SSH Authentication Agents and Secure Password-less Logins

In conventional password authentication, you prove you are who you claim to be by proving that you know the correct password. The only way to prove you know the password is to tell the server what you think the password is. This means that if the server has been hacked, or spoofed, an attacker can learn your password.

Public key authentication solves this problem. You generate a key pair, consisting of a public key (which everybody is allowed to know) and a private key (which you keep secret and do not give to anybody). The private key is able to generate signatures. A signature created using your private key cannot be forged by anybody who does not have that key; but anybody who has your public key can verify that a particular signature is genuine.

So you generate a key pair on your own computer, and you copy the public key to the server. Then, when the server asks you to prove who you are, SSH can generate a signature using your private key. The server can verify that signature (since it has your public key) and allow you to log in. Now if the server is hacked or spoofed, the attacker does not gain your private key or password; they only gain one signature. And signatures cannot be re-used, so they have gained nothing.

There is a problem with this: if your private key is stored unprotected on your own computer, then anybody who gains access to that will be able to generate signatures as if they were you. So they will be able to log in to your server under your account. For this reason, your private key is usually encrypted when it is stored on your local machine, using a passphrase of your choice. In order to generate a signature, PuTTY must decrypt the key, so you have to type your passphrase.

This can make public-key authentication less convenient than password authentication: every time you log in to the server, instead of typing a short password, you have to type a longer passphrase. One solution to this is to use an authentication agent, a separate program which holds decrypted private keys and generates signatures on request. When you begin a Windows session, you start Pageant and load your private key into it (typing your passphrase once). For the rest of your session, you can start PuTTY any number of times and Pageant will automatically generate signatures without you having to do anything. When you close your Windows session, Pageant shuts down, without ever having stored your decrypted private key on disk. Many people feel this is a good compromise between security and convenience.

(text taken from here)

Main advantage of authentication agents is not having to type or remember passwords for every system you need to login, quite a time-saver if you're using SSH on a daily basis. Additionaly, passwords can be long and cryptic – in other words, very secure.

Detailed setup instruction for Unix systems. Recommended read for users on all operating systems.

http://www.sshkeychain.org/mirrors/SSH-with-Keys-HOWTO/

SSHKeychain is a nice Mac OS X authentication client which stores private keys in the user's Keychain:

http://www.sshkeychain.org/

Windows users can use PuTTY suite of tools, containing PuTTYgen (SSH key generator) and Pageant (authentication agent). Documentation on using Pageant.

Revision #1
Created Fri, Aug 24, 2007 4:42 PM by Jovcic, Slobodan
Updated Fri, Aug 24, 2007 4:42 PM by Jovcic, Slobodan