Spectre & Meltdown

Speculative execution - performance hack or a massive vulnerability?
CS4028 Security: In-course assessment

Konrad Dryja
51552177
k.dryja.15@abdn.ac.uk
University of Aberdeen

October 10, 2019

Contents

Abstract

As part of my CS4028 assessment, I would like to delve into the Spectre & Meltdown vulnerabilities. I recognize that those are not offensive technology or tool per se, although both of them expose a major flaw which could easily be applied in one of them. I found researching this subject very interesting, as it taught me a lot about low-level CPU operations and how speculative execution can be used in nefarious ways.

1 Introduction

The computing and security environments were shook in January 2018 when Google's Project Zero collaborating with security researchers discovered a major flaw in almost all consumer CPU. The hole allowed potentially a piece of code with no special privileges to read memory outside its own sandboxed environment (or even reading privileged parts of kernel memory). Day-to-day users perhaps wouldn't be impacted to a tremendous extent (it is claimed that until the flaw was discovered, no malicious use was detected [SOURCE]), but imagine running a production server in GCP, AWS or other major cloud provider - and having other customers of Amazon being able to access your data from their own, virtualised containers.

Figure 1: Graphic for Meltdown on the left, Spectre on the right.

As a word of introduction, the first sighting consisted of Spectre variant 1 & 2, which was allocated Common Vulnerabilities and Exposures ID codes of CVE-2017-5753 [SOURCE] and CVE-2017-5715 [SOURCE] along with variant 3, which was dubbed Meltdown, with code CVE-2017-5754 [SOURCE].

2 Background

In order to understand how the vulnerabilities work, it is necessary to introduce some concepts and ideas that CPU manufacturers opted in for. In short, the biggest problems relate to Out of Order Execution (OOE), Branch Prediction (BP) and Branch Target Buffer (BTB) - all of those