

Autorisations relatives aux fichiers sous Linux

Description du projet

L'équipe de recherche de l'entreprise pour laquelle je travaille doit mettre à jour les autorisations de certains fichiers et répertoires dans le répertoire `projects`. Les autorisations actuelles ne reflètent pas le niveau d'autorisation qui devrait être accordé. La vérification et la mise à jour de ces autorisations contribueront à sécuriser le système auquel ils accèdent. Pour mener à bien cette mission, j'ai effectué les tâches suivantes :

Vérifier les détails des fichiers et des répertoires

Le code suivant montre comment j'ai utilisé les commandes Linux pour identifier les autorisations existantes relatives à un répertoire spécifique du système de fichiers.

```
(steph@kali)-[~/projects]
$ ls -la
total 1268
drwxr-xr-x  3 steph steph    4096 May 26 19:32 .
drwxr-xr-x 41 steph steph 1286144 May 26 19:30 ..
-rw--w----  1 steph steph      0 May 26 19:32 .project_x.txt
drwx--x---  2 steph steph    4096 May 26 19:31 drafts
-rw-rw-rw-  1 steph steph      0 May 26 19:32 project_k.txt
-rw-r-----  1 steph steph      0 May 26 19:32 project_m.txt
-rw-rw-r--  1 steph steph      0 May 26 19:32 project_r.txt
-rw-rw-r--  1 steph steph      0 May 26 19:32 project_t.txt
```

La première ligne de la capture d'écran affiche la commande que j'ai saisie, et les autres lignes affichent la sortie. Le code permet d'obtenir l'ensemble du contenu du répertoire `projects`. J'ai utilisé la commande `ls` avec l'option `-la` pour afficher une liste détaillée des fichiers, y compris les fichiers masqués. La sortie de ma commande indique la présence des éléments suivants : un répertoire intitulé `drafts`, un fichier masqué intitulé `.project_x.txt` ainsi que cinq autres fichiers « project ». La chaîne de 10 caractères de la première colonne correspond aux autorisations définies pour chaque fichier ou répertoire.

Décrire la chaîne des autorisations

La chaîne de 10 caractères peut être décomposée afin d'identifier qui est autorisé à accéder au fichier et quelles sont ses autorisations spécifiques. Les caractères et leur correspondance sont les suivants :

- **1er caractère** : représenté par un `d` ou un tiret (`-`), ce caractère indique le type de fichier. Le caractère `d` représente un répertoire, tandis qu'un tiret (`-`) indique un fichier standard.

- **2e au 4e caractère** : ces caractères indiquent les autorisations de lecture (**r**), d'écriture (**w**) et d'exécution (**x**) de l'utilisateur. Si l'un de ces caractères est remplacé par un tiret (**-**), cela signifie que cette autorisation n'est pas accordée à l'utilisateur.
- **5e au 7e caractère** : ces caractères indiquent les autorisations de lecture (**r**), d'écriture (**w**) et d'exécution (**x**) du groupe. Si l'un de ces caractères est remplacé par un tiret (**-**), cela signifie que cette autorisation n'est pas accordée au groupe.
- **8e au 10e caractère** : ces caractères indiquent les autorisations de lecture (**r**), d'écriture (**w**) et d'exécution (**x**) des autres utilisateurs. Ce type de propriété comprend tous les autres utilisateurs du système, à l'exception de l'utilisateur principal et du groupe. Si l'un de ces caractères est remplacé par un tiret (**-**), cela signifie que cette autorisation n'est pas accordée aux autres utilisateurs.

Par exemple, les autorisations du fichier `project_t.txt` sont `-rw-rw-r--`. Le premier caractère est un trait d'union (**-**) : cela indique que `project_t.txt` est un fichier et non un répertoire. Les deuxième, cinquième et huitième caractères sont tous **r** : l'utilisateur principal, le groupe et les autres utilisateurs disposent chacun d'autorisations de lecture. Les troisième et sixième caractères sont **w** : seuls l'utilisateur et le groupe disposent d'autorisations d'écriture. Personne ne dispose d'autorisations d'exécution pour le fichier `project_t.txt`.

Modifier les autorisations de fichiers

L'entreprise a établi que les autres utilisateurs ne devaient disposer d'aucune autorisation d'écriture relative à leurs fichiers. Pour ce faire, j'ai consulté les autorisations de fichiers que j'avais précédemment renvoyées. J'ai défini que les autorisations d'écriture relatives aux autres utilisateurs devaient être supprimées pour le fichier `project_k.txt`.

Le code suivant montre comment j'ai utilisé les commandes Linux pour y parvenir :

```
(steph@kali)-[~/projects]
$ ls -la
total 1268
drwxr-xr-x  3 steph steph    4096 May 26 19:32 .
drwxr-xr-x 41 steph steph 1286144 May 26 19:30 ..
-rw--w----- 1 steph steph      0 May 26 19:32 .project_x.txt
drwx--x---  2 steph steph    4096 May 26 19:31 drafts
-rw-rw-r--  1 steph steph      0 May 26 19:32 project_k.txt
-rw-r----- 1 steph steph      0 May 26 19:32 project_m.txt
-rw-rw-r--  1 steph steph      0 May 26 19:32 project_r.txt
-rw-rw-r--  1 steph steph      0 May 26 19:32 project_t.txt
```

Les deux premières lignes de la capture d'écran affichent les commandes que j'ai saisies, et les autres lignes affichent la sortie de la deuxième commande. La commande `chmod` modifie les autorisations relatives aux fichiers et aux répertoires. Le premier argument indique les autorisations à modifier et le deuxième argument précise le fichier ou le répertoire. Dans cet exemple, j'ai supprimé les

autorisations d'écriture des autres utilisateurs relatives au fichier `project_k.txt`. Ensuite, j'ai utilisé la commande `ls -la` pour passer en revue les mises à jour que j'ai effectuées.

Modifier les autorisations d'un fichier masqué

L'équipe de recherche de l'entreprise pour laquelle je travaille a récemment archivé le fichier `project_x.txt`. Elle souhaite que personne ne dispose d'autorisations d'écriture pour ce projet, mais l'utilisateur et le groupe doivent disposer d'autorisations de lecture.

Le code suivant montre comment j'ai utilisé les commandes Linux pour modifier les autorisations :

```
(steph@kali)-[~/projects]
$ ls -la
total 1268
drwxr-xr-x  3 steph steph    4096 May 26 19:32 .
drwxr-xr-x 41 steph steph 1286144 May 26 19:30 ..
-r--r----- 1 steph steph      0 May 26 19:32 .project_x.txt
drwx--x---  2 steph steph    4096 May 26 19:31 drafts
-rw-rw-rw-  1 steph steph      0 May 26 19:32 project_k.txt
-rw-r----- 1 steph steph      0 May 26 19:32 project_m.txt
-rw-rw-r--  1 steph steph      0 May 26 19:32 project_r.txt
-rw-rw-r--  1 steph steph      0 May 26 19:32 project_t.txt
```

Les deux premières lignes de la capture d'écran affichent les commandes que j'ai saisies, et les autres lignes affichent la sortie de la deuxième commande. Je sais que le fichier `.project_x.txt` est un fichier masqué, car son nom commence par un point (`.`). Dans cet exemple, j'ai supprimé les autorisations d'écriture de l'utilisateur et du groupe, et j'ai ajouté des autorisations de lecture au groupe. J'ai supprimé les autorisations d'écriture de l'utilisateur grâce à la commande `u-w`. Puis, j'ai supprimé les autorisations d'écriture du groupe à l'aide de la commande `g-w` et ajouté des autorisations de lecture au groupe avec la commande `g+r`.

Modifier les autorisations des répertoires

L'entreprise pour laquelle je travaille souhaite que seul l'utilisateur `researcher2` ait accès au répertoire `drafts` et à son contenu. Cela signifie qu'aucune autre personne que l'utilisateur `researcher2` ne doit disposer d'autorisations d'exécution.

Le code suivant montre comment j'ai utilisé les commandes Linux pour modifier les autorisations :

```
(steph@kali)-[~/projects]
$ ls -la
total 1268
drwxr-xr-x  3 steph steph    4096 May 26 19:32 .
drwxr-xr-x 41 steph steph 1286144 May 26 19:30 ..
-r--r----- 1 steph steph      0 May 26 19:32 .project_x.txt
drwx----- 2 steph steph    4096 May 26 19:31 drafts
-rw-rw-r-- 1 steph steph      0 May 26 19:32 project_k.txt
-rw-r----- 1 steph steph      0 May 26 19:32 project_m.txt
-rw-rw-r-- 1 steph steph      0 May 26 19:32 project_r.txt
-rw-rw-r-- 1 steph steph      0 May 26 19:32 project_t.txt
```

Les deux premières lignes de la capture d'écran affichent les commandes que j'ai saisies, et les autres lignes affichent la sortie de la deuxième commande. J'ai préalablement défini que le groupe disposait d'autorisations d'exécution. J'ai donc utilisé la commande `chmod` pour les supprimer. L'utilisateur `researcher2` disposait déjà d'autorisations d'exécution. Il n'a donc pas été nécessaire de les ajouter.

Synthèse

J'ai modifié plusieurs autorisations afin qu'elles correspondent au niveau d'autorisation relatif aux fichiers et aux répertoires du répertoire `projects` souhaité par mon entreprise. La première étape a consisté à utiliser la commande `ls -la` pour consulter les autorisations du répertoire. Cela m'a permis de prendre des décisions éclairées lors des étapes suivantes. J'ai ensuite utilisé la commande `chmod` à plusieurs reprises pour modifier les autorisations relatives aux fichiers et aux répertoires.