Moving eBPF Verification Out of the Kernel

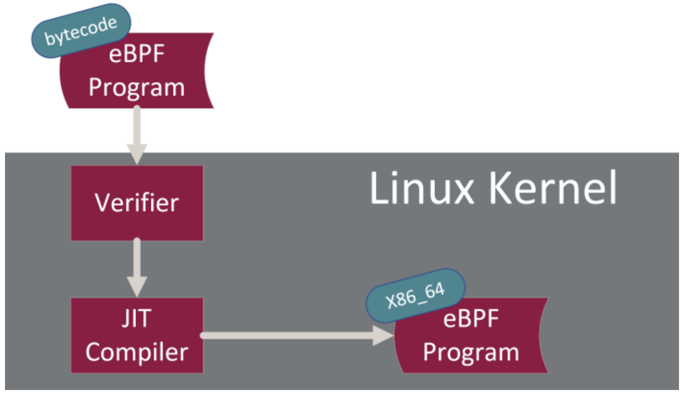
Adam Oswald, Raj Sahu, Jinghao Jia, Dan Williams, Michael V. Le, Tianyin Xu

Virginia Tech

University of Illinois Urbana-Champaign

Background

- eBPF aims to allow loading verified and safe programs into the kernel
- Verification and JIT is done in kernel space
- eBPF vulnerabilities often target the verifier and the JIT
- Lack of a rigid testing framework for the verifier

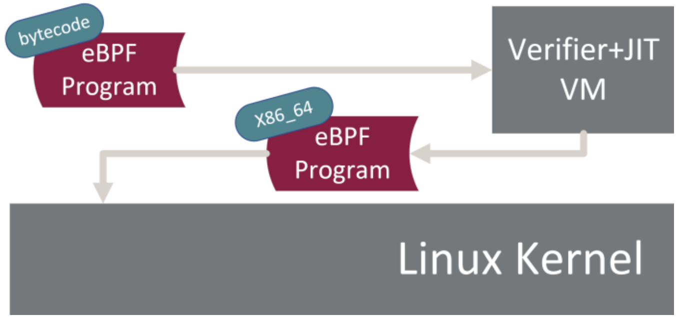


Proposal

- The verifier and JIT should be moved outside of the host kernel
- Use signing to confirm authenticity

Design

- Provide a VM with a minimal kernel that performs verification and JIT
  - Pointer leaks only affect guest kernel
  - eBPF doesn't run inside VM
  - VM restarts from fresh state for every eBPF load
  - Output: Signed native machine code (e.g., x86_64)
- Placeholders are made for relocs
- Host checks and runs output from VM in kernel
  - Safety is provided through signing
  - Only need to verify program once



The verifier doesn't only verify

- Verifier fills in address to file descriptor
- Verifier and BTF are tightly coupled
  - Type-checking
- Verifier rewrites structs members to kernel structs
  - Two versions of structs are maintained

Challenges

- Verifier does more than verification
  - Code rewrites for maps and kernel structs
- How to transition maps from guest to host?
- Supporting multiple programs?
- Host and guest Kconfig differences?
- Kernel version differences?