Supervisor Mode Execution Protection

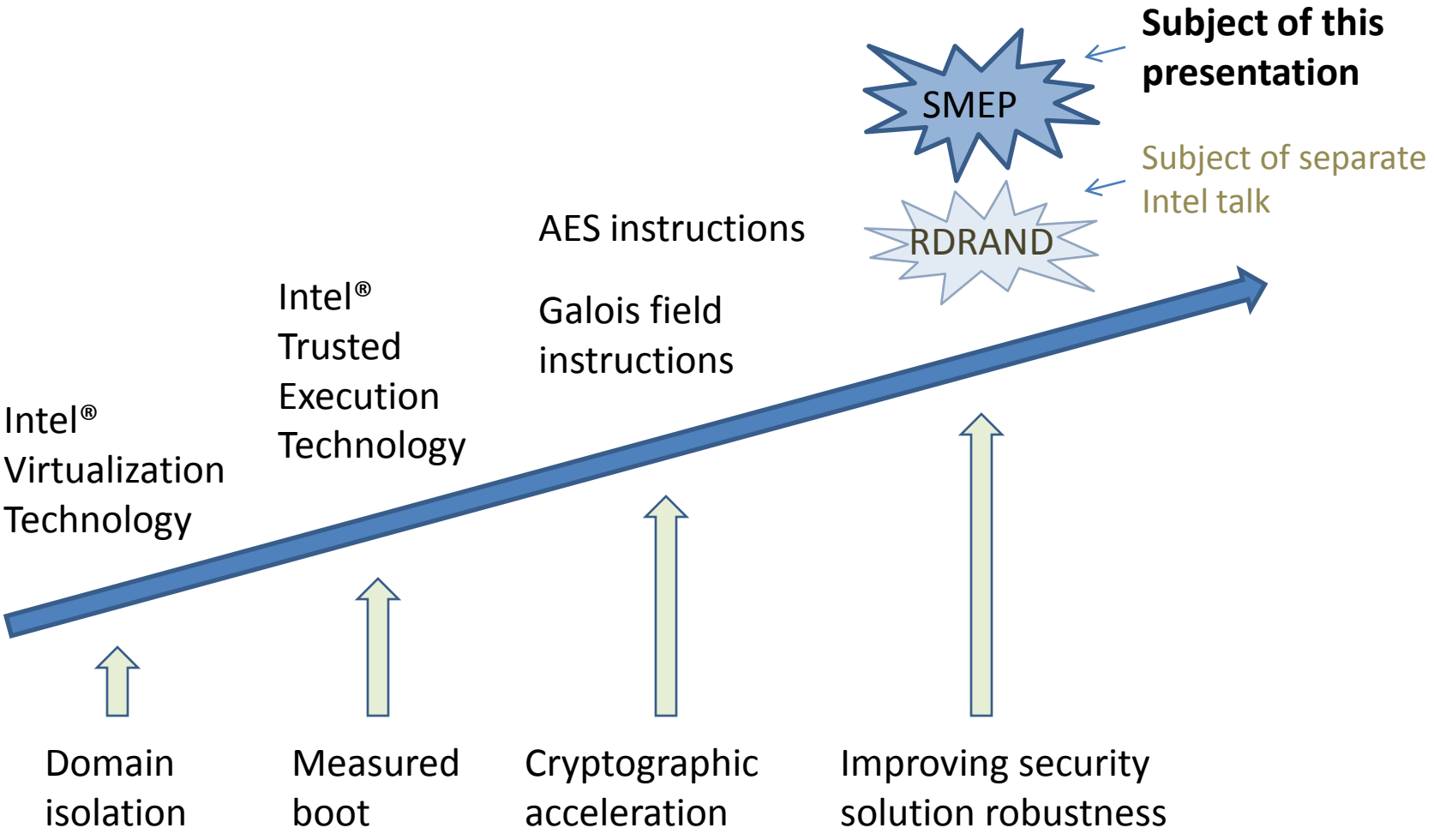
Stephen Fischer
Senior Principal Engineer
Intel® Corporation

09/21/2011

Legal Disclaimer

- INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL® PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT. INTEL PRODUCTS ARE NOT INTENDED FOR USE IN MEDICAL, LIFE SAVING, OR LIFE SUSTAINING APPLICATIONS.
- Intel may make changes to specifications and product descriptions at any time, without notice.
- All products, dates, and figures specified are preliminary based on current expectations, and are subject to change without notice.
- Intel, processors, chipsets, and desktop boards may contain design defects or errors known as errata, which may cause the product to deviate from published specifications. Current characterized errata are available on request.
- Nehalem, Merom, Wolfdale, Harpertown, Tylersburg, Penryn, Westmere, Sandy Bridge and other code names featured are used internally within Intel to identify products that are in development and not yet publicly announced for release. Customers, licensees and other third parties are not authorized by Intel to use code names in advertising, promotion or marketing of any product or services and any such use of Intel's internal code names is at the sole risk of the user
- Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of Intel products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance.
- Intel, Intel Inside, Intel Core, Intel Xeon, Intel Core2, Intel Xeon and the Intel logo are trademarks of Intel Corporation in the United States and other countries.
- *Other names and brands may be claimed as the property of others.
-
- "Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more information go to http://www.intel.com/performance"

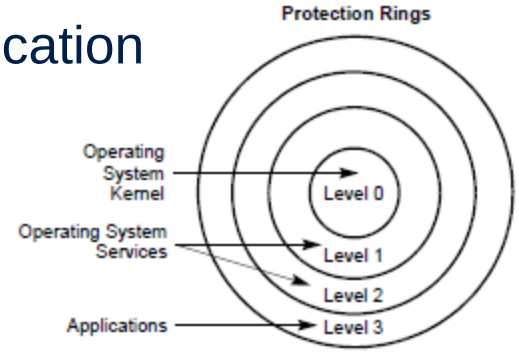
Intel® continues to provide leadership technologies in computer security

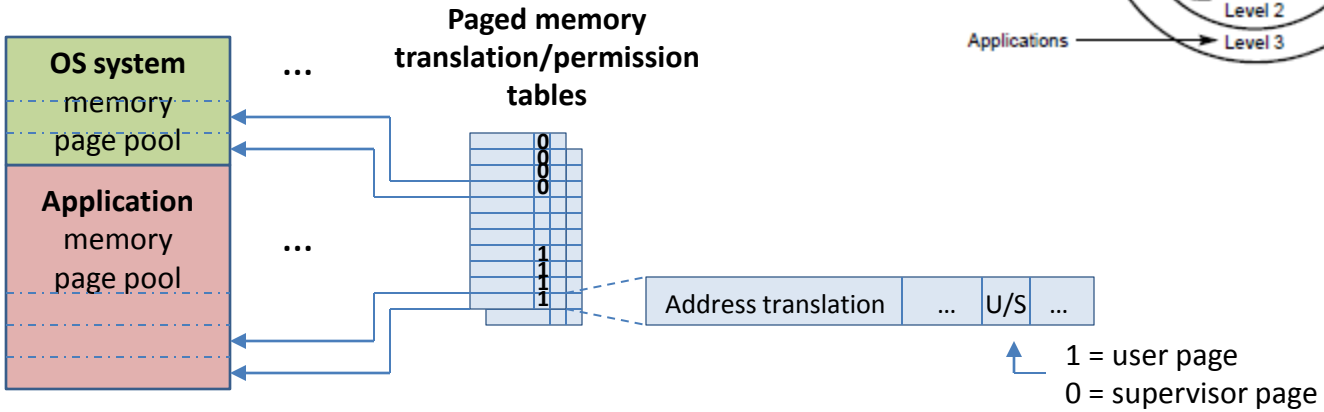


What is SMEP?

A means to prevent execution out of untrusted application memory while operating at a more privileged level

- ✓ Addresses class of Escalation of Privilege (EoP) security attacks





Historical access permission rules for code execution:

- Supervisor mode (levels 0-2): user or supervisor pages allowed (u/s==*)
- User mode (level 3): user only (u/s==1)

When SMEP is active:

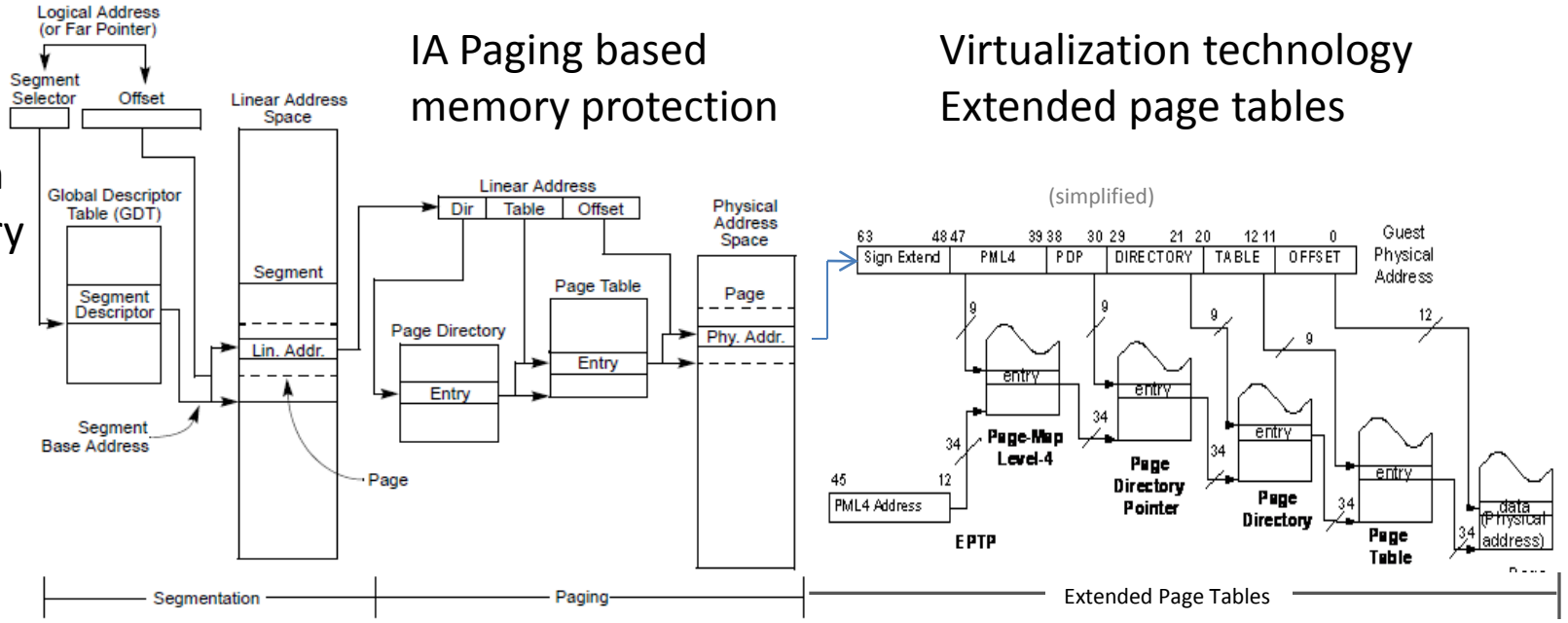
- Supervisor mode: supervisor only (u/s==0)
- User mode: user only (u/s==1)

Intel® architecture memory protection in review

Segmentation based memory protection

IA Paging based memory protection

Virtualization technology Extended page tables



Protection attribute control

Segmentation	IA paging	Extended Page Tables
Read-only	Read-only (aka XD)	Read-only
Read/write	Read/write	Read/write
Execute-only	- *	Execute-only
Execute/read	Execute/read	Execute/read
Privilege level (4)	User vs. Supervisor	

Properties modified by SMEP

* Can be accomplished via segmentation configuration

Some example escalation of privilege vulnerabilities potentially addressable by SMEP

US-CERT vulnerability 362981 - Linux kernel RDS protocol vulnerability

Kernel functions fail to properly check if a user supplied address exists in the user segment of memory. By providing a kernel address to a socket call an unprivileged user can execute arbitrary code as root

US-CERT vulnerability 537223 - GNU C library dynamic linker expands $ORIGIN in setuid library search path

Certain versions of glibc unsafely handle the $ORIGIN ELF substitution sequence which can be exploited to gain local privilege escalation.

CVE-2008-2812

The Linux kernel before 2.6.25.10 does not properly perform tty operations, which allows local users to cause a denial of service (system crash) or possibly gain privileges via vectors involving NULL pointer dereference of function pointers

CVE-2009-1863

Unspecified vulnerability in Adobe Flash Player before 9.0.246.0 and 10.x before 10.0.32.18, and Adobe AIR before 1.5.2, allows attackers to cause a denial of service (application crash) or possibly execute arbitrary code via unknown vectors, related to a "privilege escalation vulnerability.“

CVE-2010-2743

The kernel-mode drivers in Microsoft Windows XP SP3 do not properly perform indexing of a function-pointer table during the loading of keyboard layouts from disk, which allows local users to gain privileges via a crafted application, as demonstrated in the wild in July 2010 by the **Stuxnet** worm, aka "Win32k Keyboard Layout Vulnerability." NOTE: this might be a duplicate of CVE-2010-3888 or CVE-2010-3889.

- **See next slide**

CVE-2010-2743 Windows keyboard layout vulnerability

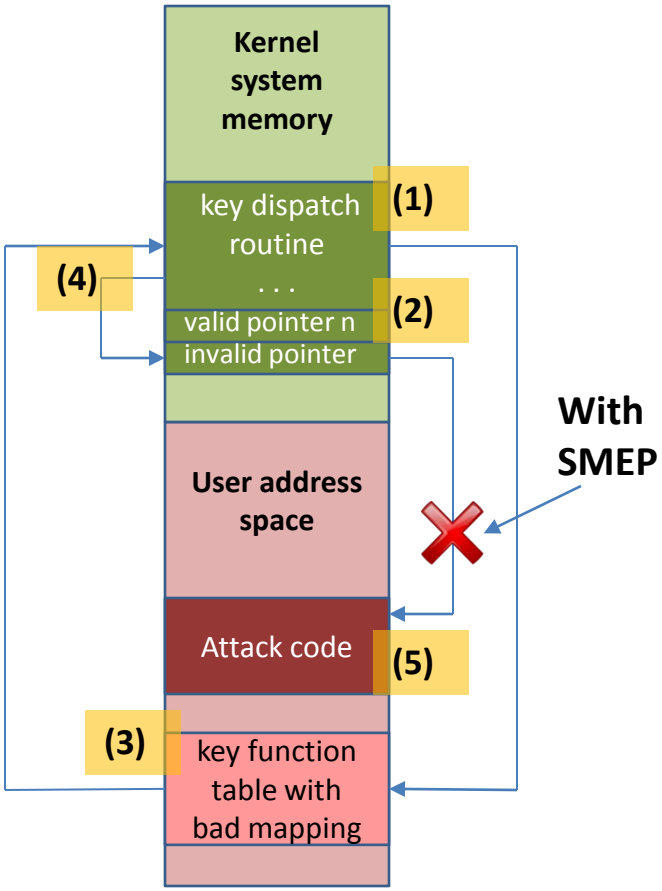
reference: http://technet.microsoft.com/en-us/security/bulletin/MS10-073

The vulnerability:

1. Keyboard dispatch routine as part of win32.sys executing with system privilege
2. Jump table exists to different key handler responses
   - Data value entry at a specific offset contains a constant that can be used to point into the user address space
3. Index to jump table controllable by user program provided keyboard layout file via function: *NtUserLoadKeyboardLayoutEx*
4. Mapping of a specific virtual keyboard code can be used to redirection execution to this offset without proper checking/validation of this input
5. Result: Execution of user provided code from supervisor mode

With SMEP:

- Attempted redirected execution to the user code space would result in the signaling of a page fault back to the OS



SMEP architectural control details

- CR4.SMEP – If 1 and in supervisor mode (CPL<3), instructions may not be executed from a linear address for which the U/S flag is 1 (user mode) in every paging-structure entry controlling the translation for the linear address
- Available in both 32 and 64 bit operating modes
- #PF: If CR4.SMEP=1, and CPL<3 and instruction is fetched from user mode page. Error code = 10001b
  - Page is present, Access was not a write (data read or code fetch), Access was in supervisor mode (CPL<3), No reserved-bit violation, Access was an instruction fetch
  - The I/D bit of the page fault error code (bit 4) will now be set when an instruction page fault occurs and ((EFER.NXE and CR4.PAE) or CR4.SMEP)
    - Previously it would be set only when EFER.NXE and CR4.PAE is true.
- SMEP is enumerated via CPUID.7.0.EBX[7]

SMEP software considerations

- TLB invalidation guidelines for software:
  - If CR4.SMEP=1, software should perform an appropriate invalidation if modifying a paging-structure entry (for an executable page) to change the U/S flag from 0 (supervisor) to 1 (user)
    - Previously this would not have been a more restrictive condition
  - If the MOV CR4 instruction is changing CR4.SMEP (0→1 or 1→0), the instruction will invalidate the TLBs.
    - Explicit TLB Invalidation is required only if virtualization software is emulating that instruction
- SMEP U/S paging attribute precedence:
  - Any page level marked as supervisor (U/S=0) will result in treatment as supervisor for SMEP enforcement
    - Existing user/supervisor privileging checking continues to require the more conservative mapping (i.e. execution in user mode (CPL=3) requires all levels to be mapped as U/S=1 (user)
- EPT and SMEP
  - Guest VMs can utilize SMEP within an EPT virtualized context
  - EPT does not support notion of ‘u/s’ attribute, and thus n/a to EPT mappings

First available with Ivy Bridge

Intel®'s Next Generation Microarchitecture
on 22nm process technology