

Conectividade em Sistemas
CiberFísicos

TDE 2

Usuários, Grupos e Permissões no Sistema de Arquivos



Atividades deste TDE

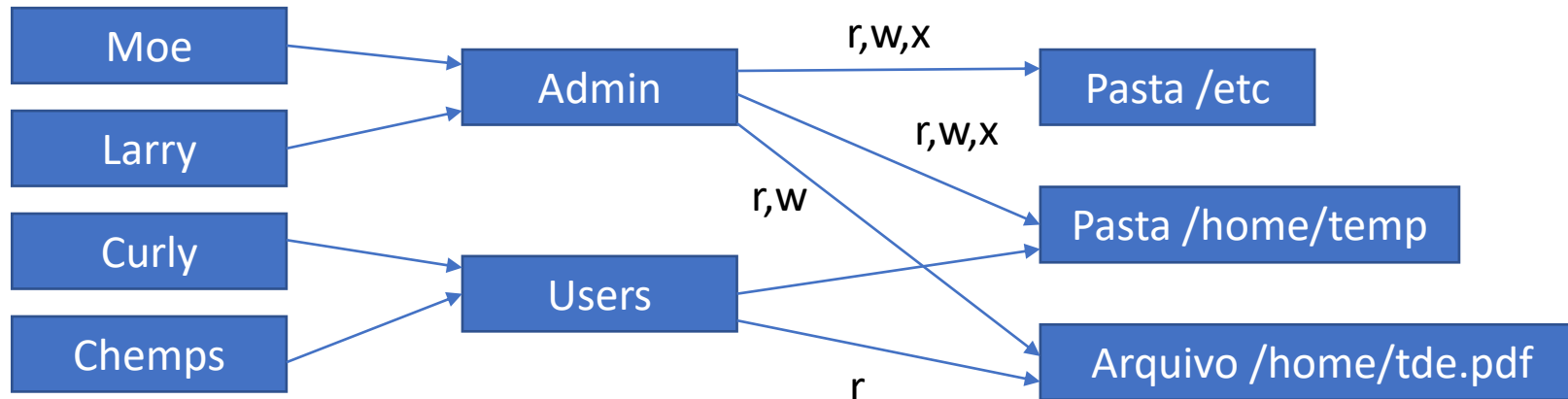
1. Ler alguns aspectos teóricos que estão apresentados nos primeiros slides.
2. Fazer os exercícios práticos conforme indicado
3. A entrega da atividade corresponde ao relatório solicitado no final deste roteiro com a resposta as **PERGUNTAS** formuladas durante o roteiro.

Permissões

- Um sistema operacional pode ser acessado pelo próprio usuário logado ou por usuários remotos, através de aplicações como SSH, Servidores Web e aplicações de Transferência de Arquivos
- É necessário controlar quais usuários tem permissões para acessar arquivos e pastas no sistema de arquivos
- O Linux define três tipos de acesso:
 - Ler (r)
 - Escrever (w)
 - Executar (x)

Usuários e Grupos

- As permissões pode ser atribuídas para cada usuário
- Contudo é mais fácil agrupar os usuários em grupos, e atribuir permissões ao grupo



Usuários e Grupos Pré-Criados

- Os programas e serviços do sistema operacional também recebem permissões através de grupos
- Por isso o sistema cria vários grupos automaticamente
- Você pode ver todos os grupos criados com o comando abaixo:
 - `cat /etc/group | cut -d: -f1`
- Você pode ver todos os usuários criados com o comando abaixo:
 - `cat /etc/passwd | cut -d: -f1`
- arquivo `/etc/group`: `groupname:password:GID:memberusers`
- arquivo `/etc/passwd`: `username:password:UID:GID:comment:home:shell`
- `cut -d: -f1`: separa os campos por ":" e retorna o primeiro campo
- Obs. Passwords estão no arquivo `/etc/shadow`

Exercício1: Informações do Usuário

- Verifique as informações do seu usuário usando o comando a seguir:
 - id login_usuario (exemplo kali, root, etc.)
- O comando traz as seguintes informações:
 - uid = identificador do usuário
 - gid = identificador do grupo primário do usuário (inserido quando ele é criado)
 - grupos = lista de grupos secundários ao qual o usuário pertence
- **PERGUNTA 1:**
 - Indique a quais grupos o usuário conectividade pertence
 - Ele é administrador do sistema?

O grupo dos administradores pode variar de acordo com a distribuição do Linux.

Geralmente é **sudo**. Mas faça uma pesquisa sobre **wheel group**.

DAC: Discretionary Access Control

- O Linux implementa um modelo de permissões denominado DAC
- Nesse modelo, o usuário que criar um arquivo é seu proprietário, e pode determinar quais outros usuários também terão acesso.
- Cada arquivo do sistema está associado a um conjunto de permissões, que podem ser verificadas com o comando abaixo;

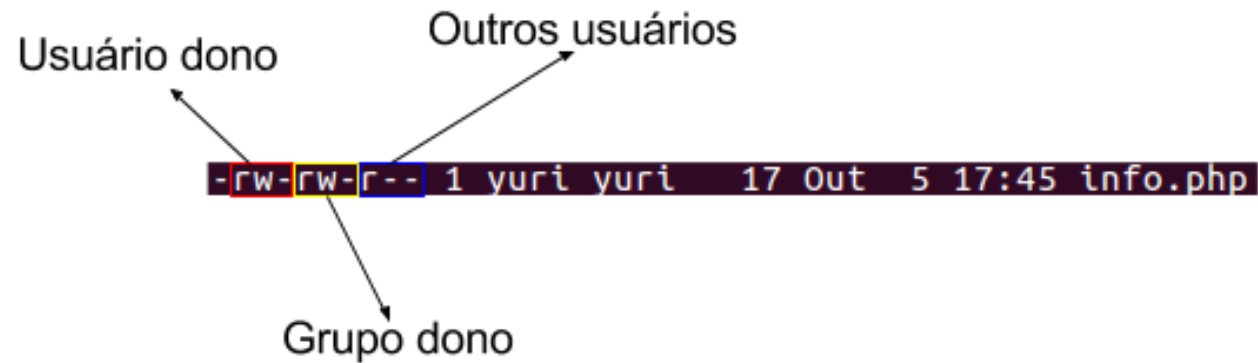
`ls -l`

- As permissões são representadas pelo conjunto de 10 caracteres da primeira coluna.

The diagram illustrates the components of the permission string in the first column of the `ls -l` output. It shows a sample output line: `-rw-rw-r-- 1 yuri yuri 17 Out 5 17:45 info.php`. Three arrows point from labels to specific parts of the permission string `-rw-rw-r--`:

- An arrow from "Usuário dono" points to the first three characters `rw-`.
- An arrow from "Grupo dono" points to the next three characters `rw-`.
- An arrow from "Outros usuários" points to the final three characters `r--`.

Permissões



- O primeiro caractere indica se o objeto listado é um arquivo (-) ou um diretório (d)
- Os nove caracteres restantes nos mostram as permissões do usuário dono, grupo dono e outros usuários respectivamente
- Observe que as permissões se aplicam aos usuários logados no sistema. Verifique o usuário logado com o comando abaixo:
 - `users`

1. O grupo primário tem o mesmo nome do usuário

uid = sharapova
gids = sharapova



cria um arquivo

2. Inicialmente o grupo primário contém apenas o próprio usuário

3. Mas Sharapova **pode adicionar** outras pessoas em seu grupo, por exemplo **Serena**

Linux

4. Suponha que o grupo Sharapova contem **Sharapova e Serena**

Serena usa as permissões do grupo do **DONO**

uid = serena
gid = serena



Federer usa as permissões do grupo **OUTROS**

uid = federer
gid = federer



Se loga no sistema e acessa o arquivo criado pela Sharapova

OBS. Esse exemplo é para um arquivo criado pela Sharapova. E se o arquivo for criado pelo Federer?

O dono (owner) é o usuário logado
uid=sharapova

DONO - GRUPO DO DONO - OUTROS

grupo primário do usuário logado
gid=sharapova

Qualquer usuário que tem conta no Linux mas não está no grupo do dono

Exercício 2: Permissão de Execução

1. Usando o gedit ou outro editor, crie o arquivo teste.sh conforme indicado:	#!/bin/bash echo "bem vindo"
2. Verifique as permissões do arquivo:	ls -l teste.sh
3. Verifique se você consegue executar o script:	./teste.sh
4. Altere as permissões do arquivo:	chmod u=rwx,g=rw,o=r teste.sh ls -l testes.sh
5. Verifique novamente se a execução é possível:	./teste.sh

- **PERGUNTA 2:**

- Indique quais permissões o comando **chmod** modificou comparando o resultado do comando **ls -l** nos passos 2 e 4.

Exercício 3: Criação de Usuários

1. Crie um novo usuário e defina sua senha como 1234:	sudo useradd -m convidado -s /bin/bash sudo passwd convidado
2. Altere o usuário logado para convidado::	su convidado
3. Verifique se ele consegue:	ler o arquivo: cat teste.sh executar o arquivo: ./teste.sh apagar o arquivo: rm teste.sh renomear o arquivo: mv teste.sh teste2.sh copiar o arquivo: cp teste.sh /home/convidado/.

-m: cria um diretório home para o usuário
-s: define o shell default

userdel -r convidado
Remove o usuário e o diretório home

OBS. A senha solicitada pelo sudo é do usuário logado e não do convidado.

• PERGUNTA 3:

- Explique os resultados do passo 3 analisando as permissões do arquivo com **ls -l teste.sh**.

Exercício 4: Permissões em Pastas

1. OBS. Para voltar ao usuário conectividade digite exit
2. Crie a estrutura de pastas com os comandos a seguir:

```
mkdir arquivos
cd arquivos
mkdir publico
mkdir privado
ls -l
cp ../teste.sh publico/teste.sh
ls -l publico
chmod u=rwx,g=rx,o=r privado
chmod -R g+rx publico
ls -l
sudo adduser convidado dono_arquivo
id convidado
```

Note que o
arquivo teste.sh
é copiado com a
permissão

Obs. **umask -S** mostra a permissão
default:

u=rwx,g=rwx,o=rx

Para novos diretórios é feito um AND
com 777:

rwX rwX r-X

111 111 111

rwX rwX r-X

Para novos arquivos é feito um AND
com 666:

rwX rwX r-X

110 110 110

rw- rx- r--

Exercício 4: Permissões em Pastas

1. Entre no diretório home do usuário owner:	cd cd arquivos
2. Mude para o usuário convidado:	su convidado
3. Verifique se ele consegue:	copiar o arquivo para pasta publica : cp ../teste.sh publico/teste2.sh copiar o arquivo para pasta privada : cp ../teste.sh privado/teste2.sh executar o arquivo na pasta pública: cd publico ./teste2.sh

IMPORTANTE:
permissões de escrita
e execução nunca
devem ser dadas a
OUTROS

- **PERGUNTA 4:**

- Explique os resultados do passo 4 analisando as informações do usuário convidado (**id convidado**) e as permissões das pastas obtidas com **ls -l**.

Entrega do TDE

- Responda as quatro perguntas e entregue apenas o relatório