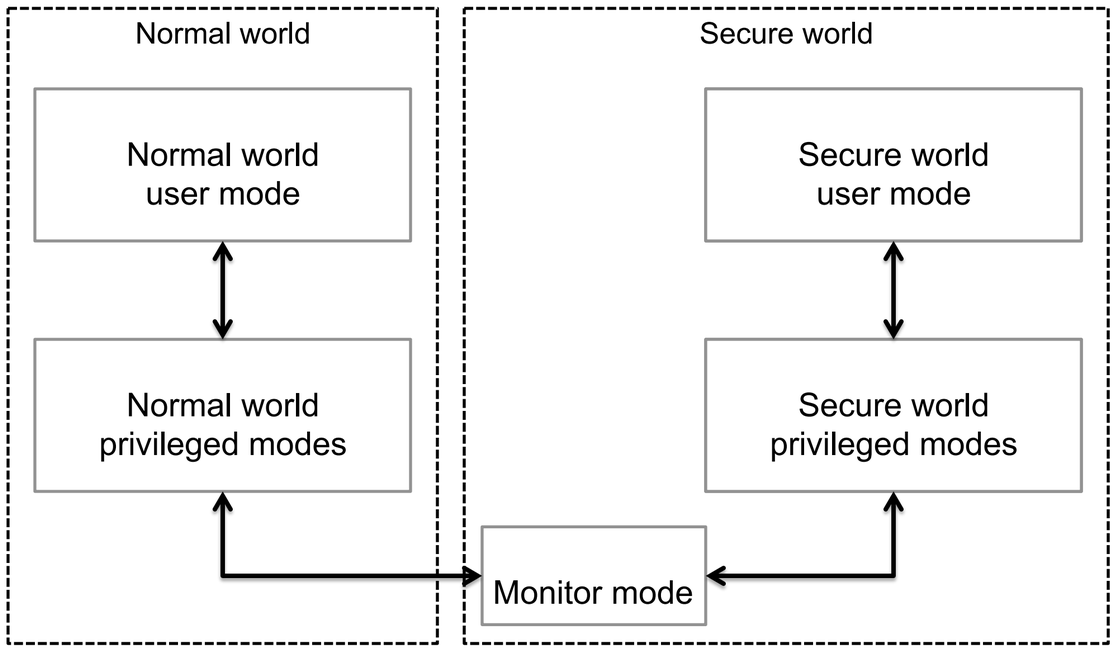
SPROBES: Enforcing Kernel Code Integrity on TrustZone Architecture

Xinyang Ge, Hayawardh Vijaykumar, Trent Jaeger

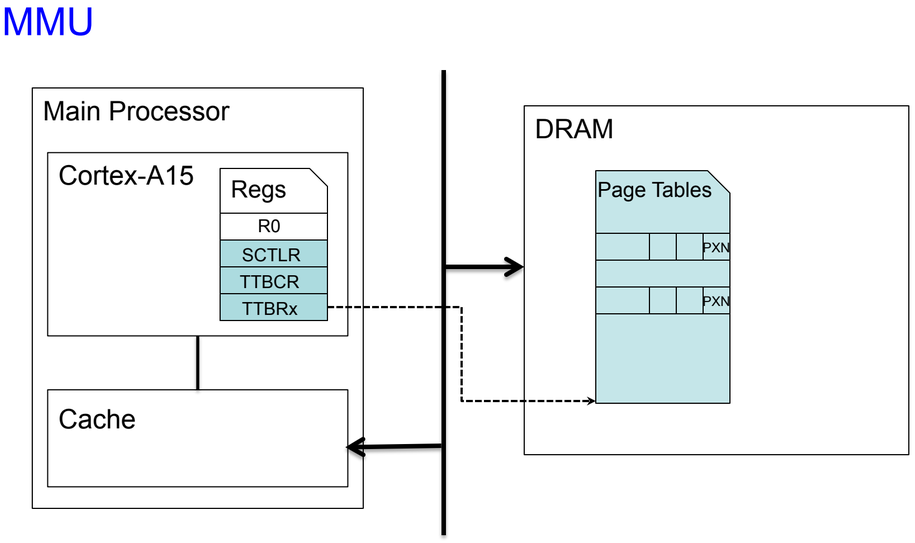
Smartphones now deploy conventional operating systems, inheriting both its functionalities and threats. Researchers have advocated using virtualization to detect attacks on operating systems but it is not practical for a smartphone due to its expenses. Current smartphone processors do have hardware support such as TrustZone for running a protected environment, but such hardware does not control the operating system operations sufficiently to enable introspection. In particular, a conventional operating system running with TrustZone still retains full control of memory management, which a rootkit can use to prevent traps on sensitive instructions or memory accesses necessary for effective introspection. In this project, we present SPROBES, a novel primitive that enables introspection of operating systems running on ARM TrustZone hardware. We demonstrate SPROBES by developing a methodology for restricting the conventional kernel (e.g., Linux) execution to approved kernel code memory.



Problem

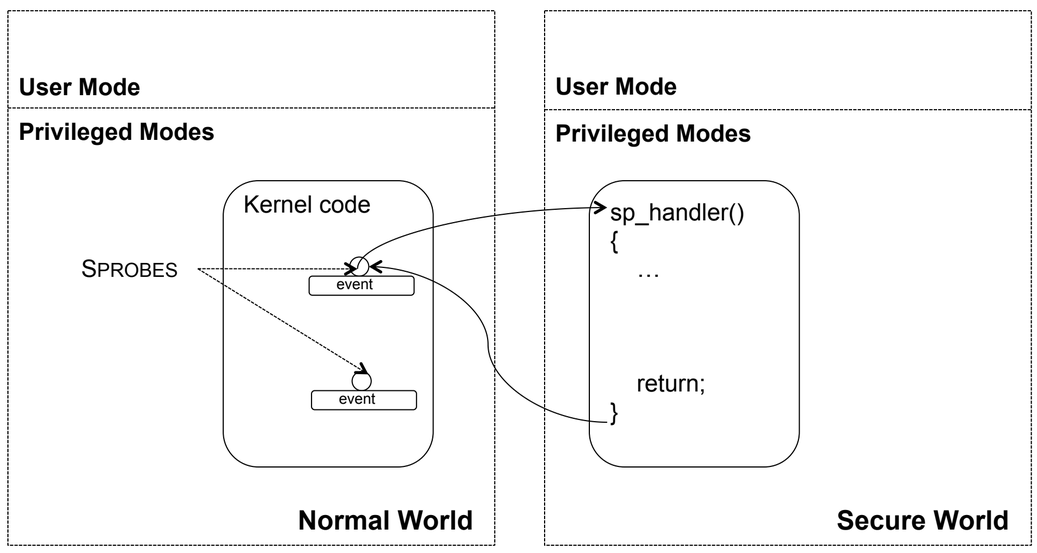
A rootkit can leverage existing code to modify virtual memory settings, allowing an attacker to modify or inject code in kernel.

- Disable the W^X protection
- Enable writing to code pages by switching to a malicious page table
- Modify page table entries
- Enable execution over user space
- Disable MMU



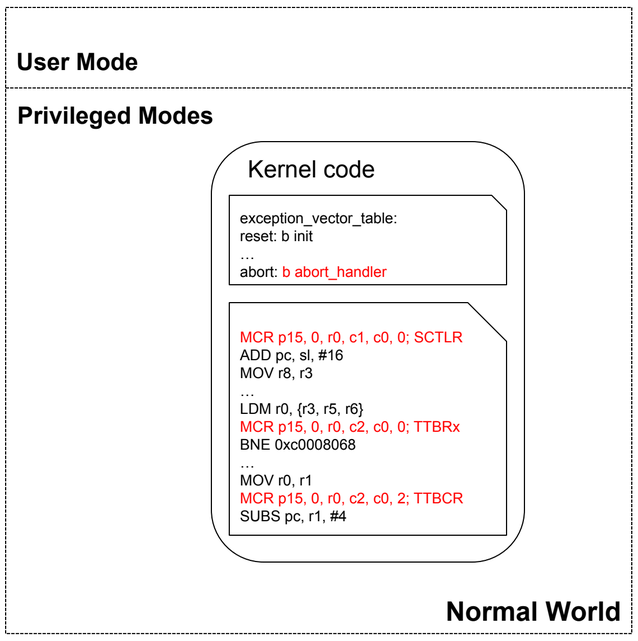
SPROBES

- Instruction-level instrumentation
- Transparent to the normal world
- Independent of normal world software



Protecting Kernel Code Integrity

- Insert SPROBES at those instructions that can disable the W^X protection.
- Insert SPROBES at those instructions that change page table root
- Write-protect the page tables and insert an SPROBE into the page fault handler
- Track modifications to the Privilege eXecute Never bits
- Insert SPROBES at those instructions that can disable the MMU.



Evaluation

We ran Linux 2.6.38 in the normal world and protected its kernel code integrity using only 12 SPROBES. Among them, 6 are for enforcing W^X protection and the MMU, 5 are to intercept changes of page table root and the last one for trapping normal world page faults to the secure world.

We also measured the hit frequency of different types of SPROBES. And the result is shown as below:

Purpose	W^X and MMU	Page table root	Page fault
Hit Frequency	N/A	313836	85982

Related Publications

Xinyang Ge, Hayawardh Vijaykumar, Trent Jaeger. "SPROBES: Enforcing Kernel Code Integrity on the TrustZone Architecture", in submission.

Sponsored by US Army CARDEC