

Ejecución de programas como root

ÍNDICE:

1. Ejecución de programas como Administrador.

1.1. Comando sudo.

1.2. Comandos gksu y gksudo.

1.3. Comando su.

1.4. Activar usuario root en Ubuntu.

1.5. Limitar los usuarios que pueden cambiar a root.

1. Ejecución de programas como Administrador.

Ya hemos comentado que en todo sistema Linux existe un usuario llamado root que tiene todos los privilegios sobre el sistema.

En realidad, el usuario root es un usuario normal, que tiene unos privilegios adicionales porque el kernel se lo permite. En el kernel Linux existen unas series de funciones denominadas "Capabilities" que determinan qué puede y que no puede hacer root en el sistema.

En Ubuntu, el usuario root viene desactivado por defecto, así que para poder realizar tareas administrativas o que necesiten permiso de administrador (root), se usa el comando sudo.

1.1. Comando sudo.

El comando "sudo" permite que usuarios normales (sin privilegios de administrador) puedan ejecutar tareas propias del usuario root.

Para ello, el comando sudo comprueba que el usuario que quiere ejecutar algo como root, aparece en el fichero de configuración de sudo y además tiene los permisos necesarios para ejecutar la tarea en cuestión.

El fichero "/etc/sudoers" es donde se indican que usuarios pueden ejecutar tareas como root y que acciones pueden llevar a cabo. El sistema mantiene un registro de todas las actividades realizadas por esos usuarios autorizados en el fichero "/var/log/auth.log".

Una vez ejecutamos sudo y el programa comprueba que tenemos permiso para ejecutar la tarea como root, nos pedirá que insertemos la contraseña del usuario que ejecuta sudo. Si es correcta iniciará la tarea como root.

Esta contraseña será recordada por defecto durante 15 minutos, por lo que si en ese tiempo volvemos a ejecutar sudo no nos pedirá la contraseña. Este tiempo puede ser cambiado en el fichero "/etc/sudoers".

Aunque el fichero "/etc/sudoers" se puede editar con cualquier editor de textos, se recomienda usar el comando "visudo", ya que este programa antes de guardar el archivo sudoers realiza una verificación para comprobar si hay errores sintácticos.

1.1.1. Uso de sudo.

⇒ Para ejecutar un comando como administrador ponemos:

`$sudo <comando>`

⇒ Si lo que queremos es loguearnos como root sin activar de manera general la cuenta podemos hacer lo siguiente:

`$sudo su –` (nos cambiará al directorio /root)

O también: `$sudo –s`

O también: `$sudo su` (En este caso permaneceremos en el directorio en el que nos encontremos al hacer sudo)

⇒ Para salirnos de la sesión como usuario root y volver a nuestro usuario:

`#exit`

1.1.2. Visudo.

Mediante visudo podremos editar el archivo `/etc/sudoers` de una manera segura, evitando ediciones simultáneas por parte de varios usuarios ya que bloquea el archivo `"/etc/sudoers"` mientras se esté editando.

Además, al finalizar la edición y antes de guardar el archivo modificado, visudo realiza una comprobación buscando errores sintácticos. Si encuentra errores nos indica donde está el error y nos pregunta que queremos hacer. Las opciones son "e" de editar, "x" de salir sin guardar los cambios y "Q" de salir guardando los cambios (opción no recomendada).

Para ejecutar visudo escribimos en la terminal:

`$sudo visudo`

1.1.3. Tareas básicas de configuración de sudo.

Registrar un usuario en sudo

Para hacer que un usuario pueda ejecutar tareas de administrador, lo añadimos al grupo sudo:

`$sudo adduser usuario sudo`

Cambiar el tiempo que sudo recuerda la contraseña

Por defecto, el tiempo que sudo recuerda la contraseña del usuario que ejecuta sudo es de 15 minutos.

En entornos donde la seguridad es importante este tiempo puede ser un agujero de seguridad. Para cambiar este tiempo hacemos lo siguiente:

1) Editamos el fichero de configuración de sudo:

`$sudo visudo`

2) En el editor, buscamos la línea que tiene el siguiente contenido:

```
Defaults    env_reset
```

y la cambiamos por:

```
Defaults    env_reset , timestamp_timeout = X
```

donde X es un valor numérico que indica los minutos que recuerda la contraseña sudo.

Si ponemos cero, pedirá la contraseña cada vez que se utilice sudo.

1.2. Comandos gksu y gksudo.

El comando "sudo" sirve para lanzar aplicaciones que no son gráficas.

Si queremos ejecutar aplicaciones gráficas (como gedit) en entornos gráficos (gnome y algunos otros) debemos usar el comando "gksu" o "gksudo". En entornos gráficos KDE debemos usar "kdesudo".

Si usamos las aplicaciones gráficas (como gedit, firefox,...) con sudo, puede ocurrir que tengamos problemas con los privilegios de usuario de nuestros archivos, pues con sudo estamos como usuario root pero con el entorno del usuario que estemos usando en ese momento, por lo que se pueden crear ficheros nuevos en nuestras carpetas en las que el propietario sea root.

Sin embargo, si usamos "gksu", además de usar la aplicación como usuario root, estamos utilizando el entorno de trabajo del usuario root, por lo que no se guardarían en nuestras carpetas de usuario, ficheros en los que el propietario fuera root que impedirían nuestro acceso.

⇒ El comando gksu lo podemos usar de dos maneras:

➤ **\$gksu**

Al ejecutarlo nos aparece un pequeño cuadro de diálogo en el que nos pide un programa a ejecutar y con que usuario quiero hacerlo.

➤ **\$gksu <nombre del programa>**

Al ejecutarlo junto con el nombre del programa lo ejecutará como su fuéramos el usuario root.



En Ubuntu, "gksudo" es sólo un enlace simbólico a "gksu", por lo que hacen lo mismo.



Recordar que desde el entorno de escritorio podemos ejecutar comandos pulsando las teclas <ALT + F2>.

1.3. Comando su.

⇒ El comando "su" permite cambiar de usuario en el mismo terminal en el que se está trabajando. Para utilizarlo:

1) Hay dos maneras de hacerlo:

- **\$su <nuevousuario>** (cambia de usuario pero mantiene el entorno del usuario anterior).
- **\$su - <nuevousuario>** (se ejecutarán los scripts de inicio de sesión para el nuevo usuario y tendremos una sesión idéntica a la obtenida con la entrada con login para ese nuevo usuario).

2) Introducimos la contraseña del usuario al que pretendemos cambiar.

3) Para terminar esa sesión escribimos:

\$exit.

⇒ Esta orden también permite al usuario normal cambiar a los privilegios de root de dos formas diferentes:

1) Para ello ponemos:

- **\$su** (en este caso seguimos en el entorno del usuario que estaba pero ahora con los permisos de root).
- **\$su -** (cambiamos al entorno de root).

2) Introducimos la contraseña de root.

1.4. Activar usuario root en Ubuntu.

Como hemos comentado antes, en Ubuntu el usuario root está deshabilitado por defecto. Si queremos usar root como en otras distribuciones Linux lo único que tenemos que hacer es crearle una contraseña. Para ello:

1) Solicitamos cambiar la contraseña de root:

\$ sudo passwd root

2) Escribimos la nueva contraseña. La pedirá dos veces por seguridad.

Desactivar cuenta root

1) Para desactivar nuevamente la cuenta root, ponemos lo siguiente:

```
$ sudo passwd -l root
```

1.5. Limitar los usuarios que pueden cambiar a root.

Si queremos limitar que usuarios pueden cambiar a root hacemos lo siguiente:

1) \$sudo nano /etc/pam.d/su

```
----- Fichero /etc/pam.d/su -----  
# línea 15: descomentar y añadir los grupos que pueden cambiar a root  
auth required pam_wheel.so group=adm  
----- Fichero /etc/pam.d/su -----
```

2) \$sudo gpasswd -a <usuario> adm

El grupo “adm” es el grupo por defecto de administración.