Semana 3 - Controle de Acesso a Arquivos

Administração de Redes e Segurança Digital
Prof. Gaio B. Oliveira

OBJETIVOS DE APRENDIZAGEM

- Entender o conceito de permissões de arquivos no Linux.
- Conhecer os diferentes níveis de acesso (dono, grupo, outros).
- Conhecer comandos para visualizar e modificar permissões.
- Exercitar os conceitos através de comandos práticos no bash.

CONTEÚDO

Introdução

O que são permissões de arquivos?

No Linux, o sistema de permissões define quem pode ler, escrever e executar um arquivo ou diretório.
Cada arquivo ou diretório tem um proprietário (dono), um grupo e outros usuários.

Como visualizar permissões?

O comando `ls -l` exibe as permissões dos arquivos:

ls -l

Saída esperada:

-rwxr-xr-- 1 user group 1234 Mar 26 12:00 arquivo.txt

Explicação dos campos:

- `-rwxr-xr--` → Permissões do arquivo.
- `1` → Número de links.
- `user` → Dono do arquivo.

- `group` → Grupo do arquivo.
- `1234` → Tamanho em bytes.
- `Mar 26 12:00` → Data e hora da última modificação.
- `arquivo.txt` → Nome do arquivo.

Entendendo as permissões

Cada conjunto de três caracteres representa permissões para:

1. **Dono** (User - "u")
2. **Grupo** (Group - "g")
3. **Outros** (Others - "o")

Os tipos de permissões:

- `r` (read) → leitura
- `w` (write) → escrita
- `x` (execute) → execução

Exemplo: `-rw-r--r--`

- Dono: pode ler e escrever (`rw-`).
- Grupo: pode apenas ler (`r--`).
- Outros: podem apenas ler (`r--`).

Modificando Permissões

1. Comando chmod (Alterar permissões)

Modo simbólico:

Adicionando permissão de execução para o dono:

chmod u+x arquivo.txt

Removendo permissão de escrita para grupo e outros:

chmod go-w arquivo.txt

Modo numérico (octal):

Cada permissão tem um valor:

- `r = 4`, `w = 2`, `x = 1`
- Somando os valores, definimos as permissões.

Exemplo:

chmod 754 arquivo.txt

Significado:

- Dono: 7 (4+2+1 = leitura, escrita, execução)
- Grupo: 5 (4+0+1 = leitura e execução)
- Outros: 4 (4+0+0 = apenas leitura)

2. Comando chown (Alterar dono do arquivo)

Mudar o dono para `joao`:

chown joao arquivo.txt

Mudar dono e grupo:

chown joao:professores arquivo.txt

3. Comando chgrp (Alterar grupo do arquivo)

Mudar apenas o grupo:

chgrp professores arquivo.txt

ATIVIDADES

Exercícios Práticos

- **Listar permissões dos arquivos no diretório atual**
    - Comando: `ls -l`
    - Perguntas: Qual arquivo tem permissão de execução? Por quê?

- **Criar um arquivo e modificar permissões**

    touch meu_arquivo.txt

    chmod 640 meu_arquivo.txt

    ls -l meu_arquivo.txt

    Pergunta: O que cada número na permissão `640` representa?

- **Dar permissão de execução ao dono**

    chmod u+x meu_arquivo.txt

    ls -l meu_arquivo.txt

    Pergunta: O que mudou?

- **Criar um novo usuário e testar permissões**

    # adduser aluno1

    # su aluno1

    cd /home/aluno

    cat meu_arquivo.txt

    Pergunta: O que aconteceu? Por quê?

- **Resolução de Problema**

    Desafio: Controle de Permissões em um Servidor de Desenvolvimento

Cenário:

Você trabalha como administrador de sistemas em uma empresa de desenvolvimento de software. A equipe de engenharia possui um servidor Linux onde armazena código-fonte, scripts e documentos técnicos.

Os usuários são divididos em três grupos:

1. **Desenvolvedores** (`devs`): Podem modificar o código-fonte.
2. **Testadores** (`qa`): Podem apenas ler o código, sem modificá-lo.
3. **Gerentes** (`gerentes`): Precisam ter acesso total para revisar e aprovar mudanças.

Dentro do diretório `/projetos/`, há um repositório de código para um projeto crítico chamado **"app_seguro"**. O arquivo principal do projeto é:

/projetos/app_seguro/main.py

Atualmente, as permissões estão assim:

ls -l /projetos/app_seguro/main.py

-rw-r--r-- 1 alice devs 3500 Mar 26 15:00 main.py

Isso significa que:

- O dono (`alice`, uma desenvolvedora) pode **ler e modificar** o arquivo.
- O grupo `devs` pode **apenas ler** o arquivo.
- Outros usuários, incluindo testadores e gerentes, também podem **ler** o arquivo.

Problema a Resolver

> **Garantir que os desenvolvedores possam modificar o arquivo:**

Todos os membros do grupo `devs` devem poder editar `main.py`.

Evitar que os testadores (QA) modifiquem o código:

Eles devem apenas ler os arquivos, sem possibilidade de alteração.

Permitir que os gerentes tenham acesso total:
Eles devem poder modificar qualquer arquivo do projeto.

Tarefas a Executar

1. Ajustar as permissões para resolver os problemas listados

2. Adicionar um novo desenvolvedor e testar o acesso

3. Criar um novo testador e testar as permissões

Perguntas para Reflexão:

1. Qual comando `chmod` pode ser usado para permitir que o grupo `devs` modifique `main.py`?
2. Como garantir que os testadores não possam modificar arquivos, mas possam lê-los?
3. Qual comando pode ser usado para garantir que os gerentes tenham acesso total?
4. Se um testador tentar editar `main.py` e receber um erro de permissão, o que pode ter causado isso?
