

Forum:[Hack Forums](#)

Thread: [larger exe sizes than expected](#)

Stumpy - March 6, 2016, 10:31 pm

I'm trying to compile a crypter I'm writing to be as small as possible.. Using visual studio with dynamic linking I'm able to get it down to 10kb, but unfortunately due to the required redistributables I need to compile it with static linking, bringing the file up to 140KB.

With the orwellDevC++ I'm able to get it to 441KB, which is very large, while using the -s flag, and dynamic linking.

What can I do to make the executable size smaller? I'm looking for something less than 50kb ideally.

Hoodie+ - March 7, 2016, 7:08 am

You could try using an alternate c run-time and statically linking it, it would reduce the size but also limit the functionality, alternatively you could remove all the dependencies and only use windows.h

I'm writing on my phone... I can explain more once I get home...

notsteve - March 7, 2016, 11:25 am

Are you turning on optimization for size and not releasing a debug build? Alternatively, you can force visual studio to use a previous library, using the lowest common denominator. [Here's someone holding your hand through it.](#)

Hoodie+ - March 7, 2016, 11:42 am

Make sure to use
Code -Werror -Wall -o2 -s

NTTITON378 - March 7, 2016, 12:51 pm

If you're that worried about size, use a previous toolset as others have suggested as well as incorporating your own c runtime and disabling default libraries. You'd dynamically load them using ntdll, or kernel32. Don't compile with debugging information, compile on release, and exclude the manifest.

You could even build a native application, but then that would have to be loaded using ntdll, or manually mapped.

And of course, not including CRT you must define your entry point.. as well as disabling security checks

0xDEAD10CC - March 7, 2016, 5:27 pm

(Mar 7, 2016 12:42 PM)Hoodie+ Wrote: [\[CITE\]](#) Make sure to use
Code -Werror -Wall -o2 -s
That isn't even for Microsoft's compiler, that's GCC.

The first thing to do would be to make sure that you switch over to Release configuration and you aren't looking at the Debug build. Compile with /MT to make the binary load the C++ runtime files when the executable is loaded (so you can remove it later). Remove all default libs

(/NODEFAULTLIB) and change the entry point in the linker settings to your own. You can also remove the manifest, and also the relocations section if you never relocate code -- turn off random relocations in linker settings, and change the randomized base address setting to disabled. Next you can save up on some unused bytes by joining sections like .rdata and .text -- Merge sections option ".rdata=.text". For optimization you can set that to /O1 to get minimum size or favor small code /Os. Unless you're not using intrinsics you can disable those too, in addition to disabling whole program optimization. Additionally, you might try to enable string pooling. There's more you can do still but that involves knowledge on how the PE format works. At that point packers like UPX will typically only increase the size of your executable too.

Stumpy - *March 7, 2016, 7:50 pm*

Awesome guys, thanks a lot. Got it down quite a bit already.