Enabling Sudo on Slackware

Sudo, substitute user of (Super User) or Super user of (acting as super user) has a big role on UNIX Likes systems, sudo allows ordinary users to temporarily get privileges from another user, widely used for the privileges of the superuser root.

There are many advantages and disadvantages of using sudo over su, for example sometimes we just need to perform a task quickly, such as updating the system with a single command, such as 'slackpkg update && slackpkg upgrade-all' or simply opening a file that has write and read permission only for root, for example.

But every superhero pays his price, as sudo allows the user to have super powers temporarily, so anyone who knows the password of your average user can have these super privileges.

This is why it is extremely important that you have a secure password for root as well as your regular user, so you can use sudo with peace of mind.

Unlike other distributions Slackware comes by default with sudo disabled, and we will learn how to enable it.

The first step is to create a group called “sudo”, for this simple task we can do it in two ways. The first is by manually creating a group, and the second is by using the 'groupadd GROUPName' command.

Let's use manual mode for better learning and notion.

The first step is to log in as root, for that use the su command, and right after we will open a groups configuration file located in / etc / group, open it with your favorite editor:

```
$ su
# nano /etc/group
```

Let's navigate to the last line of the 'group' file and add a special line to sudo, my penultimate line contains the privoxy group, your file is sure to be different:

```
privoxy:x:206:
```

In this same format we will create the sudo group, the format should be as follows:

```
groupname:x:ID:USERNAME
```

To check your user ID run the 'id -u' command, remember to be sudo enabled with the user.

```
$ id -u
1000
```

We then add in the last line of the group file:

```
sudo:x:1000:Username
```

Example:

```
nobody:x:98:nobody
nogroup:x:99:
users:x:100:
console:x:101:
tor:x:220:
privoxy:x:206:
sudo:x:1000:slackjeff
```

After this process, save and close.

Now we will need to edit the 'sudoers' file, so open with your favorite editor the file / etc / sudoers.

```
# nano /etc/sudoers
```

Open the file, find the line #%sudo ALL=(ALL)ALL, this line is commented out with the trailing '#' in front of% sudo, we need to **uncomment**, remove # in front of% sudo to have effect ... if you are using the nano editor, you can use the keys simultaneously CTRL + W, will open a search field in the lower left corner, just enter %sudo ALL=(ALL)ALL for the location to be made.

Commented sudo line:

```
## Uncomment to allow members of group sudo to execute any command
# %sudo ALL=(ALL) ALL
```

Uncommented sudo line:

```
## Uncomment to allow members of group sudo to execute any command
%sudo ALL=(ALL) ALL
```

Uncommented the line of sudo, save and close, to put the icing on the cake we need to make a last setting that is very important. We know that regular users have UID 1000, and some commands are special for the super user that contains UID 0, when we run a command the system looks in the $ PATH environment variable for the location of the command we ask.

There are directories like the example '/ sbin' which is accessible only with users of UID 0, an example is root itself. What happens if we try to execute a command like slackpkg for example that is inside directory? It will fail.

So we need to add two new lines to our PATH.

For this still as root user let's open the file 'profile' which is located in '/ etc / profile' and find the line:

```
# Set the default system $PATH:
PATH="/usr/local/bin:/usr/bin:/bin:/usr/games"
```

Note that there is an established pattern, directories have their field separated by ':'. At the end of '/usr/games' we add ':' and we add '/sbin' and after that we add ':' again and '/usr/sbin'. Save and close.

```
# Set the default system $PATH:
PATH="/usr/local/bin:/usr/bin:/bin:/usr/games:/sbin:/usr/sbin"
```

Please note that your regular user (UID 1000) can now execute the maintenance commands that are found in /sbin and /usr/sbin, whereas before these commands in these directories were not even 'shown' to a regular user (UID). 1000). But even now seeing these commands we will need the super user to continue.

Simply put, it is the same as giving a read permission to a file, for a given user it can read but cannot write or execute, for example.

Reboot your system with the shutdown command with parameters -r now, or simply exit your user and return so that you can execute the exit command or simultaneously press the **CTRL + D** keys.

Run tests by adding sudo in front of the desired command (s). Also test by updating your list and checksuns with slackpkg.

```
$ sudo ls /root/
$ sudo slackpkg update
```

Sources

- Originally written by Slackjeff

howtos, misc, sudo on slackware, enabling, author slackjeff