Effects of Spectre and Meltdown on the Internet of Things

Author
Computer Science and Engineering
Tandon School of Engineering
New York University
Brooklyn, NY
Email: author@nyu.edu

***Abstract*—There are many threats that exist in the computing world that effect the data that exists on personal computers, high performance computing clusters, and virtualized systems. A lot of these threats come in the form of software vulnerabilities that are exploited and used to exfiltrate data from a system. The general public mostly only hears of software vulnerabilities, but as seen with Spectre/Meltdown, the need to look at hardware vulnerabilities is apparent. Due to complex nature of manufacturing and the expansive supply chain of computing component development, ensuring the safety and security of the hardware that an operating system runs on can be a monumental task. Therefore, it is imperative that operating systems be able to run securely even on untrusted hardware. This will be paramount in the security of the internet of things (IOT) where relatively cheap hardware is produced at astonishing rates where time for evaluation is diminished. Being able to run secure kernels on these devices can help protect the often-sensitive data that IOT devices collect and will help safeguard the future of computing**

I. Introduction

The discovery of the Meltdown and Spectre vulnerabilities in most modern processors has been detrimental to the reliability of the security of these hardware components. The current workarounds to these vulnerabilities involve operating system (OS) patches and the use of special bound-checking functions for implementation in software programs that run on the systems [5]. Spectre, being a physical flaw in the way processors handle speculative processing, is much harder to patch via software and can only be fully rectified using hardware solutions. There has been research into comparing results from desktop and server systems pre-patch and post-patch, but not a lot of attention on the internet of things (IOT) devices and how those are being secured along with the effect of the patches on performance. Many IOT devices run on ARM processors, which are vulnerable to Spectre [1]. It has been discovered that there is performance degradation on systems with these workarounds and the same is true for IOT devices. This research uses an emulated ARM processor that is utilized in many IOT platforms and which is vulnerable, in order to test the performance of the device both pre-patch and post-patch and verify if the performance degradation is similar in this class of devices. This will demonstrate how difficult it can be to patch a large number of mostly unmanaged devices and what the effects of the patch can have on deterring vendors and users from patching these devices, thus perpetuating the cycle of insecure IOT devices.

II. Related Research

In the article written by P. Deb[2], the author discusses different metrics that previous research has done in the area of Spectre and Meltdown mitigation effects on high performance computing (HPC) clusters and servers. The author mentions that consumer grade workstations have not been studied as close and attempts to generate metrics to help show those effects. The article lays out several different metrics for consumer-grade computers and several tests to help measure their performance before and after operating system patches are applied to the test device. My research adopts similar tests and methodology but applies to a narrower set of systems that is specifically geared toward the internet of things (IOT) devices.

In the article written by A. Prout, et al.[7], the authors discuss the various variants of the Spectre and Meltdown vulnerabilities as they relate to HPC clusters. The article focuses on the MIT Lincoln Laboratory Supercomputing Center HPC platform and its performance before and after operating system kernel patches for network connection establishment, disk access, and computationally intensive MATLAB programs. My research focuses on more consumer-oriented statistics like graphics processing, CPU speed, disk I/O, and RAM speed. The focus on my research is also on consumer and enterprise grade IOT devices and their performance hit.

In the article written by M. Löw[6], the author discusses the different systems, processors, and operating systems that are affected by the Spectre and Meltdown vulnerabilities. The research collates available information on patches and performance metrics for the various effected products. The article also mentions the variants of the vulnerabilities and how they break confidentiality of the system. While ARM processors are discussed and a list of the effected processors, my research takes a closer look at IOT devices that use ARM processors and collects more specific metrics on what performance hits a device would take after receiving a kernel patch for the system to mitigate the threat posed by the vulnerabilities. My research uses the information to help decide what architecture to test and what to look for.

III. Empirical Evidence

The purpose of this research is to investigate alternative methods to secure IOT devices from Spectre and Meltdown to reduce the effect of the operating system patches on the performance of the devices. Since these devices usually run on minimal hardware, it is beneficial to be able to protect the systems from the vulnerabilities posed while still allowing the systems to run at their full performance. The research is focused on systems that use ARM processors and minimal hardware. Three performance areas, RAM, CPU, and a running a basic program[4] with and without a mitigation, are tested. The test environment for this research will be a device using an ARM64 processor that is emulated using QEMU. A vulnerable ARM processor, a dual-core Cortex A57, is running a vulnerable version of Linux, specifically Ubuntu 16.04 [8], [3]. The comparison device will be a similar virtualized system with a patched Linux operating system, specifically Ubuntu 16.04.7 [3]. The operating system was checked for vulnerabilities to Spectre/Meltdown using a tool referenced in other materials. The vulnerabilities themselves were not confirmed in this test. The virtualized system will have 1 GB of RAM and an 8 GB disk. This configuration is a reasonable setup for what might be commonly used for an IOT product. The CPU and RAM performance is measured using the Sysbench tool. The output from the tool for the vulnerable and patched operating systems can be seen in Tables I,II, III, and IV.

Table I shows the Sysbench simple CPU test and contains the results for both systems. The average execution time and data transfer rate decreases slightly from the unpatched system to the patched system. That same trend holds for the read and write memory tests using Sysbench for a 1 GB transfer using 1 KB blocks as seen in Tables II and III. The custom program created to test the performance of the speculation barrier mitigation macros was a simple C-program that takes in the number of rounds to measure and whether to use the mitigation macro. The program simply times the computer for putting 10000 numbers in an array.

From the data, it can be seen that there is a slight performance hit from the unpatched Spectre/Meltdown vulnerable OS to the patched OS. While the patched OS has an updated kernel, it is apparent that there is a marginal difference between the two systems in these tests. Interestingly, there was a performance increase from the unpatched system to the patched system when running the custom benchmark test that incorporates the speculation barrier macro function created by Arm. There is a slight increase in time when applying macro function to an array assignment from the custom test code as well which does make sense, however the amount of time could be irrelevant depending on the application that the system is utilizing.

IV. Future Work

There is plenty of room for improvement on the research performed. Utilizing virtual machines and a very simple custom performance monitoring program help to give a proof of concept to the idea that Arm-systems were not immune to the performance degradation that other processors experienced from the Spectre/Meltdown patches. A further look at more common IOT devices and running tests directly with them would help determine how the performance was affected by devices in an commercial setting. Increasing the number of devices also helps with sample size and narrows the margin of error. Looking at other versions of the ARM processors and at other common speculative processors used in the IOT market would also help to increase the pool of data in order to have a greater understanding of what is really happening with these systems. There could be real world consequences for upgrading these systems and there could be some for not upgrading as well. Giving consumers of the devices the most accurate data is always important so sound decisions are made should be the most important aspect of this reseearch.

V. Conclusion

It is difficult to say with certainty based on the small sample size how much this data applies to operations on a large scale, but from the minimal testing, there is certainly some cause for concern. The patched system takes a bit of a performance hit from the results of the performance testing tools, but appears to run the custom application in a faster time on average. This could be due to a number of reasons and could be well within a margin of error. Even with the caveats, taking the data as is still shows that there could be quite some performance loss with patching an ARM system for Spectre/Meltdown, but the macros to mitigate the speculation barrier might be a good option for developers and industries who cannot and should not upgrade their IOT devices. This solution would prevent the performance degradation while still protecting the system from Spectre/Meltdown.

References

[1] Arm Ltd., ”Speculative Processor Vulnerability – Arm Developer”, Arm Developer, 2021. [Online]. Available: https://developer.arm.com/support/arm-security-updates/speculative-processor-vulnerability. [Accessed: 20- Jun- 2021]

[2] P. Deb, ”An Analysis on Effects after Mitigating Meltdown and Spectre Vulnerabilities”, DAFFODIL INTERNATIONAL UNIVERSITY, 2018 [Online]. Available: http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/2568/P11684

[3] ”DebianSecurity/SpectreMeltdown - Debian Wiki”, Wiki.debian.org, 2021. [Online]. Available: https://wiki.debian.org/DebianSecurity/SpectreMeltdown. [Accessed: 26- Jul- 2021].

[4] Fonyi, Shane. CS-GY6233 System Benchmark Tool. https://github.com/fonyi/CSGY6233-Project, 2021

[5] J. Forissier and J. Greenhalgh, Speculation Barrier. https://github.com/ARM-software/speculation-barrier: ARM Software, 2018.

[6] M. Löw, ”Overview of Meltdown and Spectre patches and their impacts”, in Workshop on Advanced Microkernel Operating Systems, Hessen, Germany, 2018, pp. 53-61.

[7] A. Prout et al., ”Measuring the Impact of Spectre and Meltdown”, 2018 IEEE High Performance extreme Computing Conference (HPEC), 2018. Available: 10.1109/hpec.2018.8547554 [Accessed 11 July 2021].

[8] A. Wong, ”Complete List Of CPUs Vulnerable To Meltdown / Spectre Rev. 8.0 — Page 4 of 9 — Tech ARP”, Tech ARP, 2017. [Online]. Available: https://www.techarp.com/guides/complete-meltdown-spectre-cpu-list/4/. [Accessed: 26- Jul- 2021].

TABLE I
Performance Data for CPU Tests 10000 Events

System Type	Average Execution Time (s)	Average Time Per Request (ms)
Vulnerable	14.9096	1.49
Patched	15.9289	1.59

TABLE II
Performance Data for Write Memory Tests

System Type	Average Execution Time	Average Number of Events	Data Transferred MB/s	Operations per second
Vulnerable	2.6304 s	1048576	317.51 MB/s	325127.24
Patched	2.6641 s	1048576	310.66 MB/s	318117.49

TABLE III
Performance Data for Read Memory Tests

System Type	Average Execution Time	Average Number of Events	Data Transferred MB/s	Operations per second
Vulnerable	2.0357 s	1048576	441.92 MB/s	452522.54
Patched	2.0837 s	1048576	423.71 MB/s	433880.22

TABLE IV
Performance Data for Custom Test Averaged over 500 Times

System Type	Average Execution Time w/ Mitigation	Average Execution Time w/o Mitigation
Vulnerable	0.000093 s	0.000089 s
Patched	0.000087 s	0.000084 s