

Using PuTTY As An SSH Client On Windows

These instructions should, and we all know what should means, work on most versions of Microsoft Windows, including XP, Vista, Windows 7, 8, 8.1 and 10. The goal of these notes is to allow a novice PuTTY user to make use of the SSH system management methodology to connect to and operate their Raspberry Pi (RPi).

The official PuTTY website is at:

<https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>



Illustration 1: PuTTY Official Website

The best idea is to download one of the MSI (windows installer) files, as it will contain all the PuTTY executables. They are:

- putty.exe (the SSH and Telnet client itself)
- pscp.exe (an SCP client, i.e. command-line secure file copy)
- psftp.exe (an SFTP client, i.e. general file transfer sessions much like FTP)
- plink.exe (a command-line interface to the PuTTY back ends)
- pageant.exe (an SSH authentication agent for PuTTY, PSCP, PSFTP, and Plink)
- puttygen.exe (a RSA and DSA key generation utility)

Since I have a 64-bit system, I will download the 64-bit set of binaries. The version as of this writing is 0.72. The download link is:

<https://the.earth.li/~sgtatham/putty/latest/w64/putty-64bit-0.72-installer.msi>

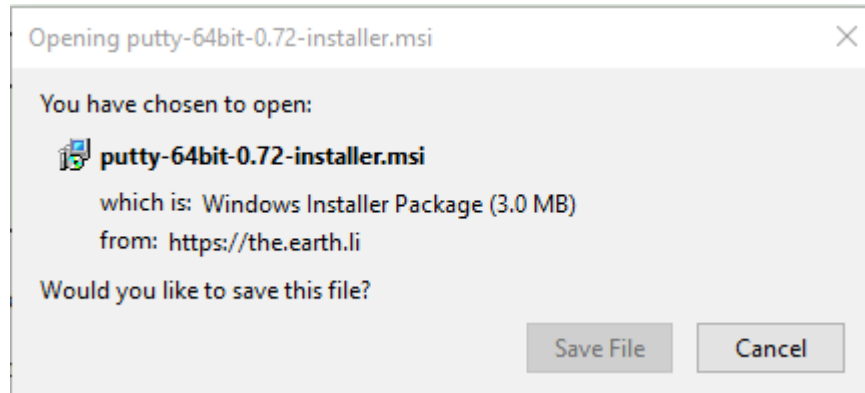


Illustration 2: PuTTY download pop-up

The file size is 3MB. It can usually be found in the Windows default Downloads folder. Alternatively, you will find it in your browser's default location for file downloads.

The documentation is fairly thorough and can be found Here:

<https://the.earth.li/~sgtatham/putty/0.72/html/doc/>

Running the Installer.

The downloaded file is the installer, here is a file manager view (note, I moved the installer to its own \ tmp folder within the Downloads folder, you don't need to do that):

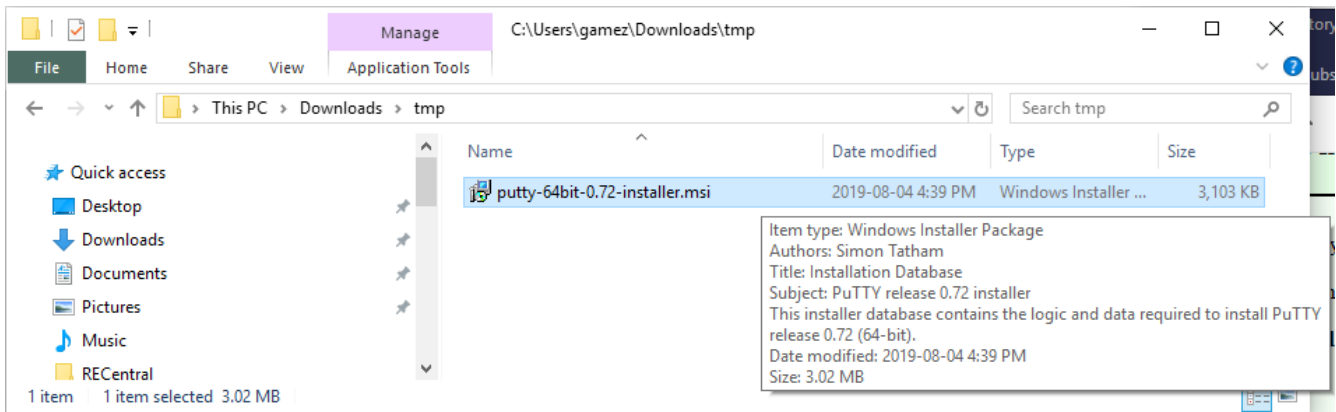


Illustration 3: PuTTY installer found in the Downloads\tmp folder

Double click the installer file. You'll see the following series of panels:

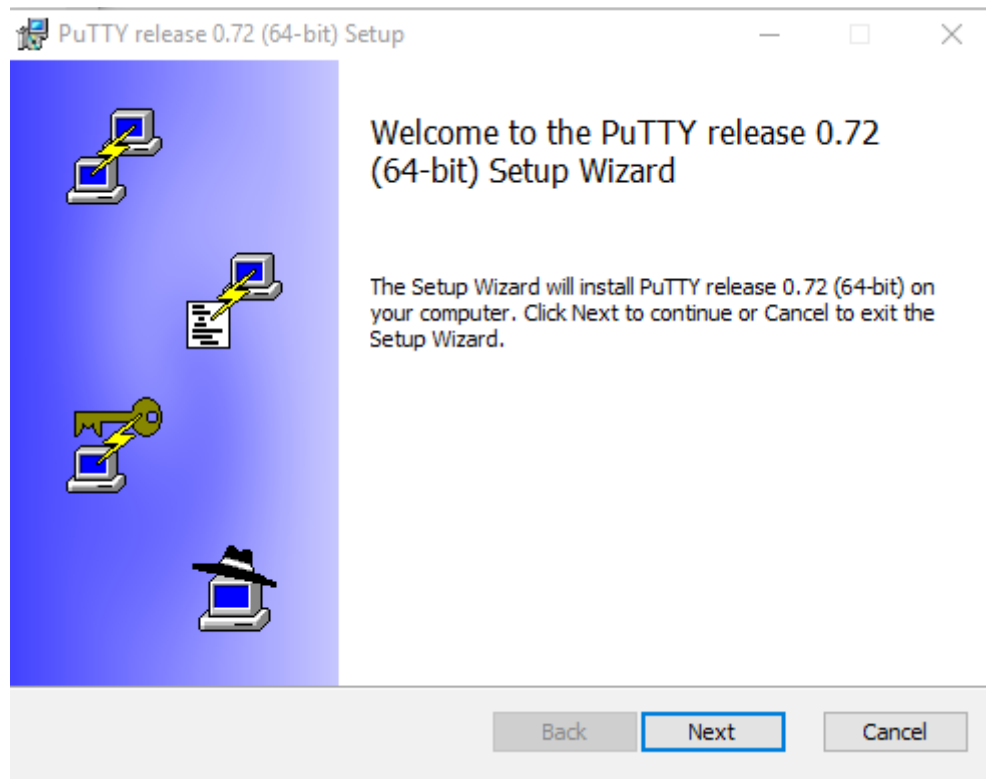


Illustration 4: PuTTY installer Welcome panel

Next:

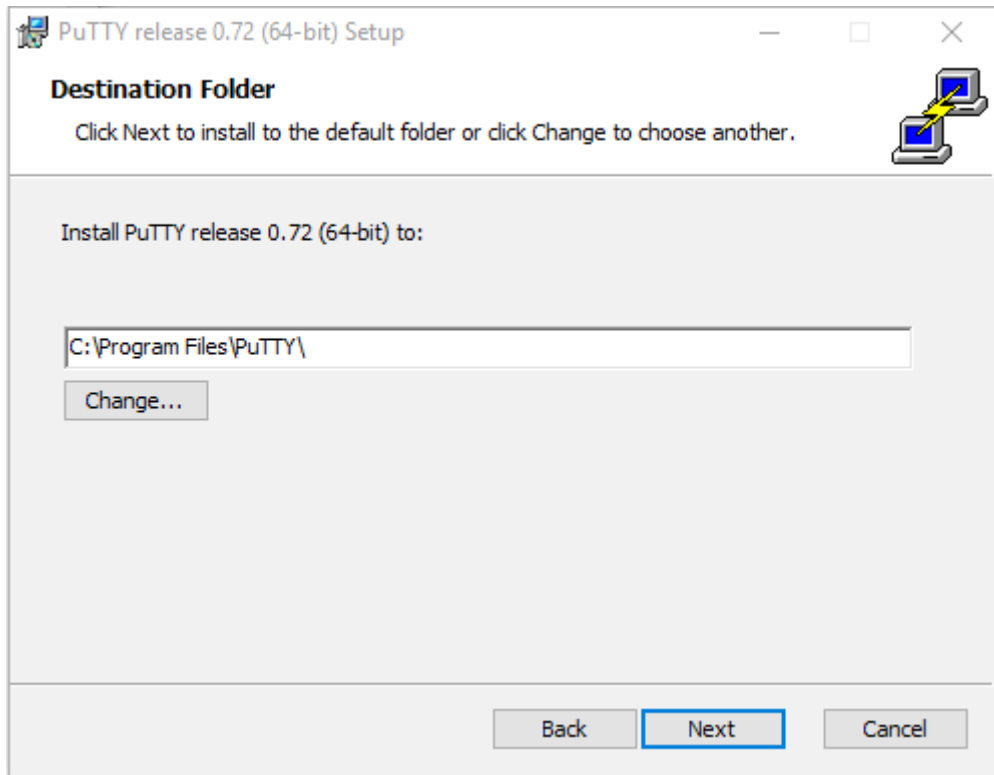


Illustration 5: PuTTY installer destination folder

Next:

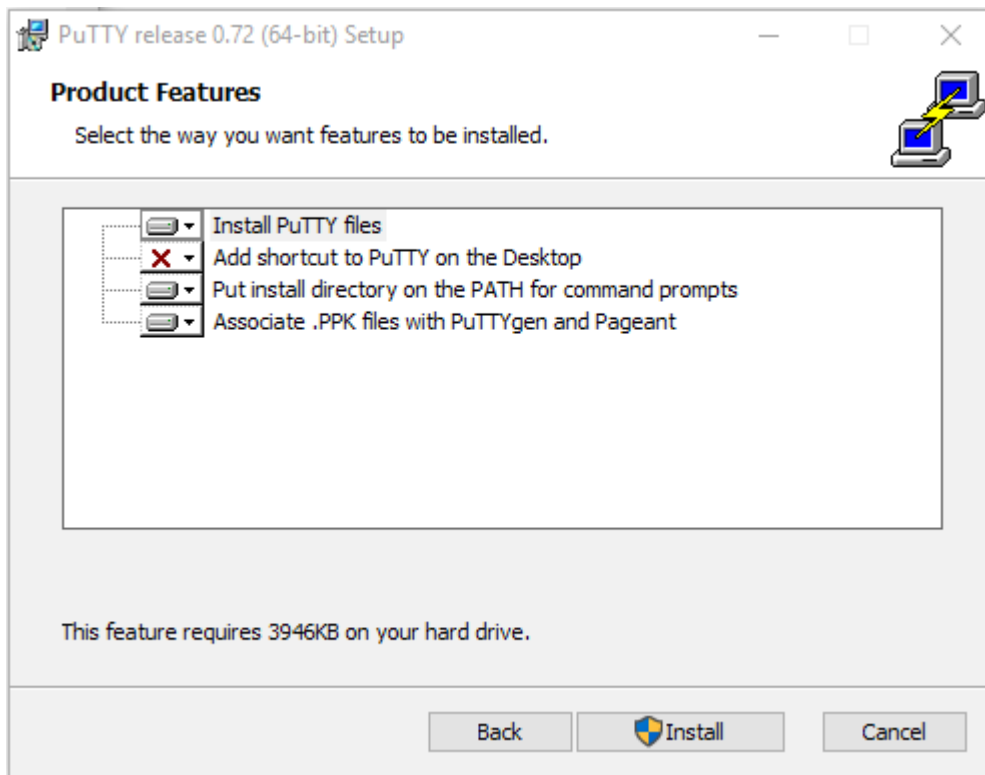


Illustration 6: PuTTY installer product features

Install:

Between steps 3 and 4, the installer asked for administrative privileges. It was granted. PuTTY is now installed.

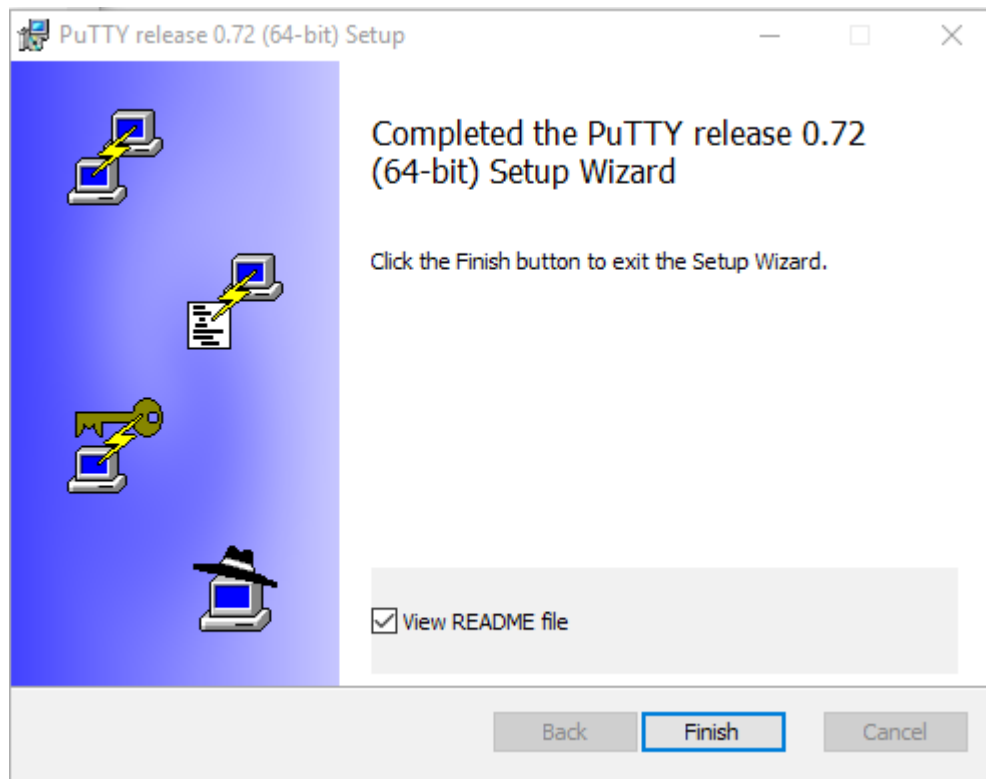


Illustration 7: PuTTY installation completed

Please skim the README.txt file for any useful information.

On my Windows 10 system, the Start Menu items for PuTTY are shown in the next image:

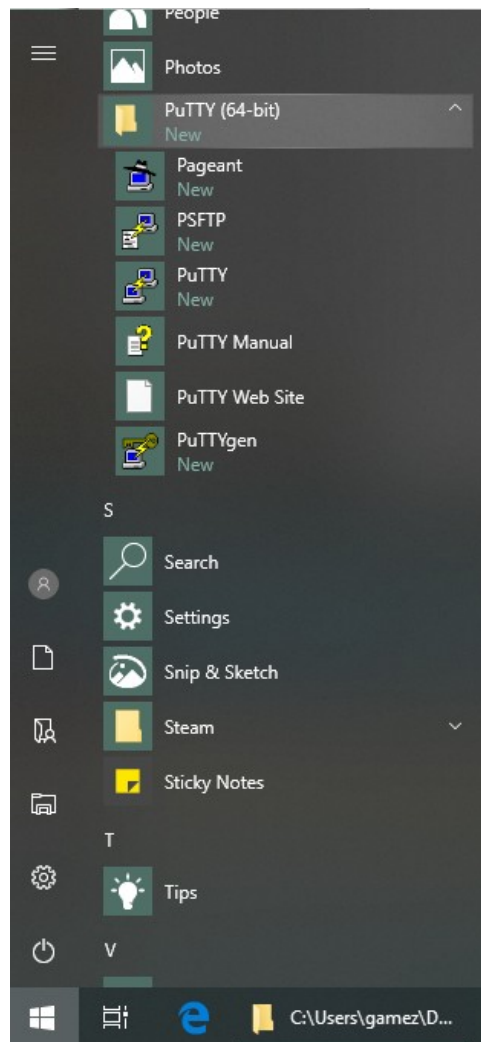


Illustration 8: PuTTY start menu items

Using PuTTY With A Password

To connect to a SSH server using password authentication, follow the steps below.

Start PuTTY and the PuTTY Configuration dialog opens.

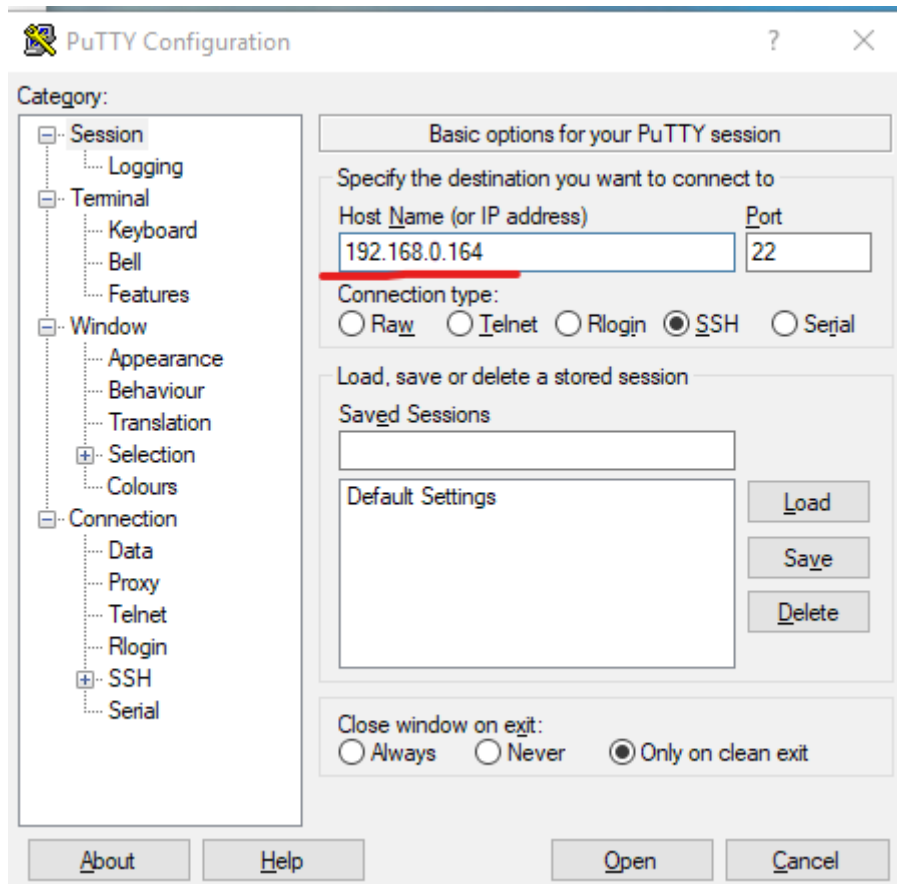


Illustration 9: The usual PuTTY startup screen showing an IP address entered by the user

On the left Category panel, the Session category is selected by default. On the right panel, basic options for a session are shown. A session is a connection from a client (you) to a server (the RPi), usually persistent.

Enter the IP address or host name of the SSH server to which you want to connect. See the red underlined value in the image above as an example.

Press the Open button near the bottom right of the PuTTY Configuration dialog to connect to the SSH server. At the first connection to an SSH server, you will receive this warning:

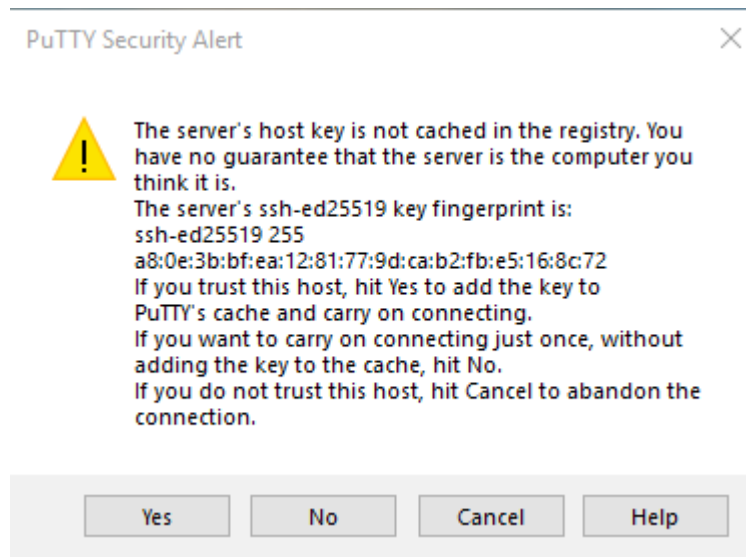


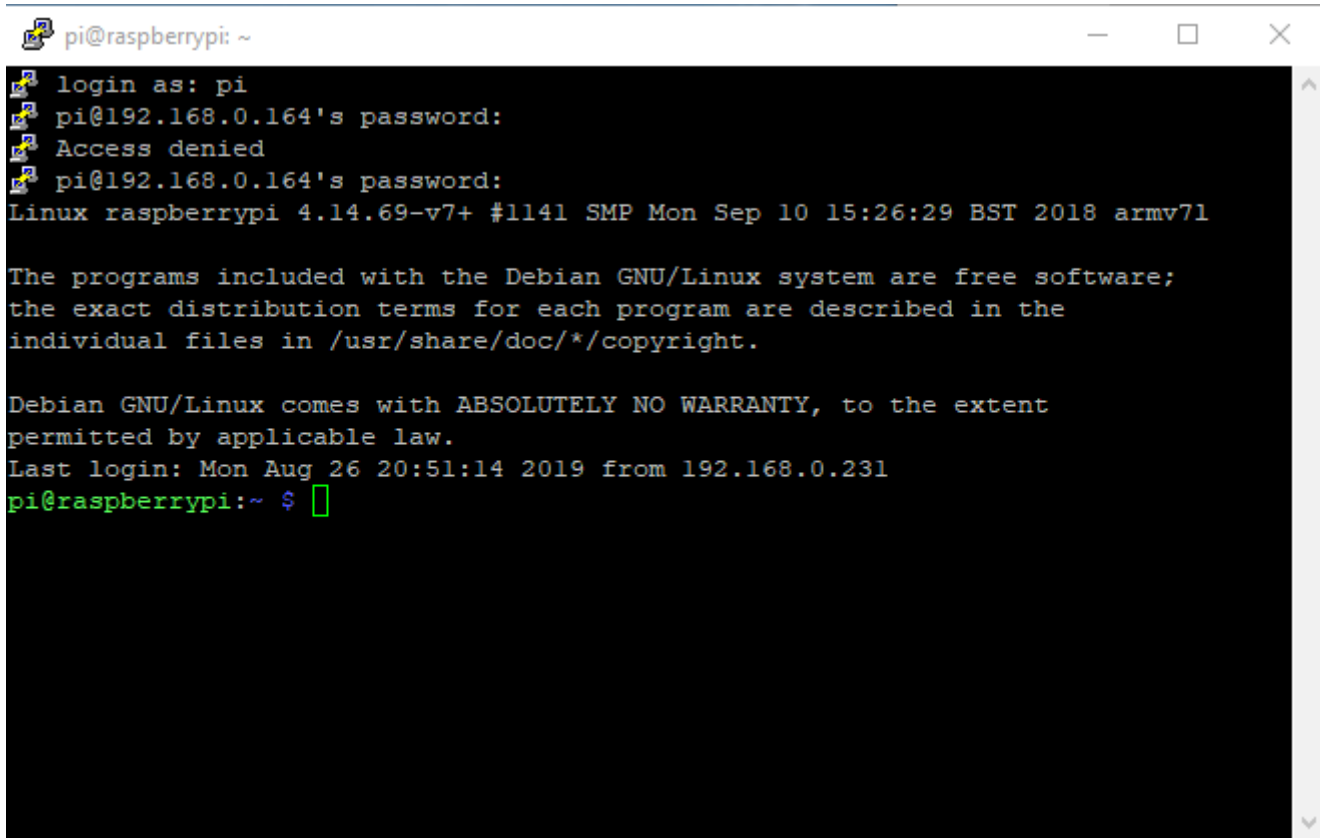
Illustration 10: Important Server Host Key Unknown or Changed Alert

It means that the server's identity key is unknown to your SSH client software (i.e. you) and that you should verify the server's identity. You should only ever accept this on the first connection to a server. If this pops up again for the same server, it can mean one of two things:

1. Someone has hijacked the network somewhere and has redirected your connection request to a server under their control. HIT CANCEL. Do not proceed.
2. The server host you use has legitimately changed it's server host key or IP address and you would know about this or would have been advised that this would happen.

This warning is the most important line of defense for man-in-the-middle attacks against SSH services. Always heed it with great seriousness and caution. You must find out exactly why the host key changed BEFORE you connect again. In our case here, this is the first connection and we can safely accept the server host key.

After pressing Yes, the PuTTY terminal will open and ask you which user you would like to login as, and then ask you for the user's password.



```
pi@raspberrypi: ~  
login as: pi  
pi@192.168.0.164's password:  
Access denied  
pi@192.168.0.164's password:  
Linux raspberrypi 4.14.69-v7+ #1141 SMP Mon Sep 10 15:26:29 BST 2018 armv7l  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Mon Aug 26 20:51:14 2019 from 192.168.0.231  
pi@raspberrypi:~ $
```

Illustration 11: PuTTY terminal login with username and password prompt

That's all that is needed for a simple password based connection. Keep this terminal open, since you will use it in the next section of these instructions.

Using PuTTY With Fancy Keys

The SSH protocol lives and breathes with fancy, cryptographic keys. It is a best practice to use them almost always.

Generating a private/public key pair with PuTTYgen

The PuTTYgen program is used to generate cryptographic key pairs for logging into SSH servers. Basically, the private key is the 'door key' and the public key, stored on the server, is the 'door lock'. You store the private key, protected by a passphrase, on your own computer. The server uses the public key, stored in a special file within your home directory on the server. This file is called `authorized_users` and is stored in the `/home/you/.ssh/` directory on the RPi.

When you generate the key pair, you'll have to copy the public key to the server using either a physical USB file transfer, or a one-off SSH password connection (like the one we did above), prior to disabling password logins. To generate a key pair, start the PuTTYgen program, found in the start menu at:

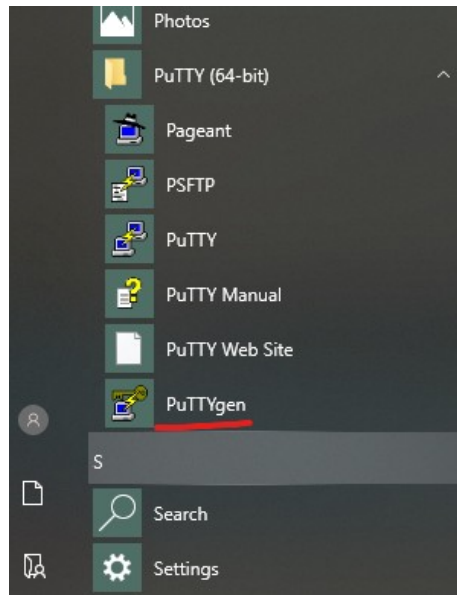


Illustration 12: PuTTYgen start menu item

PuTTYgen is underlined in red in figure 12. Once started, you will see the blank key generator panel.

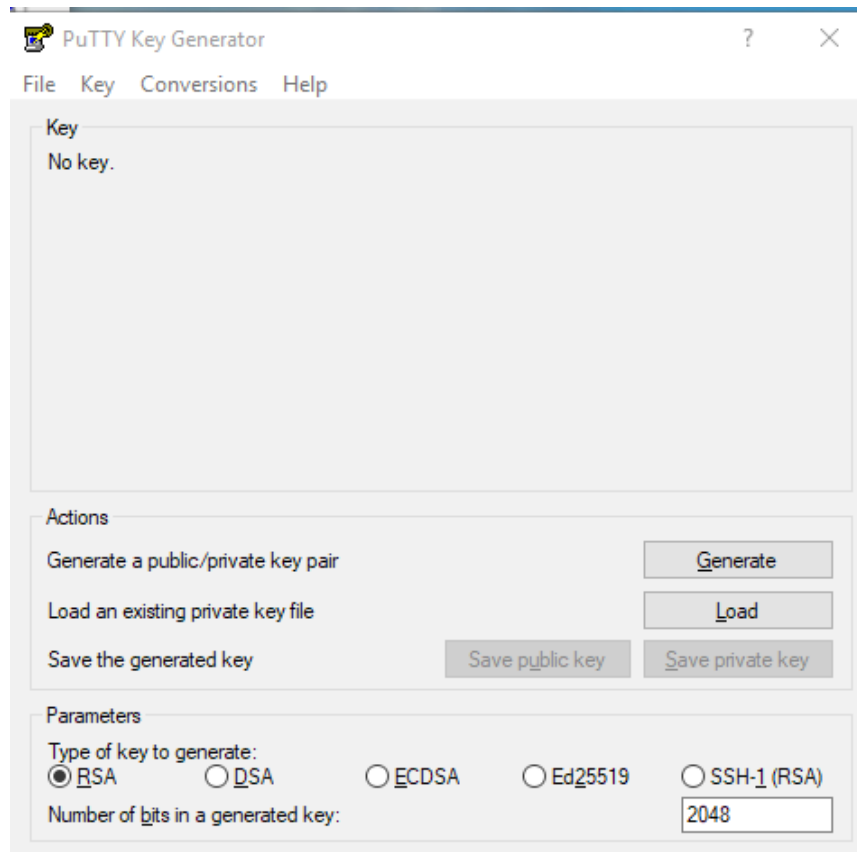


Illustration 13: PuTTYgen blank startup panel

To start, make sure at least a 2048 bit RSA key (bottom of the panel, default parameters), is selected and then click Generate. The program will require you to wiggle the mouse around the large blank area in the panel to generate randomness. It will then produce a screen similar to the next image:

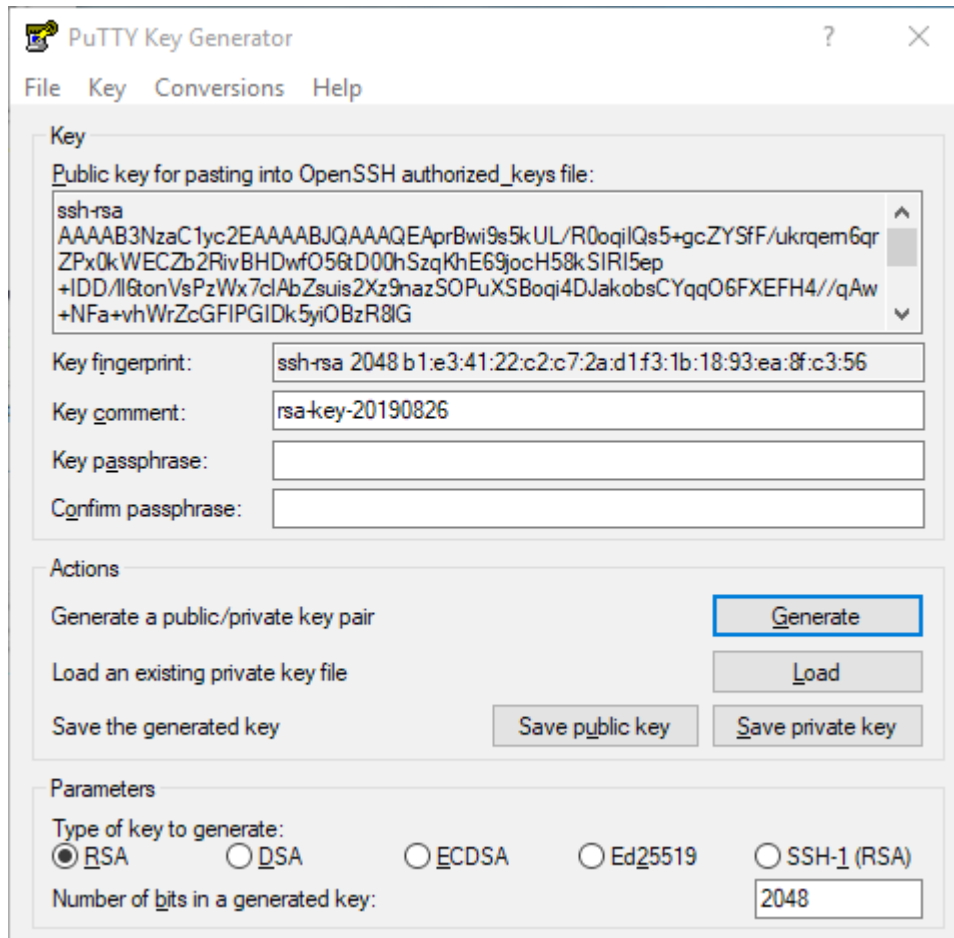


Illustration 14: PuTTYgen key pair generated, but no passphrase set yet

Note the empty "Key passphrase" field. You now need to fill it in with a strong phrase / password, along with the Confirm passphrase textbox. Optionally type in an useful "Key comment" in the text box which you can use to remind yourself in the future what the key is used for. Click "Save private key" (highlighted in light blue), and give it a functional file name (servername, usually).

As for the public key, it is right at the top of the panel under "Public key for pasting into OpenSSH authorized_keys file". We will use this. To get it onto the clipboard, use the keyboard. Holding the Alt key and then pressing the P key will select the public key's text. Use Control-C to copy the key to the clipboard. You can also left click on the text area to select the public key.

Now navigate to the PuTTY terminal you logged into via password earlier, and execute the following commands:

```
pi@raspberrypi:~ $ ls -la .ssh
ls: cannot access '.ssh': No such file or directory
```

The above means you don't have an `.ssh` folder yet. If you get an actual listing of the folder, skip the next step.

Create the `.ssh` folder and give it the proper permissions:

```
pi@raspberrypi:~ $ mkdir .ssh
pi@raspberrypi:~ $ chmod 0700 .ssh
```

Change to / enter the `.ssh` directory:

```
pi@raspberrypi:~ $ cd .ssh
```

Using your favourite editor, create a text file named `authorized_keys`:

```
pi@raspberrypi:~/ssh $ vi authorized_keys
```

Paste the Public key from the clipboard into the file, as a single line (this is important). With the vi editor, you must use Shift-Insert to paste the public key into the file. Save the `authorized_keys` file.

Close PuTTYgen, optionally saving the public key beforehand.

Testing the key

Open PuTTY to try a new session (still keeping your current terminal session open). Type in the host name or IP address as before. This time, we also modify the Connection Category's SSH sub-category's Auth section (see figure below). There is a file text box that allows a "Private key file for authentication" to be selected by the Browse mechanism. Click browse and choose the private key file you saved earlier. You will see the key path in the PuTTY Auth Configuration.

Left panel: Category → Connection → SSH → Auth

The image below doesn't actually show a private key file in the box labelled "Private key file for authentication" (but don't let this bother you).

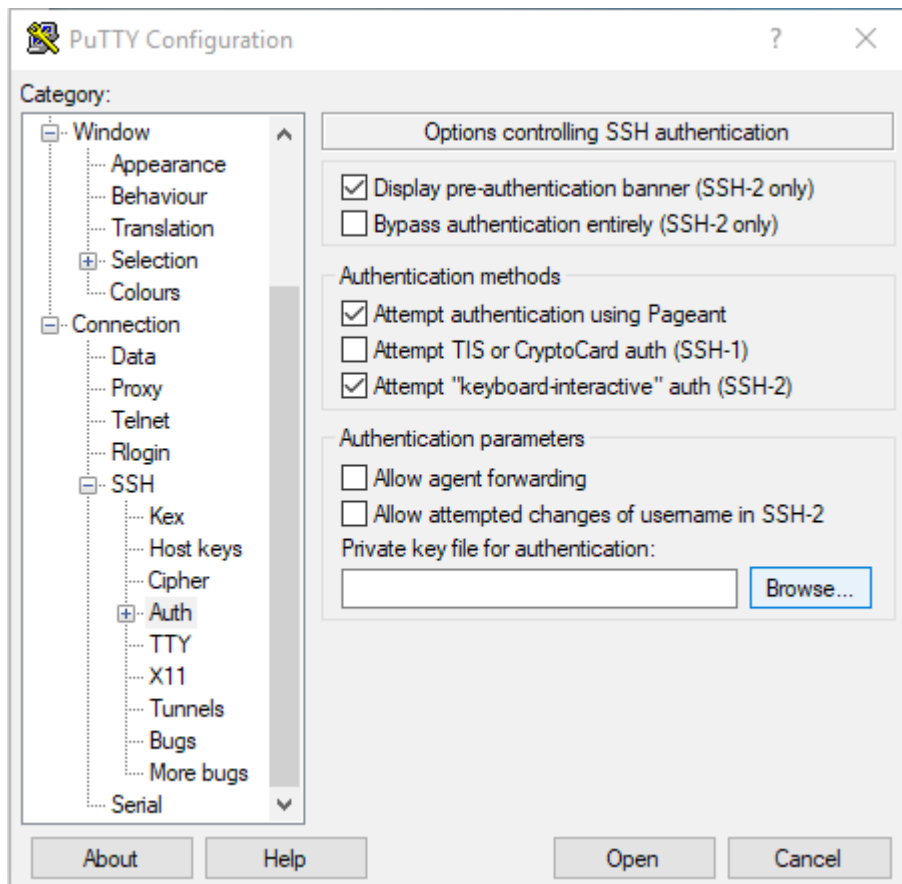


Illustration 15: PuTTY and where to set / select the private key

If you hit Open after selecting a key, a connection will be established and you will be asked to provide the username and private key passphrase. However, you will have to reselect the key each time you wish to connect. This is annoying and is fixed by saving the session parameters before you press Open.

Saving A Session

We have to type the IP address in again. Also, reselect the private key file, do not press Open, but instead return to the Session Category. You'll see a set of controls for loading, saving and deleting stored sessions. Type a name into the blank one line text box under Saved Sessions and press Save to store the session. This saves the private key path (but not the passphrase, of course) that you entered in the Auth section, as well as the IP address and any other parameters you may have set.

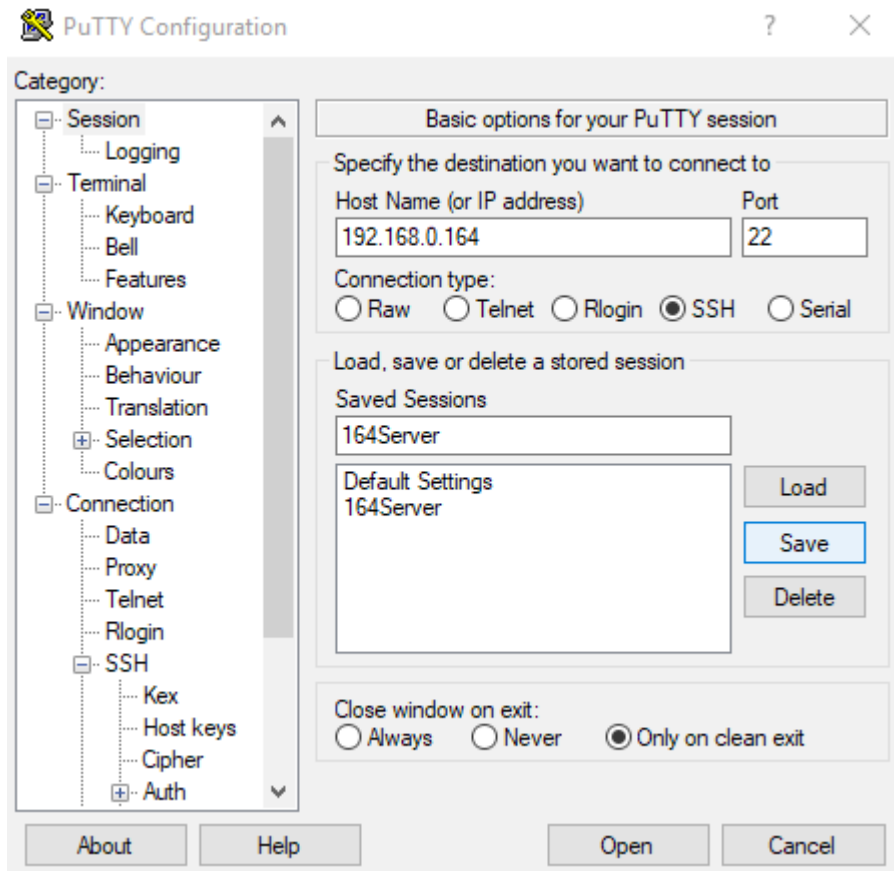
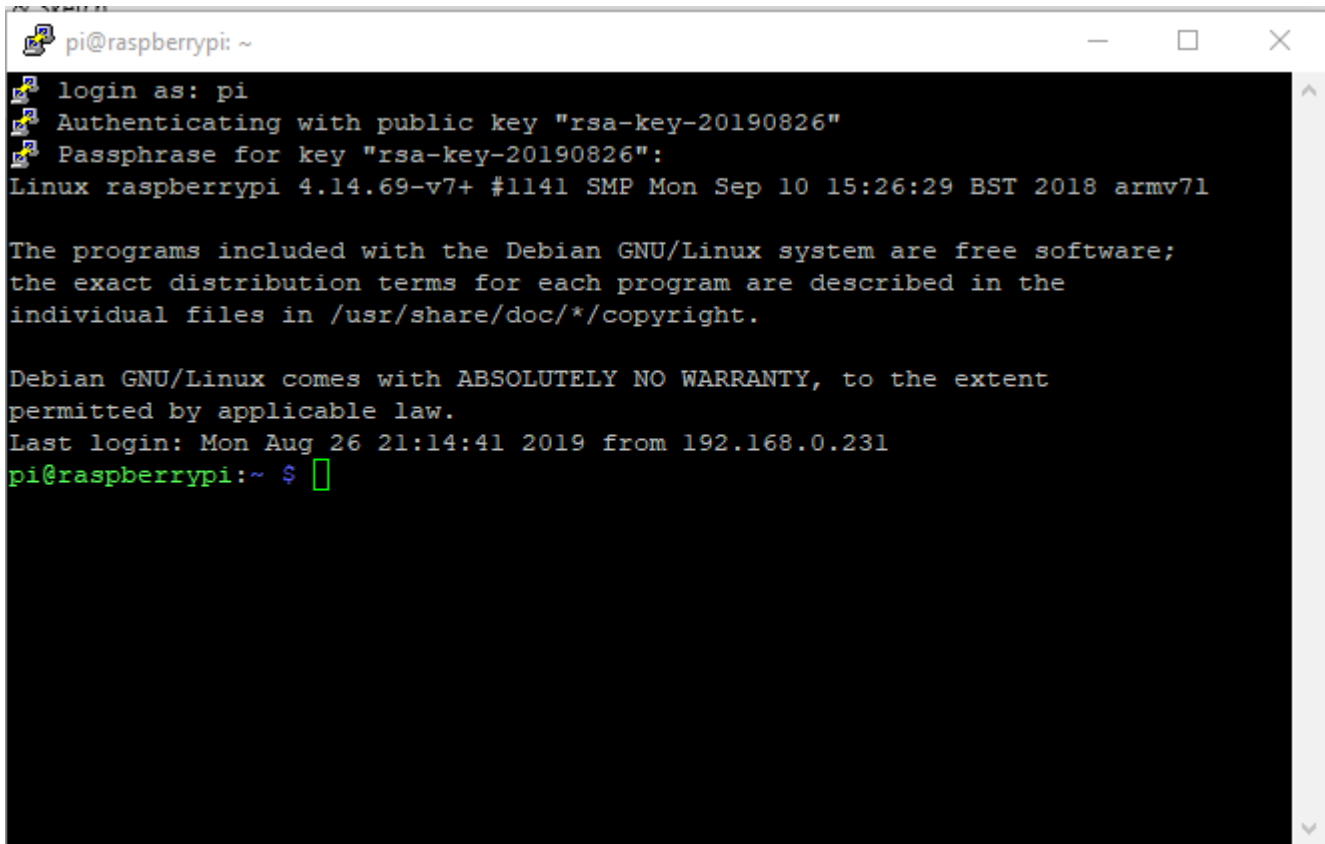


Illustration 16: PuTTY and saving a session for future connection ease

You can now easily use Load to select this server session in the future.

Now click Open, and you'll be prompted to enter a username, and then the private key passphrase.



```
pi@raspberrypi: ~  
login as: pi  
Authenticating with public key "rsa-key-20190826"  
Passphrase for key "rsa-key-20190826":  
Linux raspberrypi 4.14.69-v7+ #1141 SMP Mon Sep 10 15:26:29 BST 2018 armv7l  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Mon Aug 26 21:14:41 2019 from 192.168.0.231  
pi@raspberrypi:~ $
```

Illustration 17: PuTTY using a private key to login to a server

Having to enter the username every time is boring. You can optionally store the username as part of the saved session under the Category: Connection->Data, auto-login username text box as shown in the next image.

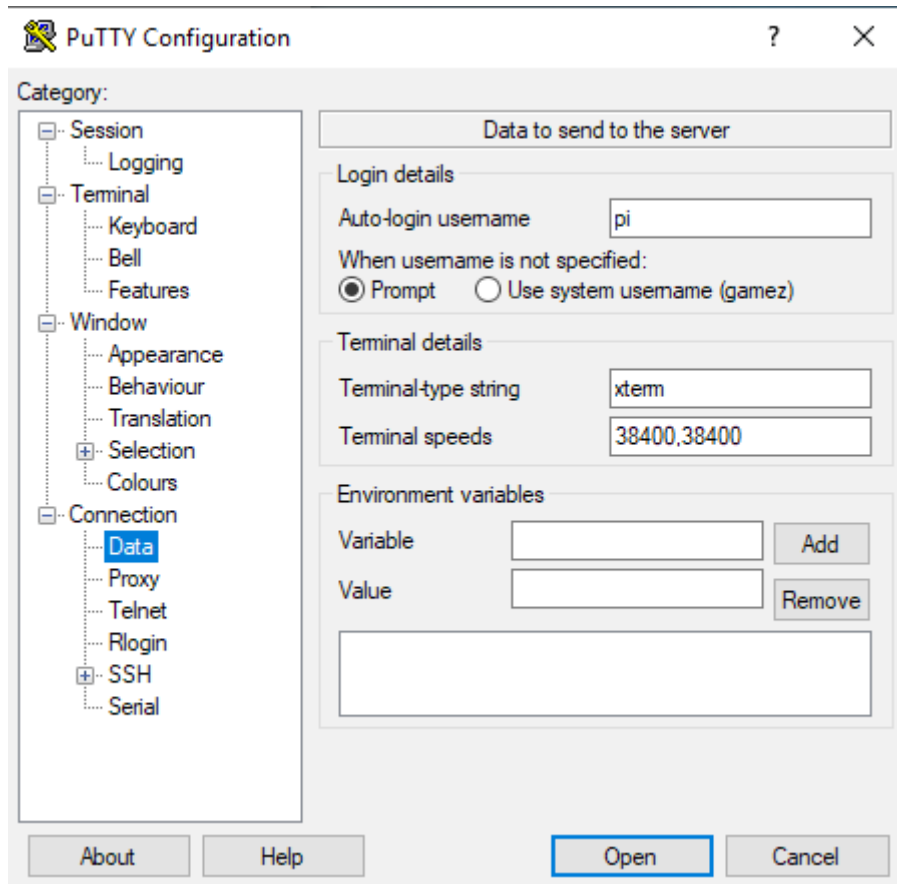
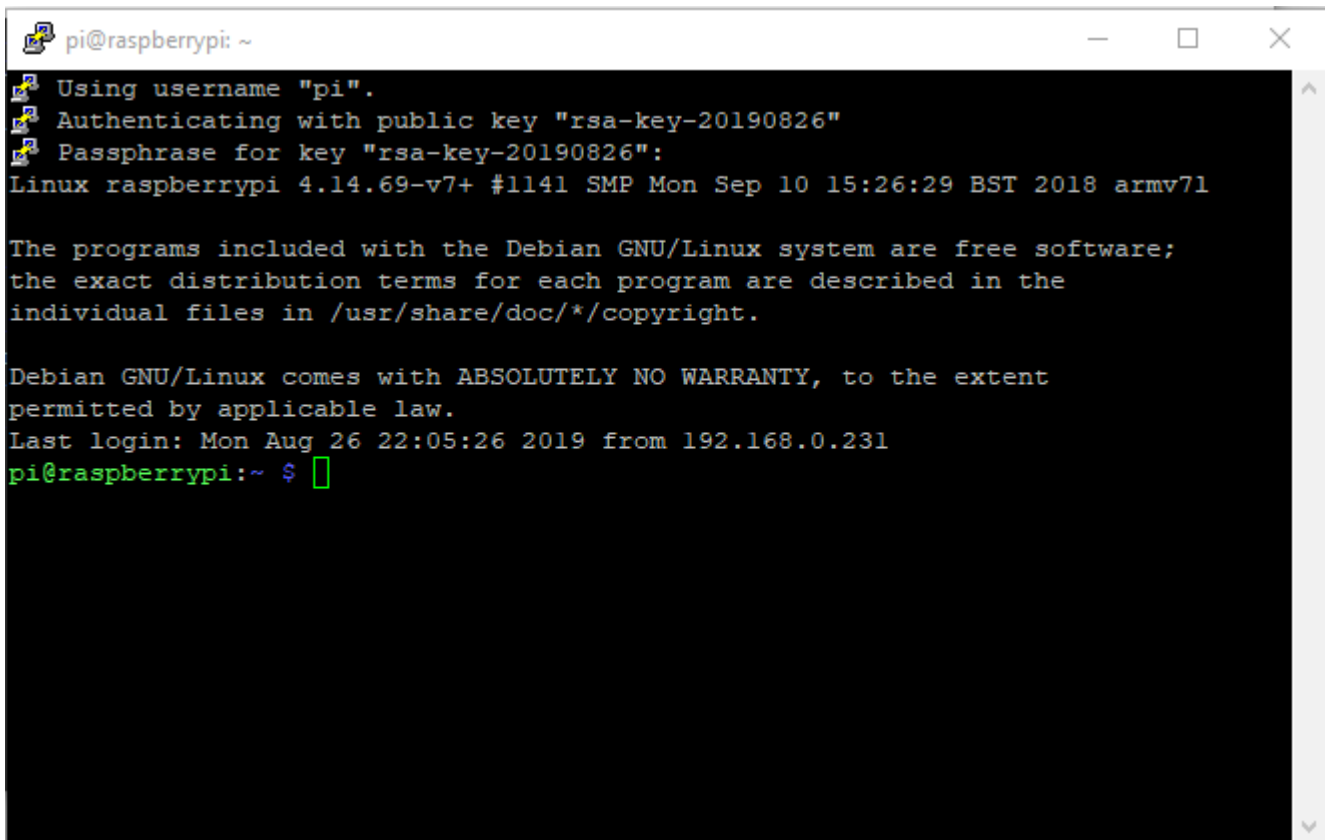


Illustration 18: PuTTY setting the session auto-login username to pi

WARNING! Don't forget to return to the Session category and re-save the stored session before you hit Open, otherwise the username entry won't be preserved. Here is the auto-login terminal:



```
pi@raspberrypi: ~
Using username "pi".
Authenticating with public key "rsa-key-20190826"
Passphrase for key "rsa-key-20190826":
Linux raspberrypi 4.14.69-v7+ #1141 SMP Mon Sep 10 15:26:29 BST 2018 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Aug 26 22:05:26 2019 from 192.168.0.231
pi@raspberrypi:~ $
```

Illustration 19: PuTTY using auto-login and private keys with a passphrase

You can now disable password based logins (a lesson for another day).

PuTTY SCP

Excellent documentation for PuTTY SCP (`pscp`) exists here:

<https://the.earth.li/~sgtatham/putty/0.72/html/doc/Chapter5.html#pscp>

To copy files from the RPi to your windows system, you can use the PuTTY SCP program named `pscp`. You have to open a Command Prompt window to do so. The command prompt is found under the "Windows System" folder in the start menu.

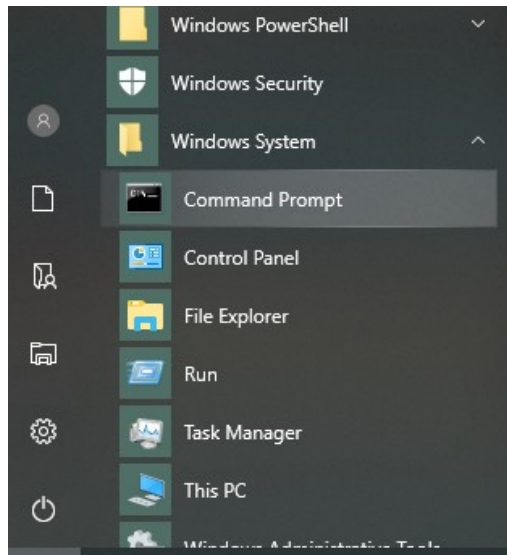


Illustration 20: Windows command prompt start menu item

The command prompt window looks like this:

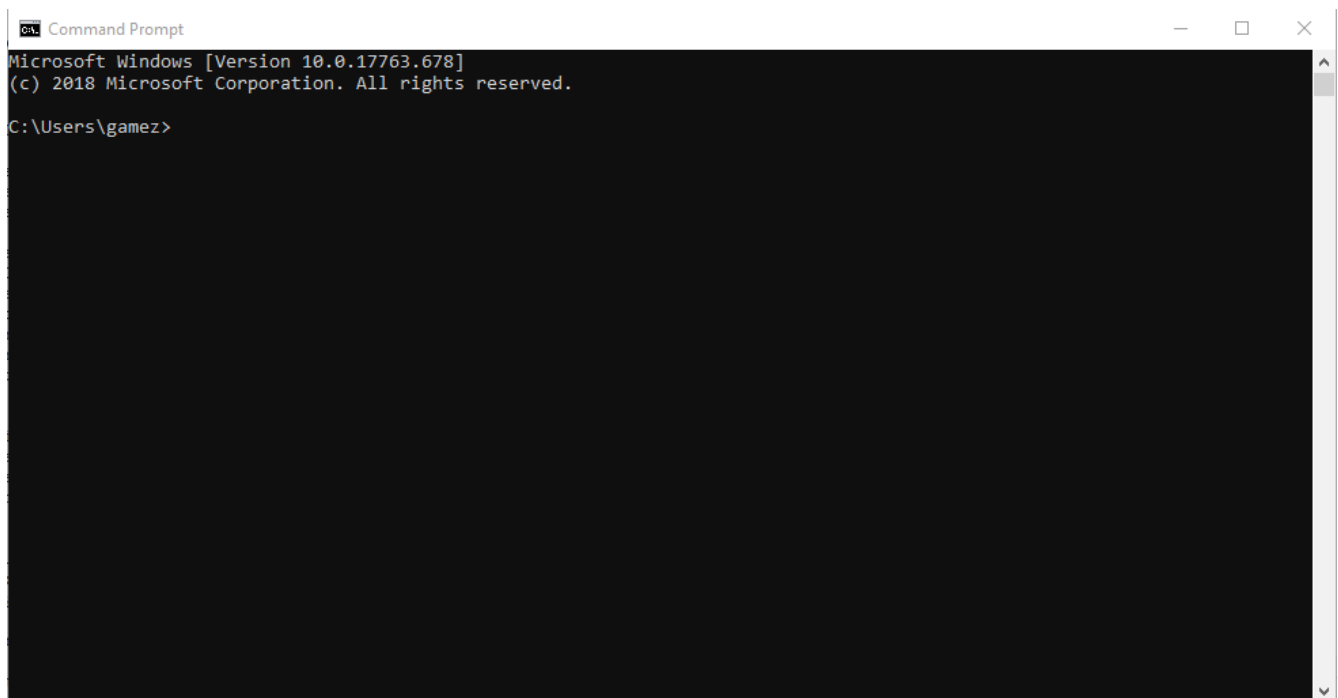


Illustration 21: Windows command prompt - similar to, but different than the PuTTY terminal

At startup, the command prompt working directory is your user home directory on windows. The command prompt uses DOS commands, so you'll need to be familiar with a couple of them. Some useful ones are:

- cd – change direcotry

- `mkdir` – create a directory
- `dir` – list a directory

Create a folder to store your files. I created a folder called `dev` located in my windows user home directory (`Users\gamez\`).

```
C:\Users\gamez>mkdir dev
C:\Users\gamez>cd dev
C:\Users\gamez\dev>
```

Now use the `pscp` command with the private key specified with the `-i` option. Also, note the relative path name to the private key file.

I am assuming the files are stored on the RPi in a directory similarly named `dev` in the user `pi`'s home directory. We will also copy the files recursively, using the `-r` option, which means descend into subdirectories. Note there is a dot `'.'` at the end of the `pscp` command, and also note that the command is displayed wrapped around, but should be on a single line.

```
C:\Users\gamez\dev>pscp -r -i ..\Documents\164server.ppk
pi@192.168.0.164:/home/pi/dev/* .
Passphrase for key "rsa-key-20190826":
C:\Users\gamez\dev>
```

Now we check to see if all files were copied:

```
C:\Users\gamez\dev>dir /s
Volume in drive C has no label.
Volume Serial Number is FED8-50A4
Directory of C:\Users\gamez\dev
2019-08-28  01:05 PM    <DIR>          .
2019-08-28  01:05 PM    <DIR>          ..
2019-08-28  01:05 PM                0 foo.c
2019-08-28  01:05 PM    <DIR>          unit1
```

1 File(s) 0 bytes

Directory of C:\Users\gamez\dev\unit1

2019-08-28	01:05 PM	<DIR>	.
2019-08-28	01:05 PM	<DIR>	..
2019-08-28	01:05 PM		0 bar.c

1 File(s) 0 bytes

That's it! Very easy.