Permissions: Qui contrôle quoi?

Objectif

Comprendre ce qui suit:

Le modèle de sécurité Unix

Comment un programme est autorisé à s'exécuter

- Ou sont stocké les informations des utilisateurs et des groupes
- Détails des permissions sur les fichiers

Utilisateurs et groupes

Unix comprend des utilisateurs et groupes

- Un utilisateur peut appartenir à plusieurs groupes
- Un fichier peut appartenir qu'à un seul utilisateur et groupe à un moment
- Un utilisateur particulier, le super-utilisateur "*root*" a des privilèges supplémentaires (uid = "0"dans / etc / passwd)
- Seul root peut changer le propriétaire d'un fichier

Utilisateurs et groupes

Les informations sur l'utilisateur se trouve dans: `/etc/passwd`

Information sur le mot de passe se trouve dans `/etc/master.passwd`

Informations sur le groupe se trouve dans `/etc/group`

`/etc/passwd` et `/etc/groupe` les donnés sont separées en utilisant ":"

/ etc / passwd:

```
auser: *: 1000:1000: A. Utilisateur: /home/user: /usr/local/bin/bash
```

/ etc / group:

```
utilisateurs: *: 99: utilisateur
```

Un programme tourne ...

Un programme peut être exécuté par un utilisateur, lorsque le système démarre ou par un autre processus.

Avant que le programme ne s'exécute le noyau examine plusieurs choses:

- Le fichier contient-il un programme accessible à l'utilisateur ou le groupe du processus qui veut l'exécuter?
- Le fichier contenant le programme permet-il l'exécution par cet utilisateur ou groupe (ou quelqu'un)?
- Dans la plupart des cas, lors de l'exécution, un programme hérite des privilèges de l'utilisateur ou processus qui l'a démarré.

Un programme en détail

Quand on tape:

```
ls -l /usr/bin/top
```

Nous allons voir:

```
-r-xr-xr-x 1 root wheel 46424 21 novembre 2009 28
/usr/bin/top
```

Que signifie tout cela?

```
-r-xr-xr-x  1   root     wheel    46424   Nov 21 2009  /usr/bin/top

---------- --- ------- ------- -------- ------------ -------------
  |    |   |    |      |      |        |
  |    |   |    |      |      |      File Name
  |    |   |    |      |      |
  |    |   |    |      |      +---  Modification Time/Date
  |    |   |    |      |
  |    |   |    |      +------------   Size (in bytes
  |    |   |    |
  |    |   |    +----------------------       Group

  |    |   |
  |    |   +-----------------------------       Owner
  |    |
  |    +-----------------------------------  "link count"
  |
  +-------------------------------------------  File Permissions
```

UNIVERSITY OF OREGON

Les droits d'accès

Les fichiers sont détenus par un utilisateur et un groupe (la propriété)

Les fichiers ont des autorisations pour l'utilisateur, le groupe, et les autres.

"autres » est souvent désignée comme «le monde»

Les autorisations sont en lecture, écriture et exécution (R, W, X)
Le propriétaire d'un fichier est toujours autorisé à changer les permissions

Quelques cas particuliers

Lorsque l'on regarde la sortie d'une commande `ls -l` ;vous avez dans la 1ere colonne

```
d = répertoire
- = fichier régulier
l = lien symbolique
s = socket de domaine Unix
p = pipe nommé
c = fichier de périphérique caractère
b = fichier de périphérique bloc
```

Quelques cas particuliers …suite

Dans la colonne Propriétaire, Groupe et autres, vous pourriez voir:

s = setuid [dans la colonne Propriétaire]

s =setgid [dans la colonne dans le Groupe]

t = sticky bit [à la fin]

Quelques références

http://www.tuxfiles.org/linuxhelp/filepermissions.html
http://www.cs.uregina.ca/Links/class-info/330/Linux/linux.html
http://www.onlamp.com/pub/a/bsd/2000/09/06/FreeBSD_Basics.html

Les permissions sur les fichiers

Il y a deux façons de définir des autorisations par l'utilisation de la commande `chmod`:
Mode symbolique:

testfile dispose des autorisations de **-r--r--r--**

U G O *

```
$ chmod g + x testfile ==>-r--r-xr--
$ Chmod u + wx testfile ==>-rwxr-xr -
$ chmod ug-x testfile ==>-rw--r--r--
U = utilisateur, G = groupe, O = autre
(monde)
```

Permissions sur les fichiers cont.

Le mode absolu:

Nous utilisons octal (base huit) les valeurs sont représentées comme ceci:

Lettre	Permission	Valeur
R	lire	4
W	écrire	2
X	exécuter	1
-	aucune	0

Pour chaque colonne, utilisateur, groupe ou autres vous pouvez définir des valeurs de 0 à 7. Voici ce que ca veut dire:

0 = **---** 1 = **--x** 2 =**-w-** 3 =**-wx**

4 = **r--** 5 = **r-x** 6 = **rw-** 7 = **rwx**

Permissions sur les fichiers cont.

Mode numérique:

Exemple avec un fichier index.html avec des valeurs typiques d'autorisation:

```
$ Chmod 755 index.html
$ ls-l Index.html
  -rwxr-xr-x 1 root wheel 0 24 mai 06:20 index.html

$ Chmod 644 index.html
$ ls-l Index.html
  -rw-r--r-- 1 root wheel 0 24 mai 06:20 index.html
```

Les autorisations héritées

Deux points critiques:

Les autorisations d'un répertoire indiquent si quelqu'un peut voir son contenu, ou ajouter ou supprimer des fichiers.

Les permissions sur un fichier déterminent ce que vous pouvez faire des données du fichier.

Exemple:

Si vous n'avez pas l'autorisation d'écriture sur un répertoire, alors vous ne pouvez pas supprimer un fichier dans le répertoire. Si vous avez accès en écriture au fichier vous pouvez mettre à jour les données dans le fichier.

Conclusion

Pour renforcer ces concepts nous allons faire quelques exercices.

En outre, une référence très agréable sur l'utilisation de la commande `chmod` est dans le document:

An Introduction to Unix Permissions -- Part Two par Dru Lavigne

http://www.onlamp.com/pub/a/bsd/2000/09/13/FreeBSD_Basics.html