Capitulo 6. Gerenciar usuarios e grupos locais

O que é um usuário?

Uma conta de *usuário* fornece limites de segurança entre pessoas e programas que podem executar comandos.

Internamente, o sistema distingue as contas de usuários pelo número de identificação exclusivo, a ID do usuário ou *UID*, atribuído a elas.

Todo arquivo tem um usuário específico como seu proprietário. Com a propriedade de arquivo, o sistema impõe o controle de acesso aos usuários dos arquivos. O usuário associado a um processo em execução determina os arquivos e diretórios acessíveis a esse processo.

Tipos de usuarios:

- SuperUsuario = administra o sistema ou seja total acesso ao sistema, é chamado de root e tem um UID 0
- Daemon (processos) = fornece serviços de suporte, não precisa ser executado como root. Eles são atribuídos a contas não privilegiadas para proteger os arquivos e outros recursos uns dos outros e de usuários regulares no sistema. Os usuários não fazem login interativamente com uma conta de usuário do sistema.
- Usuario regular = uso para trabalho diario e possui acesso limitado ao sistema

Comando ID para mostrar as informações do usuario conectado:

```
[user01@host ~]$ id
uid=1000(user01) gid=1000(user01) groups=1000(user01) context
```

Comando ID + USER para ver informações de outros usuarios:

```
[user01@host ~]$ id umalguem
uid=1002(user02) gid=1001(user02) groups=1001(user02) context
```

Comando LS -L para ver o proprietario de um arquivo:

```
[user01@host ~]$ ls -l mytextfile.txt
-rw-rw-r--. 1 user01 user01 0 Feb  5 11:10 mytextfile.txt
```

Comando ls -ld para ver o proprietario de um diretorio:

```
[user01@host]$ ls -ld Documents
drwxrwxr-x. 2 user01 user01 6 Feb  5 11:10 Documents
```

Comando ps para ver informações de processos.

Comando ps -a para ver todos os processos em um terminal

Comando ps -au para ver o usuario associado a um processo

```
[user01@host ~]$ ps -au
USER     PID %CPU %MEM    VSZ   RSS TTY    STAT START  TIME C
root    1690  0.0  0.0 220984  1052 ttyS0  Ss+  22:43  0:00 /
user01  1769  0.0  0.1 377700  6844 tty2   Ssl+ 22:45  0:00 /
user01  1773  1.3  1.3 528948 78356 tty2   Sl+  22:45  0:03 /
user01  1800  0.0  0.3 521412 19824 tty2   Sl+  22:45  0:00 /
user01  3072  0.0  0.0 224152  5756 pts/1  Ss   22:48  0:00 -
user01  3122  0.0  0.0 225556  3652 pts/1  R+   22:49  0:00 p
```

O arquivo /etc/passwda é divido em 7 campos seperados por cores e armazena informações sobre os usuarios locais.

```
[user01@host ~]$ cat /etc/passwd
...output omitted...
user01:x:1000:1000:User One:/home/user01:/bin/bash
```

Cada parte do bloco é separada por dois pontos:

- user01 = nome do usuario
- x: = senha criptografada que foi armazenada e possui um espaço reservado
- 1000 = numero UID do usuario
- 1000 = numero GUID do grupo do usuario
- User One = descrição ou nome real do usuario
- /home/user01 = diretorio pessoal do usuario
- /bin/bash = o programa de shell padrão para esse usuário que é executado no login. Algumas contas usam o shell `/sbin/nologin` para proibir logins interativos com essa conta

O que é um grupo?

É uma coleção de usuario que precisam compartilhar o acesso a arquivos e outros recursos do sistema.

Podendo conceber acesso a um unico usuario ou um grupo de usuarios.

`Identificação` = GID e usam o arquivo /etc/group para armazenar informações dos grupos locais

`O arquivo` = é divido em 4 campos separados por cor:

```
[user01@host ~]$ cat /etc/group
...output omitted...
group01:x:10000:user01,user02,user03
```

Cada parte do bloco é separada por dois pontos:

- group01 = nome do grupo
- x = senha de grupo, espaço reservado
- 10000 = numero GUID desse grupo
- user01,user02 = lista de usuarios desse grupo suplementar

Grupo primario

Cada usuário tem exatamente um grupo principal.

Para usuários locais, esse é o grupo listado por número de GID no arquivo `/etc/passwd`.

O grupo primário possui arquivos que o usuário cria.

Quando cria um usuario comum, automaticamente se cria um grupo com o mesmo nome para ser o grupo primario desse usuario sendo unico nesse grupo.

É usado para simplifica o gerenciamento de permissões de arquivo, para que os grupos de usuários sejam separados por padrão.

Grupos sumplementares

A associação em grupos suplementares é armazenada no arquivo `/etc/group`.

Os usuários recebem acesso aos arquivos com base no acesso de qualquer um de seus grupos, independentemente de os grupos serem primários ou suplementares.

Por exemplo, se o usuário `user01` tiver um grupo primário `user01` e grupos suplementares `wheel` e `webadmin`, esse usuário poderá ler arquivos legíveis por qualquer um desses três grupos.

O comando `id` pode mostrar a associação ao grupo de um usuário. No exemplo anterior, o usuário `user01` tem o grupo `user01` como seu grupo primário (`gid`). O item `groups` lista todas as associações de grupo para esse usuário, e o usuário também tem os grupos `wheel` e `group01` como grupos suplementares.

O superusuario

Pode ser usado para gerenciar e administrar o sistema.

Para executar tarefas como a instalação ou a remoção de software e o gerenciamento de arquivos e diretórios de sistema, é necessário escalonar seus privilégios para o usuário `root`.

`Usuários normais` podem controlar dispositivos removíveis, como dispositivos USB. Sendo assim, usuários normais podem adicionar e remover arquivos e também para gerenciar um dispositivo removível

Porem o `root` consegue gerenciar discos rígidos por padrão.

Alternância de contas de usuário

`Comando su` = os usuários podem alternar para uma conta de usuário diferentes.

Este exemplo usa o comando `su` da conta `user01` para alternar para a conta `user02`:

```
[user01@host ~]$ su - user02
Password: user02_password
[user02@host ~]$
```

Se você omitir o nome de usuário, os comandos `su` ou `su -` tentarão alternar para `root` por padrão.

```
[user01@host ~]$ su -
Password: root_password
[root@host ~]#
```

`Comando su` = inicia uma shell que não é de login, define o ambiente do shell como se fosse um novo login com esse usuário,

`Comando su -` = inicia um shell de login, apenas inicia um shell com esse usuário, mas usa as configurações de ambiente do usuário original.

Execução de comandos com Sudo

Os administradores de sistema configuram o usuário `root` para não ter uma senha válida.

Assim, os usuários não podem fazer login no sistema como `root` diretamente com uma senha. Além disso, você não pode usar `su` para obter um shell interativo.

`Comando sudo` = para ter acesso root.
Os usuários que usam o comando `sudo` para executar comandos como `root` não precisam saber a senha de `root`. Em vez disso, eles usam suas próprias senhas para autenticar o acesso.

	su	su -	sudo
Torna-se um novo usuario	Sim	Sim	Por comando escalado

Ambiente	Do usuario atual	Do novo usuario	Do usuario atual
Senha obrigatoria	Do novo usuario	Do novo usuario	Do usuario atual
Privilegios	Os mesmos que o novo usuário	Os mesmos que o novo usuário	Definidos pela configuração
Atividade registrada	Somente comando `su`	Somente comando `su`	Por comando escalado

Permitir que usuario comuns executem comando com sudo = Permite que `user01` execute o comando `usermod` como root:

```
[user01@host ~]$ sudo usermod -L user02
[sudo] password for user01: user01_password
```

Se um usuário tentar executar um comando como outro usuário e a configuração `sudo` não permitir, o bash bloqueia o comando, registra a tentativa e, por padrão, envia um e-mail ao usuário `root`.

```
[user02@host ~]$ sudo tail /var/log/secure
[sudo] password for user02: user02_password
user02 is not in the sudoers file.  This incident will be rep
[user02@host ~]$
```

`Sudo` = pode registrar todos comando executador em /var/log/secure

```
[user01@host ~]$ sudo tail /var/log/secure
...output omitted...
Mar  9 20:45:46 host sudo[2577]:  user01 : TTY=pts/0 ; PWD=/h
...output omitted...
```

Obtenção de um root shell interativo com o Sudo

`Comando sudo -i` = acesso a conta root e executa o shell padrão desse usuário (normalmente `bash`) e os scripts de login interativos associados.

`Comando sudo -s` = executar o shell sem os scripts interativos

Configuração do sudo

`Arquivo /etc/sudoers` = arquivo de configuração principal para o comando `sudo`.

Também inclui o conteúdo de qualquer arquivo no diretório `/etc/sudoers.d` como parte do arquivo de configuração.

Para evitar problemas se vários administradores tentarem editar o arquivo ao mesmo tempo, você só poderá editá-lo com o comando especial `visudo`.

O editor `visudo` também valida o arquivo para garantir que não haja erros de sintaxe.

Por exemplo, a linha a seguir do arquivo `/etc/sudoers` permite acesso `sudo` aos membros do grupo `wheel`.

```
%wheel        ALL=(ALL:ALL)       ALL
```

- O símbolo `%` antes da palavra `wheel` especifica um grupo. A string `%wheel` é o usuário ou grupo ao qual a regra se aplica
- O comando `ALL=(ALL:ALL)` especifica que em qualquer host com esse arquivo (o primeiro `ALL`), os usuários no grupo `wheel` podem executar comandos como qualquer outro usuário (o segundo `ALL`) e como qualquer outro grupo (o terceiro `ALL`) no sistema.
- O comando final `ALL` especifica que os usuários no grupo `wheel` podem executar qualquer comando.

Para permitir acesso `sudo` total para o usuário `user01`, você pode criar o arquivo `/etc/sudoers.d/user01` com o seguinte conteúdo:

```
user01        ALL=(ALL)       ALL
```

Para permitir acesso `sudo` total para o grupo `group01`, você pode criar o arquivo `/etc/sudoers.d/group01` com o seguinte conteúdo:

```
%group01        ALL=(ALL)       ALL
```

Para permitir que os usuários no grupo `games` executem o comando `id` como o usuário `operator`, você pode criar o arquivo `/etc/sudoers.d/games` com o seguinte conteúdo:

```
%games ALL=(operator) /bin/id
```

`Comando NOPASSWD: ALL` = configurar `sudo` para permitir que um usuário execute comandos como outro usuário sem digitar sua senha:

```
ansible         ALL=(ALL)       NOPASSWD: ALL
```

Gerenciamento de contas de usuários locais

Criação de usuários a partir da linha de comando

`Comando useradd username` = cria um usuario chamado username, tambem configura um diretorio pessoal e um grupo privado.

O arquivo `/etc/login.defs` define algumas opções padrão para contas de usuário, como o intervalo de números de UID válidos e as regras padrão para duração de senhas.

Modificação de usuários existentes a partir da linha de comando

Opções do `usermod`	Uso
-a, —apend	Use-o com a opção `-G` para adicionar os grupos suplementares ao conjunto atual de associações de grupos, em vez de substituir o conjunto de grupos suplementares por um novo conjunto.
-c, --comment COMMENT	Adiciona o texto `COMMENT` ao campo de comentários.
-d, --home HOME_DIR	Especifica um diretório pessoal para a conta de usuário.
-g, --gid GROUP	Especifica o grupo primário para a conta de usuário.
-L, --lock	Bloqueia a conta do usuário.
-m, --move-home	Move um diretório pessoal do usuário para um novo local. Você deve usá-lo com a opção `-d`.
-s, --shell SHELL	Especifica um shell de login específico para a conta de usuário.
-U, --unlock	Desbloqueia a conta do usuário.

Exclusão de usuario a partir da linha de comando

`Comando userdel username` = remove o usuário `username` de `/etc/passwd`, mas deixa o diretório pessoal do usuário intacto.

`Comando userdel -r username` = remove o usuario `username` de `/ect/passwd` e exclui o diretorio pessoal do usuario.

Importante

Ao excluir um usuário sem usar a opção "userdel -r", os arquivos do usuário excluído são atribuídos a uma UID não associada. Se um novo usuário é criado e acidentalmente recebe essa mesma UID, ele terá acesso aos arquivos do usuário anterior, representando um risco de segurança. Por isso, muitas organizações preferem bloquear contas em vez de excluí-las, para evitar tal situação.

Exemplo

```
[root@host ~]# useradd user01
[root@host ~]# ls -l /home
drwx------. 3 user01  user01    74 Mar  4 15:22 user01
```

```
[root@host ~]# userdel user01
[root@host ~]# ls -l /home
drwx------. 3    1000    1000   74 Mar  4 15:22 user01
[root@host ~]# useradd -u 1000 user02
[root@host ~]# ls -l /home
drwx------. 3 user02     user02       74 Mar  4 15:23 user02
drwx------. 3 user02     user02       74 Mar  4 15:22 user01
```

Observe que o `user02` agora é proprietário de todos os arquivos dos quais o `user01` era proprietário anteriormente. O usuário `root` pode usar o comando `find / -nouser -o -nogroup` para localizar todos os arquivos e diretórios sem proprietários.

Configuração de senhas a partir da linha de comando

`Comando passwd username` = define a senha inicial ou altera a senha existente para o usuário `username`.

Intervalos de UID

- `UID 0` = UID da conta de superusuário (`root`).
- `UID 1-200` = UIDs da conta do sistema que são atribuídas estaticamente aos processos do sistema.
- `UID 201-999` = UIDs que são atribuídas a processos do sistema que não possuem arquivos neste sistema. O software que exige uma UID sem privilégios é atribuído dinamicamente a uma UID a partir desse pool disponível.
- `UID 1000+` = o intervalo de UIDs a ser atribuído a usuários regulares e sem privilégios.

Exercicio orientado: Gerenciamento de contas de usuários locais

```
Exercício orientado: Gerenciamento de contas de usuários loca
Neste exercício, você cria vários usuários em seu sistema e d
```

```
Resultados

Configurar uma sistema Linux com contas de usuário adicionais

Com o usuário student na máquina workstation, use o comando l

Esse comando prepara seu ambiente e garante que todos os recu

[student@workstation ~]$ lab start users-user
Instruções

Na workstation, abra uma sessão de SSH em servera com o usuár

[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$ sudo -i
[sudo] password for student: student
[root@servera ~]#
Crie o usuário operator1 e confirme se ele existe no sistema.

[root@servera ~]# useradd operator1
[root@servera ~]# tail /etc/passwd
...output omitted...
operator1:x:1002:1002::/home/operator1:/bin/bash
Defina a senha para operator1 como redhat.

[root@servera ~]# passwd operator1
Changing password for user operator1.
New password: redhat
BAD PASSWORD: The password is shorter than 8 characters
Retype new password: redhat
passwd: all authentication tokens updated successfully.
Crie os usuários adicionais operator2 e operator3. Defina a s

Adicione o usuário operator2. Defina a senha para operator2 c

[root@servera ~]# useradd operator2
[root@servera ~]# passwd operator2
```

```
Changing password for user operator2.
New password: redhat
BAD PASSWORD: The password is shorter than 8 characters
Retype new password: redhat
passwd: all authentication tokens updated successfully.
Adicione o usuário operator3. Defina a senha para operator3 c

[root@servera ~]# useradd operator3
[root@servera ~]# passwd operator3
Changing password for user operator3.
New password: redhat
BAD PASSWORD: The password is shorter than 8 characters
Retype new password: redhat
passwd: all authentication tokens updated successfully.

================================================================

Atualize as contas de usuário operator1 e operator2 para incl
Operator One e Operator Two, respectivamente. Verifique se os
para as contas de usuário.

Execute o comando usermod -c para atualizar os comentários da

[root@servera ~]# usermod -c "Operator One" operator1

Execute o comando usermod -c para atualizar os comentários da

[root@servera ~]# usermod -c "Operator Two" operator2
Exiba o arquivo /etc/passwd para confirmar se existem comentá

[root@servera ~]# tail /etc/passwd
...output omitted...
operator1:x:1002:1002:Operator One:/home/operator1:/bin/bash
operator2:x:1003:1003:Operator Two:/home/operator2:/bin/bash
operator3:x:1004:1004::/home/operator3:/bin/bash
Excluir o usuário operator3 junto com quaisquer dados pessoai

Remova o usuário operator3 do sistema.
```

```
[root@servera ~]# userdel -r operator3
Confirme que o usuário operator3 não existe.

[root@servera ~]# tail /etc/passwd
...output omitted...
operator1:x:1002:1002:Operator One:/home/operator1:/bin/bash
operator2:x:1003:1003:Operator Two:/home/operator2:/bin/bash
Observe que a saída anterior não exibe as informações da cont

Confirme que o diretório pessoal do usuário operator3 não exi

[root@servera ~]# ls -l /home
total 0
drwx------. 4 devops    devops    90 Mar  3 09:59 devops
drwx------. 2 operator1 operator1 62 Mar  9 10:19 operator1
drwx------. 2 operator2 operator2 62 Mar  9 10:19 operator2
drwx------. 3 student   student   95 Mar  3 09:49 student
Saia do shell do usuário root para voltar ao shell do usuário

[root@servera ~]# exit
logout
[student@servera ~]$
Faça logoff da máquina servera.

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

Na máquina workstation, altere para o diretório pessoal do us

[student@workstation ~]$ lab finish users-user
Isso conclui a seção.
```

Gerenciamento de contas de grupos locais

Criação de grupos a partir da linha de comando

O comando `groupadd` cria grupos.

Sem opções, o comando `groupadd` usa a próxima GID disponível no intervalo que as variáveis `GID_MIN` e `GID_MAX` especificam no arquivo `/etc/login.defs`.

Por padrão, o comando atribui um valor de GID que é maior do que qualquer outra GID existente, mesmo se um valor inferior ficar disponível.

`Comando groupadd -g` = especifica uma GID para o grupo usar.

```
[root@host ~]# groupadd -g 10000 group01
[root@host ~]# tail /etc/group
...output omitted...
group01:x:10000:
```

`Comando groupadd -r` = cria grupos, como grupos normais, grupos de sistema usam uma GID do intervalo de GIDs de sistema válidos listados no arquivo `/etc/login.defs`.

Os itens de configuração `SYS_GID_MIN` e `SYS_GID_MAX` no arquivo `/etc/login.defs` definem o intervalo de GIDs do sistema.

```
[root@host ~]# groupadd -r group02
[root@host ~]# tail /etc/group
...output omitted...
group01:x:10000:
group02:x:988:
```

Modificação de grupos existentes a partir da linha de comando

`Comando groupmod` = altera as propriedades de um grupo existente

`Comando groupmod -n` = especifica um novo nome para o grupo.

```
[root@host ~]# groupmod -n group0022 group02
[root@host ~]# tail /etc/group
...output omitted...
group0022:x:988:
```

Observe que o nome do grupo é atualizado para `group0022` de `group02`

`Comando groupmod -g` = especifica uma nova GID

```
[root@host ~]# groupmod -g 20000 group0022
[root@host ~]# tail /etc/group
...output omitted...
group0022:x:20000:
```

Exclusão de grupos a partir da linha de comando

`Comando groupdel` = remove grupos

```
[root@host ~]# groupdel group0022
```

Alteração da associação do grupo na linha de comando

A associação de um grupo é controlada com o gerenciamento de usuários.

`Comando usermod -g` = para alterar o grupo principal de um usuário

```
[root@host ~]# id user02
uid=1006(user02) gid=1008(user02) groups=1008(user02)
[root@host ~]# usermod -g group01 user02
[root@host ~]# id user02
uid=1006(user02) gid=10000(group01) groups=10000(group01)
```

`Comando usermod -aG` = adicionar um usuário a um grupo suplementar.

```
[root@host ~]# id user03
uid=1007(user03) gid=1009(user03) groups=1009(user03)
[root@host ~]# usermod -aG group01 user03
[root@host ~]# id user03
uid=1007(user03) gid=1009(user03) groups=1009(user03),10000(g
```

Comparação da associação a grupos primários e suplementares

O grupo primário de um usuário é o grupo exibido na conta do usuário no arquivo `/etc/passwd`. Um usuário pode pertencer a apenas um grupo primário por vez.

Os grupos suplementares de um usuário são os grupos adicionais configurados para o usuário e exibidos na entrada do usuário no arquivo `/etc/group`. Um usuário pode pertencer a quantos grupos suplementares forem necessários para implementar o acesso e as permissões de arquivos com eficiência.

Para a configuração de permissões de arquivo baseadas em grupo, não há diferença entre os grupos primários e suplementares de um usuário. Se o usuário pertencer a qualquer grupo ao qual seja atribuído acesso a arquivos específicos, esse usuário terá acesso a esses arquivos.

UNICA DISTINÇÃO DO GRUPO PRIMARIO E SUPLEMENTAR

A única distinção entre as associações primárias e suplementares de um usuário é quando um usuário cria um arquivo. Novos arquivos devem ter um proprietário de usuário e um proprietário de grupo, que é atribuído quando o arquivo é criado. O grupo primário do usuário é usado para a propriedade do grupo do novo arquivo, a menos que seja substituído por opções de comando.

Alteração temporária do seu grupo primário

Somente o grupo primário de um usuário é usado para novos atributos de criação de arquivo.

No entanto, você pode alternar temporariamente seu grupo primário para um grupo suplementar ao qual já pertence.

Você poderá alternar se estiver prestes a criar arquivos, manualmente ou em script, e quiser atribuir um grupo diferente como proprietário quando estiverem sendo criados.

`Comando newgrp` = para alterar seu grupo primario, nesta sessão de shell, pode alternar entre qualquer grupo primário ou suplementar ao qual pertença, mas somente um grupo por vez pode ser primário. Seu grupo primário retornará ao padrão se você fizer o logout e login novamente.

Neste exemplo, o grupo `group01` torna-se temporariamente o grupo primário desse usuário.

```
[user03@host ~]# id
uid=1007(user03) gid=1009(user03) groups=1009(user03),10000(g
[user03@host ~]$ newgrp group01
[user03@host ~]# id
uid=1007(user03) gid=10000(group01) groups=1009(user03),10000
```

Exercício orientado: Gerenciamento de contas de grupos locais

```
Instruções

Na workstation, abra uma sessão de SSH em servera com o usuár
student e alterne para o usuário root.

[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$ sudo -i
[sudo] password for student: student
[root@servera ~]#
Crie o grupo suplementar operators com uma GID de 30000.
```

```
[root@servera ~]# groupadd -g 30000 operators
Crie o grupo suplementar admin sem especificar uma GID.

[root@servera ~]# groupadd admin
Verifique se os grupos suplementares operators e admin existe

[root@servera ~]# tail /etc/group
...output omitted...
operators:x:30000:
admin:x:30001:
Certifique-se de que os usuários operator1, operator2 e opera
pertencem ao grupo operators.

Adicione os usuários operator1, operator2 e operator3 ao grup
operators.

[root@servera ~]# usermod -aG operators operator1
[root@servera ~]# usermod -aG operators operator2
[root@servera ~]# usermod -aG operators operator3
Confirme se os usuários estão no grupo.

[root@servera ~]# id operator1
uid=1002(operator1) gid=1002(operator1) groups=1002(operator1
30000(operators)
[root@servera ~]# id operator2
uid=1003(operator2) gid=1003(operator2) groups=1003(operator2
30000(operators)
[root@servera ~]# id operator3
uid=1004(operator3) gid=1004(operator3) groups=1004(operator3
30000(operators)
Certifique-se de que os usuários sysadmin1, sysadmin2 e sysad
pertencem ao grupo admin. Habilite direitos administrativos p

Adicione os usuários sysadmin1, sysadmin2 e sysadmin3 ao grup

[root@servera ~]# usermod -aG admin sysadmin1
[root@servera ~]# usermod -aG admin sysadmin2
[root@servera ~]# usermod -aG admin sysadmin3
```

```
Confirme se os usuários estão no grupo.

[root@servera ~]# id sysadmin1
uid=1005(sysadmin1) gid=1005(sysadmin1) groups=1005(sysadmin1
30001(admin)
[root@servera ~]# id sysadmin2
uid=1006(sysadmin2) gid=1006(sysadmin2) groups=1006(sysadmin2
30001(admin)
[root@servera ~]# id sysadmin3
uid=1007(sysadmin3) gid=1007(sysadmin3) groups=1007(sysadmin3
30001(admin)
Examine o arquivo /etc/group para verificar as associações ao
suplementar.

[root@servera ~]# tail /etc/group
...output omitted...
operators:x:30000:operator1,operator2,operator3
admin:x:30001:sysadmin1,sysadmin2,sysadmin3
Crie o arquivo /etc/sudoers.d/admin de forma que os membros d
admin tenham privilégios administrativos totais.

[root@servera ~]# echo "%admin ALL=(ALL) ALL" >> /etc/sudoers
Alterne para o usuário sysadmin1 (um membro do grupo admin) e
verifique se você pode executar um comando sudo.

[root@servera ~]# su - sysadmin1
[sysadmin1@servera ~]$ sudo cat /etc/sudoers.d/admin
[sudo] password for sysadmin1: redhat
%admin ALL=(ALL) ALL
Retorne à máquina workstation como o usuário student.

[sysadmin1@servera ~]$ exit
logout
[root@servera ~]# exit
logout
[student@servera ~]$ exit
logout
Connection to servera closed.
```

```
[student@workstation ~]$
Encerramento

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

[student@workstation ~]$ lab finish users-group
Isso conclui a seção.
```

Gerenciamento de senhas de usuários

Senhas shadow e política de senha

No passado, as senhas criptografadas eram armazenadas no arquivo `/etc/passwd`, que podia ser lido por todos.

As senhas criptografadas com hash foram movidas para o arquivo `/etc/shadow`, que somente o usuário `root` pode ler.

```
[root@host ~]# cat /etc/shadow
...output omitted...
user03:$6$CSsXsd3rwghsdfarf:17933:0:99999:7:2:18113:
```

Cada campo deste bloco de código é separado por dois pontos:

- `user03` : nome da conta de usuário.
- `$6$CSsXsd3rwghsdfarf` : a senha criptografada com hash do usuário.
- `17933` : os dias a partir da época em que a senha foi alterada pela última vez, em que a epoch é `1970-01-01` no fuso horário UTC.
- `0` : o número mínimo de dias desde a última alteração de senha antes que o usuário possa alterá-la novamente.

- `99999` : o número máximo de dias sem uma alteração de senha antes que a senha expire. Um campo vazio significa que a senha nunca expira.
- `7` : o número de dias até o usuário ser avisado de que sua senha expirará.
- `2` : o número de dias sem atividade, começando com o dia em que a senha expirou, antes de a conta ser automaticamente bloqueada.
- `18113` : o dia em que a conta expira em dias desde a epoch. Um campo vazio significa que a conta nunca expira.
- O último campo geralmente está vazio e é reservado para uso futuro.

Formato de uma senha criptografada com hash

O campo de senha criptografada com hash armazena três informações:

- o algoritmo de hashing em uso
- o sal = adiciona dados aleatorios ao hash criptográfico para criar um hash exclusivo e fortalecer a senha com hash criptográfico
- hash criptografico

Cada informação é delimitada pelo caractere de dólar `$:`

```
$6$CSsXcYG1L/4ZfHr/$2W6evvJahUfzfHpc9X.45Jc6H30E
```

- **6** = é o algoritmo de hash usado para a senha. Um 6 indica um hash SHA-512. 1 indica MD5 e um 5 indica SHA-256
- **CSsXcYG1L/4ZfHr/** = o sal em uso para criptografar com hash a senha
- **2W6evvJahUfzfHpc9X.45Jc6H30E** = o hash criptográfico da senha do usuário, combinando o sal e a senha de texto simples e, depois, criptografando com hash para gerar o hash da senha.

Verificação de senha

Quando um usuário tenta fazer login, o sistema procura a entrada do usuário no arquivo `/etc/shadow` e combina o sal para o usuário com a senha de texto simples digitada.

Em seguida, o sistema executa um hash criptográfico da combinação do sal e da senha de texto simples com o algoritmo de hash especificado.

Se:

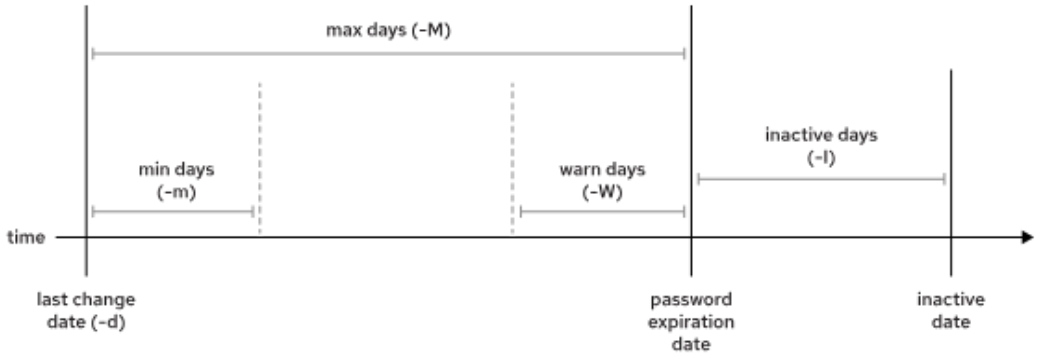
`o resultado corresponder ao hash criptográfico, o usuário digitou a senha correta.`

Senão:

`o resultado não corresponder ao hash criptográfico, o usuário digitou a senha incorreta, e a tentativa de login falhará`

Configuração do vencimento de senha

O diagrama a seguir mostra os parâmetros de vencimento de senha relevantes que podem ser ajustados usando o comando `chage` para implementar uma política de vencimento de senha. Observe que o nome do comando é `chage`, o que significa "alterar idade". Não confunda o comando com a palavra "alterar".



O exemplo a seguir demonstra o comando `chage` para alterar a política de senha do usuário `sysadmin05`.

O comando define uma idade mínima (`-m`) de zero dias, uma idade máxima (`-M`) de 90 dias, um período de aviso (`-W`) de 7 dias e um período de inatividade (`-I`) de 14 dias.

```
[root@host ~]# chage -m 0 -M 90 -W 7 -I 14 sysadmin05
```

Digamos que você gerencie as políticas de senha do usuário em um servidor Red Hat. O usuário `cloudadmin10` é novo no sistema, e você deseja definir uma política de vencimento de senha personalizada. Você deseja definir a expiração da conta para 30 dias a partir de hoje, então use os seguintes comandos:

1. Use o comando `date +%F` para obter a data atual

2. Use o comando `date -d “+30 days” +%F` para obter a data 30 dias a partir de agora
3. Use a opção `-E` do comando `chage` para alterar a data de expiração do usuário `cloudadmin10`. `chage -E $(date -d "+30 days" +%F) cloudadmin10`
4. Use a opção `chage` do comando `-l` para exibir a política de vencimento da senha para o usuário `cloudadmin10`.
   `chage -l cloudadmin10 | grep "Account expires"`

Você deseja que o usuário `cloudadmin10` altere a senha no próximo login, então use o seguinte comando:

```
[root@host ~]# chage -d 0 cloudadmin10
```

Na próxima vez que o usuário `cloudadmin10` fizer login, ele será solicitado a alterar a senha.

O comando `date` pode calcular uma data no futuro. A opção `-u` informa a hora em UTC.

```
[user01@host ~]$date -d "+45 days" -u
Thu May 23 17:01:20 UTC 2019
```

Você pode alterar a configuração de expiração de senha padrão no arquivo `/etc/login.defs`. As opções `PASS_MAX_DAYS` e `PASS_MIN_DAYS` definem a idade padrão máxima e mínima da senha, respectivamente. O `PASS_WARN_AGE` define o período de aviso padrão da senha. Qualquer alteração nas políticas padrão de vencimento de senha afeta os usuários que são criados após a alteração. Os usuários existentes continuam usando as configurações anteriores de vencimento de senha, em vez de usar as configurações mais recentes. Para obter mais informações sobre o arquivo `/etc/login.defs`, consulte o curso *Red Hat Security: Linux in Physical, Virtual, and Cloud* (RH415) e a página do man `login.defs(5)`.

Restrição de acesso

Você pode usar o comando `usermod` para modificar o vencimento da conta para um usuário. Por exemplo, a opção `-L` do comando `usermod` bloqueia uma conta de usuário, e o usuário não pode fazer login no sistema

```
[root@host ~]#usermod -L sysadmin03
[user01@host ~]$su - sysadmin03
Password:redhat
su: Authentication failure
```

Se um usuário deixar a empresa em uma determinada data, você poderá bloquear e expirar a conta com um único comando `usermod`. A data deve ser o número de dias desde `1970-01-01` ou usar o formato *AAAA-MM-DD*. No exemplo a seguir, o comando `usermod` bloqueia e expira o usuário `cloudadmin10` em `2022-08-14`

```
[root@host ~]# usermod -L -e 2022-08-14 cloudadmin10
```

Quando você bloqueia uma conta, impede que o usuário faça a autenticação com senha no sistema. Esse método é recomendado para impedir o acesso a uma conta por um ex-funcionário da empresa. Use a opção `-U` do comando `usermod` para habilitar o acesso à conta novamente.

Shell sem login

O shell `nologin` funciona como um shell de substituição para as contas de usuário que não pretendem fazer login interativamente no sistema. É uma boa prática de segurança liberar uma conta de fazer login no sistema quando a conta não exige login. Por exemplo, um servidor de e-mail pode exigir uma conta para armazenar mensagens de e-mail e uma senha para o usuário autenticar com um cliente de e-mail para obter mensagens. Esse usuário não precisa fazer login diretamente no sistema.

Uma solução comum para essa situação é definir o shell de login do usuário como `/sbin/nologin`. Se o usuário tentar fazer login no sistema diretamente, o shell `nologin` fecha a conexão.

```
[root@host ~]#usermod -s /sbin/nologin newapp
[root@host ~]#su - newapp
Last login: Wed Feb  6 17:03:06 IST 2019 on pts/0
This account is currently not available.
```

Exercício orientado: Gerenciamento de senhas de usuários

Neste exercício, você define políticas de senha para diversos usuários.

Resultados

- Forçar uma alteração de senha quando o usuário fizer login no sistema pela primeira vez.
- Forçar uma alteração de senha a cada 90 dias.
- Configurar a conta para que expire 180 dias a partir do dia atual.

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 users-password
```

Instruções

1. Na `workstation`, abra uma sessão de SSH com o usuário `student` para a máquina `servera`.

   ```
   [student@workstation ~]$ssh student@servera
   [student@servera ~]$
   ```

2. Em `servera`, use o comando `usermod` para bloquear e desbloquear o usuário `operator1`.

   a. Como o usuário `student`, use direitos administrativos para bloquear a conta `operator1`.

```
[student@servera ~]$sudo usermod -L operator1
[sudo] password for student:student
```

b. Tente fazer login como `operator1`. Esse comando deve falhar.

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

c. Desbloqueie a conta `operator1`.

```
[student@servera ~]$sudo usermod -U operator1
```

d. Tente fazer login como `operator1` novamente. Dessa vez, o comando deve funcionar.

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

e. Saia do shell do usuário `operator1` para voltar ao shell do usuário `student`.

```
[operator1@servera ~]$exit
logout
```

3. Altere a diretiva de senha da conta do usuário `operator1` para que ela solicite uma nova senha a cada 90 dias. Confirme se a idade da senha foi configurada com êxito.

a. Mude para o usuário `root`.

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

b. Defina a idade máxima da senha do usuário `operator1` para 90 dias.

```
[root@servera ~]#chage -M 90 operator1
```

c. Verifique se a senha do usuário `operator1` expira 90 dias após sua alteração.

```
[root@servera ~]#chage -l operator1
Last password change      : Mar 10, 2022
Password expires          : Jun 10, 2022
Password inactive         : never
Account expires           : never
Minimum number of days between password change    : 0
Maximum number of days between password change    : 9
0
Number of days of warning before password expires : 7
```

4. Force uma alteração de senha no primeiro login da conta `operator1`.

```
[root@servera ~]#chage -d 0 operator1
```

5. Saia como o usuário `root` na máquina `servera`.

```
[root@servera ~]#exit
logout
[student@servera ~]$
```

6. Faça login como `operator1` e altere a senha para `forsooth123`. Depois de definir a senha, retorne ao shell do usuário `student`.

a. Faça login como `operator1` e altere a senha para `forsooth123` quando solicitado.

```
[student@servera ~]$su - operator1
Password:redhat
You are required to change your password immediately
(administrator enforced)
Current password:redhat
New password:forsooth123
```

```
Retype new password:forsooth123...output omitted...
[operator1@servera ~]$
```

b. Saia do shell do usuário `operator1` para retornar ao usuário `student` e, em seguida, alterne para o usuário `root`.

```
[operator1@servera ~]$exit
logout
[student@servera ~]$sudo -i
[sudo] password for student:student
[root@servera ~]#
```

7. Configure a conta `operator1` para que expire 180 dias a partir do dia atual.

a. Determine uma data futura de 180 dias. Use o formato `%F` com o comando `date` para obter o valor exato. Essa data retornada é um exemplo, use o valor em seu sistema para as etapas posteriores.

```
[root@servera ~]#date -d "+180 days" +%F
2022-09-06
```

b. Defina a conta para que expire na data exibida na etapa anterior. Por exemplo:

```
[root@servera ~]#chage -E 2022-09-06 operator1
```

c. Verifique se a data de vencimento da conta foi definida com êxito.

```
[root@servera ~]#chage -l operator1
Last password change      : Mar 10, 2022
Password expires          : Jun 10, 2022
Password inactive         : never
Account expires           : Sep 06, 2022
Minimum number of days between password change    : 0
Maximum number of days between password change    : 9
0
Number of days of warning before password expires : 7
```

8. Defina as senhas para que expirem 180 dias a partir da data atual para todos os usuários. Use os direitos administrativos para editar o arquivo de configuração.

   a. Defina `PASS_MAX_DAYS` como `180` em `/etc/login.defs`. Use direitos administrativos ao abrir o arquivo com o editor de texto. Você pode usar o comando `vim /etc/login.defs` para executar essa etapa.

```
...output omitted...
# Password aging controls:
#
#       PASS_MAX_DAYS   Maximum number of days a pass
word may be
#       used.
#       PASS_MIN_DAYS   Minimum number of days allowe
d between
#       password changes.
#       PASS_MIN_LEN    Minimum acceptable password l
ength.
#       PASS_WARN_AGE   Number of days warning given
before a
#       password expires.
#
PASS_MAX_DAYS 180
PASS_MIN_DAYS   0
PASS_WARN_AGE   7
...output omitted...
```

   ### Importante

   A senha padrão e as configurações de vencimento da conta se aplicam a novos usuários, mas não para usuários existentes.

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

```
[root@servera ~]#exit
logout
[student@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 users-password
```

Isso conclui a seção

Laboratório Aberto: Gerenciar usuários e grupos locais

```
Defina PASS_MAX_DAYS como 30 no arquivo /etc/login.defs. Use
ao abrir o arquivo com o editor de texto. Você pode usar o co
nessa etapa.
1. No arquivo `/etc/login.defs` definir `PASS_MAX_DAYS` como

Crie o grupo consultants com uma GID de 35000.
2. groupadd -g 35000 consultants

Crie o arquivo /etc/sudoers.d/consultants e adicione o conteú
pode usar o comando vim /etc/sudoers.d/consultants nessa etap
3.  sudo vim /etc/sudoers.d/consultants
%consultants ALL=(ALL) ALL

Crie os usuários consultant1, consultant2 e consultant3 com o
grupo suplementar deles.
4. useradd -G consultants consultant1
useradd -G consultants consultant2
useradd -G consultants consultant3

Defina as senhas consultant1, consultant2 e consultant3 como
```

```
5. consultant1 passwd
consultant2 passwd
consultant3 passwd

Determine a data futura de 90 dias.Defina a data de vencimento
consultant2 e consultant3 com o mesmo valor determinado na et
6. `**date -d “+90 days” +%F`**
chage -E 2024-09-09 consultant1
chage -E 2024-09-09 consultant2
chage -E 2024-09-09 consultant3

Altere a diretiva de senha da conta consultant2 para que soli
cada 15 dias.
7. chage -M 15 consultant2

Além disso, force os usuários consultant1, consultant2 e cons
senhas no primeiro login.Defina o último dia da alteração da
usuários precisem alterar a senha quando fizerem login no sis
8. chage -d 0 consultant1
chage -d 0 consultant2
chage -d 0 consultant3
```