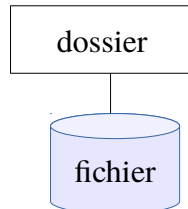


TD n°9

1) Manipulations à faire en solo

Placez-vous dans votre compte linux, par exemple dans système/td10. Créez un dossier appelé `dossier`, descendez dedans et créez-y un fichier appelé `fichier`. Mettez au moins une ligne dans le fichier, afin qu'on puisse savoir si on a accès au contenu ou pas.



On peut afficher l'état de ces éléments par `ls -ld . fichier`

Puis faire les tests suivants. Notez vos observations et tout ce qui peut vous étonner par rapport au « bon sens ».

- Retirez tous les droits sur le fichier (`chmod ugo-rwx fichier`) et retirez seulement W sur le dossier (`chmod u-w .` ou `chmod u=rx .`), (laissez R et X sur le dossier), vérifiez l'état des protections avec la commande `ls` précédente. Tentez d'afficher le contenu du fichier à l'écran, tentez de le copier, tentez de le renommer, tentez de le supprimer (recréez-le s'il est supprimé). Quels sont les droits qui sont sollicités dans chacun des cas ?
- Rajoutez tous les droits sur le fichier (`chmod u=rwx fichier`) mais toujours pas W sur le dossier, vérifiez que vous pouvez l'afficher. Tentez de le copier, de le renommer, de le supprimer, quels sont les droits nécessaires et qui manquent pour pouvoir le faire ? Rajoutez le droit W sur le dossier et tentez de renommer le fichier puis de le supprimer (recréez-le s'il disparaît).
- Vérifiez qu'il y a les droits RWX sur le dossier. Enlevez le droit W sur le fichier puis tentez de faire `echo suite >> fichier`, puis `echo vide > fichier` : qu'est-ce que c'est censé faire et que se passe-t-il ? Essayez `echo truc > autre`, puis enlevez le droit W sur le répertoire courant puis faites `echo bidule >> autre` et `echo dibule >> essai` Est-ce normal ? Remettez W sur le dossier.
- Enlevez le droit X sur le répertoire courant puis tentez d'afficher le fichier, tentez de lister le dossier courant. Remettez le droit X sur le répertoire et retentez les opérations. Que peut-on en conclure ? En fait, il faudrait aussi refaire les manipulations en étant au dessus de dossier.
- Enlevez seulement le droit R sur le fichier (`chmod u=w fichier`), en laissant RWX sur le dossier, tentez de copier le fichier : `cp fichier copie`. Tentez de renommer le fichier : `mv fichier chifrier`. Remettez alors le droit R sur `chifrier` et retentez un copie vers `fichier`. Conclusion ?
- Enlevez le droit W sur le répertoire courant et tentez de copier le fichier comme précédemment, tentez maintenant de copier le fichier dans le répertoire parent : `cp fichier ..` Enfin tentez de renommer le fichier comme précédemment. Qu'est-ce qui empêche de copier le fichier ?

2) Promenade dans le système

On vous invite maintenant à examiner quelques fichiers emblématiques du système. Vous devrez regarder les protections qui sont mises dessus (droit, propriétaire et groupe) et en déduire ce qui vous est permis de faire.

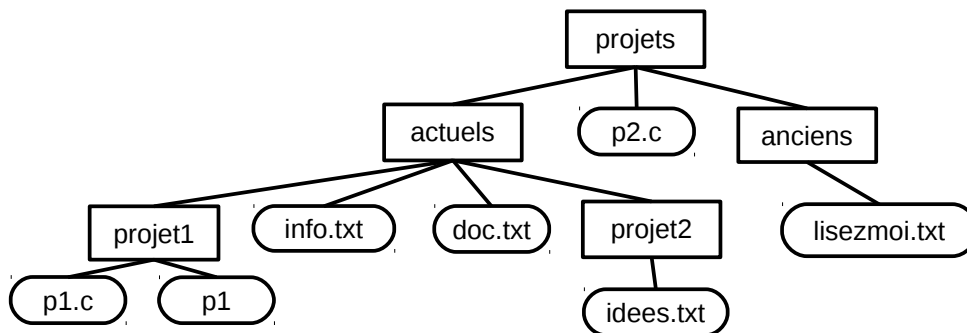
- `/etc/passwd`
- `/etc/shadow`
- `/etc/ssh/*` (listez les propriétés des fichiers, lesquels pouvez-vous afficher, lesquels ne pouvez-vous pas?)
- `/tmp` : voici un dossier bien étrange : regardez ses droits, vous les avez tous. Vous pouvez y créer des fichiers et les supprimer, mais pouvez-vous supprimer les fichiers des autres

utilisateurs? Notez le t à la place du dernier x, et lisez la documentation de chmod, au paragraphe sur le BIT STICKY (drapeau de suppression restreinte).

- /usr/bin/passwd : c'est une commande (passwd, pour changer son mot de passe). Faites `ls -l` : c'est quoi ce s à la place du x ? (doc de chmod, § sur le SETUID, ça veut dire : vous prenez l'identité du propriétaire de la commande passwd (root) juste le temps de l'exécution, ça vous donne alors le droit de modifier le fichier /etc/shadow, sinon comment serait-il possible de changer de mot de passe si vous n'avez aucun droit).

3) Droits des commandes sur les répertoires

Soit cette arborescence de fichiers. On s'intéresse aux accès RWX que font les commandes sur les différents éléments. On se situe initialement dans le répertoire projets et on ne considère pas les répertoires situés plus haut.



Donner les listes d'accès, c'est-à-dire les affirmations « telle combinaison rwx sur tel et tel fichier ou répertoire » pour les commandes suivantes en expliquant pourquoi. Par exemple, pour faire `vi p1.c`, il faut `rw` sur `p1.c` et `x` sur `projets`, `actuels` et `projet1`. Les questions sont indépendantes.

- | | |
|--|--|
| a) <code>ls</code> | f) <code>rm actuels/*.txt</code> |
| b) <code>more actuels/doc.txt</code> | g) <code>mv anciens/lisezmoi.txt actuels</code> |
| c) <code>cd actuels/projet1</code> | h) <code>mv actuels/projet1 anciens</code> |
| d) <code>ls actuels/projet1</code> | i) <code>ls -l actuels > actuels/doc.txt</code> |
| e) <code>cp actuels/doc.txt anciens</code> | j) <code>cc p2.c -o p2</code> |

4) Configuration des droits

On reprend le même arbre de fichiers que précédemment. On a différents comptes utilisateurs auxquels on veut autoriser ou interdire certaines opérations. Le but de l'exercice est de réfléchir à une attribution des propriétaires et des droits sur les répertoires pour réaliser ces vœux.

Voici les noms et les groupes des utilisateurs :

Jean	tech
Jacques	tech
Arnaud	biz
Victor	guest

- Tout le monde peut entrer dans le répertoire `actuels` et lister les fichiers. Seul Jacques peut créer un autre sous-répertoire.
- Jean peut tout faire dans `projet1`, Jacques ne peut rien y faire, Arnaud peut y aller et lister les fichiers mais pas en créer, Victor ne peut rien y faire.
- Le répertoire `projet2` peut être listé par Jacques, Arnaud et Jean. Tout le monde peut y aller, mais seul Arnaud peut effacer le fichier `infos`.
- Arnaud et Victor peuvent aller dans `anciens` mais sans pouvoir lister de fichiers. Jacques et Jean peuvent copier des fichiers dedans. Eux seuls peuvent lister les fichiers.