

Forum:[Hack Forums](#)

Thread: [Hooking .net function calls](#)

BahNahNah - October 17, 2017, 7:07 pm

The goal here is to be able to hook/redirect calls from .net code on runtime.

An example scenario is injecting a managed dll into a unity game and hooking the unity OnGui function to your newly injected library.

Hooking the jit method table ([e.g.](#)) is a possibility but i have found this kinda unreliable and would rather do it in a lower level style by hooking the .net framework/jit more directly.

The two options i have thought of is

1) Hooking the jit compiler. (like [ubbe's project](#)) and redirecting while it is being compiled
However, this would require reversing the jit compile on any method that i wish to hook that is already been jit-compiled. Wich is a likely senario.

I couldn't find any info on un-jit-compiling methods.

2) Hooking the low-level .net calling procedures.

I really dont know anything about this. I have looked around for any starting place on this but came out empty handed.

Any help on either of these solutions, or any new solutions would be greatly appreciated.

HFDone - October 18, 2017, 2:00 am

https://en.wikipedia.org/wiki/List_of_CIL_instructions

like native api hook you can hook net method by using the bytescode to do it
and its much simple then ubbe way

Washi - October 18, 2017, 2:16 am

(Oct 18, 2017 02:00 AM)HFDone Wrote: [\[CITE\]](#) https://en.wikipedia.org/wiki/List_of_CIL_instructions

like native api hook you can hook net method by using the bytescode to do it
and its much simple then ubbe way

That would require you to either patch the program on the disk, or use a jit hook anyway and somehow make that the method you're hooking is not jitted yet. since the IL code is only interpreted once by the runtime and then translated into native code.

@op: I was thinking (and not sure if it would actually work), you could maybe use the [Marshal.GetFunctionPointerForDelegate](#) to get the native function pointer to a method of your choice. You could then follow that pointer and edit the first few assembly instructions that will call or jump to your callback function, thus installing a hook. You could also have a look at hooking libraries (maybe easyhook), feed in this function pointer and let them do the work for you.

HFDone - October 18, 2017, 4:54 am

(Oct 18, 2017 02:16 AM)Washi Wrote: [\[CITE\]](#) That would require you to either patch the program on the disk, or use a jit hook anyway and somehow make that the method you're hooking is not jitted yet. since the IL code is only interpreted once by the runtime and then translated into native code.

@op: I was thinking (and not sure if it would actually work), you could maybe use the [Marshal.GetFunctionPointerForDelegate](#) to get the native function pointer to a method of your choice. You could then follow that pointer and edit the first few assembly instructions that will call or jump to your callback function, thus installing a hook. You could also have a look at hooking libraries (maybe easyhook), feed in this function pointer and let them do the work for you.
what you offer to op thats what i mean and already done ?

ForceJew - October 18, 2017, 11:05 am

You can just change the OnGui in dnspy

BahNahNah - October 18, 2017, 12:56 pm

(Oct 18, 2017 02:16 AM)Washi Wrote: [\[CITE\]](#) @op: I was thinking (and not sure if it would actually work), you could maybe use the [Marshal.GetFunctionPointerForDelegate](#) to get the native function pointer to a method of your choice. You could then follow that pointer and edit the first few assembly instructions that will call or jump to your callback function, thus installing a hook. You could also have a look at hooking libraries (maybe easyhook), feed in this function pointer and let them do the work for you.

This might work. I'll do so.e research on jitted methods and seekf I can dig anything up.

(Oct 18, 2017 11:05 AM)ForceJew Wrote: [\[CITE\]](#) You can just change the OnGui in dnspy
That is not the goal.
OnGui hook was just an example.

3dsboy08 - October 18, 2017, 2:24 pm

<https://github.com/pardeike/Harmony>

Thank me later. Works quite well, the Transcompiler is really useful.

stefsot - October 18, 2017, 3:15 pm

option one is not possible as Unity games are using mono
second option is the best imo

I would share how to do it but I'm using it on my own projects.

melk - December 18, 2019, 5:43 pm

can you pm me please need help with something

(Oct 18, 2017 02:24 PM)3dsboy08 Wrote: [\[CITE\]](#) <https://github.com/pardeike/Harmony>

Thank me later. Works quite well, the Transcompiler is really useful.
can you pm me please need help with something thanks