Contribution ID: **116** Type: **not specified**

Security Features status update

Wednesday, 18 September 2024 12:00 (40 minutes)

Another year of work is behind us, with lots of progress across GCC, Clang, and Rust to provide the Linux kernel with a variety of security features. Let's review and discuss where we are with parity between toolchains, approaches to solving open problems, and exploring new features.

Parity reached since last year:

- counted_by attribute for bounded Flexible Array Members (GCC, Clang)
- language extension to support Flexible Array Member in Unions (GCC, Clang)

In progress:

- -fbounds-safety language extension (Clang)
- arithmetic overflow protection via -fsanitize=(un)signed-integer-overflow, -fsanitize=implicit-(un)signed-integer-truncation, and idiom exclusions (Clang)
- improving -Warray-bounds warnings (GCC)

Stalled, needs driving:

- forward edge Control Flow Integrity (GCC: KCFI)
- arbitrary stack protector guard location (Clang: RISC-V, PowerPC)
- Link Time Optimization (Kernel support for GCC)
- backward edge Control Flow Integrity (x86 CET Shadow Stack in kernel mode)

Primary author: COOK, Kees (Google)

Co-authors: ZHAO, Qing; WENDLING, Bill (Google); STITT, Justin (Google)

Presenters: COOK, Kees (Google); ZHAO, Qing; WENDLING, Bill (Google)

Session Classification: Toolchains Track

Track Classification: Toolchains Track