CWE: An Outsider’s Perspective (or: a Retrospective on the new Microarchitectural Weaknesses)

Scott Constable (Intel Labs)

HW CWE SIG Meeting 2024-03-08

Outline

- Recap of microarchitectural weaknesses before CWE 4.14
- Timeline of the new microarchitectural weakness proposal
- Overview the new weaknesses, with an example
- Clarify nuances

Recap: Pre-4.14

CVEID: CVE-2021-0089

Description: Observable response discrepancy in some Intel(R) Processors may allow an authorized user to potentially enable information disclosure via local access.

CVSS Base Score: 6.5 Medium

CVSS Vector: CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:C/C:H/I:N/A:N

Source: INTEL-SA-00516

Recap: Pre-4.14

CVEID: CVE-2021-0089

Description: Observable response discrepancy in some Intel(R) Processors may allow an authorized user to potentially enable information disclosure via local access.

CVSS Base Score: 6.5 Medium

CVSS Vector: CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:C/C:H/I:N/A:N

Source: INTEL-SA-00516

Overview

An in-domain transient execution attack methodology known as Speculative Code Store Bypass (SCSB) may allow data values to be inferred during the transient execution of self-modifying code (SMC) on some Intel processors. SCSB has been assigned CVE-2021-0089 with a base score of CVSS 6.5 CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:C/C:H/I:N/A:N. Refer to the Affected Processors table for a list of processors affected by SCSB.

Source: Speculative Code Store Bypass (intel.com)

Recap: Pre-4.14

CVEID: CVE-2021-0089

Description: Observable response discrepancy in some Intel(R) Processors may allow an authorized user to potentially enable information disclosure via local access.

CVSS Base Score: 6.5 Medium

CVSS Vector: CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:C/C:H/I:N/A:N

Source: INTEL-SA-00516

CWE-204: Observable Response Discrepancy

Weakness ID: 204
Vulnerability Mapping: ALLOWED
Abstraction: Base

View customized information: Conceptual | Operational | Mapping Friendly | **Complete** | Custom

Description

The product provides different responses to incoming requests in a way that reveals internal state information to an unauthorized actor outside of the intended control sphere.

Source: CWE - CWE-204: Observable Response Discrepancy (4.14) (mitre.org)

Recap: Pre-4.14

"Speculative Code Store Bypass (SCSB) may allow data values to be inferred during the transient execution of self-modifying code (SMC)"
Source: Speculative Code Store Bypass (intel.com)

CWE	Title	Description
1037	Processor Optimization Removal or Modification of Security-critical Code	The developer builds a security-critical protection mechanism into the software, but the processor optimizes the execution of the program such that the mechanism is removed or modified.
1264	Hardware Logic with Insecure De-Synchronization between Control and Data Channels	The hardware logic for error handling and security checks can incorrectly forward data before the security check is complete.
1303	Non-Transparent Sharing of Microarchitectural Resources	Hardware resources shared across execution contexts (e.g., caches and branch predictors) can violate the expected architecture isolation between contexts.
1342	Information Exposure through Microarchitectural State after Transient Execution	The processor does not properly clear microarchitectural state after incorrect microcode assists or speculative execution, resulting in transient execution.

Recap: Pre-4.14

"Speculative Code Store Bypass (SCSB) may allow data values to be inferred during the transient execution of self-modifying code (SMC)"
Source: Speculative Code Store Bypass (intel.com)

CWE	Title	Description
1037	Processor Optimization Removal or Modification of Security-critical Code	The developer builds a security-critical protection mechanism into the software, but the processor optimizes the execution of the program such that the mechanism is removed or modified.
1264		The hardware logic for error handling and security checks can incorrectly forward data before the security check is complete.
1303	Microarchitectural Resources	Hardware resources shared across execution contexts (e.g., caches and branch predictors) can violate the expected architecture isolation between contexts.
1342	Information Exposure through Microarchitectural State after Transient Execution	The processor does not properly clear microarchitectural state after incorrect microcode assists or speculative execution, resulting in transient execution.

CWE-1037 isn't relevant. There isn't any particular software protection mechanism that is bypassed by SCSB

Recap: Pre-4.14

"Speculative Code Store Bypass (SCSB) may allow data values to be inferred during the transient execution of self-modifying code (SMC)"
Source: Speculative Code Store Bypass (intel.com)

CWE	Title	Description
1037	Processor Optimization Removal or Modification of Security-critical Code	The developer builds a security-critical protection mechanism into the software, but the processor optimizes the execution of the program such that the mechanism is removed or modified.
~~1264~~		The hardware logic for error handling and security checks can incorrectly forward data before the security check is complete.
1303	Microarchitectural Resources	Hardware resources shared across execution contexts (e.g., caches and branch predictors) can violate the expected architecture isolation between contexts.
1342	Information Exposure through Microarchitectural State after Transient Execution	The processor does not properly clear microarchitectural state after incorrect microcode assists or speculative execution, resulting in transient execution.

CWE-1264 isn't relevant. An SMC machine clear doesn't involve error handling or security checks.

Recap: Pre-4.14

"Speculative Code Store Bypass (SCSB) may allow data values to be inferred during the transient execution of self-modifying code (SMC)"
Source: Speculative Code Store Bypass (intel.com)

CWE	Title	Description
1037	Processor Optimization Removal or Modification of Security-critical Code	The developer builds a security-critical protection mechanism into the software, but the processor optimizes the execution of the program such that the mechanism is removed or modified.
1264	[illegible]	The hardware logic for error handling and security checks can incorrectly forward data before the security check is complete.
13[illegible]	[illegible] Microarchitectural Resources	Hardware resources shared across execution contexts (e.g., caches and branch predictors) can violate the expected architecture isolation between contexts.
1342	Information Exposure through Microarchitectural State after Transient Execution	The processor does not properly clear microarchitectural state after incorrect microcode assists or speculative execution, resulting in transient execution.

SCSB doesn't require shared resources to exploit, for example it could potentially be exploited within a JIT runtime (within a single process) or remotely over a network.

Recap: Pre-4.14

"Speculative Code Store Bypass (SCSB) may allow data values to be inferred during the transient execution of self-modifying code (SMC)"
Source: Speculative Code Store Bypass (intel.com)

CWE	Title	Description
1037	Processor Optimization Removal or Modification of Security-critical Code	The developer builds a security-critical protection mechanism into the software, but the processor optimizes the execution of the program such that the mechanism is removed or modified.
1264	[illegible]	The hardware logic for error handling and security checks can incorrectly forward data before the security check is complete.
1303	Microarchitectural Resources	Hardware resources shared across execution contexts (e.g., caches and branch predictors) can violate the expected architecture isolation between contexts.
1[illegible]2	Information Exposure through Microarchitectural State after Transient Execution	The processor does not properly clear microarchitectural state after incorrect microcode assists or speculative execution, resulting in transient execution.

Clearing microarchitectural state before or after an SMC machine clear is impractical for many commodity processors

Timeline:

- Oct. ’21: Priya and Scott first discuss HW CWE challenges

Timeline:

- Oct. ’21: Priya and Scott first discuss HW CWE challenges
- Apr. ’22: Intel submits initial proposal for one new HW CWE

Timeline:

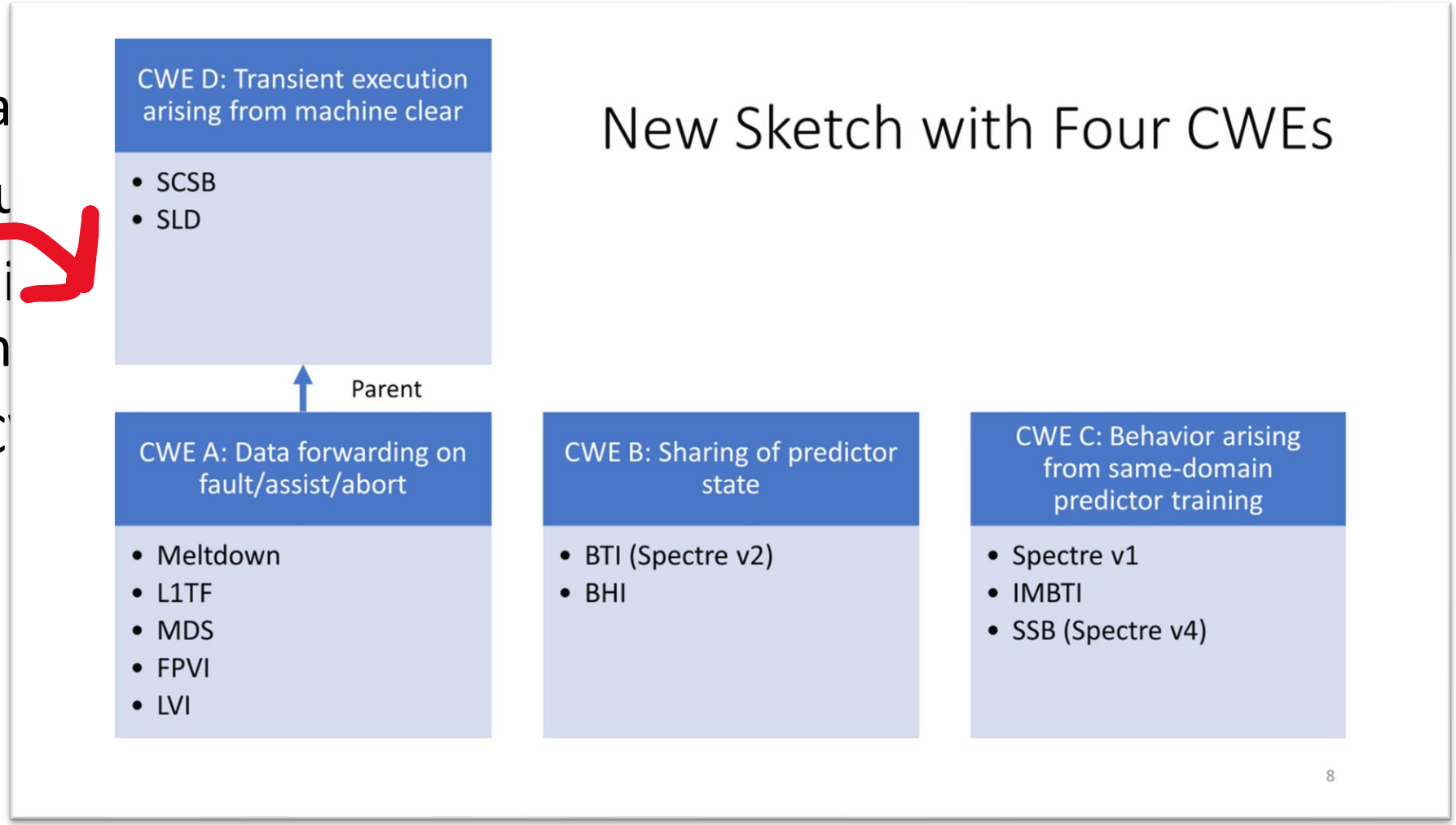
- Oct. ’21: Priya and Scott first discuss HW CWE challenges
- Apr. ’22: Intel submits initial proposal for one new HW CWE
- Sep. ’22: First discussion in the CWE HW SIG; decision to expand the proposal into multiple new CWEs

Timeline:

- Oct. ’21: Priya and Scott first discuss HW CWE challenges
- Apr. ’22: Intel submits initial proposal for one new HW CWE
- Sep. ’22: First discussion in the CWE HW SIG; decision to expand the proposal into multiple new CWEs
- Oct. ’22: New CWE hierarchy presented and discussed in HW SIG

Timeline:

- Oct. ’21: Priya a
- Apr. ’22: Intel su
- Sep. ’22: First di
  proposal into m
- Oct. ’22: New C

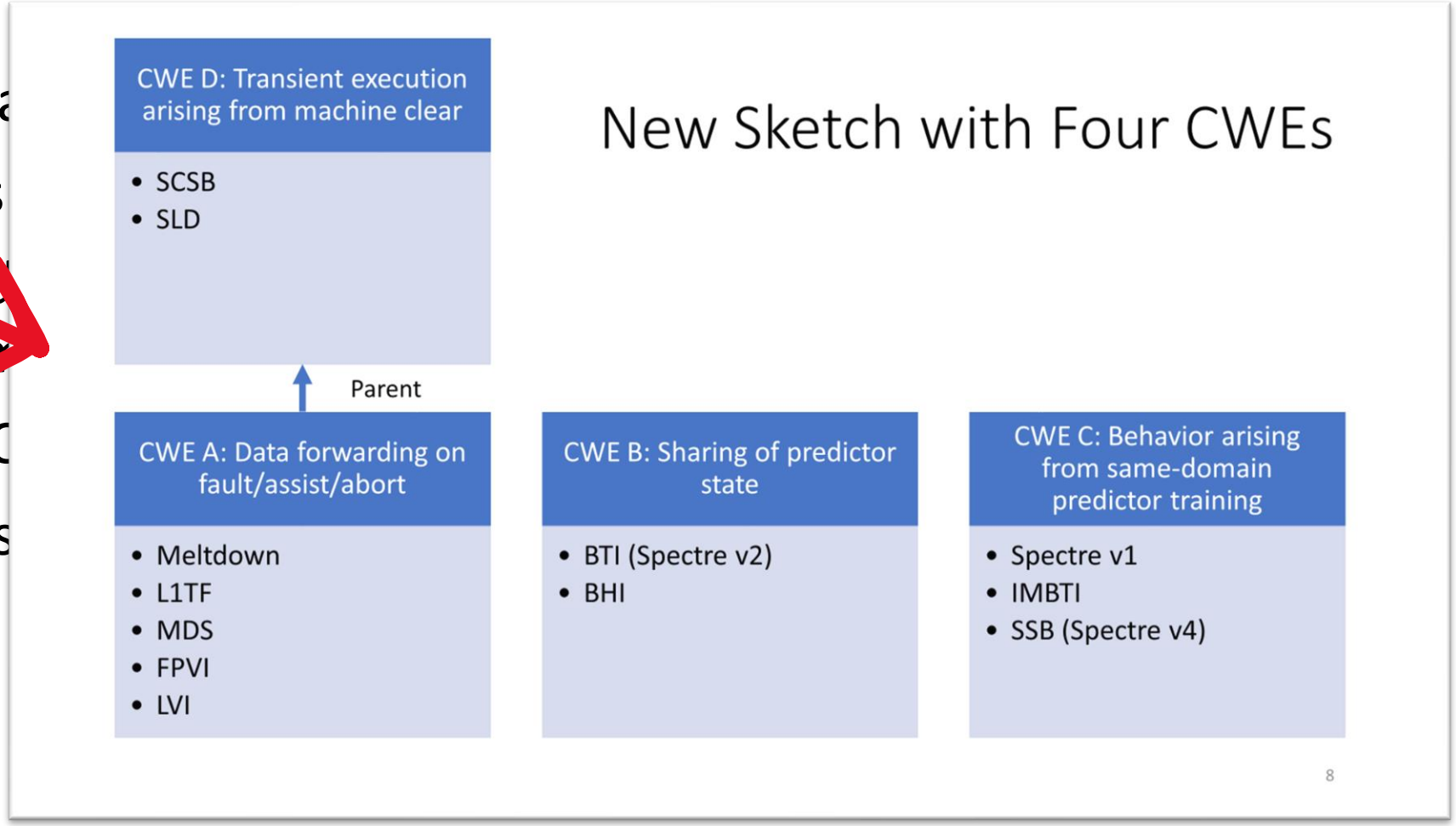


Timeline:

- Oct. ’21: Priya and Scott first discuss HW CWE challenges
- Apr. ’22: Intel submits initial proposal for one new HW CWE
- Sep. ’22: First discussion in the CWE HW SIG; decision to expand the proposal into multiple new CWEs
- Oct. ’22: New CWE hierarchy presented and discussed in HW SIG
- Dec. ’22: Intel submits draft documentation for four new CWEs

Timeline:

- Oct. ’21: Priya a
- Apr. ’22: Intel s
- Sep. ’22: First d
  proposal into m
- Oct. ’22: New C
- Dec. ’22: Intel s



Timeline:

- Oct. ’21: Priya and Scott first discuss HW CWE challenges
- Apr. ’22: Intel submits initial proposal for one new HW CWE
- Sep. ’22: First discussion in the CWE HW SIG; decision to expand the proposal into multiple new CWEs
- Oct. ’22: New CWE hierarchy presented and discussed in HW SIG
- Dec. ’22: Intel submits draft documentation for four new CWEs
- Jan.-Sep. ’23: Collaborative effort to refine proposal

Timeline:

- Oct. ’21: Priya and Scott first discuss HW CWE challenges
- Apr. ’22: Intel submits initial proposal for one new HW CWE
- Sep. ’22: First discussion in the CWE HW SIG; decision to expand the proposal into multiple new CWEs
- Oct. ’22: New CWE hierarchy presented and discussed in HW SIG
- Dec. ’22: Intel submits draft documentation for four new CWEs
- Jan.-Sep. ’23: Collaborative effort to refine proposal
- Oct. ’23: MITRE formally accepts the proposal outline

Timeline:

- Oct. ’21: Priya and Scott first discuss HW CWE challenges
- Apr. ’22: Intel submits initial proposal for one new HW CWE
- Sep. ’22: First discussion in the CWE HW SIG; decision to expand the proposal into multiple new CWEs
- Oct. ’22: New CWE hierarchy presented and discussed in HW SIG
- Dec. ’22: Intel submits draft documentation for four new CWEs
- Jan.-Sep. ’23: Collaborative effort to refine proposal
- Oct. ’23: MITRE formally accepts the proposal outline
- Feb. ’23: Four new CWEs incorporated into CWE 4.14

Today: CWE 4.14

CWE	Title	Description
1420	Exposure of Sensitive Information during Transient Execution	A processor event or prediction may allow incorrect operations (or correct operations with incorrect data) to execute transiently, potentially exposing data over a covert channel.
1421	Exposure of Sensitive Information in Shared Microarchitectural Resource during Transient Execution	A processor event may allow transient operations to access architecturally restricted data (for example, in another address space) in a shared microarchitectural resource (for example, a CPU cache), potentially exposing the data over a covert channel.
1422	Exposure of Sensitive Information caused by Incorrect Data Forwarding during Transient Execution	A processor event or prediction may allow incorrect or stale data to be forwarded to transient operations, potentially exposing data over a covert channel.
1423	Exposure of Sensitive Information caused by Shared Microarchitectural Predictor State that influences Transient Execution	Shared microarchitectural predictor state may allow code to influence transient execution across a hardware boundary, potentially exposing data that is accessible beyond the boundary over a covert channel.

Today: CWE 4.14

CWE	Title	Description
1420	Exposure of Sensitive Information during Transient Execution	A processor event or prediction may allow incorrect operations (or correct operations with incorrect data) to execute transiently, potentially exposing data over a covert channel.
1421	Exposure of Sensitive Information in Shared Microarchitectural Resource during Transient Execution	A processor event may allow transient operations to access architecturally restricted data (for example, in another address space) in a shared microarchitectural resource (for example, a CPU cache), potentially exposing the data over a covert channel.
1422	Exposure of Sensitive Information caused by Incorrect Data Forwarding during Transient Execution	A processor event or prediction may allow incorrect or stale data to be forwarded to transient operations, potentially exposing data over a covert channel.
1423	Exposure of Sensitive Information caused by Shared Microarchitectural Predictor State that influences Transient Execution	Shared microarchitectural predictor state may allow code to influence transient execution across a hardware boundary, potentially exposing data that is accessible beyond the boundary over a covert channel.

Today: CWE 4.14

"Speculative Code Store Bypass (SCSB) may allow data values to be inferred during the transient execution of self-modifying code (SMC)"
Source: Speculative Code Store Bypass (intel.com)

CWE	Title	Description
1420	Exposure of Sensitive Information during Transient Execution	A processor event or prediction may allow incorrect operations (or correct operations with incorrect data) to execute transiently, potentially exposing data over a covert channel.
1421	Exposure of Sensitive Information in Shared Microarchitectural Resource during Transient Execution	A processor event may allow transient operations to access architecturally restricted data (for example, in another address space) in a shared microarchitectural resource (for example, a CPU cache), potentially exposing the data over a covert channel.
1422	Exposure of Sensitive Information caused by Incorrect Data Forwarding during Transient Execution	A processor event or prediction may allow incorrect or stale data to be forwarded to transient operations, potentially exposing data over a covert channel.
1423	Exposure of Sensitive Information caused by Shared Microarchitectural Predictor State that influences Transient Execution	Shared microarchitectural predictor state may allow code to influence transient execution across a hardware boundary, potentially exposing data that is accessible beyond the boundary over a covert channel.

Today: CWE 4.14

"Speculative Code Store Bypass (SCSB) may allow data values to be inferred during the transient execution of self-modifying code (SMC)"
Source: Speculative Code Store Bypass (intel.com)

CWE	Title	Description
1420	Exposure of Sensitive Information during Transient Execution	A processor event or prediction may allow incorrect operations (or correct operations with incorrect data) to execute transiently, potentially exposing data over a covert channel.

New Description:

Reminder – Original Description:

CVEID: CVE-2021-0089

Description: Observable response discrepancy in some Intel(R) Processors may allow an authorized user to potentially enable information disclosure via local access.

CVSS Base Score: 6.5 Medium

CVSS Vector: CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:C/C:H/I:N/A:N

Source: INTEL-SA-00516

Today: CWE 4.14

"Speculative Code Store Bypass (SCSB) may allow data values to be inferred during the transient execution of self-modifying code (SMC)"
Source: Speculative Code Store Bypass (intel.com)

CWE	Title	Description
1420	Exposure of Sensitive Information during Transient Execution	A processor event or prediction may allow incorrect operations (or correct operations with incorrect data) to execute transiently, potentially exposing data over a covert channel.

New Description: A machine clear triggered by self-modifying code

Reminder – Original Description:

CVEID: CVE-2021-0089

Description: Observable response discrepancy in some Intel(R) Processors may allow an authorized user to potentially enable information disclosure via local access.

CVSS Base Score: 6.5 Medium

CVSS Vector: CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:C/C:H/I:N/A:N

Source: INTEL-SA-00516

Today: CWE 4.14

"Speculative Code Store Bypass (SCSB) may allow data values to be inferred during the transient execution of self-modifying code (SMC)"
Source: Speculative Code Store Bypass (intel.com)

CWE	Title	Description
1420	Exposure of Sensitive Information during Transient Execution	A processor event or prediction may allow incorrect operations (or correct operations with incorrect data) to execute transiently, potentially exposing data over a covert channel.

New Description: A machine clear triggered by self-modifying code may allow incorrect operations to execute transiently,

Reminder – Original Description:

CVEID: CVE-2021-0089

Description: Observable response discrepancy in some Intel(R) Processors may allow an authorized user to potentially enable information disclosure via local access.

CVSS Base Score: 6.5 Medium

CVSS Vector: CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:C/C:H/I:N/A:N

Source: INTEL-SA-00516

Today: CWE 4.14

"Speculative Code Store Bypass (SCSB) may allow data values to be inferred during the transient execution of self-modifying code (SMC)"
Source: Speculative Code Store Bypass (intel.com)

CWE	Title	Description
1420	Exposure of Sensitive Information during Transient Execution	A processor event or prediction may allow incorrect operations (or correct operations with incorrect data) to execute transiently, potentially exposing data over a covert channel.

New Description: A machine clear triggered by self-modifying code may allow incorrect operations to execute transiently, potentially exposing data over a covert channel.

Reminder – Original Description:

CVEID: CVE-2021-0089

Description: Observable response discrepancy in some Intel(R) Processors may allow an authorized user to potentially enable information disclosure via local access.

CVSS Base Score: 6.5 Medium

CVSS Vector: CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:C/C:H/I:N/A:N

Source: INTEL-SA-00516

Today: CWE 4.14

"Speculative Code Store Bypass (SCSB) may allow data values to be inferred during the transient execution of self-modifying code (SMC)"
Source: Speculative Code Store Bypass (intel.com)

CWE	Title	Description
1420	Exposure of Sensitive Information during Transient Execution	A processor event or prediction may allow incorrect operations (or correct operations with incorrect data) to execute transiently, potentially exposing data over a covert channel.

New Description: A machine clear triggered by self-modifying code may allow incorrect operations to execute transiently, potentially exposing data over a covert channel.

Reminder – Original Description:

CVEID: CVE-2021-0089

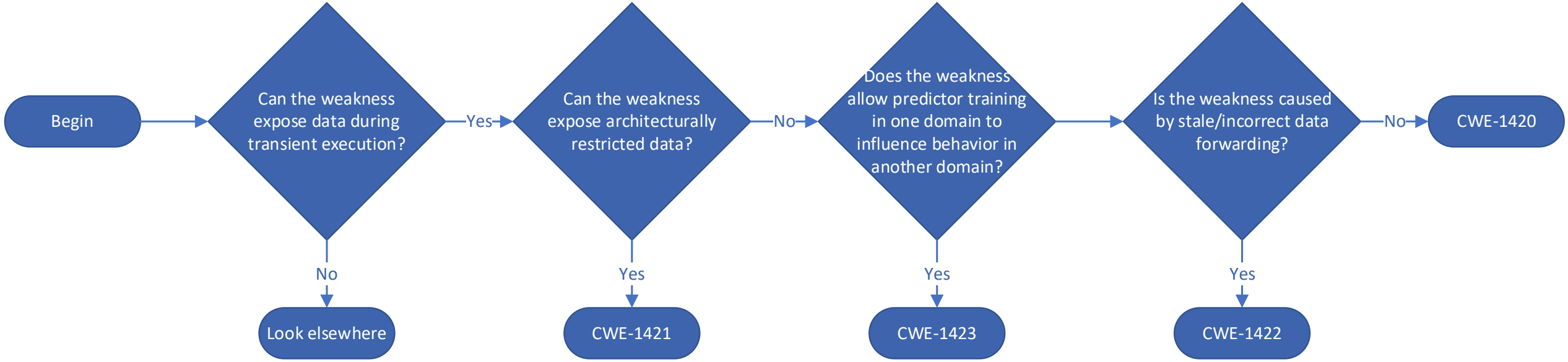
Description: Observable response discrepancy in some Intel(R) Processors may allow an authorized user to potentially enable information disclosure via local access.

CVSS Base Score: 6.5 Medium

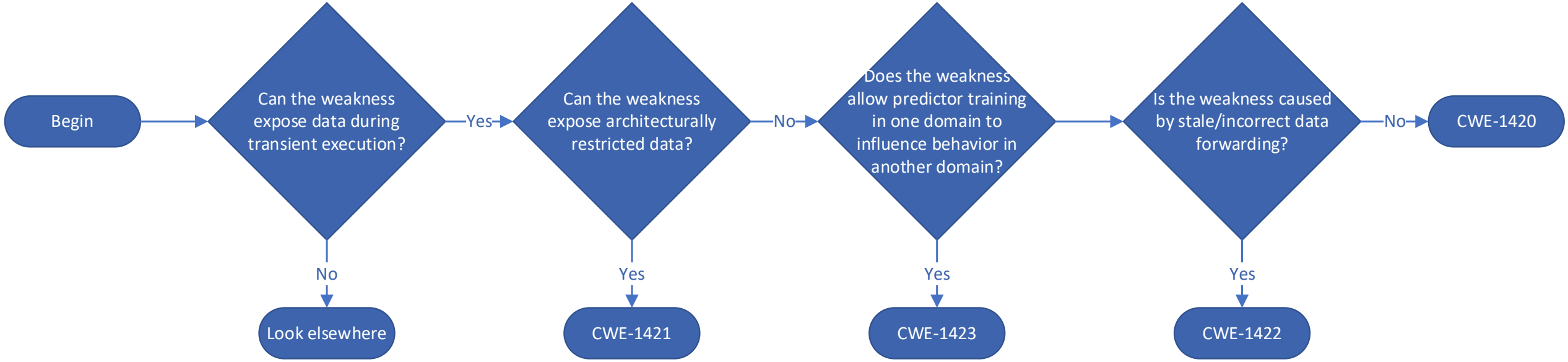
CVSS Vector: CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:C/C:H/I:N/A:N

Source: INTEL-SA-00516

Microarchitectural Weakness Flowchart



Microarchitectural Weakness Flowchart



Vulnerability Mapping Notes

Usage: ALLOWED *(this CWE ID could be used to map to real-world vulnerabilities)*

Reason: Acceptable-Use

Rationale:
This CWE entry is at the Base level of abstraction, which is a preferred level of abstraction for mapping to the root causes of vulnerabilities

Comments:
Use only when the weakness arises from forwarding of incorrect/stale data, and the data is not architecturally restricted (that is, the forwarded data is accessible within the current processor context).

If a weakness arises from forwarding of incorrect/stale data that is not accessible within the current processor context, then CWE-1421 may be more appropriate for the mapping task.

Source: CWE-1422

Other Content in these CWEs

- Modes of introduction – At what point in hardware design, system configuration, or software development (etc.) can these weaknesses potentially be introduced?
- Potential mitigations – How can transient execution weaknesses be mitigated during hardware design and, for those that can't, how can they be mitigated by software techniques? The CWEs provide more than a dozen best-known methods, many of which are currently being used in the hardware industry and by software vendors.
- Detection methods – How can transient execution weaknesses be detected in hardware designs, in post-silicon hardware samples, or in software programs?
- Demonstrative examples – These brief expository examples are derived from real CVEs.
- Additional general information about each weakness and how certain hardware behaviors can contribute to the introduction of vulnerabilities.