Système partie 1 - cours 3

Pastille 3 : les verrous

N. de Rugy-Altherre - Vincent Colotte

Protection sous Linux

Protection ressources système

- Accès à des ressources critiques ? ? ?
- Normalement seul le super-utilisateur (root) ! [UID=0]
- Exemple : fichier `/etc/shadow` contenant les mots de passe.

Bit de protection : set-uid

- À l'exécution d'un fichier binaire (si positionné)
- Donne UID du propriétaire du fichier binaire
- On parle alors UID **effective*** du processus ($\neq$ UID **réelle**).
- Les droits sont conditionnés par l'UID effective d'un processus.

[si le bit set-uid n'est pas positionné l'UID effective reste identique à l'UID réelle]

Protection sous Linux

Exemple : le binaire *passwd* change le mot de passe dans */etc/shadow*

```
⇒-rws r-x r-x root root /usr/bin/passwd
⇒-rw- r-- --- root shadow /etc/shadow
```

Processus qui fait *passwd* :

- UID réelle = 1234, donc normalement */etc/shadow* est inaccessible (---)
- set-uid positionné ⇒ UID effective devient 0 (=root)
- */etc/shadow* est accessible et modifiable pour *root* et donc aussi pour le processus
  ...mais seulement par cette fonction (l'accès est donc sous "contrôle").

Fonctions C

Quelques appels système pour accéder/modifier les droits :

- chmod(...) : Change les protections (existe comme commande en shell)
- access(...) : vérifie l'accès avec les UID et GID réels
- getuid(), geteuid() : récupère UID réelle et effective
- getgid(), getegid() : récupère GID réelle et effective

`uid` : user id
`gid` : group id

Exercice

Exercice

- Écrivez un fichier en C affichant les `gid` effectifs et réels. Compilez-le en nommant `a.out` le fichier exécutable. Éxécutez-le.
- Puis avec la commande bash `chmod` changez le groupe du propriétaire du fichier. Re-exécutez votre fichier (sans le compiler de nouveau). Que constatez-vous ?