TSX Async Abort (TAA) mitigation

Overview

TSX Async Abort (TAA) is a side channel attack on internal buffers in some Intel processors similar to Microachitectural Data Sampling (MDS). In this case certain loads may speculatively pass invalid data to dependent operations when an asynchronous abort condition is pending in a Transactional Synchronization Extensions (TSX) transaction. This includes loads with no fault or assist condition. Such loads may speculatively expose stale data from the same uarch data structures as in MDS, with same scope of exposure i.e. same-thread and cross-thread. This issue affects all current processors that support TSX.

Mitigation strategy

a) TSX disable - one of the mitigations is to disable TSX. A new MSR IA32_TSX_CTRL will be available in future and current processors after microcode update which can be used to disable TSX. In addition, it controls the enumeration of the TSX feature bits (RTM and HLE) in CPUID.

b) Clear CPU buffers - similar to MDS, clearing the CPU buffers mitigates this vulnerability. More details on this approach can be found in :ref:`Documentation/admin-guide/hw-vuln/mds.rst <mds>`.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\x86\[linux-master][Documentation][x86]tsx_async_abort.rst`, line 29); *backlink*

Unknown interpreted text role "ref".

Kernel internal mitigation modes

off	Mitigation is disabled. Either the CPU is not affected or tsx_async_abort=off is supplied on the kernel command line.
tsx disabled	Mitigation is enabled. TSX feature is disabled by default at bootup on processors that support TSX control.
verw	Mitigation is enabled. CPU is affected and MD_CLEAR is advertised in CPUID.
ucode needed	Mitigation is enabled. CPU is affected and MD_CLEAR is not advertised in CPUID. That is mainly for virtualization scenarios where the host has the updated microcode but the hypervisor does not expose MD_CLEAR in CPUID. It's a best effort approach without guarantee.

If the CPU is affected and the "tsx_async_abort" kernel command line parameter is not provided then the kernel selects an appropriate mitigation depending on the status of RTM and MD_CLEAR CPUID bits.

Below tables indicate the impact of tsx=on|off|auto cmdline options on state of TAA mitigation, VERW behavior and TSX feature for various combinations of MSR_IA32_ARCH_CAPABILITIES bits.

1. "tsx=off"

MSR_IA32_ARCH_CAPABILITIES bits			**Result with cmdline tsx=off**			
TAA_NO	**MDS_NO**	**TSX_CTRL_MSR**	**TSX state after bootup**	**VERW can clear CPU buffers**	**TAA mitigation tsx_async_abort=off**	**TAA mitigation tsx_async_abort=full**
0	0	0	HW default	Yes	Same as MDS	Same as MDS
0	0	1	Invalid case	Invalid case	Invalid case	Invalid case
0	1	0	HW default	No	Need ucode update	Need ucode update
0	1	1	Disabled	Yes	TSX disabled	TSX disabled
1	X	1	Disabled	X	None needed	None needed

2. "tsx=on"

MSR_IA32_ARCH_CAPABILITIES bits			**Result with cmdline tsx=on**			
TAA_NO	**MDS_NO**	**TSX_CTRL_MSR**	**TSX state after bootup**	**VERW can clear CPU buffers**	**TAA mitigation tsx_async_abort=off**	**TAA mitigation tsx_async_abort=full**
0	0	0	HW default	Yes	Same as MDS	Same as MDS
0	0	1	Invalid case	Invalid case	Invalid case	Invalid case
0	1	0	HW default	No	Need ucode update	Need ucode update

MSR_IA32_ARCH_CAPABILITIES bits			Result with cmdline tsx=on			
TAA_NO	**MDS_NO**	**TSX_CTRL_MSR**	**TSX state after bootup**	**VERW can clear CPU buffers**	**TAA mitigation tsx_async_abort=off**	**TAA mitigation tsx_async_abort=full**
0	1	1	Enabled	Yes	None	Same as MDS
1	X	1	Enabled	X	None needed	None needed

3. "tsx=auto"

MSR_IA32_ARCH_CAPABILITIES bits			Result with cmdline tsx=auto			
TAA_NO	**MDS_NO**	**TSX_CTRL_MSR**	**TSX state after bootup**	**VERW can clear CPU buffers**	**TAA mitigation tsx_async_abort=off**	**TAA mitigation tsx_async_abort=full**
0	0	0	HW default	Yes	Same as MDS	Same as MDS
0	0	1	Invalid case	Invalid case	Invalid case	Invalid case
0	1	0	HW default	No	Need ucode update	Need ucode update
0	1	1	Disabled	Yes	TSX disabled	TSX disabled
1	X	1	Enabled	X	None needed	None needed

In the tables, TSX_CTRL_MSR is a new bit in MSR_IA32_ARCH_CAPABILITIES that indicates whether MSR_IA32_TSX_CTRL is supported.

There are two control bits in IA32_TSX_CTRL MSR:

```
Bit 0: When set it disables the Restricted Transactional Memory (RTM)
       sub-feature of TSX (will force all transactions to abort on the XBEGIN instruction).
Bit 1: When set it disables the enumeration of the RTM and HLE feature
       (i.e. it will make CPUID(EAX=7).EBX{bit4} and CPUID(EAX=7).EBX{bit11} read as 0).
```