

## Re: Obfuscation and Salamander

**Source:** <http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.faqs/2004-05/0008.html>

---

**From:** Richard Grimes [MVP] (*read*)

**Date:** 04/30/04

Date: Fri, 30 Apr 2004 14:18:01 +0100

Matt wrote:

> *Is my only choice to use managed C++? If so what level of source is  
> exposed in its assemblies/builds?*

Managed C++ generates IL, just as do all the other .NET compilers. You could compile your secret algorithms as native code (so that the module contains x86 for the algorithm) and managed C++ does help here because you can tell the compiler that a function should be compiled as x86 rather than IL, this x86 code is stored in the assembly.

Of course, any cracker worth his salt will be able to read x86 and will be able to work out your algorithm. The problem with .NET is that (as you imply) the metadata in the assembly contains readable class and member names, this level of information is not normally available with x86 code unless a symbol file (.pdb or .dbg) is supplied. However, ...

> *what's more, for those that don't get Salamander, what is the point  
> of using obfuscation (the main use being its obscure class  
> overloading and variable renaming) when your own application  
> 'ildasm.exe' will give anyone all the original friking names, types  
> and variables??*

Not true, unless the obfuscator is really lame (in which case don't use it). In your assembly classes and class members are referred to through tokens, the compiler will add metadata to the file that gives the full (string) name of the item that the token refers to. However, for private classes or private members these string names are not required because the tokens are sufficient. An obfuscator will replace the string names of such private members with some unrelated string, so you don't get to see the original name (friking or not).

This does not work with public classes because the string names are required by compilers when they compile a class that use them.

Some obfuscators do additional work: they change the code flow to make the IL less readable.

microsoft.public.dotnet.faqs: Re: Obfuscation and Salamander

- > *My question is simple, why should I, as a freelance developer, spend*
- > *good time producing client/server software and libraries in .net when*
- > *any stupid monkey can extract all of my source using Salamander? And*

You say you are a client-server developer. Well, if your server code is installed on a secure server and is accessed through remoting or a web service then the person running the client won't have access to the IL/metadata. (They only need access to the metadata