Should I be worried about obfuscating my .NET code? [closed]

Asked 16 years, 4 months ago Modified 6 years, 4 months ago

Viewed 7k times

26

Closed. This question is opinion-based. It is not currently accepting answers.

Want to improve this question? Update the question so it can be answered with facts and citations by editing this post.

Closed 9 years ago.

Improve this question

I'm sure many readers on SO have used Lutz Roeder's .NET reflector to decompile their .NET code. I was amazed just how accurately our source code could be recontructed from our compiled assemblies.

I'd be interested in hearing how many of you use obfuscation, and for what sort of products?

I'm sure that this is a much more important issue for, say, a .NET application that you offer for download over the internet as opposed to something that is built bespoke for a particular client.

.net obfuscation

Share

Improve this question

Follow

edited Aug 8, 2018 at 17:48

bruno

2,253 ● 1 ● 19 ● 32

asked Aug 15, 2008 at 8:36

John Sibly

23k ● 7 ● 64 ● 82

10 Answers

Sorted by: Highest score (default)

21

I wouldn't worry about it too much. I'd rather focus on putting out an awesome product, getting a good user base, and treating your customers right than worry about the minimal percentage of users concerned with stealing your code or looking at the source.

Share Improve this answer

Follow

answered Aug 15, 2008 at 8:39

Chris Bunch

89.6k ● 37 ● 129 ● 127

Except if the software contains sensitive data that **should** be protected (such as PRIVATE KEYS and PASSWORDS) – marcolopes Mar 9, 2014 at 17:08

2 @marcolopes: private keys should never be delivered with the application... you probably meant public keys... and the

passwords should be hashed, not clear text anyway...
– Igor Popov Dec 30, 2014 at 10:22

1 What other solution do you have to store PRIVATE KEYS? Server access? And if there is no Internet connection?
– marcolopes Jan 3, 2015 at 3:17

10

Remember, obfuscation is not encryption. IMHO, if somebody perceives value in reverse-engineering your code, they will do it. That's true for managed code or native code, obfuscated or not. Sure, obfuscation deters the casual observer, but is your business actually threatened by such people? Every .NET obfuscation method I've seen makes your life as a developer harder.

There are services that offer true encryption, such as SLPS from Microsoft. See http://www.microsoft.com/slps/default.aspx

Share Improve this answer

Follow

answered Aug 15, 2008 at 16:46

Martin
5,452 ● 31 ● 40

1 good point about making development harder. – Lucas B Jan 22, 2010 at 21:44

1 Fantastic comment: 'obfuscation deters the casual observer, but is your business actually threatened by such people?'
– Lawrence Wagerfield Nov 7, 2012 at 10:24

7

We currently obfuscate all our output, even though we are a small outfit who sells specialist software to a small number of clients.

We made this decision for one simple reason - we discovered a disgruntled ex-employee was actively approaching our clients requesting binaries - there was some some concern he was intending to reverse engineer newer features in order to offer competing functionality.

Of course he is still able to do this if he uses the software, but there is no reason to make it easy for him.

Share Improve this answer

Follow

answered Aug 15, 2008 at 9:00

Martin

40.3k ● 20 ● 100 ● 131

5

No new obfuscation, but lots of compiler tricks since 1.1

For instance every time you use an anonymous type you get IL that compiles back with a pretty obscure name. Every time you use yield you get a whole new class that implements both IEnumerable and IEnumerator (clever optimisation, unreadable code). Every time you use an anonymous delegate you get a new method with a name that's invalid in every .Net language that I know of, but that's fine in the IL.

Share Improve this answer

answered Aug 15, 2008 at 9:14

Follow

Keith

155k ● 82 ● 306 ● 446

4

@Rob Cooper

> Having had some discussions with my manager at work, he said he doesn't obfuscate, but does NGEN on install, apparantly that should be enough to stop Reflector working on your assemblies, but I have no idea if this is true and to what extent, so please don't take it as gospel :)

This doesn't offer any kind of protection against disassembly. First I imagine its quite possible to extract raw files from any installation package like an MSI or a CAB file.

But more importantly, Ngen runs on the client machine after the assembly has been installed. Ngen just forces the assembly to compile now instead of later using the JIT. The original assembly remains and is unmodified and it must remain because Ngen might not be able to compile the entire assembly.

Ngen is for performance, not security, and does nothing to prevent disassembly or make it even slightly more difficult.

Share Improve this answer

answered Aug 15, 2008 at 12:54

Follow

Brian Ensink
11.2k • 3 • 51 • 63

> "I imagine its quite possible to extract raw files from any installation package like an MSI or a CAB file." - Yes: superuser.com/questions/307678/… – CAD bloke May 30, 2013 at 3:28

3

easy for me - if you need to protect intellectual property - obfuicate - if not dont.

Easy to do with the right tools.

Share Improve this answer

Follow

answered Aug 15, 2008 at 8:38

littlegeek

2

I think to some extent we should ALL be worrying about our IP :)

Good question though as its something I am keen to know more about (I currently do **not** obfuscate).

Having had some discussions with my manager at work, he said he doesn't obfuscate, but does NGEN on install, apparantly that should be enough to stop Reflector working on your assemblies, but I have no idea if this is true and to what extent, so please don't take it as gospel :)

Good question :) +1

Share Improve this answer

Follow

answered Aug 15, 2008 at 8:39

Rob Cooper

28.9k ● 26 ● 105 ● 142

4 NGen will not affect Reflector in any way. Using NGen tool does not remove original assemblies from the system.
– lubos hasko Sep 17, 2008 at 13:51

2

We don't use obfuscation for "non public" applications but we use it for public available applications. The obfuscated app contains plenty of highly sophisticated code which took us an exorbitant amount of time to write and that's the reason that let me think that obfuscation is a must - at least in that case.

Share Improve this answer

Follow

answered Sep 15, 2008 at 16:59

JRoppert

5,942 ● 5 ● 34 ● 38

0

Obsfucation is limited in it's effectiveness, it might keep the casual guy away. The most effective obsfucation is making only the smallest amount of code available to the user. If you can, make your app run depend heavily on a fat server.

Share Improve this answer

Follow

answered Aug 15, 2008 at 17:07

Jim

3,170 ● 4 ● 32 ● 38

...which can be a little bit of a problem if we talk about WinForms Apps. – JRoppert Sep 15, 2008 at 16:52

0

Agree, most people who know how to code even a little bit do not need to steal your code!

Share Improve this answer

Follow

answered Jan 22, 2010 at 19:41

Drupad Panchal

393 ● 3 ● 10