Contribution ID: **200**

Type: **not specified**

SYZOS: Practical KVM fuzzing

Fuzzing the Linux kernel with coverage-guided tools like syzkaller has proven to be an extremely effective method for finding kernel bugs. However, complex subsystems like KVM present unique and significant challenges that standard syscall fuzzing cannot easily address. Fuzzing KVM effectively requires managing complex state across both the host and the guest, and necessitates the coordinated execution of code in both environments.

The biggest hurdle has been the generation of meaningful guest-side code. Randomly generated instruction blobs are difficult to create, trivial to break through mutation, and nearly impossible to reliably test. This fragility has limited the fuzzer's ability to explore deep, guest-driven functionality and device interactions.

This talk introduces SYZOS, a novel framework designed to overcome these challenges. Initially prototyped on ARM64 and now being extended to x86, SYZOS reframes the problem: instead of fuzzing raw instructions, we fuzz higher-level operations within the guest. It consists of a small, immutable C library loaded into the guest that exposes a simple, fuzzer-friendly API. The fuzzer generates a sequence of calls to this API, providing stable, high-level building blocks for complex guest-side actions like interrupt controller setup, privileged register manipulation, and triggering controlled VM exits.

This 15-minute presentation will detail the SYZOS architecture, demonstrate how it enables deeper and more meaningful KVM fuzzing, and share key findings from our work on both ARM64 and x86.

Primary author: POTAPENKO, Alexander (Google)

Presenter: POTAPENKO, Alexander (Google)

Session Classification: Kernel Testing & Dependability MC

Track Classification: Kernel Testing & Dependability MC