

Forum:[Hack Forums](#)

Thread: [Finding source of function call](#)

Psyberion - July 31, 2013, 8:39 am

Hello everyone!

I'm not entirely sure where this is supposed to go, but since it's highly related to decompiling, I suppose this is the most suitable place.

Anyway, I want to decompile a program, to be able to modify it slightly. The program consist of quite a few modules and .dll files, but the UI consists solely of a .vbs file, which is great. I skimmed through the code and found the call to the function I want to manipulate. The problem is that I have no clue in what other file that function is.

The software I am tinkering with is (obviously) closed source, so just doing something like simply greping the source is impossible; I only have binaries to work with. I might be worth noting that I have checked for any inclusion of other files, but I haven't found any.

So, my question is if there is a simple, preferably reliable and fast, way of finding out what file contains the function being called in the .vbs?

Help would be much appreciated!

BigFagget - July 31, 2013, 10:21 am

Upload the files and I am crack it for you and make a tut.

Psyberion - July 31, 2013, 10:29 am

Too big to upload, I'm afraid. Also, I'd rather crack it myself, I just want to know if there is an easy way of finding the source of a function call, or if I have to do it the hard way. But thanks anyway!

BigFagget - July 31, 2013, 11:36 am

Logically you will have to pause the program and look at the stack before you are progressing. Most of the time you will notice SendMessage.

Get on top of that method and bp it and keep going backwards.

Psyberion - July 31, 2013, 1:59 pm

Yeah, that makes sense. Thanks a lot, I'll take a look at this!

Doctor Shadow - July 31, 2013, 4:43 pm

I got an idea, when a function call is made the address of the next instruction that's supposed to be executed is pushed onto the stack. While you're in the function call, observe the stack and find the return address? The function call would have been made on the previous address.

Psyberion - August 1, 2013, 1:59 am

(Jul 31, 2013 04:43 PM)Doctor Shadow Wrote: [\[CITE\]](#) I got an idea, when a function call is made the address of the next instruction that's supposed to be executed is pushed onto the stack. While you're in the function call, observe the stack and find the return address? The function call would have been made on the previous address.

This seems like a simpler idea than pausing the program, in this case at least. Thank you!

