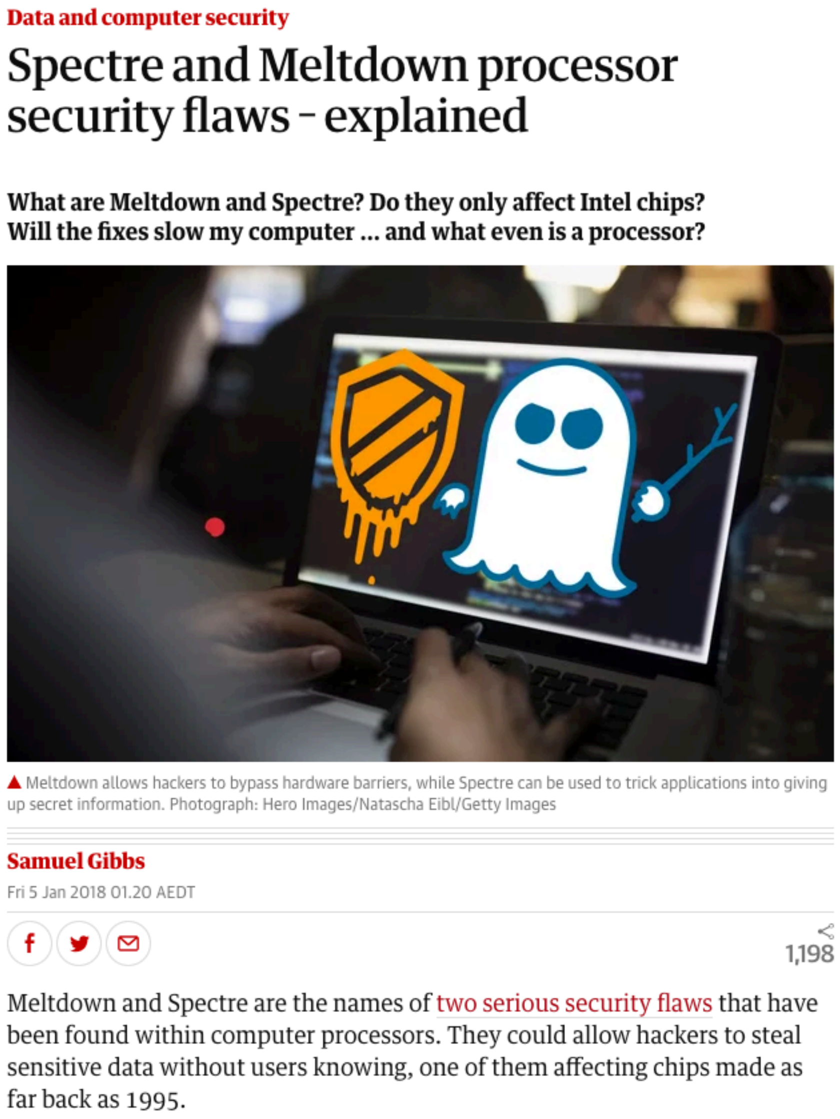
No Security Without Time Protection

We Need a New Hardware-Software Contract!

Qian Ge, Yuval Yarom, Gernot Heiser
gernot.heiser@data61.csiro.au | @GernotHeiser
APSys'18, Jeju, Korea

https://trustworthy.systems

The New Year Shock



Data and computer security

Spectre and Meltdown processor security flaws – explained

What are Meltdown and Spectre? Do they only affect Intel chips? Will the fixes slow my computer … and what even is a processor?

▲ Meltdown allows hackers to bypass hardware barriers, while Spectre can be used to trick applications into giving up secret information. Photograph: Hero Images/Natascha Eibl/Getty Images

Samuel Gibbs

Fri 5 Jan 2018 01.20 AEDT

1,198

Meltdown and Spectre are the names of two serious security flaws that have been found within computer processors. They could allow hackers to steal sensitive data without users knowing, one of them affecting chips made as far back as 1995.

NDY GREENBERG SECURITY 01.03.18 03:00 PM

A CRITICAL INTEL FLAW

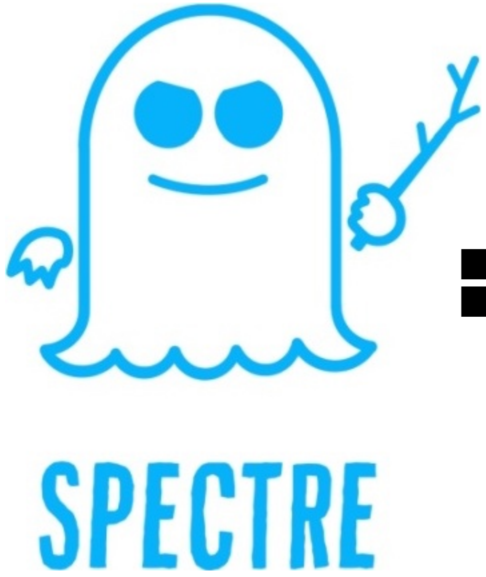
BREAKS BASIC SECURITY

TERS

rn Intel processors could

ty of most computers

Threats



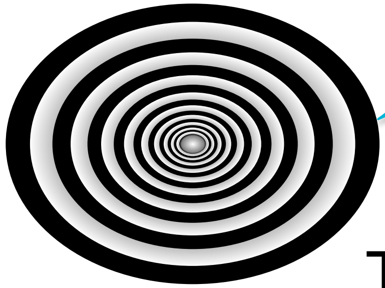
=

An "unknown unknown" until recently

Speculation

+

A "known unknown" for decades



Timing Channel

Overview

- What are timing channels?
- *Time protection:* OS must close microarchitectural channels
- How helpful is present hardware?
- What are the requirements on hardware for closing timing channels?
- Defining the new hardware-software contract – aISA

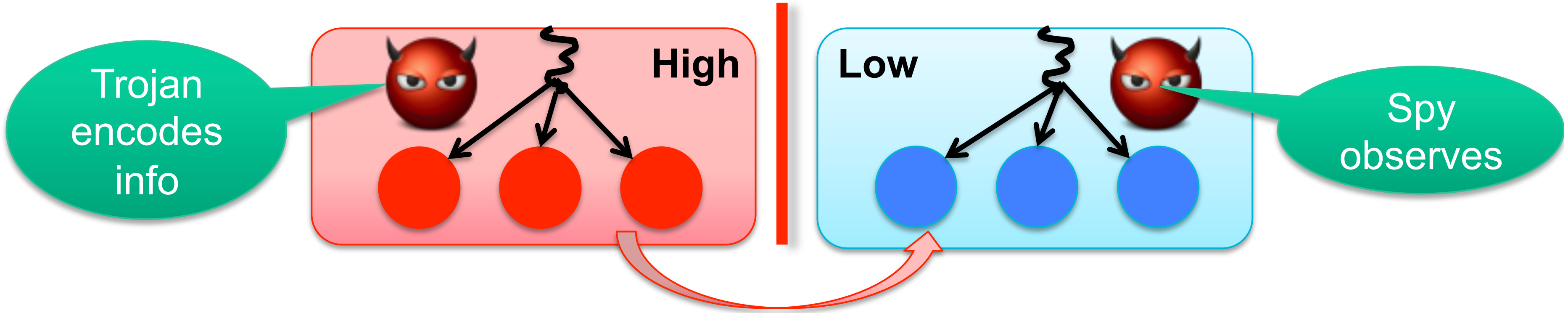
What are Timing Channels?

Timing Channels

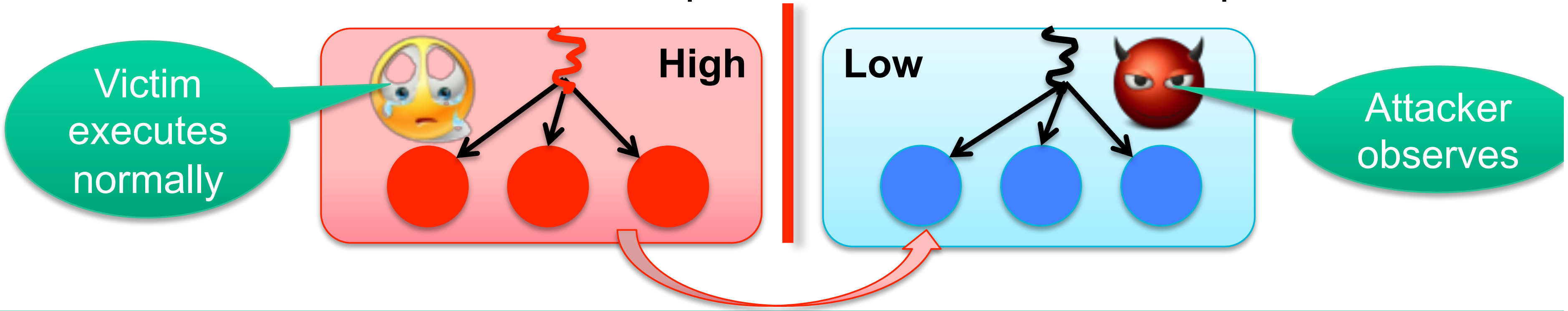
Information leakage through timing of events

- Typically by observing response latencies or own execution speed

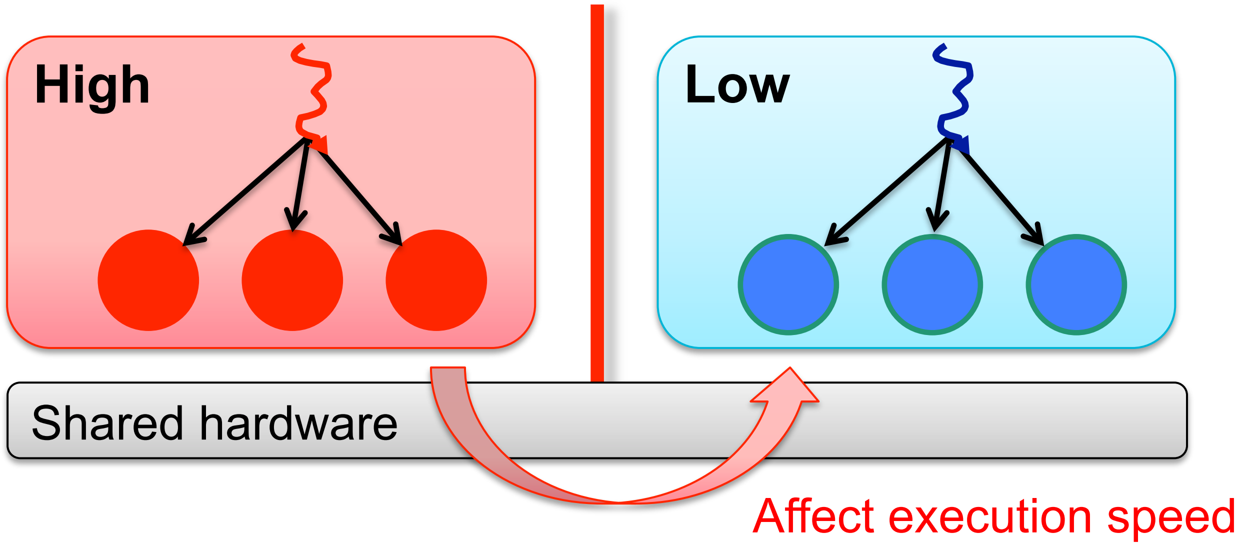
Covert channel: Information flow that bypasses the security policy



Side channel: Covert channel exploitable without insider help

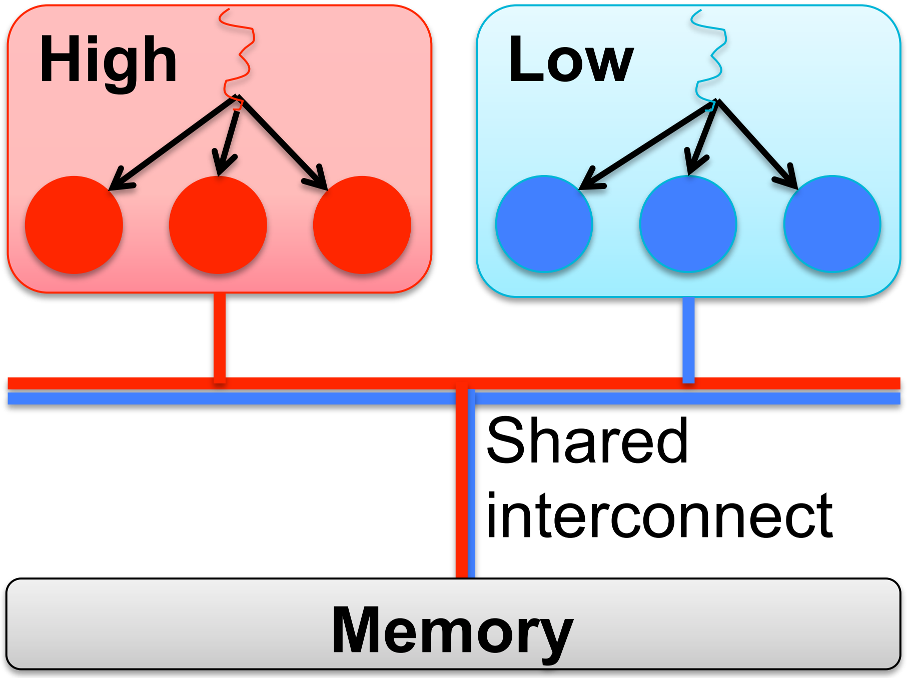


Origin of Timing Channels: Temporal Interference



- Inter-process interference
- Competing access to micro-architectural features
  - not exposed by the ISA
  - hidden by the HW-SW contract!

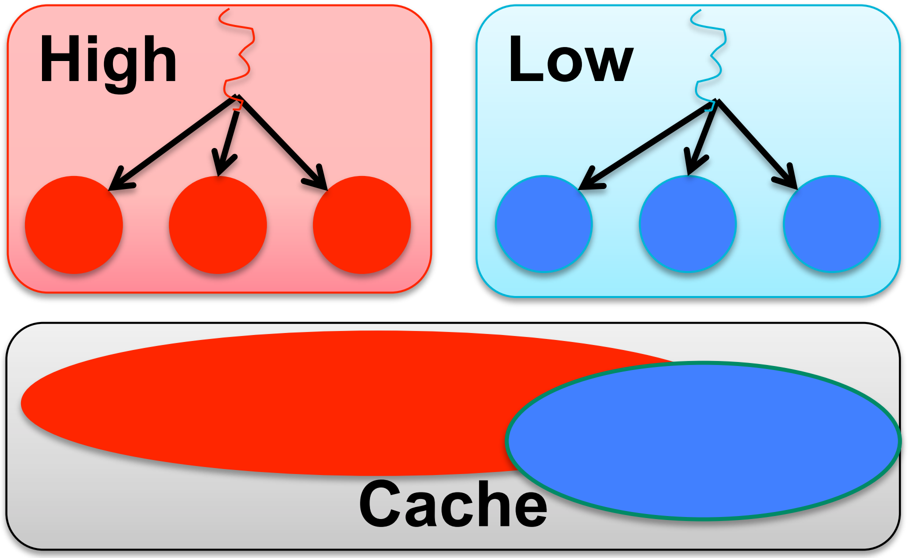
Sharing 1: Stateless Interconnect



H/W is *bandwidth-limited*

- Interference during concurrent access
- Generally reveals no data or addresses
- Must encode info into access patterns
- Only usable as covert channel, not side channel

Sharing 2: Stateful Hardware



HW is *capacity-limited*

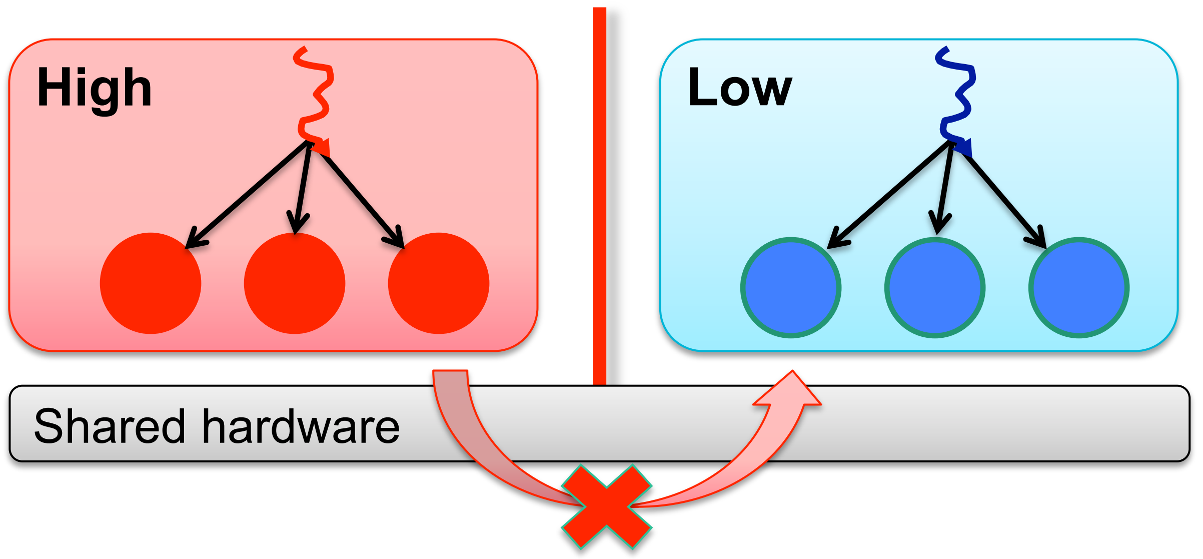
- Interference during
  - concurrent access
  - time-shared access
- Collisions reveal data or addresses
- *Usable as side channel*

Any state-holding microarchitectural feature:

- cache
- branch predictor
- pre-fetcher state machine

Time Protection

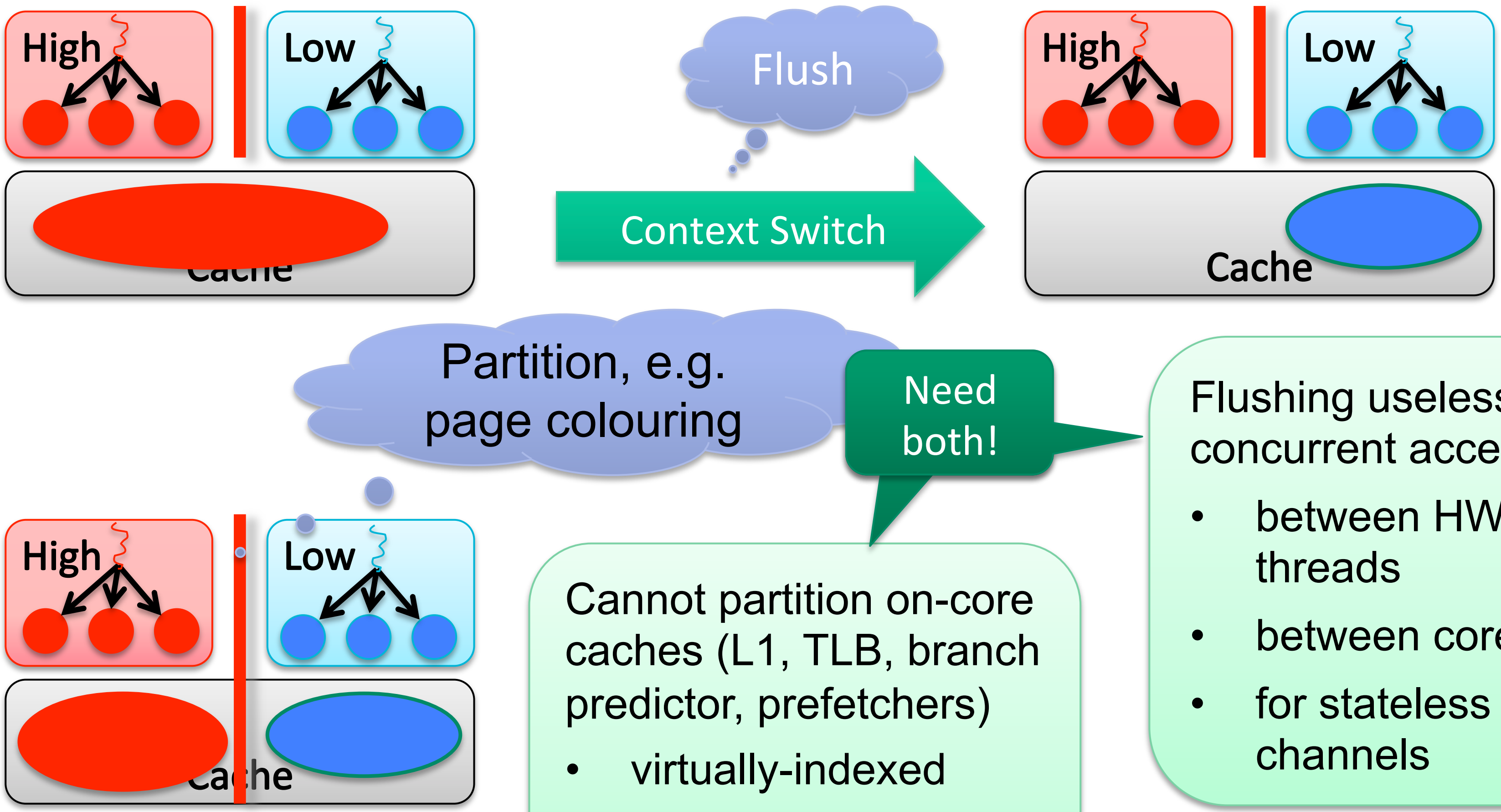
OS Must Enforce *Time Protection*



Preventing interference is core duty of the OS!

- *Memory protection* is well established
- *Time protection* is completely absent

Time Protection: No Sharing of State



Cannot partition on-core caches (L1, TLB, branch predictor, prefetchers)

- virtually-indexed
- OS cannot control access

Flushing useless for concurrent access

- between HW threads
- between cores
- for stateless channels

Requirements For Time Protection

Off-core state & stateless HW

Timing channels can be closed *iff* the OS can

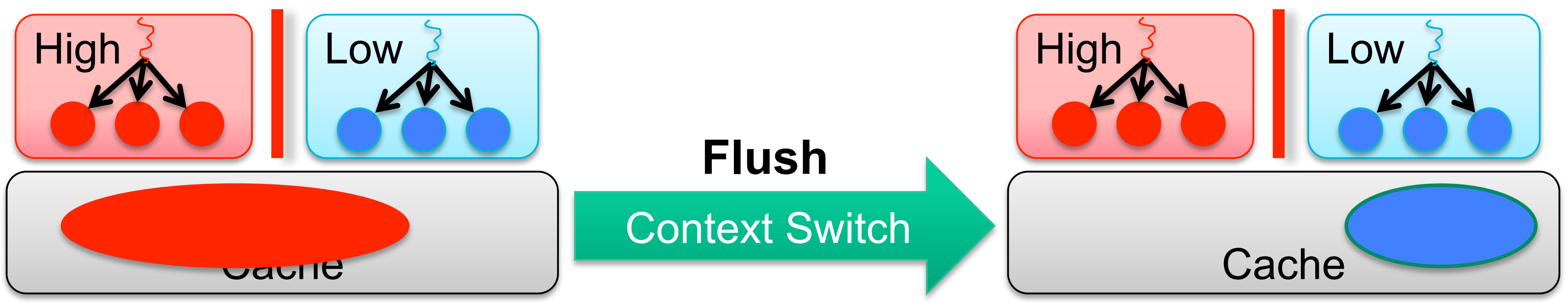
- partition or
- reset

all shared hardware

On-core state

Reality Check:
Resetting Intra-Core State

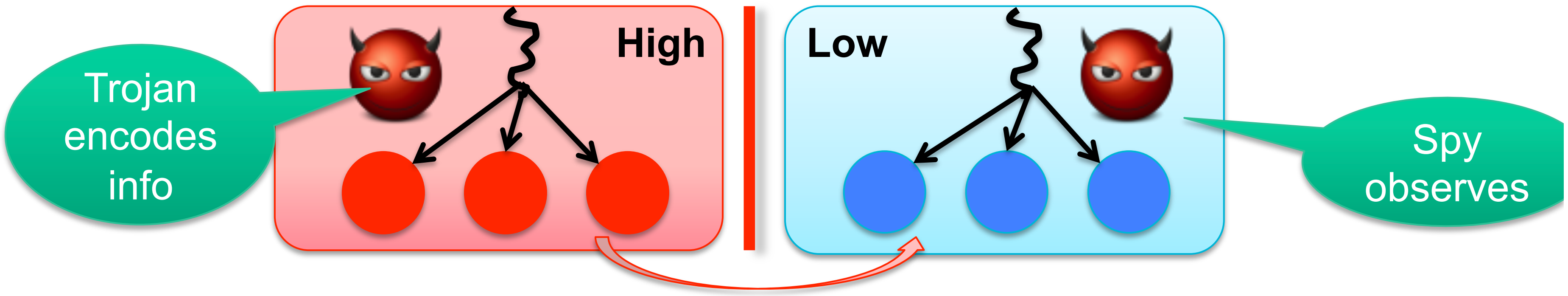
Evaluating Intra-Core Channels



Mitigation on Intel and Arm processors:

- Disable data prefetcher
- On context switch, perform all architected flush operations:
  - Intel: `wbinvd` + `invpcid` (no targeted L1-cache flush supported)
  - Arm: `DCCISW` + `ICIALLU` + `TLBIALL` + `BPIALL`

Methodology: Prime & Probe



1. Fill cache with own data

2. Touch *n* cache lines

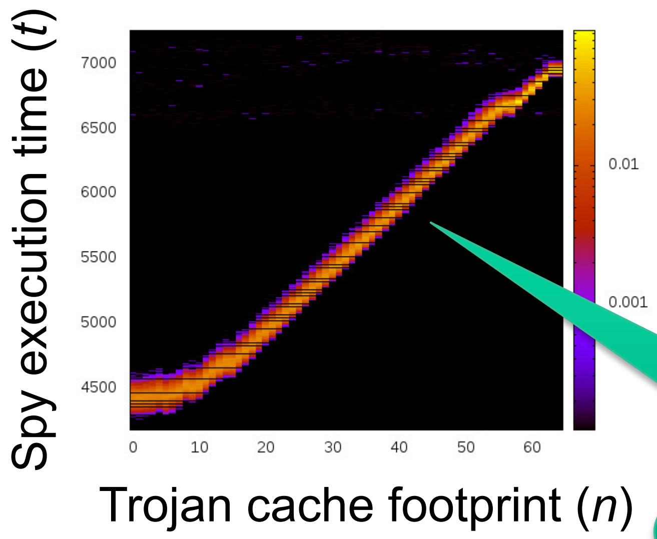
2.

3. Traverse cache, measure execution time

Output Signal

Input Signal

Methodology: Channel Matrix



Channel Matrix:

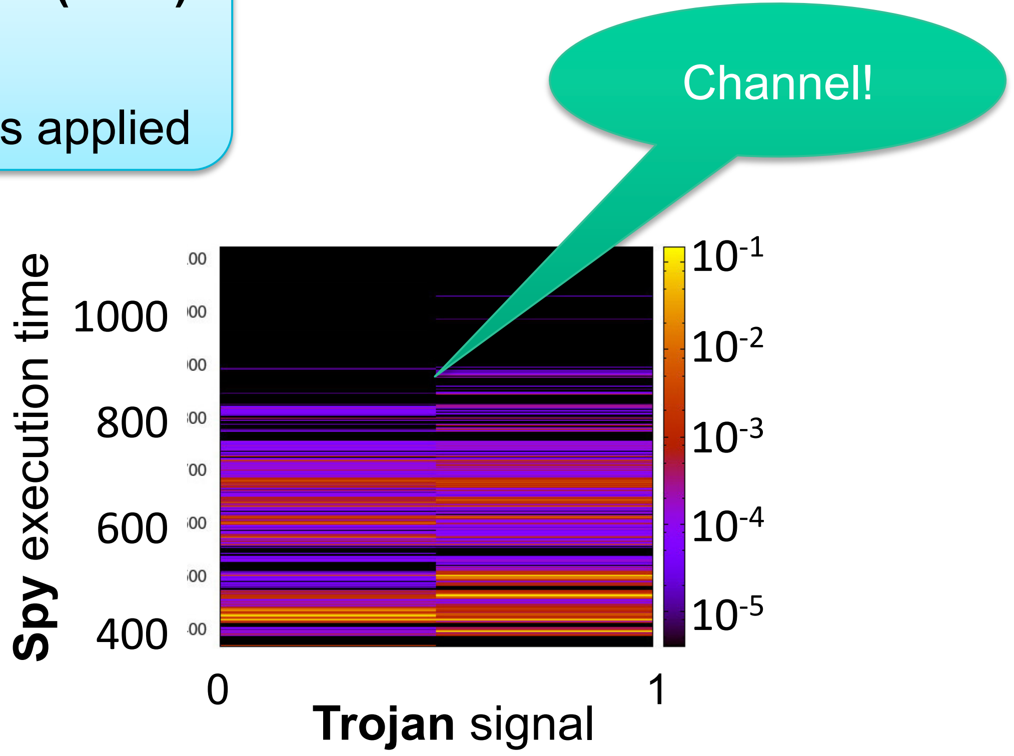
- Conditional probability of observing time, *t*, given input, *n*.
- Represented as heat map:
  - bright = high probability
  - dark = low probability

Horizontal variation indicates channel

Example: ARM A53 BHB

Branch history buffer (BHB)

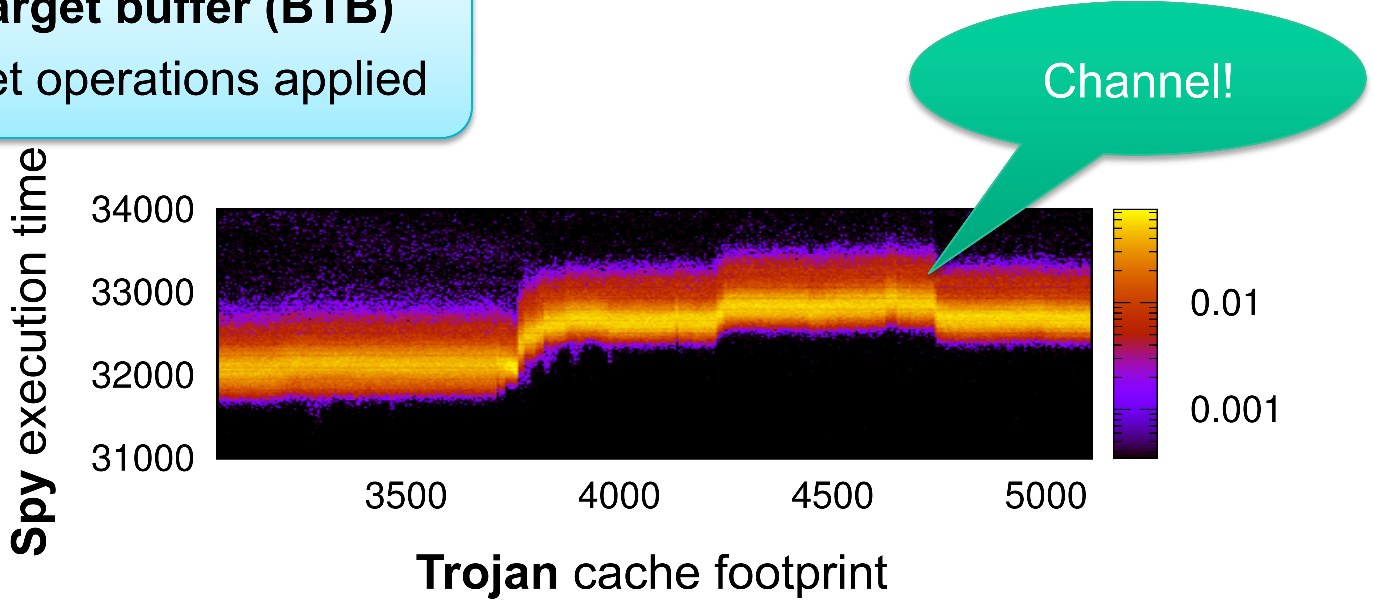
- One-bit channel
- All reset operations applied



Example: Intel Haswell BTB

Branch target buffer (BTB)

- All reset operations applied



Found residual channels in all recent Intel and ARM processors examined!

Latest Intel ISA addition (IBC) helps, but prohibitive without targeted L1 cache flush operation

Complete dataset at
https://ts.data61.csiro.au/projects/TS/timingchannels/arch-mitigation.pml

Requirements on Hardware

Hardware-Software Contract: ISA

- The ISA is a purely operational contract
  - sufficient to ensure *functional correctness*
  - abstracts away *time*
  - insufficient for ensuring either timing safety or security
- For security need an abstraction of microarchitectural state
  - essential for letting OS provide time protection

Remember: Timing channels can only be closed if all shared hardware state can be partitioned or reset

New HW/SW Contract: aISA

Augmented ISA supporting time protection

For all shared microarchitectural resources:

1. Resource must be partitionable or resetable
2. Concurrently shared resource must be partitioned
3. Resource accessed by virtual address must be reset and not concurrently accessed
   – implies cannot share HW threads across security domains!
4. Mechanisms must be sufficiently specified to allow OS to partition or reset
   – must be constant time or of specified, bounded latency
5. OS must know if resettable state is derived from data, instructions, data addresses or instruction addresses

Cost of Reset

- **Flushing on-core state** is not a performance issue:
  - no cost when not used
  - direct flush cost should for dirty L1-D in the order of 1μs
  - direct flush cost for everything else in the order of 1 cycle
  - indirect cost should be negligible, if used on security-partition switch
    - eg VM switch, 10–100 Hz rate
    - no hot data in cache after other partition's execution

- **Hardware support is essential!**

Summary

- Timing channels are a mainstream security threat
- The are based on competition for shared hardware
- This hardware is hidden by the ISA, the present HW-SW contract
  - Cannot systematically prevent timing channels based on ISA
  - OS cannot provide the required *time protection*
- Need a new, security-oriented contract, the aISA
  - aISA must expose enough microarchitecture for OS to enforce time protection

DATA 61

Thank you

Qian Ge, Yuval Yarom, Gernot Heiser
gernot.heiser@data61.csiro.au | @GernotHeiser

https://trustworthy.systems

CSIRO