Article development led by **acmqueue**
queue.acm.org

**Learn from the past to prepare
for the next battle.**

BY RICH BENNETT, CRAIG CALLAHAN, STACY JONES,
MATT LEVINE, MERRILL MILLER, AND ANDY OZMENT

How to Live in a Post-Meltdown and -Spectre World

THE WORLD OF vulnerability management is rapidly changing to keep pace with the complexity of potential threats requiring remediation. What will it look like to live in this world for the next 10 to 15 years?

In 1996, Aleph One published “Smashing the Stack for Fun and Profit.”¹ For the next decade, stack smashing was a common form of exploitation, and the security community expended significant effort to finding defenses against it. The Spectre and Meltdown vulnerabilities may constitute an equally seminal moment, ushering in a decade or more of chronic risk-management issues. Indeed, two

variants were recently released: SpectrePrime and MeltdownPrime, as detailed in a recent paper by Caroline Trippel, Daniel Lustig, and Margaret Martonosi.³ Expect these to be the first of many.

Spectre and Meltdown create a risk landscape that has more questions than answers. This article addresses how these vulnerabilities were triaged when they were announced and the practical defenses that are available. Ultimately, these vulnerabilities present a unique set of circumstances, but

for the vulnerability management program at Goldman Sachs, the response was just another day at the office.

While these vulnerabilities are theoretically fascinating, we have to live with their practical impact. As risk managers at Goldman Sachs, a large enterprise of approximately 35,000 employees, we had to respond rapidly to the announcement of the vulnerabilities. Moreover, we will have to continue managing the risks that will arise over the next decade from new variants or similar vulnerabilities.

We learned about the vulnerabilities when they were publicly announced on January 3, 2018. The announcement was made earlier than planned because word was already starting to leak. This meant that many vendors had not yet released patches or prepared customer communications about impact, mitigation strategies, and the timelines for patch availability. Vendors could not immediately help in understanding the vulnerabilities.

The first challenge when any major vulnerability is released is to gather

information: which systems are impacted, when will patches be available, what compensating controls are in place, and is the vulnerability being actively exploited? It's even better to know if the vulnerability is being exploited by threat actors who have historically targeted your firm.

Meltdown and Spectre were particularly difficult to triage. It was clear early on that certain processor families were impacted, but the full scope was suspected to be much wider. Moreover, our hardware and software inventories

focus on operating systems, applications, and the overall computer model. They are not set up for rapidly revealing the brand and model number of the processors.

It would have been simpler to patch all our machines, but we were wary of news that patches might cause significant performance impacts.

Initially, estimates of performance impact from patching ranged wildly on blogs and articles and were not directly cited in official papers. On January 18, 2018, Eric Siron of Altaero.com summarized that sentiment, saying, “We’ve all seen the estimations that the Meltdown patch might affect performance in the range of 5% to 30%. What we haven’t seen is a reliable data set indicating what happens in a real-world environment.”² Those ranges were borne out in our own testing of patches, with some systems suffering worse slowdowns than others. Moreover, roundtables with other chief information security officers indicated similar ranges.

These patches had a particularly poor risk trade-off: high potential performance impact, imperfect security benefit. Normally, a patch fixes a vulnerability. Because these are fundamental design vulnerabilities—and worse, vulnerabilities in the hardware design—the patch opportunities are limited. Rather than fixing the underlying vulnerability, they essentially put up a labyrinth to stop an adversary from exploiting it, but the underlying vulnerability remains. Moreover, our experience with complex vulnerabilities is that the first patch is often flawed, so we expected that many of the patches would be updated over time—an expectation that has since proven true.

Although patching was clearly going to be problematic, our quick triage highlighted some good news. Exploiting these vulnerabilities required executing code locally on the victim machine. That led to considering which parts of the operating environment are likely to run untrusted code: hypervisors in the public cloud, employee endpoints such as desktops and laptops, mobile devices, and the browsers or applications that often open email attachments. Since patches could have significant

performance impacts, every decision would have to involve a risk trade-off.

The conclusion was that desktops were at most risk, and testing showed that the performance impact would be manageable. We thus immediately began to patch all of our desktops. For servers, we decided to investigate further and make more nuanced, risk-based decisions. The risk of cyber-attack had to be balanced against the operational risk of the patch breaking or significantly slowing the systems.

There was no information that the vulnerabilities were being actively exploited, which was reassuring. On the other hand, the nature of the vulnerabilities is such that exploitation is difficult to detect. If we know a vulnerability is being exploited, we will try to push a patch even if there is a high risk of the patch breaking some of the systems. With these vulnerabilities, the lack of known exploitation reinforced the decision to take more time assessing our servers.

To aid in this assessment of risk, we examined our patch strategy and compensating controls through the following lenses: public cloud, servers, employee endpoints, browsers, and email. These lenses also helped communicate the risks to our business leadership.

Public cloud. Research showed that attacks leveraging Meltdown and Spectre could target a public cloud environment. In certain cases, an attacker could defeat the technology used by the public cloud providers to ensure isolation between customers’ instances. If a malicious user were able to bypass the hypervisor or container engine controls, then that user could access other customers’ data collocated on the same hardware.

Thus, our most immediate concerns were public cloud providers. The public cloud risk could be further broken into instance-to-instance attacks and within-an-instance attacks.

In an instance-to-instance attack, a customer could attack another customer on the same hypervisor. Meltdown was the most obvious vector for this attack. An attacker could theoretically just pay for an instance on the public cloud and then target any other customer on that hardware. Fortunately, several large provid-

ers—including Amazon, Google, and Microsoft—had received advance notice and had completed, or nearly completed, an initial round of patching on their hypervisors to address these concerns. Moreover, some of the providers informed us that they had patched months before the vulnerabilities were publicized without any noticeable performance impact.

For a within-an-instance attack, the attacker would have to run code on the same instance. This would require access to the system or application to exploit the vulnerability. It was not immediately clear what needed to be done to completely protect against the multiple variants that could be used in this attack. The protections implemented by the public cloud providers remediated Meltdown, but the Spectre variants required multiple mitigations. Google published a binary modification technique called Retpoline that it used to patch its systems against Spectre Variant 2. This had the benefit of minimal performance impact compared with CPU patches. Mitigations for other providers included chip firmware, hypervisor patches, operating system patches, and even application rewrites.

Spectre remediation is made even more complicated because customers and cloud providers have to work in tandem, depending on the cloud service in use. Our impact analysis determined that the within-an-instance risk was not significantly increased by running instances in the public cloud: It was essentially the same risk faced with the internal servers. Accordingly, we treated it as we treated all of our servers: by making individual, risk-based decisions.

Servers. At Goldman Sachs, server performance is critical, so we must be careful in patching our servers. In financial services, many critical applications are time-sensitive and effective only if the processing is completed rapidly—for example, applications that perform trading or large-scale, complex risk calculations. This patch could have very real-world implications. If the hundreds of thousands of public cloud processors used every night to perform complex risk calculations had their processing speed reduced by 30%, in addition to the operational risks that

could be raised and potential concerns about robust and real-time risk management, our bill could potentially see a significant increase to compensate for the lost computing power.

Server patching then actually becomes a question of understanding the firm's thousands of internal applications and making a risk-based decision on them. For some applications, performance is critical, and the likelihood of running untrusted code is low. In those cases, we—and the other major firms we talked to—decided not to patch. For those applications, we relied upon compensating controls and the fact that they are very unlikely to run untrusted code. For other applications, we assessed the risk to be higher and patched their servers. To do this type of risk-based analysis, a firm has to understand both the behavior (application profiling) and risk (risk-based categorization) of its applications.

Endpoints. Employee endpoints, such as desktops and laptops, were also a high priority within our triage process, as they have access to the Internet via the Web and email. These are key channels through which threat actors looking to exploit these vulnerabilities could attempt to deliver malware.

The Goldman Sachs endpoint response had two key themes: patching and controls. Because user endpoints are much more likely to run untrusted code than servers, we decided to patch in all but the most exceptional circumstances. We thus rapidly deployed patches to our managed Windows, macOS, and iOS devices as they became available. Because of concern over potential end-user performance impact, it was hugely beneficial to be able to run repeatable, automated testing on an isolated set of endpoints before pushing the patches across the enterprise.

Patching was not focused exclusively on the operating system. We also considered the availability of patches for components on the employee desktop that could allow for untrusted code execution—for example, applications that open business-related documents. Unfortunately, even months later many of those applications have not been patched by their vendors.

Meltdown and Spectre were particularly difficult to patch. It was clear early on that certain processor families were impacted, but the full scope was suspected to be much wider.

Our assessments included the broader control set available on the endpoint—both preventative and detective. We were most interested in determining which layers of defense could play a role in mitigating risk. For prevention, we reviewed the configuration hardening for our builds and application whitelisting capability and concluded that they did not require any changes. We also use both signature-based and heuristic-based malware detection on our endpoints and on incoming email. Of course, the signature-based tools will have value only when exploits become public.

Not only is it important to look at all of the potential options to mitigate the risk, but also to have the foundational blocks in place for controls that can be adapted to mitigate a broad set of threats in a constantly evolving landscape.

Browsers. The Web contains plenty of malicious websites that could attempt to exploit these vulnerabilities. Even legitimate websites may inadvertently host malicious advertisements. Or, in the case of a watering-hole attack, an adversary could compromise a website that a company's employee population is known to visit and use it to deliver malicious code.

At Goldman Sachs we use a Web proxy and service to categorize domain names to reduce risk. Our proxy settings are extremely conservative, blocking entire categories of Web pages that are not relevant to our business or are potentially risky. That includes many of the servers used to host advertisements, so we already have a reasonable amount of advertisement blocking. The proxies also block the downloading of executable files.

In addition, Google Chrome and Microsoft Edge have site isolation capabilities that stop malicious code from impacting more than one tab in the browser window. Like patching, this is not a perfect mitigant for these vulnerabilities, but it does provide a partial control and another layer of defense. As this feature was ready even before some patches, it was implemented rapidly. Although we feared that it would break many internal or external sites, there were actually very few problems.

More specific patches for browsers came out from days to weeks after the initial vulnerability disclosure. We pushed those patches out rapidly. A few hundred of our developers use nonstandard browsers to test their applications, so we used application whitelisting on user endpoints to ensure that only managed browsers, or approved and patched exceptions, were being used.

Browser plug-ins can also execute on untrusted code. As a partial mitigant, specific plug-ins can be locked down to a set of whitelisted sites. Very few plug-ins have released patches, so this remains an area of concern.

Some firms have also chosen to virtualize browsers to isolate the application from the operating system. A browser can be virtualized either as a stand-alone application or as the entire desktop operating system. If any mission-critical Web applications are running on legacy browsers or with plug-ins, a virtualized browser can provide a more protected mechanism for doing so.

Email is another common vector for untrusted code. It is not a likely tool for exploiting these vulnerabilities as a majority of the attack vectors have included cache-timing attacks, which are difficult or impossible to exploit over email. Nonetheless, it is important to address phishing attacks as a means of general exploitation. Most firms, including Goldman Sachs, use a variety of techniques to block email-based attacks.

The simplest technique is to block certain types of attachments. If your business supports it, this is a relatively cheap control that can have a significant impact. Unfortunately, many businesses depend upon being able to share office documents, such as PDF or Excel files, that can include macros or other types of code.

Of course, phishing emails do not necessarily contain attachments. They can also contain links to malicious websites. We rewrite incoming URLs so that outbound calls have to go through a central control point where we can quickly implement a block. Outbound Web connections also have to go through the same proxy-based controls described earlier.

In addition, we use signature-

based email blocking technologies within our layered approach. As long as there are no known exploits, however, there are no known signatures to deploy. This will be an area to track going forward when the exploits move from research proof-of-concept to being weaponized.

There will likely be more value in “combustion chambers,” which open attachments in a virtual machine and look for malicious behavior. Some combustion chamber vendors are looking at running unpatched virtual machines and using them to detect the exploitation of these vulnerabilities.

Hardware fixes. While patches and controls are the focus here, hardware fixes are not totally out of the question. Intel indicated in its Q4 earnings call that chips with silicon changes (directly addressing Spectre and Meltdown) will begin to hit the market later this year. Similar to the operating system patches, however, the first generation of hardware fixes may not fully address the vulnerabilities. Moreover, it will be years before organizations upgrade all of their hardware with the new chips.

Just Another Day of Vulnerability Management?

These vulnerabilities pushed the vulnerability management process at Goldman Sachs, but they did not break it. We are used to making risk trade-offs in this space: for example, do you patch more quickly to decrease the risk of cyber-exploitation, even if that increases the risk of an operational breakdown?

The risks posed by these vulnerabilities are a more challenging version of that scenario. The question is not just whether the patches would break a system, but whether they would have a significant performance impact. That risk has to be assessed in a distributed way, as it is unique to each application. At the same time, there is a lot of uncertainty about these vulnerabilities and how readily they could be exploited. We therefore have to balance operational risk with cyber-attack risk when both risks are unclear.

The scope of vulnerabilities such as Meltdown and Spectre is so vast that it can be difficult to address. At best,

this is an incredibly complex situation for an organization like Goldman Sachs with dedicated threat, vulnerability management, and infrastructure teams. Navigation for a small or medium-sized business without dedicated triage teams is likely harder. We rely heavily on vendor coordination for clarity on patch dependency and still have to move forward with less-than-perfect answers at times.

Good cyber-hygiene practices remain foundational—the nature of the vulnerability is different, but the framework and approach to managing it are not. In a world of zero days and multidimensional vulnerabilities such as Spectre and Meltdown, the speed and effectiveness of the response to triage and prioritizing risk-reduction efforts are vital to all organizations. More high profile and complex vulnerabilities are sure to follow, so now is a good time to take lessons learned from Spectre and Meltdown and use them to help prepare for the next battle.

Related articles on queue.acm.org

Securing the Tangled Web

Christoph Kern

<https://queue.acm.org/detail.cfm?id=2663760>

One Step Ahead

Vlad Gorelik

<https://queue.acm.org/detail.cfm?id=1217266>

Understanding Software Patching

Joseph Dadzie

<https://queue.acm.org/detail.cfm?id=1053343>

References

1. Aleph One. Smashing the stack for fun and profit. *Phrack* 49, 7 (1996); <http://phrack.org/issues/49/14.html#article>.
2. Siron, E. The actual performance impact of Spectre/ Meltdown Hyper-V updates. *Hyper-V Blog*, 2018; <https://www.altaro.com/hyper-v/meltdown-spectre-hyperv-performance/>.
3. Trippel, C., Lustig, D., Martonosi, M. Meltdown Prime and Spectre Prime: automatically synthesized attacks exploiting invalidation-based coherence protocols, 2018. arXiv:1802.03802; <https://arxiv.org/abs/1802.03802>.

Rich Bennett, Craig Callahan, Stacy Jones, Matt Levine, Merrill Miller, and Andy Ozment are on the global technology risk and information security team at Goldman Sachs.