Working with SSH key passphrases

In this article

You can secure your SSH keys and configure an authentication agent so that you won't have to reenter your passphrase every time you use your SSH keys.

Mac Windows

About passphrases for SSH keys

With SSH keys, if someone gains access to your computer, the attacker can gain access to every system that uses that key. To add an extra layer of security, you can add a passphrase to your SSH key. To avoid entering the passphrase every time you connect, you can securely save your passphrase in the SSH agent.

Adding or changing a passphrase

You can change the passphrase for an existing private key without regenerating the keypair by typing the following command:

```
$ ssh-keygen -p -f ~/.ssh/id_ed25519
> Enter old passphrase: [Type old passphrase]
> Key has comment 'your_email@example.com'
> Enter new passphrase (empty for no passphrase): [Type new passphrase]
> Enter same passphrase again: [Repeat the new passphrase]
> Your identification has been saved with the new passphrase.
```

If your key already has a passphrase, you will be prompted to enter it before you can change to a new passphrase.

Auto-launching `ssh-agent` on Git for Windows

You can run `ssh-agent` automatically when you open bash or Git shell. Copy the following lines and paste them into your `~/.profile` or `~/.bashrc` file in Git shell:

```
env=~/.ssh/agent.env

agent_load_env () { test -f "$env" && . "$env" >| /dev/null ; }

agent_start () {
    (umask 077; ssh-agent >| "$env")
    . "$env" >| /dev/null ; }
```

```
agent_load_env

# agent_run_state: 0=agent running w/ key; 1=agent w/o key; 2=agent not running
agent_run_state=$(ssh-add -l >| /dev/null 2>&1; echo $?)

if [ ! "$SSH_AUTH_SOCK" ] || [ $agent_run_state = 2 ]; then
    agent_start
    ssh-add
elif [ "$SSH_AUTH_SOCK" ] && [ $agent_run_state = 1 ]; then
    ssh-add
fi

unset env
```

If your private key is not stored in one of the default locations (like `~/.ssh/id_rsa`), you'll need to tell your SSH authentication agent where to find it. To add your key to ssh-agent, type `ssh-add ~/path/to/my_key`. For more information, see "Generating a new SSH key and adding it to the ssh-agent"

Tip: If you want `ssh-agent` to forget your key after some time, you can configure it to do so by running `ssh-add -t <seconds>`.

Now, when you first run Git Bash, you are prompted for your passphrase:

```
> Initializing new SSH agent...
> succeeded
> Enter passphrase for /c/Users/YOU/.ssh/id_rsa:
> Identity added: /c/Users/YOU/.ssh/id_rsa (/c/Users/YOU/.ssh/id_rsa)
> Welcome to Git (version 1.6.0.2-preview20080923)
>
> Run 'git help git' to display the help index.
> Run 'git help <command>' to display help for specific commands.
```

The `ssh-agent` process will continue to run until you log out, shut down your computer, or kill the process.

Saving your passphrase in the keychain

On Mac OS X Leopard through OS X El Capitan, these default private key files are handled automatically:

- *.ssh/id_rsa*
- *.ssh/identity*

The first time you use your key, you will be prompted to enter your passphrase. If you choose to save the passphrase with your keychain, you won't have to enter it again.

Otherwise, you can store your passphrase in the keychain when you add your key to the ssh-agent. For more information, see "Generating a new SSH key and adding it to the ssh-agent."