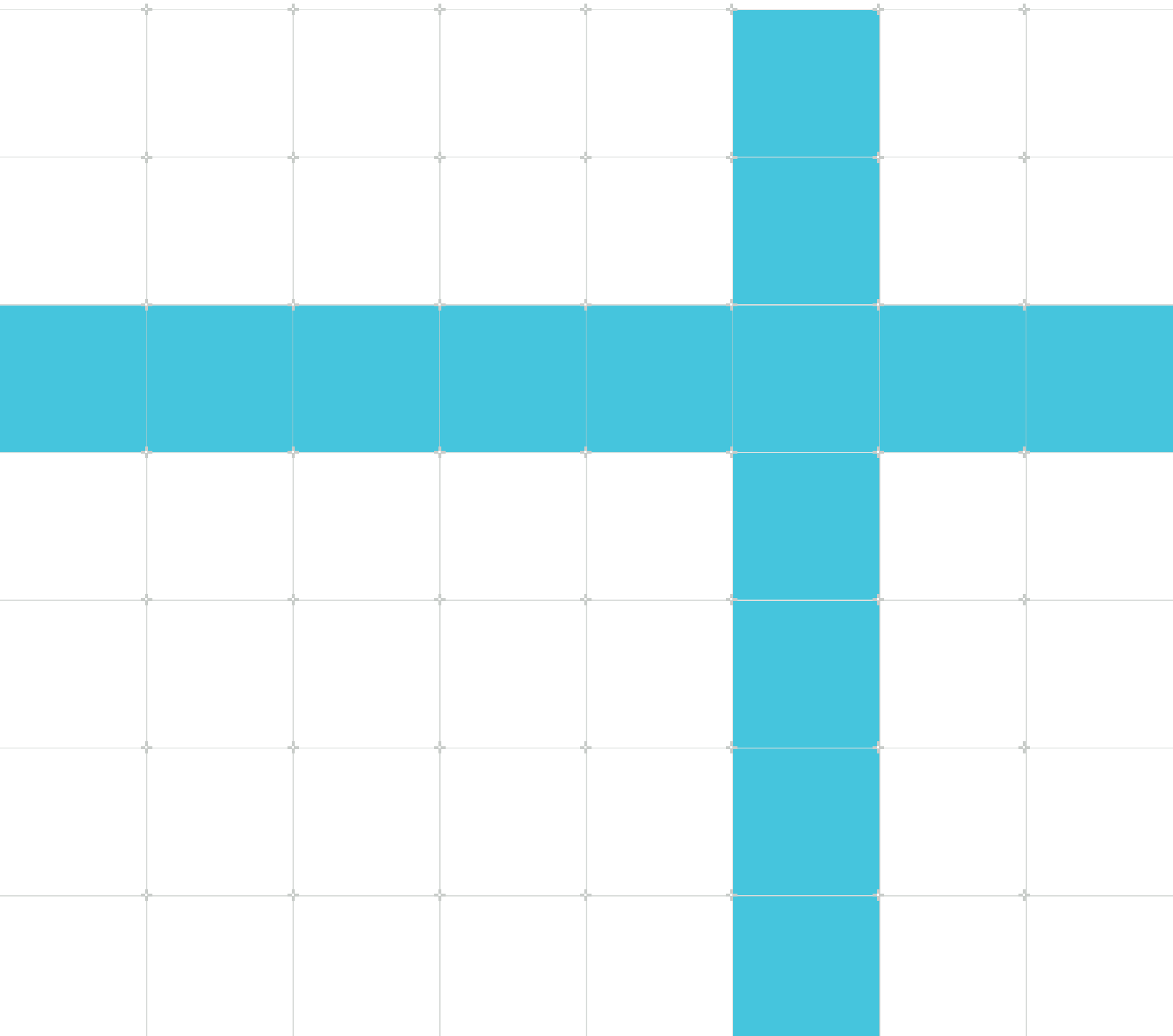
Speculative Processor Vulnerability

Revision: r1p3

Frequently Asked Questions

Non-Confidential

Issue
102587_0103_en



Speculative Processor Vulnerability

Frequently Asked Questions

Release information

Document history

Issue	Date	Confidentiality	Change
0100	24 February 2021	Non-Confidential	Initial release
0101	8 March 2022	Non-Confidential	Update for Spectre-BHB
0102	15 June 2022	Non-Confidential	Update for PMT side-channel and MMIO Stale Data
0103	9 August 2023	Non-Confidential	Update for Transient Data Gathering

Proprietary Notice

Arm Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

(LES-PRE-20349|version 21.0)

Confidentiality Status

This document is Non-Confidential. The right to use, copy and disclose this document may be subject to license restrictions in accordance with the terms of the agreement entered into by Arm and the party that Arm delivered this document to.

Unrestricted Access is an Arm internal classification.

Product Status

The information in this document is Final, that is for a developed product.

Feedback

Arm welcomes feedback on this product and its documentation. To provide feedback on the product, create a ticket on https://support.developer.arm.com

To provide feedback on the document, fill the following survey: https://developer.arm.com/documentation-feedback-survey.

Inclusive language commitment

Arm values inclusive communities. Arm recognizes that we and our industry have used language that can be offensive. Arm strives to lead the industry and create change.

We believe that this document contains no offensive language. To report offensive language in this document, email terms@arm.com.

Contents

1. Overview

The following information provides answers to some frequently asked questions about Speculative Processor Vulnerability.

2. General frequently asked questions

The following information provides answers to general questions about Speculative Processor Vulnerability.

Can you explain the problem in layman's terms?

The problem is as follows:

- Both attacks exploit existing side-channel techniques and could result in small bits of data being accessed through use of malware.
- Malware using this method and running locally could potentially allow malicious actors to improperly gather small bits of sensitive data from privileged memory (DRAM or CPU cache).
- Arm believes these exploits do not have the potential to corrupt, modify or delete data.
- All variants are based on speculative memory access causing cache allocation. Timing analysis of memory accesses can then be used to reveal data that would otherwise be kept secret. Each mechanism is slightly different, and some are microarchitecture dependent. The impact of each variant on Arm cores, plus the solutions and how they should be applied, are described in the Software Overview for Arm Cores Paper.

What is a cache side-channel attack?

When using the time taken to execute instructions that access the cache (such as loads or stores) it's possible to infer what items are in the cache. By being able to infer such information, the values of addresses that have been used to allocate items into the cache can be deduced.

What does this mean for the average mobile user?

Malicious actors would need to install malware on devices to execute any of these attacks. Users will greatly reduce their risk by following good security practices by avoiding suspicious links and downloads, and immediately installing any software updates when available from device makers.

Can you describe the variants of Spectre and Meltdown?

The variants of Spectre and Meltdown are as follows:

- Variant 1 (Spectre): Bounds Check Bypass - Use existing code with access to secrets by making it speculatively execute memory operations with out-of-range arguments.
- Variant 2 (Spectre): Branch Target Injection - Malicious code usurps properties of CPU branch prediction features to speculatively run code.
- Variant 3 (Meltdown): Rogue Data Load - Access memory controlled by the OS while running a malicious application.
- Variant 3a (Meltdown): A different form of Variant 3 identified by Arm. Variant 3a uses speculation to access a privileged system register.
- Variant 4 (Spectre): An attack using a CPU technology known as memory disambiguation.

How is TrustZone affected?

TrustZone is affected as follows:

- Arm Trusted Firmware has been updated to mitigate the known Spectre variants (1, 2 and 4) which could potentially create different levels of privilege.
- For Meltdown, this variant is only exploitable between Exception Levels within the same translation regime, for example between EL0 and EL1, therefore this variant cannot be used to access secure memory from the non-secure world and is not applicable for Arm Trusted Firmware.
- The link to the updated Trusted Firmware can be found at Arm Security Updates.

3. Affected processors frequently asked questions

The following information provides answers to questions about which processors are affected by Speculative Processor Vulnerability.

Are the Cortex-A65AE or Neoverse E1 vulnerable to SMT-based attacks such as PortSmash or TLBleed?

As multi-threaded processors, the Cortex-A65AE and Neoverse-E1 are vulnerable to attacks that are based on using one hardware thread to expose data in another. Many use cases for the Cortex-A65AE and Neoverse E1 employ a closed software stack, and so the thread model of their system does not include this style of attack. For systems which do include this thread model a number of mitigations are available, such as gang-scheduling to limit adversary access to parallel threads and disabling multi-threading for sensitive workloads.

Which Arm processors are affected by Spectre and Meltdown variants?

A comprehensive and updated list of all Arm processors impacted by all known variants can be found at Speculative Processor Vulnerability.

Will future Arm cores or architectures address this?

Future Arm cores and architectures will provide a range of features to address the issues that can be addressed within hardware and will provide more optimized mechanisms to allow software to address the issues that need to be addressed in software.

My Arm processor operates with branch prediction and speculative execution, why is my processor not on the list of affected processors?

Power efficient processor implementations tend to have shorter pipelines with fewer backend stages, and may also lack more complex micro-architectural features such as out-of-order execution. While these processors do still use speculation, they tend to be less aggressively optimized which makes these attacks unfeasible.

Arm have looked at all processors in all families to assess their vulnerability and compiled the table of processors with their perceived vulnerability that can be seen on the Speculative Processor Vulnerability overview page. Any processor not on the table has been analyzed and deemed not susceptible.

Can you comment on the susceptibility of Arm architecture partner implementations?

Arm cannot comment on architecture partner implementations. Please contact those partners directly with any questions.

How have affected Arm CPUs been updated to address these threats?

For details on mitigations added to Armv8.5-A, view the Armv8.5-A CPU Updates document.

4. Mitigations frequently asked questions

The following information provides answers to questions about mitigations for Speculative Processor Vulnerability.

Is there a fix for all the vulnerabilities?

Each variant will have a different set of mitigations. For all known variants impacting Arm cores, Arm has completed initial kernel patching, compiler work and firmware updates. We continue to monitor and work on mitigations. Detailed information on mitigations for all known variants can be found at Arm Security Updates.

Are software mitigations available, and will I get them?

The major operating systems running on Arm are aware of these issues and are deploying software mitigations. Please contact the suppliers of that software for their plans in this regard.

Arm trusted firmware patches are being made available through GitHub at Arm Trusted Firmware Security Advisory TFV 6 and Trusted Firmware A Security Advisory TFV 7.

Specifications of the changes required in Firmware are available in the Firmware interfaces for mitigating cache speculation vulnerabilities PDF.

How can mobile or PC users know whether the mitigation measure has been applied to them or not?

We encourage end-users to frequently check the security update sites for their respective operator or OEM or inquire directly as to when security updates will be available for Spectre and Meltdown.

What is Arm doing to ensure existing Android devices are updated with mitigations?

Arm has pushed out the necessary mitigations for all known Meltdown and Spectre variants to our ecosystem. In some cases, updates have been pushed to existing telephones that address the original vulnerabilities published in January 2018.

Questions about specific devices or timing for deploying security updates should be directed to OEMs.

Are all the fixes applicable to Arm architecture partner designs?

The vulnerabilities are dependent on processor micro-architecture and Arm cannot comment on how architecture partner designs are impacted. We advise you to contact Arm architecture partners directly with questions related to mitigations for their affected designs.

5. Spectre Variant 1 frequently asked questions

The following information provides answers to questions about Spectre Variant 1.

Will the Cortex-A cores impacted by Spectre variant 1 be updated for resiliency against the attack?

Spectre variant 1 is an issue that a usually safe hardware mechanism becomes exploitable in a very small number of cases that need to be identified by the software. As such, it's not possible for hardware to distinguish between apparently identical code sequences which are safe and those which are exploitable, and so it not practical to implement the generic fixes in the hardware without incurring significant performance degradation on safe code sequences. Therefore we regard it as being necessary for the software to be modified to avoid the issue.

6. Spectre Variant 2 frequently asked questions

The following information provides answers to questions about Spectre Variant 2.

Is Google's Retpoline considered effective for mitigating variant 2 vulnerability on Arm based systems?

Arm has investigated the use of retpoline and has concluded that it doesn't provide effective mitigation on Arm-based systems. Retpoline relies on specific aspects of the design of the branch prediction logic in the CPU, which do not apply to Arm-based systems.

Variant 2 mitigations for Arm systems have been implemented in the Linux kernel and Arm Trusted Firmware and can be implemented in other operating systems. For details see Arm Security Updates.

See our FAQ Are software mitigations available, and will I get them? for more information.

Will compilers provide mitigations for variant 2?

Arm is not currently aware of any effective compiler-based mitigation techniques (such as retpolines) for variant 2.

Variant 2 mitigations for Arm systems have been implemented in the Linux kernel and Arm Trusted Firmware, and can be implemented in other operating systems. For details see Arm Security Updates.

7. Meltdown Variant 3 frequently asked questions

The following information provides answers to questions about Meltdown Variant 3.

Is Arm impacted by Meltdown 3a (CVE-2018-3640)?

In January 2018, Arm provided details on the Cortex-A cores (A15, A57, and A72) impacted. Arm believes that software mitigations for Meltdown 3a are not required for the impacted Arm cores. However, an update for the latest release of the Cortex-A72 does include a hardware mitigation for this issue. For more details, please refer to our whitepaper at Arm Security Updates.

8. Spectre Variant 4 frequently asked questions

The following information provides answers to questions about Spectre Variant 4.

Please explain how Spectre Variant 4 differs from the other variants?

Spectre Variant 4 differs from the other variants in the following ways:

- Variant 4 is a Spectre-type attack utilizing a CPU technology known as memory disambiguation, a technology used in high-end CPUs to enable greater out-of-order execution and higher performance.
- Simply put, this is a race between a store and following load that target the same memory location whereby under specific conditions, a speculative load can overtake a store, resulting in the load returning stale data.
- If a chain of suitable instructions can be constructed, this stale data can be used to construct an address that drives cache allocation. This can be used to leak data to an adversary across a privilege boundary in a similar way to existing variants.

What Arm cores are impacted by Spectre Variant 4?

Four cores from the existing Cortex-A family are impacted: Cortex-A57, A72, A73 and A75. All cores impacted by all Spectre and Meltdown variants are listed at Arm Security Update.

Will the Cortex-A cores impacted by Spectre variant 4 be updated for resiliency against the attack?

The mitigation for existing cores impacted by Spectre variant 4 is a straightforward configuration setting at boot and, therefore, does not require hardware-level modifications. Arm has introduced new architectural features that make implementing software mitigations more straightforward and common across different CPUs.

Can Spectre variant 4 attacks be executed through a web browser?

Based on our testing, we believe Spectre variant 4 could target browser JavaScript engines and code on a malicious web page. Please contact browser solution providers with questions about their respective mitigations.

9. Spectre-BHB frequently asked questions

The following information provides answers to questions about Spectre-BHB.

Can you explain the problem in layman's terms?

The problem is as follows:

- This attack exploits existing side-channel techniques and could result in small bits of data being accessed through use of malware. Malware using this method and running locally could potentially allow malicious actors to improperly gather small bits of sensitive data from privileged memory (DRAM or CPU cache).
- Arm believes these exploits do not have the potential to corrupt, modify or delete data.
- Spectre-BHB is based on speculative memory access causing cache allocation. Timing analysis of memory accesses can then be used to reveal data that would otherwise be kept secret.

What is a cache side-channel attack?

When using the time taken to execute instructions that access the cache (such as loads or stores) it's possible to infer what items are in the cache. By being able to infer such information, the values of addresses that have been used to allocate items into the cache can be deduced.

What does this mean for the average user?

Malicious actors would need to install malware on devices to execute any of these attacks. Users will greatly reduce their risk by following good security practices by avoiding suspicious links and downloads, and immediately installing any software updates when available from device makers.

How does this differ from Spectre v2?

Unlike Spectre v2 by which a vulnerable CPU in one context could train the branch predictor to induce another context, Spectre-BHB forces and exploits the misprediction of the victim's own predictions. This is explained in more detail in the Spectre-BHB white paper.

Are Arm CPUs affected by the Linux kernel-only Spectre-BHB?

This issue exploits the fact that the adversary can execute code in the kernel's context by submitting unprivileged eBPF workloads. This allows the insertion of exfiltration primitives, timing measuring capabilities, and collision generation in the context of the kernel. The adversary essentially exploits Spectre-BHB directly in the kernel context. Arm CPUs affected by Spectre-BHB are also affected by this issue.

Given the broad range of attack vectors for eBPF, and the high-performance requirements, Arm strongly recommends that systems ensure that only eBPF code supplied by trusted parties is used. Please note that since eBPF is not supplied or developed by Arm, we cannot guarantee security for different instances of eBPF.

Is Arm affected by Unprivileged eBPF pointer arithmetic?

The risk of this issue is derived from the JIT compilation capabilities being able to generate exfiltration gadgets in software-managed-privilege execution environments. Arm has previously described mitigations against Variant 1.

Can you comment on the susceptibility of Arm architecture partner implementations?

Arm cannot comment on architecture partner implementations. Please contact those partners directly with any questions.

What about support for AArch32?

The Cortex-A57, Cortex-A72, and Cortex-A73 support the AArch32 execution state in EL1. In addition, the Cortex-A15 is an Armv7-A CPU, which is largely the same as AArch32. Mitigations for those CPUs will be the same regardless of whether the kernel is executing in AArch32 or AArch64 state. However, at this time, Arm does not have a schedule for updates to the AArch32 kernel to include those mitigations.

What is the mitigation status for other operating systems(OS)?

Please contact your OS vendor directly.

The white paper mentions a new Secure Monitor Call SMCCC_ARCH_WORKAROUND_3 - where is this documented?

Download the SMC Calling Convention documentation.

10. Straight-line speculation frequently asked questions

The following information provides answers to questions about straight-line speculation.

Do Arm CPUs speculatively execute past unconditional changes in control flow, as identified by the Google SafeSide project?

Following research from the Google SafeSide project, Arm is systematically documenting the possibilities for a processor to speculatively execute the instructions immediately following what should be a change in control flow, including:

- Exception generating instructions (`SVC`, `HVC`, `SMC`, `UNDEF`, `BRK`)
- Exception returns (`ERET`)
- Unconditional direct branches (`B`, `BL`)
- Unconditional indirect branches (`BR`, `BLR`)
- Function returns (`RET`)

Arm refers to this as "straight-line speculation" past and unconditional change in control flow, and has allocated CVE-2020-13844 accordingly. Arm has also published a new whitepaper on the topic here.

What does this mean in layman's terms? What is the impact?

This can be thought of as another form/variant of Spectre, several others of which have also been identified since the original publication in January 2018 as part of ongoing efforts by the industry to protect consumers and devices.

Note that at present we deem the security risk to be low as this would be difficult to exploit in practice, and a practical exploit has yet to be demonstrated. However, the possibility cannot be dismissed which is why Arm is acting now.

What does this mean for partners, and what action needs to be taken?

Where threat modelling shows that this vulnerability needs to be mitigated in a particular project, that project will need to be recompiled using tools that are aware of and can mitigate against the vulnerability.

Some manual intervention may also be required for projects that:

- Handle architectural exceptions (`ERET`)
- Make calls into more privileged software (`SVC/HVC/SMC`)
- Contain hand-written assembly sequences with unconditional changes in control flow


Note

At present we deem the security risk to be low as this would be difficult to exploit in practice, and a practical exploit has yet to be demonstrated. However, the possibility cannot be dismissed.

What patches and tools have been developed to mitigate this vulnerability and when will they be available?

Arm has developed patches for open source tooling here:

- gcc
- llvm

Patches mitigating against speculation past `ERET` developed by the open source community and merged into several projects in late 2019 and early 2020, including Trusted Firmware-A 1, 2, 3, OP-TEE, FreeBSD, and OpenBSD.

Similar mitigations for ERET were already present in the `/arch/arm64/` Linux kernel tree as part of belt-and-braces hardening measures that were developed as part of the original Spectre/Meltdown mitigations.

Further mitigations for other instructions involving a change in control flow are under investigation.

How will these mitigations impact performance?

In most cases we expect no direct impact on performance save for a reduction in code density. That said, secondary effects may include marginally increased pressures on the instruction caches and branch predictors due to the insertion of speculation barrier sequences and branch instructions.

11. Memory-Mapped IO (MMIO) Stale Data Issue

The following information provides answers to frequently asked questions about the “Memory-Mapped IO (MMIO) Stale Data Issue” published by Intel on June 14th.

Are Arm processors vulnerable to the recently disclosed issue “Memory-Mapped IO (MMIO) Stale Data Issue” published by Intel on June 14th?

It has been found that Cortex-A15 is susceptible to a variant of the Memory-Mapped IO (MMIO) Stale Data Issue described below. The Cortex-A15 may allow the leakage of cache data into MMIO (Device or Non-cacheable) writes. Write data within the bus width granularity that is not being written (that is, byte enables not asserted), will come from the output of the L1 data cache at the time this data is sampled, which happens when the request is sent to the L2 cache.

Can you summarize the issue?

The root cause of this issue is that unused bytes on a data bus (for example those not enabled or with clear strobes) are not zeroed out and therefore, under very specific implementation defined conditions, may contain stale data values.

The basic form of this mechanism is that some memory-mapped IO items might be ignoring byte strobes on writes.

For example, a byte-width write to an endpoint location that is expecting a word write, and then reading back a word from that same location might return a full word of data, where the data that was passed in the non-enabled bytes of the write is stale data from an earlier memory access.

The origin of the stale data is implementation defined, and it may or may not be easily controlled by an adversary.

Finally, it is important to remark that for this attack to be effective, it requires a path for the harvesting of stale data, which in a normal system, without cooperating devices it is generally thought to be hard to achieve deterministically.

Is my SoC vulnerable?

If the Cortex-A15 utilizes a closed software stack, then in these environments, applications or processes are strictly controlled, and therefore not exploitable.

The ability to propagate stale data around the system will depend on the memory subsystem components.

We would advise partners with open software stacks to analyze these components in their system to determine if these components further propagate unused byte lanes.

Finally, the system will need to include memory-mapped IO devices that ignore byte strobes on writes.

12. Other attacks frequently asked questions

The following information provides answers to frequently asked questions about Speculative Processor Vulnerability not covered by the other sections in this FAQ.

What is the Gather Data Sampling (GDS) attack mentioned in Downfall Attacks published on 9th August 2023?

GDS is an attack that abuses the optimistic matching and lack of context isolation in a shared temporal data buffer associated with wide operand instructions (e.g., SIMD, gather loads).

An adversary would trigger this behavior by transiently executing a gather load instruction while the victim thread is running other vulnerable wide operand instructions. The adversary would transiently gather data pertaining to the victim's domain and exfiltrate that via a cache side-channel.

Is Arm affected by the GDS attack?

No, Arm cores are not affected. Arm cores have the necessary isolation protections in hardware to prevent this behavior.

Are Arm CPUs affected by the Data-Instruction cache synchronization (SCSB) transient execution attack described in the recently published report "Rage Against the Machine Clear: A Systematic Analysis of Machine Clears and Their Implications for Transient Execution Attack" published in June 2021?

No, Arm CPUs are not affected by the SCSB attack described in this recent report. Arm has reviewed the report and our assessment is that no new mitigations are required. Arm is not aware of any implementations that require an SB instruction (in addition to the synchronization sequence required between writing new instructions and execution of the instructions) to prevent transient execution attacks caused by the old code.

Are Arm CPUs affected by the Floating Point Value Injection (FPVI) transient execution attack described in the recently published report "Rage Against the Machine Clear: A Systematic Analysis of Machine Clears and Their Implications for Transient Execution Attack" published in June 2021?

Arm is not aware of any implementations that would be vulnerable to the attack presented in this paper. We have become aware that some Arm implementations will speculatively perform calculations assuming that denormalised inputs are in fact not denormalised, and so can create transient results that can be used to form data value injection. However, Arm is not aware of any implementations that can form transient NaN values as a result of this, and thus are not vulnerable to these attacks.

Are Arm CPUs affected by the micro-op cache side channel attack described in the recently published paper "I See Dead muops: Leaking Secrets via Intel and AMD Micro-Op Caches" published in April 2021?

Arm has reviewed the paper and our assessment is that no new mitigations are required. This new covert channel would only apply to Arm CPUs with a micro-op cache (Cortex-A77, Cortex-A78, Cortex-A78-AE, Cortex-X1, Neoverse-V1 and Neoverse-N2) and the current mitigations described in the Arm Cache Speculation Side Channels white-paper already protect against the exploitation of side channels such as the micro-op cache, in a transient execution attack. Furthermore, to protect against this or other types of side channel attacks, programmers are encouraged to follow secure coding best practices and avoid secret-dependent data and control flow operations.

Are Arm CPUs affected by the Micro-architecture Data Sampling (MDS) approaches, variously described as RIDL (Rogue In-flight Data Load), Fallout and Zombieload, proposed by academic researchers and announced in May 2019?

After reviewing the paper and working with architecture licensees we are not aware of any Arm-based implementations which are affected by Micro-architecture Data Sampling (MDS) approaches. We thank Vrije Universiteit Amsterdam and TUGraz for their research and interaction with Arm.

Are Arm cores affected by the debug based attack known as Nailgun described in the paper from COMPASS Lab at Wayne State University, and if so is there a security problem with Arm debug logic?

Arm acknowledges the work of the team at COMPASS on the Understanding the Security of ARM Debugging Features paper that investigated this. We are confident that Arm debug technology is robust, and refer silicon vendors and software developers to existing recommendations in the Arm Architecture and CoreSight Architecture specifications. For more information, please see Arm Debug and Trace Configuration and Usage Models.

In the paper "Understanding the Security of ARM Debugging Features" from COMPASS at Wayne State University, the research names the Arm Juno r1 Board and implies that this implementation is not secure. Can Arm comment on this?

The Juno Arm Development Platform is an open, vendor-neutral Armv8-A development platform specifically designed to enable the development of secure software, and is not intended to be used in production deployments. More information on the Juno board can be found at Juno Development Board.

Are Arm CPUs affected by the SPOILER attack identified by academic researchers and announced in February 2019?

This attack is described in this paper. The researchers of Worcester Polytechnic Institute in Worcester, MA and the University of Lübeck in Germany have proposed a technique, SPOILER, to reveal some information about the virtual address mapping to physical memory addresses of a system, where software would normally be unaware of this information. The information revealed is not actually secret data, but can be used to facilitate attacks like Rowhammer or existing side channel methods like Prime+Probe.

The SPOILER approach can be used on certain Arm cores (Cortex-A57, Cortex-A72, and some previous revisions of Cortex-A76, and Neoverse N1) to reveal this address mapping information.

No other Arm CPUs are affected by the SPOILER attack. The commonly used mitigations within DRAM modules against Rowhammer remain effective even with this revelation of information. On the affected CPUs, the SPOILER method uses a form of Spectre variant 4 to gain access to that physical information, and existing variant 4 mitigations will prevent use of the SPOILER approach. More information on existing Spectre and Meltdown mitigations can be found in the Cache Speculation Side Channels whitepaper.

Are any Arm cores affected by CVE-2018-3665?

No Arm Cortex cores are affected by the FPU state information attack described in CVE-2018-3665.

Are any Arm cores affected by Foreshadow or Foreshadow-NG (L1 Terminal Fault)? (CVE-2018-3615, CVE-2018-3620, CVE-2018-3646)

No Arm Cortex cores are affected by the speculative data leaks known as Foreshadow and Foreshadow-NG.

Are Arm CPUs affected by the seven Spectre and Meltdown variants identified by academic researchers and announced in November 2018?

These varianets are described in this paper. Arm CPUs affected by Spectre variant 1 and variant 2 are also affected by their derivatives described in the paper, however the existing mitigation techniques, including the use of DSB SY + ISB when deployed as directed by the Cache Speculation Side-channels Whitepaper are effective against these additional vulnerabilities.

We are thankful to the teams at Graz University of Technology, imec-DistriNet, KU Leuven, and the College of William and Mary for their ongoing research and interaction with Arm.

Are Arm CPUs affected by the SplitSpectre variant, identified by academic researchers and announced in December 2018?

Arm does not speculatively take exception returns, and so SplitSpectre cannot be exhibited across different exception levels on Arm. Therefore current Arm CPUs are not vulnerable to SplitSpectre attacks where the malicious software and victim software are running at separate levels of hardware enforced privilege.

However, where privilege is only enforced by software, SplitSpectre is effectively another presentation of Spectre variant 1 and attacks are possible on current Arm CPUs. In these cases Arm continues to recommend techniques described previously in the Cache Speculation Side-channels Whitepaper, such as site isolation, and/or use of DSB;ISB sequences when returning to a less privileged context.

Is Arm aware of the security reports recorded on the Common Vulnerabilities and Exposures database which reference Trustzone, and what is Arm's response to these?

This database can be found at Common Vulnerabilities and Exposures database. Arm monitors the entries in the CVE database. The reports which reference TrustZone are largely attributable to third-party software issues occurring where TrustZone security extensions have been deployed, hence these are not vulnerabilities in the TrustZone architecture but issues observed in development of any software regardless of whether it's secure or non-secure. Arm investigates

new vulnerability reports and continuously strives to improve the TrustZone architecture for the advancement of secure world requirements.

Is Arm aware of the Privileged Access Never (PAN) mitigation bypass reported in January 2020?

Privileged Access Never (PAN) is a hardening feature that is intended to protect against other operating system vulnerabilities where the kernel unintentionally reads user-controlled memory. When PAN is implemented, this type of kernel read will generate a fault. The report describes a scenario which allows user-space to bypass PAN, meaning that a fault will not be generated when it should be.

The vulnerability that is described in the report requires the presence of an existing bug that would most likely be caught by normal PAN functionality. This means that the vulnerability cannot be directly exploited. Instead, it facilitates the exploitation of a more serious operating system vulnerability involving the kernel in these circumstances.

PAN functionality in Arm cores is designed to identify this type of operating system vulnerability and mitigate against it, significantly reducing the likelihood of any such vulnerability being exploited in practice. To ensure continued PAN protection, Arm released a Linux kernel patch to disable execute-only permissions for user applications on January 6, 2020.

Are Arm CPUs affected by any of the Load Value Injection (LVI) attacks announced by researchers in March 2020?

LVI attacks are described in this paper. Various forms of LVI have been identified corresponding to the equivalent “forward” mechanisms, such Meltdown, RDS, and speculative use of floating-point data (variant 3a), and could allow the adversary to inject selected data into the victim’s speculative execution. In principle this could include user space attempting to attack a kernel. However Cortex-A72 and Neoverse N1 cores do not speculatively use any data that is returned with an associated synchronous exception, and so are not susceptible to LVI attacks where the adversary can select the data to be used speculatively by the load.

One form of LVI, LVI-NULL describes a scenario in which a core returns a fixed value (for example, 0) with a Translation, AccessFlag or Permission Fault, and this is used speculatively for subsequent loads/stores which could be used as the basis of an attack. Cortex-A72 does not speculatively use any result from a Translation, AccessFlag or Permission Fault so this mechanism does not apply to Cortex- A72. Neoverse N1 will use the fixed value of 0 in this situation, so this situation cannot be ruled out however Arm does not believe it is practically exploitable on Arm systems. Any further updates will be published on the speculative processor vulnerability pages of the Arm Developer website. We thank the researchers for their interaction with Arm.

Are Arm CPUs affected by the Power-Management Throttling side-channel identified by Intel and others and disclosed on 14th June 2022?

Arm has reviewed the PMT side-channel and concluded that whilst this opens a new attack surface, the difficulty of side-channel analysis and exploitation reduces its likelihood and potential impact, hence Arm considers it to be low-risk.

While there does not exist a complete mitigation, countermeasures and recommendations for cryptographic code against power side-channels can be used against this attack. Arm recognizes that by adding a level of indirection and translating from the power domain to the time domain, the new PMT side-channel breaks a fundamental previous assumption about power analysis attacks: physical access. This might require re-evaluating the threat model of some security critical systems. Arm thanks Intel for their research and their interaction with Arm.