When Oblivious is Not: Attacks against OPAM

WOOT’20@USENIX-SECURITY

Nirjhar Roy (Indian Institute of Technology - Kanpur)
Nikhil Bansal (Indian Institute of Technology - Kanpur)
Gourav Takhar (Indian Institute of Technology - Kanpur)
Nikhil Mittal (Fortanix Inc)
Pramod Subramanyan (Indian Institute of Technology - Kanpur)
Presentation Roadmap

- Introduction
- Attacks on InvisiPage/OPAM
- Covert Channels using Reuse Distances and its evaluation
- Conclusion
Enclaves Demystified

Enclaves: hardware-supported environment for isolated execution with strong application-level security guarantees despite the presence of malicious/compromised privileged software
Introducing ORAMs

- **Interface** between a client and an untrusted server
- Shuffles the data from time to time
- Hides access patterns and access frequencies
- Examples: Square root ORAM, Tree-based ORAMs including Path ORAM, Ring ORAM, etc.
ORAM Meets Demand Paging and Enclaves

Threat Model

- The Host OS/apps are considered malicious trying to find out access pattern/access frequency/memory content of the pages being read or written.
- The OS observes only a random set of pages (encrypted) getting read/written after in step 3.
- The attacker can choose to tamper the pages but that will detected after step 6 in Runtime.
- Attackers having physical access to the memory will also see cipher text.
Our Contributions

- Discovering vulnerability in InvisiPage
- Implementation of a demand paging system inside Keystone
- Exploiting it to design new attacks:-
  - The reuse distance attacks
  - The level tracking attack
- Designing a covert channel using Reuse Distances
Attacks on InvisiPage

Introduction to Invisipage/OPAM

Metadata Tree

Data Tree

All dec + auth and checked

Dec + auth

Access (88, Fetch)

Position map

Updated Position map

ORAM Path Read

468 → 0, … 88→ 2, 250→ 2, 100 → 2

468 → 0, … 88→ 0, 250→ 2, 100 → 2
Introduction to Invisipage/OPAM

Access (88, Fetch)

Metadata Tree

Data Tree

468 → 0, … 88→ 0, 250→ 2, 100 → 2

Updated Position map

ORAM Path Write/Shuffle
Vulnerability in OPAM (Invisipage)

- On every page fault or ORAM access exactly one page gets transferred.

- The adversary is able to observe which page got exchanged.

- Transferred page is the page of interest and is definitely NOT a dummy page.

- Adversary can calculate number of intervening ORAM accesses.

- This in fact leaks information and makes OPAM access not oblivious.
Introducing Reuse Distance Attack

- Reuse Distance: \# of faults between the time a page gets evicted and when the page is brought back to the enclave (i.e., reused).

- This sequence of reuse distances will be different for different types of memory accesses/applications.

- We use this fact to distinguish and predict/identify the secret applications running inside the enclave.
Example of Reuse Distance Attack

- Enclave has 2 physical pages available and LRU is used.
- In Figure (a) every page is reused after 2 page faults and
- In Figure (b), the reuse distance of the root is 2 because the root node is accessed in every iteration and for non-root pages are multiples of 2 because non-root pages may or may not be accessed in successive iterations.
Attack Methodology

Training

● Collect trace of reuse distances for many apps on many inputs
● Train CNN sequence classifier on these
● Classes are the different applications

Testing

● Run app on a new input never seen before
● Measure classification accuracy
Methodology

- Execute with many (~100-200) inputs and collecte reuse distances traces
- Data divided into training and test in 3:1 ratio and evaluation repeated 10 times
- Reuse distance trace is used as the input feature
- Random splits of the data into training and test datasets
Covert Channels Using Reuse Distances
Basic Idea

- Reuse distance leakage provides a covert channel to leak secret information (e.g. an input genome data).
- Engineering the access patterns to cause a particular sequence of page faults and associated reuse distances.
- Interpret the reuse distances to leak the bits.
Threat Model

- Standard enclave threat model corresponding to a software attacker
- Enclave RT and the hardware platform are trusted and we do not use microarchitectural side-channels and/or HW access to DRAM
- Enclave app colludes with host OS to leak sensitive input data
- Host OS is aware of the encoding used by the enclave application
Example of an Encoding With Reuse Distance

- Application wants to transmit a message 1001, $n = 4$ and $k = 2$
- Page replacement policy is FIFO and enclave has $P = 4$ pages
- To transmit a bit 1, reuse distance in range $[8, 16)$ (Pages 1-8)
- To transmit a bit 0, reuse distance in the range $[0, 8)$ (Pages 9-16)
- Generate reuse distance sequence $(12, 5, 5, 14)$ corresponding to message 1001
Bit Leakage Bandwidth Analysis

- We see a peak bandwidth with arity 4
- As we increase k, more data is transmitted with each page fault, but the number of page-faults required to setup the algorithm also increases and the overheads associated with increased number of initial page faults dominate and we see a steady decline in transmission bandwidth.
Conclusions

- Introduction of a new side channel attack, The Reuse Distance attack, which is able to infer confidential information about an enclave’s execution
- Introduction of a new covert channel using reuse distances
- Found and systematically exploited a vulnerability in state-of-the-art approach to secure demand paging enclave (Invisipage/OPAM)
In Memory of
Dr. Pramod Subramanyan

8th June 1984 - 8th July 2020
Thank you