

Super User is a question and answer site for computer enthusiasts and power users. It only takes a minute to sign up.

[Sign up to join this community](#)

Anybody can ask a question

Anybody can answer

The best answers are voted up and rise to the top



Connecting to github using PuTTY generated SSH key in Windows 7

Asked 7 years, 6 months ago Active 3 years, 7 months ago Viewed 65k times

- ▲ 19 ▼
- On an otherwise pristine Windows 7 Enterprise install:
1. Install PuTTY 0.62
 2. Create public/private key pair with puttygen.exe
 3. Add public key for pasting into OpenSSH authorized_keys to github account
 - ★ 4. Add private key to pageant
 - 14 5. Install msysgit 1.7.8 with the following options:
 - Use Git from Windows Command Prompt
 - Use (Tortoise)Plink pointed to correct plink.exe
 6. Make sure github fingerprint is added to know_hosts by connecting to github.com in PuTTY and accepting fingerprint

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

```
Host github.com
  User git
  Hostname github.com
  PreferredAuthentications publickey
  IdentityFile ~/.ssh\github.ppk
```

`ssh -vvvT git@github.com` gives: <http://pastebin.com/Tu3Fc6nJ>. Note that I'm being prompted for the pass-phrase despite it being successfully loaded into pageant.

I've tried: `chmod 700 ~/.ssh; chmod 600 ~/.ssh/*` from within Git Bash but there was no effect as verified by `ls -l`.

I should note that I had this exact setup working on a previous Windows 7 install. Everything is identical as far as I can tell.

What does work is generating keys from within Git Bash. However, I cannot add those keys into pageant, which is a huge pain. In effect, it seems that any attempt to use keys generated by PuTTY is unsuccessful.

Any insight on what's preventing this from working based on the logs? Thanks in advance.

[windows-7](#) [ssh](#) [git](#) [putty](#) [github](#)

edited Jan 14 '12 at 16:49



grawity

257k

38

541

606

asked Jan 14 '12 at 15:55



Radu

229

2

3

12

with -vT, please, less verbosity level – [Lazy Badger](#) Jan 14 '12 at 16:30

3 Answers



19



You are confusing two entirely separate programs: PuTTY and OpenSSH.

- `plink` and Pageant are part of PuTTY. The `ssh` command is part of OpenSSH. It is unclear which program is being used by Git; you need to check the `%GIT_SSH%` environment variable for that.
- The programs use different agent protocols; OpenSSH cannot use PuTTY's Pageant; it has its own `ssh-agent` (which unfortunately

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

- PuTTY and `plink` store the session settings in registry, editable in PuTTY's interface. They do not use anything in `~/.ssh/`; this directory is only used by OpenSSH.
- The private key formats used by OpenSSH and PuTTY are different; you cannot use a `.ppk` key with OpenSSH. If you generated the key in PuTTYgen, you have to use its "Export → OpenSSH" command.

```
$ ssh -vvvT git@github.com
OpenSSH_4.6p1, OpenSSL 0.9.8e 23 Feb 2007
...
debug2: key_type_from_name: unknown key type 'PuTTY-User-Key-File-2:'
```

answered Jan 14 '12 at 16:47



grawity

257k 38 541 606

I am aware of the difference and as indicated in the original post, git has been set to use plink when installed. I have verified that this has changed the proper environment variable to point to plink.exe. I'd like to use PuTTY keys (and this has worked in the past!) with git, and the fact that its not working despite plink being used is weird... – Radu Jan 15 '12 at 20:46

Essentially, right now I can get it to work with OpenSSH keys, but not with PuTTY keys. I have tried reinstalling PuTTY and also generating new keys. TThat doesn't seem to be problem. Any idea what it could be? – Radu Jan 15 '12 at 20:48

- 1 @Radu: **plink does not support OpenSSH key format.** If your version does, then it's not `plink` you are using; it's something else. – grawity Jan 15 '12 at 20:56
- 1 @Radu: Also, the original post indicates the opposite of your claim. First you generate a key with PuTTYgen and load it to Pageant, but *then* you go with configuring OpenSSH and complain that " `ssh` prompts for pass-phrase despite being loaded into Pageant". You *are* confusing the two programs. – grawity Jan 15 '12 at 21:53 ✎

I never said that plink supports OpenSSH - Git Bash, however, does, since by default it uses OpenSSH. Reading back, you are correct, I have been confusing things since as you said, `~/.ssh/config` is not used by PuTTY. However, I'd like to reiterate that `GIT_SSH` is pointed to plink, and yet if I modify the ssh config file for OpenSSH the changes are reflected in Git Bash when attempting to connect to github. Essentially, I believe that the `GIT_SSH` environment variable is being ignored and I suppose it's something specific to my OS. I'll try on a fresh image tomorrow. – Radu Jan 16 '12 at 5:49

▲ I have put together a step-by-step guide to get Git setup for windows using PuTTY's Plink application for SSH authentication.

20

Follow along below:

By using our site, you acknowledge that you have read and understand our Cookie Policy, Privacy Policy, and our Terms of Service.

Setup

1. Install [putty.zip](#) which is available at the [PuTTY Download Page](#) or you can download individually.

- **PuTTY:** [putty.exe](#) (or by [FTP](#))

The SSH and Telnet client itself.

- **Plink:** [plink.exe](#) (or by [FTP](#))

A command-line interface to the PuTTY back ends.

- **Pageant:** [pageant.exe](#) (or by [FTP](#))

An SSH authentication agent for PuTTY, PSCP, PSFTP, and Plink.

- **PuTTYgen:** [puttygen.exe](#) (or by [FTP](#))

An RSA and DSA key generation utility.

2. Generate RSA and PPK Keys

1. Using the Git Bash, use `ssh-keygen` to generate a pair of RSA public/private keys. More information on how to do this can be found on the official [Generating SSH keys](#) article.

2. In PuTTYgen, import your existing `~/.ssh/id_rsa` (private) key, via → .

3. Save the imported key via the button as `~/.ssh/id_rsa.ppk`.

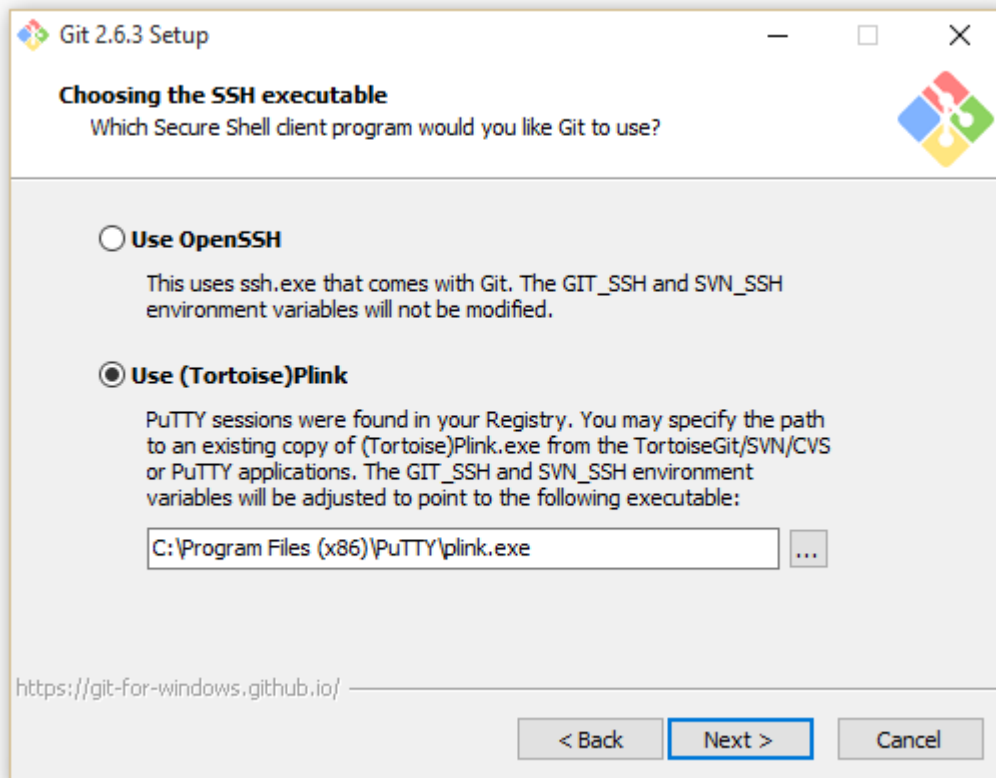
4. You should now have the following keys in your `~/.ssh` directory:

- `id_rsa` : Private (OpenSSH) RSA key
- `id_rsa.pub` : Public (OpenSSH) RSA key
- `id_rsa.ppk` : Private (PuTTY) key

3. Install [Git for Windows](#).

Make sure that you choose to use Plink.

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).



Note: If you have already installed Git, you can just run the installer again and set Plink to be your default SSH application.

4. Set your Environment paths.

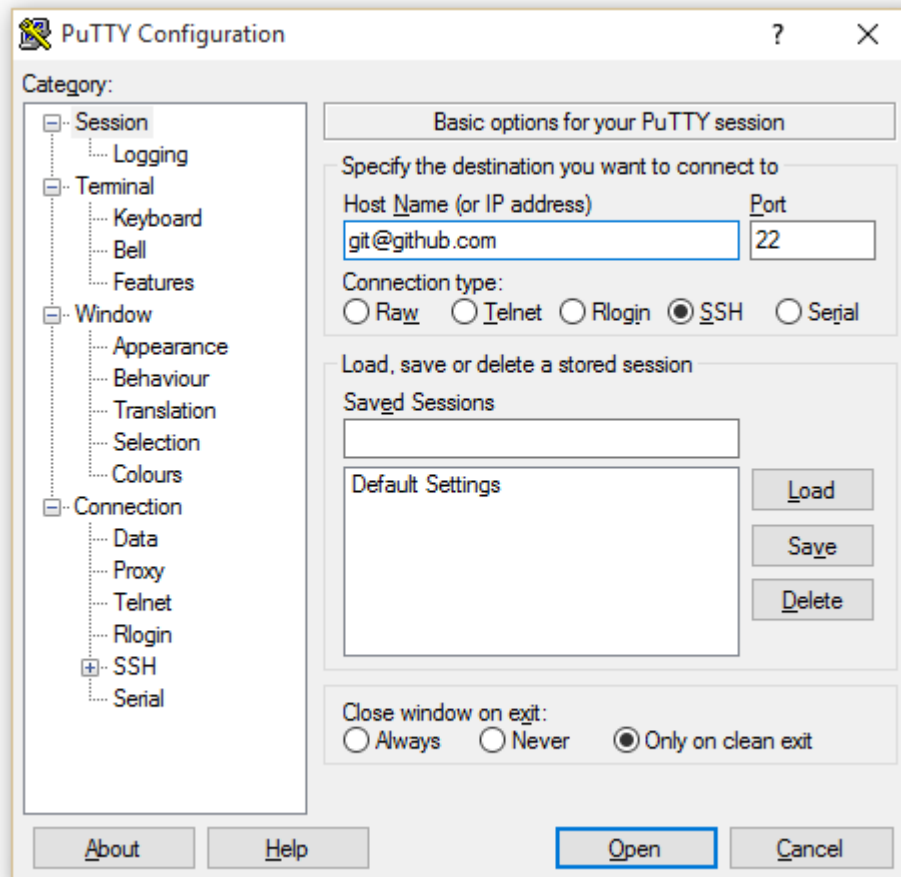
1. In *Control Panel*, navigate to the System view.
2. Choose *Advanced system settings*.
3. In the *System Properties* window, click the *Advanced* tab.
4. Click Environment variables....
5. Add the following *System variables* (if not already set):
 - **GIT_HOME** : C:\Program Files\Git
 - **GIT_SSH** : C:\Program Files (x86)\PuTTY\plink.exe

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

5. Open Pageant and load the ppk key located at `~/.ssh/id_rsa.ppk` .

Note: Once Pageant has started, you can click on its icon in the system tray located in the taskbar, next to the time, on the right.

6. Open Putty and connect to test your connection via SSH and add the server's key as a known host.



Examples hostnames:

- **GitHub:** `git@github.com:22` (or via ssh-agent `ssh -Tv git@github.com`)
- **BitBucket:** `git@bitbucket.org:22` (or via ssh-agent `ssh -Tv git@bitbucket.org`)

7. Start Git Bash.

By using our site, you acknowledge that you have read and understand our Cookie Policy, Privacy Policy, and our Terms of Service.

Shortcut

You can place a shortcut in your startup directory to auto-load your key each time you log into your Windows account.

Via Batch Script

This idea was inspired by an answer to this question:

[Super User: How to make a shortcut from CMD?](#)

```
REM /=====/
REM / Pageant AutoLoad.bat /
REM / /
REM / This script creates a shortcut for auto-loading a PPK (key) in /
REM / Pageant by writing a temporary VB script and executing it. The /
REM / following information below is added to the shortcut. /
REM / /
REM / Filename : Pageant AutoLoad /
REM / Target : pageant.exe /
REM / Arguments : id_rsa.ppk /
REM / Start in : ~/.ssh /
REM /=====/
@echo off

REM /=====/
REM / Global Values - Do not touch these! /
REM /=====/
SET VBSCRIPT="%TEMP%\%RANDOM%-%RANDOM%-%RANDOM%-%RANDOM%.vbs"
SET STARTUP_DIR=Microsoft\Windows\Start Menu\Programs\Startup
SET STARTUP_USER_DIR=%APPDATA%\%STARTUP_DIR%
SET STARTUP_ALL_USERS_DIR=%PROGRAMDATA%\%STARTUP_DIR% REM Alternative

REM /=====/
REM / Shortcut Values - You can change these to whatever you want. /
REM /=====/
SET FILENAME=Pageant AutoLoad.lnk
SET TARGET=%PROGRAMFILES(x86)%\PuTTY\pageant.exe
SET ARGUMENTS=id_rsa.ppk
SET START_IN=%USERPROFILE%\%.ssh
SET DESCRIPTION=AutoLoad PuTTY key with Pageant on startup (Ctrl+Alt+S)
SET HOTKEY=CTRL+ALT+S
```

By using our site, you acknowledge that you have read and understand our Cookie Policy, Privacy Policy, and our Terms of Service.

```

REM /===== /
ECHO Set oWS = WScript.CreateObject("WScript.Shell") >> %VBSCRIPT%
ECHO sLinkFile = "%STARTUP_USER_DIR%\%FILENAME%" >> %VBSCRIPT%
ECHO Set oLink = oWS.CreateShortcut(sLinkFile) >> %VBSCRIPT%
ECHO oLink.TargetPath = "%TARGET%" >> %VBSCRIPT%
ECHO oLink.Arguments = "%ARGUMENTS%" >> %VBSCRIPT%
ECHO oLink.WorkingDirectory = "%START_IN%" >> %VBSCRIPT%
ECHO oLink.Description = "%DESCRIPTION%" >> %VBSCRIPT%
ECHO oLink.HotKey = "%HOTKEY%" >> %VBSCRIPT%
ECHO oLink.Save >> %VBSCRIPT%
CScript //Nologo %VBSCRIPT%
DEL %VBSCRIPT% /f /q

```

Via Windows Explorer



1. Navigate to the startup directory in Windows Explorer.

- User Startup/ directory (preferred) is located at:

%AppData%\Microsoft\Windows\Start Menu\Programs\Startup


- All Users Startup/ directory is located at:

%ProgramData%\Microsoft\Windows\Start Menu\Programs\Startup

2. Right-click inside the folder and select  → 

3. In the *Create Shortcut* dialog, enter the following information.

- **Location:** "C:\Program Files (x86)\PuTTY\pageant.exe"
- **Name:** Pageant Autoload

4. Right-click the new shortcut and choose  from the context menu.

5. Modify the following fields under the *Shortcut* tab:

- **Target:** "%PROGRAMFILES(x86)%\PuTTY\pageant.exe" id_rsa.ppk
- **Start in:** %USERPROFILE%\ssh

Notes:

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

2. If you placed your shortcut in the All Users startup directory, make sure that the current user has an `id_rsa.ppk` key in their `~/.ssh` directory or the key will not auto-load.

Closing Remarks

There you have it. Next time you log into your Windows profile, you will be greeted with a Pageant prompt to enter the password for your key. If you did not set a password on your key, then your key should be loaded automatically without a prompt.

If you are not sure if your key loaded view the current keys in Pageant by selecting **View Keys** from the context menu for Pageant in the system tray.

edited Mar 20 '17 at 10:04



Community ♦
1

answered Dec 10 '15 at 3:03



Mr. Polywhirl
339 3 11

3 Great detailed instructions! +1 from me! – [Pimp Juice IT](#) Dec 13 '15 at 1:40

Importing the private key in puttygen was my missing link – [Saurabh Kumar](#) Sep 28 '16 at 16:36

Had to create a connection in Putty to enable the step in the setup process from the picture. – [user46193](#) Aug 26 '17 at 18:12

Fantastic! I had a similar issue with gitlab.com and this resolved it. Git was using putty whereas the key I had uploaded was the OpenSSH one. Thank you! – [jgalak](#) May 26 '18 at 4:53

In plain English

4

debug3: Not a RSA1 key file /c/Users/Radu/.ssh/github.ppk.

debug2: key_type_from_name: unknown key type 'PuTTY-User-Key-File-2'

Puttygen can build different of keys, Github wants SSH1-RSA (? , I use ssh2 keys with Pageant on github)

Adds

See also [this post about debugging pageant issues with Github](#)

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

```
>plink.exe -v -agent git@github.com
Looking up host "github.com"
Connecting to 207.97.227.239 port 22
Server version: SSH-2.0-OpenSSH_5.1p1 Debian-5github2
Using SSH protocol version 2
We claim version: SSH-2.0-PuTTY_Release_0.62
Doing Diffie-Hellman group exchange
Doing Diffie-Hellman key exchange with hash SHA-256
Host key fingerprint is:
ssh-rsa 2048 16:27:ac:a5:76:28:2d:36:63:1b:56:4d:eb:df:a6:48
Initialised AES-256 SDCTR client->server encryption
Initialised HMAC-SHA1 client->server MAC algorithm
Initialised AES-256 SDCTR server->client encryption
Initialised HMAC-SHA1 server->client MAC algorithm
Pageant is running. Requesting keys.
Pageant has 1 SSH-2 keys
Using username "git".
Trying Pageant key #0
Remote debug message: Forced command: gerve lazybadger
Remote debug message: Port forwarding disabled.
Remote debug message: X11 forwarding disabled.
Remote debug message: Agent forwarding disabled.
Remote debug message: Pty allocation disabled.
Authenticating with public key "github/lazybadger" from agent
Sending Pageant's response
Remote debug message: Forced command: gerve lazybadger
Remote debug message: Port forwarding disabled.
Remote debug message: X11 forwarding disabled.
Remote debug message: Agent forwarding disabled.
Remote debug message: Pty allocation disabled.
Access granted
Opened channel for session
Server refused to allocate pty
Started a shell/command
Hi lazybadger! You've successfully authenticated, but GitHub does not provide shell
access.
Server sent command exit status 1
Disconnected: All channels closed
```

edited Jan 14 '12 at 17:00

answered Jan 14 '12 at 16:25



Lazy Badger

3,496 9 12

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).