Dustin Boswell - Brain Dumps about Computers, Programming, and Everything Else

dustwell.com
About Me
Past Academic Work

Blog Posts
How to hash passwords
Snapshotting with Rsync
"An hour" vs. "A hour"
Yes/No proposition bets
div, span, and CSS "display:"
Pair programming + screen
SSH keys in 2 easy steps
Fixing Your Flaky Internet
How X-over-SSH works
Drinking Distilled Water
djb-dns installation
Vim Cheat Sheet
Oil ETFs leak money

SSH keys in 2 easy steps

February 13, 2010

These are simple instructions that will let you ssh from one Linux machine to another without needing to type your password.

Step 1) Generate your public signature

On your local machine (where you are ssh-ing *from*) type:

```
ssh-keygen
```

(Then hit ENTER to accept the default output file of `~/.ssh/id_rsa.pub` and ENTER again twice if you're lazy and want to use a blank passphrase.) Note that you only have to generate a key **once** per client machine - the same public key will be used to access all servers.

Step 2) Copy your public signature to the server

Again, from your local machine, type:

```
cat ~/.ssh/id_rsa.pub | ssh remote_user@remote.example.com "cat >> ~/.ssh/autho
```

(but replace `remote_user@remote.example.com` with your actual user and server.)

This fancy shell command **appends** the contents of your public signature to the end of the `~/.ssh/authorized_keys` file on the server. (If you did a simple `scp` it would overwrite any previous authorized keys you've stored.)

You're done!

Next time you ssh into the server

```
ssh remote_user@remote.example.com
```

It should do this without prompting for any passwords.

0 Comments **Dustin Boswell's Blog** **Javier Martín Alon...**

♡ **Recommend** Tweet Share **Sort by Oldest**

Start the discussion…

Be the first to comment.

✉ **Subscribe** **Add Disqus to your siteAdd DisqusAdd** **Disqus' Privacy PolicyPrivacy PolicyPrivacy**