

Forum:[Hack Forums](#)

Thread: [Inject DLL before program](#)

joebob - April 7, 2017, 9:55 pm

I'm trying to hook the Direct Input function "DirectInput8Create" but it happens too early so that starting the target program then manually injecting the DLL isn't an option as I would be too late to catch the event.

I know programs like Ollydbg and x64dbg can essentially load a program but stop it from executing on the first line of code until you tell it to start, would something like this be the best option? If so, does anyone have any tips on how to accomplish it? Or is there some better way to handle this?

Thanks.

edit: I found this blog post that helped explain most of what I needed. (can't post links yet)

However I'm a bit stuck on finding the entry point. Using x64dbg to find my program's entry point works great but I get a different value when I try reading the .exe file using the WinNT.h functions and doing:

```
entryPoint = ntHeader->OptionalHeader.ImageBase +  
ntHeader->OptionalHeader.AddressOfEntryPoint;
```

Anyone have any idea why that wouldn't be giving the correct address?

GlycoL - April 8, 2017, 3:06 am

Try the developers section, I don't think you'll have much luck here.

joebob - April 8, 2017, 4:35 am

Thanks I'll give that a try.

edit: or not... don't have enough prestige to post there.

Mr. Kim - April 8, 2017, 6:33 am

(Apr 8, 2017 04:35 AM)joebob Wrote: [\[CITE\]](#) Thanks I'll give that a try.

edit: or not... don't have enough prestige to post there.

If you want I can post it on your behalf in their

joebob - April 8, 2017, 6:43 am

(Apr 8, 2017 06:33 AM)M3MEMORY Wrote: [\[CITE\]](#) If you want I can post it on your behalf in their
That would be awesome, thanks.

Dragon King - April 8, 2017, 9:30 am

There are many ways you can do this:

- 1) If you can code, you can create a program that uses CreateProcess() with the flag CREATE_SUSPENDED, map your dll, then resume the process using ResumeThread on lpProcessInformation->hThread (which is the last argument passed into CreateProcess).
- 2) You can hijack a dependency (for example, if it loads asd.dll, you can replace it with your own, but if that dll exports functions the application uses / is necessary for the application to work then this is not a feasible option).

3) You can use ollydbg/x64dbg to load the application, and use it to pause the application at the entry point, and then inject your dll.

As for why the EP is different, it's probably due to ASLR mapping the application to a different base address.

joebob - April 8, 2017, 10:12 am

(Apr 8, 2017 09:30 AM)Dragon King Wrote: [\[CITE\]](#) 1) If you can code, you can create a program that uses `CreateProcess()` with the flag `CREATE_SUSPENDED`, map your dll, then resume the process using `ResumeThread` on `lpProcessInformation->hThread` (which is the last argument passed into `CreateProcess`).

That's basically what I was trying but creating a process with `CREATE_SUSPENDED` doesn't let the program load anything before being suspended so how would I inject a dll in the process if I can't use something like `LoadLibrary`?

What I'm doing now is

- 1) create a suspended process
- 2) overwrite the entry point (that I get manually from a debugger) with a loop
- 3) resume the thread and wait for it to reach the loop
- 4) suspend the thread and inject my dll
- 5) restore the old entry point code and resume

Which of course works but requires manually getting the new entry point address every time I recompile the target program.

(Apr 8, 2017 09:30 AM)Dragon King Wrote: [\[CITE\]](#) As for why the EP is different, it's probably due to ASLR mapping the application to a different base address. Wouldn't ASLR remap to a different address each launch? Which would make hard coding an address not work.

edit: ..Ignore all that. I read some bad info online and assumed they knew what they were talking about without trying it myself. Loading a dll into a process created with `CREATE_SUSPENDED` isn't an issue. All is well, no reason to bother with any of the entry point stuff. Thanks for the ideas.