

sudo

Easily prefix your current or previous commands with `sudo` by pressing `esc` twice.

To use it, add `sudo` to the plugins array in your `zshrc` file:

```
plugins=(... sudo)
```

Usage

Current typed commands

Say you have typed a long command and forgot to add `sudo` in front:

```
$ apt-get install build-essential
```

By pressing the `esc` key twice, you will have the same command with `sudo` prefixed without typing:

```
$ sudo apt-get install build-essential
```

The same happens for editing files with your default editor (defined in `$SUDO_EDITOR`, `$VISUAL` or `$EDITOR`, in that order):

If the editor defined were `vim`:

```
$ vim /etc/hosts
```

By pressing the `esc` key twice, you will have the same command with `sudo -e` instead of the editor, that would open that editor with root privileges:

```
$ sudo -e /etc/hosts
```

Previous executed commands

Say you want to delete a system file and denied:

```
$ rm some-system-file.txt
-su: some-system-file.txt: Permission denied
$
```

By pressing the `esc` key twice, you will have the same command with `sudo` prefixed without typing:

```
$ rm some-system-file.txt
-su: some-system-file.txt: Permission denied
$ sudo rm some-system-file.txt
Password:
$
```

The same happens for file editing, as told before.

Key binding

By default, the `sudo` plugin uses `EscEsc` as the trigger. If you want to change it, you can use the `bindkey` command to bind it to a different key:

```
bindkey -M emacs '<seq>' sudo-command-line
bindkey -M vicmd '<seq>' sudo-command-line
bindkey -M viins '<seq>' sudo-command-line
```

where `<seq>` is the sequence you want to use. You can find the keyboard sequence by running `cat` and pressing the keyboard combination you want to use.