



Administration, the root User and Configuring sudo

1 Aim

The aim of this activity is to understand the purpose of the `root` account, and the security risks of logging in as `root`, and how to avoid them. The student will appreciate that system administration with `sudo` is a useful help to security.

2 Background

You normally log into the computer system as your own user account. There are only two places where you can create or edit files on the computer system:

- in your *home directory*;
- in the `/tmp` directory.

Everywhere else is read only, or offers you no access.

However, to perform system administration, you sometimes need to change files outside of your home directory, and outside of the `/tmp` directory. How can you perform system administration? You need to temporarily become the `root` user. The `root` user account is sometimes called the *superuser*, since this account allows access to everything. This power is too great for most activities; with a simple typing mistake, you can delete the entire hard disk. Viruses and worms running under the `root` account can damage anywhere at all, and have access to everything on the computer. Buggy software running as the `root` user can do unlimited damage. Running all software as the `root` user is an accident waiting to happen. It is like using a chainsaw to cut a loaf of bread.

There are two tools with which you should be familiar: the program `su`, and more importantly, `sudo`. Here we see how to use them.

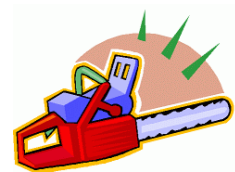


Figure 1: Doing everything as `root` is like cutting bread with a chainsaw.

3 Procedure

3.1 Using the `su` Program to Become `root`

1. Log into your Linux system with your own user account, *not* as `root`.
2. In a terminal window, type:

```
$ su -
```

Do not type the `$`; that indicates the prompt, and that you are typing this as a normal user. When you log in as `root`, the prompt becomes a hash: `#`.

The minus sign `'-'` tells `su` to run the login scripts of the user you are changing to (here, it is the `root` user).

3. When prompted, type in your `root` password (note, this is different from `sudo`).
4. Notice that your prompt has changed to a hash: `#`.
5. To run graphical programs (an example is `xclock`), you may need to type in another terminal window (or the same one before you type the `su` command) the following command:

```
$ xhost +localhost
```

6. To exit (log out) from the `root` account, type:

```
# exit
```

3.2 Advantages and Disadvantages of Using the `su` Program

There is quite a lot of convenience in opening one window, `su`ing to `root`, and leaving that window open. However, when you go to get that cup of tea, someone else may come and take advantage of this. The person who uses `su` must also know the `root` password. The more people who know a secret, the less of a secret it is. It is more secure to use the program `sudo`, which we discuss now.

3.3 The `sudo` Program

The `sudo` program allows a senior system administrator to keep the `root` password to themselves, and to delegate responsibility for various tasks to other junior system administrators.

There are some advantages to doing things this way, including the fact that every command executed using `sudo` is recorded in the system logs. This can help administrators coordinate their efforts; they can see what the other has done. Also, a cracker who breaks into the system and who gets access to the `root` account will be unlikely to use `sudo`, and the break-in will be clearly visible in the system logs (unless the person is smart enough to cover their tracks). Probably the most important thing is that the administrator does not need to know or remember the `root` password, thus making it easier to keep secret, and allowing it to be changed more often.

3.4 Configuring `sudo`

1. Become `root` using the `su` program, as described above in section 3.1 on the preceding page.
2. Type:

```
# visudo
```

3. Refer to the chapter about `vi` in the Linux Workshop notes for guidance on using the `vi` text editor. You will find the `'o'` and `'i'` commands useful.

4. Or, if you prefer emacs to vi like I do, then you could type:

```
# EDITOR=emacs visudo
```

5. Edit the file so that it looks like this, taking care to type it accurately, but put your username instead of mine:

```
# sudoers file.
#
# This file MUST be edited with the 'visudo' command as root.
#
# See the sudoers man page for the details on how to write a sudoers file.
#

Defaults timestamp_timeout = 10

# Host alias specification

# User alias specification

# Cmnd alias specification
# User privilege specification
root    ALL=(ALL) ALL
nicku   ALL=(ALL) ALL
```

Note that two lines were added, nothing else is changed. But please use *your own* LDAP user name here, not `nicku`!

See `man sudoers` and search for `Defaults`, and also `timeout` for information about the first line. It changes the time that may elapse since you last used `sudo` before you need to enter your password again. The default is normally five minutes.

6. Save it and exit by typing (in vi):

```
(Esc) :wq
```

and `sudo` is now configured. Here we have assigned full privileges to the user, but `sudo` can be set up to assign restricted administration rights to junior administrators. You can learn more by reading the `man` (manual) pages for `sudoers`, `sudo` and `visudo`.

7. Log out from being `root` by typing:

```
# exit
```

Your prompt should end with a dollar; if not, then you have run `su` - more than once. Continue typing `exit` until your prompt ends with a dollar: `$`.

3.5 Using Sudo

3.5.1 Some Background

To use `sudo`,

1. you type `sudo` in front of the command you want to execute as the `root` user.
2. The first time you do this, you will see a warning like this:

```
We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these two things:
```

- ```
#1) Respect the privacy of others.
#2) Think before you type.
```

```
Password:
```

3. At that prompt, you will type *your own password*, *not* that of the `root` user.
4. If you use `sudo` within the next ten minutes (normally five) within the same terminal window, you will not need to type your password.
5. If you walk away from your computer at all, you can cause the ten minute period to expire immediately by typing:

```
$ sudo -k
```

If `sudo` is used immediately after this, a password will be required.

6. Before running a graphical program with `sudo` for the first time in one login session, you need to type (as your own self):

```
$ xhost +localhost
```

to allow any user who is logged into your machine, to display graphics on the X server (the network graphics system on which you run your graphical user interface). This includes the user `root`, who is otherwise barred from display while you are running X from your account.

### 3.5.2 Exercise using sudo

1. Try to display the special *log* file `/var/log/secure`:

```
$ cat /var/log/secure
cat: /var/log/secure: Permission denied
```

Hmm, it says permission is denied. Only system administrators can see this log file. Let's use `sudo`:

```
$ sudo cat /var/log/secure
```

Make sure you enter *your own* password, and not that of `root`.

### 3.6 Why Use sudo?

There are four main reasons for using `sudo`:

- Only one person needs to know the root password. A secret shared between ten system administrators is no longer a secret.
- You only run as root the commands that you need to run as root. This increases security significantly. The less you do at the highest privilege level, the better.
- A senior system administrator may *delegate* only some duties to others (i.e., backup, printing administration, ...), since `sudo` allows the senior administrator to allow privileged access to only the required commands.
- Every command executed using `sudo` is recorded in a *log file*. On Red Hat 9, this is `/var/log/security`. For each command executed using `sudo`, the following details are recorded:
  - The time the command was executed
  - The user who executed the command
  - What user the command was effectively executed as (usually root, but you can change that with the `-u <user>` option to `sudo` — see `man sudo`).
  - What the current directory was that the command was executed in
  - What terminal was used (i.e., was the user logged in locally, or over the network?)
  - The exact command, with its exact location in the file system.

This log allows the system administrator to go back and find out what was done when by who. If the system stops working properly, the logs can provide information about what was done at the time, and if a mistake was made, it can be identified and rectified.

### 3.7 Warning!

From now on, if I see anyone logged in as the `root` user into the X Window system, I will make a mental note of this person as being a poor student, and I will then take careful note of any other poor behaviour! Seriously, it is *very* amateur behaviour to remain consistently logged into an X Window system as the `root` user. If I catch you doing this, expect a stern warning.

Similarly, I do not want to see you using a window `sued` to `root` for extended periods of time, and I will similarly criticise you for doing so. Such behaviour leaves the system open to security abuse by others.

I want all of you to get into the habit of using `sudo` for system administration tasks. You will be a better system administrator for this, and your system will be less easy to crack.

Note that at your workplace, you should ensure that the number of people who can use `sudo` on the system you are responsible for is limited as much as possible.