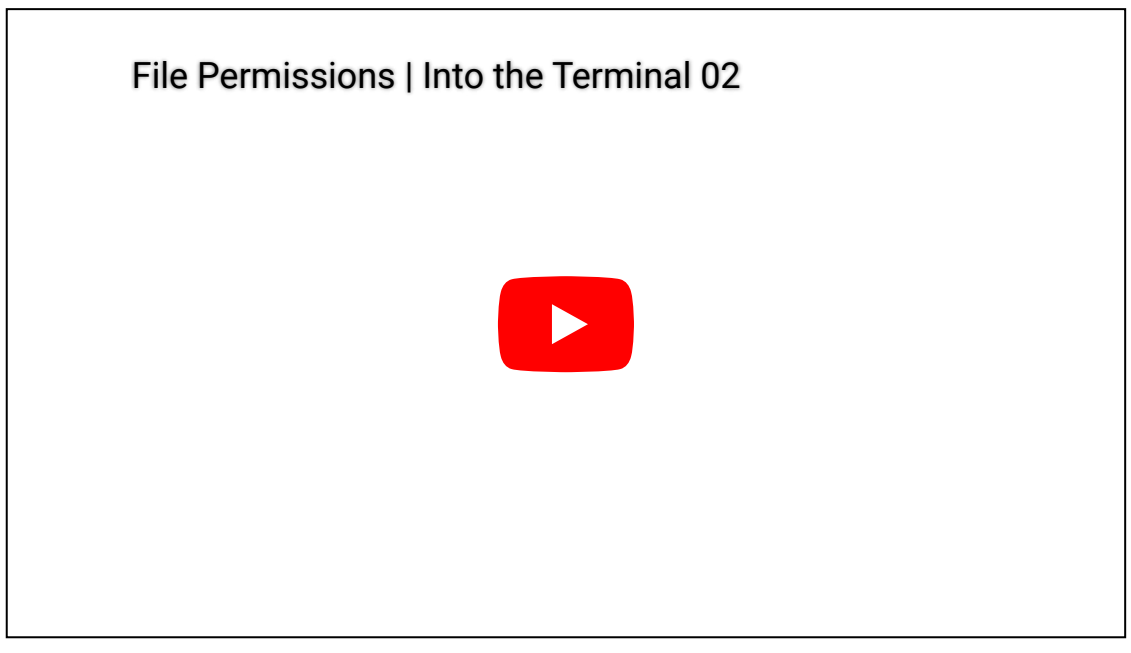
Permissões de arquivo Linux explicadas

Compreender as permissões de arquivos do Linux (como encontrá-los, lê-los e alterá-los) é uma parte importante da manutenção e segurança de seus sistemas.

Postado: 10 de janeiro de 2023 | | Scott McBrien (Chapéu Vermelho)

Foto de Eliobed Suarez no Unsplash

As permissões de arquivo são fundamentais para o modelo de segurança usado pelos sistemas Linux. Eles determinam quem pode acessar arquivos e diretórios em um sistema e como. Este artigo fornece uma visão geral das permissões de arquivo do Linux, como funcionam e como alterá-las.



Como você visualiza as permissões de arquivo do Linux?

O `ls` comando junto com sua `-l` opção (para listagem longa) mostrará metadados sobre seus arquivos Linux, incluindo as permissões definidas no arquivo.

```
$ ls -l

drwxr-xr-x. 4 root root    68 Jun 13 20:25 tuned
-rw-r--r--. 1 root root  4017 Feb 24  2022 vimrc
```

Neste exemplo, você vê duas listagens diferentes. O primeiro campo da `ls -l`saída é um grupo de metadados que inclui as permissões de cada arquivo. Aqui estão os componentes da `vimrc`listagem:

- Tipo de arquivo:`-`
- Configurações de permissão:`rw-r--r--`
- Atributos estendidos: ponto (`.`)
- Proprietário do usuário:`root`
- Proprietário do grupo:`root`

Os campos "Tipo de arquivo" e "Atributos estendidos" estão fora do escopo deste artigo, mas na saída apresentada acima, o `vimrc`arquivo é um arquivo normal, que é um tipo de arquivo `-`(ou seja, nenhum tipo especial).

A `tuned`listagem é para um `d`arquivo do tipo , ou diretório. Existem outros tipos de arquivos também, mas esses dois são os mais comuns. Os atributos disponíveis dependem do formato do sistema de arquivos no qual os arquivos estão armazenados. Para Red Hat Enterprise Linux 7, 8 e 9, o formato padrão do sistema de arquivos é XFS.

Ótimos recursos Linux

- Folha de referências de comandos avançados do Linux
- Baixe o RHEL 9 gratuitamente através do programa Red Hat Developer
- Um guia para instalar aplicativos no Linux
- Avaliação de habilidades de administração de sistema Linux
- Você conhece bem o Linux? Faça um teste e ganhe um selo

Como você lê as permissões de arquivo?

Este artigo é sobre as configurações de permissão em um arquivo. As permissões interessantes da `vimrc`listagem são:

```
rw-r--r-
```

Esta string é na verdade uma expressão de três conjuntos diferentes de permissões:

- rw-
- r--
- r--

O primeiro conjunto de permissões se aplica ao proprietário do arquivo. O segundo conjunto de permissões aplica-se ao grupo de usuários que possui o arquivo. O terceiro conjunto de permissões é geralmente chamado de “outros”. Todos os arquivos Linux pertencem a um proprietário e a um grupo.

Quando permissões e usuários são representados por letras, isso é chamado de modo simbólico. Para usuários, `u`significa proprietário do usuário, `g`proprietário do grupo e `o`outros. Para permissões, `r`significa leitura, `w`gravação e `x`execução.

[Aprenda como gerenciar seu ambiente Linux para ter sucesso .]

Quando o sistema analisa as permissões de um arquivo para determinar quais informações fornecer quando você interage com um arquivo, ele executa uma série de verificações:

1. Ele primeiro verifica se você é o usuário proprietário do arquivo. Nesse caso, você receberá as permissões do proprietário do usuário e nenhuma verificação adicional será concluída.
2. Se você não for o usuário proprietário do arquivo, em seguida sua associação ao grupo será validada para verificar se você pertence ao grupo que corresponde ao proprietário do grupo do arquivo. Nesse caso, você estará coberto pelo campo de permissões do proprietário do grupo e nenhuma verificação adicional será feita.
3. As permissões "Outros" são aplicadas quando a conta que interage com o arquivo não é a proprietária do usuário nem faz parte do grupo que possui os arquivos. Ou, dito de outra forma, os três campos são mutuamente exclusivos: Você não pode ser coberto por mais de um dos campos de configurações de permissão em um arquivo.

As permissões vão além dos diferentes tipos de pessoas que podem interagir com um arquivo. Cada usuário recebe uma expressão que inclui os três tipos básicos de permissões. No exemplo acima, o proprietário do arquivo recebe as seguintes permissões:

```
rw-
```

Cada caractere na expressão indica se uma permissão específica foi concedida ou não. No exemplo acima, `r`as permissões de leitura ( ) e gravação (`w`) foram concedidas no arquivo. Porém, a permissão de execução (`x`) não é concedida, por isso há um `-`sinal na expressão. A permissão neste campo está desabilitada.

Considere as permissões do proprietário do grupo neste exemplo:

```
r--
```

A `r`permissão de leitura ( ) é concedida aos membros do grupo, mas a gravação e a execução foram desativadas.

[Mantenha os comandos mais usados à mão com a folha de referências dos comandos do Linux .]

O que são valores octais?

Quando as permissões de arquivo do Linux são representadas por números, isso é chamado de modo numérico. No modo numérico, um valor de três dígitos representa permissões de arquivo específicas (por exemplo, 744). Eles são chamados de valores octais. O primeiro dígito é para permissões de proprietário, o segundo dígito é para permissões de grupo e o terceiro é para outros usuários. Cada permissão possui um valor numérico atribuído a ela:

- r (ler): 4
- w (escrever): 2
- x (executar): 1

No valor de permissão 744, o primeiro dígito corresponde ao usuário, o segundo dígito ao grupo e o terceiro dígito aos demais. Ao somar o valor de cada classificação de usuário, você pode encontrar as permissões do arquivo.

Por exemplo, um arquivo pode ter permissões de leitura, gravação e execução para seu proprietário e apenas permissão de leitura para todos os outros usuários. É assim:

- Proprietário: rwx = 4+2+1 = 7
- Grupo: r-- = 4+0+0 = 4
- Outros: r-- = 4+0+0 = 4

Os resultados produzem o valor de três dígitos 744.

Segurança Linux

- O que é automação de segurança?
- Perguntas frequentes sobre segurança do Red Hat OpenShift Service na AWS
- Aumente a segurança com automação
- Guia de implementação de DevSecOps
- Verificador CVE da Red Hat

O que as permissões de arquivo do Linux realmente fazem?

Já falei sobre como visualizar as permissões de arquivo, a quem elas se aplicam e como ler quais permissões estão habilitadas ou desabilitadas. Mas o que essas permissões realmente fazem na prática?

Leia (r)

A permissão de leitura é usada para acessar o conteúdo do arquivo. Você pode usar uma ferramenta como `cat`ou `less`no arquivo para exibir o conteúdo do arquivo. Você também pode usar um editor de texto como o Vi ou `view`no arquivo para exibir o conteúdo do arquivo. A permissão de leitura é necessária para fazer cópias de um arquivo, porque você precisa acessar o conteúdo do arquivo para duplicá-lo.

Escreva (w)

A permissão de gravação permite modificar ou alterar o conteúdo de um arquivo. A permissão de gravação também permite que você use os operadores de redirecionamento ou acréscimo no shell (`>`ou `>>`) para alterar o conteúdo de um arquivo. Sem permissão de gravação, não são permitidas alterações no conteúdo do arquivo.

Executar (x)

A permissão de execução permite executar o conteúdo de um arquivo. Normalmente, os executáveis seriam coisas como comandos ou aplicativos binários compilados. No entanto, a permissão de execução também permite que alguém execute scripts de shell Bash, programas Python e uma variedade de linguagens interpretadas.

[Baixe agora: um guia do administrador de sistema para scripts Bash .]

Existem outras maneiras de executar o conteúdo de um arquivo sem permissão de execução. Por exemplo, você pode usar um intérprete que tenha permissão de execução para ler um arquivo com instruções para o intérprete executar. Um exemplo seria invocar um script de shell Bash:

```
$ bash script.sh
```

O executável que está sendo executado é o `bash`. O `script.sh`arquivo é lido pelo interpretador Bash e seus comandos são executados. O conteúdo deste artigo é de uso geral, mas no Linux geralmente existem maneiras adicionais de realizar tarefas .

Como funcionam as permissões de diretório?

Os tipos de arquivo de diretório são indicados com `d`. Conceitualmente, as permissões funcionam da mesma maneira, mas os diretórios interpretam essas operações de maneira diferente.

Leia (r)

Como os arquivos normais, esta permissão permite ler o conteúdo do diretório. No entanto, isso significa que você pode visualizar o conteúdo (ou arquivos) armazenado no diretório. Essa permissão é necessária para que coisas como o `ls`comando funcionem.

Escreva (w)

Tal como acontece com os arquivos normais, isso permite que alguém modifique o conteúdo do diretório. Ao alterar o conteúdo do diretório, você está adicionando arquivos ao diretório ou removendo arquivos do diretório. Como tal, você deve ter permissão de gravação em um diretório para mover (`mv`) ou remover (`rm`) arquivos dele. Você também precisa de permissão de gravação para criar novos arquivos (usando `touch`um operador de redirecionamento de arquivo) ou copiar (`cp`) arquivos para o diretório.

Executar (x)

Essa permissão é muito diferente em diretórios em comparação com arquivos. Essencialmente, você pode pensar nisso como um fornecimento de acesso ao diretório. Ter permissão de execução em um diretório autoriza você a ver informações estendidas sobre os arquivos no diretório (usando `ls -l`, por exemplo), mas também permite alterar seu diretório de trabalho (usando `cd`) ou passar por esse diretório no caminho para um subdiretório abaixo.

A falta de permissão de execução em um diretório pode limitar as outras permissões de maneiras interessantes. Por exemplo, como você pode adicionar um novo arquivo a um diretório (aproveitando a permissão de gravação) se não consegue acessar os metadados do diretório para armazenar as informações de um arquivo novo e adicional? Você não pode. É por esse motivo que os arquivos do tipo diretório geralmente oferecem permissão de execução para um ou mais proprietários do usuário, proprietário do grupo ou outros.

[Quer testar suas habilidades de administrador de sistemas? Faça uma avaliação de habilidades hoje .]

Como você modifica as permissões de arquivo do Linux?

Você pode modificar as permissões de arquivos e diretórios com o `chmod`comando, que significa “modo de alteração”. Para alterar as permissões de arquivo no modo numérico, você insere `chmod`o valor octal desejado, como 744, ao lado do nome do arquivo. Para alterar as permissões de arquivo no modo simbólico, você insere uma classe de usuário e as permissões que deseja conceder a eles ao lado do nome do arquivo. Por exemplo:

```
$ chmod ug+rwx example.txt
$ chmod o+r example2.txt
```

Isso concede leitura, gravação e execução para o usuário e grupo, e somente leitura para outros. No modo simbólico, `chmod u`representa permissões para o proprietário do usuário, `chmod g`representa outros usuários no grupo do arquivo, `chmod o`representa outros usuários que não estão no grupo do arquivo. Para todos os usuários, use `chmod a`.

Talvez você queira alterar o proprietário do usuário. Você pode fazer isso com o `chown`comando. Da mesma forma, o `chgrp`comando pode ser usado para alterar a propriedade do grupo de um arquivo.

O que são permissões especiais de arquivo?

Permissões especiais estão disponíveis para arquivos e diretórios e fornecem privilégios adicionais sobre os conjuntos de permissões padrão que foram abordados.

- SUID é a permissão especial para o nível de acesso do usuário e sempre executa como o usuário proprietário do arquivo, independentemente de quem está passando o comando.
- SGID permite que um arquivo seja executado como o proprietário do grupo do arquivo; um arquivo criado no diretório tem sua propriedade de grupo definida como o proprietário do diretório. Isto é útil para diretórios usados de forma colaborativa entre diferentes membros de um grupo porque todos os membros podem acessar e executar novos arquivos.

O "sticky bit" é uma permissão especial em nível de diretório que restringe a exclusão de arquivos, o que significa que apenas o proprietário do arquivo pode remover um arquivo do diretório.

Quer se aprofundar nas permissões especiais? Leia as permissões do Linux: SUID, SGID e sticky bit .

Empacotando

Compreender as permissões de arquivos do Linux (como encontrá-los, lê-los e alterá-los) é uma parte importante da manutenção e segurança de seus sistemas. Você pode aprender mais sobre permissões de arquivo para o Red Hat Enterprise Linux verificando a documentação ou praticando em um laboratório individualizado sobre como usar permissões de arquivo .

[Folha de dicas: Obtenha uma lista de utilitários e comandos do Linux para gerenciar servidores e redes .]