iTLB multihit

iTLB multihit is an erratum where some processors may incur a machine check error, possibly resulting in an unrecoverable CPU lockup, when an instruction fetch hits multiple entries in the instruction TLB. This can occur when the page size is changed along with either the physical address or cache type. A malicious guest running on a virtualized system can exploit this erratum to perform a denial of service attack.

Affected processors

Variations of this erratum are present on most Intel Core and Xeon processor models. The erratum is not present on:

- non-Intel processors
- Some Atoms (Airmont, Bonnell, Goldmont, GoldmontPlus, Saltwell, Silvermont)
- Intel processors that have the PSCHANGE_MC_NO bit set in the IA32_ARCH_CAPABILITIES MSR.

Related CVEs

The following CVE entry is related to this issue:

CVE-2018-12207	Machine Check Error Avoidance on Page Size Change

Problem

Privileged software, including OS and virtual machine managers (VMM), are in charge of memory management. A key component in memory management is the control of the page tables. Modern processors use virtual memory, a technique that creates the illusion of a very large memory for processors. This virtual space is split into pages of a given size. Page tables translate virtual addresses to physical addresses.

To reduce latency when performing a virtual to physical address translation, processors include a structure, called TLB, that caches recent translations. There are separate TLBs for instruction (iTLB) and data (dTLB).

Under this errata, instructions are fetched from a linear address translated using a 4 KB translation cached in the iTLB. Privileged software modifies the paging structure so that the same linear address using large page size (2 MB, 4 MB, 1 GB) with a different physical address or memory type. After the page structure modification but before the software invalidates any iTLB entries for the linear address, a code fetch that happens on the same linear address may cause a machine-check error which can result in a system hang or shutdown.

Attack scenarios

Attacks against the iTLB multihit erratum can be mounted from malicious guests in a virtualized system.

iTLB multihit system information

The Linux kernel provides a sysfs interface to enumerate the current iTLB multihit status of the system:whether the system is vulnerable and which mitigations are active. The relevant sysfs file is:

/sys/devices/system/cpu/vulnerabilities/itlb_multihit

The possible values in this file are:

Not affected	The processor is not vulnerable.
KVM: Mitigation: Split huge pages	Software changes mitigate this issue.
KVM: Mitigation: VMX unsupported	KVM is not vulnerable because Virtual Machine Extensions (VMX) is not supported.
KVM: Mitigation: VMX disabled	KVM is not vulnerable because Virtual Machine Extensions (VMX) is disabled.
KVM: Vulnerable	The processor is vulnerable, but no mitigation enabled

Enumeration of the erratum

A new bit has been allocated in the IA32_ARCH_CAPABILITIES (PSCHANGE_MC_NO) msr and will be set on CPU's which are mitigated against this issue.

IA32_ARCH_CAPABILITIES MSR	Not present	Possibly vulnerable,check model
IA32_ARCH_CAPABILITIES[PSCHANGE_MC_NO]	'0'	Likely vulnerable,check model

IA32_ARCH_CAPABILITIES[PSCHANGE_MC_NO]	'1'	Not vulnerable

Mitigation mechanism

This erratum can be mitigated by restricting the use of large page sizes to non-executable pages. This forces all iTLB entries to be 4K, and removes the possibility of multiple hits.

In order to mitigate the vulnerability, KVM initially marks all huge pages as non-executable. If the guest attempts to execute in one of those pages, the page is broken down into 4K pages, which are then marked executable.

If EPT is disabled or not available on the host, KVM is in control of TLB flushes and the problematic situation cannot happen. However, the shadow EPT paging mechanism used by nested virtualization is vulnerable, because the nested guest can trigger multiple iTLB hits by modifying its own (non-nested) page tables. For simplicity, KVM will make large pages non-executable in all shadow paging modes.

Mitigation control on the kernel command line and KVM - module parameter

The KVM hypervisor mitigation mechanism for marking huge pages as non-executable can be controlled with a module parameter "nx_huge_pages=". The kernel command line allows to control the iTLB multihit mitigations at boot time with the option "kvm.nx_huge_pages=".

The valid arguments for these options are:

force	Mitigation is enabled. In this case, the mitigation implements non-executable huge pages in Linux kernel KVM module. All huge pages in the EPT are marked as non-executable. If a guest attempts to execute in one of those pages, the page is broken down into 4K pages, which are then marked executable.
off	Mitigation is disabled.
auto	Enable mitigation only if the platform is affected and the kernel was not booted with the "mitigations=off" command line parameter. This is the default option.

Mitigation selection guide

1. No virtualization in use

The system is protected by the kernel unconditionally and no further action is required.

2. Virtualization with trusted guests

If the guest comes from a trusted source, you may assume that the guest will not attempt to maliciously exploit these errata and no further action is required.

3. Virtualization with untrusted guests

If the guest comes from an untrusted source, the guest host kernel will need to apply iTLB multihit mitigation via the kernel command line or kvm module parameter.