Sessão 3: Usuários e grupos

As atividades desta sessão serão realizadas na máquina virtual *Client_Linux*.

Em algumas atividades, você trabalhará com a conta `root`, o que lhe dará todos os direitos sobre os recursos do sistema. Seja cauteloso antes de executar qualquer comando.

1) Criando contas de usuários

Uma das atividades que fazem parte da rotina diária de um administrador de sistemas é o gerenciamento de contas de usuários. Frequentemente, usuários são criados, modificados, desabilitados ou excluídos do sistema.

1. Descubra se o sistema faz uso de *shadow passwords* ou se ainda utiliza o esquema tradicional.

   ```
   $ ls -ld /etc/gshadow /etc/shadow
   -rw-r----- 1 root shadow  666 Ago  5 16:52 /etc/gshadow
   -rw-r----- 1 root shadow 1125 Ago  5 16:51 /etc/shadow
   ```

   O aluno deve verificar se os arquivos `/etc/shadow` e `/etc/gshadow` existem.

2. Crie uma conta para você no sistema, seguindo os passos descritos na aula teórica e no material didático.
   - Editar o arquivo `/etc/group` e inserir uma nova linha com os parâmetros relativos ao grupo do novo usuário:
     - Nome do grupo;
     - Senha ("x");
     - GID;
     - Membros do grupo.

       ```
       marcelo:x:1001:
       ```

   - Editar o arquivo `/etc/gshadow` e inserir uma nova linha com os parâmetros relativos ao grupo do novo usuário:
     - Nome do grupo;
     - Senha criptografada do grupo ("!");
     - Administradores do grupo;
     - Membros do grupo.

```
marcelo:!::
```

- Editar o arquivo `/etc/passwd` e inserir uma nova linha com os parâmetros relativos à conta do novo usuário:
  - Nome do usuário;
  - Senha ("x");
  - UID;
  - GID;
  - GECOS: campo com comentários informativos do usuário;
  - Diretório *home*;
  - Shell de login.

```
marcelo:x:1001:1001:,,,:/home/marcelo:/bin/bash
```

- Editar o arquivo `/etc/shadow` e inserir uma nova linha os parâmetros relativos à conta do novo usuário:
  - Nome do usuário;
  - Senha criptografada: inserir valor "*", que será alterado a seguir;
  - *last_change*: número de dias desde a última alteração de senha;
  - *minimum*: número mínimo de dias até que senha possa ser alterada novamente;
  - *maximum*: número máximo de dias até que a senha deva ser alterada;
  - *warning*: número de dias para aviso de expiração de senha;
  - *inactive*: número de dias após expiração em que a senha será aceita;
  - *expire*: data para expiração da senha.

```
marcelo:*:16846:0:99999:7:::
```

- Definir uma senha para a nova conta, utilizando o comando `passwd`:

```
# passwd marcelo
```

- Copiar os arquivos de inicialização contidos no diretório `/etc/skel` para o diretório *home* do usuário.

```
# cp -r /etc/skel /home/marcelo
```

- Alterar o usuário e grupo donos dos arquivos na pasta *home* do novo usuário:

```
# chown -R marcelo.marcelo /home/marcelo
```

- Configurar a *quota* de disco para o usuário, se o sistema utilizar *quotas*.
- Testar se a conta foi criada corretamente, fazendo login no sistema e verificando se o diretório corrente é o diretório *home* do usuário, definido no arquivo `/etc/passwd`.
- O *script shell* abaixo mostra uma maneira como os comandos executados manualmente nesta atividade poderiam ser automatizados por um administrador de sistemas:

```
#!/bin/bash

usage() {
  echo "  Usage: $0 -u USER -p PASSWORD"
  exit 1
}


if [[ $EUID -ne 0 ]]; then
  echo "  [*] Not root!" 1>&2
  exit 1
fi

while getopts ":u:p:" opt; do
  case "$opt" in
    u)
      user=${OPTARG}
      ;;
    p)
      pass=${OPTARG}
      ;;
    *)
      usage
      ;;
  esac
done

[ -z $user ] && { echo "  [*] No user?"; usage; }
[ -z $pass ] && { echo "  [*] No password?"; usage; }

if egrep "^${user}:" /etc/passwd &> /dev/null; then
  echo "  [*] User exists!"
  exit 1
fi

lastgid=$( getent group | grep -v 'nogroup' | cut -d':' -f3 | sort -n | tail -n1
)
((lastgid++))

echo "$user:x:$lastgid:" >> /etc/group
```

```
echo "$user:!::" >> /etc/gshadow

lastuid=$( getent passwd | grep -v 'nobody' | cut -d':' -f3 | sort -n | tail -n1
)
((lastuid++))

echo "$user:x:$lastuid:$lastgid:,,,:/home/$user:/bin/bash" >> /etc/passwd

salt="$( cat /dev/urandom | tr -dc 'a-zA-Z0-9' | fold -w 8 | head -n 1 )"
hpass="$( mkpasswd -m sha-512 -S $salt -s <<< $pass )"
echo "$user:$hpass:16842:0:99999:7:::" >> /etc/shadow

cp -r /etc/skel /home/$user
chown -R ${user}.${user} /home/$user
```

3. Agora, crie uma conta para o instrutor, utilizando, desta vez, o comando `useradd`. Faça com que a conta criada tenha sete dias de duração e com que o seu diretório de trabalho seja `/NOME`, onde `NOME` é o nome de usuário para o qual a conta deve ser aberta.

Consulte a página de manual do comando `useradd` e procure as informações necessárias para incluir a data de expiração (*expire date*) e criar o diretório de trabalho (*homedir*) em um local diferente do padrão, que é `/home/NOME`. Ainda, não se deve esquecer de escolher e atribuir uma senha para as contas que obedeça aos padrões de segurança apresentados no texto. Observe, ainda, que o diretório *home* não é criado automaticamente pelo comando `useradd`.

```
# useradd instrutor -d /instrutor -m -e 2018-08-07
# passwd instrutor
Digite a nova senha UNIX:
Redigite a nova senha UNIX:
passwd: senha atualizada com sucesso
```

Ao usar o comando `useradd`, o shell escolhido pelo sistema é o `/bin/sh`, por padrão. Para alterar o shell do usuário, pode-se editar o arquivo `/etc/passwd` diretamente, ou executar o comando `chsh`, mostrado abaixo:

```
# getent passwd | grep '^instrutor:'
instrutor:x:1002:1002::/home/instrutor:/bin/sh

# chsh instrutor
Mudando o shell de login para instrutor
Informe o novo valor ou pressione ENTER para aceitar o padrão
        Shell de Login [/bin/sh]: /bin/bash

# getent passwd | grep '^instrutor:'
instrutor:x:1002:1002::/home/instrutor:/bin/bash
```

4. O comando `useradd` não é uma boa opção para informar a senha do usuário. Por quê?

   Porque a senha criptografada deve ser digitada diretamente na linha de comando, podendo ser lida posteriormente via logs ou histórico do shell.

5. Faça um *script* que simule o comando `newusers`. Para isso, você deve criar um arquivo texto contendo as informações a respeito dos usuários, mantendo o mesmo padrão dos arquivos lidos pelo comando `newusers` (para descobrir o formato, consulte a página de manual: `$ man 8 newusers`). Como este arquivo conterá as senhas dos usuários, é importante removê-lo logo após a criação das contas.

   

   Utilize a variável de sistema `IFS` (*Internal Field Separator*) em seu *script* para definir o caractere ":" como campo que separa as informações sobre as contas.

   O *script shell* abaixo mostra um exemplo de solução para o problema proposto:

```
#!/bin/bash

IFS=':'
useradd="$( which useradd )"
groupadd="$( which groupadd )"


usage() {
  echo "  Usage: $0 -f NEWUSERS_FILE"
  echo "  File syntax: username:password:uid:gid:gecos:homedir:shell"
  exit 1
}


if [[ $EUID -ne 0 ]]; then
  echo "  [*] Not root!" 1>&2
  exit 1
fi

while getopts ":f:" opt; do
  case "$opt" in
    f)
      file=${OPTARG}
      ;;
    *)
      usage
      ;;
  esac
done

[ -z $file ] && { echo "  [*] No file?"; usage; }

while read username password uid gid gecos homedir shell; do
  if egrep "^${username}:" /etc/passwd &> /dev/null; then
    echo "  [*] User $username already exists, skipping..."
  elif getent passwd | cut -d':' -f3 | grep "$uid" &> /dev/null; then
    echo "  [*] UID $uid already exists, skipping..."
  elif getent group | cut -d':' -f3 | grep "$gid" &> /dev/null; then
    echo "  [*] GID $gid already exists, skipping..."
  else
    hpass="$( mkpasswd -m sha-512 -s <<< $pass )"
    $groupadd $username -g $gid
    $useradd $username -p $( mkpasswd -m sha-512 -s <<< $password)  -u $uid -g $gid
-c "$gecos" -d $homedir -s $shell
    cp -r /etc/skel $homedir
    chown -R $username:$username $homedir
  fi
done < "$file"
```

Um arquivo de entrada com sintaxe válida para o *script* acima seria como se segue:

```
usuario1:rnpesr:1101:1101::/home/usuario1:/bin/bash
usuario2:rnpesr:1102:1102::/home/usuario2:/bin/bash
usuario3:rnpesr:1103:1103::/home/usuario3:/bin/bash
```

2) Verificando e modificando informações de contas de usuário

Após a criação de uma conta, é fundamental que o administrador verifique se ela foi criada corretamente.

1. Entre no sistema com o usuário criado no item 3 da atividade 1 e execute os comandos indicados para verificação de uma conta.

```
$ ssh instrutor@localhost
instrutor@localhost's password:

$ id
uid=1002(instrutor) gid=1002(instrutor) grupos=1002(instrutor)
$ pwd
/instrutor
$ ls -la
total 8
drwxr-xr-x  2 instrutor instrutor 4096 Ago  7 14:42 .
drwxr-xr-x 23 root      root      4096 Ago  7 14:42 ..
```

2. Seria possível inserir o número de telefone de trabalho desse mesmo usuário, junto com a informação de quem ele é? Faça isso e torne a checar se a sua mudança surtiu efeito.

```
# chfn -w 6198765432 instrutor
# finger -l instrutor
Login: instrutor                        Name:
Directory: /instrutor                   Shell: /bin/sh
Office Phone: 619-876-5432
Last login Tue Aug  7 14:44 (-03) on pts/1 from localhost
No mail.
No Plan.
```

3) Criando grupos de usuários

O recurso de grupos de usuários é muito útil para compartilhar informações. No momento em que a conta `instrutor` foi criada, no item 3 da atividade 1 deste roteiro, o grupo primário ficou sendo o seu próprio nome de usuário. Isso ocorre sempre que não é atribuído um valor para o grupo primário, no momento da criação de um novo usuário. Como o usuário criado não faz parte de outro grupo, a não ser do seu próprio, ele somente poderá acessar seus arquivos ou aqueles

arquivos para os quais haja permissão de acesso para outros usuários.

1. Use o comando apropriado para criar um grupo chamado `grupoteste`.

```
# addgroup grupoteste
Adicionando grupo 'grupoteste' (GID 1003) ...
Concluído.
```

2. Liste o arquivo `/etc/group` e anote o `GID` que foi atribuído ao grupo criado.

```
# getent group | egrep '^grupoteste:' | cut -d':' -f3
1003
```

3. Aproveite para observar, no arquivo `/etc/group`, quais são os outros grupos existentes no sistema. Qual o grupo associado ao usuário `root`?

```
# getent group | grep root
root:x:0:
```

   O grupo `root`, que é o grupo primário do superusuário do sistema.

4. Altere o grupo primário do usuário `instrutor`, de modo que este passe a ser o grupo criado no item 1 da atividade 3, `grupoteste`.

```
# usermod -g grupoteste instrutor
# getent passwd | egrep '^instrutor:'
instrutor:x:1002:1003:,,6198765432,:/instrutor:/bin/sh
```

5. Se autentique no sistema utilizando a sua conta e inclua seu usuário como administrador do grupo `grupoteste`. Em seguida inclua o usuário `instrutor` no grupo `grupoteste`. Você conseguiu executar as tarefas propostas? Por quê? Como você deve fazer para realizar as tarefas?

```
$ gpasswd -a instrutor grupoteste
gpasswd : Permissão negada.
```

   Não, porque somente o usuário `root` pode cadastrar administradores em um grupo. Os comandos para viabilizar essa tarefa seriam:

```
# gpasswd -A aluno grupoteste
# logout

$ whoami
aluno
$ gpasswd -a instrutor grupoteste
Adicionando usuário instrutor ao grupo grupoteste
```

6. Altere novamente o grupo primário do usuário `instrutor` para o grupo `instrutor`.

```
# usermod -g instrutor instrutor
# getent passwd | egrep '^instrutor:'
instrutor:x:1002:1002:,,6198765432,:/instrutor:/bin/sh
```

4) Incluindo usuários em grupos secundários

1. Editando o arquivo `/etc/group`, inclua, no grupo `grupoteste`, o usuário criado no terceiro item da atividade 1 desse roteiro (`instrutor`). Note que o grupo primário do usuário não deve mudar; continua sendo o nome do usuário.

   Inserir após o último caractere ":" na linha referente ao grupo `grupoteste`, o *username* do usuário `instrutor`.

```
# getent group | egrep '^grupoteste:'
grupoteste:x:1003:instrutor
# groups instrutor
instrutor : instrutor grupoteste
```

2. Agora, utilize um comando apropriado para inserir nesse mesmo grupo o usuário criado para você no primeiro item da atividade 1.

```
# groups marcelo
marcelo : marcelo

# usermod -a -G grupoteste marcelo
# groups marcelo
marcelo : marcelo grupoteste
```

5) Bloqueando contas de usuários

No Linux, é possível impedir temporariamente o acesso ao sistema mesmo que o usuário esteja utilizando uma conta com acesso liberado a este.

1. Utilizando um comando apropriado, bloqueie a conta criada para o instrutor e teste se obteve

sucesso no bloqueio.

```
# passwd -l instrutor
passwd: informação de expiração de senha alterada.

# ssh instrutor@localhost
instrutor@localhost's password:
Permission denied, please try again.
```

2. Agora desbloqueie a conta e faça o teste de acesso para verificar se sua alteração surtiu efeito.

```
# passwd -u instrutor
passwd: informação de expiração de senha alterada.

# ssh instrutor@localhost
instrutor@localhost's password:
$ pwd
/instrutor
```

Também pode-se utilizar o comando `# usermod -U USERNAME` para atingir o mesmo objetivo.

6) Removendo uma conta de usuário manualmente

No Linux, é possível executar uma mesma tarefa de diversas maneiras. Para um administrador de sistemas, é importante conhecer essas alternativas, porque elas podem ser úteis em situações específicas em que não seja possível utilizar um dado recurso ou ferramenta do sistema.

1. Sem utilizar o comando `userdel`, remova a conta criada para você no segundo item da atividade 1.

   Em ordem, deve-se executar as atividades espelho das que foram feitas anteriormente, quais sejam:

   - Remover entradas referente à conta nos arquivos:
     - `/etc/group`
     - `/etc/gshadow`
     - `/etc/passwd`
     - `/etc/shadow`
   - Remover o diretório *home* do usuário;
   - Remover as configurações de *quota*, caso tenham sido configuradas anteriormente.
   - O *script shell* abaixo mostra uma maneira como os comandos executados manualmente nesta atividade poderiam ser automatizados por um administrador de sistemas:

```
#!/bin/bash

BACKUP_DIR="/root/user_backups"

usage() {
  echo "  Usage: $0 -u USER [-b]"
  echo "  Use [-b] to backup user dir to /root before deletion."
  exit 1
}


if [[ $EUID -ne 0 ]]; then
  echo "  [*] Not root!" 1>&2
  exit 1
fi

backup=false
while getopts ":u:b" opt; do
  case "$opt" in
    u)
      user=${OPTARG}
      ;;
    b)
      backup=true
      ;;
    *)
      usage
      ;;
  esac
done

[ -z $user ] && { echo "  [*] No user?"; usage; }

if ! egrep "^${user}:" /etc/passwd &> /dev/null; then
  echo "  [*] User does not exist!"
  exit 1
fi

homedir=$( getent passwd | egrep "^$user:" | cut -d':' -f6 )

if $backup; then
  [ ! -d $BACKUP_DIR ] && mkdir $BACKUP_DIR
  tar czf $BACKUP_DIR/${user}.tar.gz $homedir
fi
rm -rf /home/$user

sed -i "/^$user:/d" /etc/group
sed -i "/^$user:/d" /etc/gshadow
sed -i "/^$user:/d" /etc/passwd
sed -i "/^$user:/d" /etc/shadow
```

```
# remove user from secondary groups
sed -r -i "s/,?${user},?/,/ ; s/:,/:/ ; s/,$//" /etc/group
```

2. Certifique-se de que esse usuário foi realmente excluído do sistema, utilizando um dos comandos que fornecem informações sobre os usuários.

```
# finger marcelo
finger: marcelo: no such user.
```

3. Crie novamente a conta removida no passo 1 desta atividade. Agora, antes de removê-la, faça um backup de seus dados de modo que o instrutor possa ter sobre eles o mesmo tipo de acesso que você.

   O *script* apontado no primeiro item desta atividade já faz o backup de arquivos (via opção `-b`). Caso o usuário tenha sido removido sem que seu *home* tenha sido apagado (por exemplo, via comando `userdel`), pode-se fazer o backup dos dados da seguinte forma:

```
# tar czf /instrutor/marcelo.tar.gz /home/marcelo && rm -rf /home/marcelo
tar: Removendo `/' inicial dos nomes dos membros

# chown instrutor.instrutor /instrutor/marcelo.tar.gz

# ls /instrutor/
marcelo.tar.gz
```

7) Obtendo informações sobre usuários

Muitas vezes, é necessário obter informações sobre os usuários de um sistema. Dois comandos que fornecem informações sobre usuários são `finger` e `id`.

1. Verifique os parâmetros do usuário `instrutor` utilizando esses comandos, e descreva a diferença entre os dois a partir dos resultados obtidos. Consulte as páginas de manual para verificar as opções disponíveis nestes comandos.

```
$ id instrutor
uid=1002(instrutor) gid=1002(instrutor) grupos=1002(instrutor),1003(grupoteste)

$ finger instrutor
Login: instrutor                        Name:
Directory: /instrutor                   Shell: /bin/bash
Office Phone: 619-876-5432
Last login Tue Aug  7 15:45 (-03) on pts/1 from localhost
No mail.
No Plan.
```

O comando `id` mostra os grupos do usuário e seu UID enquanto o comando `finger` mostra informações como: diretório *home*, shell, *username*, GECOS, terminal utilizado pelo usuário, etc.

8) Removendo contas de usuários

1. Utilizando os comandos apropriados, remova a conta criada para o instrutor. Não se esqueça de que um grupo foi especialmente criado para ele e que ele também possui um grupo secundário.

```
# userdel -r instrutor
# getent passwd | egrep '^instrutor:'
# getent group  | egrep ',?instrutor,?'
#
```

9) Alterando o grupo a que um arquivo pertence

O arquivo `/etc/passwd` contém informações importantes sobre os usuários do sistema. Esse arquivo pertence ao usuário `root` e ao grupo `root`. As permissões de acesso desse arquivo definem que ele só poderá ser modificado pelo usuário `root`.

1. Faça com que esse arquivo pertença ao grupo `grupoteste`, criado na atividade 3. Com isso, os usuários desse grupo, incluindo o usuário criado na atividade 1 poderão acessar esse arquivo por meio das permissões definidas para os usuários do grupo.

```
# chgrp grupoteste /etc/passwd
# ls -ld /etc/passwd
-rw-r--r-- 1 root grupoteste 1612 Ago  7 16:12 /etc/passwd
```

10) Alterando permissões de acesso de arquivos

É muito comum o administrador ter que modificar a permissão de arquivos para possibilitar ou impedir que eles sejam lidos ou modificados por diferentes categorias de usuários. A melhor forma de fazer isso é utilizando o comando `chmod`.

1. O arquivo `/etc/passwd` tem apenas permissão de leitura para os usuários do seu grupo proprietário. Use o comando `chmod` para atribuir permissão de escrita ao grupo proprietário desse arquivo. A permissão de escrita nesse arquivo é inicialmente atribuída apenas ao usuário proprietário do arquivo.

```
# chmod 664 /etc/passwd
# ls -ld /etc/passwd
-rw-rw-r-- 1 root grupoteste 1612 Ago  7 16:12 /etc/passwd
```

Alternativamente, pode-se usar também o comando `# chmod g+w /etc/passwd` para atingir o

mesmo objetivo.

2. O setor de controladoria de uma empresa só possuía um funcionário, que pediu demissão. Como não há um diretório específico para armazenar os arquivos do setor, todos os seus arquivos de trabalho estão armazenados em seu diretório *home*. Que passos você deve fazer para disponibilizar estes arquivos para o novo funcionário que será contratado e para que este tipo de problema não volte a ocorrer?

   - Crie o grupo controladoria:

     ```
     # addgroup controladoria
     Adicionando grupo 'controladoria' (GID 1002) ...
     Concluído.
     ```

   - Crie a conta do novo funcionário e defina o grupo `controladoria` como seu grupo primário:

     ```
     # useradd -m -g controladoria funcionario
     # ls -lha /home/ | egrep ' funcionario$'
     drwxr-xr-x  2 funcionario controladoria 4,0K Ago  7 16:22 funcionario
     ```

   - Crie o diretório `/home/controladoria`:

     ```
     # mkdir /home/controladoria
     # chgrp controladoria /home/controladoria
     # chmod g+w /home/controladoria/
     # ls -lha /home/ | egrep ' controladoria$'
     drwxrwxr-x  2 root        controladoria 4,0K Ago  7 16:24 controladoria
     ```

   - Habilite o *sticky bit* para o diretório `/home/controladoria`, de forma que todos os membros do grupo `controladoria` possam criar arquivos ali, mas apenas o dono de cada arquivo possa apagá-los:

     ```
     # chmod +t /home/controladoria/
     # ls -lha /home/ | egrep ' controladoria$'
     drwxrwxr-t  2 root        controladoria 4,0K Ago  7 16:24 controladoria
     ```

   - Mova os arquivos do antigo funcionário para o diretório `/home/controladoria`:

     ```
     # cp -a /home/antigo_funcionario /home/controladoria
     # ls /home/controladoria
     antigo_funcionario
     ```

     Redefina as permissões dos arquivos do antigo funcionário:

```
# chown -R root.controladoria /home/controladoria
```

- Remova a conta do antigo funcionário:

```
# userdel -r antigo_funcionario
```

- Oriente o novo funcionário para que ele só armazene os arquivos relacionados ao setor de controladoria no diretório `/home/controladoria`, e seus arquivos pessoais em `/home/funcionario`.

Por motivos de segurança, ao final das atividades, retorne a permissão e o grupo do arquivo `/etc/passwd` para os valores originais.

```
# chown root.root /etc/passwd
# chmod 644 /etc/passwd
# ls -lh /etc/passwd
-rw-r--r-- 1 root root 1,7K Ago  7 16:22 /etc/passwd
```