
 <http://www.cs.umd.edu/~arun/misc/ssh.html>

99 captures

6 Feb 02 - 21 Jun 09



MAI JUN JUL

21

2007 2009 2010

[Close](#) [Help](#)

SSH without password: micro HOWTO - by Arunchandar Vasan

ssh is a secure clone of rsh with RSA encryption based authentication. This micro HOWTO tells you how to use ssh without having to type in your password everytime you use ssh. I had to RTFM, and I hope this will be googled for others to use.

0. The basis of using ssh without typing your password is public key based authentication. You need to generate a pair of public/private keys for this. We shall stick to version 2 of ssh.

1. Firstly, generate your public/private keys using ssh-keygen

```
% ssh-keygen -t rsa
```

You must use the -t option to specify that you are producing keys for SSHv2 using RSA. This will generate your id_rsa and id_rsa.pub in the .ssh directory in your home directory. I strongly suggest using a passphrase.

2. Now copy the id_rsa.pub to the .ssh directory of the remote host you want to logon to as authorized_keys2 .

[Note: If you have more than one host from which you want to connect to the remote host, you need to add the local host's id_rsa.pub as one line in the authorized_keys2 file of the remote host, i.e., you can have more than one entry. Thanks to Jinn Koriech for pointing this out. Also, you need to 'chmod 644 authorized_keys2' to make it unwritable to everybody apart from the user. Thanks to Matthew Lohbihler for this info. Andy Pieters writes it is best to have .ssh and associated directories on the server machine to have at most 0600 permissions.]

You are basically telling the sshd daemon on the remote machine to encrypt the connection with this public key and that this key is authorized for version 2 of the ssh protocol. Try using something secure like scp for this copying.

```
% scp ~foo/.ssh/id_rsa.pub foo@bar.cs.umd.edu:~foo/.ssh/authorized_keys2
```

3. Your public key based authentication has been setup. You won't be asked your password on the remote machine. However, you need a program that manages your keys for you called an agent. You need to start the agent, tell it your passphrase, and hook up to the agent whenever you need to connect to the remote machine.

4. We shall assume the following situation: You logon to a console and then startx as in say, an out-of-the-box Linux installation. You should figure out what exactly has to be done for your specific machine's X initialization. All the following steps are to be done on your local machine, in this case- localmachine.cs.umd.edu.

5. Fire your favourite editor, and pull up your .profile file. Add the following line to the file:

```
alias startx='ssh-agent startx'
```

This means that every child of startx (i.e. anything under X) would be able to hookup to the agent.

6. Edit your .xinitrc file by adding the following lines:

```
DISPLAY="localmachine.cs.umd.edu:0"  
SSH_ASKPASS="/usr/libexec/openssh/x11-ssh-askpass"  
ssh-add < /dev/null
```

```
# Change this to whatever window manager you use under X  
# or leave whatever was there unchanged.  
startkde
```

.xinitrc is the init file for X. Unfortunately, as ssh-add doesn't have a controlling terminal, it needs to be told to read input from an external source. When you specify, /dev/null, the program pops up a d-box program specified by \$SSH_ASKPASS and ask you for your passphrase. The x11-ssh-askpass that comes with your openssh installation is the d-box program . The DISPLAY is usually automatically set, but just in case.

A hint from Oliver Meili: If a graphical login manager like XDM,KDM, or GDM is used, ~/.xsession is started which, in turn, runs ~/.xinitrc. You can add the ssh-agent to the line starting ~/.xinitrc in the ~/.xsession file like this: ssh-agent ~/.xinitrc and the ~/.xinitrc file has everything else same as before.

If you had to create .xinitrc, then you must add something after the ssh-add statement to start the window-manager/desktop/whatever. Otherwise, X will simply terminate after asking for the password. If you don't know how to set this up, you might want to dig in your /etc/X11/init.d files for the appropriate init sequences.

7. Now when you startx, a dialog box should pop up and ask you for your passphrase. You are all set. Open up an xterm, and say

```
% ssh bar.cs.umd.edu
```

Voila ! You'll be logged in without typing in your password. You'll have to re-enter your passphrase, everytime you start X. The passphrase can be side-stepped by giving the empty string, but I'd rather you don't.

8. As a fringe benefit, you can execute any GUI based programs on the remote machine for free provided X forwarding has been enabled; no setting up \$DISPLAY , no need to xhost+ etc. Cool, eh ?

NB: No promises if this will work for you. I am not responsible if you screw up your workspace environment and/or your machine.

—AcV—
arun@cs.umd.edu