Contribution ID: **397** Type: **not specified**

Improving eBPF Complexity with a Hardware-backed Isolation Environment

Thursday, 19 September 2024 11:00 (30 minutes)

While eBPF has been used in various scenarios, it presents two issues in use. The first is the complexity issue, where legal programs may fail in the verification due to the verifier's limited capabilities. Researchers have resorted to ''verifier-oriented programming"to circumvent this issue, such as masking memory accesses to reduce the verification complexity. Even so, it remains a persistent issue highlighted by many literature; The second is the security issue, where malicious programs may pass the verification due to vulnerabilities. Over half (36/60) of eBPF's CVEs come from the verifier since 2014.

Through systematic analysis, we found that the above issues come from the full-path analysis stage of the verification. It executes symbolically the program at the entry and explores all possible execution paths to check whether the state is illegal or not. However, it encounters the well-known state explosion problem, which has become the bottleneck of eBPF.

This proposal aims to address the above challenges to expand the practical applications of eBPF. Specifically, current BPF programs are viewed as part of the kernel code, so eBPF uses the verification-based method to "review"the code to identify all abnormal behaviors. But we choose another perspective —BPF programs are a new type of kernel-mode application that interacts with the kernel through helper function calls rather than system calls, so kernel security should be achieved by isolating BPF programs, not by verification. As such, we aim to build an isolated execution environment for eBPF and enforce runtime isolation for BPF programs, thus eliminating the need for full-path analysis in the verification.

Primary author: WANG, Zhe (Institute of Computing Technology, Chinese Academy of Sciences)

Presenter: WANG, Zhe (Institute of Computing Technology, Chinese Academy of Sciences)

Session Classification: eBPF Track

Track Classification: eBPF Track