

## sudo 入门指南

你在使用 Linux 命令行时曾经得到过“<sup>Permission denied</sup>拒绝访问”的错误提示吗？这可能是因为你正在尝试执行一个需要 **root** 权限的操作。例如，下面的截图展示了当我尝试复制一个二进制文件到一个系统目录时产生的错误。

### *shell 的拒绝访问*

那么该怎么解决这个错误？很简单，使用 **sudo** 命令。

### *用 sudo 运行命令*

用户运行此命令后会被提示输入他们（自己）的登录密码。一旦输入了正确的密码，操作将会成功执行。

毫无疑问，**sudo** 是任何在 Linux 上使用命令行的人都必须知道的命令。但是，为了更负责、更有效地使用该命令，你还是要知道一些相关（及深入）的细节。这正是我们将会在这篇文章中讨论的。

*在我们继续之前，值得提一下的是，这篇文章所提到的所有命令指示都已经在 Ubuntu 14.04 LTS 下的 4.3.11 版 Bash 下通过测试。*

## 什么是 **sudo**

正如你们大部分人所知道的，**sudo** 用来执行需要提升权限（通常是作为 **root** 用户）的命令。在这篇文章之前的简介部分已经讨论过这样的例子。然而，如果你想的话，你能用 **sudo** 以其它（非 **root**）用户运行命令。

这是由工具提供的 **-u** 命令行选项所实现的。举个例子，如下例所展示的那样，我（himanshu）尝试将一个在其他用户（howtoforge）的 **Home** 目录中的文件重命名，但是得到一个“访问拒绝”的

错误。然后我加上 **sudo -u howtoforge** 后用同样的“**mv**”命令，命令成功执行了：

### *什么是 sudo*

## 任何人都能用 **sudo** 吗？

不是。一个用户要能使用 **sudo**，应该在 **/etc/sudoers** 文件里有一条跟该用户相关的信息。下述摘自 Ubuntu 网站的一段能讲得更清楚：

`/etc/sudoers` 文件控制了谁能以哪个用户的身份在哪个机器上运行什么命令，还可以控制特别的情况，例如对于特定的命令是否需要输入密码。这个文件由 <sup>aliases</sup> 别名（基本变量）和 <sup>user specifications</sup> 用户标识（控制谁能运行什么命令）组成。

如果你正在使用 **Ubuntu**，让一个用户能运行 `sudo` 命令很容易：你所需要做的就是将账户类型改成

<sup>administrator</sup> 管理员。这可直接在 <sup>System Settings</sup> 系统设置 -> <sup>User Accounts</sup> 用户账户 里完成。

*sudo 用户*

首先解锁该窗口：

*unlocking window*

然后选择你想改变用户类型的用户，然后将类型改成 <sup>administrator</sup> 管理员。

*choose sudo accounts*

然而，如果你不使用 **Ubuntu**，或者你的发行版没有提供这个特性，你可以手动编辑 `/etc/sudoers` 文件来实现此改变。要在文件中添加这样的一行：

1. `[user] ALL=(ALL:ALL) ALL`

无需赘言，`[user]` 应该用你想提升 **sudo** 权限的用户的用户名所代替。在这里值得提到的一件重要的事情是，官方建议通过 `visudo` 命令编辑该文件 —— 你需要做的就是运行下述命令：

1. `sudo visudo`

为了说清究竟是怎么一回事，这里有段从 `visudo` 手册里的摘要：

`visudo` 以安全的模式编辑 `sudoers` 文件。`visudo` 锁定 `sudoers` 文件以防多个编辑同时进行，提供基本的检查（**sanity checks**）和语法错误检查。如果 `sudoers` 文件现在正在被编辑，你将会收到一个信息提示稍后再试。

关于 `visudo` 的更多信息，前往[这里](#)。

## 什么是 **sudo** 会话

如果你经常使用 `sudo` 命令，你肯定注意到过当你成功输入一次密码后，可以不用输入密码再运行几次 `sudo` 命令。但是一段时间后，`sudo` 命令会再次要求你的密码。

这种现象跟运行 `sudo` 命令数目无关，跟时间有关。是的，`sudo` 默认在输入一次密码后 **15** 分钟内不会再次要求密码。**15** 分钟后，你会再次被要求输入密码。

然而，如果你想的话，你能改变这种现象。用以下命令打开 `/etc/sudoers` 文件：

```
1. sudo visudo
```

找到这一行：

```
1.Defaults env_reset
```

*env\_reset*

然后在这行最后添加以下变量：

```
1.Defaults env_reset,timestamp_timeout=[new-value]
```

[new-value] 为想要 `sudo` 会话持续的时间数。例如，设数值为 40。

*sudo timeout value*

如果你希望每次使用 `sudo` 命令时都要求输入密码，你可以把这个变量赋值为 0。想要 `sudo` 会话永远不过时，应赋值为 -1。

注意将 `timestamp_timeout` 的值赋为 "-1" 是强烈不推荐的。

## **sudo 密码**

你可能注意过，当 `sudo` 要求输入密码然后你开始输入时，不会显示任何东西 —— 甚至连常规的星号都没有。虽然这不是什么大问题，不过一些用户就是希望显示星号。

好消息是那有可能也很容易做到。所有你需要做的就是 在 `/etc/sudoers` 文件里将下述的行：

```
1.Defaults env_reset
```

改成

```
1.Defaults env_reset,pwfeedback
```

然后保存文件。

现在，无论什么时候输入 `sudo` 密码，星号都会显示。

*hide the sudo password*

## **一些重要的 sudo 命令行参数**

除了 `-u` 命令行参数（我们已经在这篇教程的开始部分讨论过了），还有其他重要的 `sudo` 命令行参数值得注意。在这部分，我们将会讨论其中一些。

## **-k 参数**

考虑下这种情况：输入密码后你刚刚运行了几个 `sudo` 驱动的命令。现在，如你所知，`sudo` 会话默认保持 15 分钟。假设在这会话期间，你需要让某些人访问你的终端，但你不想让他们可以使用 `sudo`，你将会怎么做？

还好，有 `-k` 命令行参数允许用户取消 `sudo` 权限。这是 `sudo` <sup>man page</sup> 帮助页面对此的解释：

```
-k, --reset-timestamp
```

不带任何命令使用时，撤销用户缓存的凭据。换句话说讲，下一次使用 `sudo` 将会要求输入密码。使用这个参数不需要密码，也可以放到一个 `.logout` 文件中来撤销 `sudo` 权限。

当与一个命令，或者一个可能需要密码的操作一起用时，这个参数将会导致 `sudo` 忽略用户缓存的凭据。结果是 `sudo` 要求输入密码（如果这是被安全策略所要求的），而且不会更新用户缓存的凭据。

## **-s 参数**

有时你的工作要求你运行一堆需要 `root` 权限的命令，你不想每次都输入密码。你也不想通过改变 `/etc/sudoers` 文件调整 `sudo` 会话的过期时限。

这种情况下，你可以用 `sudo` 的 `-s` 参数。这是 `sudo` 帮助页面对此的解释：

```
-s, --shell
```

如果设置了 `SHELL` 环境变量或者调用用户的密码数据库指定了 `shell`，就运行该 `shell`。如果指定了命令，命令将会通过 `shell` 的 `-c` 参数将命令传递给该 `shell` 执行。如果没有指定命令，会执行一个交互式 `shell`。

所以，基本上这命令参数做的是：

- 启动一个新的 `shell` - 至于是哪一个 `shell`，参照 `SHELL` 环境变量赋值。如果 `$SHELL` 是空的，将会用 `/etc/passwd` 中定义的 `shell`。
- 如果你用 `-s` 参数传递了一个命令名（例如 `sudo -s whoami`），实际执行的是 `sudo /bin/bash -c whoami`。
- 如果你没有尝试执行其他命令（也就是说，你只是要运行 `sudo -s`），你将会得到一个有 `root` 权限的交互式的 `shell`。

请记住，`-s` 命令行参数给你一个有 `root` 权限的 `shell`，但那不是 `root` 环境 —— 还是执行的你自己的 `.bashrc`。例如，在 `sudo -s` 运行的新 `shell` 里，执行 `whoami` 命令仍会返回你的用户名，而非 `root`。

## **-i 参数**

`-i` 参数跟我们讨论过的 `-s` 参数相像。然而，还是有点区别。一个重要的区别是 `-i` 给你的是 `root` 环境，意味着你的（用户的）`.bashrc` 被忽略。这就像没有显式地用 `root` 登录也能成为 `root` 。此外，你也不用输入 `root` 用户密码。

**重要：** 请注意 `su` 命令也能让你切换用户（默认切换到 `root` ）。这个命令需要你输入 `root` 密码。为了避免这一点，你可以使用 `sudo` 执行它（`sudo su`），这样你只需要输入你的登录密码。然而，`su` 和 `sudo su` 有隐含的区别 —— 要了解它们，以及它们和 `sudo -i` 的区别，请看[这里](#)。

## 总结

我希望现在你至少知道了 `sudo` 的基本知识，以及如何调整 `sudo` 的默认行为。请按我们解释过的那样去尝试调整 `/etc/sudoers` 。同时也浏览一下论坛讨论来更深入了解 `sudo` 命令。