Les utilisateurs et les droits

Les utilisateurs et les droits

Plan de la phase

Introduction
La sécurité multi-utilisateur
La gestion des utilisateurs
Les droits
Exemples

Installation d'Ubuntu

Introduction

Le système GNU/Linux étant multi-utlisateur, les personnes employant celui-ci doivent être identifiées afin d'assurer la confidentialité des informations contenue dans les fichiers. En effet, il ne serait pas acceptable que l'utilisateur « Nicolas » puisse consulter les fichiers personnels de « Stéphane » sans son accord.

Ces personnes possèderont donc un « compte utilisateur » sur le système pour l'utiliser en étant clairement identifiées.

Cependant, il est heureusement permis de partager des fichiers entre collaborateurs et une notion de « groupe d'utilisateurs » existe sous GNU/Linux.

Les utilisateurs et les droits

La sécurité multi-utilisateur

Sur un système GNU/Linux, une application accède au fichiers avec des restrictions. Par exemple, un serveur Apache ne peut transmettre une page Web que s'il a accès en lecture aux fichiers correspondant. Cette approche, appelée « sécurité multi-utilisateur » repose sur les concepts :

- L'existence d'une base de comptes utilisateurs et d'une base de comptes groupes d'utilisateurs.
- Le fait qu'un fichier possède des droits précisant les utilisateurs et les groupes qui sont habilités à y accéder.
- Une application en cours d'exécution est associée à un compte utilisateur et à des comptes groupes, ce qui détermine ses droits d'accès aux fichiers.
- Les services sont associés à des comptes grâce à leurs fichiers de configuration.
- La connexion d'un utilisateur, le « login », qui détermine les droits de son shell, et par héritage, les droits de toutes les application qu'il activera par la suite.
- L'administrateur (*root*) à tous les droits sur le système. Il peut créer des comptes et accéder à l'ensemble des fichiers sans restriction. Il peut déléguer une partie de ses prérogatives à certains utilisateurs

Les utilisateurs et les droits

La gestion des utilisateurs

Le concept de compte

Chaque utilisateur d'un système GNU/Linux est inscrit dans une base de données locale ou dans un annuaire réseau. Un compte utilisateurreprésente aussi bien une personne (Pierre, Paul, …) qu'une application (Apache, Postfix, ...).

Caractéristique d'un compte utilisateur

- Login : c'est le nom de l'utilisateur (ou de l'application)
- Mot de passe : utiliser lors de la connexion pour authentifier l'utilisateur.
- UID : numéro qui identifie l'utilisateur (« User IDentification »)
- GID : numéro qui spécifie le groupe principal de l'utilisateur (« Group IDentification »)
- Commentaire
- Répertoire de connexion
- Shell, ce logiciel, le plus souvent un véritable shell, est activé en début de session en mode texte.

L'UID 0 est réservé. Toute application ayant cet UID a tous les droits sur le système. C'est L'UID de l'administrateur root.

Caractéristique d'un compte groupe

- Le nom du groupe.
- GID, ce numéro identifie.
- Un mot de passe. Cette valeur n'est jamais renseignée.
- La liste des membres en tant que membres secondaires, ce qui exclut les comptes dont c'est le groupe principal.

Les utilisateurs et les droits

La gestion des utilisateurs

La gestion des comptes

La gestion des comptes (création...) est une prérogative de l'administrateur (root).

Les fichiers

/etc/nsswitch.conf

Ce fichier indique dans quels annuaires locaux ou réseaux sont recherchés les comptes.

/etc/passwd

Ce fichier contient la base local des comptes groupes.

/etc/shadow

Ce fichier contient les mots de passe locaux et leur durée de vie

Les commandes

- `useradd, usermod, userdel` : Ajout, modification, destruction d'un compte utilisateur.

`groupadd, groupmod, groupdel` : Ajout, modification, destruction du groupe.
`passwd` : Modifie le mot de passe d'un compte.
`id` : Affiche les identiés d'un compte.
`chsh ,chfn` :Modifie le shell, le commentaire d'un compte utilisateur.
`getent` : affiche les données d'un annuaire (passwd, group, shadow).
`pwck, grpck` : Vérifie la syntaxe des fichiers *passwd* et *group*.

Les utilisateurs et les droits

La gestion des utilisateurs

La structure des fichiers passwd et group

```
simoko:x:1000:1000:si moko,12,00 01 02 03 04 05,05 04 03 02 01 00:/home/simoko:/bin/bash
```

Chaque ligne du fichier `passwd` décrit un utilisateur. Les champs sont séparés par deux point (« : »). Cette ligne décrit l'utilisateur *simoko*, son uid est *1000*, son gid est *1000*, le commentaire contient *00 01 02 03 04 05, 05 04 03 02 01*, son répertoire de connexion est `/home/simoko` et son shell est `/bin/bash`.

Chaque ligne du fichier `group` décrit un groupe.
La dernière ligne décrit le groupe de nom « smbshare », son *GID* est *122* et *simoko* et un membre de ce groupe.

```
simoko@karmic:~$ tail -3 /etc/group
pulse-rt:x:121:
simoko:x:1000:
sambashare:x:122:simoko
simoko@karmic:~$
```

La commande useradd

La commande useradd permet de créer un compte utilisateur. Ses principales caractéristiques peuvent être précisées.

```
useradd - u 1001 -g lp -G news,mail -c vampire -d /usr/dracula -m -s /bin/bash dracula
```

La commande précédente crée le compte utilisateur *dracula*. Son *uid* est égale à *1001*, son groupe principale (son *gid*) est *lp*, il fait partie également des groupes *news* et *mail* (en tant que groupes secondaires). Le champs commentaire contient la chaîne *vampire*. Son répertoire de connexion est */usr/dracula*. Ce répertoire sera crée (option -m). Son shell est le *shell bash*

Les utilisateurs et les droits

Les droits

Les catégories d'utilisateurs

Lors de l'accès à un fichier, le noyau Linux considère trois catégories d'utilisateurs :
• Le propriétaire du fichier (user ou u).
• Les membres du groupe (group ou g) auquel est affilé le fichier.
• Les autre utilisateurs (other ou o)

pour chaque catégorie, il existe trois droits d'accès, dont la signification dépend de la nature du fichier : ordinaire ou répertoire.

Les droits pour un fichier ordinaire

• Le droit de lecture (read ou r) permet de lire les données du fichier.
• Le droits d'écriture (write ou w) permet d'ajouter, supprimer ou modifier les données d'un fichier.
• Le droit d'exécution (execute ou x) permet de considérer le fichier comme une commande.

Remarque : le droit d'exécution ne doit être utilisé que pour des binaires résultant d'une compilation ou pour des scripts.

Les droits pour un répertoire

• Le droit de lecture (r) permet de connaître la liste des fichiers du répertoire.
• Le droits d'écriture (w) permet de modifier le répertoire : créer ou supprimer des entrées dans le répertoire.
• Le droit d'exécution (x) permet d'accéder aux fichier du répertoire.

Les utilisateurs et les droits

Les droits

Le sticky bit

ce droit, réservé à root, s'applique à un répertoire et corrige une bizarrerie du système. Par défaut, un répertoire accessible en écriture à un ensemble d'utilisateurs permet à l'un d'entre eux de détruire les fichier d'un autre utilisateur. Avec le *stycky bit* il faut être propriétaire d'un fichier pour avoir le droit de le détruire.

Les droits d'endossement (SUID, SGID) pour un exécutable

- Permettent d'augmenter les privilèges des utilisateurs.
- Le droit Set-UID (SUID) sur un binaire exécutable permet à l'utilisateur de l'application d'avoir les même droits d'accès que le propriétaire du binaire.
- Le droit Set-GID (SGID permet d'endosser les dorits du groupes auquel est affilé le binaire.

Exemple : le fichier /etc/shadow n'est en théorie accessible qu'à root. Or, tout utilisateur à accès en écriture à ce fichier lorsqu'il change son mot de passe grâce à la commande `/usr/bin/passwd`. L'explication réside dans le fait que cette commande, possédée par root possède le droit SUID et donne de fait à tous les utilisateurs les mêmes droits que root.

```
simoko@karmic:~$ ls -ld /usr/bin/passwd
-rwsr-xr-x 1 root root 41232 2009-05-06 00:37 /usr/bin/passwd
simoko@karmic:~$
```

Même si les droits d'endossement sont pratiques, il n'en demeure pas moins qu'ils sont dangereux.

Les utilisateurs et les droits

Les droits

Le droit SGID pour un répertoire

lorsque l'on crée un fichier, il est automatiquement affilé à son groupe courant, qui est par défaut son groupe principal. Si l'on crée un fichier dans un répertoire qui possède le droit SGID, son groupe sera identique à celui du répertoire. La conséquence c'est que l'ensemble des fichiers du répertoire appartiendra au même groupe, ce qui est intéressant pour un répertoire accessible à plusieurs personnes.

Les commandes

- `ls -l` : listes les caractéristiques d'un fichier, dont les droits.
- `chmod` : modifie les droits d'un fichier.
- `chgrp` : change le groupe d'un fichier.
- `chown` : change le propriétaire d'un fichier.
- `umask` : fixe les droits retirés automatiquement lors de la création d'un fichier.
- `cp -p` : copie de fichiers avec conservation des attributs.

Les utilisateurs et les droits

Les droits

Les droits en octal

```
simoko@karmic:~$ mkdir rep
simoko@karmic:~$ chmod 1000 rep
simoko@karmic:~$ ls -ld rep
d--------T 2 simoko simoko 4096 2009-10-19 02:04 rep
simoko@karmic:~$
```

Le sticky-bit est présent, mais pas le droit x pour les autres.

```
simoko@karmic:~$ chmod 1001 rep
simoko@karmic:~$ ls -ld rep
d--------t 2 simoko simoko 4096 2009-10-19 02:04 rep
simoko@karmic:~$
```

Le sticky-bit et le droit x sont présent pour les autres.

```
simoko@karmic:~$ chmod 2000 rep
simoko@karmic:~$ ls -ld rep
d-----S--- 2 simoko simoko 4096 2009-10-19 02:04 rep
simoko@karmic:~$
```

Le SGID est présent, mais pas le le droit x pour le groupe

```
simoko@karmic:~$ chmod 2010 rep
simoko@karmic:~$ ls -ld rep
d-----s--- 2 simoko simoko 4096 2009-10-19 02:04 rep
simoko@karmic:~$
```

Le SGID et le droit x sont présent pour le groupe.

Les utilisateurs et les droits

Les droits

Les droits en octal

```
simoko@karmic:~$ chmod 4000 file
simoko@karmic:~$ ls -ld file
---S------ 1 simoko simoko 0 2009-10-19 02:16 file
simoko@karmic:~$
```

Le SUID est présent, mais pas le droit x pour l'utilisateur.

```
simoko@karmic:~$ chmod 4100 file
simoko@karmic:~$ ls -ld file
---s------ 1 simoko simoko 0 2009-10-19 02:16 file
simoko@karmic:~$
```

Le SUID et le droit x sont présent pour l'utilisateur.

```
simoko@karmic:~$ chmod 6110 file
simoko@karmic:~$ ls -ld file
---s--s--- 1 simoko simoko 0 2009-10-19 02:16 file
simoko@karmic:~$
```

Le SUID, le SGID et les droit x sont présent pour l'utilisateur et le groupe.

Les utilisateurs et les droits

Exemples

Les tâches essentielles de gestion des utilisateurs

1. Est-ce que les comptes utilisateurs de *daemon* et luke existent, et si oui quels sont leurs uid, gid et leurs groupes ?

```
simoko@karmic:~$ id daemon
uid=1(daemon) gid=1(daemon) groupes=1(daemon)
simoko@karmic:~$ id luke
id: luke: usager inexistant.
simoko@karmic:~$
```

2. Créez les groupes *jedi* et *rebelles*.

```
root@karmic:/home/simoko# groupadd jedi
root@karmic:/home/simoko# groupadd rebelles
root@karmic:/home/simoko#
```

Les utilisateurs et les droits

Exemples

Les tâches essentielles de gestion des utilisateurs

3. Créez des comptes utilisateurs
Le compte *luke*, appartenant au groupe *jedi* (comme groupe principal) et au groupe *rebelles* (comme groupe secondaire). Le compte *vador* appartenant au groupe *jedi*. Et enfin, le compte *solo* faisant partie du groupes *rebelles*. On visualise ensuite les groupes.

```
root@karmic:/home/simoko# useradd -g jedi -G rebelles -m luke
root@karmic:/home/simoko# useradd -g jedi -m vador
root@karmic:/home/simoko# useradd -g rebelles -m solo
root@karmic:/home/simoko# id luke
uid=1001(luke) gid=1001(jedi) groupes=1001(jedi),1002(rebelles)
root@karmic:/home/simoko# id vador
uid=1002(vador) gid=1001(jedi) groupes=1001(jedi)
root@karmic:/home/simoko# id solo
uid=1003(solo) gid=1002(rebelles) groupes=1002(rebelles)
root@karmic:/home/simoko#
```

Les utilisateurs et les droits

Exemples

Les tâches essentielles de gestion des utilisateurs

4. Mettez le mot de « password » comme mot de passe à l'utilisateur *luke*.

```
root@karmic:/home/simoko# passwd luke
Entrez le nouveau mot de passe UNIX :
Retapez le nouveau mot de passe UNIX :
passwd : le mot de passe a été mis à jour avec succès
root@karmic:/home/simoko#
```

5. Essayez de vous connecter sous le compte *luke*.

```
simoko@karmic:~$ su luke
Mot de passe :
$ whoami
luke
$
```

Les utilisateurs et les droits

Exemples

L'essentiel de la gestion des droits

1. On crée une arborescence de fichier

```
root@karmic:/home/simoko# mkdir /home/etoilenoire
root@karmic:/home/simoko# cd /home/etoilenoire/
root@karmic:/home/etoilenoire# echo "voici les plans" > plans
root@karmic:/home/etoilenoire# echo "c'est ouvert" > entree_secret
root@karmic:/home/etoilenoire#
```

2. On change les caractéristiques du répertoire *etoilenoire*.
Son propriétaire sera *luke*, son groupe *jedi*. Il sera accessible en lecture, écriture et accès au propriétaire et au groupe mais pas aux autres. On positionne le *SGID* et le *sticky-bit*.

```
root@karmic:/home/etoilenoire#  cd
root@karmic:~# chown luke /home/etoilenoire/
root@karmic:~# chgrp jedi /home/etoilenoire/
root@karmic:~# chmod 3770 /home/etoilenoire/
root@karmic:~# ls -ld /home/etoilenoire/
drwxrws--T 2 luke jedi 4096 2009-10-19 02:55 /home/etoilenoire/
root@karmic:~#
```

Les utilisateurs et les droits

Exemples

L'essentiel de la gestion des droits

3. On change les caractéristiques des fichiers
Ils seront accessibles en lectures seule pour le groupe et n'auront aucun droit pour les autres. On utilise la notation symbolique. On affilie le fichier *plans* au groupe *jedi* et le fichier *entree_secrete* au groupe *rebelles*.

```
root@karmic:~# chmod g=r,o=- /home/etoilenoire/*
root@karmic:~# chgrp jedi /home/etoilenoire/plans
root@karmic:~# chgrp rebelles /home/etoilenoire/entree_secrete
root@karmic:~# ls -l /home/etoilenoire/
total 8
-rw-r----- 1 root rebelles 13 2009-10-19 02:55 entree_secrete
-rw-r----- 1 root jedi     16 2009-10-19 02:54 plans
root@karmic:~#
```

Les utilisateurs et les droits

Exemples

L'essentiel de la gestion des droits

4. On teste les accès

(a) A partir du compte *luke*.

```
root@karmic:~# su - luke
$ ls /home/etoilenoire
entree_secrete  plans
$ cat /home/etoilenoire/plans
voici les plans
$ cat /home/etoilenoire/entree_secrete
c'est ouvert
$ exit
```

(b) A partir du compte *vador*.

```
root@karmic:~# su - vador
$ ls /home/etoilenoire
entree_secrete  plans
$ cat /home/etoilenoire/plans
voici les plans
$ cat /home/etoilenoire/entree_secrete
cat: /home/etoilenoire/entree_secrete: Permission non accordée
$
```

(c) A partir du compte *solo*.

```
root@karmic:~# su - solo
$ ls /home/etoilenoire
ls: ne peut ouvrir le répertoire /home/etoilenoire: Permission non accor
dée
$ cat /home/etoilenoire/plans
cat: /home/etoilenoire/plans: Permission non accordée
$ cat /home/etoilenoire/entree_secrete
cat: /home/etoilenoire/entree_secrete: Permission non accordée
$
```

Les utilisateurs et les droits

Exemples

L'essentiel de la gestion des droits

5. On teste les accès
Supprimez temporairement les droits d'exécution à la commande uptime.

```
root@karmic:~# whereis uptime
uptime: /usr/bin/uptime /usr/share/man/man1/uptime.1.gz
root@karmic:~# ls -l /usr/bin/uptime
-rwxr-xr-x 1 root root 5432 2009-03-18 23:17 /usr/bin/uptime
root@karmic:~# chmod o-x /usr/bin/uptime
root@karmic:~# ls -l /usr/bin/uptime
-rwxr-xr-- 1 root root 5432 2009-03-18 23:17 /usr/bin/uptime
root@karmic:~# su - luke
$ uptime
-su: uptime: Permission denied
$ exit
root@karmic:~# chmod o+x /usr/bin/uptime
root@karmic:~# ls -l /usr/bin/uptime
-rwxr-xr-x 1 root root 5432 2009-03-18 23:17 /usr/bin/uptime
root@karmic:~# su - luke
$ uptime
 03:27:00 up 13:39,  2 users,  load average: 0.38, 0.14, 0.07
$
```

Les utilisateurs et les droits

Exemples

La gestion des utilisateurs, compléments

1. Affichez les caractéristiques de l'utilisateur *luke* et du groupe *rebelles*.

```
root@karmic:~# getent passwd luke
luke:x:1001:1001::/home/luke:/bin/sh
root@karmic:~# getent group jedi
jedi:x:1001:
root@karmic:~#
```

2. Affichez les caractéristiques de l'utilisateur local *luke* et du groupe local *rebelles*.

```
root@karmic:~# grep luke /etc/passwd
luke:x:1001:1001::/home/luke:/bin/sh
root@karmic:~# grep rebelles /etc/group
rebelles:x:1002:luke
root@karmic:~#
```

Les utilisateurs et les droits

Exemples

La gestion des utilisateurs, compléments

3. On crée l'utilisateur *leia*, quel est son groupe principal ?

```
root@karmic:~# useradd leia
root@karmic:~# id leia
uid=1004(leia) gid=1004(leia) groupes=1004(leia)
root@karmic:~#
```

Remarque : par défaut, la création d'un compte utilisateur entraîne la création d'un compte groupe de même nom qui correspond au groupe principal du nouvel utilisateur.

4. On veut affecter l'utilisateur leia au groupe rebelle (comme groupe secondaire).

```
root@karmic:~# usermod -G rebelles leia
root@karmic:~# id leia
uid=1004(leia) gid=1004(leia) groupes=1004(leia),1002(rebelles)
root@karmic:~#
```

Les utilisateurs et les droits

Exemples

La gestion des utilisateurs, compléments

5. Recherchez les fichiers de l'utilisateur *luke*.

```
root@karmic:~# find /home -user luke
/home/luke
/home/luke/examples.desktop
/home/luke/.bashrc
/home/luke/.bash_logout
/home/luke/.profile
```

6. Supprimez un compte utilisateur et les fichiers de son répertoire de connexion.

```
root@karmic:~# userdel -r leia
userdel : erreur lors de l'effacement du répertoire /home/leia
root@karmic:~# id leia
id: leia: usager inexistant.
root@karmic:~#
```

Les utilisateurs et les droits

Exemples

La gestion des utilisateurs, compléments

7. On veut recréer le compte *leia* à l'identique (il doit avoir le même *uid* et *gid*).

```
root@karmic:~# groupadd -g 1004 leia
root@karmic:~# useradd -u 1004 --gid leia leia
root@karmic:~# id leia
uid=1004(leia) gid=1004(leia) groupes=1004(leia)
root@karmic:~#
```

8. Créez le compte *toor* ayant les mêmes droits que *root*

```
root@karmic:~# useradd -u 0  -o -d /root toor
root@karmic:~# id toor
uid=0(root) gid=1005(toor) groupes=0(root)
root@karmic:~#
```

Remarques : Si l'on affecte un mot de passe à cet utilisateur, il devient une sorte de secours dans le cas où l'on perd le mot de passe de root.
L'option -u permet de fixer l'uid, mais on ne peut utiliser un uid existant... sauf si l'on utilise l'option -o.

Les utilisateurs et les droits

Exemples

La gestion des droits, complément.

1. On crée des fichiers dans le répertoire *etoilenoire*.
Sous le compte *root*, on crée le fichier f1. Sous le compte *luke*, on crée le fichier f2. Et sous le compte *vador*, on crée le fichier f3.

```
root@karmic:~# ls -ld /home/etoilenoire/
drwxrws--T 2 luke jedi 4096 2009-10-19 03:05 /home/etoilenoire/
root@karmic:~# echo "fichier un" > /home/etoilenoire/f1
root@karmic:~# su luke -c "echo bonjour > /home/etoilenoire/f2"
root@karmic:~# su vador -c "echo bonjour > /home/etoilenoire/f3"
root@karmic:~# ls -l /home/etoilenoire/
total 20
-rw-r----- 1 root  rebelles 13 2009-10-19 02:55 entree_secrete
-rw-r--r-- 1 root  jedi     11 2009-10-19 04:06 f1
-rw-r--r-- 1 luke  jedi      8 2009-10-19 04:07 f2
-rw-r--r-- 1 vador jedi      8 2009-10-19 04:07 f3
-rw-r----- 1 root  jedi     16 2009-10-19 02:54 plans
root@karmic:~#
```

Remarque : du fait du droit *SGID*, l'ensemble des fichiers sont affiliés au groupe *jedi*, le groupe du répertoire .
La commande `su -c` permet à root d'exécuter une commande avec les droits d'un utilisateurs ordinaire.

Les utilisateurs et les droits

Exemples

La gestion des droits, complément.

2. *Vador* va essayer de détruire le fichier de *luke*.

a) on conserve le droit *sticky-bit*.

```
root@karmic:~# su - vador
$ rm /home/etoilenoire/f2
rm: détruire un fichier protégé en écriture fichier standard `/home/etoilenoir
e/f2'? y
rm: ne peut enlever `/home/etoilenoire/f2': Opération non permise
$
```

b) on supprime le droit *sticky-bit*.

```
root@karmic:~# chmod -t /home/etoilenoire/
root@karmic:~# su - vador
$ rm /home/etoilenoire/f2
rm: détruire un fichier protégé en écriture fichier standard `/home/etoilenoir
e/f2'? y
$ ls -l /home/etoilenoire/f2
ls: ne peut accéder /home/etoilenoire/f2: Aucun fichier ou dossier de ce type
$
```

Remarque : dans tous les cas *luke* peut détruire le fichier de *vador*, il est propriétaire du répertoire.

Les utilisateurs et les droits

Exemples

La gestion des droits, complément.

3. L'administrateur copie les fichiers du répertoire *etoilenoire* dans */tmp* en conservant leurs attributs

```
root@karmic:~# cp -p /home/etoilenoire/* /tmp
root@karmic:~# ls -l /tmp/plans  /tmp/entree_secrete
-rw-r----- 1 root rebelles 13 2009-10-19 02:55 /tmp/entree_secrete
-rw-r----- 1 root jedi     16 2009-10-19 02:54 /tmp/plans
root@karmic:~#
```

Remarque : normalement quand on copie un fichier , la copie appartient à celui qui copie. Ici, l'administrateur, conserve le propriétaire d'origine grâce à l'option *-p* de *cp*.

4. L'administrateur donne le fichier *entree_secrete* à *luke*.

```
root@karmic:~# chown luke /tmp/entree_secrete
root@karmic:~# ls -l /tmp/entree_secrete
-rw-r----- 1 luke rebelles 13 2009-10-19 02:55 /tmp/entree_secrete
root@karmic:~#
```

Les utilisateurs et les droits

Exemples

La gestion des droits, complément.

5. On teste les accès *(r, w, x)* au fichier */tmp/entre_secrete*.

a) A partir du compte *luke*.

```
root@karmic:~# su - luke
$ cat /tmp/etoilenoire
cat: /tmp/etoilenoire: Aucun fichier ou dossier de ce type
$ exit
root@karmic:~# su - luke
$ cat /tmp/entree_secrete
c'est ouvert
$ echo "==========" >> /tmp/entree_secrete
$ /tmp/entree_secrete
-su: /tmp/entree_secrete: Permission denied
$
```

Les utilisateurs et les droits

Exemples

La gestion des droits, complément.

5. On teste les accès *(r, w, x)* au fichier */tmp/entre_secrete*.

b) A partir du compte *solo*.

```
root@karmic:~# su - solo
$ cat /tmp/entree_secrete
c'est ouvert
==========
$ echo "+++++++" >> /tmp/entree_secrete
-su: cannot create /tmp/entree_secrete: Permission denied
$
```

Les utilisateurs et les droits

Exemples

La gestion des droits, complément.

5. On teste les accès *(r, w, x)* au fichier */tmp/entre_secrete*.

c) A partir du compte *root*.

```
root@karmic:~# echo "+=+=+=+" >> /tmp/entree_secrete
root@karmic:~# /tmp/entree_secrete
bash: /tmp/entree_secrete: Permission non accordée
root@karmic:~#
```

Remarque : les droits des utilisateurs ne s'appliquent pas à l'administrateur. Seule restriction, comme tout le monde, *root* a besoin du droit d'exécution pour activer une commande.

Les utilisateurs et les droits

Exemples

La gestion des droits, complément.

6. On visualise les droits du fichier *shadows* et de la commande *passwd*.

```
root@karmic:~# ls -l /etc/shadow
-rw-r----- 1 root shadow 1182 2009-10-19 03:57 /etc/shadow
root@karmic:~# ls -l /usr/bin/passwd
-rwsr-xr-x 1 root root 41232 2009-05-06 00:37 /usr/bin/passwd
root@karmic:~#
```

Remarque : tout le monde a le droit d'exécuter la commande *passwd*. Avec son droit *SUID*, un utilisateur endosse les droits de *root*, ce qui lui permet d'accéder au fichier *shadow*.

Les utilisateurs et les droits

Exemples

La gestion des droits, complément.

7. Visualisez les effets de la commande *umask*

```
root@karmic:~# rm /tmp/f?
root@karmic:~# umask 0
root@karmic:~# umask
0000
root@karmic:~# echo "bonjour" > /tmp/f
root@karmic:~# umask 77
root@karmic:~# echo "bonjour" > /tmp/f2
root@karmic:~# ls -l /tmp/f?
-rw------- 1 root root 8 2009-10-19 04:55 /tmp/f2
root@karmic:~# ls -l /tmp/f*
-rw-rw-rw- 1 root root 8 2009-10-19 04:55 /tmp/f
-rw------- 1 root root 8 2009-10-19 04:55 /tmp/f2
root@karmic:~#
```

Remarque : l'effet de la commande *umask* est limité à la durée de vie du shell. Pour avoir un effet, il faut mettre cette commande dans le fichier *.bash_profile* de l'utilisateur.

En plus de l'action du *umask*, la plupart des commande (mais pas un compilateur !) supprime le droit d'exécution.