# Trust Domain Security Guidance for Developers



## Overview

Intel® Trust Domain Extensions (Intel® TDX) introduces new, architectural elements to help deploy hardware-isolated, virtual machines (VMs) called trust domains (TDs). Intel TDX is designed to isolate such VMs from the virtual-machine manager (VMM)/hypervisor, other VMs/TDs, and any other non-TD software on the platform to protect TDs from a broad range of potential attacks.

For additional information, developers can refer to the architectural overview for Intel TDX as well as detailed architecture specifications

for Intel TDX.

Because Intel TDX is a Trusted Execution Environment, software entities such as operating systems (OSes) and hypervisors that are considered privileged in other contexts are not trusted by Intel TDX-protected workloads for their confidentiality or integrity. As such, compared to traditional environments, TD developers need to take additional steps to mitigate new potential attack surfaces caused by these different trust boundaries.

This document summarizes information from multiple sources which collectively provide a description of mitigations for side channel and other attacks that are particularly relevant for the protection of Intel TDX workloads. This includes both mitigations which are already deployed in Intel platforms, as well as mitigations which may be recommended for operating system code and workloads launched inside TDs.

The guidance in this document assumes that the latest version of Intel TDX is being used; some mitigations may not be available in older versions of Intel TDX.

This document focuses on protection of the TD workloads and the Intel TDX Trusted Computing Base (TCB) software from potentially untrusted software entities outside the Intel TDX TCB.

## Software Hardening Recommendations

While Intel TDX and hardware do provide security properties and isolation, since the threat model of Intel TDX assumes the hypervisor and/or TDs may be malicious, hypervisor and TD code should still consider the potential new attack surface and apply additional hardening and defense-in-depth measures in software. In particular, TD guest operating systems and driver code need to be aware that the data from many interfaces may no longer be trusted if a malicious hypervisor is included in the guest's threat model. The Intel TDX

module architecture specification

documents which interfaces (in particular, CPUID leaves and model specific registers (MSRs)) are implemented by the Intel TDX module itself and/or are provided by hardware. Software interacting with other interfaces may need to be hardened, including code involving hypercalls, CPUID results, shared memory, MMIO, port I/O, and MSRs.

We recommend that operating system and hypervisor vendors refer to Intel's documentation about the attack surface and kernel hardening for Linux* , which provides an example of how to analyze and harden such attack surfaces. In particular, this documentation describes how the attack surface can be minimized by disabling unnecessary drivers and functionality, and how analysis and fuzzing tools can be used to harden the remaining interface code. It also discusses other topics, such as the steps needed to enable additional kernel drivers .

Intel TDX also supports mitigation of many transient execution and side channel attacks. Details of these mitigations, as well as per-issue guidance about which mitigations TD software developers may need to apply, is described in the section below.

## Transient Execution and Side Channel Attacks

Modern high-performance processors used speculative execution as a main technique to improve performance. While speculative operations do not affect the architectural state of the processor, they can affect the microarchitectural state and create incidental side channels , which are unintended communication channels formed by execution, power consumption, use of shared resources, etc. Attackers may take advantage of these incidental channels to infer data.

This section provides an overview of some of the mitigations which hypervisors and TD guests may need to apply to mitigate potential transient execution attacks and side channel attacks. It also describes how Intel TDX components, including the processor hardware as well as software components such as the Intel TDX module, support such mitigations.

Note that the Intel TDX module preserves the value of IA32_SPEC_CTRL, which is a mitigation option for several of the potential classes of attack discussed below, across transitions (for example, the value is saved after a TD guest exit, and restored before resuming that TD). The Intel TDX module also mandates that the relevant hardware mitigations are enumerated to the guest, so that a malicious hypervisor cannot prevent TDs from using these mitigations. Further details about the mitigations supported and provided by Intel TDX can be found in the Intel TDX module architecture specification .

### Transient Execution Attack Mitigations

Bound Check Bypass (Spectre variant 1)

Mitigating Bounds Check Bypass is a software responsibility, and the existing Intel guidance still applies in the context of Intel TDX. In particular, TD software should be aware that potentially untrusted software running outside a TD may be able to influence conditional branch predictions of software running in a TD.

Note that data from previously trusted sources (such as reads from MSRs or hardware devices) may also be under attacker control in the Intel TDX threat model. TD software may wish to consider applying additional mitigations as

appropriate.

The Linux kernel hardening documentation mentioned above contains discussion of mitigating Bounds Check Bypass attacks in a TD, including details on use of static analysis. Developers can find this documentation on the Intel GitHub

*.

Intel provides mitigations for any potential Bounds Check Bypass attacks in the Intel TDX module code.

Speculative Store Bypass (Spectre variant 4) and Fast Store Forwarding Predictor

Intel's TDX module enables the Speculative Store Bypass Disable (SSBD) control while the Intel TDX module is executing, to provide defense-in-depth against any potential Speculative Store Bypass attacks against the TDX module. The IA32_SPEC_CTRL.SSBD control is also always available to TDs. Intel TDX is also designed to isolate any Fast Software Forwarding predictions.

TD software can apply standard software mitigations to mitigate Speculative Store Bypass

and/or
any disclosure gadgets which could be combined with Fast Store Forwarding Predictor

behavior, as described in Intel's existing guidance. Where this is not feasible, or where defense-in-depth is desired, the IA32_SPEC_CTRL.SSBD hardware control can be used to mitigate such gadgets.

Branch Target Injection (Spectre variant 2)

For Branch Target Injection

, Intel's TDX hardware and software work together towards the aim that:

- The predicted targets of near indirect branches executed in a TD cannot be controlled by software executing outside a TD, and the predicted targets of near indirect branches executed in a TD cannot be controlled by a different TD.
- The predicted targets of near indirect branches executed outside Secure Arbitration Mode (SEAM) cannot be controlled by software executing in a TD.
- The predicted targets of near indirect branches executed in SEAM root mode, such as the Intel TDX module and SEAMLDR, cannot be controlled by software running outside SEAM root mode.

Where relevant, TD software can apply standard mitigations for potential Branch Target Injection attacks inside a TD. The Intel TDX module ensures the relevant hardware mitigation controls (in particular, IBRS

and IBPB

) are available to TDs.

Branch History Injection and Intra-mode Branch Target Injection

For BHI and IMBTI

, the Intel TDX module currently clears branch history after TD exits and after the hypervisor uses SEAMCALL, mitigating potential Branch History Injection attacks against TDs, hypervisors, and the Intel TDX module itself. Future updates or newer processors may use different techniques to mitigate Branch History Injection attacks.

TD software can follow existing guidance to mitigate any potential Branch History Injection or intra-mode BTI attacks within a TD, using software mitigation techniques or the indirect branch predictor controls made available to the TD.

Note that an updated Intel TDX module may be needed for some indirect predictor controls to be enumerated to TDs; however, these controls can still be used by TDs if needed.

Other Attacks Mitigated in Hardware

Note that many previous attacks related to microarchitecture or transient execution have been mitigated in hardware on processors which support Intel TDX, as documented in the Intel TDX module architecture specification.

TD software should continue to consult the enumerations exposed via the CPUID and/or IA32_ARCH_CAPABILITIES.

## Other Side Channel Mitigations

Timing Side Channels

Other existing attacks (including side channel attacks based on microarchitectural resource sharing, such as caches) continue to be a concern for TD guests. The guidance and hardware features described in the documents below may be of particular interest to authors of software running in or as TDs.

- Secure Coding Guidance for Mitigating Timing Side Channels in Cryptographic Implementations
- Data Operand Independent Timing Mode Instruction Set Architecture Guidance
- Data Operand Independent Timing Instructions
- Security Best Practices for Side Channel Resistance

Running Average Power Limit (RAPL) Energy Reporting

Intel TDX attestation requires that RAPL

filtering is enabled at boot time, mitigating potential attacks based on RAPL energy reporting; such filtering cannot be disabled without a processor reset, as described in the existing RAPL guidance.

Frequency Throttling

Intel provides guidance about how cryptographic implementations can mitigate frequency throttling side-channels. It is recommended that TD developers refer to the guidance to assess cryptographic implementations inside TDs and apply mitigations accordingly. Refer to Frequency Throttling Side Channel Guidance for Cryptography Implementations

for more information.

Single-stepping and Zero-stepping of a TD

When in production mode, Intel TDX prevents the ability for the platform to single-step (instruction by instruction) through the TD or to cause the re-execution of instructions (zero-stepping). While zero- and single-stepping are not attacks by themselves, they are often used as precursors for developing attacks, or may be part of an attack scheme that is devised to amplify side-channel behaviors.

Section 17.4 of the Intel TDX Module v1.5 Base Architecture Specification describes how the Intel TDX module helps mitigate single-step and zero-step attacks against TDs, including expectations for how TD software can participate in these mitigations.

## References

- CPUID Enumeration and Architectural MSRs

- MKTME Side Channel Impact on Intel TDX