

Exit-Less Isolated Execution

Yushi Omote
Japan Society for the Promotion of Science

Takahiro Shinagawa
The University of Tokyo

Kazuhiko Kato
University of Tsukuba



Background Vulnerability of OSs

OSs have many vulnerabilities and attack surfaces but have unrestrained access to sensitive info. in Apps.

Need protecting Apps from compromised OSs.
(while preserving compatibility for commodity OSs or *OS transparency*)

Existing Solution

Hypervisor-based Isolated Execution

Hypervisor isolates Apps in different VMs.

Hypervisor switches VMs for OS/App context switching.

Costly CPU-mode changes, VM exits!
(1500 cycles~)

Proposal

Exit-Less Isolated Execution

Hypervisor isolates Apps in different address spaces.
(Using *Nested Page Tables*)

OS/App-level Code *transparently* inserted by Hypervisor switches address spaces without VM exits.
(Using *VMFUNC instruction*)

Lightweight switching without VM exits.
(≈ 300 cycles)

Implementation

OS-Transparent Exit-Less Context Switching

Context Switcher Code is always read-only executable while the rest has mutually-exclusive permissions.

Context Switcher Code is executed on starting/ending Apps, system calls or interrupts.

3 techniques for Hypervisors to transparently insert the code:
(1) User-level Trampoline (2) IDT Shadowing
(3) EPT-based GPT Switching

