

Forum: [Hack Forums](#)

Thread: [FacelessDebugger || MSIL Emulator / Sandbox](#)

BahNahNah - February 13, 2018, 5:27 pm

I havnt released anything in a while and ive had this lying around for a bit so here ya go.

Download: [Github](#)

Currently only works with console applications. Will try to get forms eventual when i feel motivated enough.

Here is an example gif of the dummy app running ([source on the github](#)):

Spoiler ([Click to View](#))

Note: The application is not being "executed". Its bytecode is being emulated by the debugger. Here is an example of editing calls on the fly:

Spoiler ([Click to View](#))

Yeah I know the GUI is hideous.

Any contribution to the project through the github is appreciated.

If you do choose to contribute, please keep all the "core" emulation/debugging things in the "Faceless.Core" project.

gigajew - February 13, 2018, 5:29 pm

This is awesome. Good job bananaboi.

HFDone - February 14, 2018, 3:22 am

good release. first i thought its done with pure c# but when i take alook i saw you using dnlib. i guess its much easy then using MethodBuilder.CreateMethodBody.

Zephyros - February 14, 2018, 12:02 pm

If you add some randomization to the VM, this could make a powerful anti-virus and protection toolkit.

I found a similar code on github a few month back, but yours is much nicer to read.

<https://github.com/neurosystemeu/Virtual-Machine/>

BahNahNah - February 14, 2018, 1:21 pm

(Feb 14, 2018 04:22 AM)HFDone Wrote: [\[CITE\]](#) good release. first i thought its done with pure c# but when i take alook i saw you using dnlib. i guess its much easy then using MethodBuilder.CreateMethodBody.

MethodBuilder.CreateMethodBody Still requires that you have the bytes for each method.
dnlib is being used to parse the executable. I dont think CreateMethodBody is a substitute.

gigajew - February 14, 2018, 3:10 pm

(Feb 14, 2018 02:21 PM)BahNahNah Wrote: [\[CITE\]](#) MethodBuilder.CreateMethodBody Still requires that you have the bytes for each method.
dnlib is being used to parse the executable. I dont think CreateMethodBody is a substitute.

[https://msdn.microsoft.com/en-us/library/system.reflection.methodbody.getilasbytearray\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.reflection.methodbody.getilasbytearray(v=vs.110).aspx)

BahNahNah - February 14, 2018, 3:43 pm

(Feb 14, 2018 04:10 PM)gigajew Wrote: [\[CITE\]](#)
[https://msdn.microsoft.com/en-us/library/system.reflection.methodbody.getilasbytearray\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.reflection.methodbody.getilasbytearray(v=vs.110).aspx)

That gave me an idea.

I might have a go at migrating over to using Assembly.ReflectionOnlyLoad

edit:

yeah, i dont think reflection is gonna cut it.

I need to load an assemblys bytes so that i can read the IL instructions of each method.

Loading it with reflection, then using MethodBuilder to load each methods bytes is gonna fuck up the reference to methods within the assembly.

So using the dummy app as an example, the entrypoint isnt gonna be able to find the "fib" and "reverse" methods.

Fixing the calls to point to the correct method is in option but you still gotta read the IL to ever get to that stage.

I think dnlib is the best approach at the moment.

HFDone - February 15, 2018, 2:48 am

(Feb 14, 2018 04:43 PM)BahNahNah Wrote: [\[CITE\]](#) That gave me an idea.

I might have a go at migrating over to using Assembly.ReflectionOnlyLoad

edit:

yeah, i dont think reflection is gonna cut it.

I need to load an assemblys bytes so that i can read the IL instructions of each method.

Loading it with reflection, then using MethodBuilder to load each methods bytes is gonna fuck up the reference to methods within the assembly.

So using the dummy app as an example, the entrypoint isnt gonna be able to find the "fib" and "reverse" methods.

Fixing the calls to point to the correct method is in option but you still gotta read the IL to ever get to that stage.

I think dnlib is the best approach at the moment.

? "i guess its much easy then using MethodBuilder.CreateMethodBody."

with MethodBody.GetILAsByteArray you get the bytes of the method but you need to fix the tokens and other things to get it work.

Edit - but it will save you using dnlib

Ninja - February 16, 2018, 12:42 pm

(Feb 15, 2018 03:48 AM)HFDone Wrote: [\[CITE\]](#) Edit - but it will save you using dnlib
Whats wrong with dnlib?

Serious question, I never messed with it or used it.

debug - February 16, 2018, 6:34 pm

(Feb 16, 2018 01:42 PM)Ninja Wrote: [\[CITE\]](#) Whats wrong with dnlib?

Serious question, I never messed with it or used it.

Nothing particularly, it's just very dense with a lack of any real documentation. So you basically have to teach yourself as there aren't many public examples.

EZLib - February 16, 2018, 6:37 pm

(Feb 16, 2018 07:34 PM)debug Wrote: [\[CITE\]](#) ...

This is a very true statement.

BahNahNah - February 19, 2018, 4:50 pm

Current task is trying to pry away from dnlib.

Currently porting the core things im unsung from dnlib (mainly [this stuff](#)) onto the [faceless.pure branch](#).

Feel free to contribute wherever you can.

BahNahNah - September 12, 2018, 2:23 am

Did a little bit of work on this project. Now can handle basic object creation/usage

The current mini-goal is to support emulation of winforms applications.

int - September 12, 2018, 6:05 pm

(Sep 12, 2018 02:23 AM)BahNahNah Wrote: [\[CITE\]](#) Did a little bit of work on this project. Now can handle basic object creation/usage

The current mini-goal is to support emulation of winforms applications.

Very nice work, I saw this earlier (a few months ago) but forgot about it until now

Zephyros - October 3, 2018, 1:52 am

Keep it up, really like that project. Also probably look at the project I posted on the first page if you need some inspiration

pinterkoding - May 19, 2021, 9:48 am

This is really cool stuff to release. This is my first post.