Matthew Michaelis, CSCI 5593

Report #1, February 5th, 2019

**Literature Review of “****Measuring the Impact of Spectre and Meltdown”**

This paper primarily focuses on the long lasting effects of the Spectre/Meltdown vulnerabilities regarding speculative execution on modern CPUs. These two bugs work on similar principles, abusing the speculative systems either by searching though the shared cache for predictions that were made that may contain sensitive user data, or by using the speculation itself to deduce user patterns or data. Because these are hardware level applications, to mitigate this risk our only course of action for existing CPUs is to disable the features entirely. This causes a substantial performance hit, as the speculative execution works very well to improve performance. In newer generations, Intel has taken steps to remove the ability for users to access speculatively generated data by putting it in a protected cache (as AMD has done for years), but the existing generations will simply have to go without. The patches themselves, though, are indiscriminate. So even computers that do not handle any sensitive data can no longer use the speculation hardware if they want to update at any point.

The paper finds that the performance impact can slow down operations from 10%-90%(In worst case) depending on the operation, and how much it benefits from speculative execution. For instance, if you know a disk access is going to occur, you can poll the disk before the instruction actually comes through, effectively increasing the speed of the access from CPU→Disk→CPU to CPU→L2 Cache→CPU. This is the same concept for any external query (Network connections, memory access, etc.) When speculative execution is disabled, we lose a lot of performance because the CPU has to wait for the instruction to come through before it’s able to execute code.

These findings represent the dangers in speculative execution, or similar advanced pipeline prediction systems that rely on previously processed data to predict the future, when improperly implemented. At a hardware level, the CPU doesn’t know the difference between a private key for a bitcoin wallet, and a random string of integers. If we don’t treat all data as potentially sensitive, even if the sensitive data is merely a needle in the haystack, we risk exposing it to people (or programs) that know exactly what they’re looking for in the shared cache. Intel could also adopt AMD’s strategy of having a protected cache for speculation, but that doesn’t solve the problem of harvesting data through monitoring what gets executed speculatively. This paper shows very well that the concept of speculative execution is good. It works really well, and allows for much higher speeds than without, but we need to rethink how we handle this data at an architectural level. We also need to design CPUs not only for speed, but for security as well.

**Reference**

1. A. Prout et al., "Measuring the Impact of Spectre and Meltdown," 2018 IEEE High Performance extreme Computing Conference (HPEC), Waltham, MA, 2018, pp. 1-5.

doi: 10.1109/HPEC.2018.8547554

URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8547554&isnumber=8547513