SSH Agent Forwarding

What is SSH-Agent Forwarding ?

What is SSH-Agent ?

SSH-Agent allows you to have your SSH key's password saved into memory. Thus, you will have to give your key's password **once** to the agent, and then you won't have to type in your password to unlock that key as long as the agent is running.

What is Agent Forwarding and why would I need it ?

Let's assume you have your machine at home. You have to administrate the machine named 'Plutonium' which sits on the local network of a service or business unit. The service is using the machine 'Tiberium' as a Gateway/Firewall. As a conscious admin, you set up Tiberium to block any unsollicited connection from the outside world to the local network, and you definitely don't want to allow any SSH connection directly from outside world to Plutonium. Basically, to reach Plutonium through SSH, you would first connect to Tiberium, and then connect from Tiberium to Plutonium. This is a valid security scheme, but it has its drawbacks:

- An SSH Key which is allowed to connect to Plutonium has to sit on Tiberium: Some other persons have physical acces to Tiberium and might steal your key, and Tiberium might get compromised – keeping a private key with administration privileges for the local network on Tiberium is definitely a **risk**.
- You are ~~lazy~~ clever and you don't want to waste your time giving a password to unlock the same key several times - especially if you have a **long & strong**... password! - And you have one, don't you ?

SSH-Agent forwarding comes in with support for both issues.
SSH-Agent Forwarding works this way:

- You start the agent on your machine and unlock your private key.
- You connect to Tiberium with Agent Forwarding enabled.
- You connect from Tiberium to Plutonium - an authentication is asked from Plutonium to Tiberium, and Tiberium simply forwards this request to your home machine. Your SSH Agent catches the request and authenticates. Bam ! You're in Plutonium !

To sum it up:

Pros

- + You don't need to keep a private SSH key on every machine in the middle of a 'chain'.
- + You don't have to provide a password everytime you connect from a machine to another (but see cons)

Cons

- - The 'root' user of each machine in the chain is able to use your SSH-Agent, thus **your key**. The hosts you connect to using SSH agent has to be **trusted**. Please note that root can **use** your key, he cannot **download** it - but he won't need the password for it since your Agent will always provide the sesame.
- - The whole chain of machines have to be **TRUSTED** (I prefer to write it twice.)
  OK, let's stress it even more :
  Yes, the machines between you and the target host have to be **TRUSTED** - but it's not worse than before: the machine has to be compromised enough to allow an attacker to gain root acces, and rebooting the machine on a live CD or a USB stick to mount the partitions and download your beloved private key won't help the attacker.

How to use SSH-Agent

Start SSH Agent

The command

```
ssh-agent
```

starts the SSH Agent and displays on screen the environment variables you have to set to use it.
Hence the most convenient way to start SSH-Agent is to use:

```
eval `ssh-agent`
```

This will automatically set the environment variables for you.
You could eventually add this 'eval `ssh-agent`' to your Session Startup.

Add and unlock an identity to use with SSH Agent

Either you are on your home machine and want to use your 'regular' SSH Identity (key ~/.ssh/id_rsa) with the Agent, then simply type:

```
[user@host:/home/user]$ ssh-add
```

You will be prompted for your **key** password and add your Identity to the Agent.

Or, you might be on a guest machine and have your SSH key on a usb stick (key /media/usbstick/private_key.key). Then you have to use:

```
[user@host:/home/user]$ ssh-add /media/usbstick/private_key.key
```

This will prompt for your **key** password and add '/media/usbstick/private_key.key' to the agent.

Connection to a host using SSH Agent Forwarding:

Once your agent is started and you have added the desired identity/key, to use it, simply type:

```
[user@host:/home/user]$ ssh -A username@targetmachine
```

This will connect to 'targetmachine' on port 22 as user 'username' and SSH-Agent Forwarding enabled.

Exemple

In the scenario described before (home machine -> Tiberium -> Plutonium), the step-by-step method would be something like:

```
[user@host:/home/user]$ eval `ssh-agent`
Agent PID 1643
[user@host:/home/user]$ ssh-add
Enter passphrase for /home/user/.ssh/id_rsa:
Identity added: /home/user.ssh/id_rsa (/home/user/.ssh/id_rsa)
[user@host:/home/user]$ ssh -A admin@tiberium
[admin@tiberium:/home/admin]$ ssh -A admin@plutonium
[admin@plutonium:/home/admin]$ _
```

Et voila! You gave your password once, and could bounce from hosts to hosts !