The microarchitectures that I saw and the ones that I hope to one day see

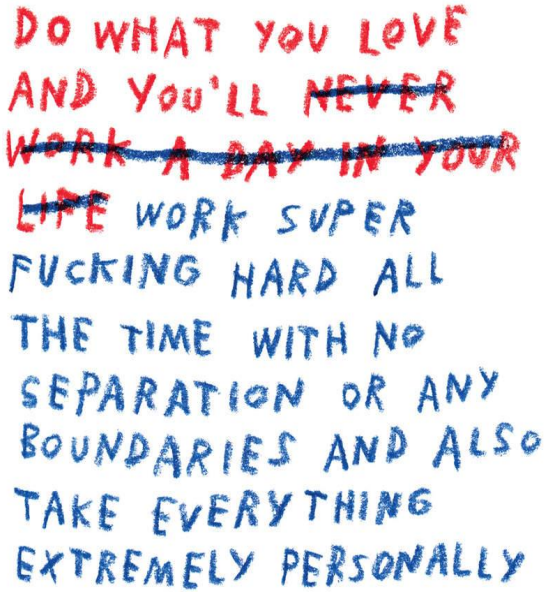
Rodrigo Branco (BSDaemon)

“The smart way to keep people passive and obedient is to strictly limit the spectrum of acceptable opinion, but allow very lively debate within that spectrum”

N. Chomsky

Disclaimers (1/2)

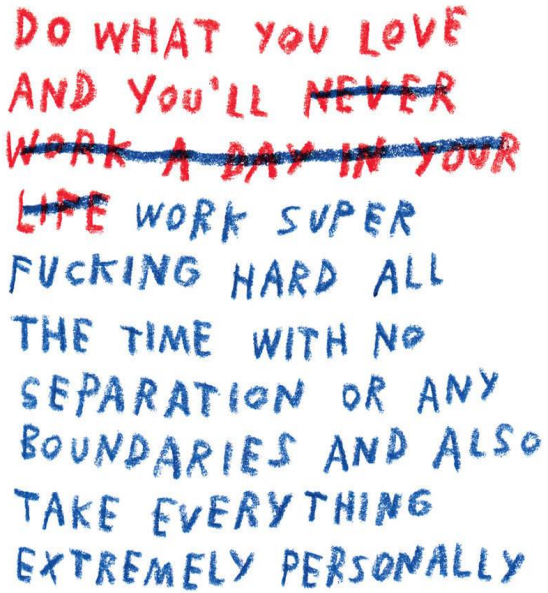
- I’m not speaking for my employer, nor representing past employers
- Everything shared is based on personal memories
- Events unfold in parallel and the experiences and perspectives highly depend on the moment (bias, past experience/knowledge, priorities and emotional situation) of the observer - I always did my best to be sure that those around/close to me are heard too - **I will always be more forgiven to those below than to those above me**



Adam J. Kurtz

Disclaimers (2/2)

- None of this is intended as personal attacks, therefore I omitted names and only talk about positions/titles when the relative power (and how it was used) of the decision maker is relevant
- I use tons of generalizations knowing very well generalizations are risky and unfair to many - In order to be fast, I hope one alert about it could cover all cases during the talk (yes, I am aware that is not how communication works)
- During the talk, I praise companies, researchers and technologies because I believe in them, not because of any personal benefit



Adam J. Kurtz

whoami (1/5)

- I’m best described as a failed farmer
- Writing exploits and finding vulnerabilities for **25 years** now
  - Started with CPU bugs by accident, while researching SMM (2007) - **so 15+ years**
  - Worked for Intel, AWS, Google doing CPU/Platform Security (**11+ years**)
    - Tons of horror stories, lots of learnings and lots of great people (my opinions are forged from those experiences and interactions)
    - Jan 2019: https://www.wired.com/story/intel-meltdown-spectre-storm/

whoami (2/5)

- Pertinent to this talk, I was involved in the following (publicly acknowledged) low-level issues (and hundreds more that were silently fixed)
  - **CVE-2023-1998, CVE-2023-00045, CVE-2020-12965, CVE-2020-0543**, CVE-2019-0185, CVE-2019-0155, CVE-2019-14590, CVE-2019-14591, CVE-2019-11089, CVE-2019-11113, CVE-2019-0151, CVE-2019-0152, CVE-2019-0117, CVE-2019-0184, CVE-2019-0155, CVE-2019-14590, CVE-2019-14591, CVE-2019-11089, CVE-2019-11113, **CVE-2018-3693, CVE-2018-12126, CVE-2018-12130, CVE-2018-12127, CVE-2019-11091**, CVE-2019-0115, CVE-2018-12209, CVE-2018-12210, CVE-2018-12211, CVE-2018-12212, CVE-2018-12213, CVE-2018-12214, CVE-2018-12215, CVE-2018-12216, CVE-2018-12217, CVE-2018-3626, CVE-2018-5736, **CVE-2019-0162, CVE-2018-3615, CVE-2018-3620, CVE-2018-3646, CVE-2018-3665, CVE-2018-3639, CVE-2018-3640, CVE-2017-5753, CVE-2017-5754, CVE-2017-5715**
  - **20+ patents (covering CFI, side-channels, encrypted memory, etc) that impacted in major Intel features**

whoami (3/5)

NONE OF THAT MEANS S**!**

LISTEN TO WHAT I HAVE TO SAY, EVALUATE, CHALLENGE, COME TO YOUR OWN CONCLUSIONS!

MY TIP THOUGH: LOTS OF NUANCES AND REFERENCES FROM BETTER SOURCES, USE AT LEAST THAT IF THE REST IS WORTHLESS :)

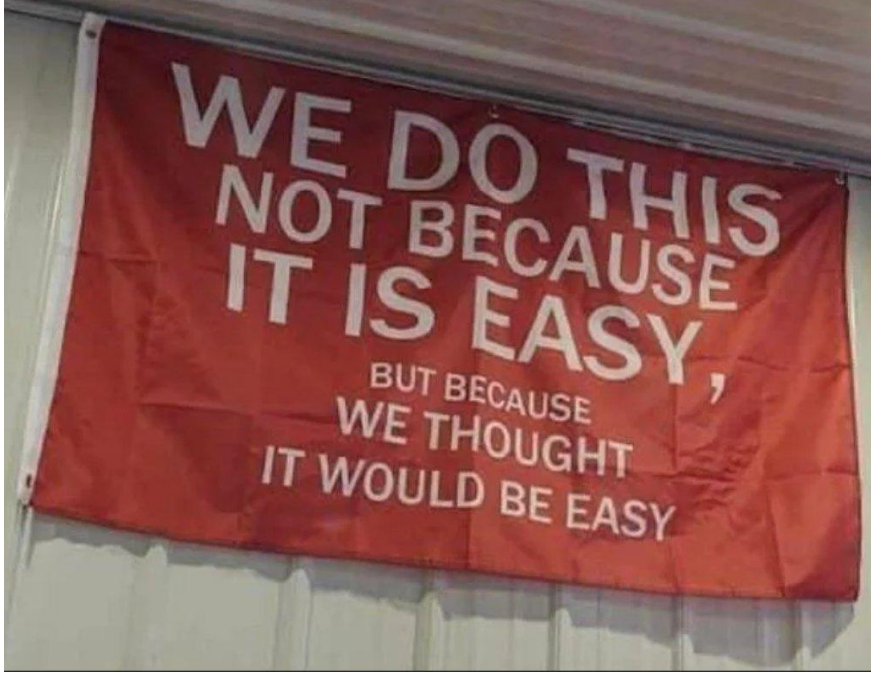
whoami (4/5)

Still, I’m a failure! And this talk is (mostly) about that failure (so hopefully you don’t have to fail as well)

whoami (5/5)

(not the farming failure)

Objectives



- Pass the responsibility over - I’m done dealing with those vendors for the foreseeable future
- Make sure more folks are aware of how bad it is, and how simple things really are (demystifying the myth of this been super hard thus why the problems persist)
- Hopefully give some ideas so ‘better than me’ individuals can follow-up and research on and impact the world :)

Objectives - Truly (1/2)

I believe in transparency, openness and always fought for it!

A hard truth: I think side-channels were/are extremely over-hyped. I tried to get out of it multiple times! At Intel, at AWS, at Google.

Objectives - Truly (2/2)

I want more researchers to understand that they can (and are) changing things (even though individuals in certain companies might dismiss their work and their impact)

My Take on Security

- Two main challenges in security
  - Assumptions
  - Composition

- I believe it's a rule rather than an exception that the potential for vulnerabilities (and therefore, exploits) is already present at the design stage, and as so, can be anticipated at that stage

- While the details matter for a given exploit to be created, a lot of patterns exist across systems and therefore can be abstracted/generalized
  - That essentially mean that the root cause is not in the specific details, but in a more general aspect of the decisions behind the implementation

My Approach to Complex Systems (1/2)

- Architecture should be defined
  - Threat Model created, reviewed
  - Important flows, dependencies (assumptions and composition)
- Start with questions
  - What we don’t know, but should know
  - What we are not able to do, but should be able to
  - What is the taxonomy of past security issues?

My Approach to Complex Systems (2/2)

- Have survivability in mind (specially for HW and hard to patch systems)
  - Simplification/TCB reduction as an explicit goal
  - All involves knowing what is really your input
    - What are your dependencies (compositions)
    - And what are your expectations (assumptions)

Now, now, give us the meat!

Timeline (1/2)

01

2007

Predicting secret keys via branch prediction

Followed by an Intel SPE talk at Oregon State University. The myth of the ‘cache-line boundary side-channel’ is a Software Problem is born

02

2013

The myth grows

2007-2013: Somehow the myth became just ‘side-channel is a Software Problem’

03

2016

DRK: TSX to bypass KASLR

Discussions initiated on if Intel should continue investing in kaslr at all and how to view this kinds of side channels

04

2017

ASLR^Cache - AnC)

I raise the alert that more exploit-dev researchers will look at it. I propose to initiate an effort to look for side channels. By march management said ‘Yes’, but the decision gets reverted back because the security ‘Fellow’ said there was no ROI

Timeline (2/2)

05

Jun2017

Jann Horn from Project Zero sends a report to Intel's PSIRT

I get the report in the same day. After reproducing and understanding (couple of days), I've raised the flag that this was a 'New Class of Issues'

06

Jul 2017

Anders Fogh "Negative Results" blog post

Due to a PoC error, Anders did not find the issue. But had the right insight. Inspired others to look at it

07

Rest 2017

First Linux Kernel Patches (Kaiser) are upstreamed

Somehow Linus Torvalds is not vocal (at all) about them. Patches start getting accepted, so folks that saw Ander's work get suspicious that there is something - Others find the issues and report

08

2018

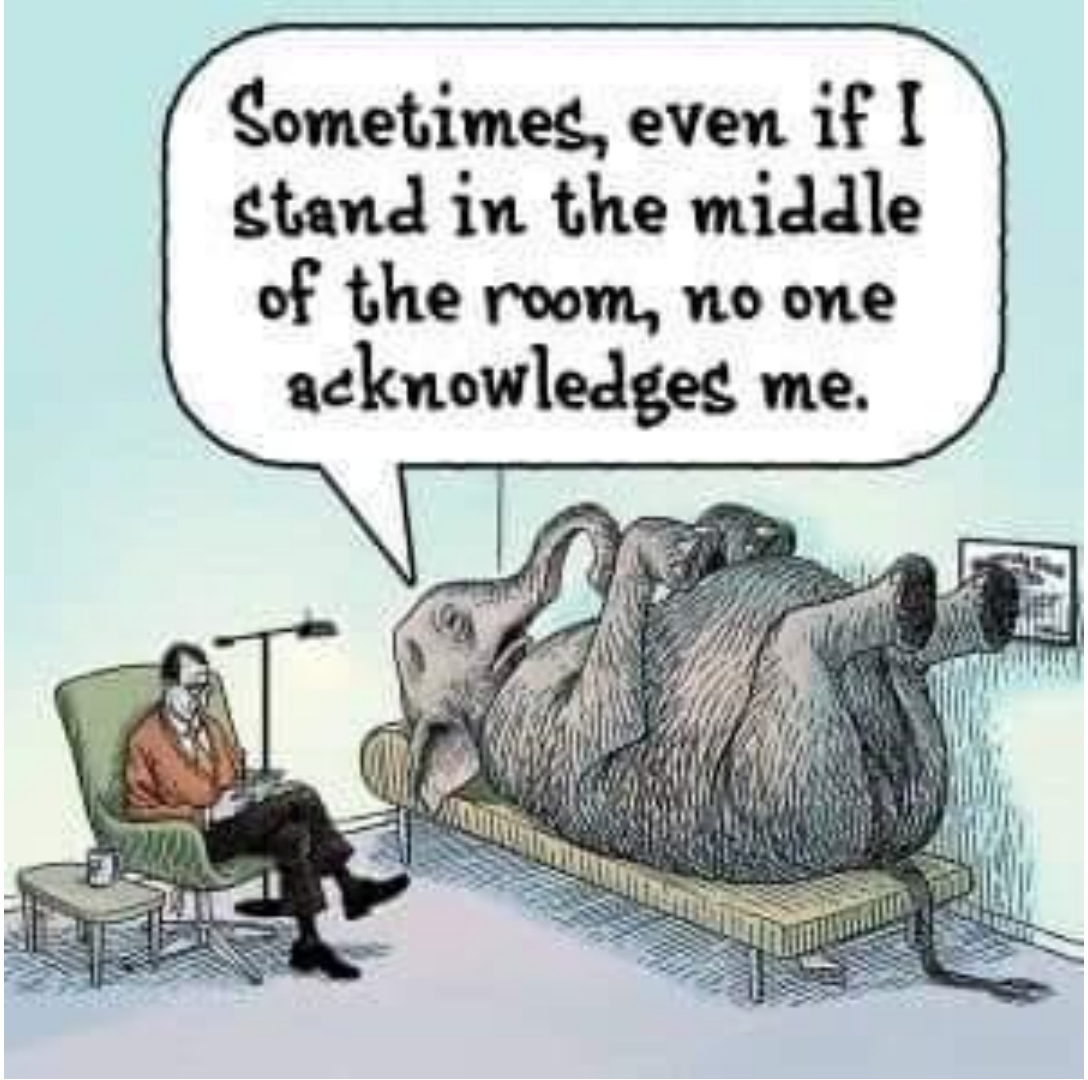
Issues go Public

New Intel organization formed (6 VPs, no employees - the generals-only army), CEO says HW fixes underway (untrue, complicates matters). Class action lawsuits (many, but a major one in US and another one in Israel). Academics jump on it

Why me or my team?

- We were the client security team (covering Client CPU and Platform, we had responsibilities over BIOS/UEFI too for a short stint)

- High Praises to those that made having a team even happen/possible
    - Thanks to Truc Nguyen for convincing me to join Intel
    - Thanks to Burzin Daruwala, Dhinesh Manoharan and David Daughty (now at AWS) for listening to me and making real changes! (that really kept me around)
    - Thanks to Gabriel Barbosa (now at AWS) for the amazing guidance when I joined and outstanding brainstormings and fun interactions over the many years we work together
    - Thanks to Matt King (now at Nvidia) for the ‘Core Dump’
    - Thanks to Yuriy Bulygin (now at Eclypsium) for the amazing guidance when I joined
    - Thanks to Shay Gueron (now at AWS) for replying to my email stating that I was bored and would leave - Shay gave me an interesting challenge to work on (Memory Encryption Attacks)! And for the overall guidance and support over the years - that is real mentorship!


Sometimes, even if I stand in the middle of the room, no one acknowledges me.

But really, why?

- Positive Technologies had found a vulnerability in Intel's CSME (Intel-SA-00086, the vulnerability that is the gift that keeps giving since many further research was enabled due to this bug)
- That came after a couple vulnerabilities were found in Intel's AMT by Embedi (CVE-2017-5689)

And here is a tale of how impactful some things really are, even behind the scenes!

But now you criticize Intel (and AMD) so much, why?? What changed?

Another missing item in the Timeline

Circa 2018 - Intel high management asked STORM to look for vulnerabilities on AMD systems

At first, limited to demonstrate side-channels (since AMD denied they were affected)

Later, overall (covering graphics, PSP, firmware, CPU)

So what?

- We were concerned with the ethics behind having their major team (working on those topics, not necessarily the most important overall) looking at a competitor
- We got commitment that all the issues would be disclosed to AMD (and they indeed were disclosed)
- It is been a few years and still some were not fixed, some were silently fixed, some were found later by others or re-reported by Intel somehow. E.g.,: CVE-2020-12965, intel-sa-00706
  - Initial responses were great, but decayed over time with the leadership change in AMD
- Kudos to IOActive and Binarly researchers for starting to look at AMD and ARM platform firmware (everyone benefits from more eyes)

Unexpected Impacts

- Unfortunately AMD replaced their head of PSIRT (not necessarily due to the reports or from the pressure from Intel), but clearly with someone aligned with Intel (as in: AMD literally consults with Intel for responses and admits - to some gang members at least - that they have no team working on those issues)
- Immediately the collaboration with AMD became problematic (I was in a cloud provider by then)
  - What in the past was engineer to engineer discussions on how to best fix issues became denial and legalize and career threats
  - Entire response integration from customers broke due to system changes (like, really stupid ones)
  - In the past, when AMD had too many issues in a given area, as a customer we would talk to them and really help their team better cover the topic (e.g.: I had a session with their graphics drivers team to help them improve coverage to stop having bugs reported by Cisco Talos)

PSIRTs became driven by MARKETING and LEGAL

- They proxy responses, literally making collaboration a one way street (there are many public examples and literally everyone complaining that they send all info but receive nothing back)
- Instead of fostering (and truly leveraging) collaboration, they created the ‘Gang of’ (initially Gang of 4, later Gang of 10, now Gang of 20+): representatives from major companies and projects that receive private information
  - This became a career benefit, and literally a ‘Gang’
  - The original idea made sense: to be sure the impact of certain issues are understood industry-wide - **Remember Intel’s deposition on Congress due to reporting issues to a OEM in China but not to US CERT?**
- They now give bounties for silence (instead of using bug bounties as an incentive to increase awareness)

Anecdotal Examples

So many examples of failures to look at, from the inconsistencies in their advisories (like even how things are root-caused or how impact/severity is defined) to really bad behavior from folks

Here are just a few examples, so folks who are unaware can have a glimpse! (not in any specific order)

VoltPillager / Plundervolt

- Collision by different researchers (paper)
- (CVE-2019-11157 / INTEL-SA-00289)
- Intel notified somewhere in June 2019, issue published on December 2019
  - Commit: a7e1f67ed29f (Jun/2020) - "x86/msr: Filter MSR writes"
  - By end of Jan/2021, another fix (“As msr register can be written by X86_IOC_WRMSR_REGS ioctl too, they should be applied to the ioctl as well”)
- Different than usual advisories (to keep up with the inconsistencies), this advisory does not specifically list anything hypervisor related, just SGX (as is the case of the papers)
  - But Intel did investigate further (I was there)
  - SMM/Hypervisor obviously affected

Hertzbleed

- CVE-2022-24436 (Intel), CVE-2022-23823 (AMD) (again, inconsistency)
- INTEL-SA-00698 - No mention about hypervisors
- I've contacted Intel PSIRT to let them know that their advisory should explicitly mention the hypervisor case and that many likely would not notice the problem (I did it before the issue became public).
  - To exemplify the matter, I pointed out that the feature was accessible to guests running on Xen
  - Their response came from their PSIRT VP, but not to me, instead, an escalation to the VP of another group in the company and was along the lines of this:

"Your security team is trying to blame Intel for their mistake. Apparently, Xen, a private hypervisor made by you is affected. No other hypervisors are."

But you promised that it is not that complicated…

Reverse BTB Poisoning

- Interesting research done by José Luiz Negreira Castro de Oliveira (the research *was* complicated)
- Jose reported it to Intel (another funny PSIRT f***-up that I will share later), but the work caught my attention (when it was published, I wasn't at Intel anymore)
  - My rationale: When we (Intel) designed IBPB we had a threat model in mind in which the attacker wants to control the branch prediction entries in order to redirect the victim to cause a leak
  - Therefore, the flow for the attack requires attacker and then victim scheduled (as close as possible to increase reliability)
  - IBPB did not consider the actual 'entries' to be secret (even though it does erase the entries)
    - So if anyone optimize the usage and only did IBPB when an untrusted entity was scheduled before a trusted one, they might have a situation in which they are still vulnerable
    - I was very surprised that Intel researchers completely missed that

ret2aslr

- In talking with Jose, it seemed that the main limitation was the actual targets. In brainstorming how to increase the applicability of the attack, we agreed that rets could also be used (and are more prominent than other indirect branches) - so we tested
- That culminated in the ret2aslr research/write-up
- As usual, we've shared with Intel/AMD. As usual, considered irrelevant

But…

- But somehow when we were working on the ret2aslr we've noticed that we saw signals we should not…
- We really wondered why/how Intel/AMD did not even mention it, so we assumed we indeed were that bad and had made some basic mistake somewhere
- Later, we spent the time to examine, to see if we could learn what we had missed…

CVE-2023-0045 - user-mode BTI mitigations 'slightly' broken for 4 years

We did not miss anything, it is just that Intel/AMD respond without really analyzing :)

Top of the Stack

Write-up has the full timeline

- Research done to demonstrate that jmp2ret mitigation was a bad idea (most folks that we discussed the issues with had a misunderstanding on how ret's speculate)
- When we reported top of the stack (Jul 07 - 2022), we had a sentence in the paper: "in case the IBPB instruction does not clear the RAS, as is the case on some microarchitectures".
  - Got a response a week later saying there was nothing new there (but asking to keep secret for 90 days). Intel gave a similar answer.
- After the 90 days, we published the write-up and got a request from AMD to delete the sentence (the only difference is we specifically said 'on some AMD microarchitectures')
- Two weeks later, AMD released an advisory about their IBPB not flushing RSB (not mentioning our work at all, of course)

BUT BEHIND THE SCENES WE GOT A THREAT OF LEGAL INVOLVEMENT AND A FORMAL DEMAND TO APOLOGIZE!

WHEN IT WAS INDEED JUST MORE EVIDENCE OF RESPONSES WITHOUT ACTUAL ANALYSIS

But it is really a GANG problem

Retpoline (1/3)

- As soon as Google shared with Intel the idea of retpoline, our team got to work in testing its effectiveness
  - We've reported to management that we were still seeing signal from PoCs - something was wrong
  - The signals got root-caused to be really deep paths inside the kernel (Linux) - mostly related to interrupt handling (and the 'Fellow' randomly decided without any actual data, analysis or even having worked on developing PoCs/exploits in their life that was uncommon)
  - We also noticed that SMI handlers were consuming RSB entries (which meant that ALL affected systems that had to do the RSB filling would need to have BIOS updates - making retpoline way less attractive/realistic) - we were told that Google and Cloud Providers controlled their own BIOS, so it was not relevant

Retpoline (2/3)

- We also reminded folks (we knew before, since the very beginning of the original analysis of Variant 2) that on some uarches, the rets were going to BTB if the RSB was empty (so ucode patches needed too)
- We also pointed out the difficulty on integrating it with CFI, more specifically CET (that was not released yet, but upcoming) - notice that we had already flagged that CET ‘endbranch’ should serialize (terminate) speculation
  - AFAIK, Open Source Security Inc. (grsecurity folks) are the only ones that really made a proper integration
- We also worked hard on proposing an improvement, which we dubbed Randpoline

All warnings got ignored. Retpoline got disclosed. When retbleed was reported, folks pretended it was all new.

Retpoline (3/3)

- The surprise was when years later, at Google, I've noticed that:
  - They forgot to stuff the RSB for their SMI handlers
  - They forgot to apply the ucode patch on broadwell systems (therefore, they were also vulnerable to ret->BTB)
  - They forgot that the RSB Filling code on the kernel was interruptible and therefore, RSB refilling had to happen in every interrupt handler return **(is that fixed in mainstream yet?)**
  - They forgot that they supported nested guests… **(that got a patch sent mainstream without credits, of course)**
- **But there were tickets… so problem solved :)**
- I used the opportunity to teach about a great operational principle I've learned in AWS: mechanisms
  - It was *NOT* well received

eIBRS

- Did not even break our tests at Intel
  - To make it easier to test across all micro-architectures, we trained in-mode
  - eIBRS does not prevent same-mode trainings
  - We immediately called it out
- During the randpoline research, we also noticed that the branches could be trained via the history and not via the entries
  - We even mention the branch history and the need to randomize it in the randpoline write-up
- Still Intel misguided folks about eIBRS:

  1-) Saying it offered the same security properties as retpoline (it does not protect against same-mode training while retpoline does)

  2-) Stating they did not know about the ability for training the branch history (which eIBRS does not protect against)

Hyperbleed - Trolling the trolls - CVE-2023-1998

“Your test is flawed due to a fundamental misunderstanding of what IBRS guarantees” - Google L8 (Principal Engineer)

- Then, after a lengthy explanation of how IBRS works on different micro-architectures

“Micro-architectural details do not matter!” - Google L8 (Principal Engineer)

FOR A MICROARCHITECTURAL BUG!!!

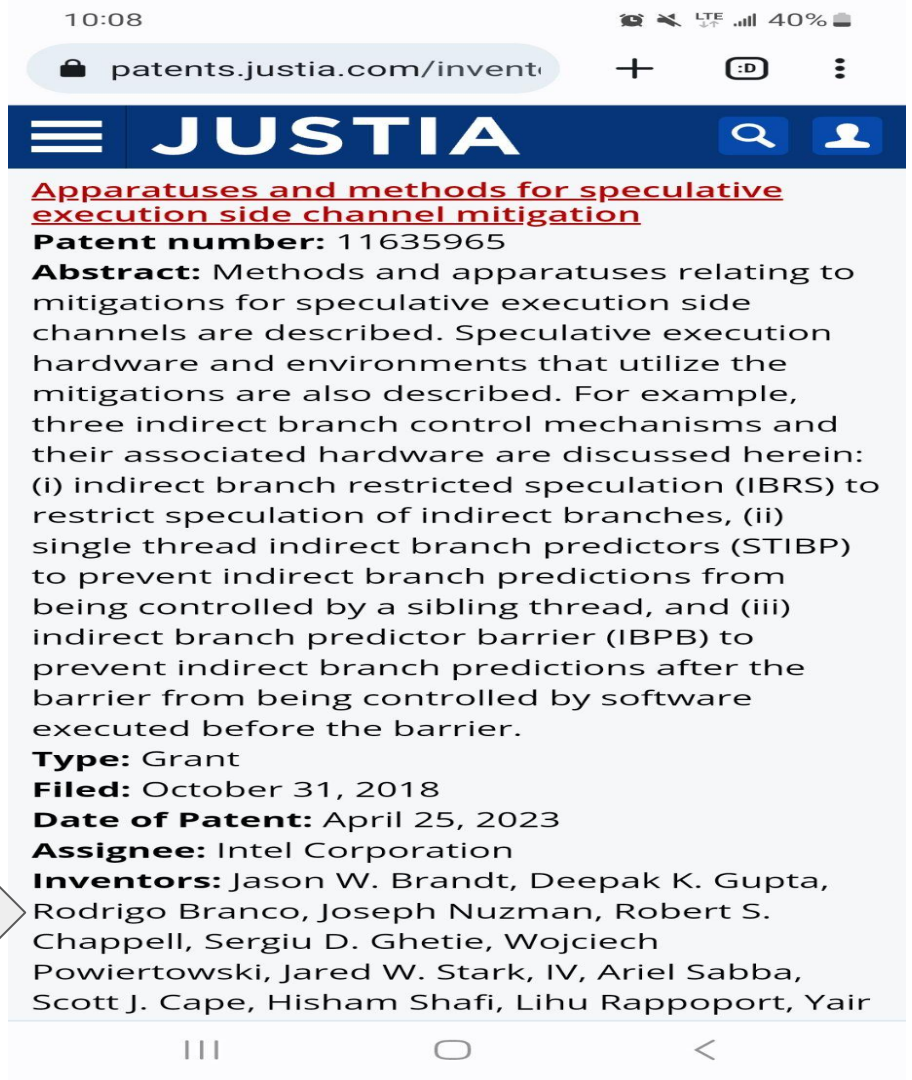
Ended-up been a real issue (of course) due to a misunderstanding of eIBRS!
The world loops around :)

One additional funny detail

That is my name, right there

on the patent for

IBRS :)



If florida alligators are able to post signs, can PSIRTs learn to email?

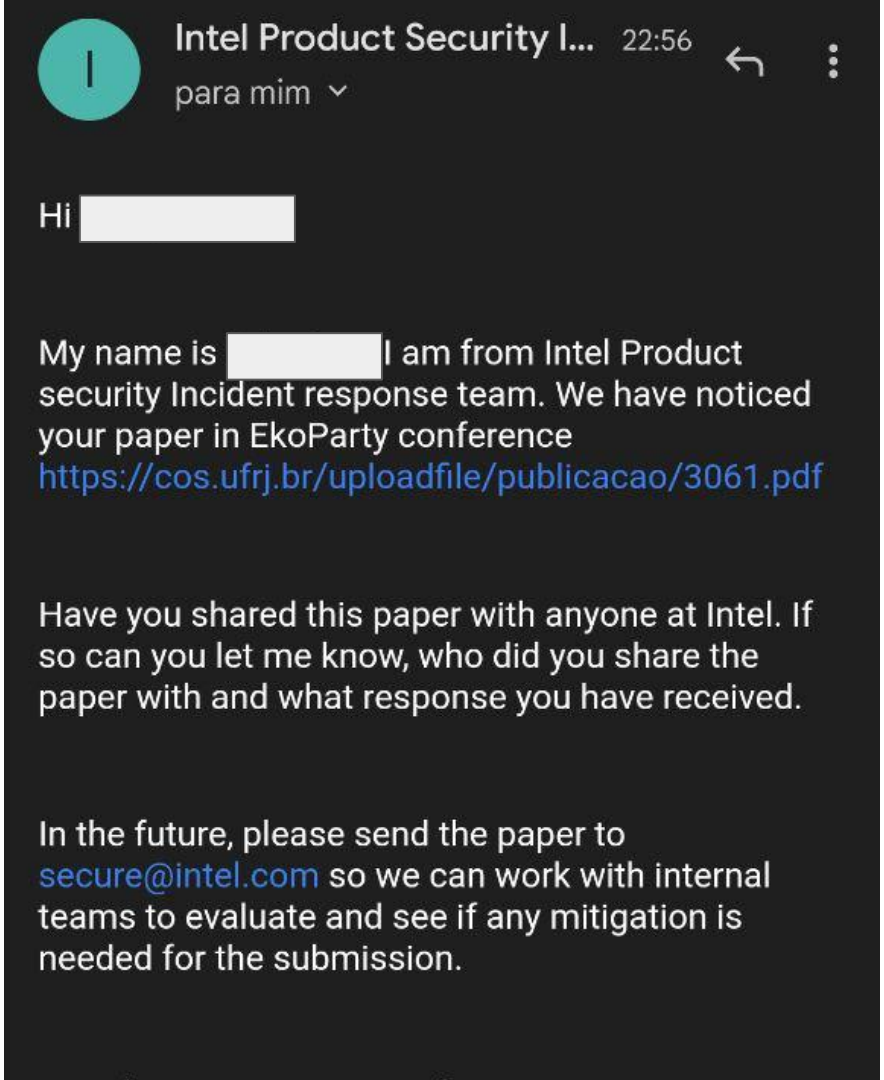
March/31/2022

[redacted]@intel.com>
Sent: Thursday, March 31, 2022 3:42 PM
To: [redacted]
Cc: Intel Product Security Incident Response Team <Intel.Product.Security.Incident.Response.Team@intel.com>
Subject: PTK0002446 Reverse Branch Target Buffer Poisoning

Hello [redacted]

We have evaluated your paper for impact beyond Spectre V2 and we believe this is a Spectre V2 exploit, and is used to attack (break) ASLR. At this time we believe this problem investigation is completed from the Intel perspective. We believe the mitigations Intel currently provides for Spectre V2 apply in this case and no further work from Intel is needed. Please do let us know when you plan to

Nov/29/2022



Intel Product Security I... 22:56

para mim

Hi [redacted]

My name is [redacted] I am from Intel Product security Incident response team. We have noticed your paper in EkoParty conference https://cos.ufrj.br/uploadfile/publicacao/3061.pdf

Have you shared this paper with anyone at Intel. If so can you let me know, who did you share the paper with and what response you have received.

In the future, please send the paper to secure@intel.com so we can work with internal teams to evaluate and see if any mitigation is needed for the submission.

Nov/29/2022


Recall: Reverse Branch Target Buffer Poisoning
Caixa de entrada
I
Intel Product Security I... 23:24
para mim
Intel Product Security Incident Response Team would like to recall the message, "Reverse Branch Target Buffer Poisoning".

Side-channels are irrelevant, Intel/AMD are good at everything else (security-wise I mean, forget delays, quality)…

- Wasn't it AMD SEV that had basic encryption problems? (not even considering the fact that registers were not encrypted and their whitepaper claimed that any memory write would lead to a crash only?)
- What about AMD's fTPM?
- What about all the UEFI flaws? Malicious actors are noticing more and more
  - Binarly finally gave an industry-wide view/ability for issues (300+ found in one year)
  - IOActive and NCC Group researcher have been constantly finding things
- What about the Boot Guard leaked keys from OEMs (that somehow someone from Intel PSIRT claimed it is an OEM problem?)
- What about transparency? (ucode updates without any specific info on what they fix)
- What about Taviso's recent work (reproducing an AMD errata and demonstrating that AMD forgot to patch in many affected systems?)

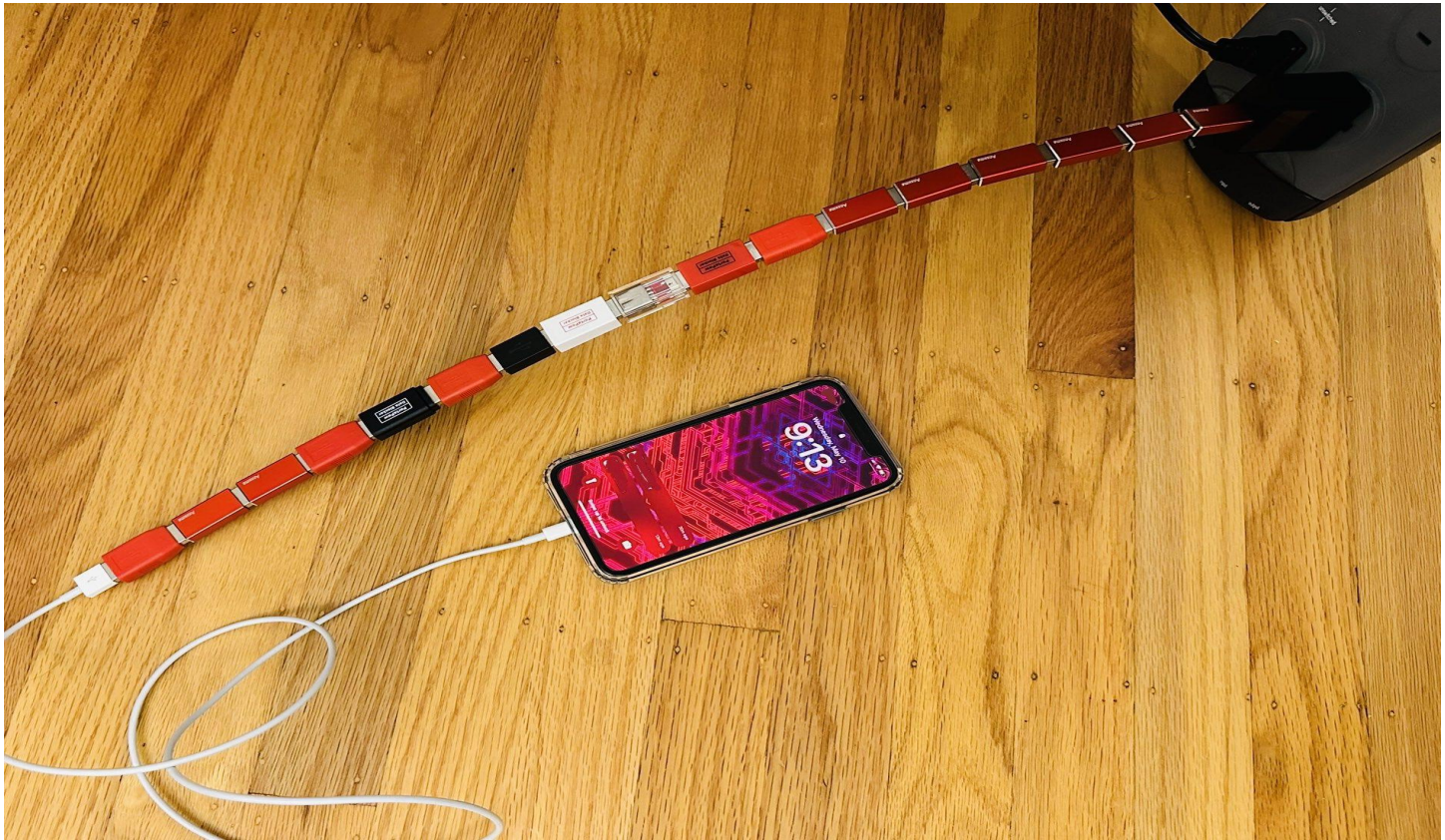
IS THIS REALLY ACCEPTABLE? ISN'T IT TIME THAT WE CHANGE THOSE FOLKS IN CHARGE?

Does the world keep re-discovering things and still accepting them?

The truth is, the most interesting bugs went completely ignored…

- CVE-2019-0151 and CVE-2019-0152 (IOMMU issue that somehow is published as SMM/TXT) - even got a pwnied awards somehow
- Issue found by Gabriel Barbosa and myself on how ucode cache works… given it does not keep mode information (instructions with different semantics between user-mode and kernel-mode end-up can't be differentiated) - MPX was the only case found, very irrelevant but super interesting, became an errata (details here)
- Open Source Security Inc. (grsecurity folks) also disclosed a nice errata that they spotted in a customer
- Look at the embedded graphics ones that Gabriel and I were involved (hint: they were not Windows DRIVERS issues) :)

Confidential Computing - “Trust me” / Defense-in-Depth



“TDX is built on the solid foundation of SGX”

- Is it? I’ve challenged Intel’s statement
- Let’s see:
  - SGX is implement in client, not servers (so the MPH is different)
  - SGX uses MEE as the encryption mechanism, that has integrity and replay protection (backed by an internal SRAM). TDX does not even have integrity/replay protection
  - While server and client microcode are unique, there are branches and technologies that are only supported on one or another (like FuSA requirements for Client SoCs and RAS features for Servers)
  - XuCode logic is different, given that:
    - mcheck has different requirements for client/server (like multi-package, hot plugging)
    - the persistent part is different, given that TDX ISA is different
  - Debugging/unlocking capabilities are different, with different base IPs
- So what is the solid foundation?
  - Above ISA architectural enclaves?
  - They are literally just software
- AWS has a nice take on Confidential Computing that is beyond jargons

Sec researchers and acade

The important interruption: I have to conclude, time is up :)

Conclusions

- Low-level research is not complicated, it is made to look complicated (the same way that in the past folks used complexity as an excuse to not improve things) - I honestly believe it is way simpler than modern exploitation on major OSes/software
- We gave a lot of power to a few individuals in the PSIRTs and in the Gang of 20 (part of key companies) and, because their unawareness of security (they are from marketing or developers), they are making the same mistakes that we've seeing in the 90s (like attacking researchers instead of trying to collaborate, assuming that collaboration = submission/acceptance)
- I didn't even discuss supply chain considerations for HW security (and how more and more things like secureboot on different devices will matter) - go look at storage controllers, switches, etc (if Intel/AMD still make trivial mistakes…)

Future - as if I could foresee it (1/3)

- The rise of confidential computing will bring new reliability problems (and hide even further compromises) - a lot of technology has yet to be developed - anyone jumping into using it should not lie to themselves they are doing it for security reasons :)

- Cloud providers are a national security risk
  - Too many secrets concentrated in one technology set
  - Research on attacks leveraging multi-tech capabilities are still non-existent
  - They can do way more than standalone companies, but savings at scale, problems in scaling talent, and others also create new challenges

Future - as if I could foresee it (2/3)

- Supply Chain security and hardware inspection specifically will become a major issue/topic

- BIOS/UEFI (and other platform firmware) will become more and more targets
  - Binarly and LVFS responded to part of my ask years ago (Troopers Keynote) to have a way to do industry-wide searches
  - We still need to fulfill the second half, which was for users/companies all over to upload their images for comparison with images on systems in other regions

Future - as if I could foresee it (3/3)

- Side-channels can be mitigated by 3 major techniques
  - Flushes (of secrets, of untrusted control entries)
  - Serialization/Stopping speculation (at key points, at certain flows, to stop speculation) - it can be a serializing instruction (like lfence), disabling it entirely (like IBRS) or marking things as non-cacheable at all
  - Partitioning (as a way to make the other two performant - like AWS using ASI-like in their hypervisor and not been affected by most of the side-channels that came after Variants 1-3)

- But the ideal implementation should consider the entire threat model and consider side-channels as *one* element
  - Intel did that early on with the endbranch change on CET or the non-cacheability for MPK (too little)
  - **Open Source Security Inc. (grsecurity)** folks went above and beyond (as usual) with their integration to full CFI (and obviously doing actual research, which led to findings like Straight Line Speculation)

Questions !? I will be around for the entirety of the con