

Forum:[Hack Forums](#)

Thread: [Wow64GetThreadContext fails with error 87 using Aeonhacks RunPE](#)

eoaikyn - August 12, 2017, 4:17 pm

I apologize if this has been solved elsewhere but I searched and was not able to find any information on this issue I have been having. To start off with, despite my account age I've actually been writing crypters and DLL injectors for a long time, and I feel I'm quite comfortable working with PE files and modifying them in memory. I learned about the RunPE method a few years ago, and haven't really had the time to mess with it until just very recently. I was surprised how easy it was to setup 32bit->32bit injection, however I'm having trouble accomplishing the same thing for 64bit->64bit. This is on Windows 8.1.

My implementation is based on a C# ported version of Aeonhacks RunPE ([hxxps://pastebin.com/Dzhad8rB](https://pastebin.com/Dzhad8rB)) because I'm more comfortable there than in VB land. I've compared it to Aeonhacks original VB.NET implementation, and it looks good to me. However whenever I try to inject into a 64-bit process, it fails at Wow64GetThreadContext with GetLastError returning 87 - The Parameter is Incorrect

What the hell is going on here? The project is compiled as 64-bit Release, The handle being passed is indeed a 64 bit handle. And I'm passing in the same parameter as I pass into GetThreadContext. Looking at MSDN the implementations of the two, they are identical, so what gives? As a last resort, I copied the original VB.NET code into a project and it ALSO fails at the same point! So that leads me to speculate that it is either something in my project settings, or something funky with windows 8.1 that I can't seem to find information about. Any guidance would be very much appreciated

Ninja - August 12, 2017, 4:59 pm

Why port to C# when you can reference the VB class if you make it a library?

eoaikyn - August 12, 2017, 5:02 pm

Mostly because I want to understand the code, either way the VB class experiences the same error so that wouldn't solve my problem. The only thing I've been able to find is that VS2017 uses the windows 10 SDK by default. I'm going to try installing the Windows 8.1 SDK and see if that changes anything, but I'm not expecting it too (they should be backwards compatible yes?)

Well, I feel like a n00b. The initial post where I had read about Aeonhack mentioned it was able to do 64 bit injections but looking at the code it's obvious that's not actually the case. I supposed I should have been tipped off by the function, since I know the purpose of the wow64 function is NOT to access 64 bit stuff specifically. Knowing this I spent the last few hours reviewing things and was easily able to modify the code to do 64-bit to 64-bit. Sorry for the stupidity! I just needed to sleep on it and then everything was clear.