TP - ASR5 système d'exploitation

Un peu de système

9 avril 2018

Pour ce TP, nous avons créé plusieurs machines virtuelles sur lesquelles vous allez vous connecter. Votre charger de TP vous indiquera l'adresse de celle que vous devez utiliser. Ces machine sont connues par leur adresse IP, dans le sujet, celle-ci est nomée `IP_VM`. Ces machines sont reliées à l'anuaire de l'université, pour vous connecter vous allez donc utiliser votre login de l'université et votre mot de passe. Dans la suite du TP, votre login sera noté `UTILISATEUR`.

Contrairement à ce qui était anoncé, il n'y aura plus de TP noté. cette séance et la suivante portera sur l'utilisation du sytème linux. Mais prenez des notes, lors du dernier contrôle, certaines questions porteront sur l'utilisation des commandes vues aujourd'hui.

I Gestion des utilisateur

Connectez-vous sur la machine fournie. Vous devez être administrateur pour faire cette exercice. Si vous n'êtes pas parvenu à le faire via les commandes du TP précédant, ne vous inquiétez pas, nous avons modifié la configuration générale.

Q.I.1) - Créez un utilisateur dont le login est `votreprenom.votrenom` via la commande `adduser`. Pouvez-vous directement utiliser la commande `adduser` ? Et via `sudo` ?

> **Solution:** `adduser` est interdit, c'est bien sur la commande `sudo` qui permet de le faire.

Q.I.2) - Regardez les fichiers de configuration `/etc/sudoers` et `/etc/sudoers.d/*`. Qu'est-ce qui vous permet d'utiliser la commande sudo ?

> **Solution:** Il y a 3 lignes de ces fichiers qui peuvent concerner les comptes étudiants :
> - `/etc/sudoers/: %admin ALL=(ALL:ALL) ALL`
> - `/etc/sudoers.d/42-tp-systeme: %admin ALL=(ALL:ALL) NOPASSWD:ALL`
> - `/etc/sudoers.d/42-tp-systeme: %etudiant ALL=(ALL:ALL) ALL`
>
> Les 2 premières concernent les étudiants qui on réussi le TP précédant et ont pu s'ajouter au groupe admin. La dernière permet à tous les étudiants de pouvoir utiliser la commande sudo sans restriction.
>
> Le premier fichier de configuration est le fichier standard de la distribution. Le second fichier a été ajouté pour permettre de faire cet exercice. C'est la modification de « configuration générale » dont il est question au début de l'exercice.

Q.I.3) - Que signifie au niveau du système la création d'un utilisateur ? Regardez dans les fichiers `/etc/group`, `/etc/passwd`, `/etc/shadow`. Quelles lignes correspondent à l'utilisateur que vous avez créé.

> **Solution:** En supposant l'utilisateur `toto` :
> - `/etc/passwd: toto:x:1005:1005:test,203,06,04,truc:/home/toto:/bin/bash` est la ligne correpondant à l'utilisateur `toto` dans le système. Toutes les information fournie lors de la création sont là, séparées par des « : ». Seul le mot de passe manque, historiquement, c'est le second champs (le « x ») qui sontenait cette valeur.
> - `/etc/shadow: toto:$6$3N1ccgt8$jJy4k4p[...]:17637:0:99999:7:::` Contient l'empreinte du mot de passe et des informations sur ce dernier (voir `man shadow`).

— `/etc/group: toto:x:1005:` est la ligne contenant le groupe `toto` qui a été crée par défaut comme groupe principal de l'utilisateur. On peux voir cela car le *gid* du groupe 1005 est le *gid* principal de l'utilisateur (le 4ème champs).

Q.I.4) - Ajoutez l'utilisateur au groupe admin (`usermod -aG admin LOGINUSER`). Qu'est-ce que cela change ?

Solution: Dans `/etc/group`, le login de l'utilisateur est ajouté à la ligne correspondant au groupe `admin`. Dans ce fichier, chaque ligne correspond à la liste des utilisateurs qui en sont membres (sauf si c'est leur groupe principal).

Q.I.5) - **Attention ne faite pas cet exercice sur votre propre ordinateur** Votre chargé.ée de TP va interdire la lecture du fichier `/etc/passwd`. Que se passe-t-il ? Essayez de changer l'utilisateur courant vers celui que vous avez créé :

5(a) - Cela fonctionne-t-il en utilisant la commande `sudo su - VOTREPREONOM.VOTRENOM` ?

5(b) - Cela fonctionne-t-il en utilisant la commande `su - VOTREPREONOM.VOTRENOM` ?

Solution: Il faut taper `chmod a-r /etc/passwd` Attention, seul le.a chargé.ée de TP doit le faire. Pour rétablir la situation, il faut taper `chmod a+r /etc/passwd`.

Les 2 doivent fonctionner. Le changement d'utilisateur est possible car `root` peut lire `/etc/passwd` quelque soit les droits du fichier. Par contre, l'utilisateur ne peut pas lire le fichier, il ne peut donc pas obtenir d'information sur lui même. Cela explique que son prompt affiche : « I have no name ! ».

I.1 Si vous êtes en avance

Q.I.6) - La manipulation précédente ne change pas le fonctionnement de votre compte universitaire. Où la machine va-t-elle chercher les informations correspondant à ce compte ? Regardez dans le fichier `/etc/nsswich.conf`.

Solution:

```
# /etc/nsswitch.conf
#
# Example configuration of GNU Name Service Switch functionality.
# If you have the ‘glibc-doc-reference’ and ‘info’ packages installed, try:
# ‘info libc "Name Service Switch"’ for information about this file.

passwd:         compat sss
group:          compat sss
shadow:         compat sss
gshadow:        files
...
```

Les informations utilisateurs de cette machine ont 2 sources (`man nsswitch.conf`) :

— `compat` est le vieu mode permettant de lire les informations dans les fichiers avec possibilité de rechercher des information dans des services `NIS`.

— `sssd` (System Security Services Daemon) est le service qui gère la récupération d'informations utilisateurs via des serveurs distants (ldap ou kerberos).

Si on bloque l'une des sources (ici les fichiers), la seconde reste utilisée (ici le serveur distant) et cela n'a pas de conséquence sur les comptes que le second gère (ici ls compte de l'université).

Q.I.7) - Que signifie l'empreinte du mot de passe stockée dans `/etc/shadow` ? Le mot de passe stocké ressemble à cela :

`$6$3N1ccgt8$jJy4k4p7SmAMUhvK1J0bSqIIq3HI7JBxkCZ5xbv/X4Gq4fdR/F4L/EWercjmfYRKXoc4zuRbKFJE3OTM4H0`
Il se décompose en 3 morceau (séparé par des $). Que signifie ces morceaux (voir le manuel de `crypt(3)`.

Solution: `man 3 crypt` (c'est le manuel de la fonction C utilisée pour générer l'empreinte. La fonction utilisée sous linux est une amélioration de la fonction de base programmée dans la bibliothèque glibc.

```
...
The glibc2 version of this function supports additional encryption algorithms.

If salt is a character string starting with the characters "$id$" followed by
a string terminated by "$":

       $id$salt$encrypted
then  instead  of  using  the  DES  machine, id identifies the encryption
method used and this then determines how the rest of the password string is
interpreted.  The following values of id are supported:

       ID  | Method
       ---------------------------------------------------
       1   | MD5
       2a  | Blowfish (not in mainline glibc; added in some
           | Linux distributions)
       5   | SHA-256 (since glibc 2.7)
       6   | SHA-512 (since glibc 2.7)

So $5$salt$encrypted is an SHA-256 encoded password and $6$salt$encrypted is
an SHA-512 encoded one.

"salt" stands for the up to 16 characters following "$id$" in the salt.  The
encrypted part of the password string is the actual  computed  password.
The size of this string is fixed:

MD5     | 22 characters
SHA-256 | 43 characters
SHA-512 | 86 characters

The  characters  in  "salt"  and  "encrypted" are drawn from the set
[a-zA-Z0-9./].  In the MD5 and SHA implementations the entire key is
significant (instead of only the first 8 bytes in DES).
```

Dans notre exemple, l'algorithme utilisé est Shamir 512 et la graine `3N1ccgt8`. La graine est une chaine tirée au hasard et qui est ajoutée au mot de passe afin que deux mots de passe identique ne donnent pas la même empreinte.

Q.I.8) - Grâce à la commande `mkpasswd` pouvez-vous regénérer l'empreinte ?

Solution: En prenant la graine de l'exemple :

```
mkpasswd -m SHA-512 -S 3N1ccgt8
```

Il faut ensuite taper le mot de passe.

II Utilisation des logs

Les fichiers de logs du système sont dans le répertoire `/var/log`. Aujourd'hui nous allons nous intéresser au fichier `/var/log/auth.log`

Q.II.1) - Qui a le droit de le lire d'y écrire et pourquoi ?

> **Solution:** `syslog` est le propriétaire. Il peut écrire dessus. En fait, `syslog` est le compte de service dont l'identité est utilisé pour gérer les logs. Les utilisateurs qui peuvent lire ce fichier sont ceux du groupe `adm` uniquement. En effet, les informations contenues dans ce fichier peuvent être considérées sensibles

Q.II.2) - Que contient-il ?

> **Solution:** `sudo less /var/log/auth.log`
> Le fichier contient toutes les sessions ouvertes dans le système via les différents moyens possible :
> — `sshd` connexion via le serveur `ssh`
> — `login` connexion via la console
> — `su` changement d'utilisateur via `su`
> — `sudo` changement d'utilisateur pour une commande via la commande `sudo`

Q.II.3) - Grâce à la la commande `grep` retrouvez les ligne correspondant à chaque fois où la commande `sudo` a été utilisée. Affichez les lignes mentionnant préciséement la commande qui a été exécutée.

> **Solution:**
>
> ```
> grep sudo /var/log/auth.log | grep COMMAND
> ```

Q.II.4) - Grâce à la commande `sed` retrouvez dans le résultat précédent chaque utilisateur et chaque commande utilisée sous la forme de 2 colonne "UTILISATEUR COMMANDE" [1].

> **Solution:**
>
> ```
> sudo grep sudo /var/log/auth.log | grep COMMAND | \
>      sed -e 's/.*sudo: *\(.*\) : TTY.*COMMAND=\(.*\)$/\1 \2/'
> ```

Q.II.5) - Grâce aux commandes `sort` et `uniq` triez par utilisateur et commande identiques puis compter le nombre d'ocurence de chaque commande.

> **Solution:**
>
> ```
> sudo grep sudo /var/log/auth.log | grep COMMAND \
>      | sed -e 's/.*sudo: *\(.*\) : TTY.*COMMAND=\(.*\)$/\1 \2/' \
>      | sort -k 1,2 | uniq -c
> ```

III Informations sur les processus

Copiez le fichier `code1.c` sur la machine virtuelle. Ils se compilent avec la commande
`gcc -g -Wall -O0 code1.c -o code1.exx`

Dans un `screen`, lancez le programme `./code1.exx`, retrouvez son `pid` et explorez le répertoire `/proc/PID/`.

1. Pensez au expréssion régulières de sed : sed −e `'s/sudo:(.*)/\1/'` remplace chaque ligne contenant `sudo: truc` par `truc`

Q.III.1) - Que contiennent les fichiers `cmdline`, `environ`, `loginuid`, `cwd`, `maps`

Solution:
- `cmdline` contient la ligne de commande utilisée pour le lancement ;
- `environ` contient les variables d'environnement ;
- `loginuid` contient l'identité de l'utilisateur qui a créé le processus ;
- `cwd` est un lien vers le répertoire courant ;
- `maps` contient le mapping mémoire

Q.III.2) - Regardez le contenu du répertoire `fd`, que contient-il ?

Solution: Pour chaque *file descriptor*, le répertoire contient un lien vers le fichier utilisé.

Q.III.3) - Comment retrouver les processus qui utilisent un fichier présent dans le répertoire `/home/` ?

Solution:

```
sudo ls -l  /proc/*/fd/* | grep home
```

On peut retrouver la même chose grâce à la commande `lsof`

Q.III.4) - Regardez le contenu du fichier `maps`. Que signifient les lignes de ce fichier ?

Solution: Chaque ligne correspond à un segment mémoire réservé par le système pour ce processus :

```
00400000-00401000 r-xp 00000000 fd:01 571756        /home/pers/fabien.rico/code1.exx
00600000-00601000 r--p 00000000 fd:01 571756        /home/pers/fabien.rico/code1.exx
00601000-00602000 rw-p 00001000 fd:01 571756        /home/pers/fabien.rico/code1.exx
00bf5000-00c16000 rw-p 00000000 00:00 0             [heap]
7f87abec5000-7f87ac085000 r-xp 00000000 fd:01 4247  /lib/x86_64-linux-gnu/libc-2.23.so
7f87ac085000-7f87ac285000 ---p 001c0000 fd:01 4247  /lib/x86_64-linux-gnu/libc-2.23.so
7f87ac285000-7f87ac289000 r--p 001c0000 fd:01 4247  /lib/x86_64-linux-gnu/libc-2.23.so
7f87ac289000-7f87ac28b000 rw-p 001c4000 fd:01 4247  /lib/x86_64-linux-gnu/libc-2.23.so
7f87ac28b000-7f87ac28f000 rw-p 00000000 00:00 0
7f87ac28f000-7f87ac2b5000 r-xp 00000000 fd:01 4230  /lib/x86_64-linux-gnu/ld-2.23.so
7f87ac49f000-7f87ac4a2000 rw-p 00000000 00:00 0
7f87ac4b4000-7f87ac4b5000 r--p 00025000 fd:01 4230  /lib/x86_64-linux-gnu/ld-2.23.so
7f87ac4b5000-7f87ac4b6000 rw-p 00026000 fd:01 4230  /lib/x86_64-linux-gnu/ld-2.23.so
7f87ac4b6000-7f87ac4b7000 rw-p 00000000 00:00 0
7ffccac42000-7ffccac63000 rw-p 00000000 00:00 0     [stack]
7ffccad16000-7ffccad19000 r--p 00000000 00:00 0     [vvar]
7ffccad19000-7ffccad1b000 r-xp 00000000 00:00 0     [vdso]
ffffffffff600000-ffffffffff601000 r-xp 00000000 00:00 0 [vsyscall]
```

- Les 3 premiers correspondent au code du programme. La partie exécutable, la partie des variables constantes et la partie des variables.
- Le 4e est le tas *heap*, c'est à dire la mémoire réservée par des new.
- Les suivant correspondent à des librairies partagées projetées en mémoire. `libc` contient les fonctions de base du C et `ld` contient les fonctions gérant les librairies.
- La pile est le segment marqué `stack`
- Les 3 derniers sont des segments utilisés pour avoir des appels systèmes simplifiés (cf `https://lwn.net/Articles/615809/` et `https://lwn.net/Articles/446528/`).

Vous pouvez aussi avoir les informations (et même plus de détails avec la commande `pmap -X PID`).