第1章 Linux用户

1.多用户与多任务

用户指的是我们陆操作系统使用的账号。多用户就是指操作系统上有多用户。
任务就是指的我们登陆系统之后进行的操作，多任务指的就是可以同时打开多个软件，而且每个软件都可以运行，互不影响。
Windows可以创建多个用户，但是同时只允许1个用户登陆。所以widonws实际上是单用户多任务。
Linux可以创建多个用户并且可以多个用户同时登陆，不同的账户可以有不同的权限，并且允许多个用户同时操作多个任务。

2.Linux用户介绍

那么我们为什么需要在Linux下使用多账户呢？
1.因为Linux默认的账户root是超级管理员账户，权限太大，多人使用容易造成误操作，所以可以通过设置权限较小的普通账户来减少风险。
2.我们在Linux运行的服务都需要分配一个账户，如果使用root分配，那么假如软件有漏洞被黑客入侵了，那么黑客就能获得root权限，所以我们一般使用普通用户来启动服务以减少安全风险。

3.Linux用户角色划分

3.1 什么是UID和GID

在Linux系统中用户是分角色的，由于用户角色不同，权限和所完成的任务也不同；值得注意的是，对于Linux系统来说，用户的角色是通过UID和GID来识别的。
那么什么是UID和GID呢？
UID：用户ID，相当于用户的身份，是系统中唯一的。
GID：用户组ID，相当于用户所属的组，也是系统中唯一的。

3.2 超级用户 root

当我们装完Linux系统后都会默认创建root用户，他的UID和GID都是0。
root用户拥有最高权限，可以操作系统中的任何文件和命令。
不过因为root用户的权限太大了，所以我们一般不建议直接使用root登陆Linux进行操作。也不建议使用root用户运行服务。

3.3 普通用户

普通用户一般是由root权限的系统管理人员添加的，例如，oldzhang这类普通用户可以登录系统，但仅具备操作自己的家目录中的文件及目录权限，除此之前其他重要目录只有浏览的权限，不能删除和修改。
不过也有特殊的情况，比如说普通用户临时需要使用root权限运行某个命令或操作，这时我们可以通过给普通用户赋予一定的权限，允许普通用户临时提升权限，这种操作就叫做提权，后面我们会介绍提权的命令sudo。

3.4 虚拟用户

```
虚拟用户和普通用户的区别就是虚拟用户一般不允许登陆系统，大部分作用仅仅是用来运行服务。
```

4.Linux用户UID说明

用户UID	角色定义
0	超级管理员 最高权限 拥有核爆炸的能力
1~200	系统用户 用来运行系统自带的进程 安装完系统就默认创建的
201~999	系统用户 用来运行用户安装的程序 这些用户无需登录系统
1000+	普通用户 可以正常登陆系统的用户 权限比较小 能执行的任务有限

5.用户相关配置文件

在Linux系统中，与用户相关的配置文件主要包含如下两个：

```
/etc/passwd          #存储用户信息的文件。
/etc/shadow          #存储用户密码信息的文件。
```

接下来我们分别解读一下这两个配置文件的内容。

5.1 存储用户信息的文件/etc/passwd

```
[root@linux ~]# head -3 /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
```

接下来以root用户举例我们来解读一下各个字段的含义。

root	:x	:0	:0	:root	:/root	:/bin/bash
用户名称	: 用户密码	: 用户 UID	: 用户组 GID	: 用户说明	: 用户家目录	:shell 解释器

接下来我们查看一下这个文件的权限信息

```
[root@linux ~]# ll /etc/passwd
-rw-r--r--. 1 root root 882 12月 13 11:19 /etc/passwd     #通过查看我们发现所有用户都有读的权限
```

因为每个用户登陆的时候都需要取得UID和GID的信息来判断权限，所以/etc/passwd文件的权限为644,但这样一来就会带来安全问题，即所有的用户都可以读取/etc/passwd文件。所以Linux系统将存储用户密码信息的问价移动到了/etc/shadow文件。

5.2 存储用户密码信息的文件/etc/shadow

通过查看/etc/shadow我们会发现，这个文件的权限竟然是000，但其实root是个特殊，这个文件只有root可以读。

```
[root@linux ~]# ll /etc/shadow
----------. 1 root root 604 12月 13 11:19 /etc/shadow
```

通过head命令查看/etc/shadow文件的前3行,我们会发现root用户后年有一串字符串，这就是密码加密后的数据。

```
[root@linux ~]# head -3 /etc/shadow
root:$6$bfoL8mdF5BmaGgL9$v3HcmxA.Ri9IWzVpyJxiP32ZUvILUBO8ofQjnFfdExjfQrvKSaOv73GzRkAKVubZoRikvH/idtcnJW7GMGy4m.::0:99999:7:::
bin:*:17834:0:99999:7:::
daemon:*:17834:0:99999:7:::
```

接下来我们来了解下每个字段分别代表了什么意思，了解即可，因为一般我们都是通过命令来创建用户，并不需要我们去编辑这个文件。

字 段 名 称	注 释 说 明
用户名称	用户名称
用户密码	该密码是加密过的字符串
最近更改密码的时间	从 1970 年 1 月 1 日起，到用户最近一次更改密码的天数
禁止修改密码的天数	从 1970 年 1 月 1 日起，到用户可以更改密码的天数
用户必须更改口令的天数	从 1970 年 1 月 1 日起，到用户必须更改密码的天数
警告更改密码的期限	在用户密码过期之前多少天提醒用户更改密码
不活动时间	在用户密码过期之后到禁用账户的天数
失效时间	从 1970 年 1 月 1 日起，到用户被禁用的天数
标志	保留

第2章 Linux用户组

1.Linux用户组介绍

```
简单来说，Linux系统中的用户组(group)就是具有相同特征的用户(user)集合；
比如我们需要让多个用户具有相同的权限，比如修改某一个文件，那么我们只需要将多个用户添加到一个组里，然后对这个组进行权限操作就相当于对这个组里所有的用户进行相同的权限操作了。
```

2.用户与用户组的关系

用户和用户组对应的关系说明如下：

```
1对1：即一个用户可以存在一个组中，也可以是组中的唯一成员。
1对多：即一个用户存在于多个组中。
```

3.用户组配置文件

在Linux系统中，与用户组直接相关的主要配置文件也有两个：

```
/etc/group          #用户组信息文件
/etc/gshadow        #用户组密码信息文件
```

3.1 存储用户组信息文件/etc/group

/etc/group文件是存储用户组相关信息的文件，内容包括用户组名称，用户组GID等属性。/etc/group和/etc/passwd文件的权限相同，也是644.

```
[root@linux ~]# ll /etc/group
-rw-r--r--. 1 root root 459 12月 13 11:19 /etc/group
```

/etc/group文件的实际内容具体如下：

```
[root@linux ~]# head -3 /etc/group
root:x:0:
bin:x:1:
daemon:x:2:
```

每一列的解释如下：

字段名称	注释说明
用户组名	用户组的名称
用户组密码	通常不需要设置该密码，基于安全原因，该密码被记录在 /etc/gshadow 中，因此，显示为“x”。这有点类似于 /etc/shadow
GID	用户组的 ID
用户组成员	加入这个组的所有用户（附加组成员，即用“-G”加入的成员）

3.2 存储用户组密码信息文件/etc/gshadow

了解即可。

第3章 Linux用户及用户组管理

1.用户及用户组命令汇总

```
#与用户直接相关的命令
useradd     #添加用户
usermod     #修改用户信息
userdel     #删除用户及用户有关联的配置

#与用户密码直接相关的命令
passwd      #为用户设置或修改密码
chpasswd    #批量更新用户密码
chage           #修改用户密码属性信息

#与其他用户相关的命令
id              #查看用户的UID，GID及所属的用户组信息
su              #切换用户角色工具
sudo            #以root用户执行命令的工具，普通用户用来提权的重要工具
visudo      #用于编辑suders配置文件的工具

#用户组常用命令
groupadd  #添加用户组
groupdel  #删除用户组
groupmod  #修改用户组信息
gpasswd   #为用户组设置密码
groups    #显示用户所属的用户组
```

2.useradd添加用户命令

重要选项

```
-u 指定创建用户的UID，不允许和已存在的用户UID重复
-g 指定要创建用户所属的组
-d 指定要创建用户家目录，默认在/home下创建
-s 指定要创建用户的shell，默认是bash
-c 指定要创建用户注释信息，默认为空
-M 创建的用户时不创建家目录
-r 创建系统账户，默认无家目录
```

命令实践

创建普通用户oldzhang

```
useradd oldzhang
```

创建普通用户mysql,用户ID为1000，组ID为1000，不创建家目录，登陆shell为/sbin/nologin

```
groupadd mysql -g 1000
useradd mysql -u 1000 -g 1000 -M -s /sbin/nologin
```

3.usermod修改用户信息

重要选项

```
-u 指定要修改用户的UID
-g 指定要修改用户组GID
-d 指定要修改用户家目录
-s 指定要修改用户的shell
-c 指定要修改用户注释信息
-l 指定要修改用户的登陆名
-L 指定要锁定的用户
-U 指定要解锁的用户
```

命令实践

检查之前的用户信息

```
[root@linux ~]# grep "mysql" /etc/passwd
mysql:x:1000:1000::/home/mysql:/sbin/nologin
```

修改mysql用户的UID，家目录，登陆shell

```
usermod -s /bin/bash -d /home/mysql -u 2000 mysql
```

修改后再次查看用户信息的变化

```
[root@linux ~]# grep "mysql" /etc/passwd
mysql:x:2000:1000::/home/mysql:/bin/bash

[root@linux ~]# id mysql
uid=2000(mysql) gid=1000(mysql) 组=1000(mysql)
```

4.userdel删除用户信息

```
#删除用户，但不删除用户家目录
userdel mysql

#删除用户，同时连用户家目录一并删除
userdel -r mysql
```

5.groupadd添加用户组

```
#创建用户组，不指定gid
groupadd nginx

#创建用户组，并指定gid为2000
groupadd -g 2000 nginx
```

6.groupmod修改用户组信息

```
#修改nginx组ID为3000
groupmod -g 2000 nginx
```

7.groupdel删除用户组

```
注意：
1.删除组的时候不能有用户正在使用这个组
2.删除用户的时候默认会连组一起删掉

groupadd nginx
groupdel nginx
```

8.passwd修改用户密码

passwd命令可以修改用户密码，是工作中很常用的命令。普通用户和超级用户都可以运行passwd命令，但普通用户只能更改自身的用户密码，超级用户root则可以设置或修改所有用户的密码。

重要选项

```
--stdin     #从标准输入读取密码字符串。
-d          #删除用户的密码，使密码为空。仅root用户有权使用该选项。
-e          #使用户密码立即过期，将在用户下次登录时强制修改密码。仅root用户有权使用该选项。
-l          #锁定用户，被锁定用户将不能登录。仅root用户有权使用该选项。
-u                  #解除对用户的锁定。仅root用户有权使用该选项。
```

命令实践

```
#修改当前登陆用户的密码
passwd

#给指定的用户设置密码
passwd oldzhang

#免交互设置密码
echo "123456"|passwd --stdin oldzhang
```

9.用户查询相关命令

```
w                       #显示已经登陆的用户，并且展示他都做了些什么信息。
id                    #显示用户的uid,gid,组信息等。
last                  #显示已登录的用户列表及登陆时间等。
whoami          #查看当前登陆的用户
lastlog          #显示最近的所有系统用户的登录信息
```

10.拓展-切换用户后显示不正常

问题现象:

```
切换用户后命令提示符显示不正常
-bash-4.2$
```

问题原因:

```
因为用户家目录下的环境变量文件缺失了，所以导致切换用户后读取不到正确的环境变量。
这里涉及到一个目录/etc/skel
/etc/skel目录是用来存放新用户环境变量文件的目录，当我们添加新用户时，这个目录下的所有文件会自动被复制到新添加的用户的家目录下；
默认情况下，/etc/skel 目录下的所有文件都是隐藏文件（以.点开头的文件）；
通过修改、添加、删除/etc/skel目录下的文件，我们可为新创建的用户提供统一的、标准的、初始化用户环境（给所有新用户的一个家默认样子）。
```

问题解决:

```
#查看用户环境变量目录
[root@linux ~]# ll -a /etc/skel/
总用量 24
drwxr-xr-x.  2 root root   62 4月  11 2018 .
drwxr-xr-x. 82 root root 8192 3月  24 20:14 ..
-rw-r--r--.  1 root root   18 10月 31 2018 .bash_logout
-rw-r--r--.  1 root root  193 10月 31 2018 .bash_profile
-rw-r--r--.  1 root root  231 10月 31 2018 .bashrc

#切换用户
[root@linux ~]# su - oldboy

#将环境变量文件都复制过来
-bash-4.2$ cp /etc/skel/.bash* .

```

```
#退出登陆后重新登陆查看发现已经正常了
-bash-4.2$ 登出
[root@linux ~]# su - oldboy
上一次登录：五 3月 26 16:40:00 CST 2021pts/0 上
[oldboy@linux ~]$
```

第3章 Linux用户身份切换su

1.su命令介绍

在Linux系统中，由于超级用户root的权限太大了，如果管理不好，就会带来严重的系统安全隐患。
因此在工作中大部分操作都是以普通用户的身份完成的，但是偶尔有些任务普通用户完成不了，需要临时使用超级管理员的权限，这种场景要怎么解决呢？
这里就不得不提到两个重要的命令su和sudo了。

su命令是直接将普通用户切换到root用户，前提是得知道root用户的密码。
sudo命令是临时以超级管理员权限运行某些命令，运行完后身份还是普通用户。

su命令可以在用户之间进行切换，超级权限用户root向普通用户切换不需要密码验证。
其他用户直接切换或者切换到root，都需要切换用户的密码验证。

2.预备知识

在使用su命令之前，我们先要学习一些预备知识

2.1 shell分类

Linux Shell主要分为如下几类：
交互式shell，等待用户输入执行的命令(终端操作,需要不断提示)
非交互式shell，执行shell脚本，脚本执行结束后shell自动退出
登陆shell，需要输入用户名和密码才能进入Shell，日常接触的最多的一种
非登陆shell，不需要输入用户和密码就能进入Shell,比如运行bash会开启一个新的会话窗口

2.2 环境变量

个人配置文件：~/.bash_profile ~/.bashrc
全局配置文件：/etc/profile /etc/profile.d/*.sh /etc/bashrc
profile类文件，设定环境变量，登陆前运行的脚本和命令。
bashrc类文件，设定本地变量，定义命令别名

2.3 环境变量生效顺序

```
登录式shell配置文件执行顺序:
1./etc/profile
2./etc/profile.d/*.sh
3.~/.bash_profile
4.~/.bashrc
4./etc/bashrc

非登陆式shell配置文件执行顺序:
1.~/.bashrc
2./etc/bashrc
3./etc/profile.d/*.sh
```

3.su命令使用案例

```
#root切换到普通用户
su - oldzhang

#普通用户切换到root
su -
su - root

#以某个用户身份运行命令
su - oldzhang -c 'whoami'
```

4.su命令总结

```
1.普通用户切换到root，可以使用su - 或 su - root，但必须输入root密码才能切换
2.root用户切换到普通用户，可以使用su - 用户名 不需要输入密码。
3.如果仅希望在某个用户下执行命令，而不是直接切换到该用户下操作，则可以使用su - 用户名 -c '命令'的方式。
```

第4章 普通用户权限提升

1.权限提升介绍

```
虽然我们可以使用su - root来切换成超级用户使用所有命令，但是这样做会有几个安全问题：
1.使用su - 切换用户，普通用户也就知道了root的密码，这显然是不太能接受的。
2.当普通用户切换到root用户之后，就可以使用所有的命令，甚至可以改掉root密码，这显然也是不可接受的。

那么我们可不可以实现既让普通用户可以一些特权执行命令又不需要告诉他root密码呢？
答案是可以的，那就是使用sudo命令来临时提升权限。
```

2.sudo命令介绍

```
通过sudo命令，我们可以把某些特权命令授权给指定的普通用户，并且普通用户不需要知道root密码就能使用。
但是具体哪些用户可以使用哪些特权命令就需要通过修改sudo的配置文件来实现了。
```

3.sudo配置使用

```
#我们可以通过修改sudo的配置文件来实现添加特权命令的配置。
[root@linux ~]# vim /etc/sudoers
[root@linux ~]# tail -1 /etc/sudoers
oldya  ALL=(ALL)       /usr/bin/yum,/usr/bin/rm

#切换到没有配置sudo的用户下并查看一下普通用户是否可以执行sudo命令
[oldboy@linux ~]$ sudo -l
[sudo] oldboy 的密码：
对不起，用户 oldboy 不能在 linux 上运行 sudo。

#直接使用rm命令测试
[oldboy@linux ~]$ rm -rf /opt/1.txt
rm: 无法删除"/opt/1.txt": 权限不够

#尝试使用sudo rm命令测试
[oldboy@linux ~]$ sudo rm -rf /opt/1.txt
[sudo] oldboy 的密码：
oldboy 不在 sudoers 文件中。此事将被报告。


#然后我们再切换到配置了sudo的用户下
[root@linux ~]# su - oldya
上一次登录：五 3月 26 17:50:05 CST 2021pts/0 上

#查看可以使用的sudo命令
[oldya@linux ~]$ sudo -l
......省略......
用户 oldya 可以在 linux 上运行以下命令：
    (ALL) /usr/bin/yum, /usr/bin/rm

#直接使用rm命令
[oldya@linux ~]$ rm -rf /opt/1.txt
rm: 无法删除"/opt/1.txt": 权限不够

#使用sudo rm命令
[oldya@linux ~]$ sudo rm -rf /opt/1.txt
```

4.visudo命令介绍

```
visudo是专门用来编辑sudo配置文件的命令，既然vim就可以编辑那为什么还需要visudo呢？
因为vim命令编辑的sudo文件并不具备语法检查功能，而使用visudo命令则可以实现对sudo配置文件的配置进行语法检查的功能。
也就是说使用visudo命令就可以直接配置sudo，并且如果配置错了还会有错误提醒。所以强烈建议使用visudo命令来配置sudo。
```

5.多用户多角色多权限命令配置

前因：

```
虽然我们通过配置sudo实现了普通用户使用特权命令，但是当我们需要配置的用户和特权命令越来越多时，一条条的添加就显得有些不灵活了。
针对这种情况，我们可以使用分类和分组的配置使的复杂情况的sudo配置变得清晰简单。
```

项目需求：

用户	角色	可以执行的命令
ops1 ops2	运维	网络命令: ifconfig|ping 软件命令: rpm|yum 服务命令:service|systemctl 进程命令:kill|killall
dev1 dev2	开发	网络命令: ifconfig|ping 服务命令:service|systemctl

配置步骤:

```
##1.首先创建4个用户并配置密码
useradd ops1
useradd ops2
useradd dev1
useradd dev2

echo "123"|passwd --stdin ops1
echo "123"|passwd --stdin ops2
echo "123"|passwd --stdin dev1
echo "123"|passwd --stdin dev2

##2.然后使用visudo命令来编辑sudo的配置文件
[root@linux ~]# visudo

##3.定义分组，注意这里的分组和Linux系统的分组没有关系，OPS为运维组 DEV为开发组
User_Alias OPS = ops1,ops2
User_Alias DEV = dev1,dev2

##4.定义可以执行命令的组，后面可以直接调用
Cmnd_Alias NETWORKING = /sbin/ifconfig, /bin/ping
Cmnd_Alias SOFTWARE = /bin/rpm, /usr/bin/yum
Cmnd_Alias SERVICES = /sbin/service, /usr/bin/systemctl start
Cmnd_Alias PROCESSES = /bin/kill, /usr/bin/kill, /usr/bin/killall

##5.给组分配权限
OPS  ALL=(ALL) NETWORKING,SOFTWARE,SERVICES,STORAGE,DELEGATING,PROCESSES
DEV  ALL=(ALL) SOFTWARE,PROCESSES

#6.切换到普通用户然后查看sudo权限
过程略....
```