Talk: Blurring Enclave Page Accesses in Space and Time with Compile-Time Instrumentation

Daan Vanoverloop[1], Andrés Sánchez[2,4], Victor Bullynck[1], Flavio Toffalini[2,3], Frank Piessens[1], Mathias Payer[2], Jo Van Bulck[1]

[1] DistriNet, KU Leuven [2] EPFL [3] RUB [4] Amazon**

Abstract. This talk overviews our recent work on TLBlur, a novel approach that leverages compiler instrumentation and the recent AEX-Notify hardware extension in modern, off-the-shelf Intel SGX processors to limit the bandwidth of controlled-channel attacks at acceptable performance overhead. We also discuss ongoing efforts to reduce instrumentation costs through binary layout optimizations.

1 Background and Problem Statement

Intel SGX safeguards application compartments, called *enclaves*, from unauthorized access, including the privileged OS. However, SGX's strong root adversary model has led to new side channels called *controlled-channel* attacks [4]. By strategically revoking access rights to enclave pages, privileged adversaries can trace enclave memory accesses at noiseless, 4 KiB spatial resolution. This page-fault channel has proven highly expressive, enabling the reconstruction of confidential text and JPEG images [4, 2].

This talk introduces TLBlur [3], the first comprehensive, fully automated solution for mitigating controlled-channel attacks on Intel SGX platforms, addressing both temporal and spatial bandwidth by prefetching recently accessed pages into the Translation Lookaside Buffer (TLB). The talk covers the background of controlled-channel attacks, TLBlur's design and the evaluation of the approach. Moreover, we will discuss additional follow-up work and future directions.

2 TLBlur: Compile-Time and Binary Instrumentation

Figure 1 gives an overview of TLBlur during runtime. TLBlur uses both compile-time and binary instrumentation to trace page accesses and maintain a software-based Page Access Map (PAM) data structure ① during runtime of the enclave application. Upon interrupt, the TLB is flushed, ② and the enclave will be resumed at the AEX-Notify [1] exception handler, which parses PAM and prefetches ③ the N most recent pages into the TLB, such that any future accesses ④ to these pages remain oblivious.

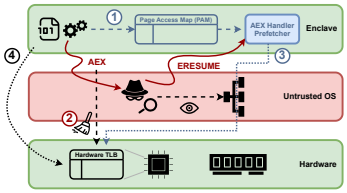

Fig. 1: Overview of TLBlur.

** All the contributions of this author were made prior to joining Amazon.

Experimental Evaluation. In terms of performance, related work by Shinde et al. reports an average slowdown of $\times 700$ for the fully-automated (non-optimized) instrumentation accross a variety of cryptographic libraries (cf. Table 2 in [2]). In contrast, our evaluation shows an average slowdown of $\times 2.8$ accross a variety of cryptographic libraries and libjpeg. This result positions TLBlur as a more practical approach for off-the-shelf Intel SGX platforms.

3 Remaining Challenges and Ongoing Work

Optimal Page Placement. In our ongoing work, we found that strategically adjusting the page layout can reduce cross-page control-flow transfers, thereby decreasing the number of instrumentation calls. We created a call graph and used clustering methods to group functions on pages. As future work, we plan to explore profile-guided optimization to assign weights to edges based on the frequency of function calls.

Efficient Constant-Time Sorting. Before resuming an enclave after an interrupt, we select the N most frequently accessed pages from the PAM to prefetch. However, to prevent leaking information about the page order in the PAM, this sorting operation must be performed in constant time. Consequently, this operation introduces significant performance overhead. We will discuss ideas to further reduce this overhead, for example, by making use of vector extensions.

4 Outlook and Lessons Learned

TLBlur demonstrates that the AEX-Notify hardware extension [1] can effectively mitigate controlled-channel attacks when combined with compiler instrumentation. This approach enables enclaves to dynamically determine their working set of pages, which is then used to blur the traceability of page access patterns by prefetching the set into the hardware TLB before resuming execution after an interrupt. However, compiler-based instrumentation introduces its own challenges in minimizing performance overhead. There remain numerous opportunities for optimization, such as improving code layout, hoisting data instrumentation out of performance-critical loops, or more efficiently selecting recently accessed pages from the PAM.

Acknowledgements

This research is partially funded by the Research Fund KU Leuven, the Research Foundation – Flanders (FWO) via grant #1261222N, and by the Cybersecurity Research Program Flanders.

References

1. Scott Constable, Jo Van Bulck, Xiang Cheng, Yuan Xiao, Cedric Xing, Ilya Alexandrovich, Taesoo Kim, Frank Piessens, Mona Vij, and Mark Silberstein. AEX-Notify: Thwarting precise single-stepping attacks through interrupt awareness for intel sgx enclaves. In *USENIX Security 23*, August 2023.
2. Shweta Shinde, Zheng Leong Chua, Viswesh Narayanan, and Prateek Saxena. Preventing page faults from telling your secrets. In *ASIA CCS'16*, 2016.
3. Daan Vanoverloop, Andres Sanchez, Flavio Toffalini, Frank Piessens, Mathias Payer, and Jo Van Bulck. TLBlur: Compiler-assisted automated hardening against controlled channels on off-the-shelf Intel SGX platforms. In *USENIX Security 25*, 2025.
4. Yuanzhong Xu, Weidong Cui, and Marcus Peinado. Controlled-channel attacks: Deterministic side channels for untrusted operating systems. In *IEEE S&P'15*, 2015.