AppInit_DLLs should be renamed Deadlock_Or_Crash_Randomly_DLLs

devblogs.microsoft.com/oldnewthing/20071213-00

December 13, 2007

Raymond Chen

I have no idea why the window manager team added this feature to Windows NT. It basically says, “Hi, use this key to violate all the rules known to mankind about what can legitimately be done in a `DllMain` function. Oh, and be an attractive malware attack vector, too.” I’ve debugged a few crashes that were traced back to the `AppInit_DLLs` key. What makes them particularly fun is that the offending DLL is usually not on the stack. Rather, the fact that a foreign DLL is being loaded inside `USER32`‘s initialization code means that you’re violating the rule against calling `LoadLibrary` inside a `DllMain` function. The result of this madness is that DLLs get initialized out of order, and typically manifests itself in some DLL crashing trying to use an object (often a critical section) that it is supposed to have initialized in its `DLL_PROCESS_ATTACH` handler. It crashed because the loader got tricked into initializing DLLs out of order. The dependent DLL received its `DLL_PROCESS_ATTACH` before the prerequisite DLL. I end up looking at these failures because the victim DLL is often a DLL that my group is responsible for.

The window manager folks came to the same conclusion about `AppInit_DLLs`, and it doesn’t work any more in Windows Vista by default. (Nick Kramer describes how to re-enable it.)

Raymond Chen

Follow