MSBuild Injection

8-9 MAR 21

Tasking

Inject arbitrary code into a running MSBuild process to be packaged with the resultant binary without negatively affecting its stability or touching disk.

Injection Potential

Last week we identified multiple vectors for injecting into MSBuild, ultimately deciding to attempt to inject as early in the stack as possible to affect the greatest number of build systems. However, the first proof of concept starts small by hooking ReadFile calls from cl.exe—the VC(++) compiler-linker stack—, searches among handles for a filepath matching the target, finds the printf() in that file, and modifies the output value. This is 100% completed, so I only need to test and clean it up.

The current obstacle is injecting the payload DLL in the first place. Compiling via cl.exe on cmd was troublesome but saves a significant amount of time. Nonetheless, the Detourer payload is not hooking calls as needed. There must be somewhere on the cl.exe stack I'm failing to deploy it, because it will capture plenty of ReadFile calls, but none to my code file. I know it's something minor because my logs show I had caught it once while shoving hooks everywhere I could.

I've tweaked everything else—the includes and definitions are all available and correct, the code is certainly compiling within this substack, I'm using the right compiler, etc.—so this has to be the only issue. I'll get it sorted out.

Future Plans

1. Validate POC.
2. Inject more code as we go, then expand to different stages of the build.
3. Finally, attempt to inline/hook in PIC during compilation.
4. Sub-project: Modify crypto calls to use a hardcoded, seemingly random key.