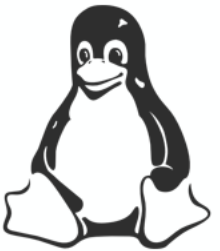

Linux系统管理

南通师范高等专科学校 朱亚林

本章导言

- Linux操作系统以高安全性著称。其中有一个显著特点就是：Linux操作系统通过对用户的分类很好地限制了“root”权限，对各类文件加上权限很好地保护了文件的安全。从而让别人不可能随意动得了你的“奶酪”。Linux操作系统的众多用户犹如微信中的用户，每位用户都可以根据实际情况为自己定制一个开放共享的准则，或是朋友圈或是工作群，而这个准则就是权限。
- 本章内容，就将详细介绍Linux操作系统中用户的管理和文件权限的管理。

第5章

朋友圈：用户、群组和权限

5.3.4 文件的特殊权限

- 一般的文件权限可以满足文件操作的一般需要，但也会有一些特殊场景出现。
- 例如：

- 一个普通用户怎样去修改属于系统管理员的配置文件？
- 在服务器上建立一个共享目录，任何人都可以向这个目录中写入文件，但每个人只能编辑和删除自己建立的文件？

这些场景应用以上的权限管理方式很难实现，此时就要考虑使用文件的特殊权限了。

SUID

- SUID是一种特殊权限，它允许普通用户使用root用户拥有的可执行程序（不包含脚本），并且使用这些程序操作那些归属于root用户、普通用户没有权限访问的文件。
- 例如，我们知道/bin/passwd是用来设置系统内用户密码的工具，它的权限信息如下图所示。

```
zz@zz-lab:~$ ls -l /bin/passwd
-rwsr-xr-x 1 root root 63960 Feb  7  2020 /bin/passwd
```

- /etc/shadow文件是用来保存用户密码信息的，它的权限信息如下图所示。

```
zz@zz-lab:~$ ls -l /etc/shadow
-rw-r----- 1 root shadow 1169 Mar 30 15:30 /etc/shadow
```

SUID

- SUID权限会允许所有的用户都可以调用passwd程序（当然它的归属权是属于root用户、root群组的），当普通用户执行passwd时，就会在执行过程中拥有和root用户一样的权限，那么对应到/etc/shadow，普通用户就拥有了和root用户一样的对它进行读和写的权限了。

```
zz@zz-lab:~$ ls -l /bin/passwd
-rwsr-xr-x 1 root root 63960 Feb  7  2020 /bin/passwd
```

SUID

SUID权限赋予的方式有两种，一种是使用字符，另一种是使用数字。

- 第一种通过字符的方式，结合之前的学习可以理解，将用户加上s，就表示拥有SUID权限了。
- 第二种通过数值的方式，可以看到是在原先三个数值的基础上，在最前面再加上一个数值。SUID、权限分别对应着数值4，如果拥有SUID，那么就在原来的三位数字表示权限的值的最左边加上4。

SUID

- 案例：为/bin/find命令设置SUID权限，然后再恢复初始权限

步骤：

（1）查询/bin/find的初始权限

（2）设置find为SUID权限

（3）查询/bin/find设置后的权限

（4）恢复find的初始权限

```
zz@zz-lab:~$ ls -l /bin/find
-rwxr-xr-x 1 root root 311008 Jan 10  2021 /bin/find
zz@zz-lab:~$ sudo chmod u+s /bin/find
[sudo] password for zz:
zz@zz-lab:~$ ls -l /bin/find
-rwsr-xr-x 1 root root 311008 Jan 10  2021 /bin/find
zz@zz-lab:~$ sudo chmod u-s /bin/find
zz@zz-lab:~$ ls -l /bin/find
-rwxr-xr-x 1 root root 311008 Jan 10  2021 /bin/find
```

SUID

SUID的特点

- SUID只针对可执行文件有效，普通文件赋予 SUID 没有意义。
- 用户需要对此可执行文件有 x 权限；
- SUID 权限赋予用户变身为文件所有者的效果，只在可执行文件运行过程中有效。

SGID

- SGID是对于群组的特殊权限，它不仅可以应用于文件，也可以应用于目录。
- 对于文件来说，SGID 表现出来的特征与SUID极其相似：拥有上述SUID的所有特点，它们的不同之处就在于，SUID 赋予用户的是文件所有者的权限，而 SGID 赋予用户的是文件所属组的权限。

SGID

- 还记得曾经使用过locate命令吗？这是一个典型的应用SGID权限的命令！
- 为什么呢？

SGID

- 对于目录而言，当一个目录被赋予 SGID 权限后，进入此目录的普通用户，其群组身份会变为该目录的所属组，这就使得用户在该目录下创建文件（或目录）时，其所属组将不再是用户的所属组，而是目录的所属组。
- 当然也要注意，只有当普通用户对具有 SGID 权限的目录具有 rwx 权限时，SGID 的功能才能完全发挥。比如说，如果用户对该目录仅有 rx 权限，则用户进入此目录后，虽然其有效群组变为此目录的所属组，但由于没有 w 权限，用户无法在目录中创建文件或目录，SGID 权限也就无法发挥它的作用。

SGID

SGID权限赋予的方式有两种，一种是使用字符，另一种是使用数字。

- 第一种通过字符的方式，结合之前的学习可以理解，将用户组加上s，就表示拥有SGID权限了。
- 第二种通过数值的方式，可以看到是在原先三个数值的基础上，在最前面再加上一个数值。SGID、权限分别对应着数值2，如果拥有SGID，那么就在原来的三位数字表示权限的值的最左边加上2。当然，如果需要给目标对象同时加上SUID权限的话，就在最左边加上数字6，6即是4+2的结果。

SGID

- 案例：SGID权限的赋予与测试

步骤：

（1） 使用zz用户，在LinuxStudy/chapter-5创建一个文件Share-4，并赋予其777的权限。

（2）给Share-4赋予SGID权限

（3）查看Share-4目录的权限。

（4）使用newbie用户登录，进入/home/zz/LinuxStudy/chapter-5/Share-4目录。

（5）使用newbie建立目录Test，并查看其权限。

```
zz@zz-lab:~/LinuxStudy/chapter-5$ mkdir Share-4
zz@zz-lab:~/LinuxStudy/chapter-5$ sudo chmod 777 -R Share-4/
zz@zz-lab:~/LinuxStudy/chapter-5$ ls -l
total 4
drwxrwxrwx 2 zz zz 4096 Apr  5 14:36 Share-4
zz@zz-lab:~/LinuxStudy/chapter-5$ sudo chmod 2777 -R Share-4/
zz@zz-lab:~/LinuxStudy/chapter-5$ ls -l
total 4
drwxrwsrwx 2 zz zz 4096 Apr  5 14:36 Share-4
```

```
zz-lab% cd /home/zz/LinuxStudy/chapter-5/Share-4
zz-lab% mkdir Test
zz-lab% ls -l
total 4
drwxr-sr-x 2 newbie zz 4096 Apr  5 14:37 Test
```

SBIT

- SBIT权限仅对目录有效，如果目录被设定了SBIT权限，那么用户在此目录下创建的文件或目录，就只有自己和文件夹属主、 root 才可以修改或删除。

SBIT

- SBIT权限的表示方法是在others权限（第三组权限）的x位上用t来替代。被设置SBIT权限的目录，其基本权限也必须是777。
- SBIT可以使用o+t这种字母组合设置；
- SBIT也可以在基本数字权限之前加上数值1，即“1777”来表示；如果与SUID、SGID进行组合，那么就是启用的权限所对应的数值相加，放在基本数字权限的最左侧。

SBIT

- 案例：SBIT权限的赋予与测试

步骤：

（1）使用zz账户在主目录LinuxStudy/chapter-5创建目录Share-5，并赋予其777权限。

（2）给Share-5赋予SBIT权限。

（3）查看Share-5目录的权限。

（4）使用newbie用户登录，进入Share-5目录。

（5）使用newbie建立文件test.txt，并查看其权限。

（6）使用newuser用户登录，进入Share-5目录，尝试删除test.txt。

（7）再次使用newbie登录，尝试删除test.txt。

```
zz@zz-lab:~/LinuxStudy/chapter-5$ mkdir Share-5
zz@zz-lab:~/LinuxStudy/chapter-5$ chmod -R 1777 Share-5
zz@zz-lab:~/LinuxStudy/chapter-5$ ls -l
total 8
drwxrwsrwx 3 zz zz 4096 Apr  5 14:37 Share-4
drwxrwxrwt 2 zz zz 4096 Apr  5 14:46 Share-5
zz@zz-lab:~/LinuxStudy/chapter-5$
```

```
zz-lab% pwd
/home/zz/LinuxStudy/chapter-5
zz-lab% ls -l
total 8
drwxrwsrwx 3 zz zz 4096 Apr  5 14:37 Share-4
drwxrwxrwt 2 zz zz 4096 Apr  5 14:46 Share-5
zz-lab% cd Share-5
zz-lab% touch test.txt
zz-lab% ls -l
total 0
-rw-r--r-- 1 newbie newbie 0 Apr  5 14:48 test.txt
zz-lab% rm test.txt
```

```
zz-lab% whoami
newuser
zz-lab% pwd
/home/zz/LinuxStudy/chapter-5/Share-5
zz-lab% ls -l
total 0
-rw-r--r-- 1 newbie newbie 0 Apr  5 14:48 test.txt
zz-lab% rm test.txt
rm: remove write-protected regular empty file 'test.txt'? y
rm: cannot remove 'test.txt': Operation not permitted
zz-lab%
```

5.3.5 su和sudo

- Linux正常使用时不建议使用root用户。一般都是使用普通用户登录，在需要提升权限时，通过su或者sudo的命令来解决。

1. su命令

- su命令是最常用的用户身份切换的命令。它可以从普通用户切换到包括root用户在内的任何用户，也可以从root用户切换到任何普通用户。
- 在进行切换时，如果是以普通用户切换，需要知道切换对象的密码；如果是以root用户切换，则无需输入密码。

```
su [选项] [用户名]
```

1. su命令

- su命令常用的选项如下表

选项	作用
-	当前用户不仅切换为指定用户的身份，同时所用的工作环境也切换为此用户的环境（包括 PATH 变量、MAIL 变量等），使用-选项可省略用户名，默认会切换为 root 用户。
-l	同-的使用类似，也就是在切换用户身份的同时，完整切换工作环境，但后面需要添加欲切换的使用者账号。
-p	表示切换为指定用户的身份，但不改变当前的工作环境（不使用切换用户的配置文件）。
-c	仅切换用户执行一次命令，执行后自动切换回来，该选项后通常会带有要执行的命令。

1. su命令

- 案例：使用su切换用户身份

（1）将当前用户切换为root用户

（2）将当前用户及工作环境切换为newbie用户

（3）使用root账户身份执行一次系统更新

2. sudo命令

- sudo的全称为：super user do，也就是超级用户可以做的事。它是允许普通用户执行一些或者全部的root命令的一个工具。

选项	作用
-l	查询用户执行sudo的权限
-k	将会强制用户在下一次执行sudo时输入密码（默认5分钟内不需要重新输入密码）
-b	将要执行的指令放在背景执行
-u	以指定用户的身份执行相关命令

3. su和sudo的比较

- su和sudo都可以用来提权，但两者又有着很大的不同。su命令的强大之处在于，调用su可以切换到任何的人身份，不足之处在于必须掌握目标用户的密码，也就是要征得调用者的授权，对于root用户而言，将密码告知他人会导致服务器存在极大的安全隐患。sudo则不然，它可以让用户在根据实际需要配置好的权限之内执行相关命令，而且只需要输入自己的密码进行确认即可，安全性大大提升。