2.2250738585072012e-308

17 February 2011 · bug reports, security, testing, tools, tricks

Meet my new friend 2.2250738585072012e-308, We've been hanging out recently. If you've not heard of him, he's about ten years old but thats pretty old in [dog and in] software years. He's getting pretty famous in his old age, but he had humble beginnings as a lowly bug report on a Sun Microsystems website.

It's rumoured he was first discovered back in 2001, but his big break didn't come until recently (http://www.exploringbinary.com/java-hangs-when-converting-2-2250738585072012e-308/), when it was realised that he has the potential to be a key component of a Denial of Service attack that could bring down many java based systems [that accept floating point numbers (http://en.wikipedia.org/wiki/Floating_point) as input]. This includes commonplace application servers like Tomcat, who accept floating point numbers as part of the HTTP protocol.

2.2250738585072012e-308 has now been placed firmly in my mental bag of tricks along with divide by zero, 2^32, null, imaginary numbers, localised floats and all the others that routinely get brought out to help me test and investigate software.

But why am I doing this now? This information has been around for years. The software has been vulnerable for years. The hard work was done 10 years ago when some clever person found this bug. Their good work was ignored, by us… by me. I've added that bug to my toolkit, that should help me catch a few instances of it as time goes by. But if we think like testers - about testing - for a moment. I'm behaving as if I'm back on the happy path: "I know about all the 'typical' bugs again." But what other bugs are there?

The most effective way to learn from this is to not only take note of this instance, but to learn how to get even better from the experience. People like Sun (now Oracle), and other organisations, publish lists of bugs found by their community (http://bugs.sun.com/bugdatabase/top25_bugs.do). Even better they often publish lists of 'recently' fixed bugs! So you can find a bug and suggest a solution! So now I'm going to start browsing these sites for useful intel' on the systems I'm testing. I recommend you do to.