Sudo sin contraseña

Configuración y uso de `sudo` en sistemas linux

Introducción sobre el comando `sudo`

¿Qué Hace `sudo`?

El comando `sudo` (Super User DO) permite a un usuario autorizado ejecutar comandos con privilegios de otro usuario, normalmente el usuario root. Es una herramienta fundamental para la administración de sistemas, ya que facilita la realización de tareas que requieren permisos elevados sin necesidad de cambiar de usuario.

Alcances en seguridad y confiabilidad

- **Seguridad**: `sudo` mejora la seguridad al permitir un control granular sobre quién puede ejecutar qué comandos. Registra todas las actividades realizadas, lo que facilita la auditoría y el seguimiento de acciones administrativas.
- **Confiabilidad**: Reduce la necesidad de compartir la contraseña de root, limitando el acceso a privilegios elevados únicamente a usuarios específicos y comandos determinados.

Ubicación de su configuración

La configuración de `sudo` se encuentra principalmente en el archivo `/etc/sudoers`. Además, se pueden añadir configuraciones específicas en el directorio `/etc/sudoers.d/`.

> En Macos lo ficheros `/etc` realmente estan en `/private/etc/`

Usar `visudo` para la edición (aconsejado)

Editar el archivo `/etc/sudoers` directamente puede ser arriegado, ya que un error de sintaxis puede bloquear el acceso administrativo. Por ello, se recomienda utilizar el comando `visudo`, que verifica la sintaxis antes de aplicar los cambios.

Hacer un backup antes de editar

Antes de modificar el archivo `sudoers`, es prudente realizar una copia de seguridad:

```
sudo cp /etc/sudoers /etc/sudoers.backup_$(date +%Y%m%d)
```

Este comando crea una copia del archivo `sudoers` con la fecha actual, facilitando la restauración en caso de errores.

Editar con `visudo`

Para editar el archivo `sudoers` de manera segura:

```
sudo visudo
```

Este comando abre el archivo en el editor predeterminado configurado para `visudo` (generalmente `nano` o `vi`) y verifica la sintaxis al guardar.

Diferentes posibilidades de configuración

Permitir `sudo` sin pedir contraseña

Para que un usuario pueda ejecutar comandos con `sudo` sin necesidad de ingresar una contraseña, añade la siguiente línea en el archivo `sudoers`:

```
usuario ALL=(ALL) NOPASSWD:ALL
```

Ejemplo:

```
javier ALL=(ALL) NOPASSWD:ALL
```

Permitir `sudo` sin pedir contraseña pero limitado a algunos comandos

Para otorgar permisos de `sudo` sin contraseña pero restringidos a comandos específicos:

```
usuario ALL=(ALL) NOPASSWD:/ruta/al/comando1, /ruta/al/comando2
```

Si tenemos mas d eun comando puede ser mas pratico usar una variable, un fichero de .conf especifico.

```
touch /etc/sudoers.d/mi_usuario_sudo
visudo -f /etc/sudoers.d/mi_usuario_sudo
```

Añade la configuración deseada (es un ejemplo)

```
Cmnd_Alias PRTG = /usr/sbin/csf, /usr/local/directadmin/scripts/letsencrypt.sh, /usr/bin/ls,
/usr/bin/cat, /usr/bin/tail
admin ALL=(ALL) NOPASSWD: PRTG
```

Guadar y cerrar.

Ejemplo:

```
javier ALL=(ALL) NOPASSWD:/usr/bin/systemctl restart nginx, /usr/bin/systemctl status nginx
```

Atención: En sistemas como macOS, una configuración incorrecta que elimina la solicitud de contraseña puede bloquear el acceso administrativo si no existe otro usuario con privilegios de superadministrador.

Usar `/etc/sudoers.d/` para configuraciones específicas de usuarios

En lugar de modificar directamente el archivo `sudoers`, es posible crear archivos individuales para cada usuario en el directorio `/etc/sudoers.d/`. Esto facilita la gestión y evita conflictos.

Creación de un Archivo de Configuración para un Usuario

1. **Crear el Archivo:**

   ```
   sudo nano /etc/sudoers.d/usuario
   ```

2. **Añadir las Reglas de `sudo`:**
   Permitir `sudo` sin contraseña:

```
usuario ALL=(ALL) NOPASSWD:ALL
```

Permitir `sudo` sin contraseña pero limitado a ciertos comandos:

```
usuario ALL=(ALL) NOPASSWD:/usr/bin/systemctl restart nginx, /usr/bin/systemctl
status nginx
```

3. **Guardar y Cerrar el Archivo:**
   Presiona `Ctrl + X`, luego `Y` y `Enter` para guardar los cambios.
4. **Verificar la Sintaxis:**
   `visudo` automáticamente verifica la sintaxis al editar el archivo. Sin embargo, puedes comprobar manualmente ejecutando:

   ```
   sudo visudo -cf /etc/sudoers.d/usuario
   ```

   Este comando validará la configuración e inmformará de cualquier error.

Resumen de comandos clave

```
# Crear una copia de seguridad del archivo sudoers
sudo cp /etc/sudoers /etc/sudoers.backup_$(date +%Y%m%d)

# Editar el archivo sudoers de manera segura
sudo visudo

# Permitir a un usuario ejecutar todos los comandos sin contraseña
usuario ALL=(ALL) NOPASSWD:ALL

# Permitir a un usuario ejecutar comandos específicos sin contraseña
usuario ALL=(ALL) NOPASSWD:/ruta/al/comando1, /ruta/al/comando2

# Crear un archivo de configuración específico para un usuario
sudo nano /etc/sudoers.d/usuario

# Verificar la sintaxis de un archivo en sudoers.d
sudo visudo -cf /etc/sudoers.d/usuario
```

Consideraciones Adicionales

- **Permisos de Archivos:** Asegúrate de que los archivos en `/etc/sudoers.d/` tengan permisos correctos (generalmente 0440) para evitar problemas de seguridad.

  ```
  sudo chmod 0440 /etc/sudoers.d/usuario
  ```

- **Evitar Errores de Sintaxis:** Siempre utiliza `visudo` o editores diseñados para manejar la configuración de `sudo` para prevenir errores que puedan bloquear el acceso administrativo.
- **Uso Responsable de `NOPASSWD`:** Otorgar permisos sin contraseña debe hacerse con cautela, limitando el acceso solo a los comandos estrictamente necesarios para minimizar riesgos de seguridad.
- **Documentación y Auditoría:** Mantén una documentación clara de las configuraciones realizadas y revisa periódicamente los permisos otorgados para asegurar que siguen siendo necesarios y seguros.
- En **MacOS** una configuración erronea del usuario administrador puede ser fatal y muy complicada la recuperación del desastre. Se recomienda por esto y po rmuchas mas cosas, tener siempre un segundo usuario SuperAdmin en un sistema MacOs

Conclusión

El uso adecuado de `sudo` es esencial para la administración segura y eficiente de sistemas Linux. Configurarlo correctamente, utilizando herramientas como `visudo` y aplicando buenas prácticas de seguridad, garantiza que los usuarios puedan realizar tareas administrativas sin comprometer la integridad y seguridad del sistema.

Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de Castris

Si necesitas soporte profesional puedes contratar con Castris soporte profesional.

Revision #5
Created 2025-02-05 07:49:12 UTC by Abkrim
Updated 2025-03-18 06:46:20 UTC by Abkrim