Google shoehorned Rust into Pixel 10 modem to make legacy code safer

Rust doesn’t have a slow garbage collector. Instead, it uses a mechanism called the borrow checker that ensures memory safety at compile time. This strict set of rules ensures that you can’t “forget” to free up memory—code simply won’t compile if the memory rules are breached. This is what makes Rust a memory-safe language.

However, not even Google can wave its proverbial hand and change how modem firmware is written—we’re talking about tens of megabytes of executable machine code, which is a lot. Not only would it be a herculean task to update decades of work, but many of the companies involved also consider the inner workings of modems to be trade secrets.

To protect the Pixel modem from zero-day attacks, Google focused on the DNS parser. As cellular features have migrated to data networks, DNS has become a core part of how phones work. Google explains that DNS requires parsing of untrusted data, and that makes it a major security concern, but it’s one you can solve with Rust.

Google chose the *hickory-proto* open source Rust DNS library, which is not particularly optimized but has broad adoption and support. The modem in Pixel phones is not a memory-constrained environment, which allowed the team to tack on a Rust component to make DNS in the existing code safer. The team stripped out the standard library dependencies, allowing it to compile to machine code for faster operation, which was then grafted onto the existing C/C++ modem code. In total, the Rust

components added 371KB, which is workable in the Pixel modem.

Under this system, any attempt to trigger a vulnerability by manipulating memory runs into the Rust wall—it can’t be affected by malicious DNS packets. The Pixel 10 phones are the first to ship with this safer modem implementation. Google hopes this work will allow other platforms to make similar improvements, but the company notes the size of its chosen Rust library could be a problem for simpler embedded systems. It may be possible to address that by making the library more modular in the future. Google also sees this work as a foundation for integrating more memory-safe components into the cellular baseband in time.

https://arstechnica.com/gadgets/2026/04/google-shoehorned-rust-into-pixel-10-modem-to-make-legacy-code-safer/