Evaluating KASLR Break on RISC-V using gem5: Microarchitectural Side-Channel Analysis of Page-Table Walks

Mahreen Khan, Maria Mushtaq, Renaud Pacalet, Ludovic Apvrille

▶ **To cite this version:**

Mahreen Khan, Maria Mushtaq, Renaud Pacalet, Ludovic Apvrille. Evaluating KASLR Break on RISC-V using gem5: Microarchitectural Side-Channel Analysis of Page-Table Walks. EICC 2025 (European Interdisciplinary Cybersecurity Conference 2025), Jun 2025, Rennes, France. pp.229-235, ⟨10.1007/978-3-031-94855-8_15⟩. ⟨hal-05097207⟩

HAL Id: hal-05097207

https://telecom-paris.hal.science/hal-05097207v1

Submitted on 4 Jun 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

HAL Authorization

Evaluating KASLR Break on RISC-V using gem5: Microarchitectural Side-Channel Analysis of Page-Table Walks

Mahreen Khan, Maria Mushtaq, Renaud Pacalet, and Ludovic Apvrille

Telecom Paris, Institut Polytechnique de Paris, France
firstname.secondname@telecom-paris.fr

Abstract. This paper leverages the gem5 simulator to analyze a microarchitectural KASLR break on RISC-V systems. Previous research [2] demonstrated the feasibility of KASLR breaks on RISC-V hardware platforms (C906 and U74). Our paper aims to provide insights that are not easily attainable through traditional hardware experiments. By employing gem5, we gain access to fine-grained metrics such as cycle counts, cache behavior, branch prediction statistics, and TLB accesses, among others. These detailed insights give a deeper analysis of the KASLR bypass and help understand the attack mechanics better.

Keywords: Microarchitectural Security · Side-Channel Attacks · gem5 Simulator · Embedded Systems · Cache Timing Analysis.

1 Introduction

Kernel Address Space Layout Randomization (KASLR) is designed to hinder kernel exploitation by making memory layout prediction difficult. However, current RISC-V implementations employ only **coarse-grained** randomization rather than the **fine-grained** per-function randomization which is under development [3]. Instead of randomizing the locations of individual kernel functions, RISC-V KASLR merely shifts the starting position of the entire kernel image. This means that once the starting address is determined, the locations of all kernel functions can be inferred [1].

Microarchitectural side channels can expose these address mappings. The timing differences during a page table walk can indicate whether an address is mapped. For mapped addresses, the walk accesses all levels and performs permission checks. For unmapped addresses, the walk aborts early, leading to lower latency.

In our paper, we leverage the gem5 simulator [5] to emulate a RISC-V system and analyze the timing behavior of page table walks. Previous work [2] demonstrated the feasibility of KASLR breaks on hardware platforms such as the C906 and U74. However, hardware experiments do not capture detailed internal state or provide comprehensive performance metrics. The gem5 simulator supplies cycle counts, cache behavior, branch prediction data, and TLB accesses. These

metrics offer a deeper view of the attack mechanics and support a thorough analysis of the KASLR bypass on RISC-V systems.

The main contributions of this paper are:

- **Simulation-based validation of Page-table walk vs no-walk impact**: We demonstrate a working proof-of-concept for breaking RISC-V KASLR in gem5, showing a 44% timing difference in `rdcycle` counts between successful and aborted page table walks (Figure 2).
- **Microarchitectural side-channel analysis using gem5**: We provide detailed measurements of:
  - TLB behavior, showing 1.5-2× more misses during page walks (Figures 3, 4)
  - Branch prediction impacts, with 30-40% increase in mispredictions during attacks (Table 2)
  - Simulation overhead characteristics (Figures 5, 6)

The remainder of this paper is structured as follows: Section 2 details the attack methodology, explaining how page table walk timing differences can break KASLR on RISC-V systems. Section 3 describes our gem5 simulation environment, including configuration parameters and experimental setup. Section 4 presents our results, analyzing timing measurements, TLB behavior, and branch prediction metrics. Section 5 discusses potential future work, including gem5-based security enhancements and defense mechanisms. Finally, Section 6 concludes with implications of our findings and the value of simulation for microarchitectural security research.

2 Attack

This section describes the exploitation of Kernel Address Space Layout Randomization (KASLR) proof-of-concept (PoC) tested on gem5 RISC-V system. We focus on the microarchitectural side channels that can be leveraged to infer whether a kernel address is physically backed.

The attack assumes an **unprivileged** adversary who has execution capabilities in user space but no direct kernel access. Existing x86 defenses like KAISER, LAZARUS, and FLARE rely on x86-specific features and are inapplicable [2].

2.1 Exploiting Page Table Walk Timing for KASLR Breaks

A **page table** is a hierarchical data structure used by modern CPUs to translate virtual addresses into physical addresses in systems implementing virtual memory. The page table consists of multiple levels, with each level refining the translation until the final physical page frame is identified.

A **page table walk** refers to the process of navigating through the *multi-level page table hierarchy* to resolve a virtual address into a physical address. This occurs whenever a **Translation lookaside buffer (TLB) miss** happens,

requiring the memory management unit (MMU) to fetch the translation manually from the page tables stored in memory.

The duration of a page table walk depends on whether all required page table entries (PTEs) exist and are valid, leading to different timing characteristics that can be exploited for **KASLR breaks**.

- **Full Page Table Walk (Walk Case)**: When all necessary PTEs are present and valid, the MMU performs a complete page table walk, traversing all levels of the hierarchy until it resolves the address. At this stage, a permission check occurs, ensuring that the accessing process has the necessary privileges to access the memory. This process incurs a measurable delay due to multiple memory accesses.
- **Early Termination (No-Walk Case):** If a required PTE is missing (e.g., the entry is not present, marked as invalid, or leads to an access violation), the page walk aborts early. This results in a page fault, which is detected significantly faster than a full page walk.

A page walk incurs a longer delay, which will happen for the mapped addresses, while no page-table walk is faster, which indicates unmapped addresses. This time measurement can be done by using an **unprivileged** `rdcycle` RISC-V instruction. These timing differences enable attackers to **de-randomize** the kernel by detecting whether a given address is backed by a physical page. If an attacker can pinpoint the base address of the kernel image, they can effectively bypass RISC-V's KASLR implementation, exposing kernel structures to further attacks [1].

3 gem5 Simulator

The gem5 simulator is a modular, open-source framework for computer architecture research. It enables detailed modeling and simulation of hardware components, offering insights into microarchitectural interactions. This section outlines the gem5's configurations and the methodology for attack analysis.

3.1 Configurations

gem5 supports multiple instruction set architectures (ISAs), including x86, ARM, RISC-V, SPARC, and MIPS [6], making it adaptable to diverse research scenarios. Simulations range from simple single-core systems to heterogeneous multi-core architectures. Users can select between two modes: full-system (FS) mode, which models the complete operating system (OS), and syscall emulation (SE) mode, which abstracts OS interactions for faster simulations [7]. CPU models include the `AtomicSimpleCPU` for fast functional emulation, the `TimingSimpleCPU` for basic timing analysis, and the out-of-order `O3CPU` [4], which models speculative execution and pipeline dynamics. Cache hierarchies, interconnects, and memory subsystems are defined via Python scripts, enabling customization of L1/L2 caches, coherence protocols, and DRAM controllers [5].

3.2 Methodology

Fig. 1. The gem5 simulator attack analysis methodology.

The gem5 workflow for analyzing page-table walk impact for breaking KASLR on RISC-V is illustrated in Figure 1. First, the attack binary is prepared and integrated into a modified RISC-V disk image. Next, the gem5 simulator is configured via Python scripts to define hardware parameters such as the out-of-order `O3CPU` model and cache hierarchies, all running in full-system (FS) mode to capture OS interactions. The detailed configuration is given in table 1.

The attack simulation is then executed, leveraging gem5's detailed timing models to track microarchitectural behaviors. Finally, metrics including branch predictor accuracy, instructions executed and TLBs behavior are analyzed to correlate the attack's execution with observable hardware-level side effects, enabling insights into vulnerabilities.

Table 1. Gem5 configuration for attack analysis.

Parameter	**Value**
Simulation Mode	Full-System (FS)
ISA	RISC-V 64-bit
CPU Model	`O3CPU` (out-of-order)
L1 Cache	32 KB I/D
L2 Cache	256 KB
Memory	4 GB DDR4
Kernel	`riscv-bootloader-vmlinux-5.10`

The gem5 simulator has certain limitations. Its execution speed is slower than real hardware, particularly for FS-mode simulations with detailed timing models. Resource demands (memory and CPU) scale with system complexity, limiting accessibility for large-scale experiments. Despite these limitations, gem5 configurable workflow balances granular data collection with flexibility, making it an excellent tool for microarchitectural research.

4 Results

Building on prior work demonstrating KASLR breaks on RISC-V hardware [2], we leverage gem5's granular metrics such as cycle counts, and TLB behavior to

dissect attack mechanics in simulated environments. By exploiting the unprivileged `rdcycle` instruction, we measure page-table walk timing discrepancies and correlate them with microarchitectural side effects. Our analysis confirms that cycle counts, TLB misses, and simulation overheads differ significantly between mapped (Walk) and unmapped (No-Walk) scenarios, validating gem5's utility for studying KASLR vulnerabilities.

Figure 2 highlights the 44% increase in `rdcycle` count during page-table walks exposes timing variations exploitable for KASLR inference.

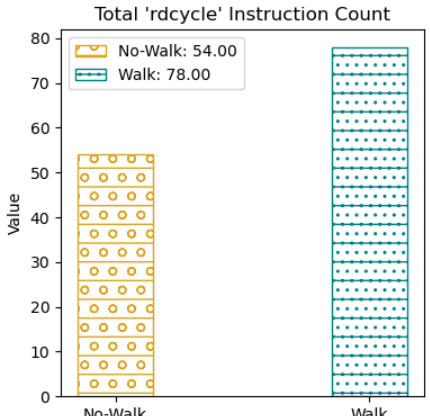

Fig. 2. `rdcycle` Counts during Page-walk and No-walk scenario.

Figures 3 and 4 illustrate increased data and instruction TLB misses in walk scenarios. This reflects address translation overheads that amplify timing leaks.

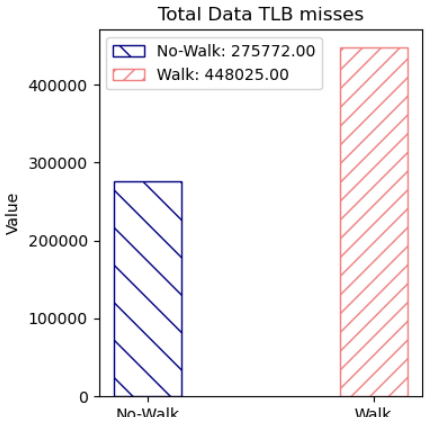

Fig. 3. Data TLB misses during Page-walk and No-walk scenario.

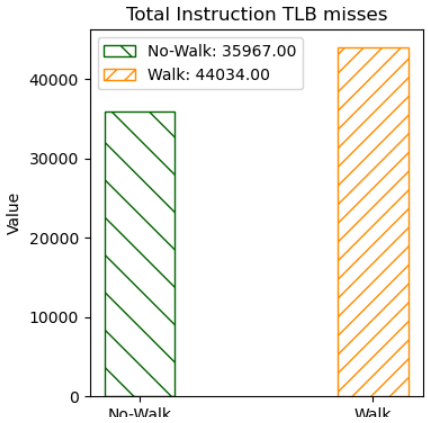

Fig. 4. Instr. TLB misses during Page-walk and No-walk scenario.

Figure 5 and 6 quantifies gem5's execution costs. Walk scenarios require longer simulation time and more instructions, reflecting the computational burden of page-table walks. These metrics validate gem5's ability to model real-world timing dynamics critical for side-channel analysis.

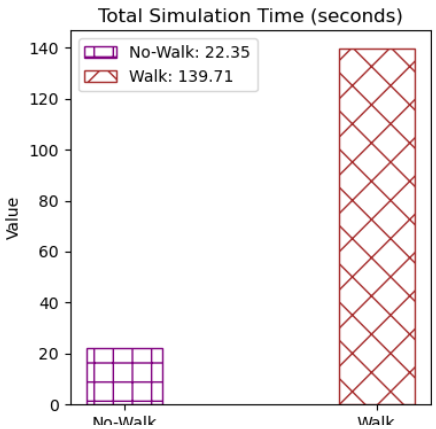

Fig. 5. Total Simulation Time.

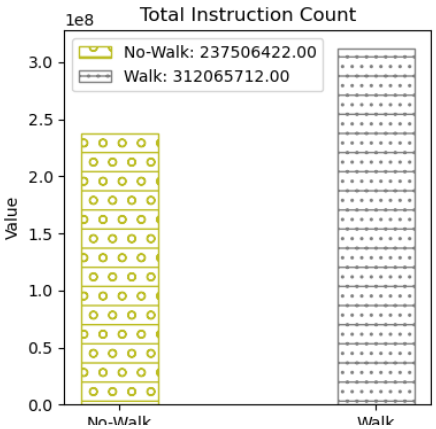

Fig. 6. Total Instruction Count.

Table 2 details branch predictor behavior. Walk scenarios experience an increase in branch prediction metrics. These metrics highlight Gem5's capacity to model complex microarchitectural interactions beyond timing.

Table 2. Comparison of Branch Predictor Metrics for Walk vs No-walk.

#	Metric Name	No Page-table Walk	Page-table Walk
1	Branch Predictor lookups (Count)	67621535	87952408
2	Squashed branches (Count)	395515	777397
3	Corrected branches (Count)	3466976	4795993
4	Committed branches (Count)	44251757	55011904
5	Mispredicted branches (Count)	2297497	3245251

5 Future Work

Future research can focus on advancing gem5 as a dedicated security platform. This involves creating enhanced visualization and tracing tools within gem5, enabling security researchers to observe and analyze microarchitectural events

with greater granularity. Further enhancements could be to develop a vulnerability assessment framework leveraging gem5-simulated hardware performance counters, thereby providing an environment for refining new defenses. Potential countermeasures include finer-grained KASLR (e.g., per-function randomization), and improved cache and TLB randomization strategies. With gem5's reconfigurability and the repeatability of experiments, these countermeasures can be thoroughly tested, ultimately leading to more effective protective measures.

6 Conclusion

KASLR is a widely used defense mechanism that randomizes the kernel's base address to prevent reliable exploitation, but current RISC-V implementations lack dedicated countermeasures and rely on coarse-grained randomization. As shown by our gem5 metrics and prior work by Gerlach [2], attackers can exploit timing variations during page walks to bypass KASLR on RISC-V CPUs using via the `rdcycle` instruction. Unlike hardware platforms, which do not provide detailed microarchitectural insights, the gem5 simulator enables comprehensive analysis, making it a valuable tool for evaluating attacks and testing countermeasures in a controlled environment. In addition to `rdcycle` count, gem5 offers detailed metrics such as instruction counts, cache behavior, and TLB access patterns, facilitating a deeper understanding of attack mechanics.

References

1. Canella, C., Schwarz, M., Haubenwallner, M., Schwarzl, M., Gruss, D.: Kaslr: Break it, fix it, repeat. In: Proceedings of the 15th ACM ACCS (2020)
2. Gerlach, L., Weber, D., Zhang, R., Schwarz, M.: A security risc: microarchitectural attacks on hardware risc-v cpus. In: 2023 IEEE S&P. IEEE (2023)
3. Larabel, M.: Fgkaslr patches revised a 10th time for improving linux kernel security (2022), https://www.phoronix.com/news/FGKASLR-Linux-v10
4. Li, J., Tufte: Out-of-order processing: a new architecture for high-performance stream systems. Proceedings of the VLDB Endowment **1**(1) (2008)
5. Lowe-Power, J.: Gem5 doc (2024), https://www.gem5.org/documentation/
6. Lowe-Power, J., Ahmad, A.M., Akram: The gem5 simulator. arXiv (2020)
7. Ta, T., Cheng, L., Batten, C.: Simulating multi-core risc-v systems in gem5. In: Workshop on Computer Architecture Research with RISC-V (2018)