

Laboratório Aberto - Gerenciamento de usuários e grupos, permissões e processos

Especificações

- Faça login no `serverb` como o usuário `student`.
- Identifique e encerre o processo que atualmente usa o maior tempo da CPU.
- Crie o grupo `database` com uma GID de 50.000.
- Crie o usuário `dbadmin1` e configure-o com os seguintes requisitos:
 - Adicione o grupo `database` como um grupo suplementar.
 - Defina a senha como `redhat` e force uma alteração de senha no primeiro login.
 - Permita que a senha seja alterada 10 dias após o último dia da alteração da senha.
 - Defina a expiração da senha para 30 dias após o último dia da alteração da senha.
 - Permita o usuário use o comando `sudo` para executar qualquer comando como o superusuário.
 - Configure o umask padrão como 007 para o usuário `dbadmin`.
- Crie o diretório `/home/dbadmin1/grading/review2` com `dbadmin1` como o usuário proprietário e o grupo `database` como o grupo proprietário.
- Configure o diretório `/home/dbadmin1/grading/review2` para que o grupo `database` seja proprietário de qualquer arquivo ou subdiretório criado nesse diretório, independentemente de qual usuário criou o arquivo. Configure as permissões no diretório para permitir que os membros do grupo `database` acessem o diretório e criem conteúdo nele. Todos os outros usuários devem ter permissões de leitura e execução no diretório.

- Certifique-se de que os usuários têm permissão para excluir somente arquivos de sua propriedade do diretório `/home/dbadmin1/grading/review2`.

Faça login no `serverb` como o usuário `student`.

```
[student@workstation ~]$ ssh student@serverb ...output omitted...
[student@serverb ~]$
```

Identifique e encerre o processo que atualmente usa o maior tempo da CPU.

1. Use o comando

`top` para ver o consumo da CPU do sistema em tempo real.

```
[student@serverb ~]$ top
```

2. Na interface interativa do comando

`top`, preste atenção à coluna `%CPU` e confirme que existe um processo `dd` consumindo a maioria dos recursos da CPU.

```
...output omitted...
PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM     TIME+ COMMAND
2303 student    20   0  217048   944   876  R   99.7    0.1  100:11.64 dd ...output
omitted...
```

O processo

`dd` na saída anterior tem a PID 2303. Esse processo está consumindo 99,7% dos recursos da CPU. A PID e a porcentagem de consumo de recursos da CPU podem variar em seu sistema.

3. Na interface interativa do comando

`top`, digite **k** para encerrar o processo `dd` com a PID 2303, como você determinou na etapa anterior. Depois de digitar **k** no comando `top`, se a PID padrão exibida no prompt corresponder à PID do processo que será encerrado, pressione a tecla **Enter**. Se a PID sugerida não corresponder, especifique a PID interativamente.

```
...output omitted...
PID to signal/kill [default pid =
2303 ] Enter ...output omitted...
```

4. Use o sinal padrão

`SIGTERM` para encerrar o processo.

```
...output omitted...  
Send pid 2833 signal [15/sigterm]  
Enter ...output omitted...
```

5. Pressione a tecla

q para sair da interface interativa do comando `top`. Crie o grupo `database` com uma GID de 50.000.

1. Mude para o usuário

```
root.
```

```
[student@serverb ~]$ sudo -i  
[sudo] password for student:  
student  
[root@serverb ~]#
```

2. Crie o grupo

`database` com uma GID de 50.000.

`[root@serverb ~]# groupadd -g 50000 database` Crie o usuário `dbadmin1`. Adicione o grupo `database` como um grupo suplementar. Defina a senha como `redhat` e force uma alteração de senha no primeiro login do usuário. Permita que a senha seja alterada 10 dias após o último dia da alteração da senha. Defina a expiração da senha para 30 dias após o último dia da alteração da senha. Permita o usuário use o comando `sudo` para executar qualquer comando como o superusuário. Configure o umask padrão como `007`.

1. Crie o usuário

`dbadmin1`. Adicione o grupo `database` como um grupo suplementar.

```
[root@serverb ~]# useradd -G database dbadmin1
```

2. Defina a senha do usuário

`dbadmin1` como `redhat`.

```
[root@serverb ~]# passwd dbadmin1  
Changing password for user dbadmin1.  
New password:  
redhat  
BAD PASSWORD: The password is shorter than 8 characters  
Retype new password:  
redhat
```

```
passwd: all authentication tokens updated successfully.
```

3. Force o usuário

`dbadmin1` a alterar sua senha na próxima vez que fizer login.

```
[root@serverb ~]# chage -d 0 dbadmin1
```

4. Defina a idade mínima da senha do usuário

`dbadmin1` para `10` dias.

```
[root@serverb ~]# chage -m 10 dbadmin1
```

5. Defina a idade máxima da senha do usuário

`dbadmin1` para `30` dias.

```
[root@serverb ~]# chage -M 30 dbadmin1
```

6. Permita o usuário

`dbadmin1` use o comando `sudo` para executar qualquer comando como o superusuário. Use o comando `vim /etc/sudoers.d/dbadmin1` para criar o arquivo e adicionar seguinte conteúdo:

```
[root@serverb ~]# vim /etc/sudoers.d/dbadmin1
dbadmin1 ALL=(ALL) ALL
```

7. Mude para o usuário

`dbadmin1` . Anexe a linha `umask 007` ao arquivo `/home/dbadmin1/.bashrc` .

```
[root@serverb ~]# su - dbadmin1
[dbadmin1@serverb ~]$
echo "umask 007" >> .bashrc
```

8. Obtenha o arquivo

`~/.bashrc` para atualizar o umask.

`[dbadmin1@serverb ~]$ source ~/.bashrc` Crie o diretório `/home/dbadmin1/grading/review2` com `dbadmin1` como o usuário proprietário e o grupo `database` como o grupo proprietário.

1. Use a opção

`-p` do comando `mkdir` para criar o diretório `/home/dbadmin1/grading/review2`.

```
[dbadmin1@serverb ~]$ mkdir -p /home/dbadmin1/grading/review2
```

2. Recursivamente, defina

`dbadmin1` e `database` como o respectivo usuário proprietário e o grupo dos subdiretórios e do diretório `/home/dbadmin1/`.

```
[dbadmin1@serverb ~]$ chown -R dbadmin1:database /home/dbadmin1/
```

3. Defina recursivamente as permissões do grupo

`execute` no diretório `/home/dbadmin1` e nos subdiretórios. Essa permissão possibilita que os membros do grupo `database` atravessem a estrutura de diretórios `/home/dbadmin1`.

```
[dbadmin1@serverb ~]$ chmod -R g+x /home/dbadmin1
```

 Configure o diretório `/home/dbadmin1/grading/review2` para permitir que os membros do grupo `database` criem conteúdo nele. Todos os outros usuários devem ter permissões de leitura e execução no diretório.

1. Aplique a permissão especial SetGID ao diretório

`/home/dbadmin1/grading/review2` para que os arquivos criados no diretório sejam de propriedade do grupo `database`.

```
[dbadmin1@serverb ~]$ chmod g+s /home/dbadmin1/grading/review2
```

2. Aplique o modo de permissão

`775` no diretório `/home/dbadmin1/grading/review2`.

```
[dbadmin1@serverb ~]$ chmod 775 /home/dbadmin1/grading/review2
```

 Certifique-se de que os usuários têm permissão para excluir somente arquivos de sua propriedade do diretório `/home/dbadmin1/grading/review2`.

1. Aplique a permissão especial sticky bit no diretório

`/home/dbadmin1/grading/review2`.

```
[dbadmin1@serverb ~]$ chmod o+t /home/dbadmin1/grading/review2
```

2. Retorne ao sistema

`workstation` como o usuário `student`.

```
[dbadmin1@serverb ~]$ exit
logout
[root@serverb ~]#
exit
logout
[student@serverb ~]$
exit
logout
Connection to serverb closed.
```

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 rhcsa-rh124-review2
```

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 rhcsa-rh124-review2
```

Isso conclui a seção.