Graham Dungan
February 14, 2025
UIN: 332001764

CSCE 413: Software Security
Self-Study: SPECTRE

What is the Vulnerability?

The SPECTRE vulnerability is an exploitation of CPU hardware that allows attackers to access restricted memory regions in a computer by abusing speculative execution. SPECTRE is not a single instance of faulty code, but the existence of a paradigm of hardware/software optimization that did not account for the security implications of the code that it is running.

What was the Root Cause?

The root cause of SPECTRE is branch prediction and speculative execution of instructions in CPUs. Based on previous instructions, the program will create a ”guess” as to which branch should be traversed. These guesses are informed by the previous behavior of a program and thus are very useful for code segments with high temporal locality. Speculative execution then uses this guesswork to attempt to fulfill instructions to fill during empty clock cycles. Suppose the CPU is waiting for some value to be retrieved from a cache. While waiting, it will use previous operations to inform the output of the current operation it is working on, making a ”guess”. Once the variable is retrieved, it will check its guess. While branch prediction and speculative execution offer a great performance benefit, the ”guesswork” of the CPU can undermine security checks on important variables. I will provide a popular example involving an out-of-bounds access vulnerability,

```
if (x < array1_size)
    y = array2[array1[x] * 512];
```

Suppose x is some `array1_size` is some uncached value that needs to be retrieved. An attacker can ”train” the CPU by previously providing x values that make the `x < array1_size` expression return true. Due to speculative execution, the trained CPU will tend to predict this expression to be true. The attacker can then provide an x value that is beyond the scope of the array, and will then access memory that is out of bounds. Since the CPU needs time to fetch `array1_size`, it will ”guess” that the expression is true, resulting in an out-of-bounds access. This out-of-bounds access can be used to perform side-channel attacks by caching desired data and then timing their retrieval later.

What is the Extent of its Impact?

When SPECTRE was first discovered, any CPU that offered speculative execution could have been exploited. By this forum asking about intel, and this forum asking about AMD, an overwhelming majority of modern CPUs including Intel, AMD, and ARM were vulnerable. A particularly dangerous effect of the vulnerability was that interpreted languages on web pages could be used to perform remote exploitation attacks in browsers. This whitepaper describes a proof-of-concept method of accessing private memory addresses from Google Chrome. Following, the discovery of SPECTRE introduced new methods of using cache timing to access secure cryptographic information.

How to Patch it?

Since this vulnerability originated in hardware and made all aspects of software, from operating systems to browsers, means of launching multi-level-attacks, hardware and software providers had to offer different patches. Hardware companies, such as CPU manufacturers, began rolling out hardware and firmware fixes for SPECTRE as soon as early 2018. Google, Linux, Windows, and Intel pushed immediate software and firmware patches, and third-party entities offered patches that clamped down on the use of speculative execution at the cost of computational speed. CPU developers offered new instructions for preventing speculative execution, and firmware/software developers offered new protections for accessing memory addresses.

How Could it Have Been Prevented?

Many experts agree that SPECTRE is a complex problem that would take time to fix, as the vulnerability predicates the nature of wanting to make optimizations for hardware. At the 2019 IEEE Symposium on Security & Privacy, Paul Kocher argues that SPECTRE is not a bug, but a symptom. Kocher sums up this issue by claiming that ”the guarantees that hardware provides is insufficient for security”. Ultimately, he believes that SPECTRE is a symptom of an unregulated drive towards increasing performance without regard for the software (and the security of the software) that is running.

References

1. Spectre Attacks Exploiting Speculative Execution
2. Cache-timing attacks on AES
3. Spectre Paper
4. Spectre & Meltdown - Computerphile
5. Geeks for Geeks Spectre Security Vulnerability
6. How Branch Prediction Works in CPUs
7. CERT CPU hardware vulnerable to side-channel attacks
8. SPECTRE Web Vulnerability Whitepaper
9. SuperUser: Which Intel CPUs are affected by Spectre/Meltdown?
10. Reddit: Do Spectre/Meltdown Attacks Affect AMD's Zen?
11. 2019 IEEE Symposium on Security & Privacy - Paul Kocher on Spectre