

TD 5 - ASR7 Programmation concurrente

Administration système

Sylvain Brandel, Guillaume Damiand, Laurent Lefèvre, Matthieu Moy, Grégoire Pichon

Printemps 2022

I Des droits et des fichiers

Sur le système considéré dans cet exercice, on trouve notamment les trois utilisateurs suivants :

- `a1400` qui fait partie des groupes `etudiant` et `user` ;
- `frico` qui fait partie des groupes `prof` et `user` ;
- `ycaiou` qui fait partie des groupes `prof`, `user` et `ycaiou`

Dans un répertoire ayant tous les droits d'accès, la commande `ls -l` retourne ce qui suit :

```
srw----- 1 ycaiou ycaiou 0   déc.  3 06:24 kdeinit4__0
lrwxrwxrwx 1 ycaiou ycaiou 33   mai   2 2014 rap.tex -> rapport.tex
drwxr-xr-x 27 frico  user   27  janv.  2 11:06 public/
drwx--x--- 70 frico  prof  4096  avril 22 17:21 prive/
-rw-rw-r-- 1 a1400  etu   5349  déc. 10 2008 rapport.tex
-rw-rw---- 1 frico  prof   336  avril 19 12:45 public/sujet.tex
-rw-rw-r-- 1 frico  prof   336  avril 19 12:45 public/sujet.pdf
-rw-r--r-- 1 ycaiou  prof   336  avril 19 18:32 public/correction.pdf
-rw-r--r-- 1 frico  prof   336  avril 19 18:32 prive/notes.ods
```

Q.I.1) - Qu'est-ce que le fichier `kdeinit4__0`? Et `rap.tex`?

Q.I.2) - Les droits sur `rap.tex` sont-ils normaux?

Q.I.3) - Représentez dans une matrice les possibilités d'accès des fichiers et répertoires pour chaque utilisateur.

Q.I.4) - Qu'est-ce qu'une telle « configuration » des droits sur `public/` et `prive/` permet de faire?

II Le bit « `setuid` »

On considère maintenant cette situation (le `$` est l'invite de commande, ou « prompt ») :

```
$ sudo ls /root
$ cat test-setuid.c
#include <fcntl.h>
#include <unistd.h>
#include <string.h>
#include <stdio.h>
#include <stdlib.h>

int main() {
    int f = open("/root/i-can-haz-write-access-to-r00t.txt",
```

```

        O_WRONLY | O_CREAT);
    if (f < 0) {
        perror("Can't open file");
        exit(1);
    }
    write(f, "pwned", strlen("pwned"));
    close(f);
}
$ gcc test-setuid.c -o test-setuid
$ ls -l test-setuid
-rwxr-xr-x 1 moy moy 8808 nov. 30 17:27 test-setuid*
$ ./test-setuid
Can't open file: Permission denied
$ sudo chown root test-setuid
$ ./test-setuid
Can't open file: Permission denied
$ sudo chmod +s test-setuid
$ ls -l test-setuid
-rwsr-sr-x 1 root moy 8808 nov. 30 17:27 test-setuid*
$ ./test-setuid
$ sudo ls /root
i-can-haz-write-access-to-r00t.txt
$

```

Q.II.1) - Pourquoi les premières exécutions de `test-setuid` échouent-elles ? Pourquoi la dernière réussit-elle ?

III Admin Système

Lorsque la ou les commandes sont propres à une distribution, nous supposons qu'il s'agit d'une distribution Debian ou dérivée (Ubuntu, comme votre VM Openstack).

Q.III.1) - Rappelez quelques variables d'environnement importantes. Comment toutes les lister avec leur affectation ?

Q.III.2) - Quelle est la différence entre sourcer un script (`source mon-script.sh` ou `. mon-script.sh`) et l'exécuter (`./mon-script.sh`) ?

Q.III.3) - Comment mettre en place la complétion *intelligente* ? À quoi sert-elle ?

Q.III.4) - Quel est le pseudo système de fichiers permettant de configurer dynamiquement le fonctionnement du noyau ?

Les commandes suivantes peuvent se faire avec `apt`, un front-end qui se veut convivial pour gérer les paquets Debian. Mais `apt` ne fournit pas encore toutes les choses des autres outils, comme la recherche du package auquel appartient un fichier donné.

Q.III.5) - Comment lister tous les packages installés ?

Q.III.6) - Comment savoir si le package `apache2` est installé ?

Q.III.7) - Où modifier les *repositories* d'où la liste des packages et leur version, ainsi que les packages eux-mêmes seront téléchargés ?

Q.III.8) - Comment savoir quelle version d'un package sera installée si on décide de l'installer ?

Q.III.9) - Comment mettre à jour l'ensemble du système ?