Invocar el poder del superusuario con sudo

Sudo Simple

Usar el comando sudo puede hacer que nuestro sistema sea más seguro. Mostraremos cómo utilizar su potencia sin temor a equivocarnos.

POR BRUCE BYFIELD

Para millones de usuarios, *sudo* [1] es la palabra mágica que les permite ejecutar ciertos comandos en Ubuntu. Sin embargo, *sudo* no es exclusivo de Ubuntu, y además con él podemos hacer mucho más que con la configuración por defecto de esta distro.

Sudo es fundamentalmente un comando de seguridad. Su propósito es ejecutar comandos como otro usuario – típicamente, como en Ubuntu, para minimizar el tiempo que una aplicación se está ejecutando con poderes de root y sea vulnerable a los ataques. Sin embargo, un usuario root también podría utilizar sudo para, por ejemplo, comprobar las configuraciones del entorno de otra cuenta. La mayoría de las veces, sudo abre una línea de comandos en la cuenta que se desea emular, aunque, con la mejora del escritorio GNU/Linux, se ha desarrollado *gksudo* para ejecutar aplicaciones gráficas utilizando *sudo*.

```
nanday:~# visudo
visudo: /etc/sudoers busy, try again later
nanday:~#
```

Figura 1: *visudo* bloquea */etc/sudoers* para que nadie más pueda editarlo mientras nosotros lo hacemos.

Cuando introducimos este comando desde una cuenta que no sea root, debemos dar una contraseña antes de poder ejecutarlo. En Ubuntu y en muchas otras distribuciones, esta contraseña es la del login para la cuenta habitual que estamos usando actualmente. Sin embargo, dependiendo de cómo esté configurado sudo, puede que tengamos que introducir en su lugar la contraseña de root. Una vez que ejecutemos un comando utilizando sudo, podemos volver a usar sudo sin una contraseña durante un rato – generalmente de 5 a 15 minutos.

Estas configuraciones y otras se establecen mediante opciones del comando básico, o a través de campos en */etc/sudoers*. Este fichero de configuración debe editarse con *visudo* por razones de seguridad.

Dentro de */etc/sudoers* existen tres tipos de entradas: alias, predeterminadas y especificaciones del usuario. Todas pueden hacer referencia a usuarios individuales, a grupos definidos en */etc/groups*, a grupos de red o a grupos creados por alias en el fichero sudoers. Todas pueden también usar un número limitado de comodines. Por ejemplo, un asterisco (*) coincide con cualquier número de caracteres, mientras que un signo de interrogación (?) corresponde a cualquier carácter individual. Habitualmente, por comodidad, se encuentran organizadas – como la mayoría de los ficheros de configuración – por líneas de comentario que comienzan con el símbolo almohadilla (#).

Antes de editar el fichero sudoers debemos tener en cuenta otras dos peculiaridades: un signo de exclamación delante del valor significa que no lo usemos. Incluso más importante, siempre debemos dar la ruta completa a cada comando que se menciona en

sudoers para asegurarnos de que se encontrará sin demora ni dificultad.

Con esta información preliminar, estamos preparados para comenzar a explorar sudo – y, si deseamos, hacer ajustes en su funcionamiento en Ubuntu o en cualquier otra distribución. No olvidemos hacer copia de seguridad de */etc/sudoers* antes de salir para que podamos recuperarlo rápidamente en caso de emergencia.

Uso de visudo

En muchas distribuciones, el fichero predeterminado */etc/sudoers* comienza notificando que los cambios deben realizarse con el comando *visudo*. Estrictamente hablando, esto no es cierto – no hay nada que nos impida abrir directamente */etc/sudoers* en un editor de texto. Sin embargo, no hay ninguna buena razón para no aprovechar las ventajas de visudo y, de hecho sería temerario hacerlo.

El comando *visudo* tiene dos virtudes. Primero, bloquea */etc/sudoers* cuando lo editamos, lo que evita que otros usuarios de sistemas multi-usuario intenten editarlo al mismo tiempo (Figura 1). En un sistema Debian o derivado, como Ubuntu, el fichero bloqueado es */etc/sudoers.tmp*, el cual guarda los cambios que hacemos hasta que lo salvemos.

En segundo lugar, visudo comprueba el fichero sudoers en busca de errores (Figura 2). Cuando lo guardamos muestra un mensaje listando cada error. Si el error es una entrada no soportada en un campo, entonces sólo lista la línea en la que ocurre y continúa con el proceso de guardado. Pero si el error está en el nombre de un campo o es algún otro error sintáctico, entonces aparece el prompt *What Now?*, teniendo que decidir si introducir *e* (de re-editar), *x* (salir sin guardar) o *q* (salir y guardar). Si elegimos *q* no podremos usar sudo hasta que se corrijan todos los errores de sintaxis.

```
Warning: undeclared Cmnd_Alias `AL' referenced near line 20
>>> sudoers file: syntax error, line 21 <<<
What now? e
nanday:~#
```

Figura 2: Una de las ventajas de *visudo* es que detecta automáticamente errores , dándonos la oportunidad de corregirlos.

```
bruce@ubuntu:~$ sudo select-editor

Select an editor.  To change later, run 'select-editor'.
  1. /bin/ed
  2. /bin/nano        <---- easiest
  3. /usr/bin/vim.tiny

Choose 1-3 [2]: 3
bruce@ubuntu:~$
```

Figura 3: Ubuntu tiene una opción especial para *sudo* que nos ayuda a seleccionar el editor usado por *visudo*.

Cuando ejecutamos visudo, éste abre uno de los editores de texto listados en */usr/bin/editor*. Por defecto, ignora las configuraciones del entorno *VISUAL* y *EDITOR*. Sin embargo, podemos configurar el editor por defecto utilizado por visudo en el fichero sudoers. En Ubuntu, también podemos utilizar el comando *sudo select-editor* para establecer el editor por defecto (Figura 3).

Las opciones para visudo son pocas. Para pedirle que compruebe errores y que informe de si sudoers tiene algún error o no, usamos la opción *-c* (Figura 4). Para una comprobación de la sintaxis más estricta escribimos *-s*. Con *-f* podemos especificar otro fichero sudoer y editarlo, lo que nos permite experimentar sin el riesgo de crear problemas con el fichero original (aunque nunca es mala idea hacer primero una copia de seguridad).

```
nanday:~# visudo -c
/etc/sudoers file parsed OK
```

Figura 4: El comando *visudo* no ofrece un informe detallado de errores - sólo un mensaje diciéndonos si existen problemas o no.

```
Below are example sudoers entries.  Admittedly, some of these are a bit
contrived.  First, we define our aliases:

 # User alias specification
 User_Alias     FULLTIMERS = millert, mikef, dowdy
 User_Alias     PARTTIMERS = bostley, jwfox, crawl
 User_Alias     WEBMASTERS = will, wendy, wim

 # Runas alias specification
 Runas_Alias    OP = root, operator
 Runas_Alias    DB = oracle, sybase

 # Host alias specification
 Host_Alias     SPARC = bigtime, eclipse, moet, anchor :\
                SGI = grolsch, dandelion, black :\
                ALPHA = widget, thalamus, foobar :\
                HPPA = boa, nag, python
 Host_Alias     CUNETS = 128.138.0.0/255.255.0.0
 Host_Alias     CSNETS = 128.138.243.0, 128.138.204.0/24, 128.138.242.0
 Host_Alias     SERVERS = master, mail, www, ns
 Host_Alias     CDROM = orion, perseus, hercules

 # Cmnd alias specification
 Cmnd_Alias     DUMPS = /usr/bin/mt, /usr/sbin/dump, /usr/sbin/rdump,\
                        /usr/sbin/restore, /usr/sbin/rrestore
 Cmnd_Alias     KILL = /usr/bin/kill
 Cmnd_Alias     PRINTING = /usr/sbin/lpc, /usr/bin/lprm
 Cmnd_Alias     SHUTDOWN = /usr/sbin/shutdown
 Cmnd_Alias     HALT = /usr/sbin/halt
 Cmnd_Alias     REBOOT = /usr/sbin/reboot
Manual page sudoers(5) line 859
```

Figura 5: La página man de *sudoers* ofrece un ejemplo detallado de alias.

Edición de /etc/sudoers: Alias

Los alias son grupos de usuarios, ordenadores y comandos (Figura 5). Se configuran de manera que puedan usarlos las opciones en sudoers, en lugar de listar grupos individuales o cada uno de los usuarios. Normalmente se utilizan sólo en redes. En una sola máquina, especialmente en casa, no tienen tanta importancia – a menos que deseemos salvaguardar nuestra instalación de las manipulaciones de nuestros hijos.

Los alias definen grupos especialmente para el uso de especificaciones de usuario. Podemos introducir cuatro tipos de alias: *User_Alias*, *Runas_Alias*, *Host_Alias* y *Cmnd_Alias*. Los alias de los usuarios listan usuarios o grupos, mientras que el alias *Runas* lista usuarios o grupos pero pueden utilizar UIDs para identificarlos – una funcionalidad que en la práctica es tan poco interesante, que el alias *Runas* raramente es utilizado. De manera similar, los alias *Host* agrupan nombres de ordenadores o direcciones y el alias *Command* los comandos.

Cada alias consta de una única línea, que comienza con el tipo de alias que está siendo definido, seguido del nombre del alias y de las entradas en el alias en una lista separada por una coma. Por ejemplo, el siguiente comando define el alias *SYSADMINS*,

```
User_Alias SYSADMINS = goulet, ➥
ng, singh
```

mientras que

```
# Override built-in defaults
Defaults                syslog=auth
Defaults>root           !set_logname
Defaults:FULLTIMERS     !lecture
Defaults:millert        !authenticate
Defaults@SERVERS        log_year, logfile=/var/log/sudo.log
Defaults!PAGERS         noexec
```

Figura 6: La página man de *sudoers* proporciona ejemplos detallados de cómo invalidar los valores predeterminados de sudo con entradas en el fichero *sudoers*.

```
Host_Alias SERVERS = firefly, ➥
serenity, buffy, angel, ➥
dollhouse
```

creará el alias *SERVERS*:

De manera similar, podríamos usar el alias *Command* para controlar el acceso a determinados comandos administrativos. Por ejemplo, la línea:

```
Cmnd_Alias SHUTDOWN = ➥
/usr/sbin/shutdown, ➥
/usr/sbin/halt, ➥
user/sbin/reboot
```

puede usarse para controlar quién puede apagar o reiniciar la máquina.

Los grupos creados por alias pueden ser definidos específicamente en el fichero sudoers. Sin embargo, si van precedidos por un signo de porcentaje (%), harán referencia a un grupo del sistema; si fuera un signo más (+), lo harán a un grupo de red.

Edición de sudoers: Invalidando los Parámetros Predeterminados

Cuando ejecutamos *sudo*, éste utiliza unos cuantos comportamientos predeterminados, como por ejemplo solicitarnos que introduzcamos nuestra contraseña antes de ejecutarse. Para modificar esto podemos usar */etc/sudoers*.

El formato para cada modificación es *Defaults: [ALIAS o USUARIO] PARÁMETRO = [VALOR]* (Figura 6). El alias o usuario es opcional pero, si no está especificado, se aplica a cada cuenta. El valor del parámetro depende de si se requiere uno. Algunas modificaciones simplemente necesitan ser introducidas para que hagan efecto; otras precisan de un valor. Por ejemplo, si queremos utilizar el parámetro *editor* con su valor por defecto de *vim* cuando usamos *visudo*, entonces añadiremos la línea *Defaults: editor*. Por otro lado, si queremos asegurarnos de que visudo siempre abra el editor *Nano*, entonces tendremos que introducir *Defaults: editor = /usr/bin/nano*.

El fichero sudoers es especialmente rico en opciones para el uso de contraseñas. El comportamiento sin modificar de sudo requiere que los usuarios introduzcan su propia contraseña antes de ejecutar nada. Pero, si hemos editado el comportamiento predeterminado, podemos restaurarlo en un caso especial con *passwd*. Para que un usuario específico o un alias ejecute un comando sin una contraseña (lo cual *no* es buena idea), podremos usar en su lugar *nopasswd*. Para incrementar la seguridad básica, *rootpw* requiere que los usuarios de sudo den la contraseña de root antes de ejecutar un comando. De forma similar, *targetpw* necesita la contraseña del usuario que va a ejecutarlo.

Para depurar aún más el uso de las contraseñas usamos *passwd_tries = [VALOR]*. Si alguien se equivoca al teclear la contraseña, podemos utilizar *badpass_message = "[VALOR]"* para cambiar el mensaje por defecto *Sorry, try again*. Si alguien falla consistentemente, podemos usar *insults* para que presente un mensaje grosero cuando se equivoque.

Para establecer el número de minutos antes de que expire el acceso garantizado por sudo usamos *timestamp_timeout = [VALOR]*. Si configuramos el valor a *0*, sudo expira inmediatamente después de que se ejecute un comando. Para alertar a los usuarios sobre los riesgos de seguridad asociados a sudo, podemos utilizar *lecture* para que nos presente un aviso (Figura 7) – con un valor de *always*, *once* o *never* – y *lecture_file* para apuntar a la ruta de un texto de aviso alternativo.

Otras opciones para añadir seguridad son *mail_always*, que envía un mensaje a quienquiera que reciba correo de root de todas las actividades de sudo, o *mail_badpass*, que remite un mensaje cuando alguien falla al introducir la contraseña. En la página *man* de sudoers se encuentra disponible una lista completa de parámetros editables. Como alternativa, podemos usar el comando *sudo -L* para ver un resumen de los parámetros disponibles.

Edición de sudoers: Especificaciones de Usuario

Las especificaciones de usuario configuradas en sudoers definen qué pueden hacer usuarios o grupos, o bien cómo funciona sudo en general (Figura 8). Estas especificaciones tienen el siguiente formato básico:

```
[USUARIO] [HOST]= ➥
([USUARIO OBJETIVO]) [COMANDOS]
```

El usuario objetivo es opcional, pero los demás valores deben estar presentes. Cualquiera de éstos puede sustituirse por un alias apropiado.

En un sistema simple, como una estación de trabajo doméstico, esta fórmula puede ser muy sencilla. Por ejemplo, en el sudoers por defecto en Ubuntu, la opción *%admin ALL = (ALL) ALL* da a todos los miembros del grupo del sistema *admin* (lo que significa desde el primer usuario creado durante la instalación, más cualquiera añadido después) el derecho sobre todas las máquinas para usar cualquier comando como cualquier usuario (Figura 9). Como probablemente sólo se trate de una única máquina y no más que un puñado de usuarios, utilizando *ALL* casi no afecta a la seguridad.

En una red, sin embargo, es probable que queramos ser más precisos en nuestras especificaciones de usuario. Si tuviéramos un alias llamado *MYSQL* para los administradores de la base de datos y un Host Alias para los servidores dedicados de la base de datos llamado *DATABASE*, la definición sería *MYSQL DATABASE = ALL*. Esta especificación de usuario daría a todos los miembros de *MYSQL* los permisos para ejecutar todos los comandos en

```
We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.
```

Figura 7: Por defecto, la primera vez que usamos *sudo* muestra un aviso con reminiscencias a Spider Man.

cualquier máquina en el grupo *DATABASE*, pero no en cualquier grupo de máquinas.

Si deseamos limitar las acciones permitidas del grupo a detener o reiniciar los servidores de bases de datos, la especificación del usuario sería *MYSQL DATABASE = /usr/sbin/shutdown, /usr/sbin/halt, /usr/sbin/reboot* o, si usamos el alias Command *SHUTDOWN* mencionado antes, la especificación de usuario tendría este aspecto: *MYSQL DATABASE = SHUTDOWN*, porque el alias Command *SHUTDOWN* contiene todos esos comandos.

Otra manera de establecer límites en *ALL* es clasificarlo mediante una lista de lo que el usuario o grupo no tiene permitido hacer. En lugar de la especificación de usuario *MYSQL DATABASE = ALL*, podríamos utilizar un signo de exclamación en la lista de comandos para definir lo que no puede hacer *MYSQL*. Por ejemplo, si la especificación de usuario es *MYSQL DATABASE = ALL, !SHUTDOWN*, entonces los usuarios de *MYSQL* pueden ejecutar todos los comandos en las máquinas en *DATABASE* excepto aquellos que reinician o apagan el ordenador.

Las especificaciones de usuario también pueden modificarse añadiéndoles una etiqueta. Las etiquetas más comunes son *NOPASSWD* y *PASSWD*, que pueden usarse para establecer cuándo un usuario necesita introducir una contraseña cada vez que ejecuta un comando. Por ejemplo, si quisiéramos que *MYSQL* usara los comandos *SHUTDOWN* sin tener que introducir una contraseña, entonces la especificación sería *MYSQL DATABASE = NOPASSWD: SHUTDOWN*.

Para ahorrar tiempo podemos introducir múltiples especificaciones para el mismo usuario en la misma línea, con cada especificación separada por una coma. Por ejemplo, la definición *valerie SERVER = SHUTDOWN : DATABASE = ALL* otorgaría al usuario *valerie* permiso para apagar todas las máquinas del grupo *SERVER* y ejecutar cualquier comando en las máquinas del grupo *DATABASE*.

Ejecutando sudo

Comparado con la configuración de sudoers, ejecutar el comando sudo es en realidad sencillo. En el abrumador número de casos, sólo necesitamos introducir *sudo <commando>*, aunque mucha gente probablemente se sorprenda cuando sepa que puede hacer mucho más.

Sudo tiene unas cuantas opciones útiles. Si usamos el parámetro *-e* seguido del nombre de un fichero, podemos abrir el fichero para editarlo – una opción especialmente útil cuando estamos editando ficheros de configuración, algo que a menudo requiere que nos registremos como root. Como alternativa, también podemos utilizar el comando *sudoedit [FICHERO]*.

```
The User specification is the part that actually determines who may run
what.

 root          ALL = (ALL) ALL
 %wheel        ALL = (ALL) ALL

We let root and any user in group wheel run any command on any host as any
user.

 FULLTIMERS    ALL = NOPASSWD: ALL

Full time sysadmins (millert, mikef, and dowdy) may run any command on any
host without authenticating themselves.

 PARTTIMERS    ALL = ALL

Part time sysadmins (bostley, jwfox, and crawl) may run any command on any
host but they must authenticate themselves first (since the entry lacks the
NOPASSWD tag).
```

Figura 8: Las páginas man de *sudoers* ofrecen ejemplos de cómo configurar las especificaciones de usuario en sudoers.

```
# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL) ALL

# Allow members of group sudo to execute any command after they have
# provided their password
# (Note that later entries override this, so you might need to move
# it further down)
%sudo ALL=(ALL) ALL
#
#includedir /etc/sudoers.d

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

Figura 9: El fichero sudoers por defecto en Ubuntu (y en la mayoría de las distribuciones) es muy breve. Sin embargo, si queremos más seguridad o flexibilidad, debemos prepararnos para editarlo.

Algunos comandos afectan a la cantidad de tiempo que podemos utilizar sudo antes de que expire. Si añadimos una *-v*, renovaremos la marca de tiempo. Por otro lado, si hemos acabado con sudo y nos preocupa la seguridad, *-K* finalizará inmediatamente la sesión sudo.

Otras opciones anulan temporalmente los funcionalidades predeterminadas de sudo establecidas en las configuraciones predefinidas en sudoers. Con la opción *-p* podemos invalidar la contraseña requerida. Por ejemplo, *-p%h* necesita una contraseña para el nombre de host local, mientras que *-p%U* la requiere para el usuario objetivo (normalmente root). Asimismo, la opción *-E* invalida cualquier configuración de entorno, forzando a sudo a utilizar nuestro entorno actual.

Buscando lo que Funciona

Como podemos ver, sudo es un comando infinitamente más flexible de lo que hubiéramos podido imaginar. Con los valores predeterminados de sudo, las opciones en sudoers y el comando *sudo* en sí, tenemos más que suficientes opciones para asegurarnos de que sudo se ejecutará del modo que mejor se adapte a nuestras necesidades.

Si nos encontramos perdidos, y los ejemplos de la página man de sudoers no nos ayudan, una búsqueda por "sudo" y "examples" nos mostrará numerosos ejemplos online, particularmente la página del proyecto sudo [2]. Incluso encontraremos una lista de programas alternativos [3].

Habituarse a sudo lleva su tiempo, pero los resultados pueden ser un sistema configurado de acuerdo a nuestras preferencias y mucho más seguro. Por tanto, aprender este comando en profundidad se merece todo el esfuerzo que podamos ponerle. ■

RECURSOS

[1] sudo: *http://www.sudo.ws*

[2] Página del proyecto sudo: *http://www.sudo.ws/sudo/sample.sudoers*

[3] Programas alternativos: *http://www.sudo.ws/sudo/other.html*