

Eliminating bugs in BPF JITs using automated formal verification

Friday, August 28, 2020 7:00 AM (45 minutes)

This talk will present our ongoing efforts of using formal verification to eliminate bugs in BPF JITs in the Linux kernel. Formal verification rules out classes of bugs by mechanically proving that an implementation adheres to an abstract specification of its desired behavior.

We have used our automated verification framework, Serval, to find 30+ new bugs in JITs for the x86-32, x86-64, arm32, arm64, and riscv64 architectures. We have also used Serval to develop a new BPF JIT for riscv32, RISC-V compressed instruction support for riscv64, and new optimizations in existing JITs.

The talk will roughly consist of the following parts:

- A report of the bugs we have found and fixed via verification, and why they escaped selftests.
- A description of how the automated formal verification works, including a specification of JIT correctness and a proof strategy for automated verification.
- A discussion of future directions to make BPF JITs more amenable to formal verification.

The following links to a list of our patches in the kernel, as well as the code for the verification tool and a guide of how to run it:

<https://github.com/uw-unsat/serval-bpf>

I agree to abide by the anti-harassment policy

I agree

Primary author: NELSON, Luke (University of Washington)

Presenter: NELSON, Luke (University of Washington)

Session Classification: Networking and BPF Summit

Track Classification: Networking & BPF Summit