Why does GetFileVersionInfo map the whole image into memory instead of just parsing out the pieces it needs?

devblogs.microsoft.com/oldnewthing/20140530-00

May 30, 2014

Raymond Chen

Commenter acq responds (with expletive deleted), "the whole file is mapped into the process' memory only for version info that's certainly only a few kilobytes to be read?" Why not map only the parts that are needed? "I don't understand the necessity to map the whole file except that it was easier to write that code without thinking too much." That was exactly the reason. But not because it was to avoid thinking. It was to make things more secure. Back in the old days, the `GetFileVersionInfo` function did exactly what acq suggested: It parsed the executable file format manually looking for the file version information. (In other words, the original authors did it the hard way.) And it was the source of security vulnerabilities because malformed executables would cause the parser to "behave erratically". This is a common problem: Parsing is hard, and parsing bugs are so common that that there's an entire category of software testing focused on throwing malformed data at parsers to try to trip them up. The general solution for this sort of thing is to establish one "standard parser" and make everybody use that one rather than rolling their own. That way, the security efforts can be focused on making that one standard parser resilient to malformed data. Otherwise, you have a whole bunch of parsers all over the place, and a bad guy can just shop around looking for the buggiest one. And it so happens that there is already a standard parser for resources. It's known as the loader. The function `GetFileVersionInfo` therefore got out of the file parsing business (it wasn't profitable anyway) and subcontracted the work to the loader.

Pre-emptive xpclient rant: "Removing the icon extractor for 16-bit DLLs was a mistake of the highest order, even worse than Component Based Servicing."

Raymond Chen

Follow