

Forum:[Hack Forums](#)

Thread: [How to load add a binary file and load it to an entry point or runpe? \[C++\]](#)

FileBoss - December 28, 2015, 7:13 pm

Hi, I'm using visual studio community 2015 (If there are any better c++ compilers that allow resource adding in a better interface please tell me). So say I managed to add a a binary (.exe) to my project in c++ as a resource (in the C++ solution), are there any working entry point solutions or run PE, as this is the most common way (not wanting to do anything more advanced than this). I would really appreciate some help in this.

excr4sh - December 28, 2015, 7:40 pm

Get the exe from resouces then use one of these methods:

- 1) Write the exe to disk and use ShellExecute function
- 2) Write the exe to disk and use CreateProcess function
- 3) Spawn a new process and inject the exe into it
- 4) Load manually the exe in memory (resolving imports, relocations, ...) and calling the entry point

FileBoss - December 28, 2015, 8:56 pm

I don't have a lot of c++ knowledge, do I add the resource in C++ Visual community like, properties, linker, input, Embed Managed Resource File? The key is that it can't drop the file on disk at all, thats why entrypoint like in c# would be better.

Goodies - December 28, 2015, 9:55 pm

(Dec 28, 2015 08:40 PM)excr4sh Wrote: [\[CITE\]](#) 1) Write the exe to disk and use ShellExecute function

2) Write the exe to disk and use CreateProcess function These two obviously touch disk, so they're not really good for any type of malware or anything. If an AV scans it, and it likely will, it's much more likely to get detected.

Quote:3) Spawn a new process and inject the exe into it
Nebbett's shuttle hasn't worked in years, but that may not be what you're referring to.

Quote:4) Load manually the exe in memory (resolving imports, relocations, ...) and calling the entry point

This is your best bet. There's actually powershell that does this. I'm NOT a windows user, so my powershell experience is limited to C API calls to basic functions for allocating memory, moving shellcode into it, and creating the thread. Not really too much else. But here's how it works according to his Defcon slide:

- 1) Powershell allocates memory for the PE (EXE or DLL)
- 2) Copies the PE's headers to the aforementioned allocated memory
- 3) Copy all of the relevant sections to memory like .text, .data, .bss, etc...
- 4) It performs "base relocations" on the sections that were loaded (I don't quite know what this means, but I'm watching the video again).
- 5) Load the DLL's that the PE requires.
- 6) Call the entry function.

Slides:

<https://www.defcon.org/images/defcon-21/dc-21-presentations/Bialek/DEFCON-21-Bialek-PowerPw>

[ning-Post-Exploiting-by-Overpowering-Powershell.pdf](#)

Video: https://www.youtube.com/watch?v=OAd68_SYQc8

Powershell Script:

<https://github.com/clymb3r/PowerSploit/blob/master/CodeExecution/Invoke-ReflectivePEInjection.ps1>

Just reading the script, I can already get a feel for how it works. He comments the structs that he is using throughout the script and talks a lot about it in the video.

He said "cavet" like "cuh-vet". Pretty sure he meant "caveat." I don't know why, but that bugs me.

excr4sh - December 28, 2015, 10:52 pm

(Dec 28, 2015 09:56 PM)FileBoss Wrote: [\[CITE\]](#) I don't have a lot of c++ knowledge, do I add the resource in C++ Visual community like, properties, linker, input, Embed Managed Resource File? The key is that it can't drop the file on disk at all, thats why entryptoint like in c# would be better. A lot of ways of storing binary data into an executable, It depends for what purpose you need this.

If you are writing a legitimate application (nothing to do with malware), you can use the method 1 or 2. If you want to code a packer, crypter or malware you better use the 3 or 4 (I recommend this). If you don't want to drop anything to disk then you bet for loading manually the PE image in memory (method 4). This method is the most difficult to code (compared to other) because you need to know very well the PE format.

I recommend reading this:

[https://msdn.microsoft.com/en-us/library/windows/desktop/ms680547\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/ms680547(v=vs.85).aspx) (official)

<http://www.csn.ul.ie/~caolan/pub/winresdump/winresdump/doc/pefile2.html> (recommended)

If you still don't know a decent C/C++, I recommend learning it before you start with these topics.