

Writing Grub2 modules in Rust

Wednesday, 22 September 2021 07:10 (30 minutes)

The grub2 bootloader is a trusted component of the secure boot process, including “traditional” GPG-based secure boot, UEFI-based secure boot, and the logical partition secure boot process being developed by IBM. Grub2 is mostly written in C and has suffered from a number of memory-unsafety issues in the past.

Rust is a systems programming language suitable for low-level code. Rust can provide valuable tools for safer code: code in ‘safe’ Rust has stronger guarantees about memory safety, while ‘unsafe’ code has to be contained in specially marked sections. It is reasonably easy for Rust code to interoperate with C.

Grub2 is based on a modular design. Potentially vulnerable components such as image and file-system parsers are written as individual modules. Can we progressively rewrite these modules in a safer language?

I will discuss my progress enabling Rust to be used as a language for grub development, issues I have encountered, decisions we will have to make as the grub community, and next steps from here.

I agree to abide by the anti-harassment policy

I agree

Primary author: AXTENS, Daniel (IBM)

Presenter: AXTENS, Daniel (IBM)

Session Classification: System Boot and Security MC

Track Classification: System Boot and Security MC