CISPA
HELMHOLTZ CENTER FOR INFORMATION SECURITY

How Secure Are Commercial RISC-V CPUs?

Lukas Gerlach, Fabian Thomas | PhD Students @ CISPA (Germany)

Meltdown

Meltdown breaks the most fundamental isolation between user applications and the operating system. This attack allows a program to access the memory, and thus also the secrets, of other programs and the operating system.

If your computer has a vulnerable processor and runs an unpatched operating system, it is not safe to work with sensitive information without the chance of leaking the information. This applies both to personal computers as well as cloud infrastructure. Luckily, there are software patches against Meltdown.

Spectre

Spectre breaks the isolation between different applications. It allows an attacker to trick error-free programs, which follow best practices, into leaking their secrets. In fact, the safety checks of said best practices actually increase the attack surface and may make applications more susceptible to Spectre

Spectre is harder to exploit than Meltdown, but it is also harder to mitigate. However, it is possible to prevent specific known exploits based on Spectre through software patches.

ZombieLoad Attack

Return of the Leaking Dead

Watch out! Your processor resurrects your private browsing-history and other sensitive data.

After Meltdown, Spectre, and Foreshadow, we discovered **more critical vulnerabilities** in modern processors. The ZombieLoad attack allows **stealing sensitive data and keys** while the computer recovers them.

Meltdown

Meltdown breaks the most fundamental isolation between user applications and the operating system. This attack allows a program to access the memory, and thus also the secrets, of other programs and the operating system.

If your computer has a vulnerable processor and runs an unpatched operating system, it is not safe to work with sensitive information without the chance of leaking the information. This applies both to personal computers as well as cloud infrastructure. Luckily, there are software patches against Meltdown.

Spectre

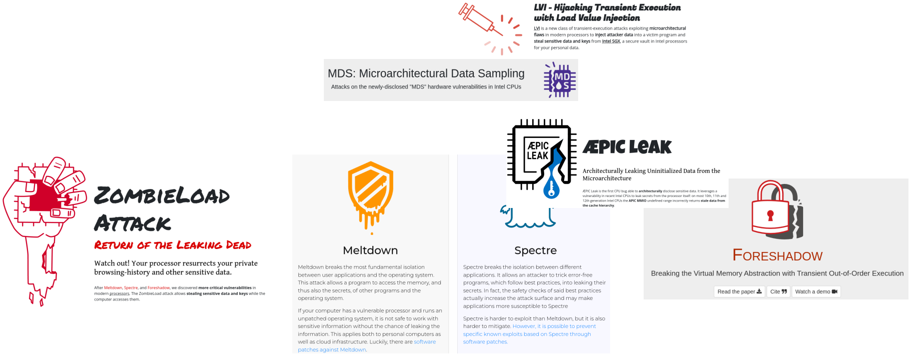
Spectre breaks the isolation between different applications. It allows an attacker to trick error-free programs, which follow best practices, into leaking their secrets. In fact, the safety checks of said best practices actually increase the attack surface and may make applications more susceptible to Spectre

Spectre is harder to exploit than Meltdown, but it is also harder to mitigate. However, it is possible to prevent specific known exploits based on Spectre through software patches.

Foreshadow

Breaking the Virtual Memory Abstraction with Transient Out-of-Order Execution

Read the paper | Cite | Watch a demo


LVI - Hijacking Transient Execution with Load Value Injection
LVI is a new class of transient-execution attacks exploiting microarchitectural flaws in modern processors to inject attacker data into a victim program and steal sensitive data and keys from Intel SGX, a secure vault in Intel processors for your personal data.
MDS: Microarchitectural Data Sampling
Attacks on the newly-disclosed "MDS" hardware vulnerabilities in Intel CPUs
MDS
ÆPIC LEAK
ÆPIC Leak
Architecturally Leaking Uninitialized Data from the Microarchitecture
ÆPIC Leak is the first CPU bug able to architecturally disclose sensitive data. It leverages a vulnerability in recent Intel CPUs to leak secrets from the processor itself: on most 10th, 11th and 12th generation Intel CPUs the APIC MMIO undefined range incorrectly returns stale data from the cache hierarchy.
ZombieLoad Attack
Return of the Leaking Dead
Watch out! Your processor resurrects your private browsing-history and other sensitive data.
After Meltdown, Spectre, and Foreshadow, we discovered more critical vulnerabilities in modern processors. The ZombieLoad attack allows stealing sensitive data and keys while the computer accesses them.
Meltdown
Meltdown breaks the most fundamental isolation between user applications and the operating system. This attack allows a program to access the memory, and thus also the secrets, of other programs and the operating system.
If your computer has a vulnerable processor and runs an unpatched operating system, it is not safe to work with sensitive information without the chance of leaking the information. This applies both to personal computers as well as cloud infrastructure. Luckily, there are software patches against Meltdown.
Spectre
Spectre breaks the isolation between different applications. It allows an attacker to trick error-free programs, which follow best practices, into leaking their secrets. In fact, the safety checks of said best practices actually increase the attack surface and may make applications more susceptible to Spectre
Spectre is harder to exploit than Meltdown, but it is also harder to mitigate. However, it is possible to prevent specific known exploits based on Spectre through software patches.
Foreshadow
Breaking the Virtual Memory Abstraction with Transient Out-of-Order Execution
Read the paper
Cite
Watch a demo

Cross-Process Information Leak

Bulletin ID: AMD-SB-7008
Potential Impact: Information disclosure
Severity: Medium

Summary

Under specific microarchitectural circumstances, a register in "Zen 2" CPUs may not be written to 0 correctly. This may cause data from another process and/or thread to be stored in the YMM register, which may allow an attacker to potentially access sensitive information.

LVI - Hijacking Transi with Load Value Inject

LVI is a new class of transient-execution attac flaws in modern processors to inject attacker steal sensitive data and keys from Intel SGX, a for your personal data.

MDS: Microarchitectural Data Sampling

Attacks on the newly-disclosed "MDS" hardware vulnerabilities in Intel CPUs

Downfall Attacks

Attack Demo FAQ Advisories Links

Downfall attacks target a critical weakness found in billions of modern processors used in personal and cloud computers. This vulnerability, identified as CVE-2022-40982, enables a user to access and steal data from other users who share the same computer. For instance, a malicious app obtained from an app store could use the Downfall attack to steal sensitive information like passwords, encryption keys, and private data such as banking details, personal emails, and messages. Similarly, in cloud computing environments, a malicious customer could exploit the Downfall vulnerability to steal data and credentials from other customers who share the same cloud computer.

ÆPIC

Architecturally
Microarchitectu

ÆPIC Leak is the first CPU bu vulnerability in recent Intel CPUs to leak secrets from the processor itself: on most 10th, 11th and 12th generation Intel CPUs the **APIC MMIO** undefined range incorrectly returns **stale data from the cache hierarchy**.

ZombieLoad Attack

Return of the Leaking Dead

Watch out! Your processor resurrects your private browsing-history and other sensitive data.

After Meltdown, Spectre, and Foreshadow, we discovered **more critical vulnerabilities** in modern processors. The ZombieLoad attack allows **stealing sensitive data and keys** while the computer accesses them.

Meltdown

Meltdown breaks the most fundamental isolation between user applications and the operating system. This attack allows a program to access the memory, and thus also the secrets, of other programs and the operating system.

If your computer has a vulnerable processor and runs an unpatched operating system, it is not safe to work with sensitive information without the chance of leaking the information. This applies both to personal computers as well as cloud infrastructure. Luckily, there are software patches against Meltdown.

Spectre

Spectre breaks the isolation between different applications. It allows an attacker to trick error-free programs, which follow best practices, into leaking their secrets. In fact, the safety checks of said best practices actually increase the attack surface and may make applications more susceptible to Spectre

Spectre is harder to exploit than Meltdown, but it is also harder to mitigate. However, it is possible to prevent specific known exploits based on Spectre through software patches.

Foreshadow

Breaking the Virtual Memory Abstraction with Transient Out-of-Order Execution

Read the paper | Cite | Watch a demo

Retbleed: Arbitrary Speculative Code Execution with Return Instructions

Retbleed (CVE-2022-29900 and CVE-2022-29901) is the new addition to the family of speculative execution attacks that exploit branch target injection to leak information, which we call Spectre-BTI. Unlike its siblings, who trigger harmful branch target speculation by exploiting indirect jumps or calls, Retbleed exploits return instructions. This means a great deal, since it undermines some of our current Spectre-BTI defenses.

Cross-Process Information Leak

Bulletin ID: AMD-SB-7008
Potential Impact: Information disclosure
Severity: Medium

Summary

Under specific microarchitectural circumstances, a register in "Zen 2" CPUs may not be written to 0 correctly. This may cause data from another process and/or thread to be stored in the YMM register, which may allow an attacker to potentially access sensitive information.

ZombieLoad Attack

Return of the Leaking Dead

Watch out! Your processor resurrects your private browsing-history and other sensitive data.

After Meltdown, Spectre, and Foreshadow, we discovered **more critical vulnerabilities** in modern processors. The ZombieLoad attack allows **stealing sensitive data and keys** while the computer revokes them.

LVI - Hijacking Transi
with Load Value Injec

LVI is a new class of transient execution attac
flaws in modern processors to **inject attacker-**
steal sensitive data and keys from Intel SGX, a
for your personal data.

MDS: Microarchitectural Data Sampling

Attacks on the newly disclosed "MDS" hardware vulnerabilities in Intel CPUs

Meltdown

Meltdown breaks the most fundamental isolation between user applications and the operating system. This attack allows a program to access the memory, and thus also the secrets, of other programs and the operating system.

If your computer has a vulnerable processor and runs an unpatched operating system, it is not safe to work with sensitive information without the chance of leaking the information. This applies both to personal computers as well as cloud infrastructure. Luckily, there are software patches against Meltdown.

Spectre

Spectre breaks the isolation between different applications. It allows an attacker to trick error-free programs, which follow best practices, into leaking their secrets. In fact, the safety checks of said best practices actually increase the attack surface and may make applications more susceptible to Spectre

Spectre is harder to exploit than Meltdown, but it is also harder to mitigate. However, it is possible to prevent specific known exploits based on Spectre through software patches.

SLAP

Data Speculation Attacks via Load Address Prediction on Apple Silicon

We present SLAP, a new speculative execution attack that arises from optimizing data dependences, as opposed to control-flow dependences. More specifically, we show that Apple CPUs (implementing the ARM ISA) are equipped with a Load Address Predictor (LAP), which improves performance by guessing the next memory address the CPU will load data from based on prior memory access patterns.

FLOP

Breaking the Apple M3 CPU via False Load Output Predictions

We present FLOP, another speculative execution attack that results from recent Apple CPUs predicting the outputs of data memory accesses. More specifically, Apple M3 CPUs (4th generation and newer CPUs) are equipped with a Load Value Predictor (LVP). The LVP improves performance on data dependencies by guessing the data value that will be returned by the memory subsystem on the next access to the CPU core, before the value is actually available.

Retbleed: Arbitrary Speculative Code Execution with Return Instructions

Retbleed (CVE-2022-29900 and CVE-2022-29901) is the new addition to the family of speculative execution attacks that exploit branch target injection to leak information, which we call Spectre-BTI. Unlike its siblings, who trigger harmful branch target speculation by exploiting indirect jumps or calls, Retbleed exploits return instructions. This means a great deal, since it undermines some of our current Spectre-BTI defenses.

Downfall Attacks

Attack Demo FAQ Advisories Links

Downfall attacks target a critical weakness found in billions of modern processors used in personal and cloud computers. This vulnerability, identified as CVE-2022-40982, enables a user to access and steal data from other users who share the same computer. For instance, a malicious app obtained from an app store could use the Downfall attack to steal sensitive information like passwords, encryption keys, and private data such as banking details, personal emails, and messages. Similarly, in cloud computing environments, a malicious customer could exploit the Downfall vulnerability to steal data and credentials from other customers who share the same cloud computer.

Architecturally Microarchitectu

AEPIC Leak is the first CPU vulnerability that leaks kernel memory from the processor itself, via recent Gen. 11th and 12th generation Intel CPUs (the **AFU MMIO** undefined range incorrectly returns **state data from the cache hierarchy.**

Foreshadow

Breaking the Virtual Memory Abstraction with Transient Out-of-Order Execution

Read the paper Cite Watch a demo

Cross-Process Information Leak

Bulletin ID: AMD-SB-7008
Potential Impact: Information disclosure
Severity: Medium

Summary

Under specific microarchitectural circumstances, a register in "Zen 2" CPUs may not be written to 0 correctly. This may cause data from another process and/or thread to be stored in the YMM register, which may allow an attacker to potentially access sensitive information.

LVI - Hijacking Transi with Load Value Injec

LVI is a new class of transient-execution attac **flaws** in modern processors to **inject attacker** **steal sensitive data and keys** from Intel SGX, a for your personal data.

MDS: Microarchitectural Data Sampling

Attacks on the newly-disclosed "MDS" hardware vulnerabilities in Intel CPUs

Downfall Attacks

Attack Demo FAQ Advisories Links

Downfall attacks target a critical weakness found in billions of modern processors used in personal and cloud computers. This vulnerability, identified as CVE-2022-40982, enables a user to access and steal data from other users who share the same computer. For instance, a malicious app obtained from an app store could use the Downfall attack to steal sensitive information like passwords, encryption keys, and private data such as banking details, personal emails, and messages. Similarly, in cloud computing environments, a malicious customer could exploit the Downfall vulnerability to steal data and credentials from other customers who share the same cloud computer.

ÆPIC

Architecturally Microarchitectu

ÆPIC Leak is the first CPU b vulnerability in recent Intel CPUs to leak secrets from the processor itself: on most 10th, 11th and 12th generation Intel CPUs the **APIC MMIO** undefined range incorrectly returns **stale data from the cache hierarchy**.

ZombieLoad Attack

Return of the Leaking Dead

Watch out! Your processor resurrects your private browsing-history and other sensitive data.

After Meltdown, Spectre, and Foreshadow, we discovered **more critical vulnerabilities** in modern processors. The ZombieLoad attack allows **stealing sensitive data and keys** while the computer accesses them.

Google researchers discover 'Reptar,' a new CPU vulnerability

November 15, 2023

Phil Venables
VP/CISO, Google Cloud

This year, Google has seen an increase in the number of vulnerabilities impacting central processing units (CPU) across hardware systems. Two of the most notable of these vulnerabilities were disclosed in August, when Google researchers discovered Downfall (CVE-2022-40982) and Zenbleed (CVE-2023-20593), affecting Intel and AMD CPUs, respectively.

Meltdown break between user ap This attack allow thus also the sec operating system

If your computer has a vulnerable processor and runs an unpatched operating system, it is not safe to work with sensitive information without the chance of leaking the information. This applies both to personal computers as well as cloud infrastructure. Luckily, there are software patches against Meltdown.

ectre

n between different tacker to trick error-free st practices, into leaking their hecks of said best practices

applications mor

Spectre is harder harder to mitigat specific known e software patches

Foreshadow

Breaking the Virtual Memory Abstraction with Transient Out-of-Order Execution

Read the paper | Cite | Watch a demo

CACHE WARP

CacheWarp is a new software fault attack on AMD SEV-ES and SEV-SNP. It allows attackers to hijack control flow, break into encrypted VMs, and perform privilege escalation inside the VM.

READ | CITE | TRY

SLAP

Data **S**peculation Attacks via **L**oad **A**ddress **P**rediction on Apple Silicon

We present SLAP, a new speculative execution attack that arises from optimizing data dependencies, as opposed to control flow dependencies. More specifically, we show that Apple CPUs starting with the M2/A15 are equipped with a Load Address Predictor (LAP), which improves performance by guessing the next memory address the CPU will retrieve data from based on prior memory access patterns.

FLOP

Breaking the Apple M3 CPU via **F**alse **L**oad **O**utput **P**redictions

We present FLOP, another speculative execution attack that results from recent Apple CPUs predicting the outcome of data dependencies. Here, we demonstrate that Apple's M3/A17 generation and newer CPUs are equipped with a Load Value Predictor (LVP). The LVP improves performance on data dependencies by guessing the data value that will be returned by the memory subsystem on the next access by the CPU core, before the value is actually available.

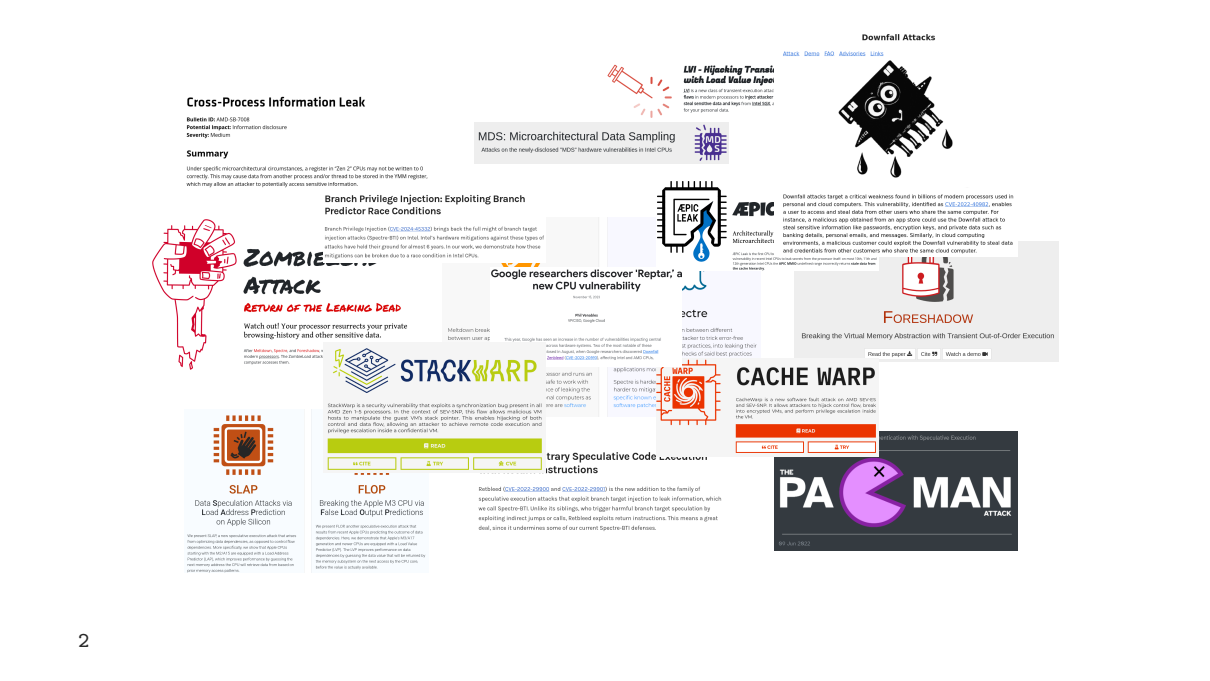
Retbleed: Arbitrary Speculative Code Execution with Return Instructions

Retbleed (CVE-2022-29900 and CVE-2022-29901) is the new addition to the family of speculative execution attacks that exploit branch target injection to leak information, which we call Spectre-BTI. Unlike its siblings, who trigger harmful branch target speculation by exploiting indirect jumps or calls, Retbleed exploits return instructions. This means a great deal, since it undermines some of our current Spectre-BTI defenses.

The PACMAN Attack

hentication with Speculative Execution

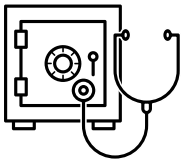
09 Jun 2022


Cross-Process Information Leak
Bulletin ID: AMD-SB-7008
Potential Impact: Information disclosure
Severity: Medium
Summary
Under specific microarchitectural circumstances, a register in "Zen 2" CPUs may not be written to 0 correctly. This may cause data from another process and/or thread to be stored in the YMM register, which may allow an attacker to potentially access sensitive information.
MDS: Microarchitectural Data Sampling
Attacks on the newly disclosed "MDS" hardware vulnerabilities in Intel CPUs
Branch Privilege Injection: Exploiting Branch Predictor Race Conditions
Branch Privilege Injection (CVE-2024-45332) brings back the full might of branch target injection attacks (Spectre-BTI) on Intel. Intel's hardware mitigations against these types of attacks have held their ground for almost 6 years. In our work, we demonstrate how these mitigations can be broken due to a race condition in Intel CPUs.
ZOMBIELOAD ATTACK
RETURN OF THE LEAKING DEAD
Watch out! Your processor resurrects your private browsing-history and other sensitive data.
Google researchers discover 'Reptar,' a new CPU vulnerability
Meltdown break
between user ap
Downfall Attacks
Attack Demo FAQ Advisories Links
Downfall attacks target a critical weakness found in billions of modern processors used in personal and cloud computers. This vulnerability, identified as CVE-2022-40982, enables a user to access and steal data from other users who share the same computer. For instance, a malicious app obtained from an app store could use the Downfall attack to steal sensitive information like passwords, encryption keys, and private data such as banking details, personal emails, and messages. Similarly, in cloud computing environments, a malicious customer could exploit the Downfall vulnerability to steal data and credentials from other customers who share the same cloud computer.
FORESHADOW
Breaking the Virtual Memory Abstraction with Transient Out-of-Order Execution
Read the paper
Cite
Watch a demo
STACKWARP
StackWarp is a security vulnerability that exploits a synchronization bug present in all AMD Zen 1-5 processors. In the context of SEV-SNP, this flaw allows malicious VM hosts to manipulate the guest VM's stack pointer. This enables hijacking of both control and data flow, allowing an attacker to achieve remote code execution and privilege escalation inside a confidential VM.
READ
CITE
TRY
CVE
CACHE WARP
CacheWarp is a new software fault attack on AMD SEV-ES and SEV-SNP. It allows attackers to hijack control flow, break into encrypted VMs, and perform privilege escalation inside the VM.
READ
CITE
TRY
SLAP
Data Speculation Attacks via Load Address Prediction on Apple Silicon
We present SLAP, a new speculative execution attack that arises from optimizing data dependences, as opposed to control-flow dependences. More specifically, we show that Apple CPUs implement a data memory-dependent prefetcher with a Load Address Predictor (LAP), which improves performance by querying the next memory address the CPU will need data from based on prior memory access patterns.
FLOP
Breaking the Apple M3 CPU via False Load Output Predictions
We present FLOP, another speculative execution attack that results from recent Apple CPUs predicting the outputs of data memory-dependent loads. Specifically, Apple M3 CPU generation and newer CPUs are equipped with a Load Value Predictor (LVP). The LVP improves performance on data dependencies by guessing the data value that will be returned by the memory subsystem on the next access to the CPU core, before the value is actually available.
trary Speculative Code Execution
ıstructions
Retbleed (CVE-2022-29900 and CVE-2022-29901) is the new addition to the family of speculative execution attacks that exploit branch target injection to leak information, which we call Spectre-BTI. Unlike its siblings, who trigger harmful branch target speculation by exploiting indirect jumps or calls, Retbleed exploits return instructions. This means a great deal, since it undermines some of our current Spectre-BTI defenses.
THE PACMAN ATTACK
tification with Speculative Execution
9th Jun 2822

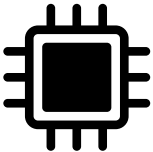
RISC-V: A Clean Slate?

Can we design a secure architecture from the start or will we repeat the same mistakes?

CPU Vulnerability Classes



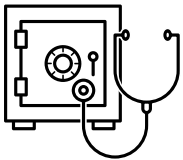
Side Channels



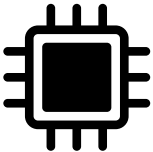
CPU Bugs

Transient Execution

CPU Vulnerability Classes



Side Channels



CPU Bugs

Transient Execution

Side Channels: RSA Square-and-Multiply

```
func exp_mod(C, d, n): /* M=C^d mod n */
    x = C
    for bit in d: /* MSB-first */
        x = x * x /* Square */
        if bit == 1:
            x = x * C /* Multiply */
    return x % n
```

Side Channels: RSA Square-and-Multiply

- d is secret

```
func exp_mod(C, d, n): /* M=C^d mod n */
    x = C
    for bit in d: /* MSB-first */
        x = x * x /* Square */
        if bit == 1:
            x = x * C /* Multiply */
    return x % n
```

Side Channels: RSA Square-and-Multiply

- d is secret
- Multiplication $\iff$ bit=1

```
func exp_mod(C, d, n): /* M=C^d mod n */
    x = C
    for bit in d: /* MSB-first */
        x = x * x /* Square */
        if bit == 1:
            x = x * C /* Multiply */
    return x % n
```

Side Channels: RSA Square-and-Multiply

- d is secret
- Multiplication $\iff$ bit=1
- How to leak?

```
func exp_mod(C, d, n): /* M=C^d mod n */
    x = C
    for bit in d: /* MSB-first */
        x = x * x /* Square */
        if bit == 1:
            x = x * C /* Multiply */
    return x % n
```

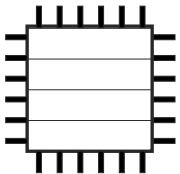
Flush+Reload

Attacker

flush

access

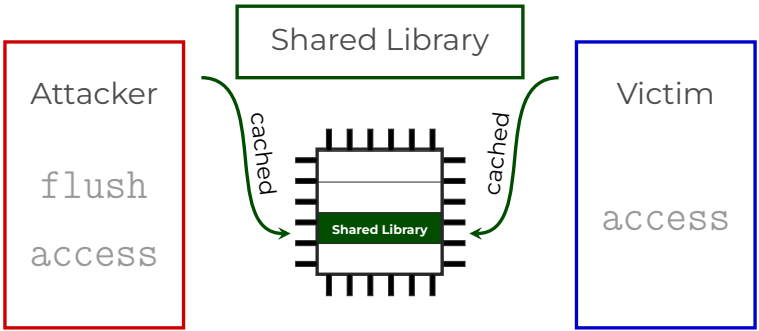
Shared Library



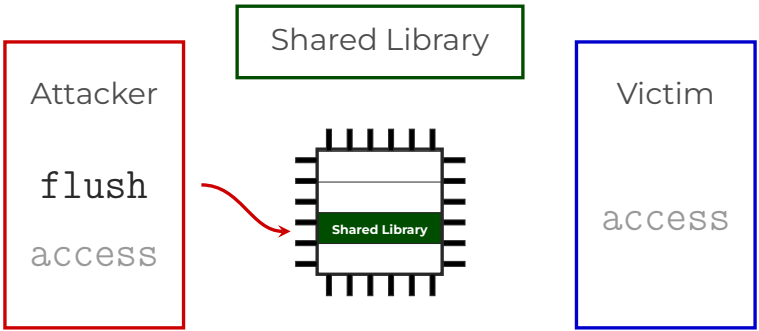
Victim

access

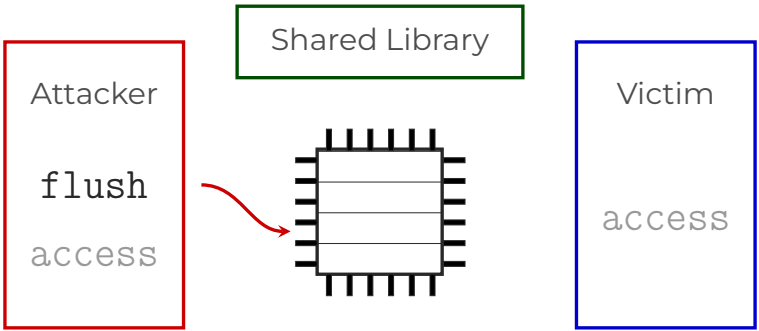
Flush+Reload



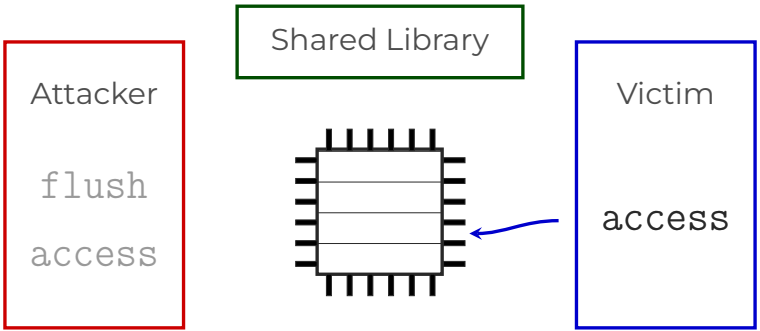
Flush+Reload



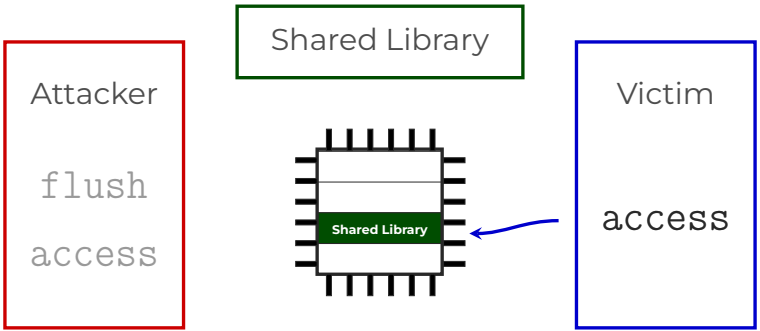
Flush+Reload



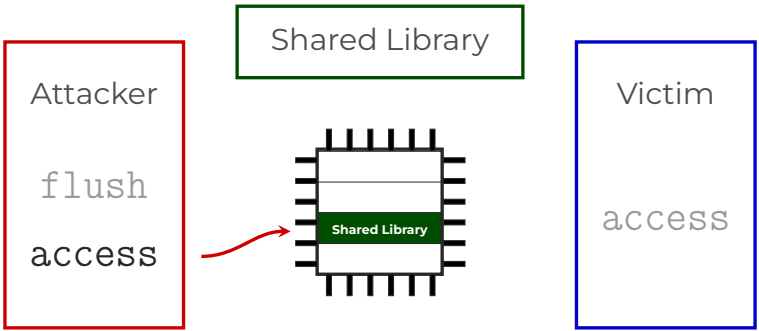
Flush+Reload



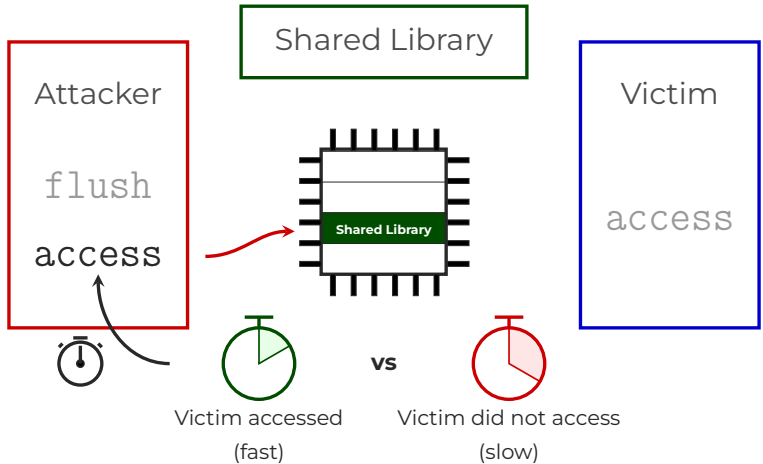
Flush+Reload



Flush+Reload



Flush+Reload



Accurate Timers?

Architectural timers

`rdcycle`	hardware cycles
`rdtime`	platform time
`rdinstret`	retired instructions

High resolution, available in user space on many cores

Cache Maintenance?

- No cache maintenance in base ISA

→ Only fences

Cache Maintenance?

- No cache maintenance in base ISA

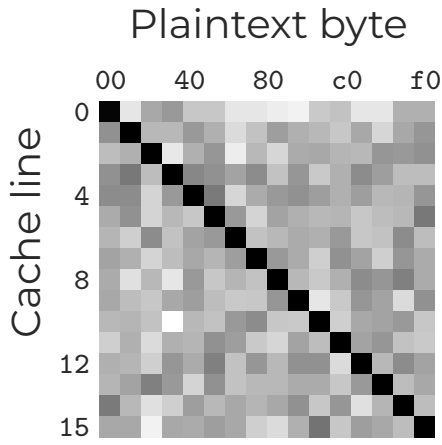
→ Only fences

Vendor-specific cache operations

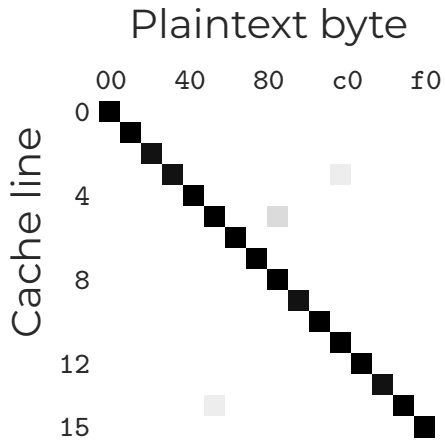
T-Head Cores:

- Unprivileged D-Cache flush by virtual address (like `clflush`)
- `fence.i` flushes entire I-Cache

AES T-Table (x86)



AES T-Table (RISC-V, C906)

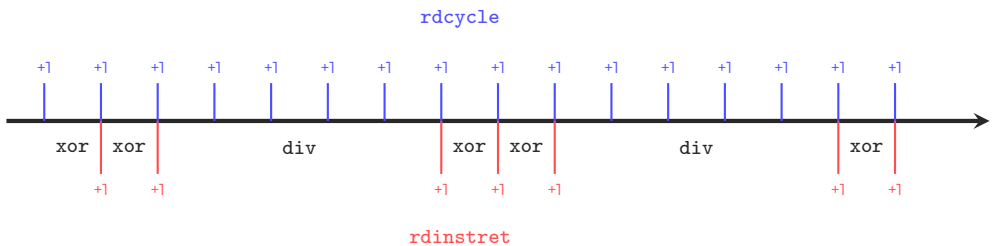


`rdcycle` **vs** `rdinstret`

CPU cycles and retired instructions can differ

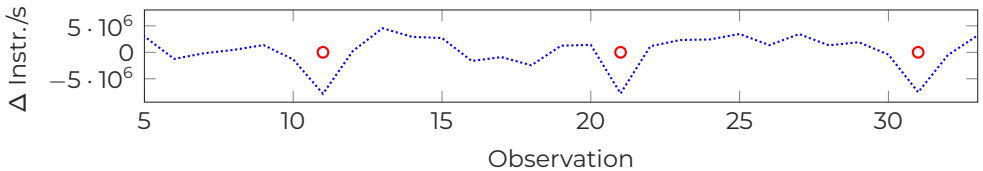
rdcycle **vs** rdinstret

CPU cycles and retired instructions can differ



CycleDrift

Drift between `rdcycle` and `rdinstret` leaks information about executed instructions (including from the kernel)



Takeaways

Accurate Timers

- ISA exposes `rdcycle`, `rdinstret` to user space
- Enables CycleDrift, precise cache timing
- ✔ Linux now disables user-space access on RISC-V

Takeaways

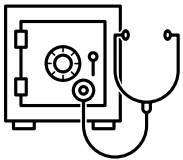
Accurate Timers

- ISA exposes `rdcycle`, `rdinstret` to user space
- Enables CycleDrift, precise cache timing
- ✔ Linux now disables user-space access on RISC-V

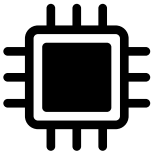
Cache Maintenance

- `fence.i` is unprivileged and flushes the entire I-Cache
- T-Head adds unprivileged D-Cache flush by virtual address (like `clflush`)
- Enables Flush+Reload on I-Cache and D-Cache.

CPU Vulnerability Classes



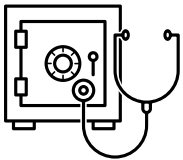
Side Channels



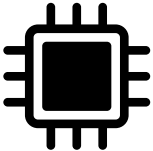
CPU Bugs

Transient Execution

CPU Vulnerability Classes



Side Channels

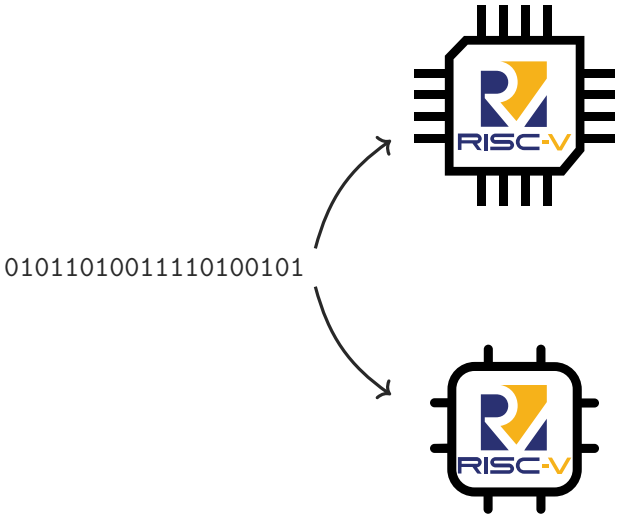


CPU Bugs

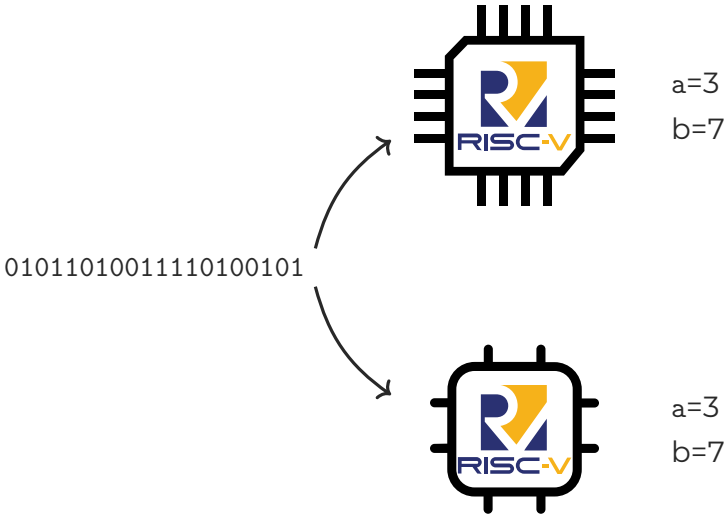
Transient Execution

Differential CPU Fuzzing

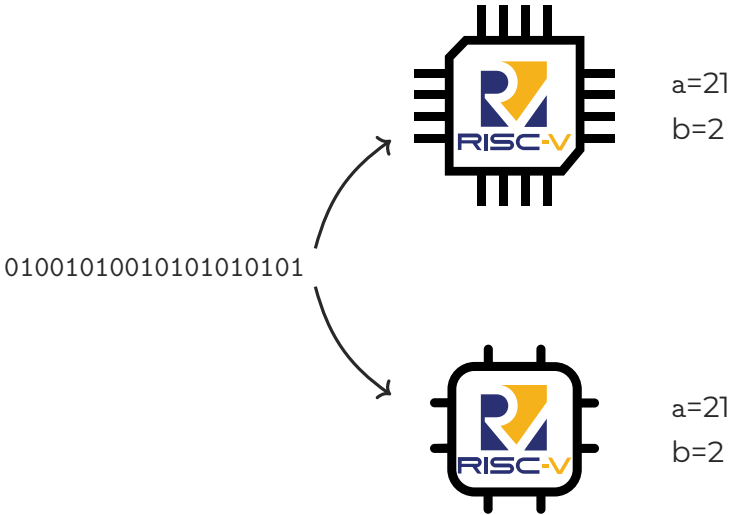
Differential CPU Fuzzing



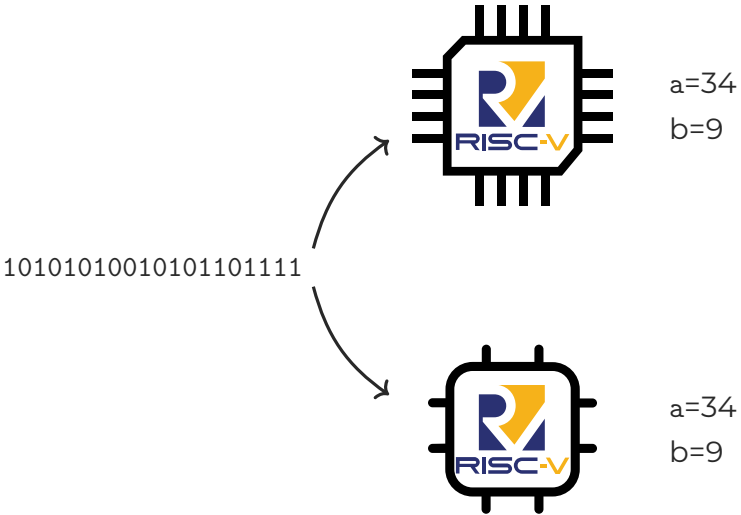
Differential CPU Fuzzing



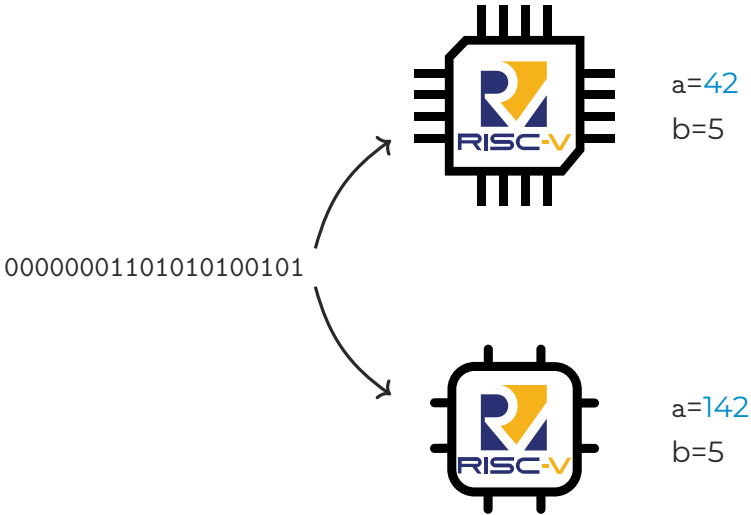
Differential CPU Fuzzing



Differential CPU Fuzzing



Differential CPU Fuzzing



Intel F00F Bug

A single (invalid) instruction deadlocked the CPU

Halt and Catch Fire (C906)

```
th.lbib t0, (t0), 0, 0
frcsr t0
li t0, 0
```

Halt and Catch Fire (C906)

```
th.lbib t0, (t0), 0, 0
frcsr t0
li t0, 0
```

CPU just hangs

Halt and Catch Fire (C906)

```
th.lbib t0, (t0), 0, 0
frcsr t0
li t0, 0
```

Synopsis

Load indexed byte, increment address before loading.

Mnemonic

`th.lbib rd, (rs1), imm5, imm2`

Description

This instruction increments the value in `rs1` by (`sign_extend(imm5)` « `imm2`) and writes the result back to `rs1`. After the increment of `rs1`, this instruction loads a sign extended 8-bit value into the GP register `rd` from the (incremented) address `rs1`.

The encoding of this instruction with equal rd and rs1 is reserved.

Halt and Catch Fire (C906)

```
th.lbib t0, (t0), 0, 0
frcsr t0
li t0, 0
```

Synopsis

Load indexed byte, increment address before loading.

Mnemonic

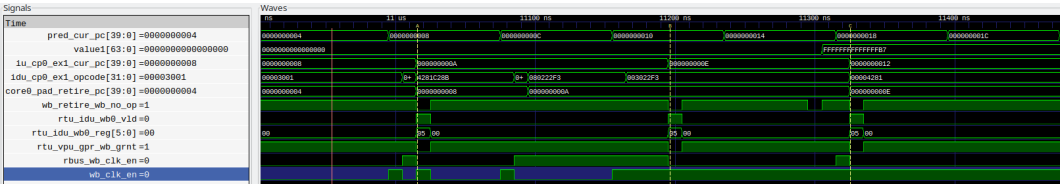
`th.lbib rd, (rs1), imm5, imm2`

Description

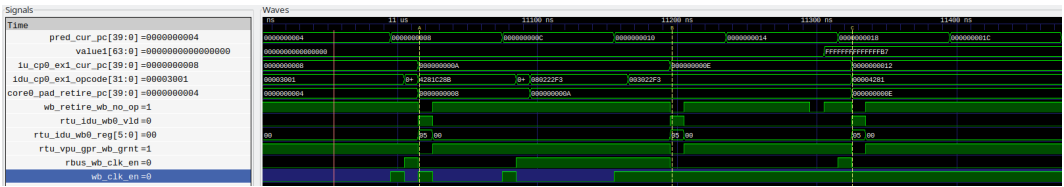
This instruction increments the value in `rs1` by (`sign_extend(imm5) « imm2`) and writes the result back to `rs1`. After the increment of `rs1`, this instruction loads a sign extended 8-bit value into the GP register `rd` from the (incremented) address `rs1`.

The encoding of this instruction with equal rd and rs1 is reserved.

Simulator Verification



Simulator Verification




```
**********************************************************************
* Error: There is no instructions retired in the last 50000 cycles! *
*                Simulation Fail and Finished!                       *
**********************************************************************
```

Other Cores?

```
.fill 1, 4, 0x20b00087
```

T-Head C908

Other Cores?

```
.fill 1, 4, 0x20b00087
```

T-Head C908

```
.fill 1, 4, 0xe0815407
```

SpacemiT X60

Mitigation Options

C908/X60 DoS

- Disable V extension in kernel
- Prevents denial-of-service on C908/X60

👎 Performance loss

👎 Breaks vector-dependent software

Mitigation Options

C908/X60 DoS

- Disable V extension in kernel
- Prevents denial-of-service on C908/X60

👎 Performance loss

👎 Breaks vector-dependent software

C906 DoS

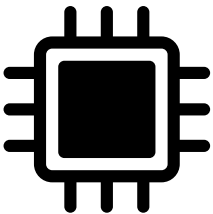
- T-Head vendor extension cannot be disabled
- “The th.sxstatus.THEADISAEE bit is not expected to be cleared. The behavior of clearing this bit is undefined”.

⚠ No known mitigation

GhostWrite

RISC-V Vector Instructions

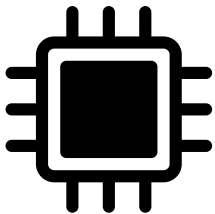
```
vse128.v v0, 0(t0)
```

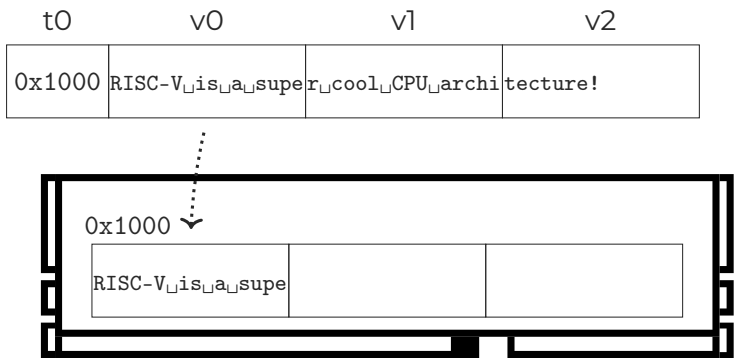


t0	v0	v1	v2
0x1000	RISC-V␣is␣a␣super	r␣cool␣CPU␣archi	tecture!

RISC-V Vector Instructions

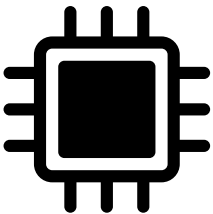
```
vse128.v v0, 0(t0)
```

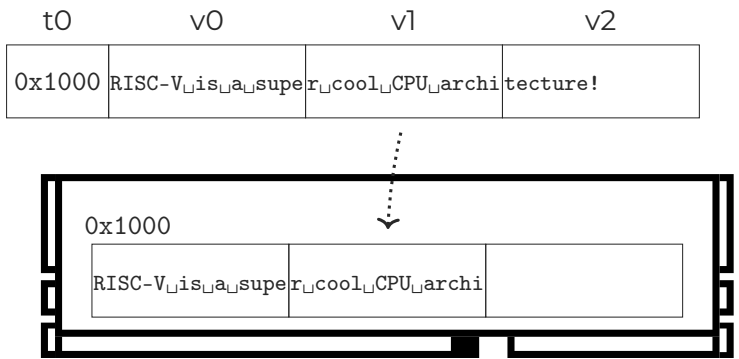




RISC-V Vector Instructions

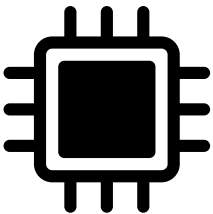
```
vse128.v v0, 0(t0)
```

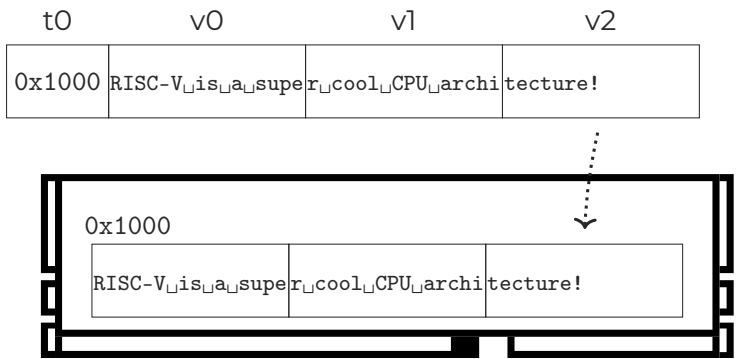




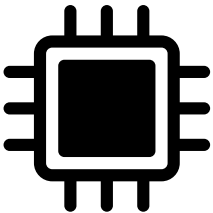
RISC-V Vector Instructions

```
vse128.v v0, 0(t0)
```

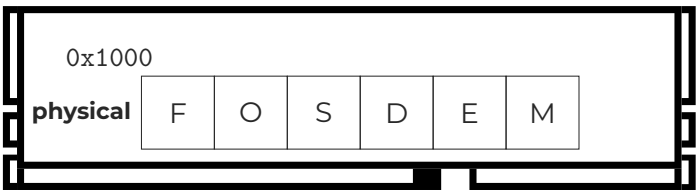




RISC-V Flawed Vector Instruction (C910)

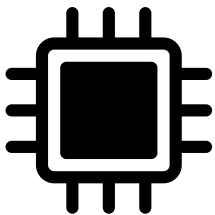


t0	v0	v1	v2	v3	v4	v5
0x1000	R	I	S	C	-	V

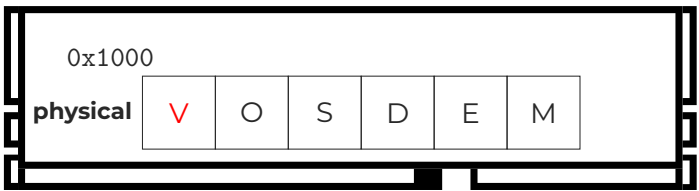


RISC-V Flawed Vector Instruction (C910)

```
vse128.v v0, 0(t0)
```

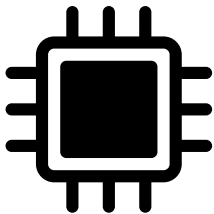


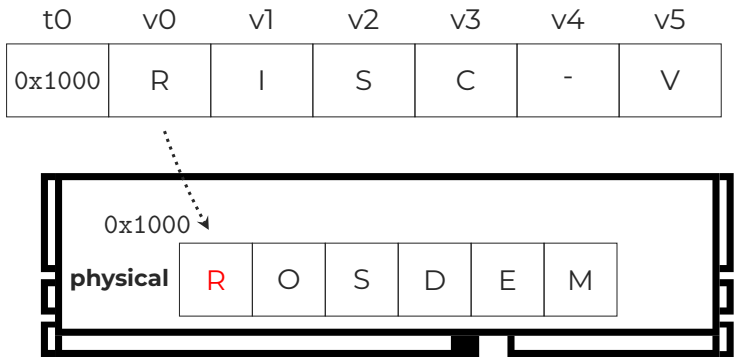
t0	v0	v1	v2	v3	v4	v5
0x1000	R	I	S	C	-	V



RISC-V Flawed Vector Instruction (C910)

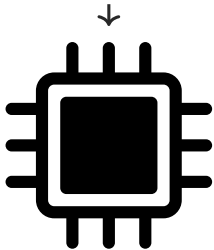
```
vse128.v v0, 0(t0)
```

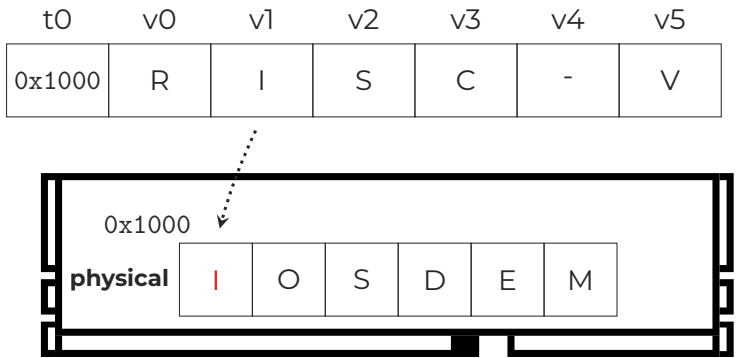




RISC-V Flawed Vector Instruction (C910)

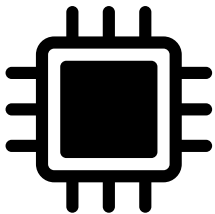
```
vse128.v v0, 0(t0)
```

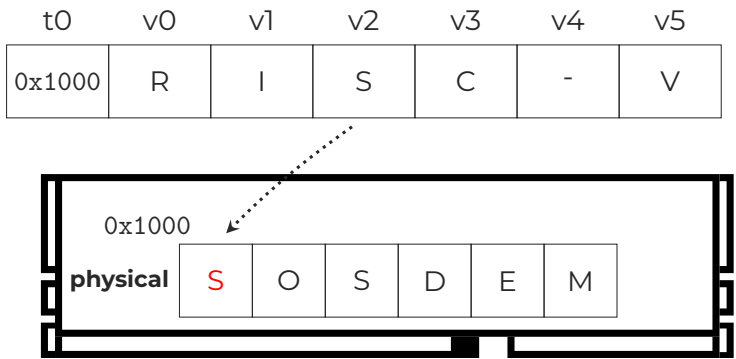




RISC-V Flawed Vector Instruction (C910)

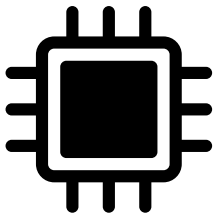
```
vse128.v v0, 0(t0)
```

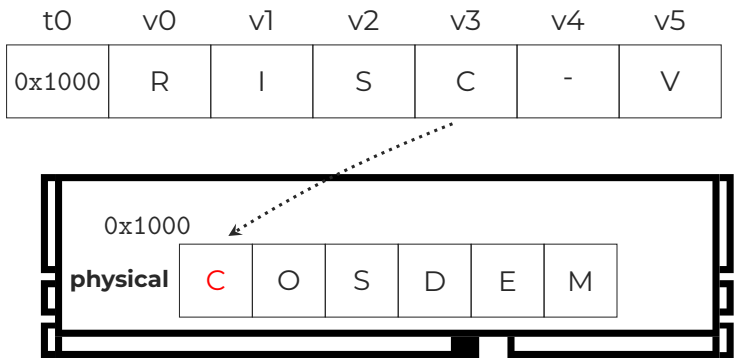




RISC-V Flawed Vector Instruction (C910)

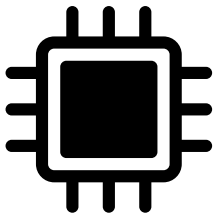
```
vse128.v v0, 0(t0)
```

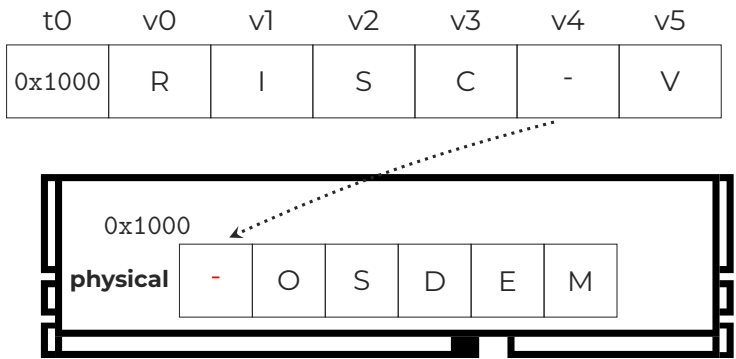




RISC-V Flawed Vector Instruction (C910)

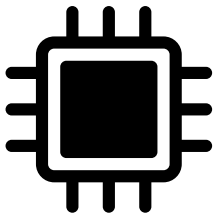
```
vse128.v v0, 0(t0)
```

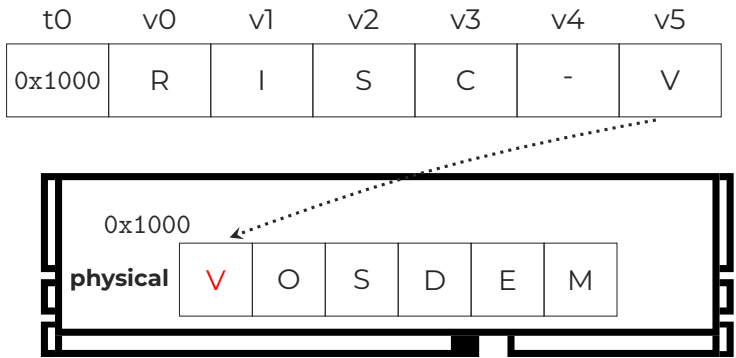




RISC-V Flawed Vector Instruction (C910)

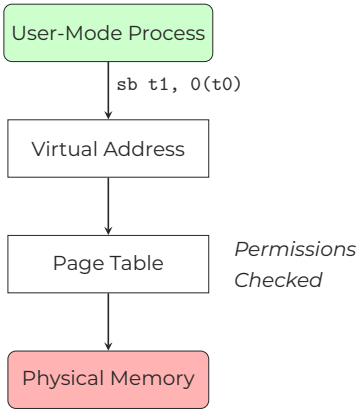
```
vse128.v v0, 0(t0)
```



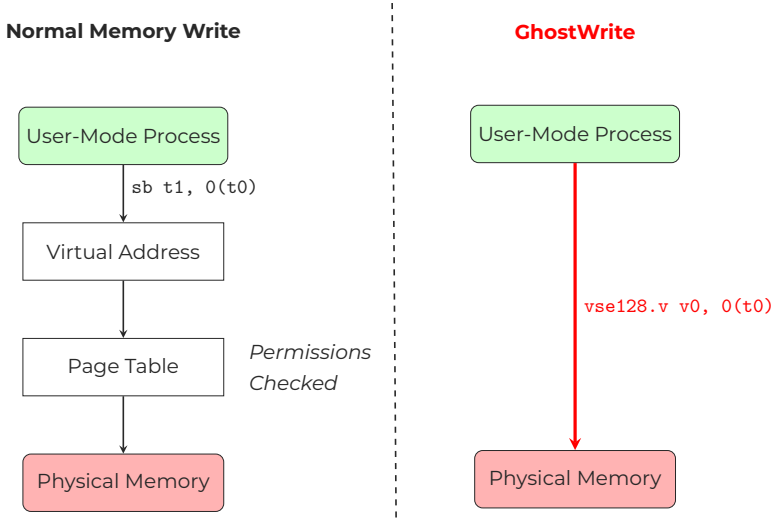


GhostWrite: Circumventing Virtual Memory

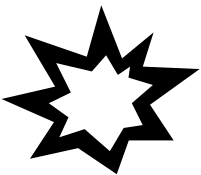
Normal Memory Write



GhostWrite: Circumventing Virtual Memory



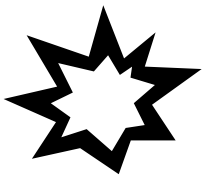
GhostWrite Exploitation



Rewrite page tables

Overwrite kernel

GhostWrite Exploitation



Rewrite page tables

Overwrite kernel

Modify M-mode firmware

Break trusted execution


unprivileged@c910:~$

Elastic Metal RV1

The world's first RISC-V servers available in the cloud.
Taste the new open processor architecture now.
Will you take the risk?

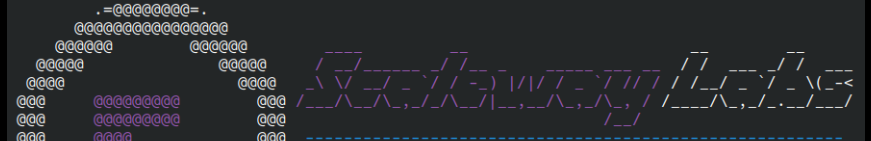
Order Now →

Read the press release →

An open RISC architecture

Compact and economical

Designed and assembled in Paris




```
Welcome to Ubuntu 23.10 (GNU/Linux 5.10.113+ riscv64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage


0 updates can be applied immediately.


Last login: Sat Mar  9 14:16:33 2024 from 87.145.163.240
ubuntu@risc:~$ sudo ./a.out
Virtual address:        3feab9f000
Phyiscal address:        3efc9b000

Value before:                caaa
Value after:                 cafe
ubuntu@risc:~$
```

GhostWrite Mitigation Options

Disable Vector Extension

- Disable V extension in kernel
- Blocks GhostWrite

👎 Performance loss

👎 No software relying on V

👎 Requires trusted kernel

GhostWrite Mitigation Options

Disable Vector Extension

- Disable V extension in kernel
- Blocks GhostWrite

👎 Performance loss

👎 No software relying on V

👎 Requires trusted kernel

```
Vulnerabilities:
  Gather data sampling:      Not affected
  Ghostwrite:                Mitigation; xtheadvector disabled
  Indirect target selection: Not affected
  Itlb multihit:             Not affected
  L1tf:                      Not affected
  Mds:                       Not affected
  Meltdown:                  Not affected
  Mmio stale data:           Not affected
  Reg file data sampling:    Not affected
  Retbleed:                  Not affected
```

GhostWrite Mitigation Options

Disable Vector Extension

- Disable V extension in kernel
- Blocks GhostWrite

👎 Performance loss

👎 No software relying on V

👎 Requires trusted kernel

Replace CPU

- Fix the bug in a new CPU revision
- Fixes GhostWrite

👎 Expensive and slow to deploy

👎 Requires hardware replacement

👍 Only real fix

GhostWrite Mitigation Options

Disable Vector Extension

- Disable V extension in kernel
- Blocks GhostWrite

👎 Performance loss

👎 No software relying on V

👎 Requires trusted kernel

Replace CPU

- Fix the bug in a new CPU revision
- Fixes GhostWrite

👎 Expensive and slow to deploy

👎 Requires hardware replacement

👍 Only real fix

Trade off: short-term software band aid (disable V) versus long-term architectural fix (new CPU revision).

Patterns From our Findings

Limited DV Scrutiny

- Hardware $\neq$ Software $\rightarrow$ DV is critical
- Differential testing across vendors

Patterns From our Findings

Limited DV Scrutiny

- Hardware $\neq$ Software $\rightarrow$ DV is critical
- Differential testing across vendors

High Diversity

- Implement only ratified extensions
- Use vendor extensions carefully
- Mandate specific behavior in ISA

Patterns From our Findings

Limited DV Scrutiny

- Hardware $\neq$ Software $\rightarrow$ DV is critical
- Differential testing across vendors

Unconfigurable Hardware

- Features should have kill switches
- Instruction hooking mechanism

High Diversity

- Implement only ratified extensions
- Use vendor extensions carefully
- Mandate specific behavior in ISA

Patterns From our Findings

Limited DV Scrutiny

- Hardware $\neq$ Software $\rightarrow$ DV is critical
- Differential testing across vendors

Unconfigurable Hardware

- Features should have kill switches
- Instruction hooking mechanism

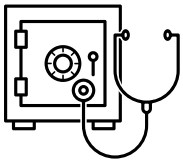
High Diversity

- Implement only ratified extensions
- Use vendor extensions carefully
- Mandate specific behavior in ISA

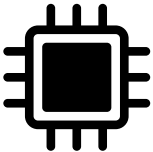
No Update Path

- Microcode or similar mechanism
- Even if microcode adds complexity
- We argue: Increases security

CPU Vulnerability Classes



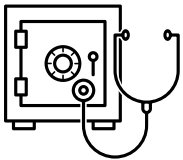
Side Channels



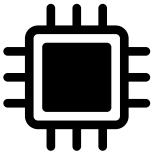
CPU Bugs

Transient Execution

CPU Vulnerability Classes



Side Channels



CPU Bugs

Transient Execution

```
lgerlach@lab53 ..eculate/experiments/pocs/e2e/userspace % CENSOR_HASHES=1 sudo -E ./hash_lookup_bpf /etc/shadow
```

Spectre on RISC-V: Implications

Out-of-order RISC-V CPUs are vulnerable to Spectre

Kernel lacks mitigations that x86 and ARM have had since 2018

No Speculation Barriers

x86

`LFENCE`

Standardized

ARM

`CSDB` / `DSB`

Standardized

RISC-V

Nothing

No dedicated fence

No Speculation Barriers

x86

`LFENCE`

Standardized

ARM

`CSDB` / `DSB`

Standardized

RISC-V

Nothing

No dedicated fence

- BPF JIT has no instruction to emit → emits no-op
- Workaround: CSR reads (`rdtime`) empirically stop speculation
- But: undocumented behavior, not guaranteed

Kernel Patches

Unprotected kernel paths on RISC-V

- Syscall dispatch – no `array_index_nospec`
- User memory access – no pointer masking
- BPF – JIT does not emmit speculation barrier
- Futex operations

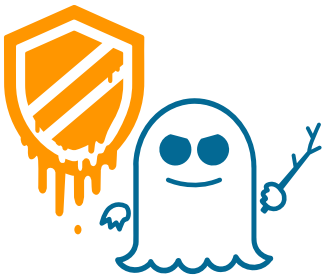
Kernel Patches

Unprotected kernel paths on RISC-V

- Syscall dispatch – no `array_index_nospec`
- User memory access – no pointer masking
- BPF – JIT does not emmit speculation barrier
- Futex operations

Patches submitted – some already merged into Linux mainline

What we need to mitigate this



- Standardized serializing fence instruction needed
- Kernel needs fine grained settings for mitigations
- Compiler needs support for retpoline equivalent
- Vendors need to document speculative behavior of CPUs

What goes wrong

- Unprivileged timers & cache maintenance
- Vendor extensions, non-ratified extensions
- Bugs: GhostWrite, DoS
- Missing kill switches, no update path
- Transient execution

What goes wrong

- Unprivileged timers & cache maintenance
- Vendor extensions, non-ratified extensions
- Bugs: GhostWrite, DoS
- Missing kill switches, no update path
- Transient execution

What we suggest

- Privileged cache maintenance
- Coarse user-space timers
- Ratified extensions only
- Kill switches
- Instruction hooking/ update mechanism
- Standardized serializing fence

What goes wrong

- Unprivileged timers & cache maintenance
- Vendor extensions, non-ratified extensions
- Bugs: GhostWrite, DoS
- Missing kill switches, no update path
- Transient execution

What we suggest

- Privileged cache maintenance
- Coarse user-space timers
- Ratified extensions only
- Kill switches
- Instruction hooking/ update mechanism
- Standardized serializing fence

What we wish for

- Documentation/Source code
- Unified perf interface
- Reproducible builds/kernel headers
- `mvendorid`/`marchid` DB

What goes wrong

- Unprivileged timers & cache maintenance
- Vendor extensions, non-ratified extensions
- Bugs: GhostWrite, DoS
- Missing kill switches, no update path
- Transient execution

What we suggest

- Privileged cache maintenance
- Coarse user-space timers
- Ratified extensions only
- Kill switches
- Instruction hooking/ update mechanism
- Standardized serializing fence

What we wish for

- Documentation/Source code
- Unified perf interface
- Reproducible builds/kernel headers
- `mvendorid`/`marchid` DB

`github.com/cispa/Security-RISC`

`github.com/cispa/RISCover`

Security Problem or Not?

CVE-2025-20103 (DoS)

Insufficient resource pool in the core management mechanism for some Intel Processors may allow an authenticated user to potentially enable denial of service via local access.

6.5 Medium

Security Problem or Not?

CVE-2025-20103 (DoS)

Insufficient resource pool in the core management mechanism for some Intel Processors may allow an authenticated user to potentially enable denial of service via local access.

6.5 Medium

CVE-2017-5754 (Meltdown)

Systems with microprocessors utilizing speculative execution and indirect branch prediction may allow unauthorized disclosure of information to an attacker with local user access via a side-channel analysis of the data cache.

5.6 Medium

Simple, but Effective

Small tests. Big failures.

Real bugs found in seconds or minutes across diverse RISC-V cores.

Privilege Escalation

C910, C920

<1 second

Denial-of-Service

C906, C908, X60

2–50 minutes

Architectural Bugs

U54, P550, C906, C908, C910

<2 minutes

Simulator Breakage

QEMU segfaults

<30 seconds