The Power of Sudo

In this chapter, you will learn how to give permissions to non-root users on the system so they can run privileged commands. In real life, the system administrator should not give the root password to any user on the system. However, some users on the system may need to run privileged commands; now, the question is: *how can non-root users run privileged commands without getting root access to the system?* Well, let me show you!

Examples of privileged commands

You would find most of the commands that require root privileges in the directories [/sbin] and [/usr/sbin]. Let's switch to user [smurf]:

```
elliot@ubuntu-linux:~$ su - smurf
Password:
smurf@ubuntu-linux:~$
```

Now let's see if [smurf] can add a new user to the system:

```
smurf@ubuntu-linux:~$ useradd bob
useradd: Permission denied.
```

User [smurf] gets a permission denied error. That's because the [useradd] command is a privileged command. OK fine! Let's try installing the [terminator] package, which is a pretty cool Terminal emulator I must say:

```
smurf@ubuntu-linux:~$ apt-get install terminator
E: Could not open lock file /var/lib/dpkg/lock-frontend - open
 (13: Permission denied)
E: Unable to acquire the dpkg frontend lock (/var/lib/dpkg/lock-frontend),
are you root?
```

Again! User [smurf] is getting an error. Life is not fun without root, I hear you saying.

Granting access with sudo

User [smurf] is now very sad as he can't add user [bob] or install the [terminator] package on the system. You can use the [visudo] command to grant user [smurf] the permissions to run the two privileged commands he wants.

Run the [visudo] command as the root user:

```
root@ubuntu-linux:~# visudo
```

This will open up the file [/etc/sudoers] so you can edit it:

```
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults            env_reset
Defaults            mail_badpass
```

```
# Host alias specification
# User alias specification
# Cmnd alias specification
# User privilege specification
root               ALL=(ALL:ALL) ALL
# Members of the admin group may gain root privileges
%admin             ALL=(ALL) ALL
# Allow members of group sudo to execute any command
%sudo              ALL=(ALL:ALL) ALL
# See sudoers(5) for more information on "#include" directives:
#includedir /etc/sudoers.d
```

All the lines that begin with the hash characters are comments, so only focus on these lines:

```
root   ALL=(ALL:ALL) ALL
%admin ALL=(ALL) ALL
%sudo  ALL=(ALL:ALL) ALL
```

The first line [root ALL=(ALL:ALL) ALL] is a rule that grants user [root] the permission to run all the commands on the system.

We can now add a rule to grant user [smurf] the permission to run the [useradd] command. The syntax specification for a rule in the [/etc/sudoers] file is as follows:

```
user hosts=(user:group) commands
```

Now add the following rule to the [/etc/sudoers] file:

```
smurf    ALL=(ALL)       /usr/sbin/useradd
```

The [ALL] keyword means no restrictions. Notice that you also have to include the full path of the commands. Now, save and exit the file then switch to user [smurf]:

```
root@ubuntu-linux:~# su - smurf
smurf@ubuntu-linux:~$
```

Now precede the [useradd] command with [sudo] as follows:

```
smurf@ubuntu-linux:~$ sudo useradd bob
[sudo] password for smurf:
smurf@ubuntu-linux:~$
```

It will prompt user [smurf] for his password; enter it, and just like that! User [bob] is added:

```
smurf@ubuntu-linux:~$ id bob
uid=1005(bob) gid=1005(bob) groups=1005(bob)
smurf@ubuntu-linux:~$
```

Cool! So [smurf] can now add users to the system; however, he still can't install any packages on the system:

```
smurf@ubuntu-linux:~$ sudo apt-get install terminator
Sorry, user smurf is not allowed to execute '/usr/bin/apt-get install
terminator' as root on ubuntu-linux.
```

Now let's fix that. Switch back to the root user and run the [visudo] command to edit the [sudo] rule for user [smurf]:

```
smurf ALL=(ALL) NOPASSWD: /usr/sbin/useradd, /usr/bin/apt-get install terminator
```

Notice that I also added [NOPASSWD] so that [smurf] doesn't get prompted to enter his password. I also added the command to install the [terminator] package. Now, save and exit then switch back to user [smurf] and try to install the [terminator] package:

```
smurf@ubuntu-linux:~$ sudo apt-get install terminator
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  gsfonts-x11 java-common
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
    terminator
```

Success! Notice that the [sudo] rule grants [smurf] permission only to install the [terminator] package. He will get an error if he tries to install any other package:

```
smurf@ubuntu-linux:~$ sudo apt-get install cmatrix
Sorry, user smurf is not allowed to execute '/usr/bin/apt-get install cmatrix'
as root on ubuntu-linux.
```

User and command aliases

You can use user aliases to reference multiple users in the [/etc/sudoers] file. For example, you can create a user alias [MANAGERS] that includes [userssmurf] and [bob] as follows:

```
User_Alias MANAGERS = smurf,bob
```

You can use a command alias to group multiple commands together. For example, you can create a command alias [USER_CMDS] that includes the commands [useradd], [userdel], and [usermod]:

```
Cmnd_Alias USER_CMDS = /usr/sbin/useradd, /usr/sbin/userdel, /usr/sbin/usermod
```

Now you can use both aliases:

```
MANAGERS ALL=(ALL) USER_CMDS
```

to grant users [smurf] and [bob] the permission to run the commands [useradd], [userdel], and [usermod].

Group privileges

You can also specify groups in the [/etc/sudoers] file. The group name is preceded by the percentage character as follows:

```
%group hosts=(user:group) commands
```

The following rule will grant the [developers] group permission to install any package on the system:

```
%developers ALL=(ALL) NOPASSWD: /usr/bin/apt-get install
```

The following rule will grant the [developers] group permission to run any command on the system:

```
%developers ALL=(ALL) NOPASSWD: ALL
```

Listing user privileges

You can use the command [sudo -lU] to display a list of the [sudo] commands a user can run:

```
sudo -lU username
```

For example, you can run the command:

```
root@ubuntu-linux:~# sudo -lU smurf
Matching Defaults entries for smurf on ubuntu-linux:
    env_reset, mail_badpass

User smurf may run the following commands on ubuntu-linux:
    (ALL) NOPASSWD: /usr/sbin/useradd, /usr/bin/apt-get install terminator
```

to list all the [sudo] commands that can be run by user [smurf].

If a user is not allowed to run any [sudo] commands, the output of the command [sudo-lU] will be as follows:

```
root@ubuntu-linux:~# sudo -lU rachel
User rachel is not allowed to run sudo on ubuntu-linux.
```

visudo versus /etc/sudoers

You may have noticed that I used the command [visudo] to edit the file [/etc/sudoers], and you might ask yourself a very valid question: why not just edit the file [/etc/sudoers] directly without using [visudo]? Well, I will answer your question in a practical way.

First, run the [visudo] command and add the following line:

```
THISLINE=WRONG
```

Now try to save and exit:

```
root@ubuntu-linux:~# visudo
>>> /etc/sudoers: syntax error near line 14 <<<
What now?
Options are:
  (e)dit sudoers file again
 e(x)it without saving changes to sudoers file
  (Q)uit and save changes to sudoers file (DANGER!)
What now?
```

As you can see, the [visudo] command detects an error, and it specifies the line number where the error has occurred.

Why is this important? Well, if you saved the file with an error in it, all the [sudo] rules in [/etc/sudoers] will not work! Let's hit [Q] to save the changes and then try to list the [sudo] commands that can be run by user [smurf]:

```
What now? Q
root@ubuntu-linux:~# sudo -lU smurf
>>> /etc/sudoers: syntax error near line 14 <<<
sudo: parse error in /etc/sudoers near line 14
sudo: no valid sudoers sources found, quitting
sudo: unable to initialize policy plugin
```

We get an error, and all the [sudo] rules are now broken! Go back and run the [visudo] command to remove the line that contains the error.

If you directly edit the file [/etc/sudoers] without using the [visudo] command, it will not check for syntax errors and this may lead to catastrophic consequences, as you saw. So the rule of thumb here: always use [visudo] when editing the [/etc/sudoers] file.

Knowledge check

For the following exercises, open up your Terminal and try to solve the following tasks:

1. Add a [sudo] rule so that user [smurf] can run the [fdisk] command.
2. Add a [sudo] rule so that the [developers] group can run the [apt-get] command.
3. List all the [sudo] commands of user [smurf].