Labo 6

Loïc Brasey, Bastien Pillonel

1 TASK 1

1.1 Interprétation des informations sur les comptes et les groupes

Exécutez la commande id pour afficher des informations sur votre compte et les groupes dont vous êtes membres :

```
bigzb@lb-laptop:~$ id
uid=1000(bigzb) gid=1000(bigzb) groups=1000(bigzb),4(adm),6(disk),20(dialout),27(sudo)
```

Quel est votre UID et quel est votre nom de compte ?

— Ici on voit que mon UID est 1000 et mon nom d'utilisateur est `bigzb`

Quel est le GID de votre groupe principal ("groupe principal") et quel est son nom ?

— Ici on voit que le GID de mon groupe principale est 1000 et le nom du groupe est `bigzb`

De combien d'autres groupes êtes-vous membre ?

— On peut voir que le champs groups contient 12 groupes au total j'appartient donc à 11 autres groupes que le principale.

1.2 Interprétation des métadonnées de contrôle d'accès sur les fichiers et répertoires

1) Pour les fichiers suivants, déterminez qui est le propriétaire et quel groupe possède le fichier et caractériser le groupe de personnes qui savent lire, qui savent écrire et qui peuvent exécuter le fichier.

File	Owner	Read	Write	Exec.
/etc/passwd	root	All	Owner	None
/bin/ls	root	All	Owner	All
~/.bashrc	bigzb	All	Owner	None
~/.bash_history	bigzb	Owner	Owner	None

Pour trouver ces informations, j'utilise la commande suivante :

```
ls -la <path/to/file>
```

2) Examinez les autorisations de votre répertoire home.

Quelles options devez-vous passer à ls pour examiner les permissions des répertoires ?

```
ls -ld ~
drwxr-x--- 75 bigzb bigzb 4096 Apr 29 13:13 /home/bigzb
```

Qui est le propriétaire et quel est le groupe propriétaire ?

— Ici c'est bigzb le propriétaire et aussi le groupe principale du propriétaire (moi).

Quelle est la configuration des autorisations ?

— Qui peut lister les fichiers ?
— Le propriétaire et les membre du groupe seulement.
— Qui peut créer des fichiers ?

— Seul le propriétaire en est capable.

3. Quelles autorisations vous permettent de créer des fichiers dans le répertoire /tmp ?

```
ls -ld /tmp
drwxrwxrwt 31 root root 12288 Apr 29 17:04 /tmp
```

Dans le dernier champs des permissions, on peut voir un `t` à la place du `x`. Cela veut dire que les fichier créer dans ce répertoire ne peuvent être renommé ou supprimé que par un utilisateur qui a la permission w sur le répertoire et qui est le propriétaire du fichier.

On appel ça le sticky bit.

1.3 Modification des droits d'accès

1. Créez un fichier et initialisez ses autorisations sur rw- — — avec les commandes suivantes :

```
touch file
chmod 600 file

# rw- r-- ---
chmod g+r file

#rwx r-x ---
chmod g+rx file

#r-- r-- r--
chmod a=r file

#rwx r-- r--
chmod a+r,u+x file

#rwx --- ---
chmod u+x file
```

2. Autorisations conflictuelles - Créez un fichier (vous en serez le propriétaire) où les autorisations sont configurées pour

— ne pas autoriser le propriétaire ou le groupe à écrire dans le fichier, mais autorisez les autres utilisateurs à écrire dans le fichier.

```
$ chmod a=,o=rwx file
$ echo "hi" > file
bash: file: Permission denied
```

1.4 Donner à d'autres utilisateurs l'accès à vos fichiers

Votre collègue est-il capable de lire les fichiers de votre répertoire personnel ? Si oui, pourquoi ? Si non, pourquoi pas ?

— Oui car par défaut la permission de read est donner à tous le monde.

Que devez-vous faire pour que votre collègue (et peut-être d'autres) puisse lire vos fichiers ?

— Il faut ajouter la permission de read pour les autre (o+r) ou bien faire en sorte que les personnes désirant lire les fichiers soit dans un groupe qui possède la permission read sur les fichiers.

Que devez-vous faire pour que personne d'autre ne puisse lire vos fichiers ?

— Il faut retiré la permissions de read pour les autres utilisateurs (0-r).

Exo avec le dossier `shared` :

```
cd
mkdir shared
chgrp proj_a shared
chmod 070 shared
```

1.5 Find

1. Que fait find avec les fichiers ou dossiers cachés ?

— Il les affichent mais par contre il n'affiche pas les dossiers `..` et seulement le `.` du répertoire courant.

2. Utiliser find pour afficher tous les fichiers de votre répertoire personnel :

— qui finissent par `.c`, `.cpp` ou `.sh` :

```
find . -type f -name "*.c" -o -name "*.cpp" -o -name "*.sh"
```

— qui sont executables :

```
find . -type f -executable # GNU-based find only (not available to Mac-OS user)
```

— qui n'ont pas été modifiés depuis plus de 2 ans :

```
find . -type f -mtime +730
```

— qui n'ont pas été accédés depuis plus de 2 ans :

```
find . -type f -atime +730
```

— qui n'ont pas été accédés depuis plus de 3 ans et plus de 3MB :

```
find . -type f -atime +1095 -size +3M
```

— les dossier qui ont pour nom `.git` :

```
find  . -type d -name ".git"
```

3. Supposons que votre répertoire actuel ne contienne aucun sous-répertoire. Vous souhaitez afficher tous les fichiers contenant le mot `root` . Laquelle des deux commandes est correcte et pourquoi ?

C'est évidement la première car la deuxième fait appelle au globing pour retrouver les fichiers or on se rappel bien que le globbing ne prend pas en compte les fichiers cachés.

```
find . -type f -exec grep -l 'root' {} \;
```

2 TASK 2

Commande find pour lister tous les fichiers et repos world writable :

```
find . -perm -o+w
```

Execution du script fix_permissions.sh :

```
./fix_permissions.sh
```

L'execution me trouve bien les fichiers et directory world writable dans l'arborescence test_dir que j'ai pris soin de modifié pour avoir certains fichier/dir en world writable.

Arborescence test_dir.txt

3 TASK3

Execution du script fix_permissions.sh :

```
./fix_permissions.sh <directory_name>
```

Arborescence test_dir.txt

4 TASK4

Pour la tâche 4 si la recherche de fichier world writable donne un résultat non vide alors on demande à l'utilisateur s'il souhaite enlever la permission world writable. Tant qu'il ne répond pas "y" ou "n" on lui demande. S'il le souhaite on fixe les permission sinon on ne fait rien.

Execution du script fix_permissions.sh :

```
./fix_permissions.sh <directory_name>
```

Arborescence test_dir.txt

Les fichiers ne réapparaisse plus lors de la 2e execution de script => fonctionne.

5 TASK5

Ici j'ai juste recopié le script d'avant sous le nom fix_group_permissions.sh. Puis changer ce dernier pour qu'il affiche les worlds writable et les group writable (sauf ceux du groupe personel) sous deux liste séparé puis le demande de fix se fait pour tous les fichiers.

J'ai aussi changer l'arborescence en conséquence test_dir_task5.txt

Les fichiers ne réapparaisse plus lors de la 2e execution de script => fonctionne.