BATAILLION Alice
MOREAU Marianne
4 ETI

TP 4 - Utilisateurs, groupes et permissions

Exercice 1. Gestion des utilisateurs et des groupes

1. Commencez par créer deux groupes groupe1 et groupe2

`sudo addgroup groupe1` Pour vérifier que le groupe a bien été créé :
`cut -d: -f1 /etc/group`

2. Créez ensuite 4 utilisateurs u1, u2, u3, u4 avec la commande useradd, en demandant la création de leur dossier personnel et avec bash pour shell. `sudo useradd -m u1`
L'option -m permet de créer un dossier personnel à l'utilisateur.
Pour vérifier que l'utilisateur a bien été créé :
`cut -d: -f1 /etc/passwd`

3. Ajoutez les utilisateurs dans les groupes créés :
- u1, u2, u4 dans groupe1
- u2, u3, u4 dans groupe2

`sudo gpasswd -a u1 groupe1` ajoute u1 au groupe1
Pour consulter les groupes auxquels appartient un utilisateur : `groups u1`

4. Donnez deux moyens d'afficher les membres de groupe2

`sudo grep groupe2 /etc/group | cut -d: -f4` : va chercher la ligne du groupe recherché dans le fichier des groupes, et sélectionne la colonne qui correspond aux noms des utilisateurs appartenant à ce groupe.
`sudo grep groupe2 /etc/gshadow | cut -d: -f4` : de même dans le fichier gshadow qui contient également ces informations.

5. Faites de groupe1 le groupe propriétaire de /home/u1 et /home/u2 et de groupe2 le groupe propriétaire de /home/u3 et /home/u4

`sudo chown -R u1:groupe1 /home/u1` Dans le dossier /home, faire `ls -l` pour vérifier que le groupe propriétaire de chaque dossier personnel d'utilisateur est bien celui attribué.

6. Remplacez le groupe primaire des utilisateurs :
- groupe1 pour u1 et u2
- groupe2 pour u3 et u4

`sudo usermod -g groupe1 u1` remplace le groupe primaire de l'utilisateur u1 (qui était par défaut le groupe u1) en groupe1. On peut vérifier que le groupe primaire a bien été modifié en faisant : `groups u1` où groupe1 ets le premier groupe à apparaître.

7. Créez deux répertoires /home/groupe1 et /home/groupe2 pour le contenu commun aux groupes, et mettez en place les permissions permettant aux membres de chaque groupe d'écrire dans le dossier associé.

`sudo mkdir /home/groupe1`

`sudo chown -R root:groupe1 /home/groupe1` On indique quels utilisateurs ont droit sur le dossier groupe1 : il s'agit de l'utilisateur root et du groupe d'utilisateurs groupe1.

`sudo chmod 775 /home/groupe1` On change les droits des différents utilisateurs : le groupe groupe1 peut maintenant écrire dans le dossier.

8. Comment faire pour que, dans ces dossiers, seul le propriétaire d'un fichier ait le droit de renommer ou supprimer ce fichier ?

`chmod +t /home/groupe1` permet d'activer le sticky bit. Ce bit indique que dans ce dossier, seuls le propriétaire d'un fichier, le propriétaire du dossier ou root ont le droit de renommer ou supprimer ce fichier.

9. Pouvez-vous vous connecter en tant que u1 ? Pourquoi ?

Non, nous n'avons pas attribué de mot de passe au compte de l'utilisateur u1. Le compte est donc inactif jusqu'à attribution d'un mot de passe.

10. Activez le compte de l'utilisateur u1 et vérifiez que vous pouvez désormais vous connecter avec son compte.

`sudo passwd u1` permet d'attribuer un mot de passe au compte de l'utilisateur u1. Nous avons attribué le mot de passe : u1.
Avec `su u1` on se connecte en tant que u1.

11. Quels sont l'uid et le gid de u1 ?

Avec `id` on obtient uid=1001 et gidup=1001.

12. Quel utilisateur a pour uid 1003 ?

`sudo cut -d: -f1-3 /etc/passwd | grep 1003 | cut -d: -f1` : permet de récupérer la ligne qui comprend un uid de 1003 et renvoie la première colonne de cette ligne (qui correspond au nom d'utilisateur). up L'utilisateur u3 a pour uid 1003. up **13. Quel est l'id du groupe groupe1 ?**

`sudo grep groupe1 /etc/group | cut -d: -f3` : on sélectionne la ligne qui correspond au groupe1 et on affiche la colonne qui correspond au gid.

Le groupe1 a pour gid 1001.

14. Quel groupe a pour gid 1002 ? (⚠ Rien n'empêche d'avoir un groupe dont le nom serait 1002...)

`awk -F":" '$3=="1002"' /etc/group | cut -d: -f1` : awk permet de trouver la ligne où la 3ème colonne (celle du gid) est égale à 1002; on sélectionne ensuite la première colonne de cette ligne qui est le nom du groupe correspondant à ce gid.

Le groupe2 a pour gid 1002.

15. Retirez l'utilisateur u3 du groupe groupe2. Que se passe-t-il ? Expliquez.

`sudo gpasswd -d u3 groupe2` retire u3 du groupe groupe2. Lorsque l'on regarde à quels groupes appartient u3 (`groups u3`), groupe2 reste son groupe primaire car un utilisateur ne peut pas ne pas avoir de groupe. Pourtant lorsque l'on vérifie les membres du groupe2 (`nano /etc/group`), u3 a bien été supprimé. Ainsi, si on change le groupe primaire de u3, l'utilisateur ne fera plus parti du groupe2.

16. Modifiez le compte de u4 de sorte que :
— il expire au 1 er juin 2020 : -E — il faut changer de mot de passe avant 90 jours : -M — il faut attendre 5 jours pour modifier un mot de passe : -m — l'utilisateur est averti 14 jours avant l'expiration de son mot de passe : -W — le compte sera bloqué 30 jours après expiration du mot de passe : -I

- `chage -E 2020-06-01 u4` permet de changer la date d'expiration du compte
- `chage -M 90 u4`
- `chage -m 5 u4`
- `chage -W 14 u4`
- `chage -I 30 u4`

17. Quel est l'interpréteur de commandes (Shell) de l'utilisateur root ?

Avec `grep root /etc/passwd | cut -d: -f7`, on obtient l'interprêteur de commande utilisé par l'utilisateur root : /bin/bash.

18. à quoi correspond l'utilisateur nobody ?

L'utilisateur nobody est un utilisateur qui ne possède aucun fichier, qui n'est dans aucun groupe et qui possède les droits de base réservés aux utilisateurs, sans aucun privilège accessible via des groupes.

19. Par défaut, combien de temps la commande sudo conserve-t-elle votre mot de passe en mémoire ?

Le mots de passe est mémorisé par défaut pour une durée de 15 minutes.

Quelle commande permet de forcer sudo à oublier votre mot de passe ?

`sudo -k`

Exercice 2. Gestion des permissions

1. Dans votre $HOME, créez un dossier test, et dans ce dossier un fichier fichier contenant quelques lignes de texte. Quels sont les droits sur test et fichier ?

Dans Marialice, on exécute :

```
 mkdir test
touch test/fichier
```

Pour voir les droits sur un éléments, on fait `ls -l` Pour test : drwxrwx-wx : d indique qu'il s'agit d'un répertoire; l'utilisateur propriétaire (nous) et le groupe propriétaire ont droit de lecture,d'écriture et d'exécution; les autres ont droit d'écriture et d'exécution. Pour fichier : -rw-rw--w- : l'utilisateur propriétaire et le groupe propriétaire ont droit de lecture et d'écriture; les autres ont droit d'écriture.

2. Retirez tous les droits sur ce fichier (même pour vous), puis essayez de le modifier et de l'afficher en tant que root. Conclusion ?

`sudo chmod 000 test/fichier` permet de supprimer tous les droits sur fichier. En utilisant `sudo`, on peut toujours avoir accès au fichier et le modifier.

3. Redonnez vous les droits en écriture et exécution sur fichier puis exécutez la commande echo "echo Hello" > fichier. On a vu lors des TP précédents que cette commande remplace le contenu d'un fichier s'il existe déjà. Que peut-on dire au sujet des droits ?

`sudo chmod 300 test/fichier` permet de se donner les droits d'écriture et d'exécution. Lorsque l'on fait : `echo "echo Hello" > fichier`, on écrit dans le fichier.
Cependant, on ne peut pas vérifier que le fichier a bien été modifié car nous n'avons pas les droits de lecture. Il faut utiliser sudo et root pour vérifier le contenu de fichier.

4. Essayez d'exécuter le fichier. Est-ce que cela fonctionne ? Et avec sudo ? Expliquez.

L'exécution grâce à `./fichier` ne fonctionne pas. Afin de pouvoir exécuter un fichier, l'utilisateur a besoin des droits de lecture. En effet, il faut pouvoir lire le contenu d'un fichier pour l'exécuter. Ici, nous n'avons pas les droits de lecture donc nous ne pouvons pas exécuter le fichier.
On réussit l'exécution graĉe à `sudo ./fichier` qui renvoie bien Hello. Le super-utilisateur root a tous les droits, il passe au-dessus des restrictions de droits. Il peut donc lire le fichier et ainsi l'exécuter.

5. Placez-vous dans le répertoire test, et retirez-vous le droit en lecture pour ce répertoire. Listez le contenu du répertoire, puis exécutez ou affichez le contenu du fichier fichier. Qu'en déduisez-vous ? Rétablissez le droit en lecture sur test

`sudo chmod u-r /home/marialice/test` permet de se retirer les droits de lecture. Nous sommes l'utilisateur propriétaire (u) auquel nous enlevons (-) les droits de lecture (r).
Un `ls -l` dans home/test est inefficace : nous n'avons pas le droit de lecture sur ce dossier. Nous ne pouvons donc pas voir les fichiers présents dans le dossier. Cependant, on peut exécuter et lire le fichier fichier car les droits de lecture du fichier sont bien actifs eux.(On s'est donné les droits de lecture sur le fichier fichier.)
`sudo chmod u+r /home/marialice/test` permet de se redonner les droits de lecture.

6. Créez dans test un fichier nouveau ainsi qu'un répertoire sstest. Retirez au fichier nouveau et au répertoire test le droit en écriture. Tentez de modifier le fichier nouveau. Rétablissez ensuite le droit en écriture au répertoire test. Tentez de modifier le fichier nouveau, puis de le supprimer. Que pouvez vous déduire de toutes ces manipulations ?

Dans test :

```
 mkdir sstest
touch nouveau
sudo chmod a-w nouveau
sudo chmod a-w test
```

Le `a` correspond à all (tous les utilisateurs potentiels) Avec `nano nouveau`, on ne peut pas écrire dans le fichier : nous n'avons pas les droits d'écriture dans le dossier test.
On a rétabli les droits d'écriture du dossier test. On ne peut toujours pas modifier le fichier nouveau : on n'a pas les droits d'écriture sur le fichier nouveau. Cependant, on peut le supprimer avec `rm nouveau` car on a les droits sur le dossier test. Cette suppression est sécurisée : on nous demande une confirmation car le dossier est protégé.
Les droits du dossier prévalent lors d'une création ou d'une suppression d'un élément en son sein. Les droits du fichier prévalent sur le contenu du fichier lui-même.

7. Positionnez vous dans votre répertoire personnel, puis retirez le droit en exécution du répertoire test. Tentez de créer, supprimer, ou modifier un fichier dans le répertoire test, de vous y déplacer, d'en lister le contenu, etc…Qu'en déduisez vous quant au sens du droit en exécution pour les répertoires ?

`sudo chmod a-x test` permet de retirer les droits d'exécution au dossier test. Pour les répertoires, le droit d'exécution correspond au droit de traverser le répertoire. Cela comprend: le droit de se déplacer dans le répertoire et dans ses sous-dossiers, le droit de créer un sous-dossier ou un fichier, le droit d'accéder au contenu du fichier du répertoire. Globalement, cel empêche toute interraction qui nécessite d epénétrer dans le répertoire.
On a tout de même la possibilité d'accéder aux noms des éléments contenus dans le répertoire car on a toujours le droit de lecture. On ne peut pas accéder aux noms au-delà d'un niveau d'arborescence.

8. Rétablissez le droit en exécution du répertoire test. Positionnez vous dans ce répertoire et retirez lui à nouveau le droit d'exécution. Essayez de créer, supprimer et modifier un fichier dans le répertoire test, de vous déplacer dans ssrep, de lister son contenu. Qu'en concluez-vous quant à l'influence des droits que l'on possède sur le répertoire courant ? Peut-on retourner dans le répertoire parent avec "cd .." ? Pouvez-vous donner une explication ?

Nous ne pouvons réaliser aucune des commandes de la consigne dans le dossier test. Contrairement à la question 7, nous ne pouvons plus lister le contenu du repertoire. En effet, `ls` étant un executable, il suit les droits du dossier depuis lequel il est appelé.
Nous pouvons retourner au dossier parent avec la commande `cd ..`, cela est possible car cd est une commande interne à bash et le dossier parent n'est pas impacté par les droits sur un dossier qui lui est inférieur dans l'arborescence. Une fois remonté à l'étage supérieur, nous ne pouvons plus redescendre.
Finalement, retirer les droits depuis le répertoire concerné est encore plus contraignant que de les retirer depuis un dossier parent.

9. Rétablissez le droit en exécution du répertoire test. Attribuez au fichier fichier les droits suffisants pour qu'une autre personne de votre groupe puisse y accéder en lecture, mais pas en écriture.

```
chmod +x test
chmod g+r-w test/fichier
```

10. Définissez un umask très restrictif qui interdit à quiconque à part vous l'accès en lecture ou en écriture, ainsi que la traversée de vos répertoires. Testez sur un nouveau fichier et un nouveau répertoire.

```
umask 077
touch le_fichier
chmod u+x le_fichier
mkdir le_dossier
```

11. Définissez un umask très permissif qui autorise tout le monde à lire vos fichiers et traverser vos répertoires, mais n'autorise que vous à écrire. Testez sur un nouveau fichier et un nouveau répertoire.

```
umask 022
touch le_fichier_2
chmod a+x le_fichier_2
mkdir le_dossier_2
```

12. Définissez un umask équilibré qui vous autorise un accès complet et autorise un accès en lecture aux membres de votre groupe. Testez sur un nouveau fichier et un nouveau répertoire.

```
umask 037
touch le_fichier_3
chmod u+x le_fichier_3
mkdir le_dossier_3
```

13. Transcrivez les commandes suivantes de la notation classique à la notation octale ou vice-versa (vous pourrez vous aider de la commande stat pour valider vos réponses) :
- chmod u=rx,g=wx,o=r fic : `chmod 534 fic`
- chmod uo+w,g-rx fic en sachant que les droits initiaux de fic sont r--r-x--- : : `chmod 602 fic`
- chmod 653 fic en sachant que les droits initiaux de fic sont 711 : `ch` mod u-x g+r o+w fic `- chmod u+x,g=w,o-r fic en sachant que les droits initiaux de fic sont r--r-x--- :` chmod 520 fic`

14. Affichez les droits sur le programme passwd. Que remarquez-vous ? En affichant les droits du fichier /etc/passwd, pouvez-vous justifier les permissions sur le programme passwd ?

Frâce à `ls -l /usr/bin/passwd` on obtient les autorisations liées au programme passwd.
On obtient : -rws r-x r-x : le groupe propriétaire et les autres ont le droit de lire et d'exécuter ce programme. L'utilisateur propriétaire (root) a le droit de lire, écrire et exécuter le programme. Le s indique que le programme peut être utilisé avec les droits propriétaires.
Lorsque l'on exécute la commande `ls -l /etc/passwd` , on obtient : -rw-r--r--. Ce sont les droits d'accès au fichier passwd.
La commande passwd a besoin d'écrire dans le fichier passwd. Ce fichier n'est accessible en écriture que par le root. La commande passwd est possédée par le root et elle a besoin d'un droit d'écriture de l'utilisateur propriétaire pour accéder au fichier passwd. Ainsi, elle pourra modifier le contenu du fichier passwd, ce qui est nécessaire à la commande passwd dans le cas d'une utlisation en root. Les autres utilisateurs et le groupe propriétaire n'ont accès qu'en lecture au fichier passwd car ils ont besoin d'exécuter la commande passwd uniquement et pas de modifier le fichier passwd.