Why doesn’t the Windows blue screen of death prominently identify the company that created the driver that crashed?

devblogs.microsoft.com/oldnewthing/20250121-00

January 21, 2025

When there is a crash in the kernel, Windows displays the famous blue screen of death. Why doesn’t the blue screen message also say, “This crash brought to you by Company X, author of Driver D”? Wouldn’t that make it easier for users to understand whom to blame for the problem?

When you assign blame, you need to be sure you are assigning blame correctly. If you mess up, then you’ll be like one of those “Find my lost phone” apps that gives the wrong location for a phone, causing some poor homeowner to be harrassed repeatedly.

What the kernel knows is that the crash occurred due to the execution of a specific instruction, and it can even figure out what memory address that instruction was attempting to access. But it is not necessarily the case that the driver that is executing that instruction is the one at fault for the crash.

One large category of failure is memory corruption. These types of failures are often quite difficult to debug because memory corruption usually does not manifest itself as a crash in the code that did the corrupting, but rather as a crash in the code that tries to use the corrupted data. If you blame the driver that executed the crashing instruction, you’ll be blaming the victim, rather than the culprit.

If you assume that memory corruption is random, then each time the system crashes, it will blame a different driver, and the conclusion of the user might be, “There must be some unknown driver that is causing all these other drivers to crash.” But it might also be “Each time I try to repair a driver that Windows blamed for the crash, I just get a crash in some other driver. Windows is so horribly broken that *all* of the drivers are crashing! And sometimes, it comes back and re-blames the driver that I just repaired. Oh, and how do I repair the `ntoskrnl.exe` driver?”

Furthermore, it’s not valid to assume that memory corruption is random. We’ve seen memory corruption bugs that consistently corrupt the same innocent victim repeatedly. So you can’t even use a rule of thumb that “Well, if ten consecutive crashes are on the same instruction, then that code is definitely at fault.”

Assigning blame for an access violation in native code is difficult because the nature of memory corruption can lead to the access violation occurring in a component unrelated to the one that is the source of the problem. An incorrect assignment of blame causes users (and technology reporters) to march on the company’s headquarters with torches and pitchforks, and now you have public relations and legal problems on your hands.

Related reading: Windows 95 provided the name of the crashing driver in its blue screen messages, giving users the incorrect impression that it was a Windows-provided driver that was crashing their system.

Bonus reading: Steve Ballmer did not write the text for the blue screen of death. But he did write the text for the `Ctrl`+`Alt`+`Del` screen in Windows 3.1.