How my lack of understanding of how processes exit on Windows XP forced a security patch to be recalled

devblogs.microsoft.com/oldnewthing/20070504-00

May 4, 2007

Raymond Chen

Last year, a Windows security update got a lot of flack for causing some machines to hang, and it was my fault. (This makes messing up a demo at the Financial Analysts Meeting look like small potatoes.) The security fix addressed a category of attacks wherein people could construct shortcut files or other items which specified a CLSID that was never intended to be used as a shell extension. As we saw earlier, lots of people mess up `IUnknown::QueryInterface`, and if you pass the CLSID of one of these buggy implementations, Explorer would dutifully create it and try to use it, and then bad things would happen. The object might crash or hang or even corrupt memory and keep running (sort of). To protect against buggy shell extensions, Explorer was modified to use a helper program called `verclsid.exe` whose job was to be the "guinea pig" and host the shell extension and do some preliminary sniffing around to make sure the shell extension passed some basic functionality tests before letting it run loose in Explorer. That way, if the shell extension went crazy, the victim would be the `verclsid.exe` process and not the main Explorer process. The `verclsid.exe` program created a watchdog thread: If the preliminary sniffing took too long, the watchdog assumed that the shell extension was hung and the watchdog told Explorer, "Don't use this shell extension." I was one of the people brought in to study this new behavior, poke holes in its design, poke holes in its implementation, review every line of code that changed and make sure that it did exactly what it was supposed to do without introducing any new bugs along the way. We found some issues, testers found some other issues, and all the while, the clock was ticking since this was a security patch and people enjoy mocking Microsoft over how long it takes to put a security patch together. The patch went out, and reports started coming in that machines were hanging. How could that be? We created a watchdog thread specifically to catch the buggy shell extensions that hung; why isn't the watchdog thread doing its job? That was a long set-up for today's lesson. After running its sanity tests, the `verclsid.exe` program releases the shell extension, un-initializes COM, and then calls `ExitProcess` with a special exit code that means, "All tests passed." If you read yesterday's installment, you already know where I messed up. The DLL that implemented the shell extension created a worker thread, so it did an extra `LoadLibrary` on itself so that it wouldn't get unloaded when COM freed it as part of `CoUninitialize` tear-down. When the DLL got its `DLL_PROCESS_DETACH`, it shut down

its worker thread by the common technique of setting a "clean up now" event that the worker thread listened for, and then waiting for the worker thread to respond with a "Okay, I'm all done" event. But recall that the first stage in process exit is the termination of all threads other than the one that called `ExitProcess`. That means that the DLL's worker thread no longer exists. After setting the event to tell the (nonexistent) thread to clean up, it then waited for the (nonexistent) thread to say that it was done. And since there was nobody around listening for the clean-up event, the "all done" event never got set. The DLL hung in its `DLL_PROCESS_DETACH`. Why didn't our watchdog thread save us? Because **the watchdog thread got killed too**! Now, the root cause for all this was a buggy shell extension that did bad things in its `DLL_PROCESS_DETACH`, but blaming the shell extension misses the point. After all, it was the fact that there existed buggy shell extensions that created the need for the `verclsid.exe` program in the first place. **Welcome Slashdot readers**. Since you won't read the existing comments before posting your own, I'll float some of the more significant ones here. The buggy shell extension was included with a printer driver for a printer that is no longer manufactured. Good luck finding one of those in your test suite.

The security update was recalled and reissued in a single action, which most people would call an *update* or *refresh*, but the word *recall* works better in a title.

Raymond Chen

Follow