

## Re: Copy protection for a .NET application

**Source:**

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.csharp/2004-12/0533.html>

---

**From:** C-Services Holland b.v. (*cs\_h\_at\_REMOVEcsh4u.nl*)

**Date:** 12/02/04

Date: Thu, 02 Dec 2004 10:25:06 +0100

William Stacey [MVP] wrote:

> *I gotta now. I see what they are doing. I would never spend time on it,  
> but from a discussion standpoint, I still wonder how much better that is  
> then encrypting the dll yourself and encrypting the key in the loader or  
> something. I think under both (hardlock and self encrypted dll) you have  
> some key used for decryption into memory and/or disk. Naturally, one could  
> use "corddb myapp.exe" to find the plain key passed to Rijndael (et al) for  
> decryption and just use that to manually decrypt the dlls. That is an  
> additional step that takes most folks out-of-the-game. But your not really  
> protecting from them. Your really protecting from the crackers that live  
> inside debuggers and read asm like the morning paper just to publish a crack  
> or keygen to their buddies on the INET.  
> Still sounds interesting, but don't think majority would ever buy sw that  
> requires dongles (as history shows us.) Also not sure how much better this  
> is (in reality) then decrypting and loading your own dll in a loader?  
>  
> Just curious, now that we spent time on this, what kind of application are  
> your protecting? Thanks for your patience and helping me understand.  
> Cheers.  
>*

The problem with decrypting the code yourself is that you have a readily accessible decryption module in your software to decrypt the DLL. That would make it a breeze for any decent cracker to break your protection.

We have engineering software for factories and such where very simply put they store every connection (power/signal lines etc) and instrument and what not in a database. It's a very expensive piece of software.

One of our other leading products is software for calculating powercables to conform to the Dutch NEN norm. That ranges from wiring in your home to streetlighting. Our main concern in the early days was decompilation protection (VB4, easily decompiled to vb source). So we used the hardlock encryption to stop people getting at our math modules. These days it's a bit harder, but we still use it for licencing purposes so we can control how many copies can be started at one time, serverlocks are available so they can just plug it onto a Novell or

microsoft.public.dotnet.languages.csharp: Re: Copy protection for a .NET application

Windows server for instance.

Now I agree with you that for most software packages this solution is way to expensive. The hardlock itself costs about as much as the average game. But the same company also provides a software solution (<http://www.ealaddin.com/HASPSL/default.asp>) which may be much more interesting to more developers since it doesn't require a piece of hardware and this distribution via the web is so much easier. How secure that is, I can't say. But I do know the hardlock encryption is very tough to crack.

In the end, any protection can be broken. I think our software is too specific to be interesting enough for any hacking group to spend it's efforts on, so we (our company that is) are reasonably safe.

--

Rinze van Huizen  
C-Services Holland b.v.