10 Useful Sudoers Configurations for Setting 'sudo' in Linux

by Aaron Kili | Published: January 6, 2017 | Last Updated: January 12, 2017

Linux Certifications - RHCSA / RHCE Certification | Ansible Automation Certification | LFCS / LFCE Certification

In Linux and other Unix-like operating systems, only the **root** user can run all commands and perform certain critical operations on the system such as install and update, remove packages, create users and groups, modify important system configuration files and so on.

However, a system administrator who assumes the role of the root user can permit other normal system users with the help of sudo command and a few configurations to run some commands as well as carry out a number of vital system operations including the ones mentioned above.

Alternatively, the system administrator can share the root user password (which is not a recommended method) so that normal system users have access to the root user account via **su** command.

sudo allows a permitted user to execute a command as root (or another user), as specified by the security policy:

- It reads and parses **/etc/sudoers**, looks up the invoking user and its permissions,
- then prompts the invoking user for a password (normally the user's password, but it can as well be the target user's password. Or it can be skipped with NOPASSWD tag),
- after that, sudo creates a child process in which it calls **setuid()** to switch to the target user
- next, it executes a shell or the command given as arguments in the child process above.

Below are ten **/etc/sudoers** file configurations to modify the behavior of **sudo** command using **Defaults** entries.

```
$ sudo cat /etc/sudoers
```

/etc/sudoers File

```
#
# 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
Defaults secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
Defaults logfile="/var/log/sudo.log"
Defaults lecture="always"
Defaults badpass_message="Password is wrong, please try again"
Defaults passwd_tries=5
Defaults insults
Defaults log_input,log_output
```

Types of Defaults Entries

```
Defaults              parameter,   parameter_list     #affect all users on any host
Defaults@Host_List    parameter,   parameter_list     #affects all users on a specific host
Defaults:User_List    parameter,   parameter_list     #affects a specific user
Defaults!Cmnd_List    parameter,   parameter_list     #affects  a specific command
Defaults>Runas_List   parameter,   parameter_list     #affects commands being run as a specific user
```

For the scope of this guide, we will zero down to the first type of **Defaults** in the forms below. Parameters may be flags, integer values, strings, or lists.

You should note that flags are implicitly boolean and can be turned off using the `'!'` operator, and lists have two additional assignment operators, `+=` (add to list) and `-=` (remove from list).

```
Defaults    parameter
OR
Defaults    parameter=value
OR
Defaults    parameter -=value
Defaults    parameter +=value
OR
Defaults    !parameter
```

1. Set a Secure PATH

This is the path used for every command run with sudo, it has two importances:

- Used when a system administrator does not trust sudo users to have a secure PATH environment variable
- To separate “root path” and “user path”, only users defined by **exempt_group** are not affected by this setting.

To set it, add the line:

```
Defaults secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin"
```

2. Enable sudo on TTY User Login Session

To enable sudo to be invoked from a real **tty** but not through methods such as **cron** or **cgi-bin** scripts, add the line:

```
Defaults  requiretty
```

3. Run Sudo Command Using a pty

A few times, attackers can run a malicious program (such as a virus or malware) using sudo, which would again fork a background process that remains on the user’s terminal device even when the main program has finished executing.

To avoid such a scenario, you can configure sudo to run other commands only from a **psuedo-pty** using the **use_pty** parameter, whether I/O logging is turned on or not as follows:

```
Defaults  use_pty
```

4. Create a Sudo Log File

By default, sudo logs through syslog(3). However, to specify a custom log file, use the logfile parameter like so:

```
Defaults  logfile="/var/log/sudo.log"
```

To log hostname and the four-digit year in the custom log file, use **log_host** and **log_year** parameters respectively as follows:

```
Defaults  log_host, log_year, logfile="/var/log/sudo.log"
```

Below is an example of a custom sudo log file:

```
aaronkilik@tecmint ~ $ cat /var/log/sudo.log
Dec 28 11:55:04 : aaronkilik : TTY=pts/0 ; PWD=/home/aaronkilik ; USER=root ;
    COMMAND=/usr/sbin/visudo
Dec 28 12:03:19 : aaronkilik : TTY=pts/0 ; PWD=/home/aaronkilik ; USER=root ;
    COMMAND=/usr/sbin/visudo
Dec 28 12:03:44 : aaronkilik : TTY=pts/0 ; PWD=/home/aaronkilik ; USER=root ;
    COMMAND=/usr/sbin/visudo
Dec 28 12:04:01 : aaronkilik : TTY=pts/0 ; PWD=/home/aaronkilik ; USER=root ;
    COMMAND=/usr/sbin/visudo
Dec 28 12:04:39 : aaronkilik : TTY=pts/1 ; PWD=/home/aaronkilik ; USER=root ;
    COMMAND=/usr/sbin/visudo
Dec 28 12:05:58 : aaronkilik : 2 incorrect password attempts ; TTY=pts/0 ;
    PWD=/home/aaronkilik ; USER=root ; COMMAND=/usr/sbin/visudo
Dec 28 12:06:58 : aaronkilik : 3 incorrect password attempts ; TTY=pts/0 ;
    PWD=/home/aaronkilik ; USER=root ; COMMAND=/usr/sbin/visudo
Dec 28 12:21:02 : aaronkilik : 1 incorrect password attempt ; TTY=pts/0 ;
    PWD=/home/aaronkilik ; USER=root ; COMMAND=/usr/sbin/visudo
Dec 28 12:46:54 : aaronkilik : TTY=pts/1 ; PWD=/home/aaronkilik ; USER=root ;
```

Create Custom Sudo Log File

5. Log Sudo Command Input/Output

The **log_input** and **log_output** parameters enable sudo to run a command in pseudo-tty and log all user input and all output sent to the screen receptively.

The default I/O log directory is **/var/log/sudo-io**, and if there is a session sequence number, it is stored in this directory. You can specify a custom directory through the **iolog_dir** parameter.

```
Defaults    log_input, log_output
```

There are some escape sequences are supported such as **%{seq}** which expands to a monotonically increasing base-36 sequence number, such as 000001, where every two digits are used to form a new directory, e.g. **00/00/01** as in the example below:

```
$ cd /var/log/sudo-io/
$ ls
$ cd  00/00/01
$ ls
$ cat log
```

```
tecmint aaronkilik # cd /var/log/sudo-io/
tecmint sudo-io # ls
00  seq
tecmint sudo-io # cd 00/00/01/
tecmint 01 # ls
log  stderr  stdin  stdout  timing  ttyin  ttyout
tecmint 01 # cat log
1483434005:aaronkilik:root::/dev/pts/2:17:67
/home/aaronkilik
/usr/bin/apt-get update
tecmint 01 #
```

Log sudo Input Output

You can view the rest of the files in that directory using the cat command.

6. Lecture Sudo Users

To lecture sudo users about password usage on the system, use the **lecture** parameter as below.

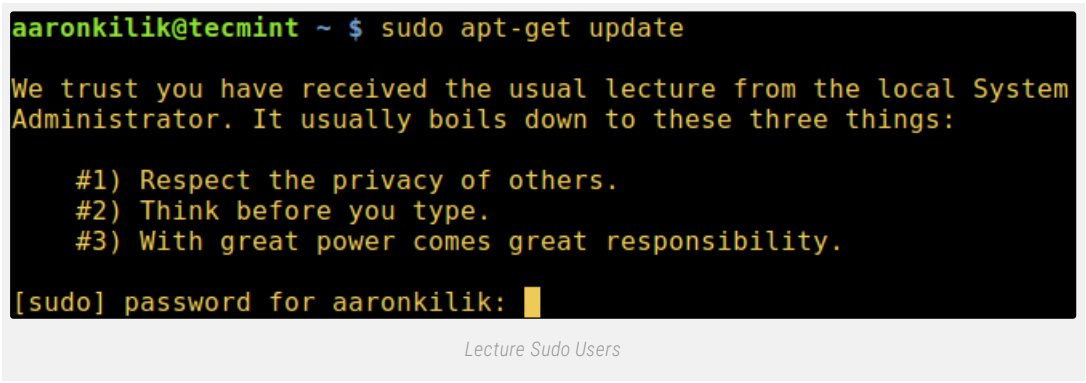
It has 3 possible values:

- always – always lecture a user.
- once – only lecture a user the first time they execute sudo command (this is used when no value is specified)
- never – never lecture the user.

```
Defaults  lecture="always"
```

Additionally, you can set a custom lecture file with the **lecture_file** parameter, type the appropriate message in the file:

```
Defaults  lecture_file="/path/to/file"
```



Lecture Sudo Users

7. Show Custom Message When You Enter Wrong sudo Password

When a user enters a wrong password, a certain message is displayed on the command line. The default message is "**sorry, try again**", you can modify the message using the **badpass_message** parameter as follows:

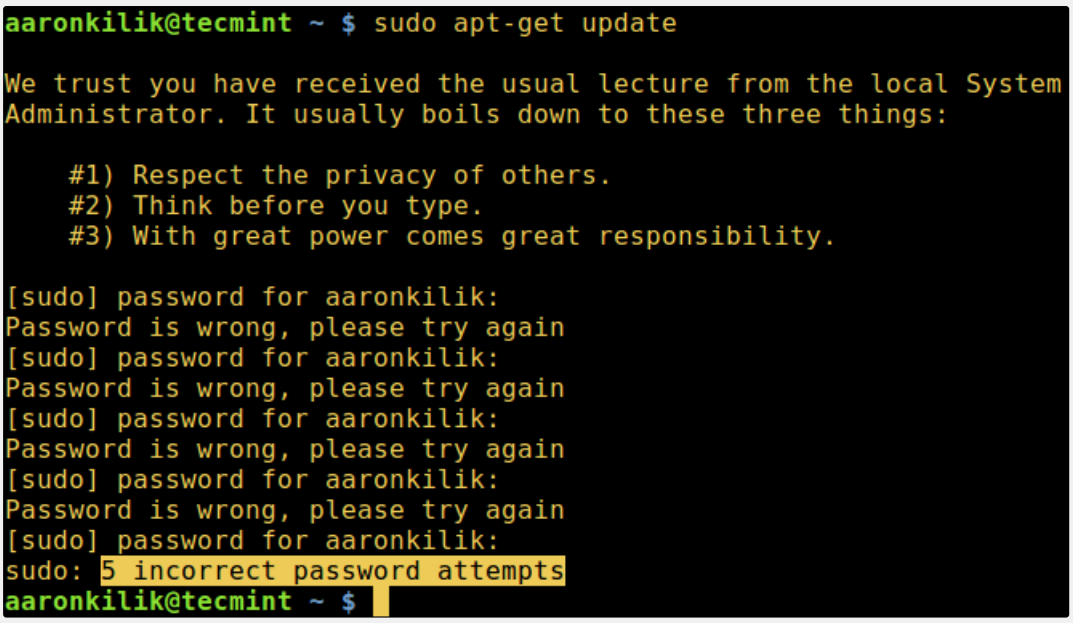
```
Defaults  badpass_message="Password is wrong, please try again"
```

8. Increase sudo Password Tries Limit

The parameter **passwd_tries** is used to specify the number of times a user can try to enter a password.

The default value is 3:

```
Defaults   passwd_tries=5
```



Increase Sudo Password Attempts

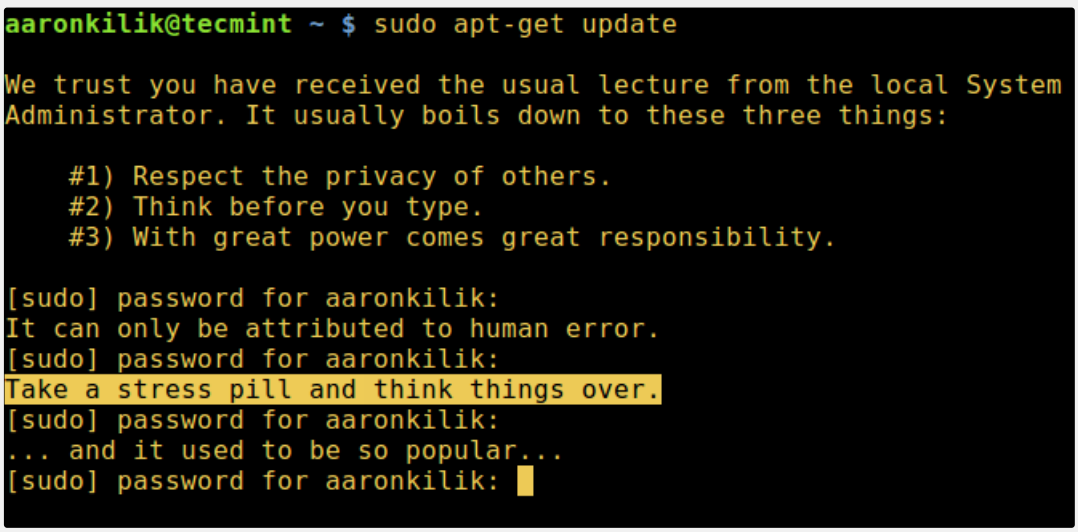
To set a password timeout (default is 5 minutes) using **passwd_timeout** parameter, add the line below:

```
Defaults   passwd_timeout=2
```

9. Let Sudo Insult You When You Enter Wrong Password

In case a user types a wrong password, sudo will display insults on the terminal with the insults parameter. This will automatically turn off the **badpass_message** parameter.

```
Defaults  insults
```



Let's Sudo Insult You When Enter Wrong Password

Read More: Let Sudo Insult You When You Enter Incorrect Password

10. Learn More Sudo Configurations

Additionally, you can learn more **sudo** command configurations by reading: Difference Between su and sudo and How to Configure sudo in Linux.

That's it! You can share other useful sudo command configurations or tricks and tips with Linux users out there via the comment section below.

Sharing is Caring...

Tags: Linux Tricks sudo commands

Aaron Kili View all Posts

Aaron Kili is a Linux and F.O.S.S enthusiast, an upcoming Linux SysAdmin, web developer, and currently a content creator for TecMint who loves working with computers and strongly believes in sharing knowledge.

Your name can also be listed here. Got a tip? Submit it here to become an TecMint author.

NEXT STORY

3 Ways to Permanently and Securely Delete 'Files and Directories' in Linux

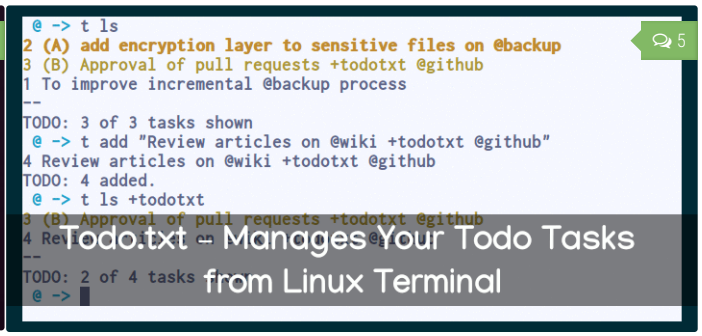
PREVIOUS STORY

12 Useful Commands For Filtering Text for Effective File Operations in Linux

YOU MAY ALSO LIKE...

How to Clear RAM Memory Cache, Buffer and Swap Space on Linux

6 JUN, 2015



Todo.txt – Manages Your Todo Tasks from Linux Terminal

27 MAR, 2019

5 RESPONSES

Comments 5 Pingbacks 0

Ryan Quezada G. March 22, 2020 at 9:42 pm

This is an excellent article. With this, I complement my knowledge about the root users and permissions.

Reply

Aaron Kili March 25, 2020 at 1:51 pm

@Ryan

Great! We are grateful that this has helped you gain more knowledge about Linux. Many thanks for the useful feedback.

Reply

Brain November 11, 2018 at 8:05 pm

Hi,

Thanks for the nice overview.

Can you help with this?

I want to mount a special source without root privileges. So I made an entry in the **/ect/sudoers** file:

username ALL = NOPASSWD: /sbin/mount.cifs, /bin/umount /mnt/folder

How can I restrict the source that I want to mount to be only one that can be mounted. Now username can mount everything.

Thanks in advance.

Bye,

Reply

Garry Garrett August 14, 2019 at 7:45 pm

I think what you may want to do is, instead of using **sudo**, add the mount to **/etc/fstab**, and include the option **"user"** (see the man page on "mount"). What this will do is allow ordinary users to mount/unmount the filesystem. That would allow ALL users to mount/unmount it. They can then say "**mount /mnt/folder**". This mount option is specific to Linux and would not work on other flavors of Unix.

Another option would be to use the automounter. You could setup a direct automount map. Then whenever a user does "**cd /mnt/folder**", it mounts. After it mounts, every 5 minutes, it half-heartedly attempts to unmount it, which will not be successful if it is still in use. Again, this would allow ALL users to mount it.

If you really want just the one user to be able to mount/unmount, then you'd need to spell out the full mount command (not **/sbin/mount.cifs**):

user ALL = NOPASSWD: /bin/mount /path-to-device /mnt/folder, /bin/umount /mnt/folder

(there might be some options you'll want to specify after "mount", e.g. "-o ro", "-t cifs", etc.). The user will then need to type the command-line exactly as it appears in sudoers (if they are not that savvy, create them an alias).

Reply

Aaron Kili August 19, 2019 at 2:16 pm

@Garry

These are practically better solutions, well explained. Am also testing them. Thanks for sharing.

Reply

GOT SOMETHING TO SAY? JOIN THE DISCUSSION.

Comment

Name *

Email *

Website

Save my name, email, and website in this browser for the next time I comment.

Notify me of followup comments via e-mail. You can also subscribe without commenting.

Post Comment

How to Add Linux Host to Nagios Monitoring Server Using NRPE Plugin

How to Install Nagios 4.4.5 on RHEL/CentOS 8/7 and Fedora 30

Install Cacti (Network Monitoring) on RHEL/CentOS 8/7 and Fedora 30

How to Install Google Chrome 75 On RHEL/CentOS 7 and Fedora 30

How to Install Ubuntu Alongside With Windows 10 or 8 in Dual-Boot

LINUX EBOOKS

- Introducing Learn Linux In One Week and Go from Zero to Hero
- RedHat RHCE/RHCSA Certification Preparation Guide
- Linux Foundations LFCS/LFCE Certification Guide
- Postfix Mail Server Setup Guide for Linux
- Ansible Setup Guide for Linux
- Django Setup Guide for Linux
- Awk Getting Started Guide for Beginners
- Citrix XenServer Setup Guide for Linux

LINUX MONITORING TOOLS

screenFetch – An Ultimate System Information Generator for Linux

How to Monitor Linux Server Security with Osquery

How to Setup Central Logging Server with Rsyslog in Linux

How to Monitor User Activity with psacct or acct Tools

How to Setup Rsyslog Client to Send Logs to Rsyslog Server in CentOS 7

LINUX INTERVIEW QUESTIONS

Practical Interview Questions and Answers on Linux Shell Scripting

25 Apache Interview Questions for Beginners and Intermediates

10 Useful Interview Questions and Answers on Linux Commands

15 Basic MySQL Interview Questions for Database Administrators

10 Interview Questions and Answers on Various Commands in Linux

OPEN SOURCE TOOLS

4 Good Open Source Log Monitoring and Management Tools for Linux

10 Best Open Source Forum Software for Linux

Best IP Address Management Tools for Linux

10 Top Open Source Artificial Intelligence Tools for Linux

7 Best Command-Line Email Clients for Linux in 2020