Computer scientists develop a framework to protect browsers from zero-day vulnerabilities in third-party libraries

December 7 2021

Credit: Pixabay/CC0 Public Domain

Researchers from the University of California San Diego, the University of Texas at Austin, and Mozilla have designed a new framework, called RLBox, to make the Firefox browser more secure. Mozilla has started deploying RLBox on all Firefox platforms this week.

RLBox increases browser security by separating third-party libraries that are vulnerable to attacks from the rest of the browser to contain potential damage—a practice called sandboxing.

Browsers like Firefox rely on third-party libraries to support different functionalities—from XML parsing, to spell checking and font rendering. These libraries are often written in low-level programming languages, like C, and, unfortunately, introducing vulnerabilities in C code is extremely easy. RLBox protects users from inevitable vulnerabilities in these libraries and supply-chain attacks that exploit these libraries.

"Well-funded attackers are exploiting zero-day vulnerabilities and supply chains to target real users," said Deian Stefan, an assistant professor in UC San Diego's Computer Science and Engineering department. "To deal with such sophisticated attackers we need multiple layers of defense and new techniques to minimize how much code we need to trust (to be secure). We designed RLBox exactly for this."

The team's effort to deploy RLBox on all Firefox platforms is detailed in a recent Mozilla Hacks blog post.

With RLBox, developers can retrofit systems like Firefox to put modules, like third-party libraries, in a fine-grained software sandbox. Like process-based sandboxing, which browsers use to isolate one site from another, software sandboxing ensures that bugs in the sandboxed

module will not create security vulnerabilities—bugs are contained to the sandbox. "Unlike process-based sandboxing, though, RLBox's sandboxing technique makes it possible for developers to isolate tightly coupled modules like Graphite and Expat without huge engineering or performance costs," said Shravan Narayan, the UC San Diego computer science Ph.D. student leading the project.

WebAssembly and sandboxing

At its core, the RLBox framework consists of two components. The first is the sandboxing technique itself: RLBox uses WebAssembly (Wasm). Specifically, RLBox compiles modules to WebAssembly and then compiles Wasm to native code using the fast and portable wasm2c compiler. "By compiling to Wasm before native code, we get sandboxing for free: We can ensure that all memory access and control flow will be instrumented to be confined to the module boundary," said Narayan.

Wasm also makes it possible for RLBox to optimize calls into and out of sandboxed code into simple function calls. In an upcoming study, to be published in the proceedings of the 2022 ACM SIGPLAN Principles of Programming Languages Symposium, the researchers show that this is safe because Wasm satisfies a set of theoretical conditions called "zero-cost conditions." This is unlike most other sandboxing techniques, which require glue code at the sandbox-application boundary to be secure. This glue code is error-prone and, in some cases, contributes to large performance overheads—the team's Wasm compiler elides this glue code, its complexity, and its overhead.

Tainted type system

The second key component of RLBox is its tainted type system. Sophisticated attackers can break out of the Wasm sandbox if the code

interfacing with the sandboxed code—the Firefox code—does not carefully validate all the data that comes out of the sandbox. RLBox's type system, which is implemented using C++ metaprogramming, prevents such attacks by marking all data coming out of the sandbox as "tainted" and ensuring, through compiler errors, that developers sanitize potentially unsafe data before using it. "Without such a type system, it would be extremely difficult to ensure that developers put all the right checks in all the right places," said Stefan.

"RLBox is a big win for Firefox and our users," said Bobby Holley, Distinguished Engineer at Mozilla. "It protects our users from accidental defects as well as supply-chain attacks, and it reduces the need for us to scramble when such issues are disclosed upstream."

The team's original work on RLBox was published in the proceedings of the USENIX Security Symposium last March. Since then they've been working on bringing RLBox to all Firefox users. RLBox will ship on all Firefox platforms, desktop and mobile, sandboxing five different modules: Graphite, Hunspell, Ogg, Expat and Woff2. The team is actively working on sandboxing more modules in future versions of Firefox and supporting use cases beyond Firefox.

More information: Matthew Kolosick et al, Isolation Without Taxation: Near Zero Cost Transitions for SFI. arXiv:2105.00033v3 [cs.CR], arxiv.org/abs/2105.00033

Provided by University of California - San Diego

vulnerabilities.html