sudo

`sudo` is a tool to execute commands with different permissions and rights. A useful tool on a standalone machine, but in the network, everything is managed centrally by the computer group, so there are no reasons to get elevated privileges through commands like this.

So, why do we document it here? Mostly to point people to this page when they use the `sudo` command anyway.

Why do people try to use ''sudo''?

1. Cut and paste from a `HOWTO` found on the internet. Possible pitfalls:
    1. the `HOWTO` is for a different type of Linux. Anything containing `apt-get` will not work on Fedora / RedHat anyway, since apt is the Debian/Ubuntu package manager. Package names are probably also different. (this is the most frequently encountered case in our `sudo` logs)
    2. the `HOWTO` may be ancient, eg asking to upgrade the `gcc` compiler to version 3.x, whereas we have version 5.3 at the moment, so such a misplaced downgrade might actually break big parts of the system, should you have succeeded.
    3. the software is already installed. Really, we try to do a good job to install everything needed for computations in astronomy, physics and mathematics. Including (most frequent case) all prerequisites for running MPI applications (OpenMPI and MPICH); AMUSE or CASA. So try running your programs before trying to install them, or look here in our wiki for documentation on what is already installed.
    4. many install notes explain how to install a package system-wide, even if there are ways to install it just for yourself, which is probably what you want anyway (see below for tips and details)
2. Mistaken identity or location (e.g. assuming you are in a terminal on your laptop, whereas you are actually logged in to our workstations). No harm done, and here it is a very good thing that your `sudo` attempt is blocked.
3. Doing something that you can already do without elevated privileges:
    1. mounting usb disks should work when you are logged in on the console (unless, see below)
    2. rebooting a desktop is possible from the menu or the login screen without password. or if that fails: `ctrl-alt-F2` to switch to a text screen, followed by `ctrl-alt-delete` to reboot
    3. if anything related to your own files gives a permission problem, there may be a problem (e.g. disk became read-only), or it is a file permission you can fix yourself (e.g. setting `execute` permission on a newly created script). Anything downloaded from the internet will not have execute permission even if it is a valid program. `chmod +x` is your tool here, no sudo required. Anything copied from cdrom/dvd has only read permission, because it came from read-only media. `chmod +w` will make sure you can edit it.

In general, if you encounter a situation where you think `sudo` is the answer, check this list first and if you still think something needs to be done using elevated privileges, contact the helpdesk.

Software installation tips

If you want to install a software package, it often has a system-wide location as default (`/usr`, `/usr/local` or `/opt`, most of the time). But in most cases, an install location can be specified.

- `configure` scripts often have a `--prefix` option to specify the install location. Check `INSTALL` notes that come with the package, and `configure --help` often lists supported options too.
- Python source packages often come with a `setup.py`. When running the `install` phase, you can also specify a –`prefix` location, or use `--user` to install in a location in your home directory.
- Installing python packages using `pip` or `easy_install` will also work if you give it a `--prefix` or `--user` option
- a package that comes with a `Makefile` often has a `install` target or some other location where you can set the destination. No real standards here, you may have to read the documentation and/or the makefile.

Problems with USB disks

In our setup, USB disks are normally mounted automatically when inserted, if you are logged in on the system locally. So it will not work when you are only logged in from your laptop; there is no way to distinguish a login from a laptop on the same desk, from a login from a laptop at the other side of the world, and you can imagine that there will be problems if someone logged in remotely can mount or unmount and access your usb disk without your knowledge. Sometimes this fails. Some common reasons:

- Updates sometimes cause a mismatch between the running software (e.g. the file manager) and underlying tools that are needed to mount the disk. You should in those cases already have seen a message about updates and a required reboot, but if you disable those messages, or you haven't logged out in a long time, you may have missed the message. Anyway, a reboot will fix this.
- A problem with the USB disk. Linux cannot do a proper file system check on disks formatted for windows, and if you connect this usb device to your laptop, you will probably be prompted to do a check and repair. Once that is done, the device should work on Linux as well
- Or perhaps the device is not usable at all, eg because it is unformatted, or formatted in a way that Linux doesn't know (e.g. MacOS time machine). No `sudo` will help you there, except when you want to reformat the disk.

From:
https://helpdesk.lorentz.leidenuniv.nl/wiki/ - **Computer Documentation Wiki**

Permanent link:
https://helpdesk.lorentz.leidenuniv.nl/wiki/doku.php?id=linux:sudo&rev=1498642639

Last update: **2017/06/28 09:37**