SSH Tips

This page contains some helps on how to do some (simple) actions with OpenSSH[1]

1 Getting a key's fingerprints

To get the fingerprints of the public key `$key`, do:

```
ssh-keygen -lf $key
```

2 What key does sshd use ?

Host keys should be indicated in the sshd_config file (`/etc/sshd_config`) by the variable `HostKey`.

If they are not, the defaults are `/etc/ssh/ssh_host_rsa_key` and `/etc/ssh/ssh_host_dsa_key` (whether we use dsa or rsa key depends on what the client wants)

3 SSH Reverse Tunnel

3.1 Introduction

This describes how I had setup SSH reverse tunneling on my machines. This service is not implemented anymore.

Reverse SSH tunnels should be used to access computers behind a NAT from the outside. They should only be used if it is useful to have access to the machine.

I will call the machine behind the NAT the client and the tunneling server the server.

3.1.1 Tools

I use openssh[2], with autossh[3] on the client to maintain the connection permantently.

3.1.2 How the tunnel is established

The client launches sshd (configured to disallow root login, as usual), and runs autossh with

- A monitor port (-M) of 20000 + k The -N and -R options to ssh,

[1] http://www.openssh.com/

[2] http://www.openssh.com/

[3] http://www.harding.motd.ca/autossh/

- establishing the tunnel on port 20000 + k + 2, where k depends on the client and is documented here.

The server hostname is always `sshtunnel.svasey.org` and the username to connect as is always sshtunnel. The authentication is made by public key.

Every ssh configuration files (`known_hosts`, `id_rsa*`, `ssh_config`) should be in `/etc/sshtunnel`. The `ssh_config` file should look like this:

```
HostName sshtunnel.svasey.org
Port 6002
User sshtunnel
BatchMode yes
GlobalKnownHostsFile /etc/sshtunnel/known_hosts
ConnectionAttempts 3
IdentityFile /etc/sshtunnel/id_rsa
PasswordAuthentication no
PubKeyAuthentication yes
```

The command establishing the ssh tunnel is:

```
autossh -f -M 20000 -N -R 20002:127.0.0.1:6002 -F /etc/sshtunnel/ssh_config remote
```

3.1.3 Port repartition

Since we cannot use the same ports for more than one tunnel, we have to assign ports arbitrarily. The convention is that for client number 1, ports 20000 and 20001 are used for monitoring and 20002 for tunneling. For client number 2, ports 20003 and 20004 are used for monitoring and 20005 for tunneling and so on.

3.1.4 How to transfer a file from client to client

Let's say you have ssh access to a server (named `$server`) and would like to download a file from a client tunneled behind it (named `$client`).

One solution is to download the file from `$client` to `$server` using sftp and then from `$server` to you, but it takes twice as much time. The other solution is to use sshfs[4] (fifos-based solutions have been tried and do not work well).

First (on `$server`), install sshfs and load up the fuse module:

```
yaourt sshfs
modprobe fuse
```

Then mount `$client` 's filesystem remotely:

```
mkdir ~/tmp/mnt
sshfs john@127.0.0.1:multimedia ~/tmp/mnt -p 20002
```

WARNING: try to mount only the directory you need to restrict privilleges

Then, download the file you want from your machine:

```
scp -P 6002 john@${server}:tmp/mnt/file/you/want .
```

[4] `http://fuse.sourceforge.net/sshfs.html`

When you are done, do not forget to umount the directory (on `$server`):

```
fusermount -u ~/tmp/mnt
```