



Élévation des privilèges (su et sudo)

Table des matières

1. Limiter le compte root	2
2. La commande su	3
3. La commande sudo	4
3.1. Avantages	4
4. Commande visudo	6
4.1. Le fichier /etc/sudoers	6
4.2. Le groupe wheel	6
4.3. Ne plus utiliser root	6
4.4. Restreindre le sudo	7
Section Host aliases	7
Section Command aliases	8
Section User Specification	8
Exemple 1 :	9
Exemple 2 :	9

Le compte du super-utilisateur (root) possède de nombreuses contraintes car :

- il a tous les droits ;
- il peut causer des dommages irréparables ;
- il est la cible privilégiée des intrus.

L'administrateur qui travaille sur le serveur, se connecte avec son compte (qui dispose de droits restreints), puis au moment où il doit réaliser une tâche d'administration, endosse l'identité de **root** de façon temporaire, exécute ses actions, puis reprend son identité d'origine.

Il convient donc d'effectuer les opérations suivantes :

- interdire le login root ;
- restreindre l'accès à la commande su ;
- privilégier la commande sudo ;
- robustesse du mot de passe :
 - il doit être complexe ;
 - non issu du dictionnaire ;
 - comporter des minuscules, majuscules, lettres, chiffres et caractères spéciaux ;
- changer régulièrement (ne pas être conservé identique indéfiniment) ;
- sécurisation (ne doit pas être inscrit à proximité des postes ou communiqué à des personnes qui n'ont pas à le connaître).



Ces principes sont les premières règles de sécurité.

Chapitre 1. Limiter le compte root

Le fichier `/etc/securetty` contient la liste des terminaux accessibles par le login **root**.

Pour interdire un terminal au login **root**, il faut :

- soit mettre la ligne en commentaire ;
- soit supprimer la ligne.

En fonction des distributions, la syntaxe du fichier `/etc/securetty` peut changer.

Pour une distribution **CentOS**, le fichier est constitué d'une ligne **tty** et d'une ligne **vc**, indispensable pour la connexion d'un utilisateur.

Chapitre 2. La commande su

su signifie **S**ubstitute **U**ser ou **S**witch **U**ser. Elle permet d'endosser l'identité d'un autre utilisateur sans se déconnecter. Cette commande utilisée sans login permet par défaut de prendre l'identité de **root**.

Syntaxe de la commande su

```
su [-] [-c commande] [login]
```

Exemple :

```
[root]# su - alain
[albert]$ su -c "passwd alain"
```

Option	Description
-	Charge l'environnement complet de l'utilisateur.
-c commande	Exécute la commande sous l'identité de l'utilisateur.

Pour la sécurité, les accès par la commande **su** sont répertoriés dans le fichier journal **/var/log/secure**.

Les utilisateurs non **root** devront taper le mot de passe de la nouvelle identité.



Il y a création de couches successives. Pour passer d'un utilisateur à un autre, il faut d'abord taper la commande **exit** pour reprendre son identité puis la commande **su** pour prendre une autre identité.

Il est conseillé de restreindre l'accès à la commande **su** à certaines personnes définies explicitement. Pour cela, il faut donner les droits à un groupe particulier pour accéder à la commande. Toutes les personnes appartenant à ce groupe obtiendront les droits réservés de celui-ci.



Le principe de restriction mis en place consiste à autoriser seulement un groupe d'utilisateurs particulier à exécuter la commande **su**.

Chapitre 3. La commande sudo

La commande **sudo** permet d'exécuter une ou plusieurs commandes avec les privilèges de root. Il n'est pas nécessaire de connaître son mot de passe.

Syntaxe de la commande sudo

```
sudo [-options] commande
```

Cet outil représente une évolution de la commande su.

Exemple :

```
[alain]$ sudo /sbin/route
Mot de passe : *****
Table de routage IP du noyau
Destination  Passerelle  Genmask      Indic Metric Ref Use Iface .....
```

Option	Description
-E	Garde l'environnement du compte appelant.
-u	Désigne le compte qui exécutera la commande.



L'environnement est constitué des paramètres de l'utilisateur comme son répertoire de connexion, ses alias ...

Les accès possibles à la commande et les possibilités offertes aux utilisateurs sont configurables par l'administrateur qui leur affecte des droits explicites.

Vérification de l'existence du paquet :

```
[root] # rpm -qa "sudo*"
sudo-1.8.6p3-12.el6.i686
```

3.1. Avantages

Par rapport à l'utilisation des restrictions des groupes :

- Le contrôle d'accès aux commandes est centralisé dans le fichier **/etc/sudoers** ;
- Tous les accès et tentatives à partir de sudo sont journalisés dans le fichier **/var/log/secure**;
- L'utilisateur doit s'identifier à sudo en donnant son mot de passe.

Le fichier **/etc/sudoers** contient pour chacun des utilisateurs :

- son login ;
- sur quel ordinateur il est autorisé ;
- quelles commandes il peut exécuter.

Chapitre 4. Commande visudo

La commande visudo permet d'éditer le fichier `/etc/sudoers` (qui est en lecture seule) afin de configurer efficacement l'accès à la commande sudo pour les utilisateurs.

Syntaxe de la commande visudo

```
visudo [-c] [-f sudoers]
```

Exemple :

```
[root]# visudo
```

Option	Description
-c	Vérifie la syntaxe du fichier.
-f file	Désigne le fichier de configuration de sudo.

4.1. Le fichier `/etc/sudoers`

```
# This file must be edited with the
# 'visudo' command.
#
# Host alias specification
# User alias specification
# Cmnd alias specification
# User privilege specification
root ALL=(ALL) ALL
%wheel ALL=(ALL) ALL
```

4.2. Le groupe wheel

Le mot **wheel** fait référence à un groupe système qui dispose de privilèges lui permettant l'exécution de commandes à accès restreint.

Le groupe **wheel** est présent par défaut sur les distributions RHEL/CentOS.

4.3. Ne plus utiliser root

1 - Autoriser les utilisateurs du groupe **wheel** à lancer toutes les commandes (via visudo) en supprimant le commentaire de la ligne suivante :


```
%wheel    ALL=(ALL)    ALL
```

2 - Ajouter le compte utilisateur **bob** dans le groupe **wheel** :

```
[root]# usermod -aG wheel bob
```

3 - L'utilisateur **bob** dispose maintenant des pleins pouvoirs :

```
[bob]$ sudo chown root:root /tmp/test  
[sudo] password for bob:
```



Il n'existe plus de raisons valables d'utiliser le compte root.

Le mot de passe **root** peut être séquestré (Keepass !) et son accès via ssh verrouillé.

4.4. Restreindre le sudo

Il est possible de ne pas donner tous les droits à un utilisateur mais de limiter ses accès et les commandes qu'il peut lancer. Par exemple, n'offrir à un compte de supervision que le droit de relancer le serveur (reboot) et uniquement celui-là.

Les accès à l'utilisation de sudo sont enregistrés dans les logs ce qui offre une traçabilité des actions entreprises.

Les restrictions se gèrent dans les différentes sections du fichier /etc/sudoers par l'intermédiaire de la commande visudo.

Section Host aliases

Définir des groupes de machines ou réseaux.

```
Host_Alias HOSTS-GROUP = host1 [,host2 ...]
```

Exemple :

```
Host_Alias FILESERVERS = 192.168.1.1, SF1, SF2
```

Cette section est utilisée pour créer des groupes de ressources réseaux autorisés à utiliser sudo.

Section User aliases

Définir des alias pour les utilisateurs.

```
User_Alias USER-GROUP = alain [,philippe, ...]
```

Exemple :

```
User_Alias ADMINS = root, AdminSF, adminPrinters
```

Cette section est utilisée essentiellement pour réunir sous un alias unique plusieurs utilisateurs ayant les mêmes besoins de la commande sudo.

Section Command aliases

Définir des alias pour les commandes.

```
Cmnd_Alias CDE-GROUP = cde1 [,cde2, ...]
```

Exemple :

```
Cmnd_Alias SOFTWARE = /bin/rpm, /usr/bin/yum  
Cmnd_Alias SERVICES = /sbin/service, /sbin/chkconfig
```

Cette section est utilisée pour réunir plusieurs commandes Linux dans un groupe de commandes sudo.

Il faut alors créer des groupes cohérents.

Section User Specification

Lier les utilisateurs aux commandes.

```
USER-GROUP HOSTS-GROUP = [(cpte-cible)] CDE-GROUP
```

Exemple :

```
root      ALL=(ALL)      ALL  
AdminSF   FILESERVERS=   SOFTWARE
```

Cette section définit qui a le droit d'utiliser des commandes particulières à partir de postes

particuliers.

Il est possible de préciser qui exécute la commande (compte cible).

Exemple 1 :

Grâce au fichier `/etc/sudoers` ci-dessous, les utilisateurs `alain`, `patrick` et `philippe` peuvent désormais exécuter les commandes **ping** et **route** et effectuer des transferts FTP comme s'ils étaient **root**.

```
# Host alias specification
Host_Alias STA = 192.168.1.1, ma.machine
# User alias specification
User_Alias CPTUSER = alain, patrick, philippe
# Cmnd alias specification
Cmnd_Alias NET = /bin/ping, /sbin/route, /usr/bin/ftp
# User privilege specification
root ALL=(ALL) ALL
%wheel ALL=(ALL) ALL
CPTUSER STA=(root) NET
```

Exemple 2 :

```
# Host alias specification
Host_Alias MACHINE = station1
# User alias specification
User_Alias ADMIN = adminunix
User_Alias UTILISAT = alain, philippe
# Cmnd alias specification
Cmnd_Alias SHUTDOWN = /sbin/shutdown
Cmnd_Alias NET = /usr/bin/ftp
# User privilege specification
root ALL=(ALL) ALL
%wheel ALL=(ALL) ALL
ADMIN MACHINE=NOPASSWD:ALL
UTILISAT MACHINE=(root) SHUTDOWN, NET
```

Explications de l'exemple 2 :

- Création d'un alias pour la station :

```
Host_Alias MACHINE=station1
```

- Création de deux alias pour deux types d'utilisateurs (adminunix étant un équivalent « root »).

```
User_Alias ADMIN = adminunix  
User_Alias UTILISAT = alain, philippe
```

- Création de deux alias de commandes qui regroupent les commandes exécutables.

```
Cmnd_Alias SHUTDOWN = /sbin/shutdown  
Cmnd_Alias NET = /usr/bin/ftp
```

- L'utilisateur « root » peut exécuter toutes les commandes sur toutes les machines

```
root ALL=(ALL) ALL
```

Les utilisateurs qui font partie de l'alias ADMIN peuvent exécuter toutes les commandes, sur toutes les machines faisant partie de l'alias MACHINE et ce sans entrer de mot de passe (NOPASSWD:).

```
ADMIN MACHINE=NOPASSWD:ALL
```

Les utilisateurs qui font partie de l'alias UTILISAT peuvent exécuter la commande /sbin/shutdown sur toutes les machines faisant partie de l'alias MACHINE. Ils doivent entrer leur mot de passe.

```
UTILISAT MACHINE = SHUTDOWN, NET
```