Why is security so hard to understand? (http://www.sandon.it/node/76)

Posted on: Sat, 11/12/2011 - 16:54 **By:** ldsandon

It looks my efforts to convince BorCodeDero to design a sound security model into Datasnap are doomed to fail. The main reason is it looks only me is worried about the actual fake secutity model. Known speakers and writers about Delphi look totally unaware about sound security. Look at this post by Bob Swart (http://www.bobswart.nl/Weblog/Blog.aspx?RootId=5%3A5039).
Because he can't find any 64 bit OpenSSL library on his system (of course Windows doesn't install it, and if you never installed any 64 bit applications using it you'll have none) he recommends to download and install a 0.9.8g build. There's a problem here. When he wrote, and I'm writing this, the latest 0.9.8 branch release is "r". The "g" release was released October 19th 2007, more than four years ago. If there is an area where you always want the latest and the ***safest***, that's security. You won't install consciounsly any old release with known bugs and security issues (http://www.openssl.org/news/vulnerabilities.html). Yes, new releases may have unknown issues, but they are anyway better than known - and exploitable - ones. That's a list of security issues fixed between 'g' and 'r', bugs fix only not included (see the full change log (http://www.openssl.org/news/changelog.html)):

- CVE-2011-0014
- CVE-2010-4180
- CVE-2010-4252
- CVE-2010-3864
- CVE-2010-2939
- CVE-2010-0742
- CVE-2010-0740
- CVE-2010-0433
- CVE-2008-1678
- CVE-2009-4355
- CVE-2009-1378
- CVE-2009-1377
- CVE-2009-1379
- CVE-2009-3555
- CVE-2009-0789
- CVE-2009-0591
- CVE-2009-0590
- CVE-2008-5077
- CVE-2009-1386
- CVE-2008-1672
- CVE-2008-0891

They may affect you or not, but are you sure you want to deploy a release with *known* issues?
Also, if you're going to deploy such unsafe releases, please ensure they're used only by your application, don't put them in directories which may come ealier in the user computer path,

and have better applications forced to use your unsafe libraries.
If Swart had searched a little more he would have find someone who builds the latest release of OpenSSL, both 32 and 64 bit, for Windows: http://www.slproweb.com/products/Win32OpenSSL.html (http://www.slproweb.com/products/Win32OpenSSL.html) (this build requires VC++ runtime deployment), and http://indy.fulgan.com/SSL/ (http://indy.fulgan.com/SSL/).
The other Datasnap issue is stil the use of the unknown PC1 algorithm. I don't have XE2 yet, but I wonder what it means it uses RSA to encrypt the keys. If filters still use the same key stored locally instead of exchanging a session key each time (and maybe changing it after a given time during the session), well, good luck Datasnap users. You're still using fake security. But almost no one will tell you, even if you buy their books. The problem IMHO is they don't sell books about Delphi only. They sell Delphi too. And consciounsly or unconsciously, they are hiding big flaws.

Update: this post (https://forums.embarcadero.com/thread.jspa?messageID=415672#415672)on Embarcadero forums really shows how hard understanding security is by Delphi developers. Read where the OP writes "*The two .DLLs I'm using are libeay32.dll and ssleay32.dll. We got them as downloads from dll-files.com*". From dll.files.com. Not from OpenSSL site, or the Indy one. Not from any reliable source, from dll-files.com!. OpenSSL is open source. Anybody can build and upload modified OpenSSL dlls to dll-files.com (did he ever read the disclaimer of that site?), who will use it blindly in any application needing SSL to protect data? It looks some people are so used to get software from unreliable sources (guess why...) they don't mind to "add security" using the same sources. "Hey, my application uses SSL thereby is safe!". Are you sure?