# The KDE su handbook

**Geert Jansen**

The KDE su handbook

# Contents

**1 Introduction** **5**

**2 Using KDE su** **6**

**3 Internals** **8**

3.1 X authentication . . . . . 8

3.2 Interface to **su** . . . . . 8

3.3 Password Checking . . . . . 8

3.4 Password Keeping . . . . . 9

**4 Author** **10**

**Abstract**

KDE su is a graphical front end for the UNIX® **su** command.

# Chapter 1

# Introduction

Welcome to KDE su! KDE su is a graphical front end for the UNIX® **su** command for the K Desktop Environment. It allows you to run a program as different user by supplying the password for that user. KDE su is an unprivileged program; it uses the system's **su**.

KDE su has one additional feature: it can remember passwords for you. If you are using this feature, you only need to enter the password once for each command. See Section 3.4 for more information on this and a security analysis.

This program is meant to be started from the command line or from `.desktop` files. Although it asks for the `root` password using a GUI dialog, I consider it to be more of a command line <-> GUI glue instead of a pure GUI program.

Since **kdesu** is no longer installed in **`$(kf5-config --prefix)`**/bin but in **`kf5-config --path libexec`** and therefore not in your `Path`, you have to use **`$(kf5-config --path libexec)kdesu`** to launch **kdesu**.

# Chapter 2

# Using KDE su

Usage of KDE su is easy. The syntax is like this:

`kdesu` [`-c`*command*] [`-d`] [`-f`*file*] [`-i`*icon name*] [`-n`] [`-p`*priority*] [`-r`] [`-s`] [`-t`] [`-u`*user*] [`--noignorebutton`] [`--attach`*winid*]

`kdesu` [KDE Generic Options] [Qt™ Generic Options]

The command line options are explained below.

**`-c command`**

This specifies the command to run as root. It has to be passed in one argument. So if, for example, you want to start a new file manager, you would enter at the prompt: **`$(kf5-config --path libexec)kdesu -c Dolphin`**

**`-d`**

Show debug information.

**`-f file`**

This option allow efficient use of KDE su in `.desktop` files. It tells KDE su to examine the file specified by *`file`*. If this file is writable by the current user, KDE su will execute the command as the current user. If it is not writable, the command is executed as user *`user`* (defaults to root).

*`file`* is evaluated like this: if *`file`* starts with a `/` , it is taken as an absolute filename. Otherwise, it is taken as the name of a global KDE configuration file.

**`-i`** ***`icon name`***

Specify icon to use in the password dialog. You may specify just the name, without any extension.

For instance to run Konqueror in filemanager mode and show the Konqueror icon in the password dialog:

```
$(kf5-config --path libexec)kdesu  -i konqueror
-c "konqueror --profile filemanagement"
```

**`-n`**

Do not keep the password. This disables the **keep password** checkbox in the password dialog.

**`-p`** ***`priority`***

Set priority value. The priority is an arbitrary number between 0 and 100, where 100 means highest priority, and 0 means lowest. The default is 50.

**-r**

Use realtime scheduling.

**-s**

Stop the kdesu daemon. See Section 3.4.

**-t**

Enable terminal output. This disables password keeping. This is largely for debugging purposes; if you want to run a console mode app, use the standard **su** instead.

**-u** ***user***

While the most common use for KDE su is to run a command as the superuser, you can supply any user name and the appropriate password.

# Chapter 3

# Internals

## 3.1 X authentication

The program you execute will run under the root user id and will generally have no authority to access your X display. KDE su gets around this by adding an authentication cookie for your display to a temporary `.Xauthority` file. After the command exits, this file is removed.

If you don't use X cookies, you are on your own. KDE su will detect this and will not add a cookie but you will have to make sure that root is allowed to access to your display.

## 3.2 Interface to su

KDE su uses the sytem's **su** for acquiring priviliges. In this section, I explain the details of how KDE su does this.

Because some **su** implementations (i.e. the one from Red Hat®) don't want to read the password from `stdin`, KDE su creates a pty/tty pair and executes **su** with its standard filedescriptors connected to the tty.

To execute the command the user selected, rather than an interactive shell, KDE su uses the `-c` argument with **su**. This argument is understood by every shell that I know of so it should work portably. **su** passes this `-c` argument to the target user's shell, and the shell executes the program. Example command: **`su root -c the_program`**.

Instead of executing the user command directly with **su**, KDE su executes a little stub program called kdesu_stub. This stub (running as the target user), requests some information from KDE su over the pty/tty channel (the stub's stdin and stdout) and then executes the user's program. The information passed over is: the X display, an X authentication cookie (if available), the `PATH` and the command to run. The reason why a stub program is used is that the X cookie is private information and therefore cannot be passed on the command line.

## 3.3 Password Checking

KDE su will check the password you entered and gives an error message if it is not correct. The checking is done by executing a test program: `/bin/true`. If this succeeds, the password is assumed to be correct.

## 3.4 Password Keeping

For your comfort, KDE su implements a 'keep password' feature. If you are interested in security, you should read this paragraph.

Allowing KDE su to remember passwords opens up a (small) security hole in your system. Obviously, KDE su does not allow anybody but your user id to use the passwords, but, if done without caution, this would lowers `root`'s security level to that of a normal user (you). A hacker who breaks into your account, would get `root` access. KDE su tries to prevent this. The security scheme it uses is, in my opinion at least, reasonably safe and is explained here.

KDE su uses a daemon, called kdesud. The daemon listens to a UNIX® socket in `/tmp` for commands. The mode of the socket is 0600 so that only your user id can connect to it. If password keeping is enabled, KDE su executes commands through this daemon. It writes the command and `root`'s password to the socket and the daemon executes the command using **su**, as describe before. After this, the command and the password are not thrown away. Instead, they are kept for a specified amount of time. This is the timeout value from in the control module. If another request for the same command is coming within this time period, the client does not have to supply the password. To keep hackers who broke into your account from stealing passwords from the daemon (for example, by attaching a debugger), the daemon is installed set-group-id nogroup. This should prevent all normal users (including you) from getting passwords from the kdesud process. Also, the daemon sets the `DISPLAY` environment variable to the value it had when it was started. The only thing a hacker can do is execute an application on your display.

One weak spot in this scheme is that the programs you execute are probably not written with security in mind (like setuid `root` programs). This means that they might have buffer overruns or other problems and a hacker could exploit those.

The use of the password keeping feature is a tradeoff between security and comfort. I encourage you to think it over and decide for yourself if you want to use it or not.

# Chapter 4

# Author

KDE su

Copyright 2000 Geert Jansen

KDE su is written by Geert Jansen. It is somewhat based on Pietro Iglio's KDE su, version 0.3. Pietro and I agreed that I will maintain this program in the future.

The author can be reached through email at g.t.jansen@stud.tue.nl. Please report any bugs you find to me so that I can fix them. If you have a suggestion, feel free to contact me.

This documentation is licensed under the terms of the GNU Free Documentation License.

This program is licensed under the terms of the Artistic License.