Previous | Contents | Index | Next

Chapter 10: Common error messages

This chapter lists a number of common error messages which PuTTY and its associated tools can produce, and explains what they mean in more detail.

We do not attempt to list *all* error messages here: there are many which should never occur, and some which should be self-explanatory. If you get an error message which is not listed in this chapter and which you don't understand, report it to us as a bug (see appendix B) and we will add documentation for it.

10.1 ‘The server's host key is not cached in the registry’

This error message occurs when PuTTY connects to a new SSH server. Every server identifies itself by means of a host key; once PuTTY knows the host key for a server, it will be able to detect if a malicious attacker redirects your connection to another machine.

If you see this message, it means that PuTTY has not seen this host key before, and has no way of knowing whether it is correct or not. You should attempt to verify the host key by other means, such as asking the machine's administrator.

If you see this message and you know that your installation of PuTTY *has* connected to the same server before, it may have been recently upgraded to SSH protocol version 2. SSH protocols 1 and 2 use separate host keys, so when you first use SSH-2 with a server you have only used SSH-1 with before, you will see this message again. You should verify the correctness of the key as before.

See section 2.2 for more information on host keys.

10.2 ‘WARNING - POTENTIAL SECURITY BREACH!’

This message, followed by ‘The server's host key does not match the one PuTTY has cached in the registry’, means that PuTTY has connected to the SSH server before, knows what its host key *should* be, but has found a different one.

This may mean that a malicious attacker has replaced your server with a different one, or has redirected your network connection to their own machine. On the other hand, it may simply mean that the administrator of your server has accidentally changed the key while upgrading the SSH software; this *shouldn't* happen but it is unfortunately possible.

You should contact your server's administrator and see whether they expect the host key to have changed. If so, verify the new host key in the same way as you would if it was new.

See section 2.2 for more information on host keys.

10.3 ‘Out of space for port forwardings’

PuTTY has a fixed-size buffer which it uses to store the details of all port forwardings you have set up in an SSH session. If you specify too many port forwardings on the PuTTY or Plink command line and this buffer becomes full, you will see this error message.

We need to fix this (fixed-size buffers are almost always a mistake) but we haven't got round to it. If you actually have trouble with this, let us know and we'll move it up our priority list.

10.4 ‘The first cipher supported by the server is ... below the configured warning threshold’

This occurs when the SSH server does not offer any ciphers which you have configured PuTTY to consider strong enough. By default, PuTTY puts up this warning only for single-DES and Arcfour encryption.

See section 4.18.5 for more information on this message.

10.5 ‘Server sent disconnect message type 2 (protocol error): "Too many authentication failures for root"’

This message is produced by an OpenSSH (or Sun SSH) server if it receives more failed authentication attempts than it is willing to tolerate.

This can easily happen if you are using Pageant and have a large number of keys loaded into it, since these servers count each offer of a public key as an authentication attempt. This can be worked around by specifying the key that's required for the authentication in the PuTTY configuration (see section 4.20.7); PuTTY will ignore any other keys Pageant may have, but will ask Pageant to do the authentication, so that you don't have to type your passphrase.

On the server, this can be worked around by disabling public-key authentication or (for Sun SSH only) by increasing `MaxAuthTries` in `sshd_config`.

10.6 ‘Out of memory’

This occurs when PuTTY tries to allocate more memory than the system can give it. This *may* happen for genuine reasons: if the computer really has run out of memory, or if you have configured an extremely large

number of lines of scrollback in your terminal. PuTTY is not able to recover from running out of memory; it will terminate immediately after giving this error.

However, this error can also occur when memory is not running out at all, because PuTTY receives data in the wrong format. In SSH-2 and also in SFTP, the server sends the length of each message before the message itself; so PuTTY will receive the length, try to allocate space for the message, and then receive the rest of the message. If the length PuTTY receives is garbage, it will try to allocate a ridiculous amount of memory, and will terminate with an 'Out of memory' error.

This can happen in SSH-2, if PuTTY and the server have not enabled encryption in the same way (see question A.7.5 in the FAQ). Some versions of OpenSSH have a known problem with this: see question A.7.16.

This can also happen in PSCP or PSFTP, if your login scripts on the server generate output: the client program will be expecting an SFTP message starting with a length, and if it receives some text from your login scripts instead it will try to interpret them as a message length. See question A.7.6 for details of this.

10.7 'Internal error', 'Internal fault', 'Assertion failed'

Any error beginning with the word 'Internal' should *never* occur. If it does, there is a bug in PuTTY by definition; please see appendix B and report it to us.

Similarly, any error message starting with 'Assertion failed' is a bug in PuTTY. Please report it to us, and include the exact text from the error message box.

10.8 'Unable to use this private key file', 'Couldn't load private key', 'Key is of wrong type'

Various forms of this error are printed in the PuTTY window, or written to the PuTTY Event Log (see section 3.1.3.1) when trying public-key authentication, or given by Pageant when trying to load a private key.

If you see one of these messages, it often indicates that you've tried to load a key of an inappropriate type into PuTTY, Plink, PSCP, PSFTP, or Pageant.

You may have specified a key that's inappropriate for the connection you're making. The SSH-1 and SSH-2 protocols require different private key formats, and a SSH-1 key can't be used for a SSH-2 connection (or vice versa).

Alternatively, you may have tried to load an SSH-2 key in a 'foreign' format (OpenSSH or `ssh.com`) directly into one of the PuTTY tools, in which case you need to import it into PuTTY's native format (`*.PPK`) using PuTTYgen - see section 8.2.12.

10.9 'Server refused our public key' or 'Key refused'

Various forms of this error are printed in the PuTTY window, or written to the PuTTY Event Log (see section 3.1.3.1) when trying public-key authentication.

If you see one of these messages, it means that PuTTY has sent a public key to the server and offered to authenticate with it, and the server has refused to accept authentication. This usually means that the server is not configured to accept this key to authenticate this user.

This is almost certainly not a problem with PuTTY. If you see this type of message, the first thing you should do is check your *server* configuration carefully. Common errors include having the wrong permissions or ownership

set on the public key or the user's home directory on the server. Also, read the PuTTY Event Log; the server may have sent diagnostic messages explaining exactly what problem it had with your setup.

10.10 'Access denied', 'Authentication refused'

Various forms of this error are printed in the PuTTY window, or written to the PuTTY Event Log (see section 3.1.3.1) during authentication.

If you see one of these messages, it means that the server has refused all the forms of authentication PuTTY has tried and it has no further ideas.

It may be worth checking the Event Log for diagnostic messages from the server giving more detail.

This error can be caused by buggy SSH-1 servers that fail to cope with the various strategies we use for camouflaging passwords in transit. Upgrade your server, or use the workarounds described in section 4.24.1 and possibly section 4.24.2.

10.11 'Incorrect CRC received on packet' or 'Incorrect MAC received on packet'

This error occurs when PuTTY decrypts an SSH packet and its checksum is not correct. This probably means something has gone wrong in the encryption or decryption process. It's difficult to tell from this error message whether the problem is in the client, in the server, or in between.

A known server problem which can cause this error is described in question A.7.16 in the FAQ.

10.12 'Incoming packet was garbled on decryption'

This error occurs when PuTTY decrypts an SSH packet and the decrypted data makes no sense. This probably means something has gone wrong in the encryption or decryption process. It's difficult to tell from this error message whether the problem is in the client, in the server, or in between.

If you get this error, one thing you could try would be to fiddle with the setting of 'Miscomputes SSH-2 encryption keys' on the Bugs panel (see section 4.24.5).

Another known server problem which can cause this error is described in question A.7.16 in the FAQ.

10.13 'PuTTY X11 proxy: *various errors*'

This family of errors are reported when PuTTY is doing X forwarding. They are sent back to the X application running on the SSH server, which will usually report the error to the user.

When PuTTY enables X forwarding (see section 3.4) it creates a virtual X display running on the SSH server. This display requires authentication to connect to it (this is how PuTTY prevents other users on your server machine from connecting through the PuTTY proxy to your real X display). PuTTY also sends the server the details it needs to enable clients to connect, and the server should put this mechanism in place automatically, so your X applications should just work.

A common reason why people see one of these messages is because they used SSH to log in as one user (let's say 'fred'), and then used the Unix `su` command to become another user (typically 'root'). The original user, 'fred', has access to the X authentication data provided by the SSH server, and can run X applications which are

forwarded over the SSH connection. However, the second user ('root') does not automatically have the authentication data passed on to it, so attempting to run an X application as that user often fails with this error.

If this happens, *it is not a problem with PuTTY*. You need to arrange for your X authentication data to be passed from the user you logged in as to the user you used `su` to become. How you do this depends on your particular system; in fact many modern versions of `su` do it automatically.

10.14 'Network error: Software caused connection abort'

This is a generic error produced by the Windows network code when it kills an established connection for some reason. For example, it might happen if you pull the network cable out of the back of an Ethernet-connected computer, or if Windows has any other similar reason to believe the entire network has become unreachable.

Windows also generates this error if it has given up on the machine at the other end of the connection ever responding to it. If the network between your client and server goes down and your client then tries to send some data, Windows will make several attempts to send the data and will then give up and kill the connection. In particular, this can occur even if you didn't type anything, if you are using SSH-2 and PuTTY attempts a key re-exchange. (See section 4.19.2 for more about key re-exchange.)

(It can also occur if you are using keepalives in your connection. Other people have reported that keepalives *fix* this error for them. See section 4.13.1 for a discussion of the pros and cons of keepalives.)

We are not aware of any reason why this error might occur that would represent a bug in PuTTY. The problem is between you, your Windows system, your network and the remote system.

10.15 'Network error: Connection reset by peer'

This error occurs when the machines at each end of a network connection lose track of the state of the connection between them. For example, you might see it if your SSH server crashes, and manages to reboot fully before you next attempt to send data to it.

However, the most common reason to see this message is if you are connecting through a firewall or a NAT router which has timed the connection out. See question A.7.10 in the FAQ for more details. You may be able to improve the situation by using keepalives; see section 4.13.1 for details on this.

Note that Windows can produce this error in some circumstances without seeing a connection reset from the server, for instance if the connection to the network is lost.

10.16 'Network error: Connection refused'

This error means that the network connection PuTTY tried to make to your server was rejected by the server. Usually this happens because the server does not provide the service which PuTTY is trying to access.

Check that you are connecting with the correct protocol (SSH, Telnet or Rlogin), and check that the port number is correct. If that fails, consult the administrator of your server.

10.17 'Network error: Connection timed out'

This error means that the network connection PuTTY tried to make to your server received no response at all from the server. Usually this happens because the server machine is completely isolated from the network, or because it is turned off.

Check that you have correctly entered the host name or IP address of your server machine. If that fails, consult the administrator of your server.

Unix also generates this error when it tries to send data down a connection and contact with the server has been completely lost during a connection. (There is a delay of minutes before Unix gives up on receiving a reply from the server.) This can occur if you type things into PuTTY while the network is down, but it can also occur if PuTTY decides of its own accord to send data: due to a repeat key exchange in SSH-2 (see section 4.19.2) or due to keepalives (section 4.13.1).

If you want to provide feedback on this manual or on the PuTTY tools themselves, see the Feedback page.

[PuTTY release 0.60]