

La sécurité des fichiers

Objectifs de cette section

Comprendre comment fonctionnent les **permissions** sur les fichiers et les répertoires.

Partager des données entre groupes et entre utilisateurs de manière sécurisée.

Apprendre ce que sont les **modes spéciaux de permissions**, et comment les utiliser.

Comment rendre un **fichier non modifiable**, y compris par l'utilisateur Root.

Utilisation des **Access Control Lists (ACLs)**.

Les **Rootkits**, comment les découvrir et les retirer de votre système.

Le système de permissions

Les permissions

Grâce à la commande **ls -l** nous pouvons voir les permissions associées aux fichiers et répertoires.

J'attire votre attention sur le premier caractère de chaque ligne, il spécifie le type de fichiers dont il s'agit :

- - : Fichier classique
- d : Répertoire
- l : Lien symbolique

```
root@debian:/home/jordan# ls -l
total 32
drwxr-xr-x 2 jordan jordan 4096 May 30 16:00 Desktop
drwxr-xr-x 2 jordan jordan 4096 May 30 16:00 Documents
-rw-r--r-- 1 root    root      0 Jun 28 10:09 jesuisunfichier
drwxr-xr-x 2 jordan jordan 4096 May 30 16:00 Music
drwxr-xr-x 2 jordan jordan 4096 May 30 16:00 Public
```

Les permissions (2)

Grâce à la commande **ls -l** nous pouvons voir les permissions associées aux fichiers et répertoires.

Les autres caractères font références aux permissions associées aux fichiers :

- **r** : Read (Lecture)
- **w** : Write (Ecriture)
- **x** : Execute (Exécution)

```
root@debian:/home/jordan# ls -l
total 32
drwxr-xr-x 2 jordan jordan 4096 May 30 16:00 Desktop
drwxr-xr-x 2 jordan jordan 4096 May 30 16:00 Documents
-rw-r--r-- 1 root    root      0 Jun 28 10:09 jesuisunfichier
drwxr-xr-x 2 jordan jordan 4096 May 30 16:00 Music
drwxr-xr-x 2 jordan jordan 4096 May 30 16:00 Public
```

Signification des permissions entre fichiers et dossiers

Permission	Fichier	Dossier
Read (r)	Autorise l'utilisateur à lire le contenu du fichier.	Autorise l'utilisateur à lire le nom des fichiers contenus dans le répertoire.
Write (w)	Autorise l'utilisateur à modifier le contenu du fichier.	Autorise l'utilisateur à ajouter, supprimer ou modifier les fichiers contenus dans le répertoire.
Execute (x)	Autorise l'utilisateur à exécuter le fichier.	Autorise l'accès au contenu et aux métadonnées pour les fichiers contenus dans le répertoire.

Les différentes catégories pour les permissions

Lorsque vous exécutez un `ls -l` les symboles w,r, et x sont répétés trois fois.

En effet, ils correspondent aux permissions pour chaque catégorie :

- u : User (Utilisateur) propriétaire du fichier
- g : Group (Groupe) associé au fichier
- o : Other (Autres)
- a : All (tous)

Les groupes

Tous les utilisateurs sont rattachés à un groupe au minimum.

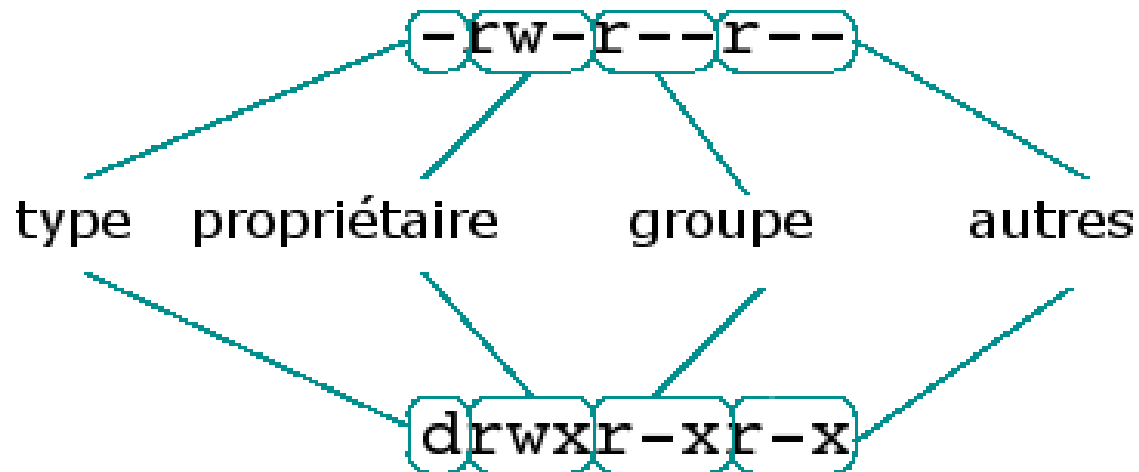
Ils peuvent également appartenir à plusieurs groupes.

Les groupes sont utilisés pour l'organisation et la gestion des utilisateurs en les regroupant par service, par type de permissions sur le système, ou même par zone géographique.

La commande **groups** permet d'afficher le, ou les groupes auxquels appartient l'utilisateur.

- On peut également utiliser la commande **id -Gn**.

Décoder les permissions associées



Le système de permissions

MODIFIER LES PERMISSIONS

Modifier les permissions

Vous pouvez modifier les permissions d'un fichier grâce à la commande **chmod** et aux options suivantes :

- u : modifier les permissions correspondantes à l'utilisateur propriétaire
- g : modifier les permissions correspondantes au groupe
- o : modifier les permissions pour les autres utilisateurs
- a : modifier les permissions pour toutes les catégories

On précise ensuite si on souhaite ajouter (+) ou enlever une permission (-), puis on ajoute la permission à modifier :

- r : read
- w : write
- x : execute

Modifier les permissions avec les caractères numériques

On peut également utiliser la notation numérique pour modifier les permissions associées à un fichier.

Les valeurs associées sont :

- r : 4
- w : 2
- x : 1

Donc pour paramétrer un droit de type :

- rwx pour l'utilisateur, rw pour le groupe et wx pour les autres
- chmod 763

Modifier les permissions avec les caractères numériques (2)

Quelques exemples supplémentaires :

Symbolic	Octal
<code>-rwx-----</code>	700
<code>-rwxr-xr-x</code>	755
<code>-rw-rw-r--</code>	664
<code>-rw-rw----</code>	660
<code>-rw-r--r--</code>	644

Modifier les groupes associés aux fichiers

Lorsque vous créez un fichier, celui-ci est automatiquement paramétré pour appartenir à l'utilisateur qui l'a créé, et au groupe primaire auquel appartient l'utilisateur.

Pour modifier le groupe associé au fichier, il suffit d'utiliser la commande **chgrp**.

Les permissions sur les répertoires

Les permissions fixées sur un répertoire peuvent affecter l'ensemble des fichiers situés à l'intérieur.

Si les permissions d'un fichier semblent correctes, alors allez toujours vérifier les permissions du répertoire qui contient ce fichier.

N'hésitez pas à remonter l'arborescence des fichiers jusqu'à la racine.

Les modes spéciaux

Le setuid

Lorsqu'un processus est démarré, il se lance en utilisant les permissions de l'utilisateur et du groupe de celui qui l'a démarré.

Il existe une permission spéciale que l'on peut utiliser appelée SetUID pour **Set User ID** qui force le processus à démarrer en utilisant **l'utilisateur propriétaire** et non celui qui l'a exécuté.

Le setuid (2)

Pour vérifier si ce mode est activé, il suffit de regarder la sortie standard du **ls -l** et de voir si un **s** figure dans la colonne concernant le propriétaire :

```
jordan@debian:~$ ls -l /usr/bin/passwd  
-rwsr-xr-x 1 root root 59680 May 17 2017 /usr/bin/passwd
```

Nous avons également les processus suivants :

- ping (besoin du root pour accéder aux services réseaux)
- chsh (permet à l'utilisateur de changer son shell en editant le /etc/passwd)

Attention, car les fichiers avec le setuid sont cibles d'attaques !

Impossible d'utiliser cette fonctionnalité sur les scripts (et heureusement !)

Les valeurs octales des modes spéciaux

On peut également utiliser la notation numérique pour modifier les modes spéciaux associées à un fichier.

Les valeurs associées sont :

- setuid : 4
- setgid : 2
- sticky : 1

Ajouter ou retirer l'attribut Setuid

Pour activer le setuid, on utilise également la commande **chmod** sur le fichier souhaité.

On a alors le choix entre la notation octale ou par texte pour ajouter le setuid :

- `chmod u+s CHEMIN_DU_FICHER`
- `chmod 4XXX CHEMIN_DU_FICHER`

On a alors le choix entre la notation octale ou par texte pour retirer le setuid :

- `chmod u-s CHEMIN_DU_FICHER`
- `chmod 0XXX CHEMIN_DU_FICHER`

Découvrir les fichiers avec le setuid

Il existe une commande permettant de découvrir les fichiers qui ont le setuid de paramétré grâce à la commande : **find / -perm /4000**

```
root@debian:/home/jordan# find / -perm /4000
/usr/bin/sudo
/usr/bin/newgrp
/usr/bin/gpasswd
/usr/bin/passwd
/usr/bin/chsh
[...]
```

Bonnes pratiques

Seul le propriétaire du fichier doit pouvoir éditer les fichiers où le Setuid a été paramétré.

Par exemple voici les bonnes permissions :

- `-rwsr-xr-x -> 4755`
- `-rwsr--r-- -> 4744`

Typiquement voici les permissions qui peuvent s'avérer dangereuses !

- `-rwsrwxr-x -> 4775`
- `-rwsrwxrwx -> 4777`

Le setgid

Le setgid (**Set Group ID**) est très proche au niveau du fonctionnement que le setuid.

Il implique le fonctionnement du processus avec les privilèges de groupe du fichier plutôt que les privilèges du groupe de l'utilisateur qui a lancé le dît processus.

```
root@debian:/home/jordan# ls -l /usr/bin/wall
```

```
-rwxr-sr-x 1 root tty 27448 Mar  7 18:29 /usr/bin/wall
```

- La commande wall permet d'afficher un message dans le terminal pour les utilisateurs qui sont logués sur le système.

Découvrir les fichiers avec le setgid

Il existe une commande permettant de découvrir les fichiers qui ont le setgid de paramétré grâce à la commande : **find / -perm /2000**

```
root@debian:/home/jordan# find / -perm /2000
/usr/bin/wall
/usr/bin/crontab
/usr/local
/usr/local/bin
/usr/local/sbin
/usr/local/include
/usr/local/lib/python2.7
[...]
```

Ajouter ou retirer l'attribut Setgid

Pour activer le setgid, on utilise également la commande **chmod** sur le fichier souhaité.

On a alors le choix entre la notation octale ou par texte pour ajouter le setuid :

- `chmod g+s CHEMIN_DU_FICHER`
- `chmod 2XXX CHEMIN_DU_FICHER`

On a alors le choix entre la notation octale ou par texte pour retirer le setuid :

- `chmod g-s CHEMIN_DU_FICHER`
- `chmod 0XXX CHEMIN_DU_FICHER`

Le Sticky Bit

Le sticky bit est utilisé sur un fichier/répertoire pour n'autoriser la suppression que par le propriétaire de ce fichier/répertoire (ou le root).

Surtout utilisé dans le cas de permissions de type 777.

Il est notamment utilisé sur le /tmp :

```
root@debian:/home/jordan# ls -ld /tmp
```

```
drwxrwxrwt 12 root root 4096 Jun 28 16:17 /tmp
```

Ajouter ou retirer l'attribut Sticky Bit

Pour activer le Sticky bit, on utilise également la commande **chmod** sur le fichier/répertoire souhaité.

On a alors le choix entre la notation octale ou par texte pour ajouter le Sticky Bit :

- `chmod o+t CHEMIN_DU_FICHER`
- `chmod 1XXX CHEMIN_DU_FICHER`

On a alors le choix entre la notation octale ou par texte pour retirer le Sticky Bit :

- `chmod o-t CHEMIN_DU_FICHER`
- `chmod 0XXX CHEMIN_DU_FICHER`

Les ACLs

LES ACCESS CONTROL LISTS

Les ACLs

ACL signifie Access Control List

Fournit des possibilités de contrôles additionnels du système de permission.

- Par exemple, n'autoriser l'accès à un fichier qu'à un utilisateur.

Pour pouvoir utiliser les ACLs, il faut s'assurer que le système de fichiers a été monté avec le support de cette fonctionnalité :

- `mount -o acl`
- `tune2fs -o acl`

Les différents types d'ACLs

Les ACLs de type **Access** permettent de gérer les permissions sur un fichier ou un répertoire en particulier.

Les ACLs de type **Default** :

- Utilisées sur les répertoires uniquement.
- Affectent les fichiers sans Access ACLs à l'intérieur de ce répertoire.

La configuration des ACLs

Les ACLs peuvent être configurées :

- Par utilisateur
- Par groupe
- Pour les utilisateurs qui ne sont pas dans le groupe associé au fichier

On utilise la commande **setfacl** pour créer, modifier et supprimer les ACLs sur les fichiers et les répertoires.

- Ps : Il est parfois nécessaire d'installer les outils d'ACLs si la commande n'est pas disponible sur votre système.

Configuration des règles pour les ACLs sur les utilisateurs

Voici la syntaxe pour configurer une ACL sur un utilisateur :

- `setfacl -m u:uid:permissions nom_du_fichier`

Par exemple pour donner les droits de lecture, écriture et exécution pour l'utilisateur jordan sur le fichier test.sh :

- `setfacl -m u:jordan:rwX test.sh`

Configuration des règles pour les ACLs sur les groupes

Voici la syntaxe pour configurer une ACL sur un groupe :

- `setfacl -m g:gid:permissions nom_du_fichier`

Par exemple pour donner les droits de lecture, écriture et exécution pour les utilisateurs du groupe IT sur le fichier test.sh :

- `setfacl -m g:IT:rwx test.sh`

Configuration des règles pour les ACLs sur les autres utilisateurs

Voici la syntaxe pour configurer une ACL sur les autres utilisateurs:

- `setfacl -m o:permissions nom_du_fichier`

Par exemple pour donner les droits de lecture pour les autres utilisateurs sur le fichier test.sh :

- `setfacl -m o:r test.sh`

Configuration des règles multiples pour les ACLs

On peut configurer plusieurs règles sur une ACL en séparant les éléments par une virgule.

Par exemple pour donner les droits :

- Lecture, écriture et exécution pour l'utilisateur Jordan
 - Lecture, exécution pour les membres du groupe IT
 - Lecture pour tous les autres
- ```
setfacl -m u:jordan:rwx,g:IT:rx,o:r test.sh
```

# Configuration des ACLs default

---

Si on souhaite que tous les fichiers qui vont être créés dans le répertoire « informatique » soient disponibles pour la lecture et l'écriture par les membres du groupe « IT » nous créons l'ACL default suivante :

- `setfacl -m d:g:IT:rw informatique`
- Ainsi tous les fichiers du répertoire « informatique » seront configurés automatiquement avec une ACL access correspondante aux règles fixées ci-dessus.

# Suppression des ACLs

---

Pour supprimer une ACL on utilise l'option `-x` :

- `setfacl -x ACL FICHIER_OU_REPERTOIRE`

Par exemple :

- `setfacl -x u:jordan test.sh`

# Voir les ACLs configurées

---

Pour voir les ACLs qui ont été configurées sur un fichier ou un repertoire, on utilise la commande **getfacl** :

```
root@debian:/home/jordan# getfacl test.sh
file: test.sh
owner: root
group: root
user::rw-
user:jordan:rwx
group::r--
mask::rwx
other::r--
```