

## 深入理解 `sudo` 与 `su` 之间的区别

### 内容

本文，我们将详细讨论关于 `su` 命令与 `sudo` 命令之间的区别。

在开始之前有必要说明一下，文中所涉及到的示例教程都已经在 Ubuntu 14.04 LTS 上测试通过。

### Linux `su` 命令

`su` 命令的主要作用是让你可以在已登录的会话中切换到另外一个用户。换句话说，这个工具可以让你在不登出当前用户的情况下登录为另外一个用户。

`su` 命令经常被用于切换到超级用户或 `root` 用户（因为在命令行下工作，经常需要 `root` 权限），但是 - 正如前面所提到的 - `su` 命令也可以用于切换到任意非 `root` 用户。

如何使用 `su` 命令切换到 `root` 用户，如下：

#### *不带命令行参数的 `su` 命令*

如上，`su` 命令要求输入的密码是 `root` 用户的密码。所以，一般 `su` 命令需要输入目标用户的密码。在输入正确的密码之后，`su` 命令会在终端的当前会话中打开一个子会话。

### **`su -`**

还有一种方法可以切换到 `root` 用户：运行 `su -` 命令，如下：

#### *`su -` 命令*

那么，`su` 命令与 `su -` 命令之间有什么区别呢？前者在切换到 `root` 用户之后仍然保持旧的（或者说原始用户的）环境，而后者则是创建一个新的环境

（由 root 用户 `~/.bashrc` 文件所设置的环境），相当于使用 root 用户正常登录（从登录屏幕登录）。

su 命令手册页很清楚地说明了这一点：

可选参数 - 可提供的环境为用户在直接登录时的环境。

因此，你会觉得使用 `su -` 登录更有意义。但是，`su` 命令也是有用的，那么大家可能会想知道它在什么时候用到。以下内容摘自 [ArchLinux wiki 网站](#) - 关于 `su` 命令的好处和坏处：

有的时候，对于系统管理员 (root) 来讲，使用其他普通用户的 Shell 账户而不是自己的 root Shell 账户更会好一些。尤其是在处理用户问题时，最有效的方法就是是：登录目标用户以便重现以及调试问题。

然而，在多数情况下，当从普通用户切换到 root 用户进行操作时，如果还使用普通用户的环境变量的话，那是不可取甚至是危险的操作。因为是在无意间切换使用普通用户的环境，所以当使用 root 用户进行程序安装或系统更改时，会产生与正常使用 root 用户进行操作时不相符的结果。例如，以普通用户安装程序会给普通用户意外损坏系统或获取对某些数据的未授权访问的能力。

注意：如果你想在 `su -` 命令的 - 后面传递更多的参数，那么你必须使用

`su -l` 而不是 `su -`。以下是 - 和 `-l` 命令行选项的说明：

`-, -l, --login`

提供相当于用户在直接登录时所期望的环境。

当使用 - 时，必须放在 `su` 命令的最后一个选项。其他选项 (`-l` 和 `--login`) 无此限制。

`su -c`

还有一个值得一提的 `su` 命令行选项为：`-c`。该选项允许你提供在切换到目标用户之后要运行的命令。

`su` 命令手册页是这样说明：

`-c, --command COMMAND`

使用 `-c` 选项指定由 Shell 调用的命令。

被执行的命令无法控制终端。所以，此选项不能用于执行需要控制 TTY 的交互式程序。

参考示例：

```
su [target-user] -c [command-to-run]
```

示例中，`command-to-run` 将会被这样执行：

```
[shell]&nbsp;-c [command-to-run]
```

示例中的 `shell` 类型将会被目标用户在 `/etc/passwd` 文件中定义的登录 `shell` 类型所替代。

## **sudo vs. su**

现在，我们已经讨论了关于 `su` 命令的基础知识，是时候来探讨一下 `sudo` 和 `su` 命令之间的区别了。

## **关于密码**

两个命令的最大区别是：`sudo` 命令需要输入当前用户的密码，`su` 命令需要输入 `root` 用户的密码。

很明显，就安全而言，`sudo` 命令更好。例如，考虑到需要 `root` 访问权限的多用户使用的计算机。在这种情况下，使用 `su` 意味着需要与其他用户共享 `root` 用户密码，这显然不是一种好习惯。

此外，如果要撤销特定用户的超级用户/`root` 用户的访问权限，唯一的办法就是更改 `root` 密码，然后再告知所有其他用户新的 `root` 密码。

而使用 `sudo` 命令就不一样了，你可以很好的处理以上的两种情况。鉴于 `sudo` 命令要求输入的是其他用户自己的密码，所以，不需要共享 `root` 密码。同时，想要阻止特定用户访问 `root` 权限，只需要调整 `sudoers` 文件中的相应配置即可。

## 默认行为

两个命令之间的另外一个区别是其默认行为。`sudo` 命令只允许使用提升的权限运行单个命令，而 `su` 命令会启动一个新的 `shell`，同时允许使用 `root` 权限运行尽可能多的命令，直到明确退出登录。

因此，`su` 命令的默认行为是有风险的，因为用户很有可能会忘记他们正在以 `root` 用户身份进行工作，于是，无意中做出了一些不可恢复的更改（例如：对错误的目录运行 `rm -rf` 命令！）。关于为什么不鼓励以 `root` 用户身份进行工作的详细内容，请参考[这里](#)。

## 日志记录

尽管 `sudo` 命令是以目标用户（默认情况下是 `root` 用户）的身份执行命令，但是它们会使用 `sudoer` 所配置的用户名来记录是谁执行命令。而 `su` 命令是无法直接跟踪记录用户切换到 `root` 用户之后执行了什么操作。

## 灵活性

`sudo` 命令比 `su` 命令灵活很多，因为你甚至可以限制 `sudo` 用户可以访问哪些命令。换句话说，用户通过 `sudo` 命令只能访问他们工作需要的命令。而 `su` 命令让用户有权限做任何事情。

## **sudo su**

大概是因为使用 `su` 命令或直接以 `root` 用户身份登录有风险，所以，一些 Linux 发行版（如 Ubuntu）默认禁用 `root` 用户帐户。鼓励用户在需要 `root` 权限时使用 `sudo` 命令。

然而，您还是可以成功执行 `su` 命令，而不用输入 `root` 用户的密码。运行以下命令：

```
sudo su
```

由于你使用 `sudo` 运行命令，你只需要输入当前用户的密码。所以，一旦完成操作，`su` 命令将会以 `root` 用户身份运行，这意味着它不会再要求输入任何密码。

**PS：**如果你想在系统中启用 `root` 用户帐户（强烈反对，因为你可以使用 `sudo` 命令或 `sudo su` 命令），你必须手动设置 `root` 用户密码，可以使用以下命令：

```
sudo passwd root
```

## 结论

当你需要可用的工具来提升（或一组完全不同的）权限来执行任务时，这篇文章以及之前的教程（其中侧重于 `sudo` 命令）应该能给你一个比较好的建议。

如果您也想分享关于 `su` 或 `sudo` 的相关内容或者经验，欢迎您在下方进行评论。

---