Capitulo 7. Controlar acesso a arquivos

Permissões do sistema de arquivos Linux

Os arquivos possuem três categorias de usuário às quais as permissões se aplicam.

O arquivo é de propriedade de um usuário, normalmente o criador.

O arquivo também é de propriedade de um único grupo, geralmente o grupo primário do usuário que criou o arquivo, mas isso pode ser alterado.

As permissões mais específicas têm prioridade. As permissões de usuário substituem as permissões de grupo, que substituem outras permissões.

Como exemplo de como a associação ao grupo habilita a colaboração entre usuários

Imagine que seu sistema tenha dois usuários: `alice` e `bob`.

- `alice` é membro dos grupos `alice` e `web`,
- `bob` é membro dos grupos `bob`, `wheel` e `web`.
- Quando `alice` e `bob` trabalham juntos, os arquivos devem ser associados ao grupo `web`, e as permissões de grupo devem permitir o acesso aos arquivos para ambos os usuários.

Três categorias de permissões se aplicam: leitura, gravação e execução.

Tabela. Efeito das permissões em arquivos e diretorios

Permissão	Efeitos em arquivos	Efeitos em diretórios
r (read)	**O conteudo pode ser lido**	**O conteudo do diretorio (nomes de arquivos) pode ser listado**
w (write)	**O conteudo pode ser alterado**	**Qualquer arquivo do diretorio pode ser criado ou apagado**

x (execute)	**Os arquivos podem ser executados como comandos**	**O diretorio pode se tornar o diretorio de trabalho atual. Pode executar o comando `cd` nele, mas tambem exigi permissão de leitura para listar os arquivos encontrados la**

- Os usuarios normalmente têm as permissões de leitura e execução em diretórios somente leitura para que possam listar o diretório e tenham acesso somente leitura a todo o conteúdo dele.
- Se um usuário tiver apenas acesso de leitura em um diretório, ele poderá listar os nomes dos arquivos nele. No entanto, o usuário não pode acessar outras informações, como permissões ou carimbos de data e hora.
- Se um usuário tiver apenas acesso de execução em um diretório, ele não poderá listar os nomes de arquivo no diretório. Se ele souber o nome de um arquivo que tem permissão para ler, poderá acessar o conteúdo desse arquivo de fora do diretório especificando explicitamente o nome do arquivo relativo.
- Qualquer pessoa que seja proprietária ou tenha permissão de gravação no diretório pode remover arquivos dele, independentemente da propriedade ou das permissões do próprio arquivo.
    - Nota:
        - No Linux, as permissões se aplicam apenas ao arquivo ou diretório no qual elas estão definidas. Os subdiretórios dentro de um diretório não herdam automaticamente as permissões do diretório pai. No entanto, as permissões de diretório poderão bloquear o acesso ao conteúdo do diretório, se forem definidas de maneira restritiva.
        - A permissão `read` em um diretório do Linux é equivalente a **List folder contents** no Windows. A permissão `write` em um diretório do Linux é semelhante a **Modify** no Windows. Ela pressupõe a capacidade de excluir arquivos e subdiretórios. No Linux, com permissões `write` e a **sticky bit** em um diretório, somente o usuário ou grupo proprietário podem excluir arquivos, o que é semelhante ao comportamento do **Write** no Windows.
        - O usuário `root` do Linux tem o equivalente à permissão **Full Control** do Windows em todos os arquivos. No entanto, a política do

SELinux pode usar contextos de segurança de arquivos e processos para restringir o acesso até mesmo ao usuário `root`.

Exibição de permissões e da propriedade de arquivos e diretórios

No primeiro exemplo:

As permissões para o usuário `student` são o primeiro conjunto de três caracteres.

O usuário `student` tem permissões de leitura e gravação no arquivo `test`, mas não tem permissão de execução.

O segundo conjunto de três caracteres são as permissões para o grupo `student`: as permissões de leitura e gravação em `test`, mas não a permissão de execução. O terceiro conjunto de três caracteres são as permissões para todos os outros usuários: somente a permissão de leitura em `test`.

O conjunto de permissões que vale é o mais específico. Então, se o usuário `student` tiver permissões diferentes do grupo `student`, e o usuário `student` também for um membro desse grupo, apenas as permissões do usuário proprietário se aplicam. Essa permissão possibilita a definição de um conjunto mais restritivo de permissões em um usuário do que a sua associação ao grupo oferece, quando não for prático remover o usuário do grupo.

A opção `-l` do comando `ls` mostra informações mais detalhadas sobre permissões e propriedade:

```
[user@host ~]$ls -l test
-rw-rw-r--. 1 student student 0 Mar  8 17:36 test
```

Segundo exemplo:

Use a opção `ls` do comando `-d` para mostrar informações detalhadas sobre um diretório em si, e não seu conteúdo.

```
[user@host ~]$ls -ld /home
drwxr-xr-x. 5 root root 4096 Feb 31 22:00 /home
```

O primeiro caractere da listagem longa é o tipo de arquivo, e é interpretado assim:

- `-` é um arquivo regular
- `d` é um diretorio
- `l` é um link simbolico
- `c` é um arquivo de dispositivo de caracteres
- `b` é um arquivo de dispositivo de bloco
- `p` é um arquivo de pipe nomeado
- `s` é um arquivo de soquete local

Os próximos nove caracteres representam as permissões de arquivo.

Esses caracteres são interpretados como três conjuntos de três caracteres:

- o primeiro conjunto são permissões que se aplicam ao proprietário do arquivo
- o segundo conjunto é para o proprietário do grupo do arquivo
- o último conjunto se aplica a todos os outros usuários (mundo)

Se um conjunto for uma string `rwx`, esse conjunto tem as três permissões: ler, gravar e executar.
Se uma letra for substituída por um traço
`-`, esse conjunto não terá essa permissão.

Após a segunda coluna (contagem de links), o primeiro nome especifica o proprietário do arquivo e o segundo nome, o grupo proprietário do arquivo.

Exemplos de efeitos de permissão

Os exemplos a seguir ilustram como as permissões de arquivo interagem. Para esses exemplos, seus sistema tem quatro usuários com as seguintes associações a grupos:

Usuario	Associação a grupos
operator1	operator1, consultant1
database1	database1, consultant1
database2	database2, operator2
contractor1	contractor1, operator2

Esses usuários trabalham com arquivos no diretório `dir`. Esta é uma listagem longa dos arquivos nesse diretório:

```
[database1@host dir]$ls -la
total 24
drwxrwxr-x.  2 database1 consultant1   4096 Mar  4 10:23 .
drwxr-xr-x. 10 root      root          4096 Mar  1 17:34 ..
-rw-rw-r--.  1 operator1 operator1     1024 Mar  4 11:02 ap
p1.log
-rw-r--rw-.  1 operator1 consultant1   3144 Mar  4 11:02 ap
p2.log
-rw-rw-r--.  1 database1 consultant1  10234 Mar  4 10:14 db
1.conf
-rw-r-----.  1 database1 consultant1   2048 Mar  4 10:18 db
2.conf
```

A opção `-a` do comando `ls` mostra as permissões de arquivos ocultos, incluindo os arquivos especiais para representar o diretório e seu pai. Nesse exemplo, o diretório especial `.` reflete as permissões de `dir` em si e o diretório especial `..` reflete as permissões do diretório pai.

Para o arquivo `db1.conf`, o usuário proprietário do arquivo (`database1`) tem permissões de leitura e gravação, mas não tem permissão de execução.

O grupo proprietário do arquivo (`consultant1`) tem permissões de leitura e gravação, mas não tem permissão de execução. Todos os outros usuários têm permissão de leitura, mas não têm permissões de gravação ou execução.

A seguinte tabela explora alguns dos efeitos desse conjunto de permissões para esses usuários:

Efeito	Por que esse efeito é verdade?
O usuário `operator1` pode alterar o conteúdo do arquivo `db1.conf`.	O usuário `operator1` é membro do grupo `consultant1`, e esse grupo tem permissões de leitura e gravação no arquivo `db1.conf`
O usuário `database1` pode visualizar e modificar o conteúdo do arquivo `db2.conf`.	O usuário `database1` é proprietário do arquivo `db2.conf` e têm acesso de leitura e de gravação.
O usuário `operator1` pode visualizar, mas não modificar o conteúdo do arquivo `db2.conf`.	O usuário `operator1` é membro do grupo `consultant1`, e esse grupo tem apenas acesso de leitura para o arquivo `db2.conf`.
Os usuários `database2` e `contractor1` não têm acesso ao conteúdo do arquivo `db2.conf`.	As permissões `other` se aplicam aos usuários `database2` e `contractor1`, mas não incluem o acesso de leitura nem de gravação.
O usuário `operator1` é o único usuário que pode alterar o conteúdo do arquivo `app1.log`.	O usuário `operator1` e os membros do grupo `operator1` têm permissão de gravação no arquivo, enquanto os outros usuários não têm. No entanto, o único membro do grupo `operator1` é o usuário `operator1`.
O usuário `database2` pode alterar o conteúdo do arquivo `app2.log`.	O usuário `database2` não é o proprietário do arquivo `app2.log` e não está no grupo `consultant1`, por isso, as permissões `other` se aplicam. As permissões `other` concedem permissão de gravação ao arquivo.

O usuário `database1` pode ver o conteúdo do arquivo `app2.log`, mas não modificar o conteúdo do arquivo `app2.log`.	O usuário `database1` é membro do grupo `consultant1`, e esse grupo tem somente permissões de leitura no arquivo `app2.log`. Mesmo que as permissões `other` incluam permissão de gravação, as permissões do grupo têm prioridade.
O usuário `database1` pode excluir os arquivos `app1.log` e `app2.log`.	O usuário `database1` tem permissões de gravação no diretório `dir`, o que o diretório especial `.` mostra, e, por isso, pode excluir qualquer arquivo nesse diretório. Essa operação é possível mesmo se o usuário `database1` não tiver permissão de gravação nos arquivos diretamente.

Gerenciamento de permissões do sistema de arquivos a partir da linha de comando