(Why) should I use obfuscation? [closed]

Asked 16 years, 3 months ago Modified 1 year, 7 months ago

Viewed 3k times

7

votes

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

Closed last year.

Locked. This question and its answers are locked because the question is off-topic but has historical significance. It is not currently accepting new answers or interactions.

It seems to me obfuscation is an idea that falls somewhere in the "security by obscurity" or "false sense of protection" camp. To protect intellectual property, there's copyright; to prevent security issues from being found, there's *fixing those issues*. In short, I regard it as a technical solution to a social problem. Those almost never work.

However, I seem to be the only one in our dev team to feel that way, so I'm either wrong, or just need convincing arguments. Our product uses .NET, and one dev suggested .NET Reactor (which, incidentally, was suggested in this SO thread as well).

> .NET Reactor completely stops any decompiling by mixing any pure .NET assembly (written in C#, VB.NET, Delphi.NET, J#, MSIL...) with native machine code.

So, basically, you throw all advantages of bytecode away in one go?

Are there good *engineering* benefits to obfuscation?

.net **security** **obfuscation**

Share

edited Apr 27, 2023 at 12:54

Zoe - Save the data dump ♦

28.1k ● 22 ● 127 ● 158

asked Aug 28, 2008 at 9:08

Sören Kuklau

19.9k ● 8 ● 55 ● 90

Comments disabled on deleted / locked posts / reviews

8 Answers

Sorted by: Highest score (default)

14 votes

You asked for engineering reasons, so this is not strictly speaking an answer to the question. But I think it's a valid clarification.

As you say, obfuscation is intended to address a social problem. And social (or business) problems, unlike technical ones, rarely have a complete solution. There are only degrees of success in addressing or minimising the problem.

In this case, obfuscation will raise the barriers to someone decompiling and stealing your code. It will discourage casual attacks and, through inertia, may make your intellectual property less likely to be stolen. To make a tiresome analogy, an immobiliser doesn't prevent your car being stolen, but it will make it less likely.

Of course there is a cost, in maintainability, (possibly) in performance and most importantly in making it harder for users to accurately submit bug reports.

As GateKiller said, obfuscation won't prevent a determined team from decompiling, but (and it depends what your product is) how determined a team is likely to be attacking you?

So, this is not a technical solution to a social problem, it's a technical decision which adds one influence to a complex social structure.

Share

answered Aug 28, 2008 at 9:28

Leigh Caldwell

11k ● 4 ● 26 ● 31

8
votes

If a big team of programmers really want to get at your source code and that had the time, money and effort, then they would be successful.

Obfuscation, therefore, should stop people who don't have the time, money or effort to get your source, passers by you might call them.

Share

answered Aug 28, 2008 at 9:14

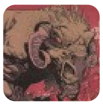
GateKiller
75.8k ● 75 ● 175 ● 204

4
votes

If you stick to pure managed code obfuscation, you can shave off quite a bit of an assembly size, and obfuscated classes/function names (collapsed to single letters) mean smaller memory footprint. This is almost always negligible, but does have an impact (and is used) on some mobile/embedded devices (though mostly in java).

Share

answered Aug 28, 2008 at 9:13



skolima
32.7k ● 27 ● 118 ● 152

3
votes

One potential engineering benefit is that in some cases obfuscation can create smaller executables or other artifacts -- e.g. obfuscating javascript results in smaller files (because all of the variables are named "a" and "b" instead of "descriptiveNameOne" and all the whitespace is stripped, etc). This results in faster load times for the web pages that use obfuscated javascript. Obviously this doesn't apply (as much) in the .NET world, but it's an example of a situation in which there is an direct engineering benefit.

Share

answered Aug 28, 2008 at 9:13

John

15.3k ● 12 ● 59 ● 57

3

votes

While not related to .net, I would consider obfuscation in Javascript, and possibly other interpeted languages. Javascript benefits well from obfuscation because it reduces the bandwith needed, and the tokens the parser has to read.

But obfuscating compiled bytecode doesn't really seem that usefull to me. I mean what would you try and achieve? I can only see obfuscation beeing slightly usefull in license checking code to avoid it beeing circumvented too easily.

Share

answered Aug 28, 2008 at 9:16

Staale

28k ● 23 ● 68 ● 85

2

votes

✓

I posted a question which might help you as it discusses some of the issues: should-i-be-worried-about-obfuscating-my-net-code

Share

edited May 23, 2017 at 12:17

Community Bot
1 ● 1

answered Aug 28, 2008 at 10:26
John Sibly
23k ● 7 ● 64 ● 82

2
votes

The main reason to use obfuscation is to protect intellectual property as you have indicated. It is generally much more cost effective to a business to purchase an obfuscation product like .NET Reactor than it is to try and legally enforce your copyrights.

Obfuscation can also provide other more incidental benefits such as performance improvements and assembly size reduction. These would the *engineering* benefits you are looking for.

Share

answered Aug 28, 2008 at 9:17
John Hunter
4,132 ● 4 ● 27 ● 35

1
vote

Use encryption to protect information on the way.

Use obfuscation to protect information while your program still has it.

Share

answered Aug 28, 2008 at 9:41
Jorge Córdoba
52.1k ● 11 ● 82 ● 130