Gerenciamento de permissões padrão e acesso aos arquivos

Permissões especiais

As *permissões especiais* são um quarto tipo de permissão além dos tipos usuário, grupo e outros. Como o nome indica, as permissões especiais fornecem recursos relacionados a acesso além do que os tipos básicos de permissão permitem. Esta seção descreve o impacto das permissões especiais, que estão resumidas na tabela a seguir.

Tabela 7.2. Efeitos das permissões especiais em arquivos e diretórios

Permissão	Efeitos em arquivos	Efeitos em diretórios
u+s (suid)	O arquivo é executado como o usuário proprietário do arquivo, não como o que o executou.	Sem efeito.
`g+s` (sgid)	O arquivo é executado como o grupo proprietário do arquivo.	Arquivos criados no diretório têm um proprietário de grupo que corresponde ao proprietário do grupo do diretório.
`o+t` (sticky)	Sem efeito.	Os usuários com acesso de gravação no diretório só podem apagar arquivos que pertencem a si mesmos. Eles não podem apagar ou forçar o salvamento de arquivos de outros usuários.

A permissão *setuid* em um arquivo executável significa que os comandos são executados como o usuário que é proprietário desse arquivo, e não o usuário que executou o comando. Um exemplo seria o comando `passwd`:

```
[user@host ~]$ls -l /usr/bin/passwd
```

```
-rwsr-xr-x. 1 root root 35504 Jul 16  2010 /usr/bin/passwd
```

Em uma listagem longa, você pode identificar as permissões `setuid` por um caractere em letra minúscula `s` onde você normalmente esperaria que houvesse um caractere `x` (o proprietário executa permissões). Se o proprietário não tiver permissões de execução, esse caractere é substituído por uma letra maiúscula `S`.

A permissão especial *setgid* em um diretório significa que os arquivos criados no diretório herdam a propriedade do grupo do diretório em vez de herdá-la do usuário que o criou.

Esse recurso é frequentemente usado em diretórios colaborativos de grupo para alterar de maneira automática um arquivo do grupo privado padrão para o grupo compartilhado ou se um grupo específico precisar sempre ser o proprietário de arquivos em um diretório. Um exemplo desse comportamento é o diretório `/run/log/journal`:

```
[user@host ~]$ls -ld /run/log/journal
drwxr-sr-x. 3 root systemd-journal 60 May 18 09:15 /run/lo
g/journal
```

Se `setgid` estiver definido em um arquivo executável, os comandos serão executados como o grupo que é proprietário desse arquivo, e não o usuário que executou o comando. Essa condição é semelhante à maneira como `setuid` funciona. Um exemplo seria o comando `locate`:

```
[user@host ~]$ls -ld /usr/bin/locate
-rwx--s--x. 1 root slocate 47128 Aug 12 17:17 /usr/bin/loca
te
```

Em uma listagem longa, você pode identificar as permissões `setgid` por um caractere em letra minúscula `s` onde você normalmente esperaria que houvesse um caractere `x` (o grupo executa permissões). Se o grupo não tiver permissões de execução, esse caractere é substituído por uma letra maiúscula `s`.

Por fim, o *sticky bit* para um diretório define uma restrição especial na exclusão de arquivos. Apenas o proprietário do arquivo (e o usuário `root`) consegue excluir arquivos dentro do diretório. Um exemplo é o diretório `/tmp`:

```
[user@host ~]$ls -ld /tmp
drwxrwxrwt. 39 root root 4096 Feb  8 20:52 /tmp
```

Em uma listagem longa, você pode identificar as permissões sticky por um caractere em letra minúscula `t` onde você normalmente esperaria que houvesse um caractere `x` (outro executa permissões). Se o outro não tiver permissões de execução, esse caractere é substituído por uma letra maiúscula `T`.

Definição de permissões especiais

- **Simbolico =** `setuid` = `u+s`; `setgid` = `g+s`; sticky = `o+t`
- **Octal =** no quarto dígito anterior adicionado; `setuid` = 4; `setgid` = 2; sticky = 1

Exemplos de permissões especiais

Adicione o bit `setgid` no diretório `example` usando o método simbólico:

```
chmod g+s example
```

Remova o bit `setuid` no diretório `example` usando o método simbólico:

```
chmod u-s example
```

Defina o bit `setgid` e adicione permissões de leitura, gravação e execução ao usuário e grupo, sem acesso para outros, no diretório `example` usando o método octal:

```
chmod 2770 example
```

Remova o bit `setgid` e adicione permissões de leitura, gravação e execução ao usuário e grupo, sem acesso para outros, no diretório `example` usando o método octal. Observe que você precisa adicionar um `0` adicional no início do valor de permissões ao remover permissões especiais usando o método octal:

```
chmod 0770 example
```

Permissões de arquivo padrão

Ao ser criado, um arquivo recebe permissões iniciais. Dois fatores afetam essas permissões iniciais:

- A primeira é se você está criando um arquivo regular ou um diretório
- O segundo é o `umask` atual, o que significa máscara de criação de arquivos do usuário.

Se você criar um diretório, suas permissões octais iniciais serão 0777 (`drwxrwxrwx`).

Se você criar um arquivo regular, suas permissões octais iniciais serão 0666 (`-rw-rw-rw-`).

Você sempre precisa incluir explicitamente a permissão de execução em um arquivo regular. Essa etapa torna mais difícil para um invasor comprometer um sistema, criar um arquivo malicioso e executá-lo.

Umask

Além disso, a sessão do shell define um umask para restringir ainda mais as permissões iniciais de um arquivo.

O umask é um bitmask octal que limpa as permissões de novos arquivos e diretórios que um processo cria. Se um bit for definido no umask, a permissão correspondente será desmarcada nos novos arquivos.

Por exemplo, o umask 0002, limpa o bit de gravação para outros usuários.

Os zeros à esquerda indicam que as permissões especial, de usuário e de grupo não são limpas.

O umask de 0077 limpa todo o grupo e outras permissões de arquivos recentemente criados.

O comando `umask` sem argumentos exibe o valor atual do umask do shell:

```
[user@host ~]$umask
0022
```

Use o comando `umask` com apenas um argumento octal para alterar o umask do shell atual. O argumento deve ser um valor octal correspondente ao novo valor de umask. Você pode omitir zeros à esquerda no umask. Por exemplo, `umask 077` é o mesmo que `umask 0077`.

Os valores de umask padrão do sistema para usuários do shell Bash estão definidos nos arquivos `/etc/login.defs` e `/etc/bashrc`.

Os usuários podem substituir os padrões do sistema em seus arquivos `.bash_profile` ou `.bashrc` nos diretórios pessoais.

Efeito do utilitário umask nas permissões

O exemplo a seguir explica como o umask afeta as permissões de arquivos e diretórios. Observe as permissões padrão do umask para os arquivos e diretórios no shell atual.

Os exemplos a seguir pressupõem que o umask do shell esteja definido como 0022.

Se você criar um arquivo regular, suas permissões octais iniciais serão 0666 (000 110 110 110, na representação binária).

Em seguida, o umask 0022 (000 000 010 010) desativa o bit de permissão de gravação para o grupo e outros. Por isso, o proprietário tem permissões de leitura e gravação nos arquivos, e o grupo e outros estão configurados para leitura (000 110 100 100)

	Symbolic	Numeric octal	Numeric binary
Initial file permissions	rw-rw-rw-	0666	000 110 110 110
umask	----w--w-	0022	000 000 010 010
Resulting file permissions	rw-r--r--	0644	000 110 100 100

```
[user@host ~]$umask
0022
[user@host ~]$touch default.txt
[user@host ~]$ls -l default.txt
-rw-r--r--. 1 user user 0 May  9 01:54 default.txt
```

Se você criar um diretório, suas permissões octais iniciais serão 0777 (000 111 111 111).

Em seguida, o umask 0022 (000 000 010 010) desativa o bit de permissão de gravação para o grupo e outros.

Por isso, o proprietário tem permissões de leitura, gravação e execução nos diretórios e o grupo e outros estão configurados para leitura e execução (000 111 101 101).

	Symbolic	Numeric octal	Numeric binary
Initial directory permissions	rwxrwxrwx	0777	000 111 111 111
umask	----w--w-	0022	000 000 010 010
Resulting directory permissions	rwxr-xr-x	0755	000 111 101 101

```
[user@host ~]$umask
0022
[user@host ~]$mkdir default
```

```
[user@host ~]$ls -ld default
drwxr-xr-x. 2 user user 0 May  9 01:54 default
```

Definindo o valor umask como 0, as permissões de arquivo para outras alterações são de leitura a leitura e gravação. As permissões do diretório para outra alteração de leitura e execução a leitura, gravação e execução.

```
[user@host ~]$umask 0
[user@host ~]$touch zero.txt
[user@host ~]$ls -l zero.txt
-rw-rw-rw-. 1 user user 0 May  9 01:54 zero.txt
[user@host ~]$mkdir zero
[user@host ~]$ls -ld zero
drwxrwxrwx. 2 user user 0 May  9 01:54 zero
```

Para mascarar todas as permissões de arquivo e diretório para outros, defina o valor umask como 007.

```
[user@host ~]$umask 007
[user@host ~]$touch seven.txt
[user@host ~]$ls -l seven.txt
-rw-rw----. 1 user user 0 May  9 01:55 seven.txt
[user@host ~]$mkdir seven
[user@host ~]$ls -ld seven
drwxrwx---. 2 user user 0 May  9 01:54 seven
```

Um umask com o valor de 027 garante que os novos arquivos tenham permissões de leitura e gravação para usuário e permissão de leitura para grupo. Novos diretórios têm permissões de leitura e execução para o grupo e nenhuma permissão para outros.

```
[user@host ~]$umask 027
[user@host ~]$touch two-seven.txt
[user@host ~]$ls -l two-seven.txt
-rw-r-----. 1 user user 0 May  9 01:55 two-seven.txt
[user@host ~]$mkdir two-seven
```

```
[user@host ~]$ls -ld two-seven
drwxr-x---. 2 user user 0 May  9 01:54 two-seven
```

Alteração das permissões padrão

No Red Hat Enterprise Linux 9, o arquivo `/etc/login.defs` define o umask padrão para os usuários. Por padrão, a linha `UMASK` especifica que o umask padrão é 0022.

O usuário `root` pode alterar o umask padrão para shells interativos sem login adicionando um script de inicialização do shell `local-umask.sh` no diretório `/etc/profile.d/`. O seguinte exemplo mostra um arquivo `local-umask.sh`:

```
[root@host ~]#cat /etc/profile.d/local-umask.sh
# Overrides default umask configuration asda sda
if [ $UID -gt 199 ] && [ "`id -gn`" = "`id -un`" ]; then
    umask 007
else
    umask 022
fi
```

O exemplo anterior configura o umask como 0007 para usuários com uma UID maior que 199 e com um nome de usuário e um nome de grupo primário correspondentes e como 0022 para todos os demais. (Os zeros iniciais podem ser omitidos.) Para definir o umask de todos como 0022, crie esse arquivo com o seguinte conteúdo:

```
# Overrides default umask configuration
umask 022
```

O umask atual de um shell se aplica até que você faça o logout do shell e faça login novamente ou até que você o altere manualmente com o comando `umask`.

Exercício orientado: Gerenciamento de permissões padrão e acesso aos arquivos

Neste exercício, você controla as permissões de arquivos criados em um diretório usando as configurações de umask e a permissão `setgid`.

Resultados

- Criar um diretório compartilhado no qual o grupo `operators` automaticamente é proprietário de novos arquivos.
- Experimentar com várias configurações de umask.
- Ajustar as permissões padrão para usuários específicos.
- Verifique a alteração.

Com o usuário `student` na máquina `workstation`, use o comando `lab` para preparar seu sistema para este exercício.

Esse comando prepara seu ambiente e garante que todos os recursos necessários estejam disponíveis.

```
[student@workstation ~]$lab start perms-default
```

Instruções

1. Faça login no sistema `servera` como o usuário `student`.

   ```
   [student@workstation ~]$ssh student@servera...output omi
   tted...
   [student@servera ~]$
   ```

2. Alterne para o usuário `operator1` usando `redhat` como senha.

   ```
   [student@servera ~]$su - operator1
   Password:redhat
   [operator1@servera ~]$
   ```

3. Liste o valor umask padrão do usuário `operator1`.

   ```
   [operator1@servera ~]$umask
   0022
   ```

4. Crie um novo diretório `/tmp/shared`. No diretório `/tmp/shared`, crie um arquivo `defaults`. Observe as permissões padrão.

a. Crie o diretório `/tmp/shared`. Liste as permissões do novo diretório.

```
[operator1@servera ~]$mkdir /tmp/shared
[operator1@servera ~]$ls -ld /tmp/shared
drwxr-xr-x. 2 operator1 operator1 6 Feb  4 14:06 /tm
p/shared
```

b. Crie um arquivo `defaults` no diretório `/tmp/shared`.

```
[operator1@servera ~]$touch /tmp/shared/defaults
```

c. Liste as permissões do novo arquivo.

```
[operator1@servera ~]$ls -l /tmp/shared/defaults
-rw-r--r--. 1 operator1 operator1 0 Feb  4 14:09 /tm
p/shared/defaults
```

5. Altere a propriedade do grupo do diretório `/tmp/shared` para o grupo `operators`. Confirme a nova propriedade e as permissões.

a. Altere a propriedade do grupo do diretório `/tmp/shared` para o grupo `operators`.

```
[operator1@servera ~]$chown :operators /tmp/shared
```

b. Liste as permissões do diretório `/tmp/shared`.

```
[operator1@servera ~]$ls -ld /tmp/shared
drwxr-xr-x. 2 operator1 operators 22 Feb  4 14:09 /tm
p/shared
```

c. Crie um arquivo `group` no diretório `/tmp/shared`. Liste as permissões do arquivo.

```
[operator1@servera ~]$touch /tmp/shared/group
[operator1@servera ~]$ls -l /tmp/shared/group
-rw-r--r--. 1 operator1 operator1 0 Feb  4 17:00 /tm
p/shared/group
```

Nota

O proprietário do grupo do arquivo `/tmp/shared/group` não é `operators`, mas `operator1`.

6. Certifique-se de que o grupo `operators` é proprietário de arquivos criados no diretório `/tmp/shared`.
   a. Defina a ID do grupo do grupo `operators` para o diretório `/tmp/shared`.

```
[operator1@servera ~]$chmod g+s /tmp/shared
```

   b. Crie um arquivo `ops_db.txt` no diretório `/tmp/shared`.

```
[operator1@servera ~]$touch /tmp/shared/ops_db.txt
```

   c. Verifique se o grupo `operators` é o proprietário do grupo para o novo arquivo.

```
[operator1@servera ~]$ls -l /tmp/shared/ops_db.txt
-rw-r--r--. 1 operator1 operators 0 Feb  4 16:11 /tm
p/shared/ops_db.txt
```

7. Crie um arquivo `ops_net.txt` no diretório `/tmp/shared`. Registre a propriedade e as permissões. Altere o umask para o usuário `operator1`. Crie um arquivo `ops_prod.txt`. Registre a propriedade e as permissões do arquivo `ops_prod.txt`.
   a. Crie um arquivo `ops_net.txt` no diretório `/tmp/shared`.

```
[operator1@servera ~]$touch /tmp/shared/ops_net.txt
```

   b. Liste as permissões do arquivo `ops_net.txt`.

```
[operator1@servera ~]$ls -l /tmp/shared/ops_net.txt
-rw-r--r--. 1 operator1 operators 5 Feb  0 15:43 /tm
p/shared/ops_net.txt
```

   c. Altere o umask do usuário `operator1` para 027. Confirme a alteração.

```
[operator1@servera ~]$umask 027
[operator1@servera ~]$umask
0027
```

d. Crie um arquivo `ops_prod.txt` no diretório `/tmp/shared/`. Verifique se os arquivos criados recentemente têm acesso somente leitura para o grupo `operators` e não têm acesso para outros usuários.

```
[operator1@servera ~]$touch /tmp/shared/ops_prod.txt
[operator1@servera ~]$ls -l /tmp/shared/ops_prod.txt
-rw-r-----. 1 operator1 operators 0 Feb  0 15:56 /tm
p/shared/ops_prod.txt
```

8. Abra uma nova janela de terminal e faça login no `servera` como `operator1`.

```
[student@workstation ~]$ssh operator1@servera...output o
mitted...
[operator1@servera ~]$
```

9. Liste o valor de umask para `operator1`.

```
[operator1@servera ~]$umask
0022
```

10. Altere o umask padrão para o usuário `operator1`. O novo umask proíbe todo o acesso de usuários que não estejam no grupo. Confirme se o umask foi alterado.

a. Altere o umask padrão do usuário `operator1` para 007.

```
[operator1@servera ~]$echo "umask 007" >> ~/.bashrc
[operator1@servera ~]$cat ~/.bashrc
# .bashrc

# Source global definitions
if [ -f /etc/bashrc ]; then
    . /etc/bashrc
```

```
fi
...output omitted...umask 007
```

b. Faça o logout e faça login novamente como o usuário `operator1`. Confirme que a alteração é permanente.

```
[operator1@servera ~]$exit
logout
Connection to servera closed.
[student@workstation ~]$ssh operator1@servera...outpu
t omitted...
[operator1@servera ~]$umask
0007
```

11. Crie um arquivo `ops_prod2.txt` no diretório `/tmp/shared/`. Verifique se os arquivos criados recentemente têm acesso leitura e gravação para o grupo `operators` e não têm acesso para outros usuários devido ao novo umask de 007.

```
[operator1@servera ~]$touch /tmp/shared/ops_prod2.txt
[operator1@servera ~]$ls -l /tmp/shared/ops_prod2.txt
-rw-rw----. 1 operator1 operators 0 Feb  0 15:56 /tmp/sh
ared/ops_prod2.txt
```

12. Em `servera`, feche todos os shells de usuário `operator1` e `student`. Retorne ao sistema `workstation` como o usuário `student`.

Atenção

Não sair de todos os shells `operator1` fará com que o script final falhe.

```
[operator1@servera ~]$exit
logout
Connection to servera closed.
[student@workstation ~]$
```

Encerramento

Na máquina `workstation`, altere para o diretório pessoal do usuário `student` e use o comando `lab` para concluir este exercício. Essa etapa é importante para

garantir que recursos de exercícios anteriores não afetem exercícios futuros.

```
[student@workstation ~]$lab finish perms-default
```

Isso conclui a seção.

Laboratório Aberto: Controlar acesso a arquivos

Neste laboratório, você configura permissões em arquivos e define um diretório que os usuários em um determinado grupo poderão usar para compartilhar arquivos no sistema de arquivos local.

Resultados

- Criar um diretório no qual os usuários possam trabalhar colaborativamente nos arquivos.
- Criar arquivos que são atribuídos automaticamente à propriedade do grupo.
- Criar arquivos que não são acessíveis fora do grupo.

Com o usuário `student` na máquina `workstation`, use o comando `lab` para preparar seu sistema para este exercício.

Esse comando prepara seu ambiente e garante que todos os recursos necessários estejam disponíveis.

```
[student@workstation ~]$lab start perms-review
```

Instruções

1. Faça login no `serverb` como o usuário `student`. Execute o comando `sudo -i` no prompt do shell para se tornar o usuário `root`. Use `student` como a senha do usuário `student`.

   ```
   [student@workstation ~]$ ssh student@serverb ...output omitted...
   [student@serverb ~]$
   sudo -i
   [sudo] password for student:
   student
   [root@serverb ~]#
   ```

   Ocultar solução

2. Crie um novo diretório `/home/techdocs`.

1. Use o comando `mkdir` para criar um diretório `/home/techdocs`.

```
[root@serverb ~]# mkdir /home/techdocs
```

Ocultar solução

3. Altere a propriedade do grupo do diretório `/home/techdocs` para o grupo `techdocs`.

   1. Use o comando `chown` para alterar a propriedade de grupo do diretório `/home/techdocs` para o grupo `techdocs`.

```
[root@serverb ~]# chown :techdocs /home/techdocs
```

   Ocultar solução

4. Verifique se os usuários no grupo `techdocs` não podem criar arquivos no diretório `/home/techdocs`.

   1. Use o comando `su` para alternar para o usuário `tech1`.

```
[root@serverb ~]# su - tech1
[tech1@serverb ~]$
```

   2. Crie um arquivo `techdoc1.txt` no diretório `/home/techdocs`. Essa etapa deve falhar. Embora o diretório `/home/techdocs` seja de propriedade do grupo `techdocs` e `tech1` seja parte do grupo `techdocs`, não é possível criar um novo arquivo nesse diretório. Isso ocorre porque o grupo `techdocs` não tem permissão de gravação.

```
[tech1@serverb ~]$ touch /home/techdocs/techdoc1.txt
touch: cannot touch '/home/techdocs/techdoc1.txt': Permission denied
```

   3. Liste as permissões do diretório.

```
[tech1@serverb ~]$ ls -ld /home/techdocs/
 drwxr-xr-x. 2 root techdocs 6 Feb  5 16:05 /home/techdocs/
```

   Ocultar solução

5. Defina as permissões no diretório `/home/techdocs`. No diretório `/home/techdocs`, configure `setgid` (2), permissões de leitura, gravação e execução (7) para o proprietário/usuário e grupo e sem permissões (0) para outros usuários.

   1. Saia do shell do usuário `tech1`.

```
[tech1@serverb ~]$ exit
logout
[root@serverb ~]#
```

   2. Defina as permissões de grupo para o diretório `/home/techdocs`. Configure `setgid`, permissões de leitura, gravação e execução para o proprietário e grupo e sem permissões para outros.

```
[root@serverb ~]# chmod 2770 /home/techdocs
```

   Ocultar solução

6. Verifique se as permissões foram definidas corretamente.

   O grupo `techdocs` agora tem permissão de gravação.

```
[root@serverb ~]# ls -ld /home/techdocs
drwxrws---. 2 root techdocs 6 Feb 4 18:12 /home/techdocs/
```

   Ocultar solução

7. Confirme se os usuários no grupo `techdocs` agora conseguem criar e editar arquivos no diretório `/home/techdocs`. Os usuários que não estão no grupo `techdocs` não conseguem editar ou criar arquivos no diretório `/home/techdocs`. Os usuários `tech1` e `tech2` estão no grupo `techdocs`. O usuário `database1` não está nesse grupo.

   1. Mude para o usuário `tech1`. Crie um arquivo `techdoc1.txt` no diretório `/home/techdocs`. Adicione texto ao arquivo `/home/techdocs/techdoc1.txt`. Saia do shell do usuário `tech1`.

```
[root@serverb ~]# su - tech1
[tech1@serverb ~]$
touch /home/techdocs/techdoc1.txt
[tech1@serverb ~]$
ls -l /home/techdocs/techdoc1.txt
-rw-r--r--. 1 tech1 techdocs 0 Feb  5 16:42 /home/techdocs/techdoc1.txt
[tech1@serverb ~]$
echo "This is the first tech doc." > /home/techdocs/techdoc1.txt
```

```
[tech1@serverb ~]$
exit
logout
[root@serverb ~]#
```

2. Mude para o usuário `tech2`. Exiba o conteúdo do arquivo `/home/techdocs/techdoc1.txt`. Crie um arquivo `techdoc2.txt` no diretório `/home/techdocs`. Saia do shell do usuário `tech2`.

```
[root@serverb ~]# su - tech2
[tech2@serverb ~]$
cd /home/techdocs
[tech2@serverb techdocs]$
cat techdoc1.txt
This is the first tech doc.
[tech2@serverb techdocs]$
touch /home/techdocs/techdoc2.txt
[tech2@serverb techdocs]$
ls -l
total 4
-rw-r--r--. 1 tech1 techdocs 28 Feb  5 17:43 techdoc1.txt
-rw-r--r--. 1 tech2 techdocs  0 Feb  5 17:45 techdoc2.txt
[tech2@serverb techdocs]$
exit
logout
[root@serverb ~]#
```

3. Mude para o usuário `database1`. Exiba o conteúdo do arquivo `/home/techdocs/techdoc1.txt`. Você recebe uma mensagem `Permission Denied`. Verifique se o usuário `database1` não tem acesso ao arquivo. Saia do shell do usuário `database1`.
Digite o seguinte comando longo `echo` em uma única linha:

```
[root@serverb ~]# su - database1
[database1@serverb ~]$
cat /home/techdocs/techdoc1.txt
cat: /home/techdocs/techdoc1.txt: Permission denied
[database1@serverb ~]$
ls -l /home/techdocs/techdoc1.txt
ls: cannot access '/home/techdocs/techdoc1.txt': Permission denied
[database1@serverb ~]$
exit
```

```
logout
[root@serverb ~]#
```

Ocultar solução

8. Modifique o arquivo `/etc/login.defs` para ajustar o umask padrão para shells de login. Os usuários normais devem ter uma configuração umask que permita ao usuário e ao grupo criar, gravar e executar arquivos e diretórios, e evita que outros usuários visualizem, modifiquem ou executem novos arquivos e diretórios.

   1. Determine o umask do usuário `student`. Alterne para o shell de login `student`. Ao terminar, saia do shell.

```
[root@serverb ~]# su - student
[student@serverb ~]$ umask
0022
[student@serverb ~]$ exit
logout
[root@serverb ~]#
```

   2. Edite o arquivo `/etc/login.defs` e defina um umask de `007`. O arquivo `/etc/login.defs` já contém uma definição de umask. Pesquise o arquivo e atualize com os valor apropriado.

```
[root@serverb ~]# cat /etc/login.defs ...output omitted...
UMASK           007

...output omitted...
```

   3. Como o usuário `student`, verifique se o umask global muda para `007`.

```
[root@serverb ~]# exit
logout
[student@serverb ~]$ exit
logout
Connection to serverb closed.
[student@workstation ~]$ ssh student@serverb ...output omitted...
[student@serverb ~]$ umask
```

`0007`

4. Retorne ao sistema `workstation` como o usuário `student`.

```
[student@serverb ~]$ exit
logout
Connection to serverb closed.
[student@workstation ~]$
```

Ocultar solução

Avaliação

Com o usuário `student` na máquina `workstation`, use o comando `lab` para avaliar seu trabalho. Corrija todas as falhas relatadas e execute novamente o comando até que ele seja concluído com êxito.

```
[student@workstation ~]$lab grade perms-review
```

Encerramento

Na máquina `workstation`, altere para o diretório pessoal do usuário `student` e use o comando `lab` para concluir este exercício. Essa etapa é importante para garantir que recursos de exercícios anteriores não afetem exercícios futuros.

```
[student@workstation ~]$lab finish perms-review
```

Isso conclui a seção.