linux su和sudo命令的区别

作者：佚名 字体：[增加 减小] 来源：互联网 时间：04-13 11:02:50 我要评论

> 一. 使用 su 命令临时切换用户身份 1、su 的适用条件和威力　　su命令就是切换用户的工具，怎么理解呢？比如我们以普通用户beinan登录的，但要添加用户任务，执行useradd ，beinan用户没有这个权限，而这个权限恰恰由root所拥有。解决办法无法有两个，一是退出beinan用户

一. 使用 su 命令临时切换用户身份

1、su 的适用条件和威力

su命令就是切换用户的工具，怎么理解呢？比如我们以普通用户beinan登录的，但要添加用户任务，执行useradd ，beinan用户没有这个权限，而这个权限恰恰由root所拥有。解决办法无法有两个，一是退出beinan用户，重新以root用户登录，但这种办法并不是最好的；二是我们没有必要退出beinan用户，可以用su来切换到root下进行添加用户的工作，等任务完成后再退出root。我们可以看到当然通过su 切换是一种比较好的办法；

通过su可以在用户之间切换，如果超级权限用户root向普通或虚拟用户切换不需要密码，什么是权力？这就是！而普通用户切换到其它任何用户都需要密码验证；

2、su 的用法：

su [OPTION选项参数] [用户]

-, -l, ——login 登录并改变到所切换的用户环境；

-c, ——commmand=COMMAND 执行一个命令，然后退出所切换到的用户环境；

至于更详细的，请参看man su ；

3、su 的范例：

su 在不加任何参数，默认为切换到root用户，但没有转到root用户家目录下，也就是说这时虽然是切换为root用户了，但并没有改变root登录环境；用户默认的登录环境，可以在/etc/passwd 中查得到，包括家目录，SHELL定义等；

[beinan@localhost ~]?$ suPassword:[root@localhost beinan]# pwd/home/beinan

su 加参数 - ，表示默认切换到root用户，并且改变到root用户的环境；

[beinan@localhost ~]?$ pwd/home/beinan[beinan@localhost ~]?$ su -Password:[root@localhost ~]# pwd/root

su 参数 - 用户名

[beinan@localhost ~]?$ su - root 注：这个和su - 是一样的功能；

Password:

[root@localhost ~]# pwd

/root

[beinan@localhost ~]?$ su - linuxsir 注：这是切换到 linuxsir用户

Password: 注：在这里输入密码；

[linuxsir@localhost ~]?$ pwd 注：查看用户当前所处的位置；

/home/linuxsir

[linuxsir@localhost ~]?$ id 注：查看用户的UID和GID信息，主要是看是否切换过来了；

uid=505(linuxsir) gid=502(linuxsir) groups=0(root),500(beinan),502(linuxsir)

[linuxsir@localhost ~]?$

[beinan@localhost ~]?$ su - -c ls 注：这是su的参数组合，表示切换到root用户，并且改变到root环境，然后列出root家目录的文件，然后退出root用户；

Password: 注：在这里输入root的密码；

anaconda-ks.cfg Desktop install.log install.log.syslog testgroup testgroupbeinan testgrouproot

[beinan@localhost ~]?$ pwd 注：查看当前用户所处的位置；

/home/beinan

[beinan@localhost ~]?$ id 注：查看当前用户信息；

uid=500(beinan) gid=500(beinan) groups=500(beinan)

4、su的优缺点；

su 的确为管理带来方便，通过切换到root下，能完成所有系统管理工具，只要把root的密码交给任何一个普通用户，他都能切换到root来完成所有的系统管理工作；但通过su切换到root后，也有不安全因素；比如系统有10个用户，而且都参与管理。如果这10个用户都涉及到超级权限的运用，做为管理员如果想让其它用户通过su来切换到超级权限的root，必须把root权限密码都告诉这10个用户；如果这10个用户都有root权限，通过root权限可以做任何事，这在一定程度上就对系统的安全造成了威协；想想Windows吧，简直就是恶梦；“没有不安全的系统，只有不安全的人”，我们绝对不能保证这 10个用户都能按正常操作流程来管理系统，其中任何一人对系统操作的重大失误，都可能导致系统崩溃或数据损失；所以su 工具在多人参与的系统管理中，并不是最好的选择，su只适用于一两个人参与管理的系统，毕竟su并不能让普通用户受限的使用；超级用户root密码应该掌握在少数用户手中，这绝对是真理！所以集权而治的存在还是有一定道理的；

二. sudo 授权许可使用的su，也是受限制的su

1. sudo 的适用条件

由于su 对切换到超级权限用户root后，权限的无限制性，所以su并不能担任多个管理员所管理的系统。如果用su 来切换到超级用户来管理系统，也不能明确哪些工作是由哪个管理员进行的操作。特别是对于服务器的管理有多人参与管理时，最好是针对每个管理员的技术特长和管理范围，并且有针对性的下放给权限，并且约定其使用哪些工具来完成与其相关的工作，这时我们就有必要用到 sudo。

通过sudo，我们能把某些超级权限有针对性的下放，并且不需要普通用户知道root密码，所以sudo 相对于权限无限制性的su来说，还是比较安全的，所以sudo 也能被称为受限制的su ； 另外sudo 是需要授权许可的，所以也被称为授权许可的su；

sudo 执行命令的流程是当前用户切换到root（或其它指定切换到的用户），然后以root（或其它指定的切换到的用户）身份执行命令，执行完成后，直接退回到当前用户；而这些的前提是要通过sudo的配置文件/etc/sudoers来进行授权；

比如我们想用beinan普通用户通过more /etc/shadow文件的内容时，可能会出现下面的情况；

[beinan@localhost ~]?$ more /etc/shadow/etc/shadow: 权限不够

这时我们可以用sudo more /etc/shadow 来读取文件的内容；就就需要在/etc/soduers中给beinan授权

于是我们就可以先su 到root用户下通过visudo 来改/etc/sudoers ；（比如我们是以beinan用户登录系统的）

[beinan@localhost ~]?$ su

Password: 注：在这里输入root密码

下面运行visodu；

[root@localhost beinan]# visudo 注：运行visudo 来改 /etc/sudoers

加入如下一行，退出保存；退出保存，在这里要会用vi，visudo也是用的vi编辑器；至于vi的用法不多说了；beinan ALL=/bin/more 表示beinan可以切换到root下执行more 来查看文件；

退回到beinan用户下，用exit命令；

[root@localhost beinan]# exit

exit

[beinan@localhost ~]?$

查看beinan的通过sudo能执行哪些命令?

[beinan@localhost ~]?$ sudo -l

Password: 注：在这里输入beinan用户的密码

User beinan may run the following commands on this host: 注：在这里清晰的说明在本台主机上，beinan用户可以以root权限运行more ；在root权限下的more ，可以查看任何文本文件的内容的；

(root) /bin/more

最后，我们看看是不是beinan用户有能力看到/etc/shadow文件的内容；

[beinan@localhost ~]?$ sudo more /etc/shadow

beinan 不但能看到 /etc/shadow文件的内容，还能看到只有root权限下才能看到的其它文件的内容，比如；

[beinan@localhost ~]?$ sudo more /etc/gshadow

对于beinan用户查看和读取所有系统文件中，我只想把/etc/shadow 的内容可以让他查看；可以加入下面的一行；

beinan ALL=/bin/more /etc/shadow

题外话：有的弟兄会说，我通过su 切换到root用户就能看到所有想看的内容了，哈哈，对啊。但咱们现在不是在讲述sudo的用法吗？如果主机上有多个用户并且不知道root用户的密码，但又想查看某些他们看不到的文件，这时就需要管理员授权了；这就是sudo的好处；

实例五：练习用户组在/etc/sudoers中写法；

如果用户组出现在/etc/sudoers 中，前面要加%号，比如%beinan ，中间不能有空格；%beinan ALL=/usr/sbin/*,/sbin/*

如果我们在 /etc/sudoers 中加上如上一行，表示beinan用户组下的所有成员，在所有可能的出现的主机名下，都能切换到root用户下运行 /usr/sbin和/sbin目录下的所有命令；

实例六：练习取消某类程序的执行：

取消程序某类程序的执行，要在命令动作前面加上!号； 在本例中也出现了通配符的*的用法；

beinan ALL=/usr/sbin/*,/sbin/*,!/usr/sbin/fdisk 注：把这行规则加入到/etc/sudoers中；但您得有beinan这个用户组，并且beinan也是这个组中的才行；

本规则表示beinan用户在所有可能存在的主机名的主机上运行/usr/sbin和/sbin下所有的程序，但fdisk 程序除外；

[beinan@localhost ~]?$ sudo -l

Password: 注：在这里输入beinan用户的密码；

User beinan may run the following commands on this host:(root) /usr/sbin/*(root) /sbin/*(root) !/sbin/fdisk[beinan@localhost ~]?$ sudo /sbin/fdisk -lSorry, user beinan is not allowed to execute '/sbin/fdisk -l' as root on localhost.

注：不能切换到root用户下运行fdisk 程序；

如果有sudo 的权限而没有su的权限: sudo su;