

# Using Touch ID on Macbook Pro for sudo Authentication

Full credit goes to [Cabel Sasser \(@cabel\)](#) on this one for sharing the original tip. I'm simply sharing it here and showing the process to prove the awesomeness of this capability.

Pro MacBook Pro Tip: have a Touch Bar with Touch ID? If you edit `/etc/pam.d/sudo` and add the following line to the top...

```
auth sufficient pam_tid.so
```

...you can now use your fingerprint to sudo!

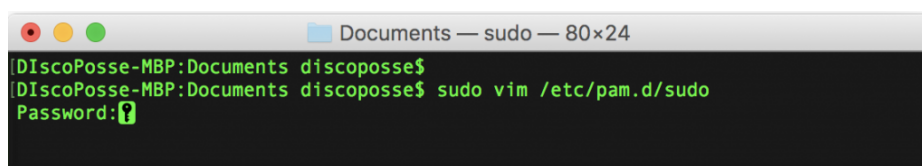
— Cabel Sasser (@cabel) [November 16, 2017](#)

If you run a MacBook Pro with the Touch ID option, you have already discovered the speed at which you can authenticate for a number of GUI-driven products. Running `sudo` in the command line does not give you that luxury, usually.

By making a small change

First, you have to edit the `/etc/pam.d/sudo` file with your editor of choice. It's a read only file and you need admin privileges to do so. Oh the irony!

I'm going to use `sudo vim /etc/pam.d/sudo` to open up the file. This prompts me for credentials in the terminal session, as it should:

A terminal window titled "Documents — sudo — 80x24" showing a shell prompt. The user enters the command "sudo vim /etc/pam.d/sudo". The prompt changes to "Password:" and a cursor is visible, indicating the password prompt is active.

```
DiscoPosse-MBP:Documents discoposse$  
DiscoPosse-MBP:Documents discoposse$ sudo vim /etc/pam.d/sudo  
Password:█
```

Add the following to the first line in the file after the comment:

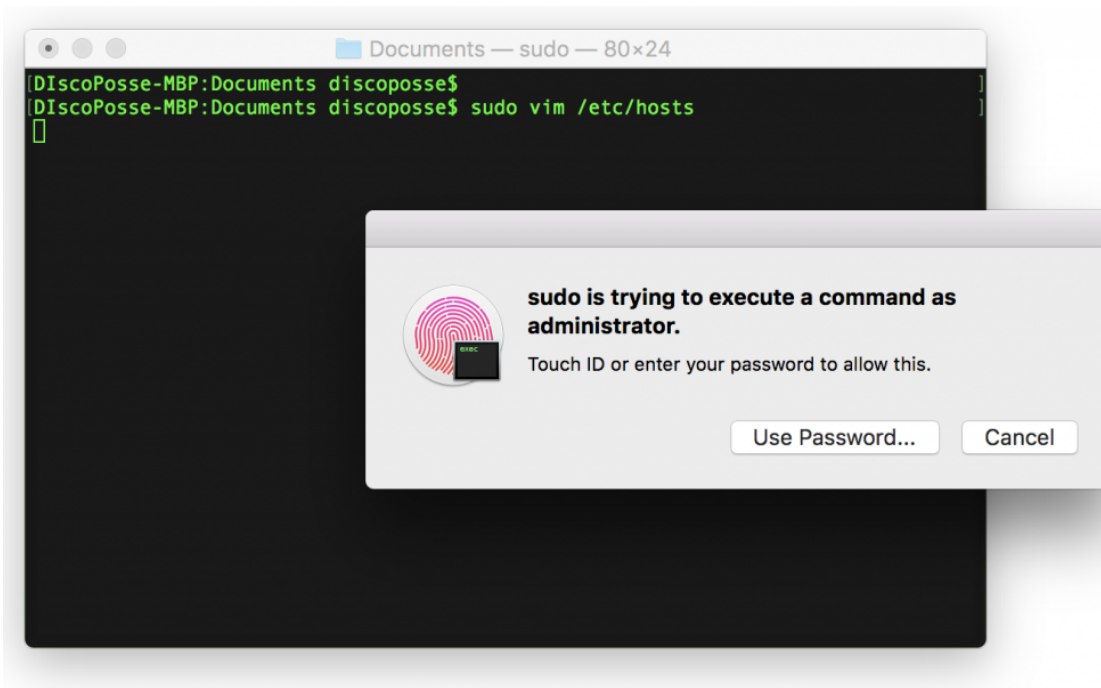
```
auth sufficient pam_tid.so
```

You can space it out for consistency with the other lines:

```
Documents — vim — sudo — 80x24
# sudo: auth account password session
auth    sufficient    pam_tid.so
auth    required        pam_opendirectory.so
account required        pam_permit.so
password required       pam_deny.so
session required      pam_permit.so
~
~
~
~
~
~
~
~
~
~
-- INSERT --
```

Save the file. It's read-only, so I have to use `w!` to save, and then exit back to the shell and close your terminal.

Launch a new terminal session so that you have no cached sudo session credentials and try a new sudo command such as `sudo vim /etc/hosts` and watch the magic happen:



This should be a nice time saver for you, especially when you use complex passwords...like you should ☐