CPU Vulnerability Watch

On 5/27, at the COMPUTEX 2019 conference, AMD announced its latest line of CPUs: the RYZEN 3000 series.

In order to adapt to large-scale changes to multicore processors, CPUs have been undergoing great structural changes, as well as adopting new 7nm process rules. Though the technical potential of these CPUs is quite interesting, perhaps even more surprising is the associated price tag.

Starting off with the specs, up until now, most CPUs sold to the general public on the PC market have at most only had eight cores, yet this time, 12-core processors have been added to the lineup.

At nominal value, the 12-core Ryzen Threadripper 2920X goes for $649, and the Core i9-9920x for $1199. As per the intention of the creators, these CPUs are being sold on the high-end desktop market, but the newly revealed Ryzen 3900X goes for as little as $499.

Though one can't help but wonder what kind of new technology could make such drastic price decreases possible, that is an entirely different topic, so it is a discussion for another time.

If you are interested, make sure to take a look at the Zen2 architecture. I for one am looking forward to its release date on July 7th.

Though I have started off with the topic of CPUs, it is time to turn our attention to the main subject matter; in this column, I would like to examine two recently-discovered CPU vulnerabilities.

Please note that the following is not a compilation of all known CPU vulnerabilities, nor is it a plea to accelerate the development of solutions to said vulnerabilities. At this point in time, if one were to employ unofficial solutions or install unofficial software patches, the performance of the CPU could be negatively impacted. This article is not a discussion of the necessity of countermeasures as much as an appreciation of a technical topic.

SPOILER

Along with Meltdown and Spectre, this vulnerability is caused by speculative execution. However, as the root cause of SPOILER is different from the other two, it cannot be solved in the same way. The target of SPOILER is a speculative execution technique called "Store Forwarding". In order to correct this issue, it is necessary to have some knowledge of the assembly, so I will do my best to provide a surface-level explanation.

The two operations carried out by a CPU are store and load; that is writing data into memory and reading data from memory, respectively.

If a load operation is carried out without waiting on the results of a different operation (speculative execution), there is a possibility that the results of this load operation will be incorrect. However, even if this happens it will not be written to memory, therefore it will not directly influence any other operations.

As for the case in which the result is incorrect, if a speculatively executed load operation takes place while a store operation is currently executing on the same memory address, the load operation will be caught.

The store buffer will confirm whether or not the load operation is referencing the same memory address as the store operation, and if it is, the data in the store operation will be forwarded to the load operation (store forwarding).

This interaction is quite simple if it takes place in a physical memory address, but if this occurs in a virtual memory address, the calculating power of a physical memory address will become needed, and because the calculations taking place in a physical address will finish much later, whether this store forwarding will be conducted or not will also be judged speculatively.

If this following speculative execution is successful and the store forward is conducted, it is possible to speed up the process to make it execute in advance, but if it fails, it will be re-executed. Because of this, there is a major difference in the execution time between the two cases.

Not limited just to SPOILER, many CPU vulnerabilities caused by speculative execution have the ability to observe these differences, and with this information they become able to read data which they would not be able to otherwise.

If you wish to know more, please refer to the following URL.

External website: Cornell University

Microarchitectural Data Sampling (MDS)

This vulnerability also stems from speculative execution, and the details are divided into the following four categories:

CVE-2018-12126：Microarchitectural Store Buffer Data Sampling（MSBDS）

CVE-2018-12127：Microarchitectural Load Port Data Sampling（MLPDS）

CVE-2018-12130：Microarchitectural Fill Buffer Data Sampling（MFBDS）

CVE-2018-11091：Microarchitectural Data Sampling Uncacheable Memory（MDSUM）

Among these, MFBDS is referred to as a ZombieLoad Attack, and there is proof that it can be used as a method of attack.

However, a fix has already been established by Intel and various OS programmers.

As for the cause of the ZombieLoad Attack, in the case that an exception arises from a load operation, there is a possibility that normally unusable old cached data from previous speculative executions will become usable again.

In the same manner as the Meltdown vulnerability, it is possible for succeeding operations to make use of this data before the previous process is suspended, or before any rollback takes place, and therein lies the problem.

Without being able to read the address pinpointed by attackers, ZombieLoad is fundamentally unable to read the desired data, even if it is being leaked by the current load operation.

However, the discoverers of ZombieLoad have demonstrated that it is theoretically capable of acquiring data such as passwords when combined with other methods of attack. Please refer to the URL below to learn more.

External website: ZombieLoad in Action: Spying on your visited websites

*It is a video, so please be mindful of your volume settings

A report has also been written on the subject, so if you would like to know more, please refer to the following URL as well.

External website: Cornell University

Summary

This time I covered SPOILER and MDS (ZombieLoad), however several other CPU vulnerabilities have been discovered as well. Since the discoveries of Meltdown and Spectre at the start of last year, there has been a great number of CPU vulnerabilities and attacks relating to speculative execution. As researchers are currently focusing on the issue, it appears that this trend will continue.

Since it is important to carefully observe the data of these attacks, and carrying out a successful attack is not something that can be done in just one day, I would like to avoid rushing the matter to the extent that it affects other affairs.

However, because the code has been revealed to the public and proven to be able to be used for attacks, it is difficult to stay clear of the issue.

I believe that just like with dealing with other matters of security, it is important to first carefully compare the risks and costs of the solution before making a proper decision.

With that aside, vulnerabilities such as these are dependent on the architecture and structure of the CPU. If researched, you would be able to deepen your understanding of your current CPU.

Perhaps most people are not interested in a topic of this nature, however if you were to wonder just how this everyday object functions, wouldn’t you possibly find something of interest?

With that said, let us meet again next time!

Written by Evangelist Matsuoka Masayuki