SSH Public Key authentication on Debian “Sarge”

written by: Axel Werner (mail@awerner.homeip.net)

a Step by Step HowTo to make the “PuttY – a free SSH Client Suite for Windows” work with Debian SSHd

Version/date: 20.2.2004

Abstract

This Document shows and explains the Steps to make PuttY, a FREE SSH Client Suite for Windows work with your SSH-Daemon on a DEBIAN SARGE Installation using PublicKexy Authentication insted of Logonname/Password combinations. With the information provided within this document you should also be able to get things fixed if you want to configure your Linux SSH Clients to work with PublicKey Authentication on SSH. So i guess its worth to read this anyway.

Please excuse my very bad English, typos and all the crap i typed in here. I am not a native english speaker neither well familiar with english. However i wanted this manual to be international readable. If you found any Typos, false grammar or any sort of bullshit i wrote, please help me fixing this manual and send in corrections that i can paste into this document. See my email Adress at the footer of each page.

If you need help on this manual or you have questions, dont hasitate to mail me. I am not a Linux specialist at all, but ill try my best to you out. Anyway the best way would be to contact one of the good Debian Forums or Debian Newsgroups around! There are lots and lots of good people around.

Preparing your Installation

I assume that you know how to install a plain Debian Sarge platform to start up with. The SSH-Daemon should already be installed and running and you also should be aleady able to logon to your linux box with any SSH client using SSH2 protocol. This is our BASE Installation where we can start. If your System doesnt seem to run a properly configured SSHD please consult your manuals or debian forums. A good start for german debian users is www.debianforum.de !!

Understanding Public Key Encryption and Authentication on SSH

Encryption, Authentication, Signature and all that stuff is pretty complicating but important to understand. However you dont need to understand all the Detail, but some basics are required. A pretty good and quick start is the Putty Manual i think. So make sure to read the Chapter „*8.1 Public key authentication - an introduction*„ of the Putty Documentation Homepage at http://www.chiark.greenend.org.uk/~sgtatham/putty/docs.html. This will help you A LOT!

Now that you understand the basics of Public-Key Encryption stuff there is one Important point that is not covered by lots of other manuals. The KEY Format! As it seems to be, there are several Versions and Branches of SSH developements. There is the commercial SSH.COM thing, the OpenSSH group, PuttY SSH Client Suite and many more. While the SSH Protocol itself seems to be a common Standard that is defined in some RFCs, the Way the Private and Public Key-Files must look like seems not to be that clear and differs from Suite to Suite. So my experiments showed that you can not use an PuttyGen generated Key-Pair directly with an Debian-Sarge OpenSSH Suite and vice versa. On OpenSSH used with a Debian Sarge the tool to generate Key-Pairs is „ssh-keygen“. So if you work one day with another SSH suites you might need to check the key-format and convert it to the right format.

This is the Point why i though this manual is nessesary! It took me about 8hrs to get those few things to work and i hated it :) - so i hope with this manual you can manage it within an hour or even less.

Configure the SSHd to enable PublicKey Authentication

Use an texteditor of your choice to modify your SSHd configuration. Make sure the following Parameters are set to those shown. Dont mess up your config file! Make a backup copy of it before you start modifying. ONLY change or add the parameters! Dont delete anything else from the file.

/etc/ssh/sshd_config

```
1. RSAAuthentication no
2. PubkeyAuthentication yes
3. AuthorizedKeysFile      %h/.ssh/authorized_keys
```

Explainations:

1. this line is for SSH Protocol Version 1 only. SSH1 should not be used anymore for security reasons so we disable the use of PublicKeys Authentication for that protocol version.
2. This line will enable publicKey authentication for the SSH2 protocol and is required.
3. Enable this line to tell the SSHd where it will find the Users Public-Key

Generate your Key-Pair (Private & Public Key)

I told you already that there are multiple Key-File Formats around. So i need to tell you that i was not able to make my PuttyGen created Keys work with the OpenSSH Package that comes with Debian Sarge. And i still dont know how to make those work toghether. But the other way around it will work. And so i will show you now how to make a working key-pair for your linux AND your PuttY SSH.

1. On your Debian Sarge Linux Box logon as the user you want to make a Key-Pair for. For Example „awerner“.
2. Enter the following command like:
   `ssh-keygen -b 1024 -t rsa`
   This line will create a KeyPair with 1024bit Keylength and RSA encryption method. You should not use DSA because its lack of security!
3. The Programm will ask you now where to save your key files. Just press Enter to use the defaults that are quite ok. This means ssh-keygen will create TWO (of course!) Keyfiles named **„~/.ssh/id_rsa“** and **„~/.ssh/id_rsa.pub“** within your Homedirectory. More Details later.
4. Now it asks your about a „Passphrase“ - Now..whats that? A Passphrase is like a Password. Its purpose is just the same. But a Passphrase can be more complex and bigger/longer and can contain special characters and whitespaces if u like. So this „Password“ is been used to protect one of your Key-Files (private Key) so nobody or nothing can use your keyfile without your permission. If you like you can enter a passphrase here, but you also can add it later and keep it empty now. I recomment to keep it empty now! Thats easier. So Press Enter twice to skip passphrase.
5. You are done! Your Keypair is created and saved in separate plain-text files. Those Files are:

~/.ssh/id_rsa	Your Private Key – Keep this file locked and never give it away! Its like your personal ID Card.
~/.ssh/id_rsa.pub	Your Public Key – This part you can give away or spread around. It does not contain any secrets or critical stuff. But its makes you or the owner able to verify your identity.

If you wonder what they look like just view them with „cat filename“ or whatever your prefer. There is nothing „magic“ with it.

Convert Key-Format and distribute public Key

At this Step we bring your Keys „to life“. Therefor we need to do a few more things.

1. Distribute your Public-Key file to all your Linux Servers you want to logon with it
2. convert your private keyfile to a format that PuttY SSH Client understands

Distribute your public Key

To logon to your Linux Server using SSH with Public-Key authentication you will need to distribute your public key to all Systems that are required to verify your identity. To do this just copy your public keyfile „**~/.ssh/id_rsa.pub**“ to „**~/.ssh/authorized_keys**“ on any System you would like to logon. You can do that with FTP, SCP or by Floppy-Disk. Whatever you prefer. Make sure only the Owner/User itself does have READ permissions on the file! So do a „chmod 600 ~/.ssh/authorized_keys“.

Remember the String „**~/.ssh/authorized_keys**“ from your SSHd config file? It is where your SSHd will look up for a users public key to authenticate!! Thats why we copied it here.

Convert your private keyfile into a PuttY-usable format

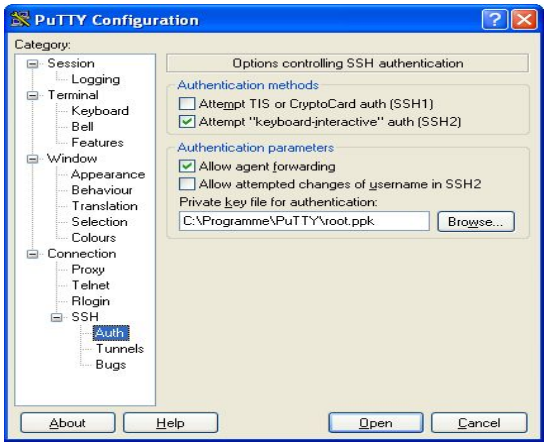
Again, there is somehow a problem with the key-format if you try to use your private keyfile directly within PuttY. We need to convert it to a Putty-Format private Key first. But no fear. Its a snap if you know where and what.

1. Take your private Keyfile from your Linux Server at **~/.ssh/id_rsa** and copy it over to your Windows PuttY machine. The PuttY Programm folder would be a cool place for that.
2. Rename the file to something more expressive, like „awerner-ssh-rsa-private-key.ppk“
3. Now start your PuttyGen.exe and „load“ this private key using the Menus „File / Load Private Key „ - PuttyGen will report in that it sucessfully imported an „foreign“ OpenSSH private key. Click OK to close the note.
4. Now click the „Save Private Key“ button and name the thing again something usefull like „awerner-ssh-rsa-privatekey-putty.ppk“ or whatever you like to. The extention PPK is for putty to recognize its private keys.You now successfully converted an OpenSSH Kexyfile into a PuttY Keyfile! Bravo!
5. Save your Putty-Private-Keyfile at a safe place AND on your Putty-Client Maschine where you want to make SSH connections from. Dont give away those private Key files! They are the PASSPORT and Entrance to your Linux boxes now!!

Configure PuttY SSH Client to allow PublicKey Authentication

We are not at your Windows Client Machine with Putty SSH Suite installed. You already copied your putty-converted private key to the same local machine. So we can now configure Putty-SSH Client itself.

1. Start Putty.exe
2. create a „stored session“, a connection setup within putty as you would do it with a normal SSH logon session. Dont forget to save this session and give it a name.
3. Load your session and switch to the configuration page „Connection / SSH / Auth “. At „Private Key File for Authentication“ click „Browse“ and fetch your previously saved putty-converted-privatekey.ppk. It should look like this screenshot:



4. Switch Back to Config-Page „Session“ and SAVE your altered Session Setup.

We are done!

Now start your preconfigured SSH Session and see that you can logon to your linux box without entering a password! All you still need to provide is the username you would like to logon with. But this you will usualy give putty as a command like option or add it to the host adress like „user@hostname“ so it wont ask you again as who you would like to logon. This is the same as on linux when you go and use „ssh username@hostname“.

If you have entered a „Passphrase“ to protect your private key once, you will be asked any time you connect to your linux host for that passphrase again. You may now think there is no benefit if i am again required to enter a password or passphrase. Yes and no! Its not that convenient as without a passphrase, but its a lot secure! A simple password is easy to guess or to spy out. Once the hacker got your logon name (which is no secret!) and passwort he could logon as you would do. In a public key authentication configuration you cannot easily logon even somebody gots your passphrase! He also needs your private-keyfile. And Vice Versa! But there is a way to make it more convenient for you if you like to protect your private key with a passphrase. See putty or openSSH manuals for „ssh agent“. This will help a lot!

Now i hope you successfully made it here and that this manual could help.
Greets from Germany!
Yours
Axel Werner