Paper - SCALE: Secure and Scalable Cache Partitioning

Authors-Nadja Ramhoj Holtryd, Madhavan Manivannan, and Per Stenström

Venue: IEEE International Symposium on Hardware Oriented Security a Trust (HOST), 2023

1. Problem Addressed - This paper deals with the security vulnerability that is caused by dynamically partitioned last level caches (LLCs) in multi-core processors. In more detail, it discusses how predictable cache allocation policies become a side-channel that enables attackers to infer sensitive information based on observable changes in cache allocation.

2. Motivation for the Problem - In fact ,utilizing dynamically partitioned caches for improving performance nowadays maximizes the chance of executing timing-based side-channel attacks. For example ,an attack could be launched based on predictable changes in the allocation of the cache to leak sensitive information. So ,it is of great importance to develop methods capable enough to ensure performance without compromising security.

3. Motivation for the Solution -Early secure cache partitioning techniques are either static or, otherwise ,still lagging behind in getting the right efficiency-security trade-off. The problem necessitated a dynamic and scalable solution that effectively embodies randomness into cache allocation decisions .The applicability of non-determinism and insights into differential privacy guided the development of SCALE toward securing cache allocation policies from side-channel attacks with safe performance.

4. Details of the Solution Proposed -

- **Theory**:To manage cache partitioning security issues, the randomness-based solution involves randomization within the allocation of caches to guard against side-channel attacks. The key constituents are as follows:

Heuristics (Randomized Cache Allocation Process):

- **Utility-Based Cache Partitioning (UCP)**:It extends the existing UCP that traditionally depends on utility-guided cache allocation based on the efficiency of the application's use of cache resources. The UCP technique widely applies optimization of cache performance of different workloads.
- **Randomization for Security**: To counteract side-channel attacks (like conflict-based or occupancy-based), the solution adds random elements:
    - **Noise via Laplace Distribution**: This technique introduces noise in the cache allocations on the basis of the Laplace distribution. It is implemented such that it masks the usage pattern within the caches that otherwise could have been exploited by an attacker. It makes it difficult for the attackers to trace the exact location of requests by altering the pattern of access through noisy cache allocations.
    - **Randomized Change Rates**: In this system, the rate of change of allocations in the cache is randomized. This makes the system unpredictable at an additional level; it further secures the system since it's much more difficult for the attackers to predict when a cache is going to reallocate or change.
    - **Variable Reconfiguration Periods**: The policy introduces randomness in the timing of the reconfigurations. This means cache partitions are reallocated at non-regular periods. This irregularity further obfuscates the usage patterns that side-channel attackers could try to analyze.
    - **Combined Defenses**:By integrating noise ,change rate and reconfiguration randomization ,SCALE balances security with cache utilization.
    - **Configurability**:Allows adjustments to noise and change parameters based on security and performance needs.
    - **Secure and Non-Secure Applications**:Supports both secure and non-secure apps ,ensuring secure applications cache allocations remain protected.

Here, **heuristics** are used to decide about optimal noise levels (how much randomization to add), the rate of change (how frequently do cache allocations change), and the reconfiguration period (how frequently is cache reallocated) based on system requirements. These decisions balance performance, security, and fairness.

- **The SCALE Secure Enforcement Mechanism** uses the DELTA tile -based cache mapping system to achieve improved security through the following adaptations :
    - **Domain ID Tagging**: Tags cache requests with a domain ID ,ensuring secure processes only access their designated cache ways and preventing cross-domain data leaks .

    - **Replacement Metadata Isolation**:It replaces metadata using isolated LRU state or random replacement policies that prevent metadata leakage.
    - **Cache Coherence**: it assigns the same domain ID to threads of multi-threaded applications for maintaining coherence ,and duplicates cache lines for cross-domain shared data.
    - **Context Switches**:It maintains domain-specific cache mappings even across context switches by keeping CBT info in the kernel .

These measures secure against cache-based side-channel attacks while maintaining performance.

- **Implementation**:

  Scale is implemented and evaluated using the Sniper simulator. It models a 16-core tiled chip multi-processor (CMP) architecture.

  Dataset: The suite of benchmarks from SPEC CPU2006 was used for performance evaluation.

  <u>Metrics of Comparison</u>

  Compare this work with other security-aware cache solutions: SecDCP, OPTIMUS, UCP, and ScatterCache. The comparison is done with the baseline of an unpartitioned static non-uniform cache architecture of SNUCA.

  Metrics: The performance, normalized for weighted speedup, is measured against the SNUCA base. It measures the total performance of the system.

<u>5. Critique</u> -

- **Novelty**: SCALE offers Randomness in cache allocation and leverages differential privacy for security guarantees. This itself is a big further step than previous approaches, which provided their guarantees based on pure randomization or static isolation.
- **Strengths and Weaknesses**:
    - **Strengths**:

        - SCALE, improved performance by as much as 39% and 14% on average, over state-of-the-art secure caches.
        - Differential privacy comes along with strong theoretical security guarantees.
    - **Weaknesses**:
        - The gain in performance varies along with the configuration parameters, in some sets it is low.
        - The complexity of this system may take some time to deploy or install in other existing systems.
- **Reasons on Outliers**:
    - Performance variations in those results, most especially configuration differences in some of those, will hence be due to typical design trade-offs: between security and performance.
- **Possible Improvements**:
    - Further optimization of the configuration parameters could enhance performance consistency.
    - Ease of configuration and deployment would make it more pragmatic to use.

6. Related Works -

Picking Related Works:

On basis of the interest criteria of both cache security and partitioning ,the selected related works were chosen.The authors compared studies using both **quantitative** and **qualitative** methods:

- **Quantitative Comparison**:Compares the performance metrics of cache hit rates, latency ,throughput and resistance to side-channel attacks .This allowed assessment of these techniques based on efficiency and security impact.
- **Qualitative Comparison**:Compare the methodologies and frameworks ,particularly in aspects of theory-based approaches ,strategies for implementation and their implications on system security.

Influential Related Works and Recent Impactful Works:

ScatterCache and Jumanji are most effective for those who develop randomized cache set designs and dynamic NUCA strategies.

MIRAGE is a practical fully associative cache design to mitigate conflict-based cache attacks.

These works inform the current research by addressing key advancements and ongoing challenges with respect to cache security and partitioning.