7. /etc/passwd 文件的第四个字段是什么?

a. 主目录
b. UID
c. 登录 shell
d. 主要组

海量视频题库 myitpub.com QQ:5565462
www.52myit.com

解决方案

描述用户和组概念

选择以下问题的正确答案：

1. 哪一项表示在最基本的层面上标识用户的编号？

a. 主要用户
b. UID
c. GID
d. 用户名

2. 哪一项表示提供用户命令行提示符的程序？

a. 主 shell
b. 主目录
c. 登录 shell
d. 命令名称

3. 哪一项目或文件表示本地组信息的位置？

a. 主目录
b. `/etc/passwd`
c. `/etc/GID`
d. `/etc/group`

4. 哪一项目或文件表示用户个人文件的位置？

a. 主目录
b. 登录 shell
c. `/etc/passwd`
d. `/etc/group`

5. 哪一项表示在最基本的层面上标识组的编号？

a. 主要组
b. UID
c. GID
d. groupid

6. 哪一项目或文件表示本地用户帐户信息的位置？

a. 主目录
b. `/etc/passwd`
c. `/etc/UID`
d. `/etc/group`

海量视频题库 myitpub.com QQ:556462
www.52myit.com

7. /etc/passwd 文件的第四个字段是什么?

a. 主目录
b. UID
c. 登录 shell
d. 主要组

海量视频题库 myitpub.com QQ:5565462
www.52myit.com

获取超级用户访问权限

培训目标

学完本节后，您将能够切换到超级用户帐户来管理 Linux 系统，并通过 **sudo**命令授予其他用户超级用户访问权限。

超级用户

大多数操作系统具有某种类型的超级用户，即具有系统全部权限的用户。在红帽企业 Linux 中，此为 `root` 用户。该用户的特权高于文件系统上的一般特权，用于管理系统。要执行诸如安装或删除软件以及管理系统文件和目录等任务，必须将特权升级到 `root` 用户。

普通用户能控制大部分的设备，但也有一些例外，只有此时才需要 `root` 用户。例如，普通用户可以控制可移动设备，如 USB 设备。因此，虽然普通用户可以添加和删除文件并可管理可移动的设备，但默认情况下，只有 `root` 用户才能管理“固定的”硬盘。

尽管如此，这种无限制的特权也带来了职责问题。`root` 用户拥有足以破坏系统的无限制权限：删除文件和目录、删除用户帐户，以及添加后门等。如果 `root` 帐户泄露，则其他人就有可能拥有系统的管理控制权限。在本课程中，建议管理员以普通用户身份登录，仅在需要时升级到 `root` 用户特权。

Linux 上的 `root` 帐户大致相当于Microsoft Windows 上的本地 Administrator 帐户。在 Linux 中，大多数系统管理员都作为无特权的用户登录，然后使用各种工具临时获得 `root` 特权。

> **警告**
> 过去在 Microsoft Windows 上的一种常见做法是本地 `Administrator` 用户直接登录系统，以履行系统管理员职责。虽然这可以在 Linux 实现，但红帽建议系统管理员不要直接以 `root` 身份登录。取而代之，系统管理员应当以普通用户登录，然后使用其他机制（例如 **su**、**sudo** 或 PolicyKit）临时获得超级用户特权。
>
> 以超级用户身份登录时，整个桌面环境都不必要地以管理员特权运行。在那种情形中，任何通常仅威胁用户帐户的安全漏洞就有可能威胁整个系统。

切换用户

su 命令可让用户切换至另一个用户帐户。如果您从普通用户帐户运行 **su**，系统会提示您输入要切换的帐户的密码。当以 `root` 用户身份运行 **su** 时，则无需输入用户密码。

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

如果省略用户名，则默认情况下 **su** 或 **su -** 命令会尝试切换到 `root`。

海量视频题库 myitpub.com QQ:556462
www.52myit.com

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

命令 **su** 将启动非登录 shell，而命令 **su -**（带有短划线选项）会启动登录 shell。两个命令的主要区别在于，**su -** 会将 shell 环境设置为如同以该用户身份重新登录一样，而 **su** 仅以该用户身份启动 shell，但使用的是原始用户的环境设置。

在大多数情况下，管理员应该运行 **su -** 以获得包含目标用户常规环境设置的 shell。有关详细信息，请参阅 **bash**(1) man page。

> **注意**
> **su** 命令最常用于获得以另一个用户身份（通常是 root）运行的命令行界面（shell 提示符）。但是，如果结合 **-c** 选项，该命令的作用将与 Windows 实用程序 **runas** 一样，能够以另一个用户身份运行任意程序。运行 **info su** 来查看更多详情。

通过 SUDO 运行命令

有时为安全起见，root 用户的帐户可能根本没有有效的密码。在这种情况下，用户无法使用密码直接以 root 身份登录系统，也不能使用 **su** 获取交互式 shell。在这种情况下，一个可以用于获取 root 访问权限的工具是 **sudo**。

与 **su** 不同，**sudo** 通常要求用户输入其自己的密码以进行身份验证，而不是输入他们正尝试访问的用户帐户的密码。也就是说，用户使用 **sudo** 以 root 运行命令时，不需要知道 root 密码。相反，他们将使用自己的密码来验证访问权限。

此外，**sudo** 可以配置为允许特定用户像某个其他用户一样运行任何命令，或仅允许以该用户身份运行部分命令。

例如，如果 **sudo** 已配置为允许 user01 用户以 root 身份运行命令 **usermod**，那么 user01 就可运行以下命令来锁定或解锁用户帐户：

```
[user01@host ~]$ sudo usermod -L user02
[sudo] password for user01:
[user01@host ~]$ su - user02
Password:
su: Authentication failure
[user01@host ~]$
```

如果某用户尝试以另一用户身份运行命令，但 **sudo** 配置不允许，该命令将被阻止，这次尝试也会被记录下来，并且默认情况下还会向 root 用户发送一封电子邮件。

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

使用 **sudo** 的另一个优点在于，执行的所有命令都默认为将日志记录到 **/var/log/secure** 中。

海量视频题库 mvitpub.com QQ:556462
www.52mvit.com

```
[user01@host ~]$ sudo tail /var/log/secure
...output omitted...
Feb  6 20:45:46 host sudo[2577]:  user01 : TTY=pts/0 ; PWD=/home/user01 ;
 USER=root ; COMMAND=/sbin/usermod -L user02
...output omitted...
```

在红帽企业 Linux 7 和红帽企业 Linux 8 中，`wheel` 组的所有成员都可以使用 **sudo** 以任何用户身份运行命令，包括 `root` 在内。系统将提示用户输入其自己的密码。这是与红帽企业 Linux 6 及更早版本相比的变化，在那些版本中，`wheel` 组的成员默认情况下不会获得这种管理访问权限。

> **警告**
> RHEL 6 默认为不授予 `wheel` 组任何特权。过去一直出于非标准意图在使用这个组的网站可能会因为 RHEL 7 和 RHEL 8 自动授予 `wheel` 组所有成员完整的 **sudo** 特权而感到讶异。这可能会导致未经授权的用户获得 RHEL 7 和 RHEL 8 系统的管理访问权限。
>
> 在过去，类 UNIX 系统利用 `wheel` 组成员资格授予或控制超级用户访问权限。

通过 Sudo 获取交互式 Root Shell

如果系统上的非管理员用户帐户能够使用 **sudo** 来运行 **su** 命令，则可以从该帐户运行 **sudo su -** 来获取 `root` 用户的交互式 shell。这是因为 **sudo** 将以 `root` 用户身份运行 **su -**，而 `root` 用户无需输入密码即可使用 **su**。

通过 **sudo** 访问 `root` 帐户的另一种方式是使用 **sudo -i** 命令。这将切换到 `root` 帐户并运行该用户的默认 shell（通常为 **bash**）及关联的 shell 登录脚本。如果您只想运行 shell，可使用 **sudo -s** 命令。

例如，管理员可以获取 AWS EC2 实例上 `root` 用户的交互式 shell，方法是使用 SSH 公钥身份验证作为普通用户 `ec2-user` 登录，再运行 **sudo -i** 来获取 `root` 用户的 shell。

```
[ec2-user@host ~]$ sudo -i
[sudo] password for ec2-user:
[root@host ~]#
```

sudo su - 命令与 **sudo -i** 的行为不完全相同。相关内容会在本节的末尾部分进行简要讨论。

配置 Sudo

sudo 的主配置文件为 **/etc/sudoers**。如果多个管理员试图同时编辑该文件，为了避免出现问题，只应使用特殊的 **visudo** 命令进行编辑。

例如，以下来自 **/etc/sudoers** 文件的一行可为组 `wheel` 的成员启用 **sudo** 访问权限。

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

在该行中，**%wheel** 是规则适用的用户或组。**%** 指定这是一个组，即组 `wheel`。**ALL=(ALL)** 指定在可能包含此文件的任何主机上，`wheel` 可以运行任何命令。最后的 **ALL** 指定 `wheel` 可以像系统上的任何用户一样运行这些命令。

海量视频题库 mytpub.com QQ:5565462
www.52mvt.com

默认情况下，**/etc/sudoers** 还包含 **/etc/sudoers.d** 目录中所有文件的内容，作为配置文件的一部分。这样管理员只需将相应的文件放入该目录中，即可为用户添加 **sudo** 访问权限。

> **注意**
> 使用 **/etc/sudoers.d** 目录下的补充文件比较简单方便。只需将文件复制到目录中或从目录中删除文件，即可启用或禁用 **sudo** 访问权限。
>
> 在本课程中，您将在 **/etc/sudoers.d** 目录中创建和删除文件，从而为用户和组配置 **sudo** 访问权限。

若要为用户 `user01` 启用完整的 **sudo** 访问权限，可创建含有以下内容的 **/etc/sudoers.d/user01**：

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

若要为组 `group01` 启用完整的 **sudo** 访问权限，可创建含有以下内容的 **/etc/sudoers.d/group01**：

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

此外，也可以设置 **sudo**，以允许用户在不输入密码的前提下以其他用户身份运行命令：

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

虽然授予用户或组此级别访问权限存在明显的安全风险，但它经常用于云实例、虚拟机和调配系统，以帮助配置服务器。必须仔细保护具有该访问权限的帐户；它可能需要 SSH 公钥身份验证，以便远程系统上的用户能够访问它。

例如，Amazon Web Services Marketplace 中适用于红帽企业 Linux 的官方 AMI 在出厂时锁定了 `root` 和 `ec2-user` 用户的密码。`ec2-user` 用户帐户设置为允许通过 SSH 公钥身份验证进行交互式远程访问。用户 `ec2-user` 也可以作为 `root` 用户运行任何命令，且不必输入密码，因为 AMI 的 **/etc/sudoers** 文件的最后一行设置如下：

```
ec2-user  ALL=(ALL)  NOPASSWD: ALL
```

可以为 **sudo** 重新启用输入密码的要求或进行其他更改来增强安全性，以此作为系统配置流程的一个部分。

海量视频题库 myitpub.com QQ:5565462
www.52myit.com

注意

在本课程中，常用的是 **sudo su -**，而非 **sudo -i**。这两个命令都有效，但它们之间存在一些细微差别。

sudo su - 命令可以完全像正常登录那样设置 root 环境，因为 **su -** 命令会忽略 **sudo** 所做的设置并从头开始设置环境。

而 **sudo -i** 命令的默认配置实际上会设置在一些细节上与正常登录不同的 root 用户环境。例如，它设置的 PATH 环境变量便略有不同。这会影响 shell 查找命令的位置。

通过用 **visudo** 编辑 **/etc/sudoers**，可以让 **sudo -i** 的行为与 **su -** 更为相似。找到行

```
Defaults    secure_path = /sbin:/bin:/usr/sbin:/usr/bin
```

并替换为以下两行：

```
Defaults      secure_path = /usr/local/bin:/usr/bin
Defaults>root secure_path = /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin
```

在大多数情况下，这不是主要区别。但是，为了让系统上的 PATH 设置与默认的 **/etc/sudoers** 文件保持一致，本课程的作者在示例中主要使用了 **sudo su -**。

参考文献

su(1)、**sudo**(8)、**visudo**(8) 和 **sudoers**(5) man page

info libc persona （GNU C 库参考书册）

· 第 30.2 节：进程的角色

（请注意，必须安装 glibc-devel 软件包，此 info 节点才可用。）

海量视频题库 myitpub.com QQ:556546 2
www.52myit.com

指导练习

获取超级用户访问权限

在本练习中，您将练习切换到 root 帐户，以及以 root 身份运行命令。

成果

您应能够：

- 使用 **sudo** 切换到 root，并以 root 身份访问交互式 shell，而不必知道超级用户的密码。
- 解释 **su** 和 **su -** 可以如何通过运行或不运行登录脚本来影响 shell 环境。
- 使用 **sudo** 以 root 身份运行其他命令。

在你开始之前

以 student 用户身份并使用 student 作为密码登录 workstation。

在 workstation 上，运行 **lab users-sudo start** 脚本来开始本练习。此脚本会创建必要的用户帐户和文件，以正确设置环境。

```
[student@workstation ~]$ lab users-sudo start
```

1. 从 workstation，以 student 用户身份打开连接 servera 的 SSH 会话。

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

2. 探索 student 的 shell 环境。查看当前的用户和组信息，并显示当前的工作目录。此外，还要查看指定用户主目录的环境变量以及用户可执行文件的位置。

2.1. 运行 **id** 来查看当前的用户和组信息。

```
[student@servera ~]$ id
uid=1000(student) gid=1000(student) groups=1000(student),10(wheel)
 context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

2.2. 运行 **pwd** 以显示当前的工作目录。

```
[student@servera ~]$ pwd
/home/student
```

2.3. 显示 HOME 和 PATH 变量的值，以分别确定主目录和用户可执行文件的路径。

海量视频题库 myitpub.com QQ:5565462
www.52myit.com

```
[student@servera ~]$ echo $HOME
/home/student
[student@servera ~]$ echo $PATH
/home/student/.local/bin:/home/student/bin:/usr/local/bin:/usr/bin:/usr/local/
sbin:/usr/sbin
```

3. 在非登录 shell 中切换到 `root`，并探索新的 shell 环境。

3.1. 在 shell 提示符下运行 **sudo su**，变为 `root` 用户。

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

3.2. 运行 **id** 来查看当前的用户和组信息。

```
[root@servera student]# id
uid=0(root) gid=0(root) groups=0(root)
 context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

3.3. 运行 **pwd** 以显示当前的工作目录。

```
[root@servera student]# pwd
/home/student
```

3.4. 显示 HOME 和 PATH 变量的值，以分别确定主目录和用户可执行文件的路径。

```
[root@servera student]# echo $HOME
/root
[root@servera student]# echo $PATH
/sbin:/bin:/usr/sbin:/usr/bin:/usr/local/sbin:/usr/local/bin
```

> 如果您已有一些 Linux 和 **su** 命令相关的经验，您可能会预计到，不带短划线 (**-**) 选项使用 **su** 来变为 `root` 会导致您保留 `student` 的当前 PATH。这并未发生。正如您在下一步中所见，这也不是 `root` 的正常 PATH。
>
> 发生了什么情况？差别在于您没有直接运行 **su**。相反，您使用 **sudo** 以 `root` 身份运行了 **su**，因为您没有掌握超级用户的密码。出于安全方面的原因，**sudo** 命令最初覆盖来自初始环境的 PATH。在初次覆盖之后运行的任何命令仍然可以更新 PATH 变量，您会在后续步骤中看到这一点。

3.5. 从 `root` 用户的 shell 退出，以返回到 `student` 用户的 shell。

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

4. 在登录 shell 中切换到 `root`，并探索新的 shell 环境。

4.1. 在 shell 提示符下运行 **sudo su -**，变为 `root` 用户。

海量视频题库 myitpub.com QQ:556462
www.52myit.com

```
[student@servera ~]$ sudo su -
[root@servera ~]#
```

注意 shell 提示符与上一步中 **sudo su** 的相比的区别。

sudo 可能会提示您输入 student 密码，也可能不提示，这取决于 **sudo** 的超时期限。默认的超时期限是五分钟。如果您在最近五分钟内通过 **sudo** 的身份验证，**sudo** 不会提示您输入密码。如果自您通过 **sudo** 的身份验证起已超过五分钟，您需要输入 student 作为密码以通过 **sudo** 的身份验证。

4.2. 运行 **id** 来查看当前的用户和组信息。

```
[root@servera ~]# id
uid=0(root) gid=0(root) groups=0(root)
 context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

4.3. 运行 **pwd** 以显示当前的工作目录。

```
[root@servera ~]# pwd
/root
```

4.4. 显示 HOME 和 PATH 变量的值，以分别确定主目录和用户可执行文件的路径。

```
[root@servera ~]# echo $HOME
/root
[root@servera ~]# echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin
```

如前一步中所见，在 **sudo** 重置了 student 用户 shell 环境设置的 PATH 变量后，**su -** 命令运行了 root 的 shell 登录脚本，并将 PATH 变量设置为另一个值。如果不带短划线 (**-**) 选项，**su** 命令不会那么做。

4.5. 从 root 用户的 shell 退出，以返回到 student 用户的 shell。

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

5. 验证 operator1 用户是否配置为可利用 **sudo** 以任何用户身份运行任何命令。

```
[student@servera ~]$ sudo cat /etc/sudoers.d/operator1
operator1 ALL=(ALL) ALL
```

6. 变为 operator1，再查看 **/var/log/messages** 的内容。将 **/etc/motd** 复制到 **/etc/motdOLD**，并将它删除 (/etc/motdOLD)。这些操作需要管理权限，因此请使用 **sudo** 以超级用户身份运行这些命令。不要使用 **sudo su** 或 **sudo su -** 切换到 root。使用 redhat 作为 operator1 的密码。

6.1. 切换到 operator1。

海量视频题库 myitpub.com QQ:5565462 www.52myit.com

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

6.2. 尝试查看 **/var/log/messages** 的最后五行，但不使用 **sudo**。这应该会失败。

```
[operator1@servera ~]$ tail -5 /var/log/messages
tail: cannot open '/var/log/messages' for reading: Permission denied
```

6.3. 尝试使用 **sudo** 来查看 **/var/log/messages** 的最后五行。这应该会成功。

```
[operator1@servera ~]$ sudo tail -5 /var/log/messages
[sudo] password for operator1: redhat
Jan 23 15:53:36 servera su[2304]: FAILED SU (to operator1) student on pts/1
Jan 23 15:53:51 servera su[2307]: FAILED SU (to operator1) student on pts/1
Jan 23 15:53:58 servera su[2310]: FAILED SU (to operator1) student on pts/1
Jan 23 15:54:12 servera su[2322]: (to operator1) student on pts/1
Jan 23 15:54:25 servera su[2353]: (to operator1) student on pts/1
```

注意
上述输出可能在您的系统上有所不同。

6.4. 尝试为 **/etc/motd** 制作副本 **/etc/motdOLD**，但不使用 **sudo**。这应该会失败。

```
[operator1@servera ~]$ cp /etc/motd /etc/motdOLD
cp: cannot create regular file '/etc/motdOLD': Permission denied
```

6.5. 尝试使用 **sudo** 来为 **/etc/motd** 制作副本 **/etc/motdOLD**。这应该会成功。

```
[operator1@servera ~]$ sudo cp /etc/motd /etc/motdOLD
[operator1@servera ~]$
```

6.6. 尝试删除 **/etc/motdOLD**，但不使用 **sudo**。这应该会失败。

```
[operator1@servera ~]$ rm /etc/motdOLD
rm: remove write-protected regular empty file '/etc/motdOLD'? y
rm: cannot remove '/etc/motdOLD': Permission denied
[operator1@servera ~]$
```

6.7. 尝试使用 **sudo** 来删除 **/etc/motdOLD**。这应该会成功。

```
[operator1@servera ~]$ sudo rm /etc/motdOLD
[operator1@servera ~]$
```

6.8. 从 operator1 用户的 shell 退出，以返回到 student 用户的 shell。

海量视频题库 myitpub.com QQ:5565462
www.52myit.com

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

6.9. 从 servera 注销。

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

完成

在 workstation 上，运行 **lab users-sudo finish** 来完成本练习。此脚本将删除在练习开始时创建的用户帐户和文件，以确保环境清理干净。

```
[student@workstation ~]$ lab users-sudo finish
```

本引导式练习到此结束。

海量视频题库 myitpub.com QQ:556462
www.52myit.com

管理本地用户帐户

培训目标

学完本节后，您应能够创建、修改和删除本地用户帐户。

管理本地用户

可使用很多命令行工具来管理本地用户帐户。

从命令行创建用户

- **useradd username** 命令将创建一个名为 username 的新用户。它会设置用户的主目录和帐户信息，并为名为 username 的用户创建一个专用组。此时，该帐户尚未设置有效的密码，用户也要等设置了密码后才能登录。
- **useradd --help** 命令将显示可用于覆盖默认值的基本选项。在大多数情形中，可以将相同的选项用于 **usermod** 命令，以修改现有的用户。
- 一些默认值从 **/etc/login.defs** 文件中读取，如有效 UID 编号的范围和默认密码过期规则。此文件中的值仅在创建新用户时使用。更改此文件对现有用户毫无影响。

从命令行修改现有的用户

- **usermod --help** 命令显示可用于修改帐户的基本选项。一些常见选项包括：

USERMOD 选项：	用法
-c, --comment COMMENT	将用户的真实姓名添加到注释字段。
-g, --gid GROUP	为用户帐户指定主要组。
-G, --groups GROUPS	为用户帐户指定补充组的逗号分隔列表。
-a, --append	利用 **-G** 选项将补充组添加到用户当前的组成员集合中，而不是将补充组集合替换为新的集合。
-d, --home HOME_DIR	为用户帐户指定特定的主目录。
-m, --move-home	将用户的主目录移到新的位置。必须与 **-d** 选项搭配使用。
-s, --shell SHELL	为用户帐户指定特定的登录 shell。
-L, --lock	锁定用户帐户。
-U, --unlock	解锁用户帐户。

从命令行删除用户

- **userdel username** 命令从 **/etc/passwd** 中删除 username 的详细信息，但保留用户的主目录不变。

海量视频题库 myitpub.com QQ:556462
www.52myit.com

- **userdel -r username** 命令从 **/etc/passwd** 中删除 username 的详细信息，同时删除用户的主目录。

> **警告**
>
> 如果在未指定 **-r** 选项的情况下使用 **userdel** 删除某一用户，系统将具有未分配 UID 所拥有的文件。当由已删除用户拥有的文件在该用户的主目录之外存在时，也会发生这种情况。这种情况将导致信息泄露和其他安全问题。
>
> 在红帽企业 Linux 7 和红帽企业 Linux 8 中，**useradd** 命令为新用户分配第一个大于或等于 1000 的可用 UID，除非您通过 **-u** 选项进行明确指定。
>
> 这正是信息泄露如何产生的原因。如果第一个可用 UID 先前被分配给了一个从系统删除的用户帐户，那么旧用户的 UID 将被重新分配给新用户，以便为新用户提供旧用户剩余文件的所有权。
>
> 下列情景演示了这种情况。
>
> ```
> [root@host ~]# useradd user01
> [root@host ~]# ls -l /home
> drwx------. 3 user01  user01    74 Feb  4 15:22 user01
> [root@host ~]# userdel user01
> [root@host ~]# ls -l /home
> drwx------. 3    1000    1000   74 Feb  4 15:22 user01
> [root@host ~]# useradd user02
> [root@host ~]# ls -l /home
> drwx------. 3 user02     user02       74 Feb  4 15:23 user02
> drwx------. 3 user02     user02       74 Feb  4 15:22 user01
> ```
>
> 请注意，user02 现在拥有之前归 user01 所有的所有文件。
>
> 根据情况，解决这一问题的一个方案是，在删除创建了文件的用户时，同时从系统删除所有这些无人拥有的文件。另一种方案是手动为不同用户分配无人拥有的文件。root 用户可以使用 **find / -nouser -o -nogroup** 命令来查找所有无人拥有的文件和目录。

从命令行设置密码

- **passwd username** 命令可为 username 设置初始密码，或更改其现有的密码。
- root 用户可以将密码设为任何值。如果密码不符合最低建议标准，系统将显示消息；不过，之后会显示提示要求重新键入该新密码，所有令牌也会成功更新。

```
[root@host ~]# passwd user01
Changing password for user user01.
New password: redhat
BAD PASSWORD: The password fails the dictionary check - it is based on a
 dictionary word
Retype new password: redhat
passwd: all authentication tokens updated successfully.
[root@host ~]#
```

- 普通用户必须选择长度至少为 8 个字符，并且还不以字典词语、用户名或上一密码为基础的密码。

海量视频题库 myitpub.com QQ:556462
www.52myit.com

UID 范围

特定的 UID 编号和编号范围供红帽企业 Linux 用于特定的目的。

- UID 0 始终分配至超级用户帐户 **root**。
- UID 1-200 是一系列“系统用户”，由红帽静态分配给系统进程。
- UID 201-999 是一系列“系统用户”，供文件系统中没有自己的文件的系统进程使用。通常在安装需要它们的软件时，从可用池中动态分配它们。程序以这些“无特权”系统用户身份运行，以便限制它们仅访问正常运行所需的资源。
- UID 1000+ 是可供分配给普通用户的范围。

注意
在 RHEL 7 之前的惯例是，UID 1-499 用于系统用户，而 UID 500+ 则用于普通用户。由 **useradd** 和 **groupadd** 使用的默认范围可以在 **/etc/login.defs** 文件中更改。

参考文献
useradd(8)、**usermod**(8)、**userdel**(8) man page

海量视频题库 myitpub.com QQ:556546 2
www.52myit.com

指导练习

管理本地用户帐户

在本练习中，您将在系统上创建几个用户并为这些用户设置密码。

成果

您应能够为 Linux 系统配置额外的用户帐户。

在你开始之前

以 `student` 用户身份并使用 `student` 作为密码登录 `workstation`。

在 `workstation` 上，运行 **`lab users-manage start`** 脚本来开始本练习。此脚本会确保环境设置正确无误。

```
[student@workstation ~]$ lab users-manage start
```

1. 从 `workstation`，以 `student` 用户身份打开连接 `servera` 的 SSH 会话。

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

2. 在 `servera` 上，使用 **`sudo`** 切换到 `root`，以转换到 `root` 用户的 shell 环境。

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

3. 创建 `operator1` 用户，再确认它存在于系统中。

```
[root@servera ~]# useradd operator1
[root@servera ~]# tail /etc/passwd
...output omitted...
operator1:x:1001:1001::/home/operator1:/bin/bash
```

4. 将 `operator1` 的密码设为 **`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.
```

海量视频题库 myitpub.com QQ:556462
www.52myit.com

5. 创建另外两个用户 operator2 和 operator3。将它们的密码设为 **redhat**。

5.1. 添加 operator2 用户。将 operator2 的密码设为 **redhat**。

```
[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.
```

5.2. 添加 operator3 用户。将 operator3 的密码设为 **redhat**。

```
[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.
```

6. 更新 operator1 和 operator2 用户帐户，使它们分别包含 **Operator One** 和 **Operator Two** 注释。验证是否已成功添加注释。

6.1. 运行 **usermod -c**，以更新 operator1 用户帐户的注释。

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

6.2. 运行 **usermod -c**，以更新 operator2 用户帐户的注释。

```
[root@servera ~]# usermod -c "Operator Two" operator2
```

6.3. 确认 operator1 和 operator2 用户的注释已分别反映到其用户记录中。

```
[root@servera ~]# tail /etc/passwd
...output omitted...
operator1:x:1001:1001:Operator One:/home/operator1:/bin/bash
operator2:x:1002:1002:Operator Two:/home/operator2:/bin/bash
operator3:x:1003:1003::/home/operator3:/bin/bash
```

7. 删除 operator3用户，以及该用户的任何个人数据。确认已成功删除该用户。

7.1. 从系统中删除 operator3 用户。

```
[root@servera ~]# userdel -r operator3
```

7.2. 确认已成功删除 operator3。

海量视频题库 myitpub.com QQ:556462
www.52myit.com

```
[root@servera ~]# tail /etc/passwd
...output omitted...
operator1:x:1001:1001:Operator One:/home/operator1:/bin/bash
operator2:x:1002:1002:Operator Two:/home/operator2:/bin/bash
```

请注意，前面的输出不显示 operator3 的用户帐户信息。

7.3. 从 root 用户的 shell 退出，以返回到 student 用户的 shell。

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

7.4. 从 servera 注销。

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

完成

在 workstation 上，运行 **lab users-manage finish** 来完成本练习。此脚本会确保环境清理干净。

```
[student@workstation ~]$ lab users-manage finish
```

本引导式练习到此结束。

海量视频题库 myitpub.com QQ:556546 2
www.52myit.com

管理本地组帐户

培训目标

学完本节后，学员应能够创建、修改和删除本地组帐户。

管理本地组

组必须存在后，用户才能添加到其中。可以使用多款命令行工具管理本地组帐户。

从命令行创建组

- **groupadd** 命令用于创建组。不带选项时，**groupadd** 命令会在创建组时使用 **/etc/login.defs** 文件中指定范围内的下一个可用 GID。
- **-g** 选项指定要使用的组的特定 GID。

```
[user01@host ~]$ sudo groupadd -g 10000 group01
[user01@host ~]$ tail /etc/group
...output omitted...
group01:x:10000:
```

> **注意**
> 由于用户私有组（GID 1000 以上）是自动创建的，因此通常建议预备一系列 GID 待用于补充组。较高的范围可以避免与系统组 (GID 0-999) 产生冲突。

- **-r** 选项使用 **/etc/login.defs** 文件中所列有效系统 GID 范围内的 GID 创建系统组。**/etc/login.defs** 中的 SYS_GID_MIN 和 SYS_GID_MAX 配置项定义系统 GID 的范围。

```
[user01@host ~]$ sudo groupadd -r group02
[user01@host ~]$ tail /etc/group
...output omitted...
group01:x:10000:
group02:x:988:
```

从命令行修改现有的组

- **groupmod** 命令可更改现有组的属性。**-n** 选项指定组的新名称。

```
[user01@host ~]$ sudo groupmod -n group0022 group02
[user01@host ~]$ tail /etc/group
...output omitted...
group0022:x:988:
```

注意组名称已从 group02 更新为 group0022。

- **-g** 选项可指定新的 GID。

海量视频题库 myitpub.com QQ:556462
www.52myit.com

```
[user01@host ~]$ sudo groupmod -g 20000 group0022
[user01@host ~]$ tail /etc/group
...output omitted...
group0022:x:20000:
```

注意 GID 已从 **988** 更新为 **20000**。

从命令行删除组

· **groupdel** 命令可删除组。

```
[user01@host ~]$ sudo groupdel group0022
```

> **注意**
> 如果组是任何现有用户的主要组，则您无法删除它。与 **userdel** 一样，请检查所有文件系统，确保系统上不遗留由该组拥有的任何文件。

从命令行更改组成员资格

· 组成员资格通过用户管理进行控制。使用 **usermod -g** 命令来更改用户的主要组。

```
[user01@host ~]$ id user02
uid=1006(user02) gid=1008(user02) groups=1008(user02)
[user01@host ~]$ sudo usermod -g group01 user02
[user01@host ~]$ id user02
uid=1006(user02) gid=10000(group01) groups=10000(group01)
```

· 使用 **usermod -aG** 命令，将用户添加到某一补充组。

```
[user01@host ~]$ id user03
uid=1007(user03) gid=1009(user03) groups=1009(user03)
[user01@host ~]$ sudo usermod -aG group01 user03
[user01@host ~]$ id user03
uid=1007(user03) gid=1009(user03) groups=1009(user03),10000(group01)
```

> **重要**
> 使用 **-a** 选项可使 **usermod** 函数处于 append 模式。如果没有 **-a**，就会从当前未包含在 **-G** 选项列表中的任何补充组中删除该用户。

参考文献

group(5)、**groupadd**(8)、**groupdel**(8) 和 **usermod**(8) man page

海量视频题库 myitpub.com QQ:5565462
www.52myit.com

指导练习

管理本地组帐户

在本练习中，您将创建组，将其用作某些用户的补充组而不更改这些用户的主要组，并且为其中一个组配置 sudo 访问权限以便以 root 身份来运行命令。

成果

您应能够：

- 创建组并将其用作补充组。
- 为组配置 sudo 访问权限。

在你开始之前

以 student 用户身份并使用 student 作为密码登录 workstation。

在 workstation 上，运行 **lab users-group-manage start** 脚本来开始本练习。此脚本会创建必要的用户帐户，以正确设置环境。

```
[student@workstation ~]$ lab users-group-manage start
```

1. 从 workstation，以 student 用户身份打开连接 servera 的 SSH 会话。

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

2. 在 servera 上，使用 **sudo** 切换到 root，继承 root 用户的完整环境。

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

3. 创建 operators 补充组，其 GID 为 30000。

```
[root@servera ~]# groupadd -g 30000 operators
```

4. 创建 admin，作为另外一个补充组。

```
[root@servera ~]# groupadd admin
```

5. 验证 operators 和 admin 这两个补充组是否都存在。

海量视频题库 myitpub.com QQ:556462
www.52myit.com

```
[root@servera ~]# tail /etc/group
...output omitted...
operators:x:30000:
admin:x:30001:
```

6. 确保用户 operator1、operator2 和 operator3 属于组 operators。

6.1. 将 operator1、operator2 和 operator3 添加到 operators。

```
[root@servera ~]# usermod -aG operators operator1
[root@servera ~]# usermod -aG operators operator2
[root@servera ~]# usermod -aG operators operator3
```

6.2. 确认这些用户已成功添加到该组中。

```
[root@servera ~]# id operator1
uid=1001(operator1) gid=1001(operator1) groups=1001(operator1),30000(operators)
[root@servera ~]# id operator2
uid=1002(operator2) gid=1002(operator2) groups=1002(operator2),30000(operators)
[root@servera ~]# id operator3
uid=1003(operator3) gid=1003(operator3) groups=1003(operator3),30000(operators)
```

7. 确保用户 sysadmin1、sysadmin2 和 sysadmin3 属于组 admin。为 admin 的所有组成员启用管理权限。验证 admin 组的任何成员都能运行管理命令。

7.1. 将 sysadmin1、sysadmin2 和 sysadmin3 添加到 admin。

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

7.2. 确认这些用户已成功添加到该组中。

```
[root@servera ~]# id sysadmin1
uid=1004(sysadmin1) gid=1004(sysadmin1) groups=1004(sysadmin1),30001(admin)
[root@servera ~]# id sysadmin2
uid=1005(sysadmin2) gid=1005(sysadmin2) groups=1005(sysadmin2),30001(admin)
[root@servera ~]# id sysadmin3
uid=1006(sysadmin3) gid=1006(sysadmin3) groups=1006(sysadmin3),30001(admin)
```

7.3. 检查 **/etc/group**，以验证补充组成员资格。

```
[root@servera ~]# tail /etc/group
...output omitted...
operators:x:30000:operator1,operator2,operator3
admin:x:30001:sysadmin1,sysadmin2,sysadmin3
```

7.4. 创建 **/etc/sudoers.d/admin** 文件，使得 admin 的成员拥有完整的管理特权。

海量视频题库 myitpub.com QQ:556462
www.52myit.com

```
[root@servera ~]# echo "%admin ALL=(ALL) ALL" >> /etc/sudoers.d/admin
```

7.5. 切换到 sysadmin1（admin 的成员），再验证您能够以 sysadmin1 身份运行 **sudo** 命令。

```
[root@servera ~]# su - sysadmin1
[sysadmin1@servera ~]$ sudo cat /etc/sudoers.d/admin
[sudo] password for sysadmin1: redhat
%admin ALL=(ALL) ALL
```

7.6. 从 sysadmin1 用户的 shell 退出，以返回到 root 用户的 shell。

```
[sysadmin1@servera ~]$ exit
logout
[root@servera ~]#
```

7.7. 从 root 用户的 shell 退出，以返回到 student 用户的 shell。

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

7.8. 从 servera 注销。

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

完成

在 workstation 上，运行 **lab users-group-manage finish** 来完成本练习。此脚本将删除在练习开始时创建的用户帐户。

```
[student@workstation ~]$ lab users-group-manage finish
```

本引导式练习到此结束。

海量视频题库 myitpub.com QQ:5565462
www.52myit.com

管理用户密码

培训目标

学完本节后，您应能够为用户设置密码管理策略，并且手动锁定和解锁用户帐户。

阴影密码和密码策略

加密的密码一度存储在全局可读的 **/etc/passwd** 文件中。这曾被认为具有合理的安全性，直到对加密密码的字典式攻击变得常见。那时，加密的密码被移动到只有 root 用户才能读取的独立 **/etc/shadow** 文件中。这种新文件也允许实施密码期限和到期功能。

与 **/etc/passwd** 相似，每个用户在 **/etc/shadow** 文件中都有对应的一行。下方显示了来自 **/etc/shadow** 的示例行，以及其九个冒号分隔字段。

```
❶user03:❷$6$CSsX...output omitted...:❸17933:❹0:❺99999:❻7:❼2:❽18113:❾
```

❶ 此密码所属帐户的用户名。
❷ 此用户的加密密码。加密密码的格式将在本节后面讨论。
❸ 上次更改密码的日期。其设置值为自 1970 年 1 月 1 日起的天数，并按 UTC 时区计算。
❹ 自用户上次更改密码以来到可以再次更改之前必须经过的最短天数。
❺ 在密码过期之前不进行密码更改可以经过的最长天数。空字段表示它不会根据上次更改以来的时间失效。
❻ 警告期。当用户在截止日期之前登录达到该天数时，会收到有关密码过期的警告。
❼ 非活动期。一旦密码过期，在这些天内仍可以接受登录。过了这一时期后，帐户将被锁定。
❽ 密码过期日。其设置值为自 1970 年 1 月 1 日起的天数，并按 UTC 时区计算。空字段表示它不会在特定的日期失效。
❾ 最后一个字段通常为空，预留给未来使用。

加密密码的格式

加密密码字段中存储了三段信息：所用的哈希算法、salt 及加密哈希值。每段信息由 **$** 符号分隔。

```
$❶6$❷CSsXcYG1L/4ZfHr/$❸2W6evvJahUfzfHpc9X.45Jc6H30E...output omitted...
```

❶ 此密码所用的哈希算法。数字 **6** 表示 SHA-512 哈希算法，这是红帽企业 Linux 8 中的默认算法。**1** 表示 MD5 哈希算法，**5** 表示 SHA-256 哈希算法。
❷ 用于加密密码的 salt。这原先是随机选取的。
❸ 用户密码的加密哈希值。salt 和未加密密码组合并加密，生成加密的密码哈希。

使用 salt 可以防止两个密码相同的用户在 **/etc/shadow** 文件中拥有相同的条目。例如，即使 user01 和 user02 都将 redhat 用作其密码，如果 salt 不同，那么它们在 **/etc/shadow** 中的加密密码也是不同的。

密码验证

用户尝试登录时，系统在 **/etc/shadow** 中查询用户的条目，将用户的 salt 和键入的未加密密码组合，再使用指定的哈希算法加密。如果结果与已加密哈希匹配，则用户键入了正确的密码。如果结

海量视频题库 myitpub.com QQ:3565462 www.52myit.com