



# Segurança LINUX

---

## RESPOSTAS

1. **Você obteve sucesso em todas as etapas desta atividade? Se não, quais foram os problemas encontrados?**

Eu consegui realizar quase todas as etapas da atividade, exceto a última, onde deveríamos programar em Python.

2. **Você conseguiu realizar a atividade sem consultar outras fontes (colegas, internet, bibliografia...)?**

Tirando a parte que não consegui realizar, só tive que pesquisar sobre o comando "setfacl", que estava escrito errado no roteiro.

3. **Quanto tempo (em minutos) você dedicou à esta atividade?**

Eu realizei a atividade em várias partes, escrevendo tudo o que fazia no relatório abaixo. Portanto não marquei o tempo ao começar, mas em 2 dias estava tudo pronto.

4. **Qual é a sua opinião sobre esta atividade?**

Eu achei ela muito divertida e muito explicativa, principalmente para quem nunca teve contato com o SO Linux e máquinas virtuais. Uma prática importante para aprendermos comandos básicos e a necessidade deles para a segurança.

5. **Você tem sugestões para melhorar atividades práticas desta natureza?**

Acredito que a única parte que ficou confusa da atividade foi a do programa em Python. O resto estava muito bem explicado e fácil de realizar, até mesmo para aqueles não familiarizados com Linux.

## RELATÓRIO DA ATIVIDADE

Comandos	Descrição	Sintaxe
cd	Acessar um diretório	cd pasta1
cd ..	Voltar um nível acima do diretório corrente	cd ..
cd ~	Acessar o diretório home do usuário	cd ~
cd -	Acessar o diretório anterior	cd -
pwd	Verificar o diretório corrente	pwd
mkdir	Criar um diretório	mkdir pasta2
ls	Listar arquivos e diretórios	ls
ls -l	Listar arquivos e diretórios de forma detalhada	ls -l
ls -la	Listar arquivos e diretórios de forma detalhada incluindo arquivos ocultos	ls -la
rm	Remover arquivo ou diretório	rm arquivo1.txt rm -rf pasta1
touch	Criar um arquivo vazio	touch arquivo1.txt
chmod	Alterar permissões básicas	sudo chmod <máscara> <objeto>
useradd	Criar um novo usuário	useradd <usuário>
groupadd	Criar um novo group	groupadd <grupo>
chown	Modificar um proprietário de um arquivo	chown <usuário>:<grupo> <objeto>
setfacl	Estender permissão da lista de controle de acesso (ACL). Definir (-m) ou Remover (-b)	setfacl -m u:<usuário>: <permissão> <objeto> setfacl -b <objeto>
getfacl	Consultar a permissão ACL estendida	getfacl <objeto>

1. **Bloco de destino:** usuário (u), grupo (g), outros (o) e todos (a);
2. **Operação:** incluir (+) ou remover permissão (-);

3. **Permissão:** leitura (r), escrita (w) e execução (x).

4. Para aplicar em diretórios colocar -R logo após o nome do comando

Permissão	Binário	Decimal
---	000	0
--x	001	1
-w-	010	2
-wx	011	3
r--	100	4
r-x	101	5
rw-	110	6
rwX	111	7

- **Usuário → Grupo → Outros**
- Execução (1) - Escrita (2) - Leitura (4)

## Atividade Manipulação de Arquivos

1. Verifique qual o seu diretório corrente

```
pwd → /home/kali
```

2. Crie dois diretórios "tsi\_pasta1" e outro "tsi\_pasta2".

```
mkdir tsi_pasta1 / mkdir tsi_pasta2
```

3. Acesse o diretório "tsi\_pasta1"

```
cd tsi_pasta1
```

4. Crie dois arquivos "arquivo1.txt" e "arquivo2.txt".

```
touch arquivo1.txt / touch arquivo2.txt
```

5. Retorne um nível na hierarquia de diretório. (cd ..)

```
cd ..
```

6. Verifique qual o seu diretório corrente.

```
pwd → /home/kali
```

7. Retorne ao diretório anterior. (cd -)

```
cd -
```

8. Verifique o seu diretório corrente.

```
pwd → /home/kali/tsi_pasta1
```

9. Liste os arquivos sem detalhes.

```
ls → arquivo1.txt arquivo2.txt
```

10. Remova o arquivo "arquivo1.txt".

```
rm arquivo1.txt
```

## Atividade Permissões Básicas

1. Verifique qual o seu diretório corrente.

```
pwd → /home/kali/tsi_pasta1
```

2. Acesse o diretório home do usuário. (cd ~)

```
cd ~
```

3. Acesse o diretório "tsi\_pasta2"

```
cd tsi_pasta2
```

4. Crie dois arquivos "arquivo3.txt" e "arquivo4.txt".

```
touch arquivo3.txt / touch arquivo4.txt
```

5. Liste os arquivos no diretório e verifique as permissões padrões.

```
ls -l
```

```
(kali@kali)-[~/tsi_pasta2]
$ ls -l
total 0
-rw-r--r-- 1 kali kali 0 Mar 23 13:37 arquivo3.txt
-rw-r--r-- 1 kali kali 0 Mar 23 13:37 arquivo4.txt
```

6. Adicione a permissão de execução para o usuário e outros no arquivo arquivo3.txt. -

```
chmod uo+x arquivo3.txt
```

7. Remova a permissão de escrita do grupo no arquivo arquivo3.txt. -

```
chmod g-w arquivo3.txt
```

8. Adicione a permissão de execução para o grupo no arquivo arquivo4.txt e remova a permissão de escrita do usuário utilizando uma única linha de comando (separar por vírgula). -

```
chmod g+x,u-w arquivo4.txt
```

9. Crie o diretório tsi\_pasta3.

```
mkdir tsi_pasta3
```

10. Remova a permissão de execução de todos no diretório tsi\_pasta3. (utilizar o "a" - todos). -

```
chmod -R a-x tsi_pasta3/
```

11. Adicione a permissão de escrita do usuário, grupo e outros no diretório tsi\_pasta3. (utilizar a permissão "ugo" – usuário, grupo e outros). -

```
chmod -R ugo+w tsi_pasta3/
```

12. Liste os arquivos e diretórios e verifique as permissões concedidas.

```
ls -l
```

```
(kali㉿kali)-[~/tsi_pasta2]
$ ls -l
total 4
-rwxr-xr-x 1 kali kali  0 Mar 23 13:37 arquivo3.txt
-r--r-xr-- 1 kali kali  0 Mar 23 13:37 arquivo4.txt
drw-rw-rw- 2 kali kali 4096 Mar 23 13:58 tsi_pasta3
```

## Atividade Permissões Especiais

1. Verifique qual o seu diretório corrente.

```
pwd → /home/kali/tsi_pasta2
```

2. Acesse o diretório home do usuário. (cd ~)

```
cd ~
```

3. Acesse o diretório "tsi\_pasta1"

```
cd tsi_pasta1
```

4. Crie o novamente o arquivo "arquivo1.txt"

```
touch arquivo1.txt
```

5. Crie o diretório "tsi\_pasta4"

```
mkdir tsi_pasta4
```

6. Liste os arquivos no diretório e verifique as permissões padrões.

```
ls -l
```

```
(kali㉿kali)-[~/tsi_pasta1]
$ ls -l
total 4
-rw-r--r-- 1 kali kali    0 Mar 23 14:17 arquivo1.txt
-rw-r--r-- 1 kali kali    0 Mar 23 13:17 arquivo2.txt
drwxr-xr-x 2 kali kali 4096 Mar 23 14:18 tsi_pasta4
```

- Adicione a permissão Set User ID sobre o arquivo "arquivo1.txt".

```
chmod u+s arquivo1.txt
```

- Adicione a permissão Set Group ID sobre o diretório "tsi\_pasta4".

```
chmod -R g+s tsi_pasta4/
```

- Adicione a permissão Sticky Bit sobre o arquivo "arquivo2.txt".

```
chmod o+t arquivo2.txt
```

- Liste os arquivos e diretórios e verifique as permissões concedidas.

```
ls -l
```

```
(kali㉿kali)-[~/tsi_pasta1]
$ ls -l
total 4
-rwSr--r-- 1 kali kali    0 Mar 23 14:17 arquivo1.txt
-rw-r--r-T 1 kali kali    0 Mar 23 13:17 arquivo2.txt
drwxr-sr-x 2 kali kali 4096 Mar 23 14:18 tsi_pasta4
```

## Atividade Permissões Representação Binária

- Verifique qual o seu diretório corrente.

```
pwd → /home/kali/tsi_pasta1
```

- Acesse o diretório home do usuário. (cd ~)

```
cd ~
```

- Crie o diretório "tsi\_pasta5".

```
mkdir tsi_pasta5
```

- Acesse o diretório "tsi\_pasta5".

```
cd tsi_pasta5
```

- Crie os arquivos "arquivo5.txt", "arquivo6.txt" e "arquivo7.txt".

```
touch arquivo5.txt arquivo6.txt arquivo7.txt
```

- Crie o diretório "tsi\_pasta6".

```
mkdir tsi_pasta6
```

- Liste os arquivos e diretórios e verifique as permissões concedidas.

```
ls -l
```

```
(kali@kali)-[~/tsi_pasta5]
$ ls -l
total 4
-rw-r--r-- 1 kali kali    0 Mar 23 14:36 arquivo5.txt
-rw-r--r-- 1 kali kali    0 Mar 23 14:34 arquivo6.txt
-rw-r--r-- 1 kali kali    0 Mar 23 14:34 arquivo7.txt
drwxr-xr-x 2 kali kali 4096 Mar 23 14:36 tsi_pasta6
```

8. Adicione a permissões no arquivo "arquivo5.txt", usuário: escrita e execução; grupo: leitura e escrita, outros: leitura. (Utilizar a representação binária).

```
chmod 364 arquivo5.txt
```

9. Adicione a permissões no arquivo "arquivo6.txt", usuário: leitura e execução; grupo: leitura; outros: sem nenhuma permissão. (Utilizar a representação binária). -

```
chmod 540 arquivo6.txt
```

10. Adicione a permissões no arquivo "arquivo7.txt", usuário: leitura e escrita; grupo: leitura e execução, outros: execução. (Utilizar a representação binária). -

```
chmod 651 arquivo7.txt
```

11. Adicione a permissões no diretório "tsi\_pasta6", usuário: leitura, escrita e execução; grupo: leitura e escrita, outros: execução. (Utilizar a representação binária). -

```
chmod -R 761 tsi_pasta6/
```

12. Liste os arquivos e diretórios e verifique as permissões concedidas. -

```
ls -l
```

```
(kali@kali)-[~/tsi_pasta5]
$ ls -l
total 4
--wxrw-r-- 1 kali kali    0 Mar 23 14:36 arquivo5.txt
-r-xr----- 1 kali kali    0 Mar 23 14:34 arquivo6.txt
-rw-r-x--x 1 kali kali    0 Mar 23 14:34 arquivo7.txt
drwxrw---x 2 kali kali 4096 Mar 23 14:36 tsi_pasta6
```

## Atividade Modificar Proprietário

1. Verifique qual o seu diretório corrente.

```
pwd → /home/kali/tsi_pasta5
```

2. Acesse o diretório home do usuário. (cd ~)

```
cd ~
```

3. Acesse o diretório "tsi\_pasta1".

```
cd tsi_pasta1
```

4. Liste os arquivos e diretórios e verifique o proprietário do arquivo.

```
ls -l
```

```
(kali@kali)-[~/tsi_pasta1]
$ ls -l
total 4
-rwSr--r-- 1 kali kali 0 Mar 23 14:17 arquivo1.txt
-rw-r--r-T 1 kali kali 0 Mar 23 13:17 arquivo2.txt
drwxr-sr-x 2 kali kali 4096 Mar 23 14:18 tsi_pasta4
```

5. Altere o proprietário do arquivo "arquivo1.txt" para "user\_tsi" e grupo para "grupo\_tsi".

```
sudo useradd user_tsi
```

```
sudo groupadd grupo_tsi
```

```
sudo chown user_tsi:grupo_tsi arquivo1.txt
```

6. Altere o proprietário do diretório "tsi\_pasta4" para "user\_tsi" e grupo para "grupo\_tsi".

```
sudo chown -R user_tsi:grupo_tsi tsi_pasta4/
```

7. Liste os arquivos e diretórios e verifique o proprietário do arquivo.

```
ls -l
```

```
(kali@kali)-[~/tsi_pasta1]
$ ls -l
total 4
-rw-r--r-- 1 user_tsi grupo_tsi 0 Mar 23 14:17 arquivo1.txt
-rw-r--r-T 1 kali kali 0 Mar 23 13:17 arquivo2.txt
drwxr-sr-x 2 user_tsi grupo_tsi 4096 Mar 23 14:18 tsi_pasta4
```

## Atividade Lista de Controle de Acesso Estendida:

Nesse exercício, o roteiro dizia para utilizássemos um comando chamado **setfact**, porém ele não é encontrado. Após pesquisar, verifiquei que o comando correto é **setfacl**.



1. Verifique qual o seu diretório corrente.

```
pwd → /home/kali/tsi_pasta1
```

2. Acesse o diretório home do usuário. (cd ~)

```
cd ~
```

3. Crie um diretório "acl\_estendida1".

```
cd acl_estendida
```

4. Acesse o diretório "acl\_estendida1".

```
cd acl_estendida
```

5. Crie os arquivos "arquivos1.txt" e "arquivos2.txt".

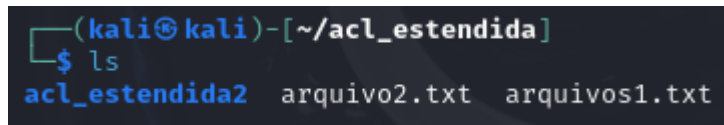
```
touch arquivos1.txt arquivos2.txt
```

6. Crie o diretório "acl\_estendida2".

```
mkdir acl_estendida2
```

7. Liste os arquivos e diretório para verificar as propriedades da pasta

```
ls
```



```
(kali㉿kali)-[~/acl_estendida]
$ ls
acl_estendida2  arquivo2.txt  arquivos1.txt
```

8. Adicione permissão de leitura e escrita para o usuário "user\_tsi" utilizando a ACL estendida sobre o arquivo "arquivos1.txt".

```
setfacl -m u:user_tsi:rw arquivos1.txt
```

9. Adicione permissão de leitura e execução para o grupo, "grupo\_tsi" utilizando a ACL estendida sobre o arquivo "arquivos2.txt".

```
setfacl -m g:grupo_tsi:rx arquivos2.txt
```

10. Adicione permissão de leitura, escrita e execução para o usuário "user\_tsi" utilizando a ACL estendida sobre o diretório "acl\_estendida2".

```
setfacl -m u:user_tsi:rwx acl_estendida2
```

11. Liste os arquivos e diretório, verifique se foi adicionado um sinal de adição (+) ao lado dos objetos que possuem permissão de ACL estendida.

```
ls -l
```

```
(kali㉿kali)-[~/acl_estendida]
$ ls -l
total 4
drwxrwxr-x+ 2 kali kali 4096 Mar 23 20:36 acl_estendida2
-rw-rwxr--+ 1 kali kali    0 Mar 23 20:35 arquivo2.txt
-rw-rw-r--+ 1 kali kali    0 Mar 23 20:35 arquivos1.txt
```

12. Verifique a permissão do arquivo "arquivos1.txt" utilizando o comando getfacl.

```
getfacl arquivos1.txt
```

```
(kali㉿kali)-[~/acl_estendida]
$ getfacl arquivos1.txt
# file: arquivos1.txt
# owner: kali
# group: kali
user::rw-
user:user_tsi:rw-
group::r--
mask::rw-
other::r--
```

13. Verifique a permissão do arquivo "arquivos2.txt" utilizando o comando getfacl.

```
getfacl arquivos2.txt
```

```
(kali㉿kali)-[~/acl_estendida]
$ getfacl arquivo2.txt
# file: arquivo2.txt
# owner: kali
# group: kali
user::rw-
group::r--
group:group_tsi:rw-
mask::rw-
other::r--
```

14. Verifique a permissão do diretório "acl\_estendida2" utilizando o comando getfacl.

```
getfacl -R acl_estendida2/
```

```
(kali㉿kali)-[~/acl_estendida]
$ getfacl -R acl_estendida2/
# file: acl_estendida2/
# owner: kali
# group: kali
user::rwx
user:user_tsi:rwx
group::r-x
mask::rwx
other::r-x
```

15. Remova a ACL estendida definida sobre o arquivo "arquivos2.txt".

```
setfacl -b arquivos2.txt
```

16. Liste os arquivos e diretório, verifique se foi adicionado um sinal de adição (+) ao lado dos objetos que possuem permissão de ACL estendida.

```
ls -l
```

```
(kali㉿kali)-[~/acl_estendida]
$ ls -l
total 4
drwxrwxr-x+ 2 kali kali 4096 Mar 23 20:36 acl_estendida2
-rw-r--r-- 1 kali kali 0 Mar 23 20:35 arquivo2.txt
-rw-rw-r--+ 1 kali kali 0 Mar 23 20:35 arquivos1.txt
```

## Definindo Permissão em Arquivos Utilizando Python

Para definir permissões em arquivos utilizando o Python vamos utilizar o comando `chmod` da biblioteca "os". Este comando funciona de maneira semelhante as permissões definidas no Linux, porém a máscara é definida em formato octal. Existe uma segunda que fornece as permissões no formato correspondente, a biblioteca "stat". As permissões básicas definidas na biblioteca "stat" são listadas abaixo:

Comando	Descrição
stat.S_IRWXU	Leitura, escrita e execução pelo proprietário.
stat.S_IRUSR	Leitura pelo proprietário.
stat.S_IWUSR	Escrita pelo proprietário.
stat.S_IXUSR	Execução pelo proprietário.
stat.S_IRWXG	Leitura, escrita e execução pelo grupo.
stat.S_IRGRP	Leitura pelo grupo.

Comando	Descrição
stat.S_IWGRP	Escrita pelo grupo.
stat.S_IXGRP	Execução pelo grupo.
stat.S_IRWXO	Leitura, escrita e execução por outros
stat.S_IROTH	Leitura por outros
stat.S_IWOTH	– Escrita por outros
• stat.S_IXOTH	– Execução por outros

## ESPECIAIS

stat.S_ISUID	Execução como Set user ID.
stat.S_ISGID	Execução como Set group ID.
stat.S_ENFMT	Bloqueio de registro aplicado.
stat.S_ISVTX	Salva a imagem do texto após a execução.
stat.S_IREAD	Leitura pelo proprietário.
stat.S_IWRITE	Escrita pelo proprietário.
stat.S_IEXEC	Execução pelo proprietário

## Atividade Permissão Arquivos Pelo Python

Para auxiliar você nesta atividade baseie-se no código da figura acima ou no arquivo "Permissao.py".

1. Crie um programa em Python para armazenar a data e horário que o arquivo foi executado.
2. A informação da data e horário deve ser armazenada em um arquivo denominado "permissao.txt"
3. Sempre que o arquivo for executado você deve verificar se este arquivo existe.
4. Caso o arquivo exista você deve modificar a permissão deste arquivo para leitura, escrita e execução do proprietário.
5. Obtenha as informações de data e hora do sistema e armazene em uma variável
6. Abra o arquivo para escrita.

7. Grave as informações das variáveis no arquivo "permissão.txt".
8. Modifique as permissões do arquivo "permissão.txt" apenas para escrita.