

Permissões para arquivos no Linux

Descrição do projeto

Neste projeto, verifico e atualizo as permissões de arquivos e diretórios dentro de uma organização. A habilidade de manipulação dessas permissões através do terminal garante confidencialidade, seguindo sempre o princípio do privilégio mínimo reafirmando conformidade na segurança de organizações. Busquei explicar de forma didática, sem pular etapas. Espero que faça sentido para você!

Verificar detalhes de arquivos e diretórios

O código a seguir demonstra como utilizei um comando, que explicarei mais a fundo como funciona, para exibir as permissões atuais tanto de arquivos comuns como até mesmo arquivos ocultos, para dessa forma analisar como estão essas permissões na organização.

```
researcher2@3b197108bdba:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Jun 24 10:22 .
drwxr-xr-x 3 researcher2 research_team 4096 Jun 24 10:54 ..
-rw--w--- 1 researcher2 research_team  46 Jun 24 10:22 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Jun 24 10:22 drafts
-rw-rw-r-- 1 researcher2 research_team  46 Jun 24 10:22 project_k.txt
-rw----- 1 researcher2 research_team  46 Jun 24 10:22 project_m.txt
-rw-rw-r-- 1 researcher2 research_team  46 Jun 24 10:22 project_r.txt
-rw-rw-r-- 1 researcher2 research_team  46 Jun 24 10:22 project_t.txt
```

Está vendo aquele cifrão (\$) ali em cima? É ali que começa o comando que eu digitei no terminal — no caso, `ls -la`.

Antes de explicar o que ele mostra, vale entender por partes:

- `ls`: lista os arquivos e pastas do diretório atual.
- `-l`: mostra a listagem completa, com permissões, dono, grupo, tamanho e data.
- `-a`: inclui os arquivos ocultos (os que começam com ponto, como `.project_x.txt`).

Quando combino tudo como `ls -la`, ele exibe todos os arquivos, inclusive os ocultos, e com todos os detalhes importantes sobre eles. Cada linha nessa captura de tela é um arquivo com suas respectivas permissões.

Mas onde estão as permissões, e como interpretá-las? Vamos para o próximo tópico para entender isso!

Descrição das permissões demonstradas pela string

Observe na imagem anterior aquela string de 10 caracteres ali do lado esquerdo, cheia de `r`, `w`, `x` e traços (-). Ela mostra quem tem permissão para fazer o quê com aquele arquivo ou diretório.

Funciona assim:

- O 1º caractere indica o tipo:
 - - é um arquivo
 - `d` é um diretório

Os 9 caracteres seguintes vêm em blocos de 3:

```
rw- r-- r--
↑  ↑  ↑
|  |  |
|  |  |___ Permissões para *outros*
|  |___ Permissões para o *grupo*
|___ Permissões para o *usuário*
```

Cada letra tem um **significado em relação a sua permissão ao arquivo**:

- `r` → read (leitura)
- `w` → write (escrita/edição/modificação)

- **x** → execute (execução)
- **-** → sem permissão

Então, quando você vê algo como **-rw-r--r--**, significa:

- É um arquivo
- O usuário pode ler e escrever/editar
- O grupo pode só ler
- Outros usuários também só podem ler

Essa string é essencial para saber como o sistema está controlando o acesso aos arquivos.

Alterar permissões de arquivos

A organização determinou que **outros** não deveriam ter acesso de gravação a nenhum de seus arquivos. Para cumprir com isso, consultei as permissões de arquivo que retornei anteriormente. Determinei que `project_k.txt` precisa ter o acesso de escrita/alteração (write) removido para o tipo proprietário do arquivo “**outros**”.

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$
```

Repare na primeira linha o comando utilizado: “**chmod**”

“Chmod” é utilizado justamente para alterar permissões em arquivos do linux, e ele funcionou ali sendo formatado da seguinte forma:

chmod o-w (nome do arquivo)

Mas o que são essas letras e esse traço novamente? Agora é mais simples do que parece:

Quem recebe a permissão:

- u → user (usuário/dono do arquivo)
- g → group (grupo do usuário)
- o → others (outros usuários)

Tipos de permissão:

- r → read (leitura)
- w → write (escrita/edição/modificação)
- x → execute (execução)

E os símbolos de operações o que fazem?

- + → adiciona a permissão
- - → remove a permissão

Exemplos práticos:

- chmod o-w arquivo.txt → remove escrita de others
- chmod u+x script.sh → dá permissão de execução para o usuário
- chmod g+r projeto.txt → adiciona leitura para o grupo

Após isso, utilizei mais um ls-la só para verificar se minhas alterações de permissão foram concluídas com sucesso.

Alterar permissões de arquivo oculto

A equipe de pesquisa da minha organização arquivou recentemente o .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 desse arquivo oculto:

```
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$
```

Perceba que .project_x.txt é um arquivo oculto por começar com “.”, ou seja, ele nem seria exibido ali caso eu só utilizasse ls -l para verificar os arquivos do sistema, apenas ls-la me permite essa visualização.

As duas primeiras linhas são os comandos que dei, o primeiro para de fato alterar as permissões e o segundo para confirmar a efetivação da mudança. Neste exemplo, removi as permissões de escrita/edição/modificação no arquivo por parte do usuário e do grupo. Além disso, adicionei permissões de leitura ao grupo. Removi as permissões de edição do usuário com u-w. Em seguida, removi as permissões de edição do grupo com g-w e adicionei permissões de leitura ao grupo com g+r.

Mudar as permissões de diretório

Minha organização deseja que apenas o usuário researcher2 tenha acesso ao diretório drafts e seu conteúdo. Isso significa que ninguém além de researcher2 deve ter permissões de execução.

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

```
researcher2@3b197108bdba:~/projects$ chmod g-x drafts
researcher2@3b197108bdba:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Jun 24 10:22 .
drwxr-xr-x 3 researcher2 research_team 4096 Jun 24 10:54 ..
-r--r----- 1 researcher2 research_team  46 Jun 24 10:22 .project_x.txt
drwx----- 2 researcher2 research_team 4096 Jun 24 10:22 drafts
-rw-rw-r-- 1 researcher2 research_team  46 Jun 24 10:22 project_k.txt
-rw----- 1 researcher2 research_team  46 Jun 24 10:22 project_m.txt
-rw-rw-r-- 1 researcher2 research_team  46 Jun 24 10:22 project_r.txt
-rw-rw-r-- 1 researcher2 research_team  46 Jun 24 10:22 project_t.txt
researcher2@3b197108bdba:~/projects$
```

Usei o comando “chmod g-x drafts” para remover a permissão de execução do grupo no diretório drafts.

Isso garante que somente o usuário researcher2 consiga acessar o conteúdo da pasta, já que sem a permissão de execução (x), o grupo não consegue nem abrir o diretório.

Resumo

Espero que minha explicação tenha feito sentido. Tentei ser o mais didático possível para que até quem não está acostumado a usar o terminal Linux consiga entender o básico sobre permissões.

Durante essa atividade, realizei verificações com `ls -la` e usei o comando `chmod` para ajustar as permissões de leitura, escrita e execução de arquivos e diretórios. Com isso, consegui controlar melhor quem pode acessar ou modificar determinados conteúdos.

Essas mudanças ajudaram minha organização a garantir a segurança dos arquivos, promovendo mais confiança e integridade nos dados armazenados.

Feito por: Mathias Petry Peixoto

