Contribution ID: **173** Type: **not specified**

The journey of BPF from restricted C language towards extended and safe C.

Monday, 12 September 2022 10:00 (30 minutes)

BPF programs can be written in C, Rust, Assembly, and even in Python. The majority of the programs are in C. The subset of C usable to write BPF programs was never strictly defined. It started very strict. Loops were not allowed, global variables were not available, etc As BPF ecosystem grew the subset of C language became bigger. But then something interesting happened. The C language itself became a limiting factor. Compile Once Run Everywhere technology required new language features and the intrinsics were added. Then the type tagging was added. More compiler and language extensions are being developed. BPF programs are now written in what can be considered a flavor of C language. The C language is becoming a safe C. Other languages rely on garbage collection (like golang) or don't prevent memory leaks (like C or C++) the extended and safe C is addressing not only this programmer's concern, but other bugs typical in C code. This talk will explore whether BPF's safe C one day will become the language of choice for the core kernel and user space programs.

I agree to abide by the anti-harassment policy

Yes

Primary author: STAROVOITOV, Alexei (Meta)

Presenter: STAROVOITOV, Alexei (Meta)

Session Classification: eBPF & Networking

Track Classification: eBPF & Networking Track