** que usuarios ejecutan cuáles comandos.

sudo

sudo (SUperuser DO) lo ejecuta un usuario normal, al que se supone tiene permisos para ejecutar cierto comando. Entonces, **sudo** requiere que los usuarios se autentifiquen a si mismos a través de su contraseña para permitirles la ejecución del comando. Veamos un ejemplo:

```
$ sudo /sbin/ifconfig
Password:
eth0      Link encap:Ethernet  HWaddr 4C:00:10:60:5F:21
          inet addr:200.13.110.62  Bcast:200.13.110.255  Mask:255.255.255.0
          inet6 addr: fe80::4e00:10ff:fe60:5f21/64 Scope:Link
...
```

Como se podrá observar se usa el comando **sudo** seguido del comando (con toda su ruta si es que este no esta en el PATH del usuario) al que se tiene permiso. **sudo** pregunta por la contraseña del usuario que ejecuta el comando y listo.

Por defecto, después de hacer lo anterior tendrás 5 minutos para volver a usar el mismo comando u otros a los que tuvieras derecho, sin necesidad de ingresar la contraseña de nuevo. Si se quiere extender el tiempo por otros 5 minutos usa la opción *-v* (validate). Por el contario, si ya terminaste lo que tenías que hacer, puedes usar *-k* (kill) para terminar con el tiempo de gracia de validación.

Ahora bien, ¿Qué comandos son los que puedo utilizar?, pues la opción *-l* es la indicada para eso:

```
$ sudo -l
User sergio may run the following commands on this host:
    (root) /sbin/ifconfig
    (root) /sbin/lspci
```

En el caso anterior se ejecutó un comando de root, pero no tiene que ser asi, también es posible ejecutar comandos de otros usuarios del sistema indicando la opción *-u*:

```
$ sudo -u ana /comando/de/ana
```

Una de las opciones más interesantes es la que permite editar archivos de texto de root (claro, con el permiso otorgado en 'sudoers' como se verá más adelante), y esto se logra con la opción *-e*, esta opción esta ligada a otro comando de **sudo** llamado **sudoedit** que invoca al editor por defecto del usuario, que generalmente es **vi**.

```
$ sudo -e /etc/inittab
(Permitira modificar el archivo indicado como si se fuera root)
```

Cuando se configura **sudo** se tienen múltiples opciones que se pueden establecer, estás se consultan a través de la opción *-L*

```
$> sudo -L
Available options in a sudoers ``Defaults'' line:

syslog: Syslog facility if syslog is being used for logging
syslog_goodpri: Syslog priority to use when user authenticates successfully
syslog_badpri: Syslog priority to use when user authenticates unsuccessfully
long_otp_prompt: Put OTP prompt on its own line
ignore_dot: Ignore '.' in $PATH
mail_always: Always send mail when sudo is run
mail_badpass: Send mail if user authentication fails
mail_no_user: Send mail if the user is not in sudoers
mail_no_host: Send mail if the user is not in sudoers for this host
mail_no_perms: Send mail if the user is not allowed to run a command
tty_tickets: Use a separate timestamp for each user/tty combo
lecture: Lecture user the first time they run sudo
lecture_file: File containing the sudo lecture
authenticate: Require users to authenticate by default
root_sudo: Root may run sudo
...
varias opciones más
```

Bastante útil, ya que nos muestra las opciones y una pequeña descripción, estás opciones se establecen en el archivo de configuración 'sudoers'.

Una de las opciones más importantes de consulta es *-V*, que permite listar las opciones (defaults) establecidas por defecto para **sudo** todos los usuarios, comandos, equipos, etc. Más adelante en este tutorial, aprenderemos como establecer opciones específicas para ciertos usuarios, comandos o equipos. NOTA: tienes que ser 'root' para usar esta opción.

```
# sudo -V
Sudo version 1.6.9p5

Sudoers path: /etc/sudoers
Authentication methods: 'pam'
Syslog facility if syslog is being used for logging: local2
Syslog priority to use when user authenticates successfully: notice
Syslog priority to use when user authenticates unsuccessfully: alert
Send mail if the user is not in sudoers
Lecture user the first time they run sudo
Require users to authenticate by default
Root may run sudo
Log the hostname in the (non-syslog) log file
Allow some information gathering to give useful error messages
Visudo will honor the EDITOR environment variable
Set the LOGNAME and USER environment variables
Reset the environment to a default set of variables
Length at which to wrap log file lines (0 for no wrap): 80
Authentication timestamp timeout: 5 minutes
Password prompt timeout: 5 minutes
Number of tries to enter a password: 3
Umask to use or 0777 to use user's: 022
Path to log file: /var/log/sudo.log
...
varias opciones más listadas
```

Con intención, trunque el listado anterior en la línea "Path to log file: /var/log/sudo.log", donde se indica cual es el archivo 'log' o de bitacora por defecto de **sudo**, en este archivo se loguea absolutamente todo lo que se haga con **sudo**, que usuarios ejecutaron que, intentos de uso, etc.

visudo

Permite la edición del archivo de configuración de **sudo** sudoers. Invoca al editor que se tenga por defecto que generalemente es **vi**. **visudo** cuando es usado, bloquea el archivo /etc/sudoers de tal manera que nadie más lo puede utilizar, esto por razones obvias de seguridad que evitarán que dos o más usuarios administradores modifiquen accidentalmente los cambios que el otro realizó.

Otra característica importande de **visudo** es que al cerrar el archivo, verifica que el archivo este bien configurado, es decir, detectará si hay errores de sintaxis principalmente en sus múltiples opciones o reglas de acceso que se tengan. Por esta razón no debe editarse /etc/sudoers directamente (perfectamente posible ya que es un archivo de texto como cualquier otro) sino siempre usar **visudo**.

Si al cerrar **visudo** detecta un error nos mostrará la línea donde se encuentra, y la pregunta "What now?":

```
>>> sudoers file: syntax error, line 15 <<<
What now?
```

Se tienen tres opciones para esta pregunta:

- e - edita de nuevo el archivo, colocando el cursor en la línea del error (si el editor soporta esta función.)
- x - salir sin guardar los cambios.
- Q - salir y guarda los cambios.

Por defecto el archivo de configuración es /etc/sudoers pero se pueden editar otros archivos que no sean ese y que se aplique la sintaxis de **sudo**, y esto se logra con la opción *-f* (**visudo -f /otro/archivo**).

Si tan solo se desea comprobar que /etc/sudoers esta bien configurado se usa la opción *-c*, toma por el archivo de configuración por defecto o si no se indica algún otro.

```
#> visudo -c
/etc/sudoers file parsed OK
```

La opción *-s* activa el modo 'estricto' del uso de **visudo**, es decir no solo se comprobará lo sintáctico sino también el orden correcto de las reglas, por ejemplo si se define el alias para un grupo de comandos y este se usa antes de su definición, con esta opción se detectará este tipo de errores.

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 defincón 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* esta 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 como 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 refiridos 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 tipo 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 varian 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, asi 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

Aqui 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, asi 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.

Opciones (defaults)

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, asi 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 requerira auténticarse. 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.

Reglas de acceso

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 utiliar iptables.

ADMIN ALL = ALL

Los usuarios definifos 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 contendido de los archivos que contenga el directorio /var/log.

Lo anterior intoduce 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 asi. 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 mas conveniente establecerlo por comando a ejcutarse 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/ (http://www.sudo.ws/) - sitio oficial de sudo
- onlamp (http://www.onlamp.com/pub/a/bsd/2002/08/29/Big_Scary_Daemons.html) - sitio en inglés con una explicación muy completa de como funciona sudo.

¿Requieres de una instalación o configuración de Linux o sus servicios?
¿Un desarrollo WEB empresarial a la medida?
¿Un curso o capacitación a la medida?
Revisa el sitio de SERVICIOS (index.php?cont=servicios) de LinuxTotal

Copyright © LinuxTotal.com.mx 2006-2019
info@linuxtotal.com.mx · linuxtotal.com.mx@gmail.com