

Linux 命令 su 和 sudo 的区别

Python开发者 2023-08-05 17:10 发表于浙江

作者: wljslmz

在 Linux 系统中，有两个常用的命令用于切换用户身份和执行特权操作，它们分别是 **su** 和 **sudo**。虽然它们都可以实现权限提升，但在使用方式、安全性和适用场景等方面存在一些区别。本文将介绍 **su** 和 **sudo** 命令的区别，以帮助用户正确选择和使用这两个命令。



1. su 命令

su（切换用户）命令是在 Linux 系统中切换到其他用户身份的命令。通常，它需要超级用户（root）的密码才能切换到其他用户。使用 **su** 命令时，用户可以切换到其他用户的账户，并获得该账户的权限。例如，可以使用以下命令切换到超级用户（root）：

```
su
```

在输入密码后，用户将切换到 root 用户，并获得 root 用户的权限。

2. sudo 命令

sudo（以超级用户身份执行命令）命令是在 Linux 系统中以超级用户（root）的权限执行特定命令的命令。与 **su** 命令不同，**sudo** 命令允许普通用户以其自己的密码执行特权操作，而无需知道超级用户的密码。使用 **sudo** 命令时，用户需要在命令前加上 **sudo** 关键字，并输入自己的密码以确认身份。

例如，以下命令将以超级用户的权限执行 **apt-get update** 命令：

```
sudo apt-get update
```

在输入密码后，用户将获得执行特权操作的权限。

3. 区别对比

3.1 用户身份

su 命令需要超级用户（root）密码，用户可以切换到其他用户的身份并获得其权限。而 **sudo** 命令允许普通用户以其自己的密码执行特权操作。

3.2 权限范围

su 命令切换到其他用户后，用户将获得该用户的全部权限。而 **sudo** 命令可以通过配置文件（sudoers 文件）控制特权操作的范围，可以精确指定用户可以执行哪些命令以及以哪些用户的身份执行。

3.3 安全性

由于 `su` 命令需要共享 root 密码，这可能存在一些安全风险。如果其他人知道 root 密码，他们就可以切换到 root 用户，并拥有完全控制系统的权限。而 `sudo` 命令通过使用自己的密码来执行特权操作，可以避免共享 root 密码，提高系统的安全性。

3.4 记录日志

`sudo` 命令会记录每个特权操作的日志，包括执行的命令和执行者的身份。这可以帮助系统管理员进行安全审计和追踪。而 `su` 命令没有内置的日志功能，无法追踪用户切换和执行的命令。

3.5 使用方式

`su` 命令可以在命令行中直接输入，然后输入目标用户的密码即可切换用户。而 `sudo` 命令需要在命令前加上 `sudo` 关键字，然后输入自己的密码确认身份。

4. 适用场景

使用 `su` 命令切换用户适用于需要长时间以其他用户身份操作的情况，比如需要在其他用户的环境下执行一系列命令或长时间工作。

使用 `sudo` 命令适用于临时需要执行特权操作的情况，比如安装软件、更新系统或执行重要的系统维护任务。

5. 结论

`su` 命令和 `sudo` 命令是在 Linux 系统中用于切换用户身份和执行特权操作的常用命令。它们在使用方式、安全性和适用场景等方面存在一些区别。`su` 命令需要超级用户密码来切换用户身份，而 `sudo` 命令允许普通用户以自己的密码执行特权操作。

使用 `sudo` 命令可以避免共享 root 密码和记录特权操作的日志，提高系统的安全性和可追踪性。在选择使用 `su` 命令或 `sudo` 命令时，需要根据具体需求和安全考虑来进行选择。希望本文对您了解 `su` 命令和 `sudo` 命令的区别有所帮助，并能在实际使用中正确选择和使用这两个命令。

sudo使用当前用户临时设置的环境变量(sudo -E)

2023-06-12 02:26:33

1、 sudo -E

-E选项在man page中的解释是：

-E

The -E (preserve environment) option indicates to the security policy that the user wishes to preserve their existing environment variables. The security policy may return an error if the -E option is specified and the user does not have permission to preserve the environment.

简单来说，就是加上-E选项后，用户可以在sudo执行时保留当前用户已存在的环境变量，不会被sudo重置，另外，如果用户对于指定的环境变量没有权限，则会报错。

出于安全方面的考虑，使用sudo执行命令将在一个最小化的环境中执行，环境变量都重置成默认状态。所以PATH这个变量不包括用户自定义设置的内容

知道了作用下面看看解决办法：

```
# See sudoers(5) for more information on "#include" directives:
#include_dir /etc/sudoers.d
abc1234 ALL=(ALL) NOPASSWD:SETENV: /bin/whoami
~
```

编辑/etc/sudoers

将原来的默认abc1234用户的NOPASSWD增加SETENV:

这时候再执行sudo -E就可以直接运行了

被问Linux命令su 和sudo的区别？

鸭哥聊Java 2022-01-04 16:30

之前一直对 su 和 sudo 这两个命令犯迷糊，最近专门搜了这方面的资料，总算是把两者的关系以及用法搞清楚了，这篇文章来系统总结一下。

1. 准备工作

因为本篇博客中涉及到用户切换，所以我需要提前准备好几个测试用户，方便后续切换。

Linux 中新建用户的命令是 useradd，一般系统中这个命令对应的路径都在 PATH 环境变量里，如果直接输入 useradd 不管用的话，就用绝对路径名的方式：/usr/sbin/useradd

useradd 新建用户命令只有 root 用户才能执行，我们先从普通用户 ubuntu 切换到 root 用户（如何切换后文会介绍）：

```
1 ubuntu@VM-0-14-ubuntu:~$ su -
2 Password: # 输入 root 用户登录密码
3 root@VM-0-14-ubuntu:~# useradd -m test_user # 带上 -m 参数
4 root@VM-0-14-ubuntu:~# ls /home
5 test_user ubuntu # 可以看到 /home 目录下面有两个用户了
```

因为还没有给新建的用户 test_user 设置登录密码，这就导致我们无法从普通用户 ubuntu 切换到 test_user，所以接下来，我们需要用 root 来设置 test_user 的登录密码。需要用到 passwd 命令：

```
1 root@VM-0-14-ubuntu:~# passwd test_user
2 Enter new UNIX password:                # 输出 test_user 的密码
3 Retype new UNIX password:
4 passwd: password updated successfully
5 root@VM-0-14-ubuntu:~#
```

接着我们输入 exit 退出 root 用户到 普通用户 ubuntu:

```
1 root@VM-0-14-ubuntu:~# exit
2 logout
3 ubuntu@VM-0-14-ubuntu:~$
```

可以看到，命令提示符前面已经由 root 变成 ubuntu，说明我们现在的身份是 ubuntu 用户。

2. su 命令介绍及主要用法

首先需要解释下 su 代表什么意思。

之前一直以为 su 是 super user，查阅资料之后才知道原来表示「switch user」。

知道 su 是由什么缩写来的之后，那么它提供的功能就显而易见了，就是「切换用户」。

2.1 - 参数

su 的一般使用方法是：

```
1 su <user_name>
```

或者

```
1 su - <user_name>
```

两种方法只差了一个字符 -，会有比较大的差异：

如果加入了 - 参数，那么是一种 login-shell 的方式，意思是说切换到另一个用户 <user_name> 之后，当前的 shell 会加载 <user_name> 对应的环境变量和各种设置；

如果没有加入 - 参数，那么是一种 non-login-shell 的方式，意思是说我现在切换到了 <user_name>，但是当前的 shell 还是加载切换之前的那个用户的环境变量以及各种设置。

光解释会比较抽象，我们看一个例子就比较容易理解了。

我们首先从 ubuntu 用户以 non-login-shell 的方式切换到 root 用户，比较两种用户状态下环境变量中 PWD 的值（su 命令不跟任何 <user_name>，默认切换到 root 用户）：

```
1 ubuntu@VM-0-14-ubuntu:~$ env | grep ubuntu
2 USER=ubuntu
3 PWD=/home/ubuntu # 是 /home/ubuntu
4 HOME=/home/ubuntu
5 # 省略.....
6 ubuntu@VM-0-14-ubuntu:~$ su # non-login-shell 方式
7 Password: # 输入 root 用户登录密码
8 root@VM-0-14-ubuntu:/home/ubuntu# env | grep ubuntu
9 PWD=/home/ubuntu # 可以发现还是 /home/ubuntu
10 root@VM-0-14-ubuntu:/home/ubuntu#
```

我们的确是切换到 root 用户了，但是 shell 环境中的变量并没有改变，还是用之前 ubuntu 用户的环境变量。

接着我们从 ubuntu 用户以 login-shell 的方式切换到 root 用户，同样比较两种用户转台下环境变量中 PWD 的值：

```
1 ubuntu@VM-0-14-ubuntu:~$ env | grep ubuntu
2 USER=ubuntu
3 PWD=/home/ubuntu # 是 /home/ubuntu
4 HOME=/home/ubuntu
5 # 省略.....
6 ubuntu@VM-0-14-ubuntu:~$ su - # 是 login-shell 方式
7 Password:
```



```
8 root@VM-0-14-ubuntu:~# env | grep root
9 USER=root
10 PWD=/root # 已经变成 /root 了
11 HOME=/root
12 MAIL=/var/mail/root
13 LOGNAME=root
14 root@VM-0-14-ubuntu:~#
```

可以看到用 login-shell 的方式切换用户的话，shell 中的环境变量也跟着改变了。

「总结」：具体使用哪种方式切换用户看个人需求：

如果不想因为切换到另一个用户导致自己在当前用户下的设置不可用，那么用 non-login-shell 的方式；

如果切换用户后，需要用到该用户的各种环境变量（不同用户的环境变量设置一般是不同的），那么使用 login-shell 的方式。

切换到指定用户

前面已经介绍了，如果 su 命令后面不跟任何 <user_name>，那么默认是切换到 root 用户：

```
1 ubuntu@VM-0-14-ubuntu:~$ su -
2 Password: # root 用户的密码
3 root@VM-0-14-ubuntu:/home/ubuntu#
```

因为我们在 1. 准备工作 部分已经新建了一个 test_user 用户，并且我们也知道 test_user 用户的登录密码（root 用户设置的），我们就能从 ubuntu 用户切换到 test_user 用户：

```
1 ubuntu@VM-0-14-ubuntu:~$ su -  
2 Password:          # root 用户的密码  
3 root@VM-0-14-ubuntu:/home/ubuntu#
```

2.3 -c 参数

前面的方法中，我们都是先切换到另一个用户（root 或者 test_user），在哪个用户的状态下执行命令，最后输入 exit 返回当前 ubuntu 用户。

还有一种方式是：不需要先切换用户再执行命令，可以直接在当前用户下，以另一个用户的方式执行命令，执行结束后就返回当前用户。这就得用到 -c 参数。

具体使用方法是：

```
1 su - -c "指令串" # 以 root 的方式执行 "指令串"
```

我么看个例子：

```
1 ubuntu@VM-0-14-ubuntu:~$ cat /etc/shadow
```

```
2 cat: /etc/shadow: Permission denied           # ubuntu 用户不能直接查看 /etc/shadow 文件内容
3
4 ubuntu@VM-0-14-ubuntu:~$ su - -c "tail -n 4 /etc/shadow"
5 Password:                                     # 输入 root 用户密码
6 ubuntu:$1$fZKcWEDI$uwZ64uFvVbwpHTbCSgim0/:18352:0:99999:7:::
7 ntp*:17752:0:99999:7:::
8 mysql!:18376:0:99999:7:::
9 test_user:$6$.ZY1lj4m$ii0x9CG8h.JH1h6zKbfBXRuo1JmIDBhAd5eqhvW71bUQXTRS//89jcuTzRi1KqRkP8YbYW4VPxmTVHWRLYNGS/:18406:0:99999:7:::
10 ubuntu@VM-0-14-ubuntu:~$                    # 执行完马上返回 ubuntu 用户而不是 root 用户
```

这种执行方式和后面要介绍的 `sudo` 很像，都是临时申请一下 `root` 用户的权限。但还是有差异，我们接着往后看。

3.sudo命令介绍及主要用法

首先还是解释下 `sudo` 命令是什么意思。

`sudo` 的英文全称是 `super user do`，即以超级用户（`root` 用户）的方式执行命令。这里的 `sudo` 和之前 `su` 表示的 `switch user` 是不同的，这点需要注意，很容易搞混。

我们先介绍 `sudo` 命令能做什么事情，然后说明为何能做到这些，以及如何做到这些。

我们开始。

3.1 主要用法

我们在 Linux 中经常会碰到 Permission denied 这种情况，比如以 ubuntu 用户的身份查看 /etc/shadow 的内容。因为这个文件的内容是只有 root 用户能查看的。

那如果我们想要查看怎么办呢？这时候就可以使用 sudo：

```
1 ubuntu@VM-0-14-ubuntu:~$ tail -n 3 /etc/shadow
2 tail: cannot open '/etc/shadow' for reading: Permission denied      # 没有权限
3 ubuntu@VM-0-14-ubuntu:~$ sudo !!                                     # 跟两个惊叹号
4 sudo tail -n 3 /etc/shadow
5 ntp:!:17752:0:99999:7:::
6 mysql:!:18376:0:99999:7:::
7 test_user:$6$.ZY1lj4m$ii0x9CG8h.JH1h6zKbfBXRuolJmIDBHAd5eqhvW7lbUQXTRS//89jcuTzRilKqRkP8YbYW4VPxmTVHWRLYNGS/:18406:0:99999:7:::
8 ubuntu@VM-0-14-ubuntu:~$
```

实例中，我们使用了 sudo !! 这个小技巧，表示重复上面输入的命令，只不过在命令最前面加上 sudo。

因为我已经设置了 sudo 命令不需要输入密码，所以这里 sudo !! 就能直接输出内容。如果没有设置的话，需要输入当前这个用户的密码，例如本例中，我就应该输入 ubuntu 用户的登录密码。

两次相邻的 sudo 操作，如果间隔在 5min 之内，第二次输入 sudo 不需要重新输入密码；如果超过 5min，那么再输入 sudo 时，又需要输入密码。所以一个比较省事的方法是设置 sudo 操作不需要密码。后面介绍如何设置。

sudo 除了以 root 用户的权限执行命令外，还有其它几个用法，这里做简单介绍。

切换到 root 用户：

```
1 sudo su -
```

这种方式也能以 login-shell 的方式切换到 root 用户，但是它和 su - 方法是有区别的：

前者输入 sudo su - 后，需要提供当前用户的登录密码，也就是 ubuntu 用户的密码；

后者输入 su - 后，需要提供 root 用户的登录密码。

还有一个命令：

```
1 sudo -i
```

这个命令和 sudo su - 效果一致，也是切换到 root 用户，也是需要提供当前用户（ubuntu 用户）的登录密码。

我们现在切换到 test_user 用户，尝试显示 /etc/shadow 文件的内容：

```
1 ubuntu@VM-0-14-ubuntu:~$ su - test_user
2 Password:                                     # test_user 的密码
3 $ sudo cat /etc/shadow
4 [sudo] password for test_user:                # test_user 的密码
```

```
5 test_user is not in the sudoers file. This incident will be reported.  
6 $
```

我们会看到倒数第二行中的错误提示信息，我们无法查看 `/etc/shadow` 的内容，这是为什么？为什么 `ubuntu` 可以使用 `sudo` 但是 `test_user` 不行呢？

这就涉及到 `sudo` 的工作原理了。

3.2 sudo 工作原理

一个用户能否使用 `sudo` 命令，取决于 `/etc/sudoers` 文件的设置。

从 3.1 节中我们已经看到，`ubuntu` 用户可以正常使用 `sudo`，但是 `test_user` 用户却无法使用，这是因为 `/etc/sudoers` 文件里没有配置 `test_user`。

`/etc/sudoers` 也是一个文本文件，但是因其有特定的语法，我们不要直接用 `vim` 或者 `vi` 来编辑它，需要用 `visudo` 这个命令。输入这个命令之后就能直接编辑 `/etc/sudoers` 这个文件了。

需要说明的是，只有 `root` 用户有权限使用 `visudo` 命令。

我们先来看下输入 `visudo` 命令后显示的内容。

输入（`root` 用户）：

```
1 root@VM-0-14-ubuntu:~# visudo
```

输出：

```
1 # User privilege specification
2 root    ALL=(ALL:ALL) ALL
3
4 # Members of the admin group may gain root privileges
5 %admin   ALL=(ALL) ALL
6
7 # Allow members of group sudo to execute any command
8 %sudo    ALL=(ALL:ALL) ALL
9
10 # See sudoers(5) for more information on "#include" directives:
11
12 #includedir /etc/sudoers.d
13 ubuntu  ALL=(ALL:ALL) NOPASSWD: ALL
```

解释下每一行的格式：

- 1、第一个表示用户名，如 root 、 ubuntu 等；
- 2、接下来等号左边的 ALL 表示允许从任何主机登录当前的用户账户；
- 3、等号右边的 ALL 表示：这一行行首对一个的用户可以切换到系统中任何一个其它用户；
- 4、行尾的 ALL 表示：当前行首的用户，能以 root 用户的身份下达什么命令，ALL 表示可以下达任何命令。

我们还注意到 ubuntu 对应的那一行有个 NOPASSWD 关键字，这就是表明 ubuntu 这个用户在请求 sudo 时不需要输入密码，到这里就解释了前面的问题。

同时我们注意到，这个文件里并没有 test_user 对应的行，这也就解释了为什么 test_user 无法使用 sudo 命令。

接下来，我们尝试将 test_user 添加到 /etc/sudoers 文件中，使 test_user 也能使用 sudo 命令。我们在最后一行添加：

```
1 test_user ALL=(ALL:ALL) ALL      # test_user 使用 sudo 需要提供 test_user 的密码
```

接下来我们再在 test_user 账户下执行 sudo：

```
1 ubuntu@VM-0-14-ubuntu:~$ su - test_user
2 Password:
3 $ tail -n 3 /etc/shadow
4 tail: cannot open '/etc/shadow' for reading: Permission denied
5 $ sudo tail -n 3 /etc/shadow      # 加上 sudo
6 ntp:!:17752:0:99999:7:::
7 mysql:!:18376:0:99999:7:::
8 test_user:$6$.ZY1lj4m$ii0x9CG8h.JH1h6zKbfBXRuolJmIDBAd5eqhvW71bUQXTRS//89jcuTzRilKqRkP8YbYW4VPxmTVHWRLYNGS/:18406:0:99999:7:::
9 $
```


可以看到，现在已经可以使用 `sudo` 了。

3.3 思考

我们已经看到了，如果一个用户在 `/etc/sudoers` 文件中，那么它就具有 `sudo` 权限，就能通过 `sudo su -` 或者 `sudo -i` 等命令切换到 `root` 用户了，那这时这个用户就变成 `root` 用户了，那这不对系统造成很大的威胁吗？

实际上的确是这样的。所以如果在编辑 `/etc/sudoers` 文件赋予某种用户 `sudo` 权限时，必须要确定该用户是「可信任」的，不会对系统造成恶意破坏，否则将所有 `root` 权限都赋予该用户将会有非常大的危险。

当然，`root` 用户也可以编辑 `/etc/sudoers` 使用户只具备一部分权限，即只能执行一小部分命令。有兴趣的读者可以参考 Reference 部分第二条，这篇文章不再赘述。

4. 二者的差异对比

我们已经看到：

使用 `su -`，提供 `root` 账户的密码，可以切换到 `root` 用户；

使用 `sudo su -`，提供当前用户的密码，也可以切换到 `root` 用户

两种方式的差异也显而易见：如果我们的 Linux 系统有很多用户需要使用的话，前者要求所有用户都知道 `root` 用户的密码，这显然是非常危险的；后者是不需要暴露 `root` 账户密码的，用户只需要输入自己的账户密码就可以，而且哪些用户可以切换到 `root`，这完全是受 `root` 控制的（`root` 通过设置 `/etc/sudoers` 实现的），这样系统就安全很多了。

来源：网络