

# Permissões de arquivos em Linux

## Descrição do projeto

Trata-se de um cenário em laboratório que retrata um analista de segurança examinando e gerenciando as permissões de arquivos e diretórios de uma empresa por meio do terminal Bash no Linux. Irei mostrar, passo a passo, cada etapa por meio de prints, comandos e explicações.

## Checar os detalhes dos arquivos e diretórios

Ao iniciar o laboratório, executei o comando “pwd” para o terminal me retornar o endereço do diretório onde eu estava trabalhando, sendo ele: “/home/researcher2”. Após isso, utilizei o comando “ls” para listar os arquivos e diretórios, retornando apenas o diretório “**projects**”. Por fim, entrei no diretório utilizando “**cd projects**” e utilizei novamente o comando ls, dessa vez com os argumentos “-la” para me listar os arquivos ocultos e as permissões dos arquivos.

```
researcher2@0b4da87354f2:~$ pwd
/home/researcher2
researcher2@0b4da87354f2:~$ ls
projects
researcher2@0b4da87354f2:~$ cd projects
researcher2@0b4da87354f2:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Sep 12 20:14 .
drwxr-xr-x 3 researcher2 research_team 4096 Sep 12 20:30 ..
-rw--w--- 1 researcher2 research_team  46 Sep 12 20:14 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Sep 12 20:14 drafts
-rw-rw-rw- 1 researcher2 research_team  46 Sep 12 20:14 project_k.txt
-rw-r----- 1 researcher2 research_team  46 Sep 12 20:14 project_m.txt
-rw-rw-r-- 1 researcher2 research_team  46 Sep 12 20:14 project_r.txt
-rw-rw-r-- 1 researcher2 research_team  46 Sep 12 20:14 project_t.txt
```

## Descreva a string das permissões

No Linux, as permissões dos arquivos e diretórios são representadas por uma string de 10 caracteres. Exemplo: “**drwxrwxrwx**” - o primeiro caractere indica o **tipo** de arquivo (no exemplo, um diretório), o segundo, terceiro e quarto caracteres indicam as **permissões do usuário**, o quinto, sexto e sétimo caracteres indicam as **permissões do grupo**, o oitavo, nono e décimo caracteres indicam as **permissões dos outros** (todos os usuários do sistema).

Caso alguma char da string esteja faltando, significa que aquela permissão não está concedida para o tipo de proprietário, exemplo: “-rw-rw-rw-” - a permissão de execução do arquivo não foi concedida para nenhum tipo de proprietário.

```
drwxr-xr-x 3 researcher2 research_team 4096 Sep 12 20:14 .
drwxr-xr-x 3 researcher2 research_team 4096 Sep 12 20:30 ..
-rw--w---- 1 researcher2 research_team  46 Sep 12 20:14 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Sep 12 20:14 drafts
-rw-rw-rw- 1 researcher2 research_team  46 Sep 12 20:14 project_k.txt
-rw-r----- 1 researcher2 research_team  46 Sep 12 20:14 project_m.txt
-rw-rw-r-- 1 researcher2 research_team  46 Sep 12 20:14 project_r.txt
-rw-rw-r-- 1 researcher2 research_team  46 Sep 12 20:14 project_t.txt
```

## Mude as permissões dos arquivos

Com base na listagem passada, examinou-se que o arquivo “**project\_k.txt**” estava com permissões indevidas, visto que o tipo de usuário “**outros**” possuía permissão de escrita. Para corrigir isso, utilizei o comando “**chmod**” para retirar a permissão. Utilizei o argumento “**o-w**” para retirar, daquele arquivo, a permissão de escrita para o tipo de usuário “outros”, e depois utilizei o comando “**ls -l**” para verificar se a permissão foi alterada.

```
researcher2@c1e228f89bf5:~/projects$ chmod o-w project_k.txt
researcher2@c1e228f89bf5:~/projects$ ls -l
total 20
drwx--x--- 2 researcher2 research_team 4096 Sep 12 19:22 drafts
-rw-rw-r-- 1 researcher2 research_team  46 Sep 12 19:22 project_k.txt
-rw-r----- 1 researcher2 research_team  46 Sep 12 19:22 project_m.txt
-rw-rw-r-- 1 researcher2 research_team  46 Sep 12 19:22 project_r.txt
-rw-rw-r-- 1 researcher2 research_team  46 Sep 12 19:22 project_t.txt
```

## Mude as permissões de um arquivo oculto

Agora foi necessário utilizar o comando ls com os argumento “**-la**” para procurar por arquivos ocultos e mostrar as suas permissões. O arquivo oculto “**.project\_x.txt**” deveria ter apenas permissões de leitura para o usuário proprietário e para o grupo, nada além disso, então foi necessário utilizar o comando “**chmod u-w,g-w,g+r .project\_x.txt**” para retirar as permissões de escrita do usuário e do grupo, e adicionar a de leitura para o grupo. Por fim, utilizei novamente o ls para verificar se as mudanças foram implementadas.

```
researcher2@644d83d7b986:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Sep 12 21:14 .
drwxr-xr-x 3 researcher2 research_team 4096 Sep 12 22:18 ..
-rw--w--- 1 researcher2 research_team  46 Sep 12 21:14 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Sep 12 21:14 drafts
-rw-rw-r-- 1 researcher2 research_team  46 Sep 12 21:14 project_k.txt
-rw----- 1 researcher2 research_team  46 Sep 12 21:14 project_m.txt
-rw-rw-r-- 1 researcher2 research_team  46 Sep 12 21:14 project_r.txt
-rw-rw-r-- 1 researcher2 research_team  46 Sep 12 21:14 project_t.txt
researcher2@644d83d7b986:~/projects$ chmod u-w,g-w,g+r .project_x.txt
researcher2@644d83d7b986:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 Sep 12 21:14 .
drwxr-xr-x 3 researcher2 research_team 4096 Sep 12 22:18 ..
-r--r----- 1 researcher2 research_team  46 Sep 12 21:14 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 Sep 12 21:14 drafts
-rw-rw-r-- 1 researcher2 research_team  46 Sep 12 21:14 project_k.txt
-rw----- 1 researcher2 research_team  46 Sep 12 21:14 project_m.txt
-rw-rw-r-- 1 researcher2 research_team  46 Sep 12 21:14 project_r.txt
-rw-rw-r-- 1 researcher2 research_team  46 Sep 12 21:14 project_t.txt
```

## Mude as permissões de um diretório

Nesta ocasião, a exigência para o subdiretório “**drafts**” é que apenas o usuário tenha privilégios de execução, ou seja, apenas ele terá a permissão de executar (acessar o diretório e seus conteúdos). Utilizei o comando “**ls -l**” para verificar as permissões do diretório, e após constatar que o grupo possui permissões indevidas, utilizei o comando “**chmod g-x drafts**” para retirar a permissão de executar do grupo, deixando apenas o usuário com as permissões necessárias e mantendo o privilégio mínimo.

```
researcher2@644d83d7b986:~/projects$ ls -l
total 20
drwx--x--- 2 researcher2 research_team 4096 Sep 12 21:14 drafts
-rw-rw-r-- 1 researcher2 research_team  46 Sep 12 21:14 project_k.txt
-rw----- 1 researcher2 research_team  46 Sep 12 21:14 project_m.txt
-rw-rw-r-- 1 researcher2 research_team  46 Sep 12 21:14 project_r.txt
-rw-rw-r-- 1 researcher2 research_team  46 Sep 12 21:14 project_t.txt
researcher2@644d83d7b986:~/projects$ chmod g-x drafts
researcher2@644d83d7b986:~/projects$ ls -l
total 20
drwx----- 2 researcher2 research_team 4096 Sep 12 21:14 drafts
-rw-rw-r-- 1 researcher2 research_team  46 Sep 12 21:14 project_k.txt
-rw----- 1 researcher2 research_team  46 Sep 12 21:14 project_m.txt
-rw-rw-r-- 1 researcher2 research_team  46 Sep 12 21:14 project_r.txt
-rw-rw-r-- 1 researcher2 research_team  46 Sep 12 21:14 project_t.txt
```

## Sumário

Em resumo, esta atividade foi feita com o objetivo de realizar o gerenciamento de permissões em um ambiente Linux pelo terminal Bash. Passou pela análise de permissões dos arquivos e diretórios, pela explicação de conceitos e pelo uso de comandos.