Toward Efficient Fuzzing of Nested Virtualization

Reima Ishii
The University of Tokyo
ishiir@g.ecc.u-tokyo.ac.jp

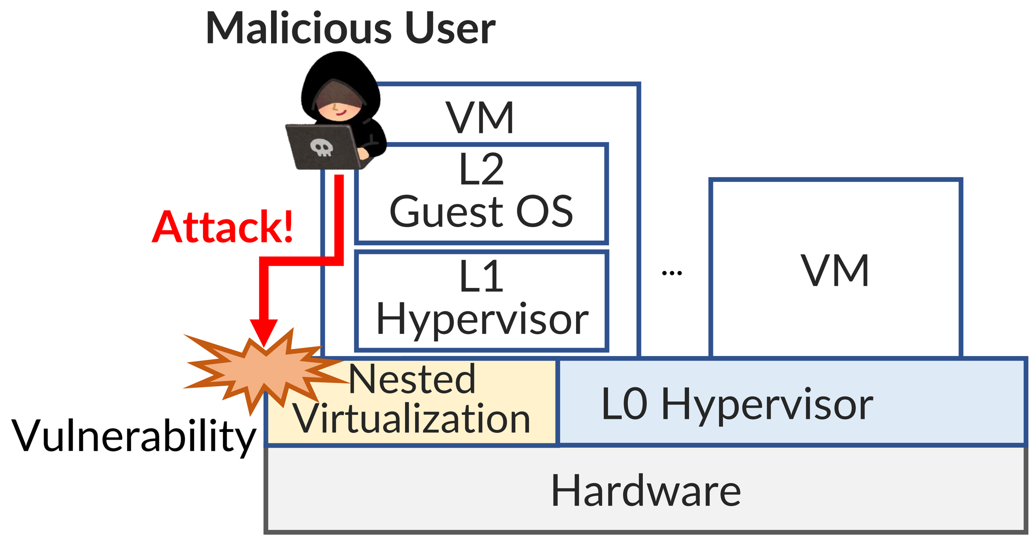
Takaaki Fukai
National Institute of Advanced Industrial Science and Technology
takaaki.fukai@aist.go.jp

Takahiro Shinagawa
The University of Tokyo
shina@ecc.u-tokyo.ac.jp

Abstract:

1 Background

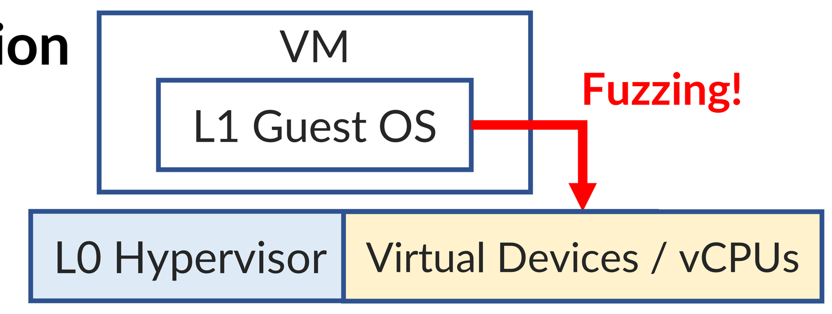
- Recent IaaS providers support nested virtualization
  - Users run their own L1 hypervisors and L2 guest OSs
- Ensuring security in nested virtualization is crucial



2 Previous Hypervisor Fuzzing

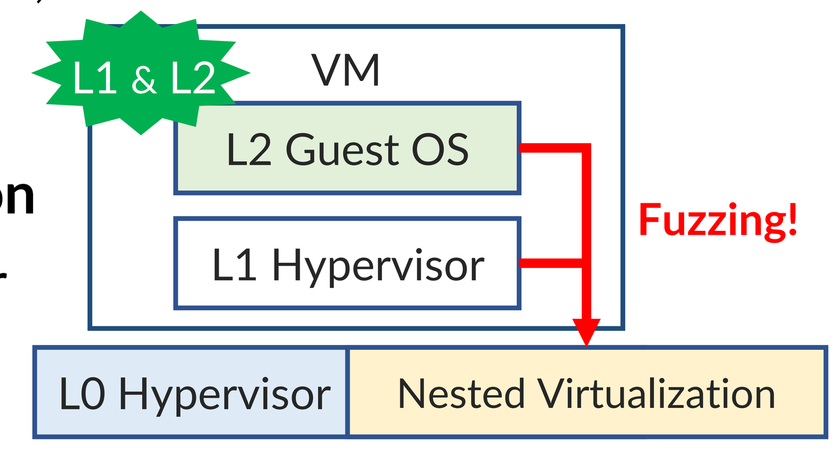
Focus Was Not on Nested Virtualization

- Virtual Devices
  - PIO, MMIO, DMA, etc.
- Virtual CPU
  - Task Switch, APIC Emu, MSR Emu, etc.

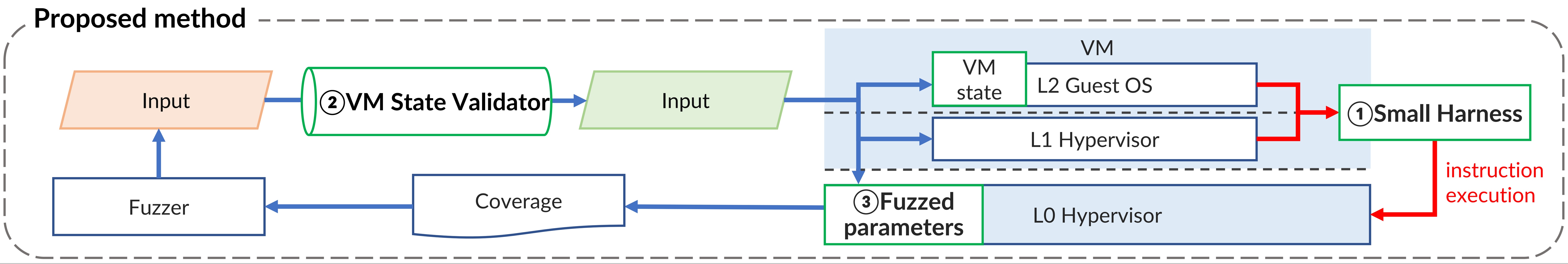


Insufficient Coverage of Nested Virtualization

- Even advanced fuzzer like Syzkaller

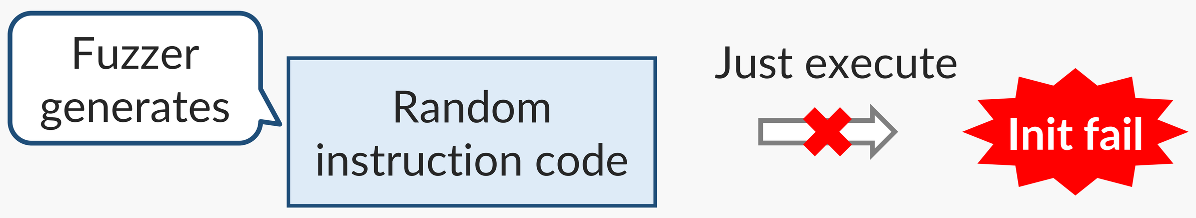


3 Proposal: Specialized Fuzzing for Nested Virtualization



Challenge 1: Proper Initialization of L2

- Randomly generated instruction sequences struggle to complete initialization

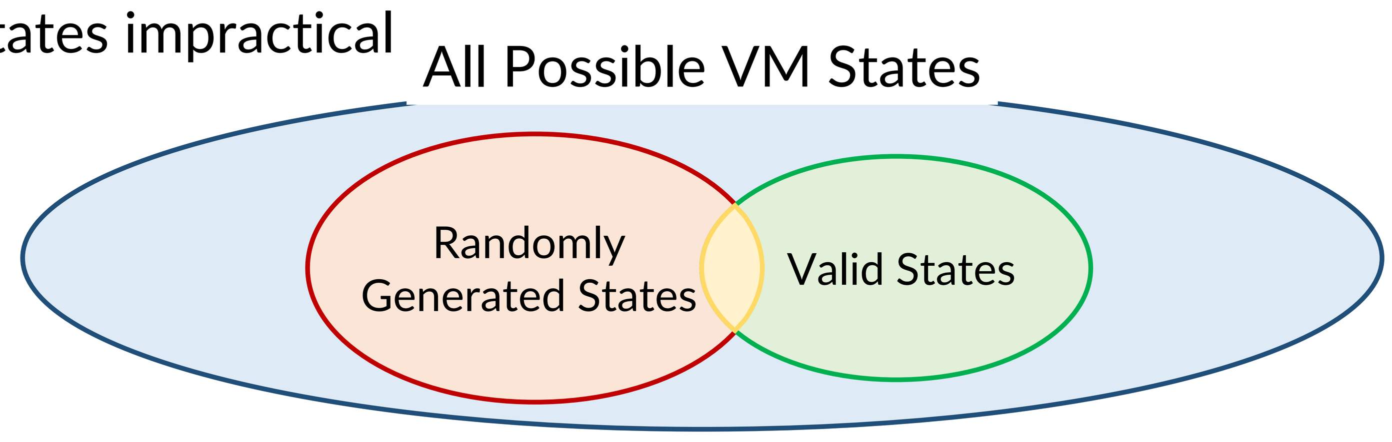


Proposal 1: Small Harness

- Executes an instruction sequence mutated from a template of correct initialization code

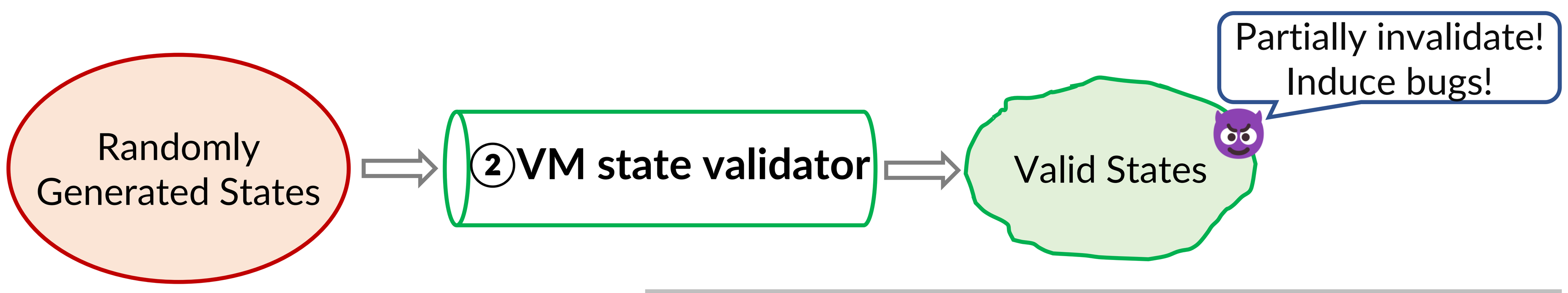
Challenge 2: Huge VM State Space

- Enormous state space of VMs makes testing all possible states impractical



Proposal 2: VM State Validator

- Randomly generated VM states are rounded to valid states

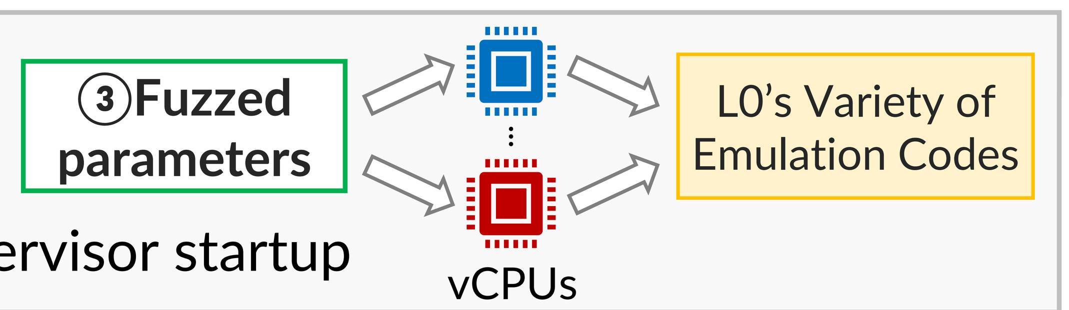


Challenge 3: Various vCPU Configurations

- vCPU behavior are determined at L0 hypervisor startup

Proposal 3: Parameter Fuzzing

- Fuzzing parameters at L0 hypervisor startup



4 Implementation

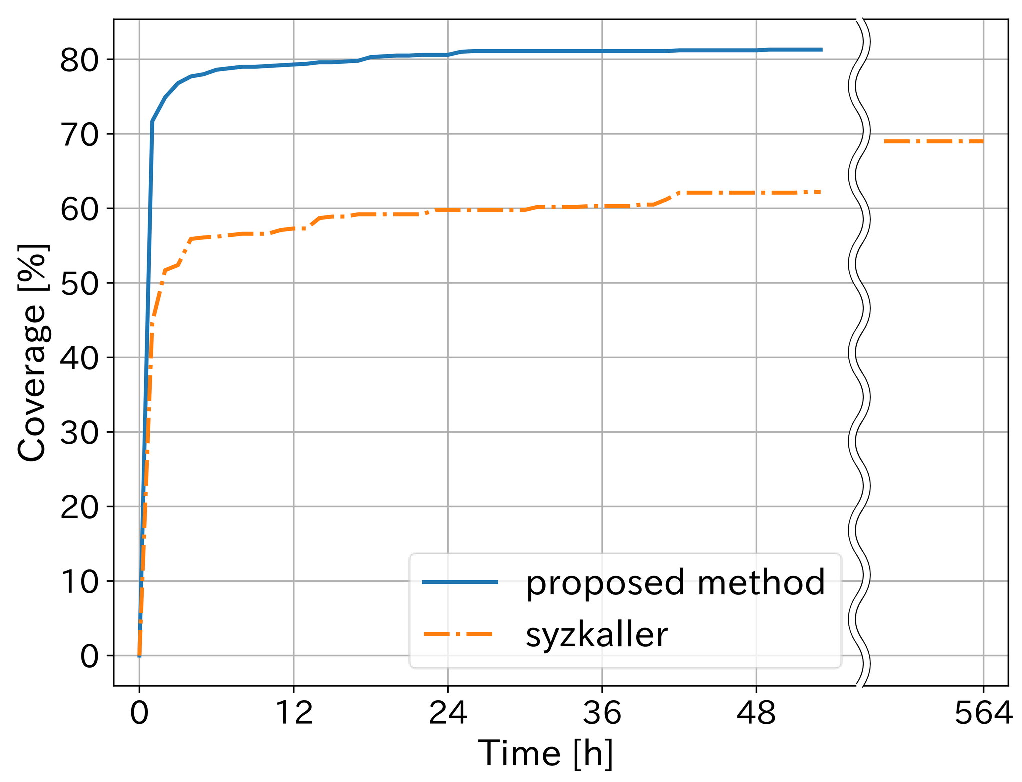
- Small Harness: based on VMXbench
- VM State Validator: ported from Bochs' VMCS checker code
- Fuzzer (Input generation): AFL++, an existing fuzzer
- Coverage collection: kcov
- Bug detection: KASAN, KCSAN and UBSAN

5 Fuzzing Experiment on KVM

- Focused on Intel VT-x nested virtualization
- Compare code coverage of KVM's nested virtualization

6 Evaluation

- The proposed method achieved 81.3% code coverage in 54 hours



- Discover a new vulnerability (CVE-2023-30456) in KVM