



## Re: Protecting .NET assemblies (runtime)

I have a collection of various .NET assemblies I authored used in various applications within our corporate Intranet. The assemblies are used in fat-client apps, asp.net apps, etc, therefore many of the assemblies are distributed to end user systems (they are not installed in the GAC).

It is my understanding that anyone can basically copy a .NET assembly, create a reference to it and consume it's public methods if CAS is not implemented in some fashion? I understand .NET assemblies are just MSIL code and meta data and can be reverse engineered quite easily (based on what I've read) using tools like the .NET reflector, etc. if they are not obfuscated. I'm not so concerned with this security aspect as developers/end users reusing my .NET assemblies in their own applications.

In short, I've read where with Framework 1.1 one could use the `[StrongNameIdentityPermission ( SecurityAction.Demand , PublicKey="public key" ...)` declaration at a class/function level which would throw a runtime error if any consumer tried to use a strong-named assembly where the caller was not also signed with the same strong-name key. I then found this is no longer the case in .NET 2.0 where if the caller is fully trusted the `StrongNameIdentifyPermission` check is completey disregarded as discussed here?  
[http://msdn2.microsoft.com/en-us/library/aa480477.aspx#pagguidelines0003\\_class3](http://msdn2.microsoft.com/en-us/library/aa480477.aspx#pagguidelines0003_class3)

How can I protect my .NET 2.0 assemblies from being consumed by other applications?

Re: Protecting .NET assemblies (runtime)