

Permissões de arquivo no Linux

Descrição do projeto

A equipe de pesquisa da minha organização precisa atualizar as permissões de arquivo para determinados arquivos e diretórios no `diretório de projetos`. As permissões atualmente não refletem o nível de autorização que deve ser concedido. Verificar e atualizar essas permissões ajudará a manter o sistema seguro. Para concluir esta tarefa, executei as seguintes tarefas:

Verifique os detalhes do arquivo e do diretório

O código a seguir demonstra como usei comandos do Linux para determinar as permissões existentes definidas para um diretório específico no sistema de arquivos.

```
researcher2@5d738f0f927b:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Dec  2 15:27 .
drwxr-xr-x 3 researcher2 research_team 4096 Dec  2 15:27 ..
-rw--w---- 1 researcher2 research_team  46 Dec  2 15:27 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Dec  2 15:27 drafts
-rw-rw-rw- 1 researcher2 research_team  46 Dec  2 15:27 project_k.txt
-rw-r----- 1 researcher2 research_team  46 Dec  2 15:27 project_m.txt
-rw-rw-r-- 1 researcher2 research_team  46 Dec  2 15:27 project_r.txt
-rw-rw-r-- 1 researcher2 research_team  46 Dec  2 15:27 project_t.txt
researcher2@5d738f0f927b:~/projects$
```

A primeira linha da captura de tela exibe o comando que inseri e as outras linhas exibem a saída. O código lista todo o conteúdo do diretório de `projetos`. Usei o comando `ls` com a opção `-la` para exibir uma lista detalhada do conteúdo do arquivo que também retornou arquivos ocultos. A saída do meu comando indica que há um diretório chamado `rascunhos`, um arquivo oculto chamado `project_x.txt` e cinco outros arquivos de projeto. A cadeia de caracteres de 10 caracteres na primeira coluna representa as permissões definidas em cada arquivo ou diretório.

Descrever a cadeia de caracteres de permissões

A cadeia de caracteres de 10 caracteres pode ser desconstruída para determinar quem está autorizado a acessar o arquivo e suas permissões específicas. Os personagens e o que eles representam são os seguintes:

- **1º caractere:** Este caractere é um `d` ou hífen (`-`) e indica o tipo de arquivo. Se for um `d`, é um diretório. Se for um hífen (`-`), é um arquivo normal.

- **2º a 4º caracteres:** esses caracteres indicam as permissões de leitura (**r**), gravação (**w**) e execução (**x**) para o usuário. Quando um desses caracteres é um hífen (**-**), isso indica que essa permissão não é concedida ao usuário.
- **5º a 7º caracteres:** esses caracteres indicam as permissões de leitura (**r**), gravação (**w**) e execução (**x**) para o grupo. Quando um desses caracteres é um hífen (**-**), isso indica que essa permissão não é concedida para o grupo.
- **8º a 10º caracteres:** esses caracteres indicam as permissões de leitura (**r**), gravação (**w**) e execução (**x**) para outros. Esse tipo de proprietário consiste em todos os outros usuários no sistema, além do usuário e do grupo. Quando um desses caracteres é um hífen (**-**), isso indica que essa permissão não é concedida para outro.

Por exemplo, as permissões de arquivo para `project_t.txt` são `-rw-rw-r--`. Como o primeiro caractere é um hífen (**-**), isso indica que `project_t.txt` é um arquivo, não um diretório. O segundo, quinto e oitavo caracteres são todos **r**, o que indica que usuário, grupo e outros têm permissões de leitura. O terceiro e o sexto caracteres são **w**, o que indica que apenas o usuário e o grupo têm permissões de gravação. Ninguém tem permissões de execução para `project_t.txt`.

Alterar permissões de arquivo

A organização determinou que outros não deveriam ter acesso de gravação a nenhum de seus arquivos. Para cumprir isso, referi-me às permissões de arquivo que retornei anteriormente. Determinei `project_k.txt` deve ter o acesso de gravação removido para outro.

O código a seguir demonstra como usei comandos do Linux para fazer isso:

```
researcher2@5d738f0f927b:~/projects$ chmod o-w project_k.txt
researcher2@5d738f0f927b:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Dec  2 15:27 .
drwxr-xr-x 3 researcher2 research_team 4096 Dec  2 15:27 ..
-rw--w---- 1 researcher2 research_team   46 Dec  2 15:27 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Dec  2 15:27 drafts
-rw-rw-r-- 1 researcher2 research_team   46 Dec  2 15:27 project_k.txt
-rw-r----- 1 researcher2 research_team   46 Dec  2 15:27 project_m.txt
-rw-rw-r-- 1 researcher2 research_team   46 Dec  2 15:27 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 Dec  2 15:27 project_t.txt
researcher2@5d738f0f927b:~/projects$
```

As duas primeiras linhas da captura de tela exibem os comandos que inseri e as outras linhas exibem a saída do segundo comando. O comando `chmod` altera as permissões em arquivos e diretórios. O primeiro argumento indica quais permissões devem ser alteradas e o segundo argumento especifica o arquivo ou diretório. Neste exemplo, removi as permissões de gravação de outros para o arquivo `project_k.txt`. Depois disso, usei `ls -la` para revisar as atualizações que fiz.

Alterar permissões de arquivo em um arquivo oculto

A equipe de pesquisa da minha organização arquivou recentemente `project_x.txt`. Eles não querem que ninguém tenha acesso de gravação a este projeto, mas o usuário e o grupo devem ter acesso de leitura.

O código a seguir demonstra como usei comandos do Linux para alterar as permissões:

```
researcher2@3213bbc1d047:~/projects$ chmod u-w,g-w,g+r .project_x.txt
researcher2@3213bbc1d047:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Dec 20 15:36 .
drwxr-xr-x 3 researcher2 research_team 4096 Dec 20 15:36 ..
-r--r----- 1 researcher2 research_team  46 Dec 20 15:36 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Dec 20 15:36 drafts
-rw-rw-rw- 1 researcher2 research_team  46 Dec 20 15:36 project_k.txt
-rw-r----- 1 researcher2 research_team  46 Dec 20 15:36 project_m.txt
-rw-rw-r-- 1 researcher2 research_team  46 Dec 20 15:36 project_r.txt
-rw-rw-r-- 1 researcher2 research_team  46 Dec 20 15:36 project_t.txt
researcher2@3213bbc1d047:~/projects$
```

As duas primeiras linhas da captura de tela exibem os comandos que inseri e as outras linhas exibem a saída do segundo comando. Eu sei `project_x.txt` é um arquivo oculto porque começa com um ponto (.). Neste exemplo, removi as permissões de gravação do usuário e do grupo e adicionei permissões de leitura ao grupo. Removi as permissões de gravação do usuário com `u-w`. Em seguida, removi as permissões de gravação do grupo com `g-w` e adicionei permissões de leitura ao grupo com `g+r`.

Alterar permissões de diretório

Minha organização só quer que o usuário `researcher2` tenha acesso ao diretório de `rascunhos` e seu conteúdo. Isso significa que ninguém além do `researcher2` deve ter permissões de execução.

O código a seguir demonstra como usei comandos do Linux para alterar as permissões:

```
researcher2@5d738f0f927b:~/projects$ chmod g-x drafts
researcher2@5d738f0f927b:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Dec  2 15:27 .
drwxr-xr-x 3 researcher2 research_team 4096 Dec  2 15:27 ..
-r--r----- 1 researcher2 research_team  46 Dec  2 15:27 .project_x.txt
drwx----- 2 researcher2 research_team 4096 Dec  2 15:27 drafts
-rw-rw-r-- 1 researcher2 research_team  46 Dec  2 15:27 project_k.txt
-rw-r----- 1 researcher2 research_team  46 Dec  2 15:27 project_m.txt
-rw-rw-r-- 1 researcher2 research_team  46 Dec  2 15:27 project_r.txt
-rw-rw-r-- 1 researcher2 research_team  46 Dec  2 15:27 project_t.txt
researcher2@5d738f0f927b:~/projects$
```

As duas primeiras linhas da captura de tela exibem os comandos que inseri e as outras linhas exibem a saída do segundo comando. Anteriormente, determinei que o grupo tinha permissões de execução, então usei o comando `chmod` para removê-las. O usuário `researcher2` já tinha permissões de execução, portanto, elas não precisavam ser adicionadas.

Resumo

Alterei várias permissões para corresponder ao nível de autorização que minha organização queria para arquivos e diretórios no `diretório de projetos`. O primeiro passo foi usar `ls -la` para verificar as permissões do diretório. Isso informou minhas decisões nas etapas a seguir. Em seguida, usei o comando `chmod` várias vezes para alterar as permissões em arquivos e diretórios.