Using ZeBeDee with Firebird to Encrypt and Compress Network Traffic

Table of Contents

1. Introduction

Using Firebird with a direct connection to the Internet brings always two questions: performance and security. Firebird does not have internal support to compress IP packets, nor the possibility to encrypt the data that is transmitted over the wire. Opening the port 3050 directly to the Internet is an idea that will not please anyone.

A solution to this problem will be always a third-party product – it could be a software solution, a hardware solution or both. Personally, I was looking for a solution that I could use as a standard in all my Firebird installations, so I start looking for a software solution that could encrypt and compress TCP/IP data. Open-source and support for both Linux and Windows will be the perfect solution: something with the Firebird "soul". I found ZeBeDee.

2. What is ZeBeDee?

ZeBeDee it's a software tool to establish an encrypted, compressed "tunnel" for TCP/IP or UDP data transfer between two systems. ZeBeDee works in both Linux and Windows platforms, it's completely free for commercial and non-commercial use, and it's distributed under the terms of the GNU General Public License.

ZeBeDee installation is fast and easy: in a few minutes we can connect a Firebird Server to the Internet using encryption and data compression.

Like Neil Winton, ZeBeDee developer, says in ZeBeDee's homepage, there are other software products that do the same thing, but ZeBeDee pleases me by it's easy of configuration, simplicity, small footprint, and power (features that we, Firebird developers, are very used to).

ZeBeDee is simple to understand to a Firebird user: a ZeBeDee server will listen to connections from ZeBeDee clients. ZeBeDee Server will encrypt/compress any data that is send over a port and send it to the client. The ZeBeDee client will pick up that data, decompress/un-encrypt it and send it forward to its ultimate destination.

3. ZeBeDee Configuration to work with Firebird

ZeBeDee is a generic tool: simply runs on the traffic between two ports. It can be used in a mix of situations, but this document will focus just the use with Firebird.

A ZeBeDee server does not need to be on the same equipment of the Firebird Server. A ZeBeDee client does not need to be on the same equipment of the Firebird Client Application. There are many possible combinations, so I pick two of them: using the ZeBeDee Server on the same machine that runs the Firebird server (Option A), and using ZeBeDee on another machine (Option B).

I assume also that there will be lots of remote clients, and the ZeBeDee client will be in all of them. That's the most common environment for a Firebird Server connected to the Internet.

There are other situations that ZeBeDee could be extremely useful: connecting remote offices to a main office using "low-bandwidth" links. Such cases could benefit if they use one machine running as a ZeBeDee client for all remote workstations.

Option A)

```
/-----------------\   secure tunnel  /-----------------\
| Firebird Server |-------- // ------| Firebird Client |
| ZeBeDee Server  |                  | ZeBeDee Client  |
\-----------------/                  \-----------------/
```

Option B)

```
/-----------------\
| Firebird Server |
\-----------------/
        |
        / not-secure connection
        |
/----------------\    secure tunnel /----------------\
| ZeBeDee Server |-------- // ------| Firebird Client|
\----------------/                  | ZeBeDee Client |
                                    \----------------/
```

Personally, I never use option A, mainly by two reasons: to avoid a direct connection between my Firebird server and the Internet, and keep my Firebird Server on a dedicated machine. If someone wants to use this option should take care of blocking port 3050 away for the Internet. If not, that port will be still available for Firebird direct-connections. The only port that needs to be available to the Internet is the port used by ZeBeDee (11965 by default).

I always use option B: ZeBeDee is working on a machine that is direct connected to the internet, and independent of the Firebird Server. On the ZeBeDee Server the only port that needs to be available for the outside world is the ZeBeDee port (11965 by default).

Remember that there are more options; These are just the most common situations.

Start by installing ZeBeDee, doing the download of the installation files specific for your environment. You can have a mix of these, such as using ZeBeDee in a Linux server and Windows ZeBeDee for clients. Just run the installation for your operating system by copying the necessary files to the hard disk and do some minor configurations (such as registering the 'zbd' files as a ZeBeDee application in Windows environments). ZeBeDee is never configured 'out-of-the-box' – you must configure it to your personal needs.

ZeBeDee could be loaded using command line parameters or a configuration file. This last one is recommended, but I will start to show these two examples using a simple command line.

I have tested all examples with a Linux Mandrake Box and a W2K as ZeBeDee servers. Client applications were always Windows machines (Win98/W2K). I think that it will run smoothly in any other Linux or Windows.

I assume in all of the examples that the ZeBeDee server has a public internet address. If you want to test this on a local network it's pretty simple: just use the IP address of the ZeBeDee server.

Option A)

In ZeBeDee Server, run:

```
Zebedee –s  localhost:3050
```

On the client run:

```
Zebedee 3051:fbzebedee.com:3050
```

(change fbzebedee.com to your server's name)

ZeBeDee is now configured and ready to go!

To connect to your Firebird Server use the connection string:

```
localhost/3051:PathToYourDataBase
```

Option B)

In this option, the only thing that will change is the location of the Firebird Server. On the ZeBeDee server run the command:

```
Zebedee –s 3050: IP_Address_Of_Your_Firebird_Server:3050
```

The command for the client is the same, as well the connection string.

This is the simplest way to get ZeBeDee working with Firebird. But I recommend that you use a configuration file, because you will have more control in other ZeBeDee parameters.

ZeBeDee configuration files are text files, by default with 'zbd' extension. ZeBeDee can use a configuration file in client or server mode. Just use:

```
zebedee –f NameOfConfigFile
```

The above examples are full working ones. You just need to cut & paste this into a text file and change the address of the servers to meet yours.

Configuration File - ZeBeDee Server and Firebird Server on the same machine

```
#-------------------------------------------------------------------------------------------------
# Config file to use on a Server running Firebird & ZeBeDee
#

verbosity 2              # Message type
server true              # Server Mode
detached true            # Free from console
udpmode false            # we don't need UDP

logfile './zebedee.log'    # I love log files
keygenlevel 2            # Key level


# To validate private keys, as mention on the next chapter
# comment out the next line and use the file name you want
```

```
#checkidfile './clients.ids'

redirect none   # Close all redirection ports

# Allows Firebird connections
redirect 3050            # Firebird default port

targethost localhost     # target is on the same machine


# Some other parameters

compression zlib:9           # maximum zlib compression
keylength 256                # 256 bits keys
keylifetime 36000            # share keys last for 10 hours
maxbufsize 16383             # Maximum buffer length
#---------------------------------------------------------------------------------------------------
```

Configuration Files - ZeBeDee Server in a different machine that Firebird Server

You just need to do a change in the config file above. The line:

```
redirect 3050            # Firebird default port
```

should be:

```
redirect 3050:IP_Address_Of_Firebird_Server:3050      #Firebird port
```

Also, you need another line to allow ZeBeDee to talk with the Firebird Server:

```
targethost IP_Address_Of_Firebird_Server       # Firebird Server will be the target
```

Client Config File

```
#---------------------------------------------------------------------------------------------------
# Config File - Client
#

verbosity 1      # Basic messages only
server false     # Client
detached true    # Closes Console


# You should change the next line to your environment
# with this example, the connection string should be localhost/3051
# You can change port 3051 to another one
tunnel 3051: fbzebedee.com:3050

# If you will use a private key, comment out
# the next line, and personalize the file name, if need it
# include './myclient.key'
#---------------------------------------------------------------------------------------------------
```

4. Using Private Keys

On the examples, we don't use a private key. ZeBeDee still encrypts the data, using a key created in the moment of the connection, but there is no way of authentication into the ZeBeDee server.

If you do not specify the use of private keys, ZeBeDee just establishes an encrypted tunnel between the two points. The ZeBeDee server will accept connections for any ZeBeDee client. Even without private keys ZeBeDee does eliminate the problem of using a 'sniffer' on your packets – the username/password and your database path are encrypted, making the job of a possible attacker more difficult.

But if you want you can use fixed private keys. This will create an authentication process in the ZeBeDee server, allowing just the connections of clients that have a special key.

To create a key use the command:

```
zebedee –p
```

You will get something like:

```
privatekey "410dea0cbd9c10da057848c43a610f6bb859b769"
```

This result changes each time you run the command, so you must redirect the results to a file:

```
zebedee -p > myclient.key
```

This key should be kept secret: keep it safe. With this key you can create the associated 'fingerprint':

```
zebedee -P -f myclient.key > myclient.id
```

The myclient.id file will be something like:

```
135f04050961d37553731250d5c6f7495f088b32 myclient
```

The 'myclient' text is a remark, it just helps keeping a reference to the key. You can change it to something more usefull, such as:

```
135f04050961d37553731250d5c6f7495f088b32 Lydia Computer
```

Now, you should change your config file on the client and add the line:

```
include "path-to-dir/myclient.key"
```

where "path-to-dir" it's the directory where the key file is (myclient.key).

On other way, myclien.id data could be used in any ZeBeDee server that you want the client to attach. This is not a secret file, and should be copied to the ZeBeDee server. A typical file that contains all identities is something like:

```
 ba077f6a42bea502f517cab5685e476a713d9621  Lydia's Computer
 3ad38cb1f16957d5c535272ce27557bdaa4389c6 John's Computer
 135f04050961d37553731250d5c6f7495f088b32  Rachel's Computer
```

On ZeBeDee server you should change the configuration file adding the line:

```
checkidfile "path-to-dir/clients.ids"
```

("clients.ids" is the file that contains all id's)

At this moment, the ZeBeDee server will accept only a connection from a client that uses a corresponding private key. If you remove one line, that client will not be able to connect.

5. Transparent use with Delphi

It's simple to call the ZeBeDee client from inside your Delphi application. You just need to install ZeBeDee on the machine, and inside your application use something like:

```
   WinExec( "ZeBedee 3051: ZeBeDeeServerAddress:3050', 0 );
```

And change your connection component to use localhost/3051 as the Firebird server address. (It works very well with IBObjects.)

6. Final Notes

Security in communications over the net is a subject that makes books, and books, and books… I will not write a line in those books – I'm a developer using Firebird, not a security specialist.

A powerful use of ZeBeDee in a Firebird internet-solution is, without a doubt, the ability to compress packets. In client/server applications data transferred over the wire should be small, but in most applications we have the need to transfer large amounts of data (reports are good examples). You must try ZeBeDee with your Firebird application and measure the results. If you work with data already compressed you will not notice any difference. But if you are working with large transfers of data, using VARCHAR fields, I'm sure ZeBeDee will give a new life to your application over the internet.

Using ZeBeDee in a local network is useful, even in 10Mbits networks. You will not get any performance improvements for 99% of the cases. The time spend compressing/decompressing data will kill any benefit of the compression. You can use it for encryption, but there will be just a few installations that need to be protected for communications 'in house'.

I write this document just to show how ZeBeDee could be useful with Firebird. I strongly recommend that you read ZeBeDee manual, and learn about other configuration parameters.

7. Additional Information

Firebird it's a great Open Source RDBMS. You can get more info in:

http://www.firebirdsql.org
http://www.ibphoenix.com

ZeBeDee lives in:

http://www.winton.org.uk/zebedee

Artur Trindade Anjos
artur@arsoft.pt
Julho de 2002