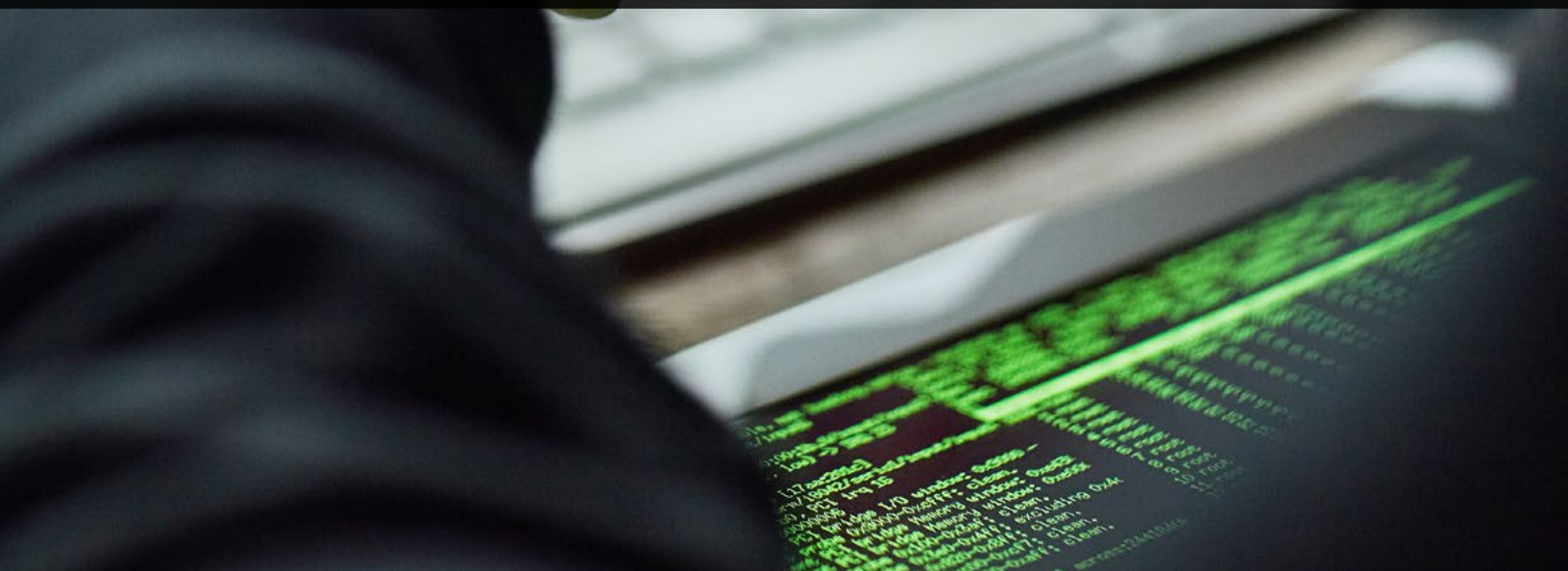
CenturyLink®

WHITE PAPER

High Fidelity Threat Intelligence: A Beginner's Guide to Spectre and Meltdown

Dave Dubois, Global Security Product Management

Version: 1.0, Jan 2019



Spectre and Meltdown Basics

In June 2017, three independent groups of researchers nearly simultaneously discovered potential security vulnerabilities in current microprocessor designs and disclosed them to Intel, the world's leading designer and provider of microprocessors. These vulnerabilities relate to architectural advancements in the processors to work around inherent delays in data transfers to and from memory. The specific vulnerability is called "branch predicition." Branch prediction occurs when a processor, under the right circumstances, "guesses" which path a code branch is going to take when the branch is dependent on data that is currently inbound from memory. When the processor guesses incorrectly, it performs a "revert" operation to trace back to the branch and re-execute the correct branch. During this revert operation, data that was in use gets flushed to a cache area that is accessible to all applications running on the processor.

Researchers have developed code that demonstrates it is possible for critical data (account names and numbers, passwords, etc.) to be exfiltrated, but this code is purpose-built for this demonstration. It is important to understand that in the normal course of events, the data that will end up in the revert cache is highly nondeterministic. Given that, to exploit this vulnerability, it would be necessary to rapidly extract data from the revert cache and transmit it to a command and control (C2) server (or an area designated by the C2 server) for post-exfiltration processing.

Neither Spectre nor Meltdown have been witnessed "in the wild," so their attack vectors and infiltration methods are not yet known. But it is anticipated that threat intelligence sources will identify and report on C2 and malware servers that carry this threat as well as spam and phishing sites that look to penetrate peripheral defenses to deliver malware with these exploits.

To gain a deeper understanding of the potential impact, let's examine some microprocessor basics and how a potential cyber threat could take advantage of these vulnerabilities.

Moore's Law and Microprocessor Basics

Moore's Law predicts that microprocessor speed will double approximately every 18 months. These technological advancements are primarily driven by new generations of photolithographic equipment that "etch" the transistors into silicon base materials. Each new generation of photolithographic equipment is capable of thinner etches which creates smaller transistors that can be packed tighter onto the chip. Creating more transistors per square inch of material increases the processing power of the chip. Reducing the distance between transistors reduces signal latency between them.

Aside from the photolithographic advances, microprocessor developers have implemented several architectural techniques that maximize how the core processing logic on the chip accesses memory. Perhaps you are aware of microprocessor speeds quoted in multiples of gigahertz (GHz). At the time of this writing, processor "base frequencies" are in the 2 GHz to 8 GHz range. This is the rate at which simple operations occur within the central "thinking" part of the chip called the arithmetic logic unit or ALU. A simple operation or instruction might be to add 2 registers together – add register 1 to register 2 and put the result in register 3. A 2.4GHz chip can execute 2.4 billion of simple instructions in a single second given that the data is available on the chip.

Registers are data locations in the ALU on which operations can take place. To affect data in memory, it first needs to be loaded into a register. Given this, adding two numbers together from memory locations X and Y and storing the result in location Z is done as follows:

- Load register 1 with memory location X
- Load register 2 with memory location Y
- Add register 1 to register 2 and store the result in register 3
- Store register 3 in memory location Z

The seemingly simple (but critical) operation of "load register 1 with the memory location X" is key to understanding many of the memory access acceleration techniques built into modern day microprocessors.

As mentioned earlier, simple arithmetic and logic operations can be executed in a single cycle. However, if the ALU needs to reach out to memory to get a value, the operation could easily expand to 400 processor cycles or more. The operation of accessing main memory takes about 400 times longer than a simple ALU operation.

Architecture Considerations

Most memory acceleration techniques rely on having a memory space on the microprocessor itself. This space is called "cache memory." Typically, a microprocessor will have between 128 KBs to 1 MB of cache memory while a typical home PC has 4 GBs or more of main memory. Because the cache is a small fraction of the overall memory, getting the right data into the cache is critical to high performance operation.

There are several techniques to getting the right data into the cache, including "reading ahead." This is where the memory logic observes that the application is reading memory sequentially, so it will pre-fetch data from memory before it is needed, generating a high "cache hit" rate. The illustration below is a simplified version of how the central core logic interacts with memory.

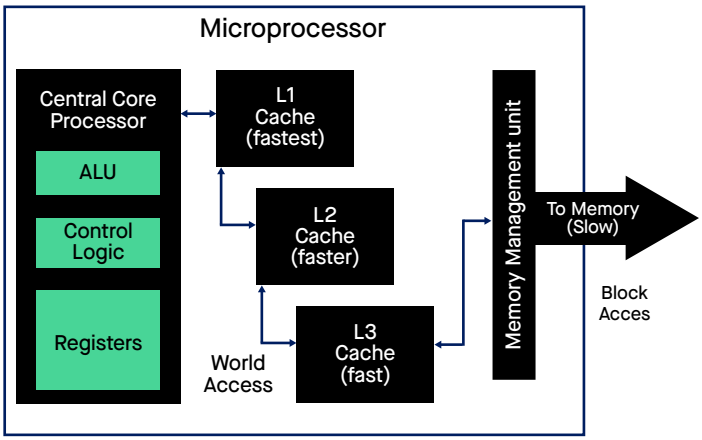


Figure 1: Microprocessor/Memory Interaction

Branch prediction is another key acceleration and the one that is most relevant to Spectre and Meltdown. Most code can only execute a dozen or so instructions before needing to make a "branch" decision, i.e. "If X is greater than zero do thing-1, otherwise do thing-2."

If they were all that simple, then the magic of cache memory would obviate the need for branch prediction (X would be in cache and zero is a constant).

But often, a branch decision relies on data from a more complex structure, which may not be currently in the cache. In this case, the branch prediction logic would attempt to predict which choice would be made and continue down the predicted path. The prediction would be based on the previous iterations through the code. For instance, if the last "n" times through this code took "path A," then we can predict iteration "n+1" will do the same. We call this "training the branch prediction logic."

While the code is executing down a predicted path, a copy of any data that gets changed is kept in the revert cache. Once the branch data arrives from memory, the prediction can be tested for accuracy. If the prediction is incorrect, a revert is executed, leaving the revert cache data in place.

When an application starts on a processor and memory is allocated for that application, the processor configures the Memory Management Unit (MMU) so that the application cannot reach beyond its allocated memory. If the application erroneously or purposely attempts to access memory outside its bounds, a memory exception will occur and, unless the application has an exception handler in place, the application will abruptly end. The revert cache, however, is not protected by the MMU. This is the essence of the vulnerability that makes Spectre and Meltdown possible. Any application can read from the revert cache. This makes it possible to write an application that aims to obtain sensitive or interesting data by continuously reading from this revert cache.

Because the data in the revert cache is nondeterministic, incomplete and of varying value, it is likely attackers will stream its content to a storage area for post processing. Once in a secondary storage area, the data can be analyzed using big data analytics, artificial intelligence and machine learning. However, these capabilities are likely to be beyond the common DDoS attacker who can simply rent a botnet for tens of dollars per hour to perpetrate their typical brand of chaos.

Spectre and Meltdown in the Wild

As stated, we have yet to see these exploits in the wild, but they are likely to use similar attack vectors and infiltration methods as most other malware. To take effect, malware will need to be downloaded to a compromised device. Cyber threat intelligence providers will utilize established malware detection methods to track and identify this code. Scanning for open ports, distributing spam and automated phishing will be used to infiltrate the enterprise peripheral defenses. A possible watch list of techniques to look for include:

- Contact with known Malware or C2 (Command and Control) devices that distribute these exploits
- Contact with phishing sites that attempt to direct contact to malware distribution sites
- Spam that attempts to direct contact to malware distribution sites
- Scanning for open ports by malware sites that distribute these exploits
- Contact with anonymous proxy (or TOR) exit nodes used to obfuscate
- Unwanted or unauthorized applications being distributed
- Unexpected file hash check or length check results on executable files

Once the malware makes its mark on a compromised device, there should be a steady stream of data exfiltration to a secondary storage area for post processing. This is another "tell" that can be used to identify sites that are compromised by these exploits. Once identified, typical blocking techniques (IP address, port, domain or URL) can be used to thwart further exfiltration.

Conclusion

Because Spectre and Meltdown take advantage of optimization techniques architected into modern day microprocessor designs, it is difficult to patch operating systems or other system software to prevent malware from exploiting these vulnerabilities once it has made its mark. That is what makes these exploits unique. However, typical attack vectors and infiltration methods will be used to distribute any malware associated with these techniques.

The best method to protect against the effects of this malware is upfront prevention, including:

- Visualization of activity with known malicious or proxy sites using advanced threat intelligence tools like Adaptive Threat Intelligence from CenturyLink
- Education and testing of the user community to identify and avoid phish lures and spam
- Identification and removal of unauthorized applications
- Identification and removal of applications with unexpected file hashes or file lengths
- Visualization and identification of unexpected data streams to unauthorized sites
- Keep current with threat intelligence bulletins that will identify when these exploits have been seen in the wild
- Ensure your asset inventory and vulnerability assessments are current

As always, active prevention is the best alternative when it comes to defending against the most advanced cyber threat attacks. Using the best visualization tools to identify network interactions that may compromise your environment is critical in the day-to-day defense of your company's critical data assets. Please contact your local CenturyLink sales representative to get a briefing on the best preventative techniques available to you.

Call 1.877.453.8353 | **Click** centurylink.com | **Email** info@centurylink.com

CenturyLink®