sudo命令详解

作者：斯巴达克斯 | 时间：January 4, 2014 | 分类：Linux

1.sudo是什么？

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执行的操作，而不需要知道root的密码，它依赖于/etc/sudoers这个文件，可以授权于那个用户在那个主机上能够以管理员的身份执行什么样的管理命令，而且是有限的。这个文件相当于就是一个授权表。

2./etc/sudoers 文件的语法

可以使用 man sudoers 来查看其帮助信息

由于这个文件是一个授权文件，那么其权限必定是很严格

```
[root@Linux178 ~]# ll /etc/sudoers
-r--r----- 1 root root 3381 Feb 23  2012 /etc/sudoers
[root@Linux178 ~]#

[root@Linux178 ~]# lsattr /etc/sudoers
------------- /etc/sudoers
[root@Linux178 ~]#
```

看到这个文件的权限是root和root组 只有读的权限，那也就是，编辑这个文件是有单独的命令的 visudo（这个文件我们最好不要使用vim命令来打开），是因为一旦你的语法写错会造成严重的后果，这个工具会替你检查你写的语法,这个文件的语法遵循以下格式：

```
who where whom command
```

说白了就是 那个用户在哪个主机以谁的身份执行那些命令，那么这个where,是指允许在那台主机ssh连接进来才能执行后面的命令，文件里面默认给root用户定义了一条规则， 看例子：

```
root    ALL=(ALL)       ALL
```

root root用户
ALL 所有的主机上都可以
(ALL) 是以谁的身份来执行，ALL就代表root可以任何人的身份来执行命令
ALL 所有的命令

那么整个一条规则就是root用户可以在任何主机以任何人的身份来执行所有的命令，也就是不限定。

再来看一条里面的规则：

```
%users  ALL=/sbin/mount /mnt/cdrom, /sbin/umount /mnt/cdrom
jerry   192.168.100.0/24=(root) /usr/sbin/useradd
```

%users 就是代表users这个组里面的所有成员
ALL 代表可以这所有的主机上
= 后面没有括号，也就是代表默认是以root身份
/sbin/mount /mnt/cdrom, /sbin/umount /mnt/cdrom 可以执行挂载的命令

3.查看用户可以执行的命令

已经授权的普通用户可以使用

```
sudo -l
```

来查看自己可以执行那些命令

要执行命令要在执行命令之前加上 sudo ，然后输入用户自己的密码，这是因为要验证，执行命令的用户确实是该用户。

sudo命令还有这个机制，就是在你正确输入密码并成功执行命令的5分钟内，再执行命令是不需要输入密码的，过了5分钟，就需要再次验证该用户的自己的密码，当然也可以手动让该期限过期，看下面sudo命令的语法

4.sudo命令语法

```
sudo [-bhHpV][-s ][-u <用户>][指令]
或
sudo [-klv]
参数
  -b  在后台执行指令。
  -h  显示帮助。
  -H  将HOME环境变量设为新身份的HOME环境变量。
  -k  结束密码的有效期限，也就是下次再执行sudo时便需要输入密码。
  -l  列出目前用户可执行与无法执行的指令。
  -p  改变询问密码的提示符号。
  -s  执行指定的shell。
  -u <用户>  以指定的用户作为新的身份。若不加上此参数，则预设以root作为新的身份。
  -v  延长密码有效期限5分钟。
  -V  显示版本信息。
  -S   从标准输入流替代终端来获取密码
```

5.场景

思考这么一个场景，看下面的规则

```
jerry   192.168.100.0/24=(root) /usr/sbin/useradd
```

这里面我如果想很多台主机上登录并执行命令（但是并不是所有的主机上），那这里岂不是要写很多的主机在这里吗？ 你执行useradd命令，但是这只是添加，不能为用户指定密码，那岂不是也不行？如果还有执行很多的命令，那是不是这里又要写很多的命令？

答案是否定的

sudo 是支持 主机别名、用户别名、whom别名（就是以谁的身份）、命令别名

有了别名，规则就变得很清爽，就是把同类的对象放到一个组里面，组名必须全部大写

主机别名 通过 Host_Alias 关键字来定义 例如下面的：

```
# Host_Alias     FILESERVERS = 192.168.100.0/24, 127.0.0.1
```

用户别名 通过 User_Aliases 关键字来定义，例如下面的：

```
# User_Alias ADMINS = jsmith, mikem
```

whom别名（就是以谁的身份） 通过 RunAs_Aliases 关键字来定义

```
这个通常是root或者是ALL 就不用定义了。
```

命令别名 通过 Cmnd_Alias 关键字来定义，例如下面的：

```
#Cmnd_Alias SERVICES = /sbin/service, /sbin/chkconfig
```

6.限定执行的命令

假设你允许一个普通用户执行passwd，那这就危险了，那么它是可以直接修改root的密码，这就有背于我们的初衷了，这就需要做限制了，如下设置：

```
pete    127.0.0.1 = /usr/bin/passwd [A-Za-z]*, !/usr/bin/passwd root
```

[A-Za-z]* 是一个正则表达式，代表是大小写字母组成的用户名

这一条就是限制pete这个用户，只能修改以字母组成的用户，而不能修改root的密码，！就是代表不能执行此命令。

在某个命令之前加!,就代表该用户或组不能执行该命令

7.设定那些命令执行的时候不需要输入密码

当然也是可以设置在执行某些命令的时候不用输入密码,例如：

```
fred          ALL = (DB) NOPASSWD: ALL
```

这就代表这个fred用户执行所有的命令时不需要输入密码

再如：

```
tom ALL = (root) PASSWD:/usr/sbin/useradd,/usr/sbin/usermod NOPASSWD:/usr/sbin/gourpadd
```

这一条就是代表tom这个用户在执行uesradd和usermod的时候是需要输入密码，而执行gourpadd时不需要

凡是 PASSWD 后面跟的命令都需要输入密码，而NOPASSWD后面的命令都不需要输入密码，前提是sudo记住密码的期限已过。

标签：linux, sudo

添加新评论