

Re: Java Security

Source: <http://coding.derkeiler.com/Archive/Java/comp.lang.java.help/2005-03/0935.html>

From: KiLVaiDeN (KiLVaiDeN_at_CaRaMaiL.CoM)

Date: 03/29/05

Date: Tue, 29 Mar 2005 08:27:21 +0200

"James" <james@enliva.com> a écrit dans le message de
news:d03b3fa0.0503280501.506da632@posting.google.com...

> *Hi all,*

>

> *My company is trying to decide make a platform desicion between
> C++/Java. I am in favor of Java however I am compelled to answer a
> question yet I am unable to find a solution.*

>

> *The problem is as follows: The application will have a two secret keys
> (A 128 bit constants) and a public encryption algorihtm (AES). It will
> encrypt some data offline and send via public methods to some other
> place at a later time(not our server). Obviously, the security of this
> data is extremely important. (A financial application). Application
> will only be provided to trusted entities therefore I don't have to go
> thru authentication. (ie. verify the sender)*

>

> *Our concern is one could decompile the Java class files and see what
> these constants are and hence break the whole system. I have checked
> out various solutions to see how can we avoid this issue and not yet
> come up with a 100% secure solution.*

>

> *Obsfucation doesn't work as it doesn't really hide the constants.*

>

> *Encryption of the constants: If we did this, someone can enrypt these
> constantants. This solution is nothing more than adding another layer
> to the difficulty. (We can pick a private algorithm but decompiling
> would expose algorithm)*

>

> *I also can not change the JVM to add extensions as I would have to
> deploy multiple extensions for various platforms.*

>

> *I appreciate any pointers.*

>

> *Thanks,*

> *James*

Never give encrypt keys on an application.

You can make your application gather encrypted data from network, but find another solution to provide the keys to decrypt it. I'd suggest emails, but they are not really secure (if you are paranoid about security), so better give them by phone or letter, or use a SSL http website with the user login, that'll show him "his" decrypt key, and not everybody's key. I am not sure, but I think you already planned that, just pointing it, because in my opinion, it's the biggest flaw to give keys in your application (even if they are encrypted), as then potential "hackers" would have all data they need single packaged.

Now that said, concerning the algorithm for decryption, I'd say that no matter what you do, there is always a way to reverse engineer the class files (or the exe files). However, it has been clear that class files can be much easily interpreted than native source code, as there is tools that allow you to check the java code, tools which are very much harder to find concerning native code (you'll be able to disassemble code, but assembly is completely another thing than Java). This said, I'd suggest that once again, you make your data go through a SSL connection, and gather all the data you need with xml or something like that.. Got no real idea how to implement that the best way, but I think SSL is the real thing you need for your application, even though i'm unsure whether it's possible to send a file with SSL encryption.. But the idea is to have algorithms at home, and only allow trusted users to use the online objects.

K