

By David B. 2006 Linux Editor

## Grant Root Privileges to Regular Users

*The proper way to allow certain regular users to run commands as root.*

A multi-user system is a Linux computer which is used by other persons besides you. If you run such a multi-user system, you probably know that users sometimes need to run certain commands as root. Of course, you can't just give them the root password so wouldn't it be nice to allow particular users to run certain commands that require root privileges without having to tell them the root password? There's a well-known tool which will solve this problem, called **sudo**. **sudo** (**SU**peruser **DO**) is a tool for Unix-like operating systems that allows normal users to run programs with the security privileges of the system's superuser in a secure manner. Users will have to confirm their identity to sudo by typing-in their password before running the target program. However, which users are allowed to use sudo, what commands are they allowed to execute, as well as other related settings can be configured through the `/etc/sudoers` file. The `/etc/sudoers` content will differ from distribution to distribution but the structure, however, will most likely be the same in all distributions. The default sudoers file will look like this:

```
[CODE=0]# Host alias specification#
User alias specification#
Cmnd alias specification#
Defaults#
User privilege specificationroot    ALL=(ALL)
ALL[CODE=1]
```

**Allow a local user to run root commands-** Take for example the `/sbin/shutdown` command: by default, you won't be able to execute it unless you have root privileges. In order to allow a local user (I'll use `softpedia` as the example user) to shut down the computer, you'll have to define the alias which represents the shutdown command by adding this line in the *Cmnd alias* section:

```
[CODE=0]# Cmnd alias
specificationCmnd_Alias    SHUTDOWN = /sbin/shutdown[CODE=1]-
```

Then, in the *User privilege* section, you'll have to add the line:

```
[CODE=0]# User privilege specificationsoftpedia ALL = SHUTDOWN[CODE=1]-
```

Now, the user `softpedia` will be able to shutdown the computer by using the command: **# sudo shutdown -h now**. The `sudo` program will prompt users for their own password (not the root's) before executing the command. If you wish to setup `sudo` for not prompting users for any password, edit the line in the *User privilege* section to look like this:

```
[CODE=0]softpedia ALL = NOPASSWD: SHUTDOWN[CODE=1]
```

**Allow a local user to run root commands without sudo or su**  
**NOTE:** This will allow the specified user to run any command as root, without having to use `su`, `sudo` or constantly type in his or the root password. - First, add the **OWNER** alias by adding the following lines in the *User alias* and *User privilege* sections of `/etc/sudoers` file:

```
[CODE=0]# User alias specificationUser_Alias OWNER = softpedia#
User privilege specificationOWNER
ALL = NOPASSWD: ALL[CODE=1]
```

**Allow users that are part of certain groups to run root commands-** You can define a group of people who are allowed to perform certain administration commands which require root privileges. This can be achieved by adding the following lines in the *User alias*, *Cmnd alias* and *User privilege* sections:

```
[CODE=0]# User alias specificationUser_Alias ADMINS = softpedia,john,david#
Cmnd alias specificationCmnd_Alias UPDATE = /usr/bin/yumCmnd_Alias REBOOT = /usr/bin/reboot#
User privilege specificationADMINS ALL = UPDATE, REBOOT[CODE=1]-
```

This example will allow the user `softpedia`, `john` and `david` to periodically update the system through **yum** and reboot it after the update is complete. Of course, you can define other commands and other users, but keep the same structure. All commands defined can be executed by those users without having the root password. - The commands are fully logged via `syslog`.