GNU - Linux // Gestion des droits & Permissions

L'Anatomie de la Sécurité (Lecture des droits)

Avant de modifier quoi que ce soit, il faut comprendre ce que vous regardez. Sous Linux, tout est fichier, et chaque fichier a un propriétaire et un groupe.

Pour auditer la situation, nous utilisons la commande sacrée :

```
ls -l
```

Structure d'une ligne de droits

```
ls -l *
```

```
-rw-rw-r-- 2 user users   12 Jan 12 11:29 aliasFichier.txt
lrwxrwxrwx 1 user users   23 Jan 12 11:30 symbolicFichier.txt -> dossierTest/fichier.txt

dossierTest:
total 4
-rw-rw-r-- 2 user users 12 Jan 12 11:29 fichier.txt
```

Segment	**Signification**
Type (1er char)	- (Fichier), d (Répertoire), l (Lien)
User (u)	rwx (Droits du propriétaire)
Group (g)	r-x (Droits du groupe)
Others (o)	r-x (Droits des autres/le reste du monde)

La Trinité des Permissions (Syntaxe)

Il existe trois niveaux d'accréditation. C'est binaire, froid et logique.

Permission	Lettre	Valeur Octale	Effet sur Fichier	Effet sur Répertoire
Lecture	r	4	Afficher le contenu (cat)	Lister les fichiers (ls)
Écriture	w	2	Modifier/Sauvegarder	**Créer/Supprimer** des fichiers
Exécution	x	1	Exécuter (Script/Prog)	**Traverser/Entrer** (cd)

> **Note :** L'addition est simple. Lecture (4) + Ecriture (2) + Exécution (1) = 7 . Si vous mettez 777, vous donnez les clés de Poudlard à Voldemort. C'est une **faille de sécurité critique**.

Commandes d'Intervention (L'Arsenal)

Voici les sortilèges pour modifier la réalité.

CHMOD (Change Mode) - Modifier les droits

C'est la commande pour définir qui peut faire quoi.

Méthode Octale (La plus rapide)

```
chmod 750 fichier
```

- 7 (User) : 4+2+1 (Tout permis).
- 5 (Group) : 4+0+1 (Lecture + Exécution).
- 0 (Others) : Rien. "Vous n'avez pas dit le mot magique".

Méthode Symbolique (La plus chirurgicale)

- u (user), g (group), o (other), a (all).
- + (ajouter), - (retirer), = (fixer).

Donner l’exécution au groupe (autorise la traversée du dossier)

```
chmod g+x script.sh
```

Retirer l’écriture aux autres

```
chmod o-w fichier
```

Full access, for ALL (DANGER)

```
chmod 777 fichier
```

CHOWN & CHGRP - Changer la propriété

Parce que parfois, il faut rendre à César ce qui appartient à Root.

Changer le propriétaire

```
chown user fichier
```

Changer le groupe

```
chgrp group fichier
```

Le Combo (Propriétaire + Groupe)

```
chown user:group fichier
```

Récursif (Tout le dossier)

```
chown -R user dossier
```

> **Avertissement de Sécurité (Winston Wolf Protocol) :** Lancer un chown -R ou un chmod -R sur la racine / ou des dossiers systèmes (/bin, /etc) n'est pas une erreur, c'est un suicide numérique. Le système ne redémarrera pas. Soyez précis.

Les Droits Spéciaux (Magie Noire)

Parfois, les règles standard ne suffisent pas. On entre dans la zone des droits étendus.

SUID (Set User ID) - Bit 4 : Le fichier s'exécute avec les droits du *propriétaire* (souvent root), peu importe qui le lance.

- *Indicateur :* un s à la place du x utilisateur (-rwsr-xr-x).
- *Risque :* Élevé. Si le programme est buggé, l'utilisateur devient root.

SGID (Set Group ID) - Bit 2 : Le fichier s'exécute avec les droits du *groupe*.

- *Indicateur :* un s à la place du x groupe.

Sticky Bit - Bit 1 : Sur un dossier, empêche un utilisateur de supprimer un fichier qu'il n'a pas créé (ex: /tmp).

- *Indicateur :* un t ou T à la fin (drwxrwxrwt).

Best Practices & Audit (Procédure de Survie)

- **Le Principe du Moindre Privilège :** Ne donnez jamais plus de droits que nécessaire. Si r-- (4) suffit, ne mettez pas rw- (6).
- **L'UMASK :** C'est le filtre par défaut à la création d'un fichier. Par défaut 022, ce qui retire les droits d'écriture au groupe et aux autres.
- **Vérification :** Si un utilisateur se plaint d'un "Permission Denied" (Accès refusé), vérifiez d'abord ses groupes avec la commande groups ou id et les droits du dossier parent (il faut x pour traverser un dossier !).