

Forum:[Hack Forums](#)

Thread: [Spaghetti - Arithmetic Obfuscation](#)

Barret - October 22, 2017, 11:34 pm

A small release of my latest work that has only one purpose, masking constant numbers into fully customizable equations inside .net assemblies.

This might be not something for beginners as i deem this fairly advanced and might 'confuse' the hell out of ya ?

So basically this is just one small feature that can be implemented into a larger obfuscation or confuser projects, that was the whole idea of it.

But i simply suck at programming obfuscators but i do know how to make these little things from time to time, just for the heck of it.

Approach and execution

The equation solver requires 3 types of input, pool of random numbers, functions and an input number, from there on it will build the expression from the solver and generate true ILcode from this. Once that is done it will replace all numbers within your assembly with new method calls inside a new type.

Before:

After:

Variations

The pool is used within the equations and the size of this pool impacts the size of the equation also the functions are dynamically implemented that are semi hard coded.

You have to encounter float and integers so we add a random serie from 1 to 10, 100, and perhaps a 1000 depending what numbers we are aiming for, if we deal with floats we also

add a few decimal numbers but they are not required if the functions pool contain certain mathematical formula's that will return decimals as well (cos, sin, tan, etc)

Available Functions

Basic operators

Tan

Sin

Cos

square root

Squared

Log

and Round

Remarks

Sometimes certain parameters take longer, there is a setting that determines the tolerance for decimal neglecting, meaning that a tolerance of 0.0004 will accept 10.00009 as 10.
You have to play with the settings to find a sweet spot.

Example of solved constants within an assembly:

I was planning on not releasing this but i changed my mind ? let me know what you think about it and i would appreciate if you write a comment when your using it.

Github: <https://github.com/TheBarret/Spaghetti>

gigajew - October 23, 2017, 7:38 am

Check out Noisette-Obfuscator that sam made, it has some basic examples of obfuscation (it's on GitHub)

Also ConfuserEx has a ton of examples too

Very nice job on this, looks nice

Barret - October 23, 2017, 8:17 am

(Oct 23, 2017 07:38 AM)gigajew Wrote: [\[CITE\]](#) Check out Noisette-Obfuscator that sam made, it has some basic examples of obfuscation (it's on GitHub)

Also ConfuserEx has a ton of examples too

Very nice job on this, looks nice

Yeah seen those, but they don't incorporate dynamic math functions that are compiled. So thats why its not exactly same in my view i assume you have not seen the source code.

gigajew - October 23, 2017, 8:23 am

(Oct 23, 2017 08:17 AM)Barret Wrote: [\[CITE\]](#) Yeah seen those, but they don't incorporate dynamic math functions that are compiled. So thats why its not exactly same in my view i assume you have not seen the source code.

No, but they do show examples of different types of obfuscation, I thought that's what you wanted. There's also one method that I've only come across one time reverse engineering, and that's a method that keeps increasing an integer throughout the flow, and if you remove or change a method that increases or decreases that integer, the program will no longer function, because it uses that integer for some methods to function