

Forum:[Hack Forums](#)

Thread: [\[Tutorial\]Setting up Private SSH keys \[Tutorial\]](#)

Loki123 - January 18, 2012, 6:18 am

Hi, im Loki123 and in this tutorial I am going to show you how you can secure SSH on your server with private SSH key so no one that does not have a key that is accepted by the server can log in.

I will show you how to generate a key and transfer it over to the server and the configuration of the server. All that will be shown in this TUT will be from a OpenSuSE Desktop computer and the server is Ubuntu 11.10 Desktop. (Server is desktop because i don't know how to configure Wifi in terminal in server edition ?)

help.Ubuntu.com Wrote:If your SSH server is visible over the Internet, you should use public key authentication instead of passwords if at all possible. If you don't think it's important, try logging all of the malicious login attempts you get for the next week. My computer - a perfectly ordinary desktop PC - had over 4,000 attempts to guess my password and almost 2,500 break-in attempts in the last week alone. How many thousand random guesses do you think it will take before an attacker stumbles across your password?

With public key authentication, every computer has a public and a private "key" (a large number with particular mathematical properties). The private key is kept on the computer you log in from, while the public key is stored on the .ssh/authorized_keys file on all the computers you want to log in to. When you log in to a computer, the SSH server uses the public key to "lock" messages in a way that can only be "unlocked" by your private key - this means that even the most resourceful attacker can't snoop on, or interfere with, your session. As an extra security measure, most SSH programs store the private key in a passphrase-protected format, so that if your computer is stolen or broken in to, you should have enough time to disable your old public key before they break the passphrase and start using your key. [Wikipedia has a more detailed explanation of how keys work.](#)

1. How to generate a SSH key !

You generate the key in your Desktop computer.

Its really simple to generate a ssh key, just open up your terminal, and type in this :

Code `ssh-keygen -t rsa -b 4096`This will generate a 4096 bit rsa private key that you will in the next step copy over to the server.

2. Transfer the key over to the server

Now, here you will copy the 4096 bit RSA key we generated earlier over to the server. to do that we simply go to our terminal and use this line :

Code `ssh-copy-id USER@HOST`no more you need to do on the desktop machine.

3. configuring the SSH server.

Now we SSH over to our server by doing this :

Code `ssh USER@Host`there you may be prompted with a passphrase for the key you generated, just insert that one and you will be able to log in :)

then we need to find our SSH configure files, usually they are in /etc/ssh but its difference after what dist you are using. Mine is /etc/ssh/sshd_config

So I will edit that one with NANO (I like it best but you can use any one you like)

Now remember, my server is Ubuntu, therefore I use `sudo` COMMAND some of you would use `su` to become root.

Code `sudo nano /etc/ssh/sshd_config` Now make sure these lines are in that file and are UNCOMMENTED !!!

Quote: `PubkeyAuthentication yes`

`RSAAuthentication yes`

Error Denied (public key).

If you did everything above right but still getting this error, do the following lines in terminal

Code `chmod go-w ~/`

`chmod 700 ~/.ssh`

`chmod 600 ~/.ssh/authorized_keys`

Encrypted home directory

If you have an encrypted home directory, SSH cannot access your `authorized_keys` file because it is inside your encrypted home directory and won't be available until after you are authenticated.

Therefore, SSH will default to password authentication.

To solve this, create a folder outside your home named `/etc/ssh/<username>` (replace "`<username>`" with your actual username). This directory should have 755 permissions and be owned by the user. Move the `authorized_keys` file into it. The `authorized_keys` file should have 644 permissions and be owned by the user.

Edit your `sshd_config` file and add this/make sure it is there and uncommented:

Code `AuthorizedKeysFile /etc/ssh/%u/authorized_keys`

then restart your SSH service ?

Code `sudo service ssh restart`

and you have set up SSH keys and now you should be able to log in to your SSH server WITH OUT any password, but any one that will try to log in to that SSH server with out the key you generated earlier will be denied ?

Loki123

Sobranye - January 18, 2012, 10:37 am

Nice tutorial, thanks for the share.

Tirus - December 8, 2014, 12:31 pm

Such a little response. Nice tutorial, thank you very much for sharing this with us.

Loki123 - December 8, 2014, 12:33 pm

(Dec 8, 2014 01:31 PM)Tirus Wrote: [\[CITE\]](#) Such a little response. Nice tutorial, thank you very much for sharing this with us.

Thanks.

I feel like I should add to it though, How to do it with Windows machine also.

cmind - December 21, 2014, 3:35 pm

thank you i really found this post really helpfull i think i will use these advices

skyn3t - December 21, 2014, 6:25 pm

Very good tutorial that was needed for this forum section. IMO this is one of the first things you NEED to do after a fresh install.

NoMemes - December 21, 2014, 10:06 pm

Heh, nice tut Loki. This guy knows what he is talking about.

Tirus - December 22, 2014, 12:14 am

(Dec 21, 2014 07:25 PM)skyn3t Wrote: [\[CITE\]](#) Very good tutorial that was needed for this forum section. IMO this is one of the first things you NEED to do after a fresh install.

What about doing a system upgrade and update? Security isn't needed first, stability is needed first. You can have all the security you want, but is your Linux distribution stable?