Can CHERI Stop Transient-Execution Attacks?

Toward Transient-Execution Attack Mitigations on CHERI Compartments

Franz A. Fuchs, Robert N. M. Watson, Simon W. Moore, Peter Sewell, Peter G. Neumann
Hesham Almatary, Jonathan Anderson, Alasdair Armstrong, Peter Blandford-Baker,
John Baldwin, Hadrien Barrel, Thomas Bauereiss, Ruslan Bukin, David Chisnall, Jessica Clarke, Nirav Dave, Brooks Davis,
Lawrence Esswood, Nathaniel W. Filardo, Dapeng Gao, Khilan Gudka, Brett Gutstein, Alexandre Joannou,
Mark Johnston, Robert Kovacsics, Ben Laurie, A. Theo Markettos, J. Edward Maste, Alfredo Mazzinghi,
Alan Mujumdar, Prashanth Mundkur, Steven J. Murdoch, Edward Napierala, George Neville-Neil, Robert Norton-Wright,
Philip Paeps, Lucian Paul-Trifu, Allison Randal, Ivan Ribeiro, Alex Richardson, Michael Roe, Colin Rothwell, Peter Rugg,
Hassen Saidi, Peter Sewell, Thomas Sewell, Stacey Son, Domagoj Stolfa, Andrew Turner, Munraj Vadera,
Konrad Witaszczyk, Jonathan Woodruff, Hongyan Xia, and Bjoern A. Zeeb

University of Cambridge and SRI International

RISC-V Summit in Europe

Barcelona, 6 June 2023

Approved for public release; distribution is unlimited. This research is sponsored by the Defense Advanced Research Projects Agency (DARPA) and the Air Force Research Laboratory (AFRL), under contract FA8750-10-C-0237. The views, opinions, and/or findings contained in this article/presentation are those of the author(s)/presenter(s) and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

Approved for public release; distribution is unlimited.

This work was supported by the Defense Advanced Research Projects Agency (DARPA) and the Air Force Research Laboratory (AFRL), under contract FA8750-10-C-0237 ("CTSRD"), with additional support from FA8750-11-C-0249 ("MRC2"), HR0011-18-C-0016 ("ECATS"), and FA8650-18-C-7809 ("CIFV") as part of the DARPA CRASH, MRC, and SSITH research programs. The views, opinions, and/or findings contained in this report are those of the authors and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

This work was supported in part by the Innovate UK project Digital Security by Design (DSbD) Technology Platform Prototype, 105694.

This work was supported by the Engineering and Physical Sciences Research Council EP/S030867/1.

We also acknowledge the EPSRC REMS Programme Grant (EP/K008528/1), the ERC ELVER Advanced Grant (789108), the Isaac Newton Trust, the UK Higher Education Innovation Fund (HEIF), Thales E-Security, Microsoft Research Cambridge, Arm Limited, Google, Google DeepMind, HP Enterprise, and the Gates Cambridge Trust.

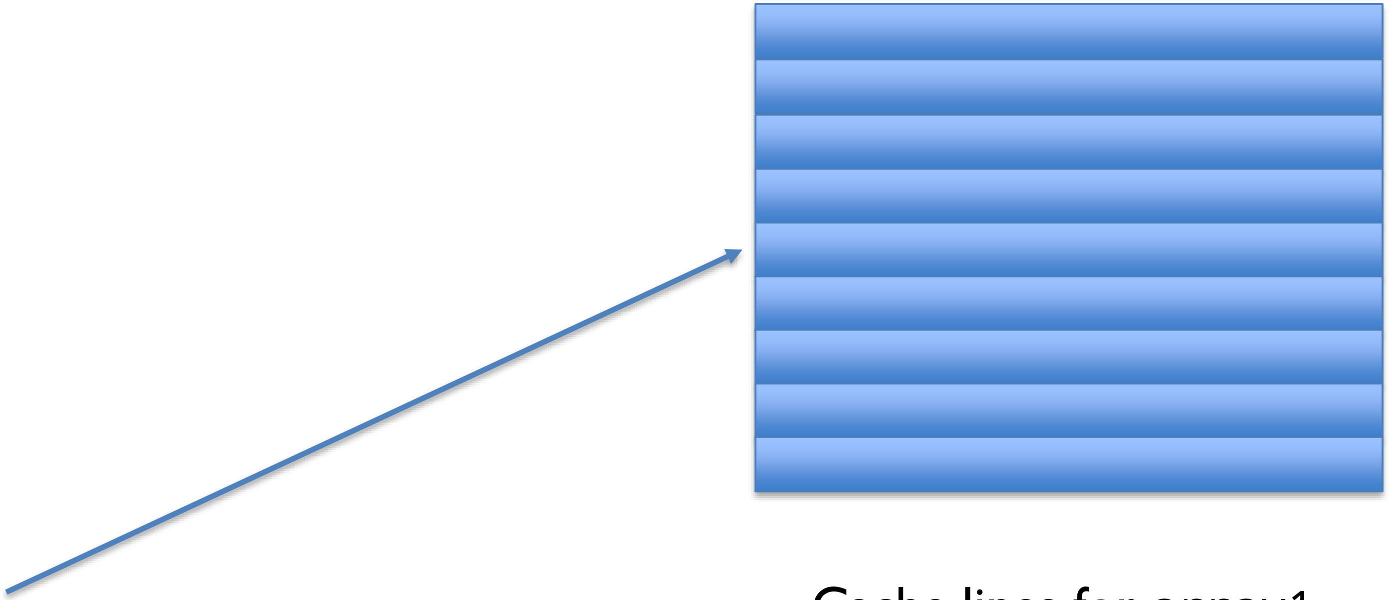
Transient-Execution Attacks

Transient-execution attacks combine:

- Directed speculative execution
- Side-channels, e.g., cache timing

Spectre v1 is most infamous example:

```
if (idx0 < size){
        int idx1 = array0[idx0];
        int idx2 = array1[idx1];
        …
}
```



Cache lines for `array1`

Transient-Execution Attacks

Transient-execution attacks are:

- Numerous: multiple subclasses (Spectre, Meltdown, Microarchitectural Data Sampling)
- Widely applicable: RISC-V, x86, AArch64
- Dangerous: can potentially leak any microarchitectural state, speed of leaking secrets going up to multiple hundred KB/s

Does this hold for secure systems as well?

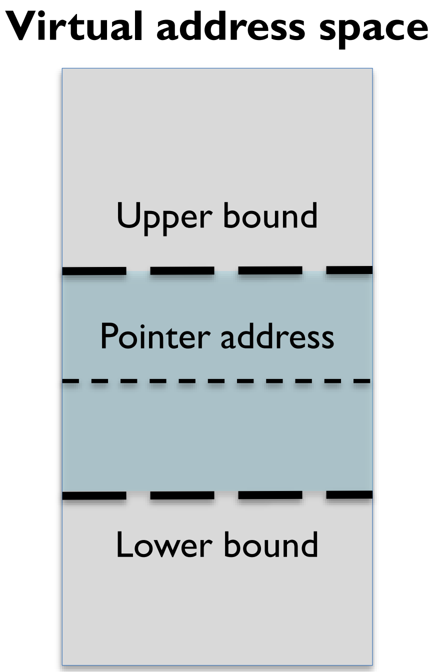
Leakage sources and reasons:

- Branch direction prediction
- Indirect jump target prediction
- Return address prediction
- Memory disambiguation
- Speculative load forwarding
- Instruction scheduling
- Out-of-order execution
- Reading from store buffers
- …

CHERI

Capability Hardware Enhanced RISC Instructions (CHERI):

- Capability: pointer extended by metadata
- CHERI capabilities both describe memory regions and authorise access to them
- Abstract CHERI ISA is mapped as an extension to conventional ISAs, e.g., RISC-V and AArch64 (Morello)



Transient-Execution Attacks on CHERI

Previous research shows that CHERI systems can mitigate some transient-execution attacks, but might be vulnerable to others:

- Traditional Spectre attacks mostly work
- Attempted Meltdown attacks all mitigated
- With careful microarchitectural design, we believe that a superscalar CHERI processor can mitigate all attacks while remaining performant

	CHERI-RISC-V
Spectre-PHT	Safe*
Spectre-BTB	Vulnerable
Spectre-RSB	Vulnerable
Spectre-STL	Vulnerable
Meltdown-US	Safe
Meltdown-GP	Safe

*only with precise capability bounds

Results obtained on CHERI-Toooba

Transient-Execution Attacks on CHERI

"The main observations"

Microarchitectural design properties rather than (CHERI) ISA determine whether a processor is vulnerable

Transient-execution attacks are harmful when they cross domain borders

This holds not only for CHERI systems, but **generally** for all architectures

Current ISAs and Speculation (1)

"ISAs do not care about microarchitectures!"

Main ISA design properties:

- Generality: needs to suit all implementations, avoid over-constraining
- Sequential programming model: execute one instruction at a time

Today's microarchitectures:

- Execute multiple instructions at the same time
- Extensive use of speculation to gain performance

→ A large gap has opened up between ISAs and microarchitectures

Current ISAs and Speculation (2)

“Insufficient ISA specifications”

Speculative behaviour generally unspecified:

- Rudimentary barriers, e.g., x86 has the `lfence` instruction preventing speculative out-of-order execution
- Switching off predictors, e.g., AArch64 offers a bit to switch off memory disambiguation

RISC-V:

?

Architectural Guarantees (1)

"Fill the architectural specification vacuum"

Currently, the architecture does not constrain speculative execution

We need ISA-level guarantees about speculative execution for security

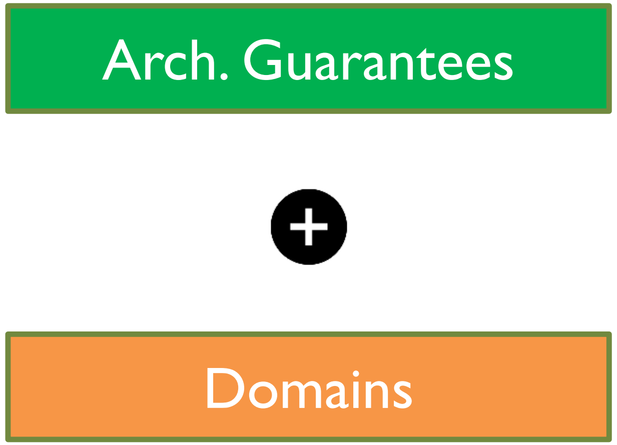
Guarantees are needed:

- To test key security properties of the microarchitecture
- To allow secure software to be built on top of ISA-level guarantees

Architectural Guarantees (2)

“Fill the architectural specification vacuum”

- Preventing all speculative execution would severely impact performance
- Solution: allow speculation within a domain, but not allow speculation to cross domain borders



Arch. Guarantees

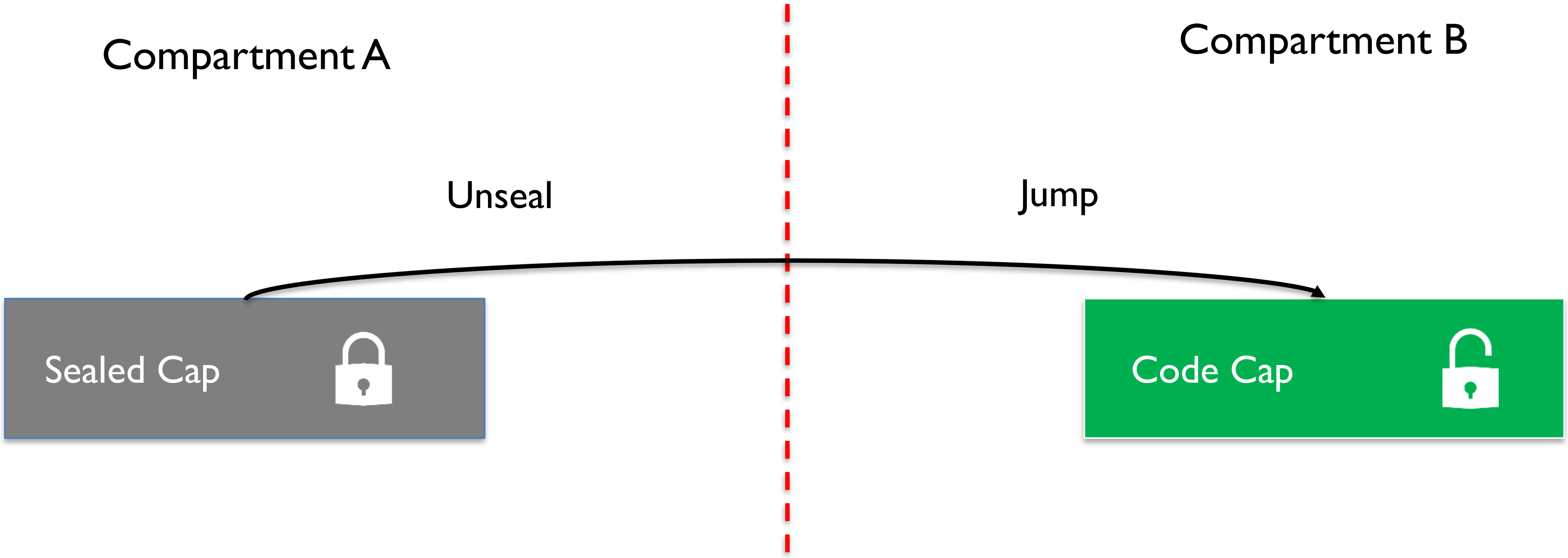
+

Domains

→ Perform compartmentalisation to create domains

How do we prevent transient-execution attacks from crossing domains?

CHERI Compartmentalisation



Sealed capability is unsealed and becomes a code capability in another domain in one atomic operation

→ Open question: How can we enforce these compartments in speculation?

CHERI Compartmentalisation

Advantages:

- Atomic domain transitions allow for clean compartmentalisation
- Fine-grained decomposition with capabilities

→ CHERI can allow for secure compartmentalisation mechanisms

Conclusions

- Transient-execution attacks are a threat to many systems
- Current ISA specifications fail to constrain the microarchitecture – so software lacks security guarantees
- Specification of ISA-level security guarantees that microarchitectures must uphold is crucial
- CHERI can support secure, fine-grained and efficient compartmentalisation