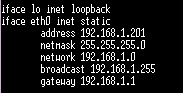
SECURING RASPBERRY PI using SSH

**Set static address**

My internet provider is Verizon FIOS and the tech support told me that you can assign static address above 100, then it is safe. DHCP will be using lower addresses.

Edit /etc/network/interfaces and modify “iface eth0 inet **dhcp**” to the following:

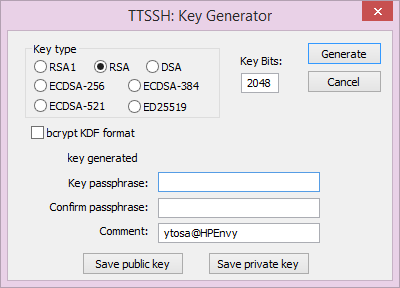


where address 192.168.1.201 is the one I decided to use for Raspberry PI and 192.168.1.1 is the router address.

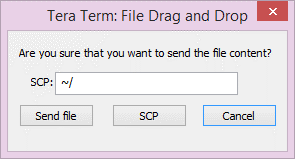
**Install SSH Terminal on Windows 8.1**

Install “Tera Term”( <http://ttssh2.sourceforge.jp/index.html.en> ) and connect it to Raspberry PI.dfdf

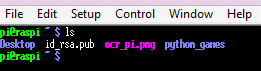
“Tera Term” has the SSH Key Generator (SetUp -> SSH Keygnerator). You have to remember the passphrase and save both public and private key to some directory.



Drag and drop “id\_rsa.pub” to “Tera Term” and use scp (press “SCP”).



It is copied to the home directory of user “pi”.



Issue the following command to create a .ssh directory and change permission:

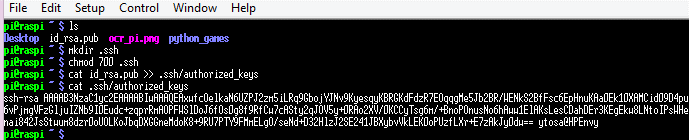
$ mkdir .ssh

$ chmod 700 .ssh

Then write the public key to .ssh/authorized\_keys by

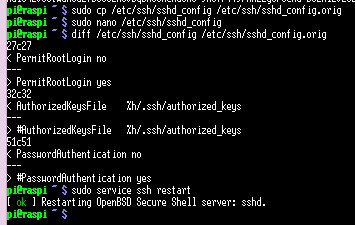
$ cat id\_rsa.pub >> .ssh/authorized\_keys

Here is the command line screen image:

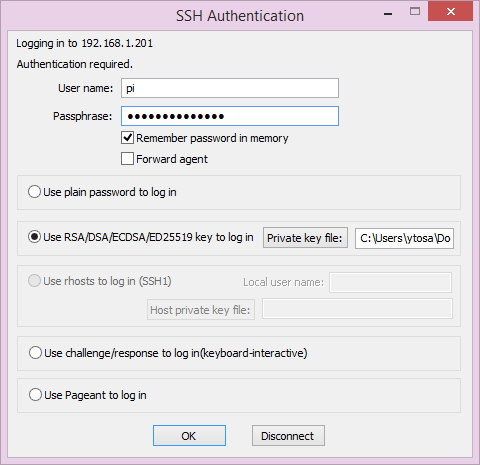


Now we forbid password login.

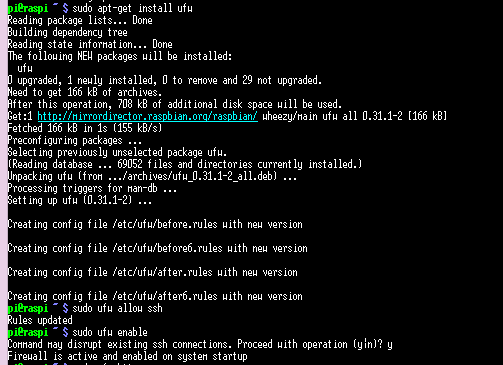
Edit /etc/ssh/sshd\_config and modify “**PermitRootLogin no”** and “**PasswordAuthentication no**”. Use “nano” to edit the file. First save the original so that you can compare after and before. Note that NANO uses the commands: write = Ctrl o, exit = Ctrl x. After confirmation, you restart sshd.



Now you can login to Raspberry PI only with the public key login. Now you have to **use the passphrase** you used to create private/public key and **specify the private key** you have:



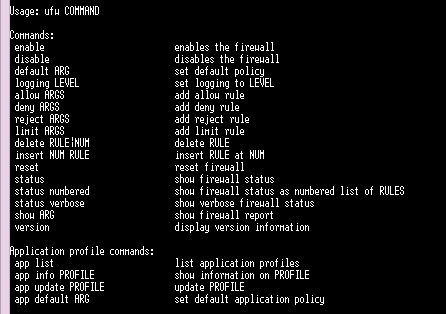
We install the firewall “ufw” and allow only ssh and http. After the install, first you allow ssh and then enable ufw.



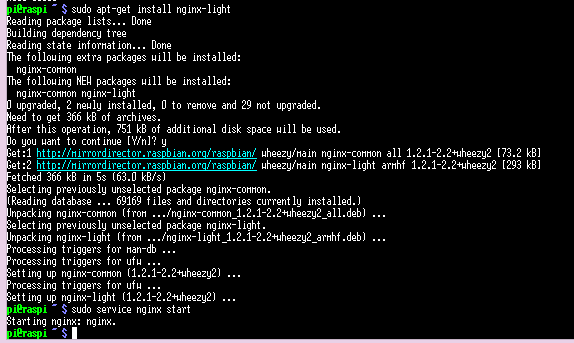
We also allow http.



Here is the command list for ufw.



Installing web server named “nginx-light” (called EngineX) and you start it.



You confirm that it is running by using the explicit Raspberry PI IP address (in my case 192.168.1.201).

