next up previous contents

Next: telnet: Connecting to Remote **Up:** SSH: Connecting Securely to **Previous:** Escape sequences
Contents

`ssh-keygen`: password-less SSH login

SSH is often used to login from one system to another without requiring passwords.

A number of methods may be used for that to work properly, one of which is to setup a `.rhosts` file (permission 600) with its content being the name of the remote system you trust, followed by the username your trust:

```
nickel.sao.nrc.ca cantin
```

would mean you trust user `cantin` from `nickel.sao.nrc.ca` to connect to your account, without requiring a password.

But for that to work, SSH itself must be configured to trust `.rhosts` files (which it does not for most OpenSSH installations - but we do on most systems RCSG maintains), and the private/public key pair of each system must be properly set in the system-wide `ssh_known_hosts` public key file.

This, of course, requires help from the local systems administrator.

The second method does not require any help from the systems administrator. And it does not require modifications to the `.rhosts` file. Instead, it requires you generate your own personal set of private/public pair.

`ssh-keygen` is used to generate that key pair for you. Here is a session where your own personal private/public key pair is created:

```
cantin@sodium:~> ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/cantin/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/cantin/.ssh/id_rsa.
Your public key has been saved in /home/cantin/.ssh/id_rsa.pub.
The key fingerprint is:
f6:61:a8:27:35:cf:4c:6d:13:22:70:cf:4c:c8:a0:23 cantin@sodium
```

The command `ssh-keygen -t rsa` initiated the creation of the key pair.

No passphrase was entered (`Enter` key was pressed instead).

The private key was saved in `.ssh/id_rsa`. This file is read-only and only for you. No one else must see the content of that file, as it is used to decrypt all correspondence encrypted with the public key.

The public key is save in `.ssh/id_rsa.pub`.

In this case, the content of file `id_rsa.pub` is

```
ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAIEArkwv9X8eTVK4F7pMlSt45pWoiakFkZMw
G9BjydOJPGH0RFNAy1QqIWBGWv7vS5K2tr+EEO+F8WL2Y/jK4ZkUoQgoi+n7DWQVOHsR
ijcS3LvtO+50Np4yjXYWJKh29JL6GHcp8o7+YKEyVUMB2CSDOP99eF9g5Q0d+1U2WVdB
WQM= cantin@sodium
```

It is one line in length.

Its content is then copied in file `.ssh/authorized_keys` of the system you wish to SSH to without being prompted for a password.

The example shown here generated keys on `sodium` by user `cantin`. If the public key generated, file `.ssh/id_rsa.pub`, was copied to your account, file `.ssh/authorized_keys` on `nickel.sao.nrc.ca`, then user `cantin@sodium` is allowed to SSH into your own account on `nickel.sao.nrc.ca` without the use of a password.

To summarize, a personal private/public key pair is generated using the `ssh-keygen` command. The public key is then copied onto a remote systems' `.ssh/authorized_keys` file. And you can now SSH to the remote systems's account without the use of a password.

---

next up previous contents

Next: telnet: Connecting to Remote **Up:** SSH: Connecting Securely to **Previous:** Escape sequences
Contents

Claude Cantin 2010-10-24