20.权限管理进阶

前面我们讲过基础的权限命令，chmod，chown，chgrp，umask命令，通过这些命令我们能够对目录或文件进行最基本的权限设置，但是在实际的应用中还有很多是这些命令解决不了的权限设置问题，所以，我们这里进一步地对linux的权限管理进行讲解！

一、Linux权限管理的ACL权限

1、什么是 ACL 权限？

ACL的全称是 Access Control List（访问控制列表），一个针对文件/目录的访问控制列表。它在UGO权限管理的基础上为文件系统提供一个额外的、更灵活的权限管理机制。ACL允许你给任何的用户或用户组设置任何文件/目录的访问权限。

2、ACL有什么用

既然是作为UGO权限管理的补充，ACL自然要有UGO办不到或者很难办到的本事，例如：

- 可以针对用户来设置权限
- 可以针对用户组来设置权限
- 子文件/目录继承父目录的acl权限

3、检查是否支持ACL

ACL需要Linux内核和文件系统的配合才能工作，当前大多数Linux发行版本默认都是支持的。

CentOS7.X创建的xfs文件系统默认内置支持ACL功能！

CentOS7.X之前版本，手工创建的ext4文件系统可能没有开ACL功能。**如果没有需手动重新挂载开启，但一般情况都是开着的。**

CentOS7.X之前版本，最好还是查看一下，我们看某个文件/目录否支持 ACL 权限，首先要看文件所在的分区是否支持 ACL 权限。

①、查看当前系统有哪些分区：df -h

②、查看指定分区详细文件信息：dumpe2fs -h 分区路径 （只针对于CentOS7.X之前版本，xfs文件系统不支持这个命令了，xfs里用xfs_info 分区路径，查看分区的详细信息）

```
[root@CentOS6 ~]# dumpe2fs -h /dev/sda3
dumpe2fs 1.41.12 (17-May-2010)
Filesystem volume name:   <none>
Last mounted on:          /
Filesystem UUID:          8e411d32-58df-4366-bddf-02f
Filesystem magic number:  0xEF53
Filesystem revision #:    1 (dynamic)
Filesystem features:      has_journal ext_attr resize
large_file huge_file uninit_bg dir_nlink extra_isize
Filesystem flags:         signed_directory_hash
Default mount options:    user_xattr acl
```

有acl表示开启的，如果没有就要手工开一下：

①、临时开启分区 ACL 权限，注意xfs文件系统已经不支持acl挂载方式，不需要怎么挂载开启acl，因为它内置支持acl

mount -o remount,acl /

②、永久开启分区 ACL 权限，Centos7上部要改，上面的命令都报错，就不要画蛇添足了！

修改配置文件 /etc/fstab

```
UUID=490ed737-f8cf-46a6-ac4b-b7735b79fc63 /                       ext4    defaults,acl        1 1
UUID=b9a4dcfe-b646-483d-aebd-796828b6b00d /boot                   ext4    defaults            1 2
UUID=5a7b789e-c6f3-4bcf-b868-c03811cdb818 /home                   ext4    defaults            1 2
UUID=f871cc80-26a3-43ce-9859-60939d17bd44 swap                    swap    defaults            0 0
tmpfs                   /dev/shm                tmpfs   defaults        0 0
devpts                  /dev/pts                devpts  gid=5,mode=620  0 0
sysfs                   /sys                    sysfs   defaults        0 0
proc                    /proc                   proc    defaults        0 0
~
```

想让某个分区拥有acl权限，只需在defaults后面加上 ,acl
其实一般default权限是默认开启acl

重新挂载文件系统或重启系统，使得修改生效

mount -o remount /

4、检查系统安装acl没的，一般情况都是安装了的，以防万一看一下：

rpm -qa | grep acl

acl-2.2.51-14.el7.x86_64

libacl-2.2.51-14.el7.x86_64

#没看到就用yum安装一下这两个软件

5、设定 ACL 权限：setfacl 选项 文件名

选项：

-m ：配置后面的 acl 参数给文件/目录使用，不可与 -x 合用；

-x ：删除后续的 acl 参数，不可与 -m 合用；

-b ：移除所有的 ACL 配置参数；

-k ：移除默认的 ACL 参数；

-R ：递归配置 acl；

-d ：配置“默认 acl 参数”，只对目录有效，在该目录新建的数据会引用此默认值；

①、给用户设定 ACL 权限：setfacl -m u:用户名:权限 指定文件名

②、给用户组设定 ACL 权限:setfacl -m g:组名:权限 指定文件名

看实例：

```
# cd /
# mkdir a
# groupadd tg
# useradd user1
# useradd user2
# useradd user3
# gpasswd -a user1 tg
# gpasswd -a user2 tg
# chgrp tg a
# chmod 770 a
drwxrwx---    2 root tg      6 2月   17 11:37 a
```

单独位user3赋予acl权限rx：

```
# setfacl -m u:user3:rx /a
```

赋予acl权限后，在ll，就可以看到一个加号，这个加号就表示存在acl权限：

```
drwxrwx---+   2 root tg     17 2月   17 11:50 a
```

6、查看 ACL 权限：getfacl 文件名

acl权限具体是什么，用getfacl查看：

```
[root@CentOS7 /]# getfacl a
# file: a
# owner: root
# group: tg
user::rwx
user:user3:r-x
group::rwx
mask::rwx
other::---
```

7、最大有效权限 mask

ACL 权限存在一个最大有效权限的概念，我们给用户或用户组设定 ACL 权限其实并不是真正我们设定的权限，是与 mask 的权限“相与”之后的权限才是用户的真正权限，一般默认mask权限都是rwx，与我们所设定的权限相与就是我们设定的权限。

这个值是可以修改的：

setfacl -m m:权限 文件名

删除 ACL 权限

①、删除指定用户的 ACL 权限

setfacl -x u:用户名 文件名

②、删除指定用户组的 ACL 权限

setfacl -x g:组名 文件名

③、删除文件的所有 ACL 权限

setfacl -b 文件名

递归 ACL 权限

通过加上选项 -R 递归设定文件的 ACL 权限，所有的子目录和子文件也会拥有相同的 ACL

权限。

setfacl -m u:用户名:权限 -R 文件名

默认 ACL 权限

如果给父目录设定了默认的 ACL 权限，那么父目录中所有新建的子文件会继承父目录的 ACL 权限。

setfacl -m d:u:用户名:权限 文件名

二、sudo命令的使用

简单的说，sudo 是一种权限管理机制，管理员可以授权于一些普通用户去执行一些 root 才能执行的操作，而不需要知道 root 的密码。

严谨些说，sudo 允许一个已授权用户以超级用户或者其它用户的角色运行一个命令。

当然，能做什么不能做什么都是通过安全策略来指定的。

默认的安全策略记录在 /etc/sudoers 文件中。

1、编辑配置文件命令:visudo

注意：编辑sudo的配置文件/etc/sudoers是一般不要直接使用vi（vi /etc/sudoers）去编辑，因为sudoers配置有一定的语法，直接用vi编辑保存系统不会检查语法，如有错也保存了可能导致无法使用sudo工具，最好使用visudo命令去配置。虽然visudo也是调用vi去编辑，但是保存时会进行语法检查，有错会有提示。

2、配置文件/etc/sudoers的具体配置

root ALL=(ALL) ALL

#用户名 被管理主机的地址=（可使用的身份） 授权命令（绝对路径）**（注意：这里写的越具体，授权就越安全）**

%wheel ALL=(ALL) ALL

#%组名 被管理主机的地址=（可使用的身份） 授权命令（绝对路径）**（注意：这里写的越具体，授权就越安全）**

可使用的身份：（ALL）代表可以切换成任意身份。不写默认是root

举例，授权给user1用户可以添加新账号，和修改普通用户的密码：

user1 ALL=/usr/sbin/useradd 赋予 user1 添加用户权限.命令必须写入绝对路径

user1 ALL=/usr/bin/passwd 赋予改密码权限，只是这样做要遭！ 必须取消对 root 的密码修改

user1 ALL=/usr/bin/passwd [A-Za-z]*, !/usr/bin/passwd ""，!/usr/bin/passwd root #注意：逗号后必须有空格

3、普通用户查看自己能执行的sudo命令：sudo -l

用户 user1 可以在 CentOS7 上运行以下命令：

(root)/usr/bin/passwd [A-Za-z]*, !/usr/bin/passwd ""，

!/usr/bin/passwd root

4、普通用户执行自己能执行的sudo命令：sudo /usr/bin/passwd user2

三、特殊权限SUID、SGID、Sticky（了解，学习这个可以让我们更深刻地理解linux的运行原理，工作中不建议使用）

1、为什么要使用特殊权限?

比如系统中假如有超过四类人然而每一类人都需要一种独特权限.只有三种独特权限的基础权限系统就会明显不够用.

特殊权限可以扩展系统基础权限的功能,使得linux权限更加强大灵活.

在理解特殊权限之前,需要先具备几个有关的认知:

前提：进程有属主和属组；文件有属主和属组；

(1) 任何一个可执行程序文件能不能启动为进程：取决于发起者对程序文件是否拥有执行权限；

(2) 启动为进程之后，其进程的属主为发起者；进程的属组为发起者所属的组；

(3) 进程访问文件时的权限，取决于进程的发起者：

(a) 进程的发起者，同文件的属主：则应用文件属主权限；

(b) 进程的发起者，属于文件的属组；则应用文件属组权限；

(c) 其他，应用文件“其它”权限；

看一个现象passwd修改密码这个命令：

```
$ ll /bin/passwd
-rwsr-xr-x. 1 root root 27832 6月  10 2014 /bin/passwd
$ ll /etc/shadow
---------- 1 root root 748 2月  17 14:57 /etc/shadow
```

密码保存在 /etc/shadow文件里，这个文件只能root才能修改，

问：在上面的理论前提下，普通用户为什么可以用passwd命令修改密码？

2、SUID

①权限设定方法：

字母表示法:

chmod u+s FILE...

chmod u-s FILE...

数字表示法:

chmod 4755 FILE 添加SUID权限到二进制程序文件(添加到DIR无意义)

在普通三位数字权限位之前,用4代表添加的SUID位

chmod 755 .... 可以删除文件的SUID

②文件SUID 权限表示:

```
$ ll /bin/passwd
-rwsr-xr-x. 1 root root 27832 6月  10 2014 /bin/passwd
```

文件属主的x权限,用s代替.表示被设置了SUID

如果属主位没有x权限,会显示为大写S,表示有故障(权限无效)

③SUID相关说明:

1. 启动为进程之后，其进程的属主为发起者;
2. 只能作用在二进制程序上,能作用在脚本上,且设置在目录上无意义 ;
3. 执行suid权限的程序时,此用户将继承此程序的所有者权限; 这就是普通用户为什么能修改密码的原因!

④SUID工作原理

环境前提:

a. linux中有一个二进制程序cat,属主属组均为root
b. linux中有一个系统文件/etc/shadow,属主属组均为root
c. 我们创建一个普通用户叫bz
d. bz具有对cat的执行权限
e. bz不具有对/etc/shadow的任何权限

默认情况下

a. bz执行cat,系统创建一个cat进程,进程的属主属组取程序发起者,也就是bz
b. cat进程访问/etc/shadow,由于进程属主属组是bz,
与/etc/shadow的属组属主都不匹配,所以被拒绝访问.

给cat设置SUID之后

a. bz执行cat.系统创建一个cat进程,进程的属主取cat的属主,属组取程序发起者,就是root
b. cat进程访问/etc/shadow,由于进程属主是root,与/etc/shadow

的属主匹配,所以被允许访问.

把④的过程实验一遍，注意再次强调，特殊权限很肯能造成安全隐患，工作中不建议使用！

3、SGID

权限设定方法：

字母表示法

chmod g+s DIR/FILE...

chmod g-s DIR/FILE...

数字表示法 :

chmod 2755 DIR/FILE 添加SGID到目录或文件

在普通数字权限位前,用2代表添加SGID位

chmod 755 DIR/FILE 删除SGID

文件权限表示:

ll locate

-rwx--s--x 1 root slocate 40520 4月 11 2018 locate

文件属组的x权限,用s代替.表示被设置了SGID

如果属组位没有x权限,会显示为大写S,表示有故障(权限无效)

SGID相关说明:

作用在二进制程序上时:

执行sgid权限的程序时,此用户将继承此程序的所属组权限（分析一下locate命令）

作用于目录上时:

此文件夹下所有用户新建文件都自动继承此目录的用户组.

SGID实例:

作用在二进制程序上时,原理同SUID,只是由user位变为group位.

作用于目录上时演示:

帐户root在/tmp中创建一个目录叫testdir,添加SGID.作用在目录上必须设定权限777

切换到普通用户user1,在testdir目录中创建一个文件和一个目录

结果显示,user1在testdir目录下创建的文件和目录都自动继承了testdir的属组

而且新目录的权限也继承了SGID.

所以设定了SGID的目录中的所有新建文件和目录都会自动属于root组.

所以这个SGID没什么意义，了解一下就ok！

4、Sticky粘滞位

权限设定方法：

字母表示法:

```
chmod o+t DIR...
chmod o-t DIR...
```

数字表示法:

```
chmod 1755 DIR
```

在普通数字权限位前,**用1代表添加Sticky位**

文件权限表示:

```
drwxrwxrwt.  27 root root 4096 2月  17 15:52 tmp
```

文件other位的x权限,用t代替.表示被设置了Sticky

如果other位没有x权限,会显示为大写T,表示有故障(权限无效)

Sticky相关说明:

1. **对于一个多人可写的目录，如果设置了sticky，则每个用户仅能删除和修改自己的文件或目录；**
2. 只能作用在目录上.普通文件设置无意义,且会被linux内核忽略
3. 用户在设置Sticky权限的目录下新建的目录不会自动继承Sticky权限

Sticky实例:

目的:

系统中创建一个很多用户可以共同使用的目录,但是要求用户之间不能互相删除改变对方的文件.

实现:

root用户先创建一个777权限目录/tmp/testdir

给/tmp/testdir这个文件夹设定一个Sticky位.

```
# chmod 1777 /tmp/testdir
# ll -d /tmp/testdir
drwxrwxrwt. 5 root root 97 Nov 20 00:49 /tmp/testdir
```

设定完毕可见目录tmp的other权限中的x位已经显示为t.说明已设定成功.

普通用户在设定了Sticky位的目录下创建的子目录不会继承这个Sticky权限,所以要注意设定好自己目录的权限.

四、文件隐藏属性：chattr，lsattr

Linux文件的隐藏属性在保护系统文件的安全性上非常重要。**它妖怪的地方在于，这个权限是可以限制root的，也就是说加了这个属性，不能删除文件，即使是root都删除不了！当然root是可以去除掉这个属性后在删除的！**

chattr（设置文件的隐藏属性）

```
# chattr [+-=] [ASacdistu] File/Directory
```

参数

+：增加某个参数

-：删除某个参数

=：仅有后面接的参数

最常用的的参数：

a：如果对文件设置 a 属性，那么只能在文件中增加数据，但是不能删除也不能修改数据；

如果对目录设置 a 属性，那么只允许在目录中建立和修改文件，但是不允许删除。

i： 如果对文件设置 i 属性，那么不允许对文件进行删除、改名，也不能添加和修改数据；

如果对目录设置 i 属性，那么只能修改目录下文件的数据，但不允许建立和删除文件。

lsattr（显示文件隐藏属性）

```
# lsattr 选项  文件名
```

选项：

-a 显示所有文件和目录

-d 若目标是目录，仅列出目录本身的属性，而不是子文件的