A gem5-Based Simulation Platform for Evaluating RISC-V Security Against Microarchitectural Side-Channel Attacks

Mahreen KHAN

Telecom Paris, Institut Polytechnique de Paris

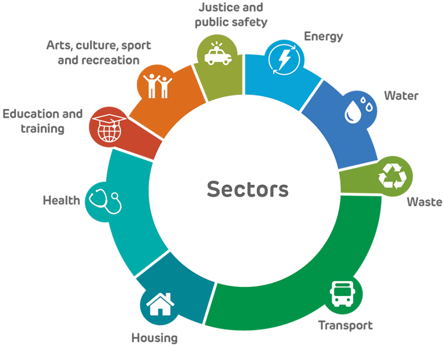
Agenda of Presentation

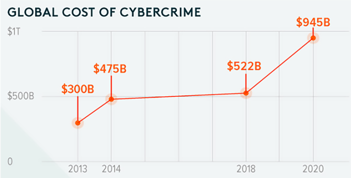
1. Security Basics
2. Side-Channel Attacks
3. RISC-V attack
4. Gem5 Analysis
5. HPCs
6. ML

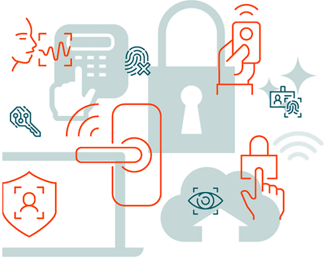
Security Basics

Information Security Perspective

Why Security Matters?

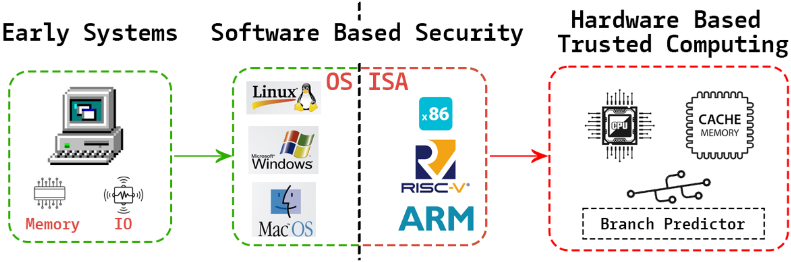






https://www.visualcapitalist.com/sp/thematic-investing-3-key-trends-in-cybersecurity/

Secure Software: Can data still be leaked?



- Consider CPU as a black box
- Assume no bugs in software

Can data still be leaked?

Yes - Through hardware vulnerabilities and side channel attacks

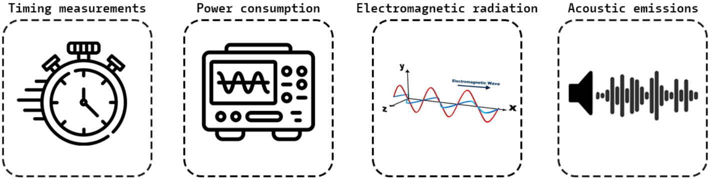
Side-Channel Attacks

Focus on Micro-architectural Side-Channels

Side-Channel: Major Security Concern

Side Channel Attacks

Side channel information can be collected from the physical behavior of a system and exploited by attackers to extract sensitive data.



Timing measurements

Power consumption

Electromagnetic radiation

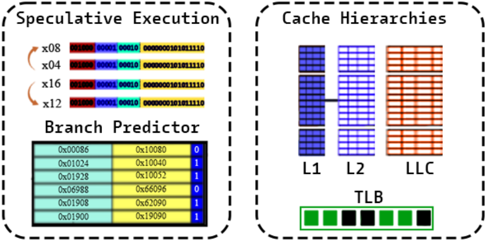
Acoustic emissions

Different types of Side-channels

Focus for this presentation:
Microarchitectural timing side-channel attacks

Understanding Microarchitecture

- Performance optimization elements:
  - Speculative execution
  - Cache hierarchies
  - Branch prediction



x08	001000	00001	00010	0000000101011110
x04	001000	00001	00010	0000000101011110
x16	001000	00001	00010	0000000101011110
x12	001000	00001	00010	0000000101011110

0x00086	0x10080	0
0x01024	0x10040	1
0x01928	0x10052	1
0x06988	0x66096	0
0x01908	0x62090	1
0x01900	0x19090	1

Typical CPU microarchitecture components

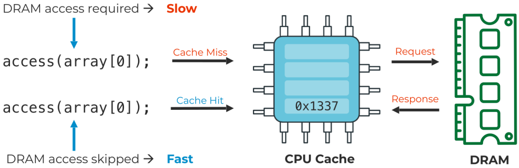
Cache Optimization

Benefits:

- Reduces memory latency (10-100x shorter than DRAM)
- Improves power efficiency

Risks:

- Creates timing side-channels
- Leaks access patterns
- Reveals cryptographic secrets



Famous Attack: Flush+Reload

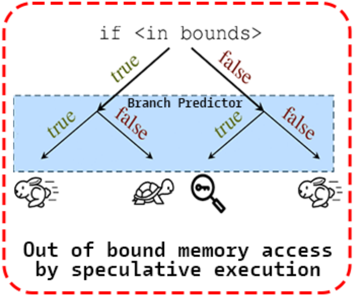
Speculative Execution and Branch Prediction

Benefits:

- Improves pipeline utilization

Risks:

- Out of bound memory access



Famous attack: Spectre

RISC-V attack

Focus on Flush+Fault attack on RISC-V

RISC-V: Emerging Architecture

Why RISC-V Matters

62.4 billion RISC-V cores forecast by 2026
(Market projection across IoT, AI and security-sensitive domains)

RISC-V: Challenges

A lot of work is done to understand Intel x86 and ARM vulnerabilities.

BUT what about RISC-V?

RISC-V Challenges: Critical Gap

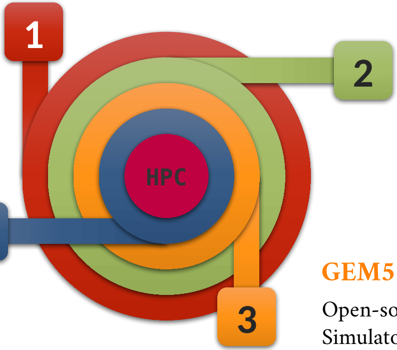
- Less mature RISC-V security analysis
- Custom extension security
- Verification complexity

RISC-V: Virtual Security Testing Platform

Bridging the Gap

- Open-source virtual platform for RISC-V security research
- Enables testing of microarchitectural attacks and defenses
- Prototypes Hardware Performance Counters (HPCs) for security use

Overall Methodology



DETECTION

Using HPC with or without Machine Learning

ATTACK

Micro-architectural side-channel attacks

RISC-V

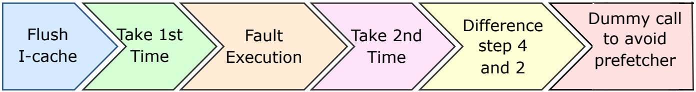
Target Architecture

GEM5

Open-source Simulator

I will focus on the analysis of the **Flush+Fault** attack on RISC-V.
A similar methodology was applied to the **Evict+Spec+Time** attack.

Flush+Fault attack: Detailed Attack Steps



Flush I-cache

Take 1st Time

Fault Execution

Take 2nd Time

Difference step 4 and 2

Dummy call to avoid prefetcher

1. Flush the instruction cache using fence.i.
2. Record a precise timestamp immediately after the flush.
3. Triggers a fault or a return by jump to a victim instruction.
4. Record second timestamp after the fault or return.
5. Calculate the time delta between both timestamps:
   - Shorter time indicates a cache hit.
   - Longer time indicates a cache miss.
6. To avoid speculative prefetching, the attacker issues multiple calls to dummy locations outside the targeted cache line.

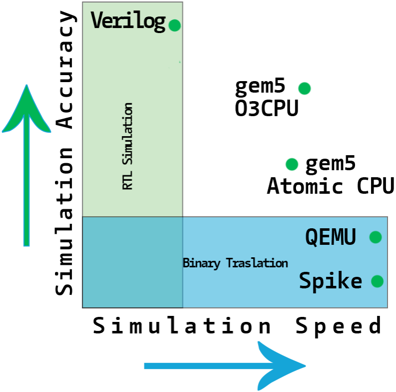
Gem5 Analysis

Flush+Fault analysis using Gem5

Our Approach: Simulated Environment for Attack Assessment

Tool Comparison

- ✓ gem5: Full-system, cycle-accurate but moderate speed
- QEMU: Fast emulation but less accurate
- Spike: ISA simulator but no timing accuracy
- Verilog Simulators: Highly accurate but very slow.

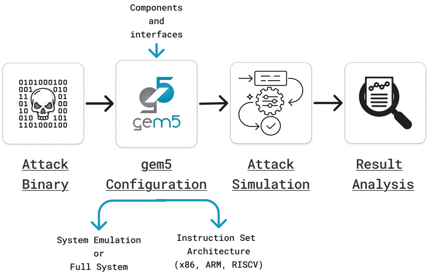


Gem5 simulator: a tool for security analysis

Why Choose gem5?

- Full-system simulation
- Rich microarchitectural stats

Attack Analysis using gem5



Components and interfaces

Attack Binary

gem5 Configuration

Attack Simulation

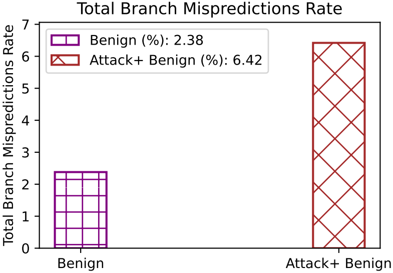
Result Analysis

System Emulation or Full System

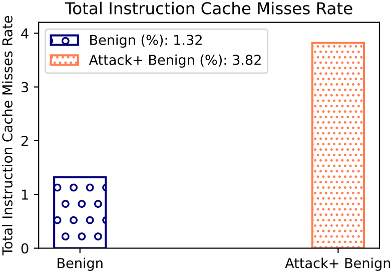
Instruction Set Architecture (x86, ARM, RISCV)

Gem5 Analysis Results

Tested **Flush+Fault** which exploits instruction cache flushing and branch mispredictions.



Total Branch Mispredictions Rate



Total Instruction Cache Misses Rate

Hardware Performance Counters (HPCs)

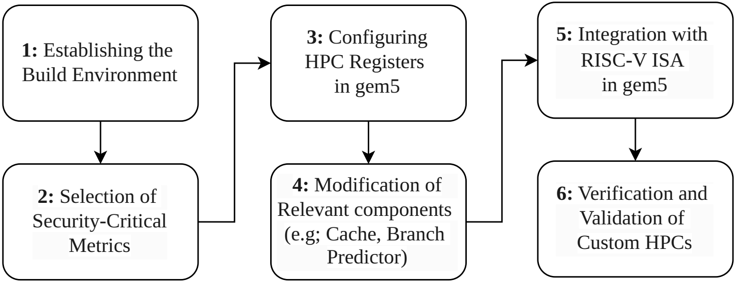
Custom HPCs within Gem5

Custom HPCs in gem5 : Motivation

Why Custom HPCs (Security-Centric)?

- gem5 currently does not have HPCs for observing cache or branch predictor misses. The metrics are important for detecting/analyzing microarchitectural side channel attacks.
- Need to create a gem5-based virtual platform with custom HPCs for branch misprediction, cache misses etc.
- Useful for attack detection, close to a real hardware scenario.

Creating Custom HPCs in gem5



1: Establishing the Build Environment

2: Selection of Security-Critical Metrics

3: Configuring HPC Registers in gem5

4: Modification of Relevant components (e.g; Cache, Branch Predictor)

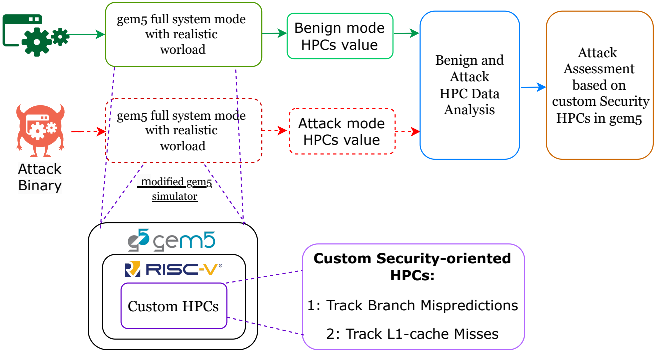
5: Integration with RISC-V ISA in gem5

6: Verification and Validation of Custom HPCs

Workflow for creating custom HPCs into gem5.

Gem5-Based Hardware Performance Counter: Methodology

A novel framework developed for attack assessment that uses gem5-based custom HPC for RISC-V security analysis.

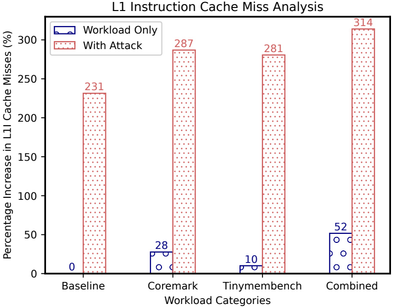

gem5 full system mode with realistic worload
Benign mode HPCs value
Attack Binary
gem5 full system mode with realistic worload
Attack mode HPCs value
Benign and Attack HPC Data Analysis
Attack Assessment based on custom Security HPCs in gem5
modified gem5 simulator
gem5
RISC-V
Custom HPCs

Custom Security-oriented HPCs:

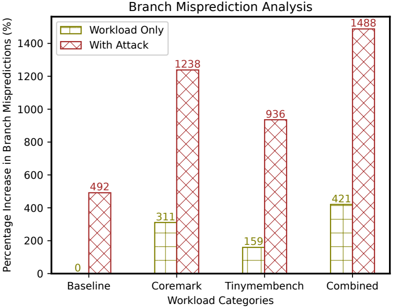
1: Track Branch Mispredictions

2: Track L1-cache Misses

HPC Analysis Results Across Various Workloads



L1 instruction cache miss analysis across various workloads (with and without Flush+Fault attack) using gem5-simulated HPCs.



Branch misprediction analysis across various workloads (with and without Flush+Fault attack) using gem5-simulated HPCs.

Machine Learning (ML)

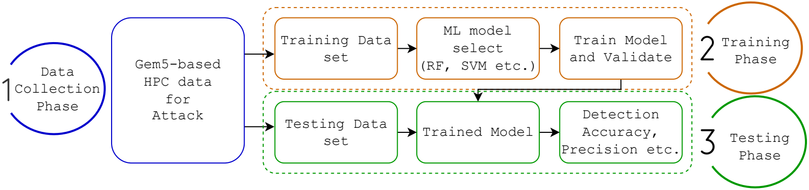
ML for attack detection

Motivation: ML for Side-Channel Detection

Why ML with gem5-based HPC traces?

- ML can learn attack vs. benign patterns automatically
- Enables virtual platform–based detection before hardware implementation
- Supports rapid prototyping of detection models

Methodology for Side-Channel Attack Detection using gem5 and ML



Methodology for HPC trace generation and ML-based detection using gem5.

ML-Based Attack Detection: Results

Evaluation Metrics:

- **Accuracy:** Proportion of correct predictions out of all predictions.
- **Precision:** Proportion of predicted attacks that were actual attacks.
- **Recall:** Proportion of actual attacks that were correctly predicted.

Model	Accuracy	Precision	Recall
RF	0.99	0.99	0.99
SVM	0.96	0.95	0.97
NB	0.95	0.92	0.96

Publications

1. Paper accepted at **SECRYPT 2025, Spain**: "Assessing Security RISC: Analyzing Flush+Fault Attack on RISC-V using gem5"
2. Paper accepted at **EICC 2025, France**: "Evaluating KASLR Break on RISC-V using gem5"
3. Paper accepted at **IOLTS 2025, Italy**: "Detection using gem5 and Machine Learning: A Case Study on Fault-based Attacks in RISC-V"
4. Paper accepted at **SAMOS 2025, Greece**: "Prototyping Custom Hardware Performance Counters in gem5 Simulator: A Framework for RISC-V Side-Channel Attack Assessment"
5. Paper accepted at **IEEE CSR HACS 2025, Greece**: "SpectreShield: Design and Analysis of Spectre Countermeasures on RISC-V Using gem5"
6. Paper accepted at **28th Euromicro Conference Series on Digital System Design (DSD), Italy**: "Evict+Spec+Time on RISC-V: Gem5-Based Implementation and Microarchitectural Analysis"

Future Work

Goal

Build a flexible RISC-V platform to evaluate microarchitectural attacks and defenses.

- **Wide Attack Coverage**
  - Support for cache attacks, speculative execution, branch prediction, and TLB-based side channels
- **ML-Based Detection Techniques**
  - Use gem5 statistics for selecting HPCs to train models for detecting attack patterns in execution traces
- **Countermeasure Evaluation**
  - Implement and test branch predictor partitioning, cache isolation, and locking
  - Measure effectiveness and performance trade-offs

Question answers

Thank You!

Contact:
mahreen.khan@telecom-paris.fr