O'REILLY®

HACKS
hack (hăk) *n.*, A clever solution to an interesting problem.

oreilly.com | O'Reilly Network | Safari Bookshelf | Conferences | Sign In/My Account | View Cart

Books | O'Reilly Gear | Newsletters | Press Room | Jobs | About O'Reilly

Search

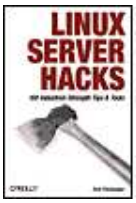
Hacks Site
- List of Titles
- Got a Hack?
- Suggestion Box
- About Hacks

Resource Centers
Bioinformatics
C/C++
Databases
Digital Media
Enterprise Development
Game Development
Java
Linux/Unix
Macintosh/OS X
.NET
Open Source
Oracle
Perl
Python
Scripting
Security
SysAdmin/Networking
Web
Web Services
Windows
Wireless
XML

Book Series
Annoyances
CD Bookshelves

LINUX SERVER HACKS

Buy the book!



HACK #66 Quick Logins with ssh Client Keys

Using ssh keys instead of password authentication to speed up and automate logins

[Discuss (11) | Link to this hack]

By Rob Flickenger
January 2003

More Info

When you're an admin on more than a few machines, being able to navigate quickly to a shell on any given server is critical. Having to type "ssh my.server.com" (followed by a password) is not only tedious, but it breaks one's concentration. Suddenly having to shift from "where's the problem?" to "getting there" and back to "what's all this, then?" has led more than one admin to premature senility. It promotes the digital equivalent of "why did I come into this room, anyway?" (In addition, the problem is only made worse by /usr/games/fortune!)

At any rate, more effort spent logging into a machine means less effort spent solving problems. Recent versions of ssh offer a secure alternative to endlessly entering a password: public key exchange.

To use public keys with an ssh server, you'll first need to generate a public/private key pair:

```
$ ssh-keygen -t rsa
```

You can also use -t dsa for DSA keys, or -t rsa1 if you're using Protocol v1. (And shame on you if you are! Upgrade to v2 as soon as you can!)

After you enter the above command, you should see something like this:

```
Generating public/private rsa key pair.
Enter file in which to save the key (/home/rob/.ssh/id_rsa):
```

Just hit Enter there. It will then ask you for a pass phrase; just hit enter twice (but read the Security note below). Here's what the results should look like:

```
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/rob/.ssh/id_rsa.
Your public key has been saved in /home/rob/.ssh/id_rsa.pub.
The key fingerprint is:
a6:5c:c3:eb:18:94:0b:06:a1:a6:29:58:fa:80:0a:bc rob@localhost
```

This created two files, ~/.ssh/id_rsa and ~/.ssh/id_rsa.pub. To use this keypair on a server, try this:

```
$ ssh server "mkdir .ssh; chmod 0700 .ssh"
```

Cookbooks
Developer's Notebooks
Hacks
Head First
In A Nutshell
Missing Manuals
Pocket References
Personal Trainer
Technology & Society

Publishing Partners
No Starch Press
Paraglyph Press
Pragmatic Bookshelf
SitePoint
Syngress Publishing

Online Publications
LinuxDevCenter.com
MacDevCenter.com
ONDotnet.com
ONJava.com
ONLamp.com
OpenP2P.com
Perl.com
WebServices.XML.com
WindowsDevCenter.com
XML.com

Special Interest
Ask Tim
Beta Chapters
Events
From the Editors List
Letters
MAKE
Open Books
tim.oreilly.com

Special Sales
Academic
Corporate Services
Government

Inside O'Reilly
About O'Reilly
Bookstores
Catalog Request
Contact Us
International
User Groups
Writing for O'Reilly

```
$ scp .ssh/id_rsa.pub server:.ssh/authorized_keys2
```

Of course, substitute your server name for `server`. It should ask for your password both times. Now, simply ssh server and it should log you in automagically without a password. And yes, it will use your shiny new public key for scp, too.

If that didn't work for you, check your file permissions on both ~/.ssh/* and server:~/.ssh/*. Your private key (`id_rsa`) should be 0600 (and only be present on your local machine), and everything else should be 0655 or better.

Terrific. So you can now ssh server quickly and with a minimum of fuss. Is it possible to make it even quicker to connect to machines you frequently touch? You bet, just check out "`Turbo-mode' ssh Logins" (#67).

Security Concerns

Some consider the use of public keys a potential security risk. After all, one only has to steal a copy of your private key to obtain access to your servers. While this is true, the same is certainly true of passwords.

Ask yourself, how many times a day do you enter a password to gain shell access to a machine (or scp a file)? How frequently is it the same password on many (or all) of those machines? Have you ever used that password in a way that might be questionable (on a web site, on a personal machine that isn't quite up to date, or possibly with an ssh client on a machine that you don't directly control). If any of these possibilities sound familiar, then consider that an ssh key in the same setting would make it virtually impossible for an attacker to later gain unauthorized access (providing, of course, that you keep your private key safe).

See also:

- SSH: The Definitive Guide
  (O'Reilly)
- Logins" (#67).
- "F

Comment on this hack
You must be logged in to the O'Reilly Network to post a comment.

Post Comment Main Thread Only Old First

Showing messages 1 through 10 of 10.

nice
2005-03-10 11:55:15 PocketMan [Reply | View]

Traveling to a tech show?

Lubbock Hotels
Kihei Hotels
Ann Arbor Hotels
Fargo Hotels
Florence Hotels
Fredericksburg

Hotels
Santa Cruz Hotels
Englewood Hotels

Oh, cool, but how to do it in windows, I using putty client and have many ssh-accs.

Don't use plaintext keys for interactive logins
2003-04-07 09:43:31 anonymous2 [Reply | View]

There's no reason to use a plaintext (passwordless) key for interactive logins, ever. Despite what the author says, a passwordless key is NOT "the same" level of security as a password, since the key sits ON DISK, unencrypted, and a password does not. A better analogy would be putting your sensitive login password into a file named "StealMe.txt". Instead, use a strong passphrase and run ssh-agent to avoid the need to type passwords. For interactive use, this is a far better solution: now an attacker would need TWO secrets (your key and your passphrase) to impersonate you, and you still get passwordless logins.

Also the method of transferring your public key to the server (scp .ssh/id_rsa.pub server:.ssh/authorized_keys2) is not good. Keys should be *appended* to the file, not overwrite the file, in case the file exists. (Yes, I know this hack is just for setting up your first key, but a novice might easily repeat the command, overwriting an existing key file.) Finally, authorized_keys2 is deprecated in favor of authorized_keys in recent versions of OpenSSH. (And of course you should always run a recent version of an important security product like this.)

Novice - almost caught by existing authorized_keys2
2003-12-30 22:32:09 anonymous2 [Reply | View]

I'm a novice, and tried to use this command today.

Fortunately, I remembered at the last moment that the consultant who had set up my system had an existing keyring. I went in and looked prior to running the command. Sure enough - there was authorized_keys2 just setting there, waiting to be overwritten.

But now I'm screwed, because the hack doesn't tell me - a novice - how to append the key onto the existing key. And I'm afaid to just go scp .ssh/id_rsa.pub >> server:.ssh/authorized_keys2 in case it doesn't work right, and I screwup my existing file.

Help!

Plus what's this about authorized_keys2 itself being depreciated? I thought this was a new book?! I would have assumed we'd be getting the very latest in technology.

Can someone (in desending priority order) please:

a: give me the command to safely append to the existing authorized_key2 file,
b: point me to how I can create a key-pair I can split between Linux and Windows, so I can use this trick with PuTTY to connect from Windows to my Linux servers, and
c: if we shouldn't be using authorized_key2 because its depreciated, correct the hack to the proper usage, and tell me how to fix my current authorized_key2 configuration to the proper usage/configuration.

Thank you very much.

Novice - almost caught by existing authorized_keys2
2005-03-11 14:24:38 rodlinux [Reply | View]

I tried this with solaris. I scp using the reccomended file. But then I did:

cat authorized_keys2 >> authorized_keys

Don't use plaintext keys for interactive logins
2003-04-07 09:43:00 anonymous2 [Reply | View]

There's no reason to use a plaintext (passwordless) key for interactive logins, ever. Despite what the author says, a passwordless key is NOT "the same" level of security as a password, since the key sits ON DISK, unencrypted, and a password does not. A better analogy would be putting your sensitive login password into a file named "StealMe.txt". Instead, use a strong passphrase and run ssh-agent to avoid the need to type passwords. For interactive use, this is a far better solution: now an attacker would need TWO secrets (your key and your passphrase) to impersonate you, and you still get passwordless logins.

Also the method of transferring your public key to the server (scp .ssh/id_rsa.pub server:.ssh/authorized_keys2) is not good. Keys should be *appended* to the file, not overwrite the file, in case the file exists. (Yes, I know this hack is just for setting up your first key, but a novice might easily repeat the command, overwriting an existing key file.) Finally, authorized_keys2 is deprecated in favor of authorized_keys in recent versions of OpenSSH. (And of course you should always run a recent version of an important security product like this.)

different filenames to note
2003-04-02 07:46:32 anonymous2 [Reply | View]

The above procedure didn't work initially on my FreeBSD server. Checking the /etc/ssh/sshd_config file; I noticed that PubkeyAuthentication was a 'no' and AuthorizedKeysFile was .ssh/authorized_keys .

Not sure if these are defaults, but the former should be yes and the latter should match the filename that's copied from your host to the user's ~/.ssh directory.

(if you change this file, best restart your ssh server, and 'killall -HUP sshd' while logged in remotely might be a BAD way to do that!)

great tip, thanks!

cosmo

Server asks for passphrase...
2003-03-28 13:39:48 anonymous2 [Reply | View]

Hi,

what did i wrong? After I try to login with

ssh server

I am asked for the passphrase...???

Server asks for passphrase...
2003-04-16 21:39:39 anonymous2 [Reply | View]

Check out ssh-agent. It manages passphrases for you so you don't have to type it.

Works Great
2003-03-19 08:56:17 anonymous2 [Reply | View]

Worked like a charm for ssh'ing from my linux to openbsd box.

Works Great

2005-03-16 21:51:51 t00tah [Reply | View]

speaking of killing all existing ssh connections while testing a new config, on Linux (I primarily use RH), I'd stay logged-in with an existing session and do a '/etc/rc.d/init.d/sshd restart' and then try my changes with a new log-in attempt. I know it sounds basic, but I _have_ logged myself out of machines before and the above is a lesson learned the hard way ;)
Peace,
t00tah

Showing messages 1 through 10 of 10.

O'Reilly Home | Privacy Policy

© 2005, O'Reilly Media, Inc.
Website: webmaster@oreilly.com | Customer Service: orders@oreilly.com | Book issues: booktech@oreilly.com

All trademarks and registered trademarks appearing on oreilly.com are the property of their respective owners.