Contribution ID: **454** Type: **not specified**

Security Features status update

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:

- arbitrary stack protector guard location (Clang: RISC-V, PowerPC)
- counted_by attribute for Pointer Members (GCC, Clang)

Compiler-specific features landed since last year:

- improved diagnostics for -Warray-bounds and related warnings (GCC)
- kcfi hash salt attribute (Clang)

In progress:

- -fbounds-safety language extension (Clang)
- arithmetic overflow protection via Overflow Behavior Types and __strong typedef (Clang)
- forward edge Control Flow Integrity (GCC: KCFI)

Stalled, needs driving:

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

Presenters: STITT, Justin (Google); COOK, Kees (Google)

Session Classification: Toolchains MC

Track Classification: Toolchains MC