

RE: Copy protection for a .NET application

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.csharp/2004-11/6191.html>

From: Salih Goncu (*Goncu_at_discussions.microsoft.com*)

Date: 11/29/04

Date: Sun, 28 Nov 2004 20:19:02 -0800

Hi, all.

Well, reading all the thread, there's a really good way to make a copy protection for a IL (or bytecode for Java) available. Besides obfuscation, which is good for many of the cases, the approach I'll tell you is for **real** paranoids.

In protecting some piece of code from analysis requires that it is impossible to reach that piece of code from within the context of application, that is, run that code somewhere else.

The best candidate, as far as I know, is smartcards with crypto co-processors.

Have the smartcard generate a key pair for you and store the private key inside itself, never telling it even you. Then, put a piece of code in the smartcard and give it the access credentials with only the public key that the card delivered you. Use a secure communication protocol between the smartcard and the computer, so that nobody will be able to intercept the data transmitted.

When your program is running as shareware, it won't require the vital piece of code that needs to execute in the smartcard. But when it is licensed, you give your customer the card and the corresponding public key. So, when in licensed mode, the card executes the vital code, and returns only the required results meaningful only for that license or context. (The code inside the smartcard may be dependent on some data or other code in your application, but your application **must** be dependent on this code piece.)

I used this approach on some projects, and proved well. You can check out possible cards supported by Windows CSP from Gemplus, Schlumberger and Orga. The capacity of cards range from 8 to 128 Kbytes, which is sufficient for many types of applications.

As this is the hardest nut to crack, this is also very hard to code, so you decide if you want it or not...

Salih

"Massimo" wrote:

> *I'm planning to develop a .NET application using C#, in order to sell it as*
> *a shareware and/or as a full package, so I'll need a good way to protect it*
> *against piracy. I know some ways to protect it (activation, serial keys,*
> *etc.), but my concern is: how can any copy protection mechanism work when*
> *you can always disassemble it and read the source code? Even if I use a*
> *native C++ DLL for my copy protection, the point where it's called from the*
> *C# code can always be found.*
> *Any good tips about this?*
>
> *Thanks*
>
>
> *Massimo*
>
>