Making your life easier and more secure when working remotely

Python Module of the Week

Dennis Terhorst

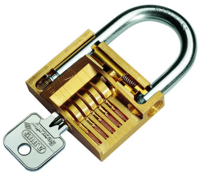
26 April 2019

Overview

Topics

- General considerations
- SSH keys & agents
- .ssh/config
- Usage examples

General considerations

-  Trust-model using keys:
  – access is granted to whoever has the key
- Keys are generated as *key-pair*, one is used to crypt and one de-crypt data
- you can have as many keys as you want

SSH keys & agents

Generate a key

```
ssh-keygen -b 4096 -C "y.name@fz-juelich.de"
```

- `-t` specifies the type of the key (default: rsa)
- `-b` gives the nuber of bits (default 2048)
- `-C` gives a useful identifier in the comment field (default `$USER@$HOST`)

- **never** create a key without a password
- give a useful filename, e.g. the identifier
- protect your private key!

This will produce two files, for example

- `y.name@fz-juelich`
- `y.name@fz-juelich.pub`

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQAB...Da9V08Ew== y.name@fz-juelich.de
```

Using a key

Add your public key to the file `~/.ssh/authorized_keys` on the destination machine

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQAB...Da9V08Ew== y.name@fz-juelich.de
```

Then use your private key to log into the machine (`man ssh`)

```
ssh -i y.name@fz-juelich  y.name@login.inm.kfa-juelich.de
```

Note

- The password you give is the *password of the key* not of the target machine
- The private key is never transfered, the password is processed only *locally*

The SSH Agent

When frequently accessing a system it may be impractical to re-type the potentially longisch key password for each connection, esp. when connecting to other hosts in scripts.

Start the agent

NOTE: This is not necessary if the gnome-keyring is available!

wrong

```
$ ssh-agent
SSH_AUTH_SOCK=/tmp/ssh-hHmOg2ChTmr4/agent.31758; export SSH_AUTH_SOCK;
SSH_AGENT_PID=31759; export SSH_AGENT_PID;
echo Agent pid 31759;
```

right

```
$ eval $(ssh-agent)
```

Using the agent

- if `$SSH_AUTH_SOCKET` is in your environment, the agent will be used

```
ssh  y.name@login.inm.kfa-juelich.de
```

- if you need to do more ssh connections on the remote side, you can
    1. tell `ssh` to "forward the ssh-agent socket" to the remote side

       ```
       ssh -A  y.name@login.inm.kfa-juelich.de
       ```

       or configure `ForwardAgent yes` (see next section)
    2. copy your encrypted private key to the intermediate machine

Make your life easier: `.ssh/config`

The problem

Even with the SSH agents lines can become annoying to re-type/remember

- which key for which host
- different required options
- different usernames
- ...

```
ssh -X -o ForwardAgent=yes  y.name@login.inm.kfa-juelich.de
ssh -o TCPKeepAlive=yes  yname1@jureca.fz-juelich.de
...
```

Basic config

Create/edit your `~/.ssh/config` (`man ssh_config`):

```
Host login
    Hostname login.inm.kfa-juelich.de
    User y.name
    IdentitiesOnly yes
    IdentityFile ~/.ssh/y.name@fz-juelich
```

This lets you do

```
ssh login
```

which is then equivalent to

```
ssh -i ~/.ssh/y.name@fz-juelich -o IdentitiesOnly=yes \ y.name@login.inm.kfa-juelich.de
```

Usage examples

cluster/HPC

```
Host hambach
    HostName hambach.inm.kfa-juelich.de
    User terhorst
    IdentityFile ~/.ssh/y.name@fz-juelich
    IdentitiesOnly yes

Host jureca
    HostName jureca.fz-juelich.de
    User jinb3326
    KbdInteractiveAuthentication no
    IdentityFile ~/.ssh/my-global-key
    IdentitiesOnly yes
```

gitty/Github

gitty

```
Host gitty
    Hostname gitty.inm.kfa-juelich.de
    User git
    IdentitiesOnly yes
    IdentityFile ~/.ssh/y.name@fz-juelich
```

then run `ssh gitty`

Github

Go to your **github settings page** “SSH and GPG keys” and add your public key

cloning repositories

```
git clone gitty:csn_toolbox
git clone github:INM-6/equipment-overview
```

sshfs vs. rsync

Sync remote files

Using the agent and ssh-config things become very easy:

```
ssh login
date >somefile
rsync -avi login:somefile .
```

Using `sshfs`

```
mkdir remote
sshfs login: remote
ls remote
fusermount -u remote
```

NOTE

With sshfs processes still run locally! Only files are transfered! Huge difference in speed and responsiveness

Finally

Summary

- no additional software/aliasses/scripts required
- things become much easier and safer

→ Discuss!

Not covered

- different users on same host
- Control channels
- Connection forwarding

Thanks!