

Re: puTTY: Coonnection reset by peer

## Re: puTTY: Coonnection reset by peer

---

*Source:* <http://www.derkeiler.com/Newsgroups/comp.security.ssh/2006-04/msg00193.html>

---

- *From:* BearItAll <spam@xxxxxxxxxxxxxx>
  - *Date:* Tue, 25 Apr 2006 11:50:45 +0100
- 

Richard E. Silverman wrote:

Has anyone in any of these groups had this same problem with an ISP that blocks port 3306???? I'm always getting a "connection reset by peer" error when trying to connect to the offending ISP? Are there any other ssh clients that I could use where I might actually have a successful connection?

Home System: Windows XP Home  
puTTY version: 0.58

Event Log:  
Looking up website.com  
Connecting to 24.229.1.5 (not a real IP address, just part of the example)  
Failed to connect to 24.229.1.5  
Network Error: Connection reset by peer.

Your problem has nothing to do with "port 3306," or anything with your port forwarding setup at all. The initial TCP connection to the server, over which SSH would run, is either being refused or closed immediately after opening. The first suggests that the server is not running SSH (or it's firewalled off); the second, that the server is using libwrap and your source address is not allowed.

– Richard

I'd agree with that.

GGG,

You should go into your ISP control panel and look at your ssh/ssl options.

On both of my hosts ssh was disabled by default. So I just had to go in and

Re: puTTY: Coonnection reset by peer

## Re: puTTY: Coonection reset by peer

enable it. I use putty too by the way, it is what I use to access my servers from users computers. But for day to day you would find that an 'ssh -l username yourhost' from a Linux command line is easier, plus they are no term problems because you generally have the same term settings on the host as you have local, if not though the host will certainly emulate at least one of the main terms in use today.

Enabling the ssh may be all you need to do for your putty access.

If you need to go further, in the ssl section yours will hopefully give you key creating options. Where you can take a copy of the public key, and create ca's for your user or one just for your application.

As for special ports for MySQL that is a tricky area. But it might also have an easy answer.

First, when you created the user/s that your MySQL application uses, which connection did you give them? You might have only given local access. Remember when you first setup MySQL you create two user access points for the root or admin user like this,

```
mysqladmin -u root password 'yourpassword'  
mysqladmin -u root hostname.com password 'yourpassword'
```

The difference is that the first is localhost access and the second is from port. In your application, since all of it's access to the data is localhost then you may not have created the second access via the url.

Generally in a database you create,

admin or root -> All access, use localhost and remote host. Though some, including myself, reduce this to localhost only unless for some reason I have to do my admin remotely, rather than ssh'ing in first.

application user -> nearly always is localhost only. If someone does manage to grab a copy of your code then they will not get access to the data because the code's user does not have a login via a remote host. There are occasions when this isn't true, and yours may be one of those.

Then none or many users -> with various access rights.

So I would look using what ever you use to administer your MySQL and check the username that you use for this can actually come in from a remote host.

I just re-read your post and think you have two different questions on this. One being user access and the other being updates between your local and remote copy of your MySQL databases.

Some ISPs lock down on these access ports because of cause everyone knows the numbers and every hacker will have a go. But with the access rights given as I said above, then the only danger is a remote user that has full

## Re: puTTY: Coonection reset by peer

rights. Unfortunately some users have tried to use their root access to MySQL for remote updates, even for day to day work, when it would be more correct to use an admin like access to the database, who only has enough rights to do the remote updates. Or better still, pull the updates local as in sftp then perform the update using a localhost root only.