

# ***Root, SU and SUDO***

**By Whil Hentzen**

The Linux "root" user has what many system admins refer to as 'god' powers – complete control over the machine and environment. In the next breath, any competent sysadmin will admonish you to avoid logging in as root unless absolutely necessary, and then they'll add for effect, "and it's almost never necessary." But the new Linux admin or user will find that root access seems to be necessary a lot more than their admin friend lets on. The reason the experienced admin doesn't need to log on as root is that they've got a couple of root access tricks up their sleeve, namely, the 'su' and 'sudo' commands. In this article, I'll explain how to use 'su' and 'sudo' to avoid a lot of root logins.

## 1. Preface

### 1.1 Copyright

Copyright 2006 and beyond Whil Hentzen. Some rights reserved. This work is licensed under the Creative Commons Attribution-NonCommercial-NoDerivs License, which basically means that you can copy, distribute, and display only unaltered copies of this work, but in return, you must give the original author credit, you may not distribute the work for commercial gain, nor create derivative works based on it without first licensing those rights from the author. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-nd/2.0/>.

### 1.2 Revisions

#### 1.2.1 History

Version	Date	Synopsis	Author
1.0.0	2006/11/04	Original	WH

#### 1.2.2 New version

The newest version of this document will be found at [www.hentzenwerke.com](http://www.hentzenwerke.com).

#### 1.2.3 Feedback and corrections

If you have questions, comments, or corrections about this document, please feel free to email me at 'articles@hentzenwerke.com'. I also welcome suggestions for passages you find unclear.

### 1.3 References and acknowledgments

Thanks to many MLUG members for various tips and tricks, and for not laughing when I asked dumb questions.

### 1.4 Disclaimer

No warranty! This material is provided as is, with no warranty of fitness for any particular purpose. Use the concepts, examples and other content at your own risk. There may be errors and inaccuracies that in some configurations may be damaging to your system. The author(s) disavows all liability for the contents of this document.

Before making any changes to your system, ensure that you have backups and other resources to restore the system to its state before making those changes.

All copyrights are held by their respective owners, unless specifically noted otherwise. Use of a term in this document should not be regarded as affecting the validity of any trademark or service mark. Naming of particular products or brands should not be seen as endorsements.

### 1.5 Prerequisites and assumptions

This document was written using Fedora Core 6, and assumes a beginner's familiarity with use of Linux via the GUI and the Command Window. I also assume that you, as a regular user, have root access to your machine. The rest of this article uses the Command Window for its examples and exercises.

## 2. Command prompts for the root user vs. regular users

The command window prompt ends either in a dollar sign (\$) or a pound sign (#). The \$ means that you are logged on as a regular user. The # means you are logged on as root. How do you remember the difference? Some people refer to the pound sign as a 'splat', which is exactly what you'll do as a root user if you aren't sure about a command and you're not careful.

## 3. Changing to root inside the command window

You can change to root after opening a command window in one of two ways: via "su" and via "su -".

### 3.1. The "su" command

The first is by using the 'su' command:

```
[bob@mymachine ~]$ su
Password:
```

```
[root@mymachine bob]#
```

The password is for the root user, not the regular user who was originally logged in. Note that you're still in the user's home directory, but now you're root, which means that you have godlike powers. You'll also notice that the title bar of the command window has also changed. You can make any change you want to the system, such as install software, delete files, get into anyone else's home directory, and generally cause all sorts of havoc if you're not careful. For example, "rm -rf /" will destroy every file on the machine in about five blimptoseconds.

### 3.2. The "su -" command

The second way is using the "su -" command:

```
[bob@mymachine ~]$ su -  
Password:  
[root@mymachine ~]#
```

If you use this method, you'll note that both the prompt string and the title bar have changed. Also note that you're now in the root user's home directory ("/root") instead of the original user's home directory ("/home/bob"). The reason is that the "-" after the "su" command brings root's environment along with it. What's an environment? It's a group of settings, such as the prompt, system variables, and so on, that belong to a specific user. The home directory, for example, is one such setting, and the command prompt string is another. These can be different (and often are) for different users.

When you use the "su" command, you're just gaining root privileges, but you're still working in the original user's environment, with the original user's path, prompt, home directory, and system variables.

When you use the "su -" command, you switch your environment to that of root, and that's why you're suddenly lauched into root's home directory. "su -" is short for "su -l" or "su -login", by the way.

## 4. Using SUDO for a single "su" command

Oftentimes you'll find yourself switching to the root user via 'su' in order to run just one command. It can be a nuisance if you're doing this over and over. You might want to keep a spare terminal window open and set up as root, but on the other hand, you might find that to be too much of a temptation, or just too dangerous.

The "sudo" command allows you to get around this. Specifically, "sudo" ("SuperUser DO"), after being set up by a system administrator, gives a user the ability to run a command as another user. When set up to allow a user to run a command as root, "sudo" obviates the need for the regular user to "su" to root before running that command. As an aside, the commands and arguments that the regular user executes via "sudo" are logged.

### 4.1 Set up "sudoers"

In order to use "sudo" as a regular user, the root user has to make an entry in a special file, "/etc/sudoers", that provides permission for the user to run a specifically designated command as another designated user on a designated host. This is done with a special program called "visudo". While "sudoers" is just a plain text file, you are advised to edit it as advised, using the 'visudo' program instead of just opening "/etc/sudo" with your favorite text editor is recommended for a couple of reasons. First, "visudo" will do syntax checking to make sure that the entry or entries you've made are correct, and second, conveniently, "visudo" knows where "sudoers" is located. Some folks will claim that they edit "sudoers" with their own editor and have never had problems, but this isn't for inexperienced or the faint of heart.

In order to give the user "bob" the ability to run the "kill" command on host 'mymachine', run "visudo" as root. You'll see that the "/etc/sudoers" file is opened with the vi text editor, so you've got access to standard vi commands in command mode and vi editing in insert mode.

The manual for "sudo" (see Section 5 for more info) provides a wealth of information on how to construct entries in "sudoers", as they can be robust and extremely flexible, providing the ability to create aliases for users, hosts and commands. The manual can thus also be overwhelming, so here are a couple of simple examples.

To allow "bob" to run the script "/etc/somescript"

```
bob mymachine=/etc/somescript
```

To allow "bob" to run the /usr/bin/kill command

```
bob mymachine=/usr/bin/kill
```

and to allow "bob" to run the /bin/ls command,

```
bob mymachine=/bin/ls
```

## 4.2 Run "sudo"

Once the "sudoers" command has been modified, it's time for "bob" to use sudo to snoop around. Normally, bob can't access the contents of the root user's home directory:

```
[bob@mymachine ~]$ ls /root
ls: /root: Permission denied
```

as expected. But using "sudo" and entering root's password, bob can be nosy:

```
[bob@mymachine ~]$ sudo ls /root
Password:
anaconda-ks.cfg  Desktop  install.log  scsrn.log  something_secret
[bob@mymachine ~]$
```

After the command executes, the user is returned to their original permissions; sudo allowed 'bob' to run just one command as root, as desired.

## 5. Where to go for more information

Naturally, "man sudo" will provide detailed reference information, but there's a great manual with examples galore at sudo's home page, [www.sudo.ws](http://www.sudo.ws).

## 6. About the author

Whil Hentzen started out life in the early '80's as a custom software developer using dBASE II (he still has the original 8 1/2 x 11 grey binder of documentation, much to the chagrin of his wife), and switched to FoxPro in 1990. Besides billing 15,000 hours in the 90's, he presented more than 70 papers at conferences throughout North America and Europe, edited FoxTalk, Pinnacle Publishing's high end technical journal for 7 years, hosted the Great Lakes Great Database Workshop since 1994. He's written 7 books and published 30 more on a variety of software development topics. He was a Microsoft Most Valuable Professional from 1995 through 2003 for his contributions to the FoxPro development community, and received the first Microsoft Lifetime Achievement Award for Visual FoxPro in 2001.

Whil began using Linux on the desktop when OpenOffice.org became a standard in the mainstream distributions, as it spelled potential for custom application development in the future, and has been a Linux user, developer, and evangelist ever since. His first book on Linux, Linux Transfer for Windows Power Users, was published in early 2004.

He is available for new and legacy Visual FoxPro application development as well as Web and desktop development on Linux.

## 7. A word from our sponsor

This free whitepaper is published and distributed by Hentzenwerke Publishing, Inc. We have the largest lists of "Moving to Linux", OpenOffice.org, and Visual FoxPro books on the planet.

We also have oodles of free whitepapers on our website and more are being added regularly. Our Preferred Customer mailing list gets bi-monthly announcements of new whitepapers (and gets discounts on our books, first crack at special deals, and other stuff as we think of it.)

Click on "Your Account" at [www.hentzenwerke.com](http://www.hentzenwerke.com) to get on our Preferred Customer list.

**If you found this whitepaper helpful, check out these Hentzenwerke Publishing books as well:**

**Linux Transfer for Windows® Network Admins:  
A roadmap for building a Linux file and print server  
Michael Jang**

**Linux Transfer for Windows® Power Users:  
Getting started with Linux for the desktop**

**Whil Hentzen**