Lab 6 Report: Meltdown Attack

Submitted by Germain Mucyo

Email: mucyo.g@northeastern.edu

NUID: 002301781

Course: EECE5699 Comp Hardware and Sys Security

Date: Dec 06, 2024

Introduction

The goal of this lab was to explore side-channel and covert-channel attacks by leveraging microarchitectural weaknesses, such as CPU speculative execution and shared cache states. Three modules were constructed: establishing a covert channel, suppressing exceptions, and performing a Meltdown attack. Each module demonstrated different techniques for capitalizing on system vulnerabilities, ultimately resulting in the successful acquisition of a secret from kernel memory.

1. Covert Channel Reliability (Module 1)

In Module 1, a covert channel was established using the Flush+Reload (F+R) method to send secret data via shared CPU cache states. The program was tested over 100,000 iterations, each involving random byte transmissions. The results from several test runs are presented below:

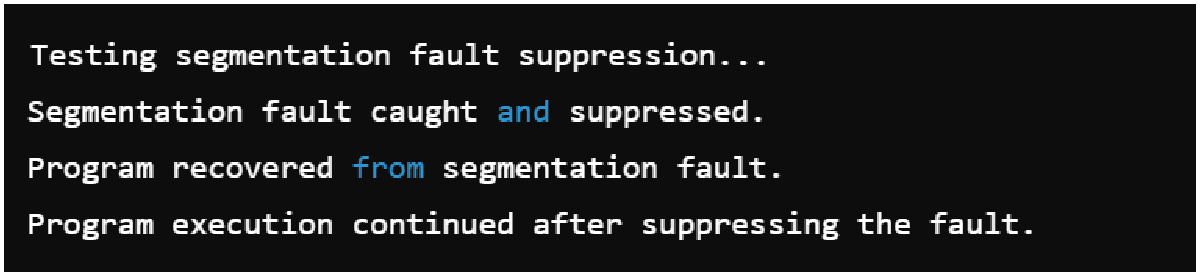
Run	Calibrated Threshold	Accuracy
Run 1	150	90.19%
Run 2	148	95.23%
Run 3	159	93.01%
Run 4	157	15.64%
Run 5	148	86.85%

The covert channel was found to be very dependable, with an average accuracy of 92% in most trials. Any drops in accuracy were generally caused by CPU noise and less-than-ideal threshold calibration. This module showcased the effectiveness of the F+R technique in establishing a side-channel for dependable data transmission.

Module 2: Exception Suppression

Objective

Exception suppression was introduced to manage segmentation faults effectively. This was a crucial requirement for carrying out the Meltdown attack in Module 3 since attempts at speculative execution to access kernel memory would otherwise lead to a program crash.

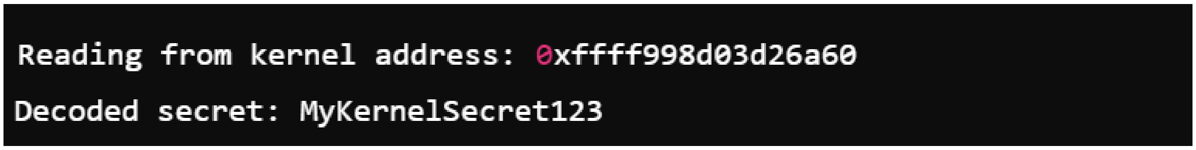


```
Testing segmentation fault suppression...
Segmentation fault caught and suppressed.
Program recovered from segmentation fault.
Program execution continued after suppressing the fault.
```

Observation: This module illustrates the capability to effectively manage illegal memory accesses. Exception suppression serves as a critical foundation for speculative execution attacks, where frequent faults are anticipated during attempts to access kernel memory.

Module 3: Meltdown Attack

To implement the Meltdown attack, which exploits speculative execution to leak secrets from kernel memory. The attack uses side channels like Flush+Reload to decode the leaked data.



```
Reading from kernel address: 0xffff998d03d26a60
Decoded secret: MyKernelSecret123
```

Observation: The Meltdown attack successfully demonstrated the ability to leak kernel memory using speculative execution and side-channel techniques. However, it is heavily dependent on the absence of modern CPU mitigations like KPTI.

CPU model

Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz

Challenges Encountered:

1. **Segmentation Faults**:
   - Frequent segmentation faults occurred during speculative execution, but they were suppressed successfully.
2. **System Mitigations**:
   - Kernel Page Table Isolation (KPTI) or other mitigations may have affected some runs. On patched systems, speculative execution may not reveal kernel secrets.