Memory corruption from outside the process looks like space aliens

devblogs.microsoft.com/oldnewthing/20250123-00

January 23, 2025

Chasing down memory corruption is one of the more frustrating parts of debugging. You can use debugger write breakpoints and tools like Address Sanitizer (ASAN), Valgrind, Application Verifier (AppVerifier), and Page heap to try to identify memory corruption bugs in real time. And you can use tools like rr, and Time Travel Debugging (TTD) to record the execution of a program and replay it. But all of these tools can only track writes that were issued by the program being debugged.

If the offending write comes from outside the process, then all your program sees is a mysterious change in the value even though the program never modified it. (As far as you can determine, it was changed by space aliens.)

You can use this knowledge to your advantage: If you see a memory change that is not detected by a write breakpoint or Time Travel Debugging, then you can add to the list of scenarios the possibility that the memory is being updated from outside the process, say by kernel mode (example 1, example 2), or more rarely, by another process doing `WriteProcessMemory` as some crude form of interprocess communication (not recommended).

Next time, I'll do a quick comparison of some of these diagnostic tools I mentioned above.