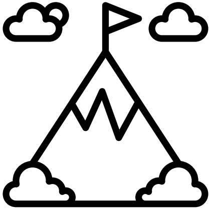
A Systematic Evaluation of Transient Execution Attacks and Defenses

Claudio Canella (@cc0x1f)[1], **Jo Van Bulck**[2], **Michael Schwarz**[1], **Moritz Lipp**[1], **Benjamin von Berg**[1], **Philipp Ortner**[1], **Frank Piessens**[2], **Dmitry Evtyushkin**, **Daniel Gruss**[1]

August 14, 2019

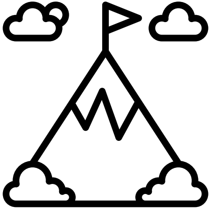
[1] Graz University of Technology, [2] imec-DistriNet, KU Leuven, [3] College of William and Mary

Motivation



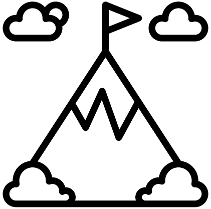
- Clear up naming confusion

Motivation



- Clear up naming confusion
- Systematic analysis shows new variants

Motivation



- Clear up naming confusion
- Systematic analysis shows new variants
- Show defenses cost performance and do not fully work

Motivation

- Clear up naming confusion
- Systematic analysis shows new variants
- Show defenses cost performance and do not fully work
- Gadget prevalence in Linux kernel

Meltdown

- CPU uses data in out-of-order execution before permission check

Meltdown

- CPU uses data in out-of-order execution before permission check
- Meltdown can read any kernel address

Meltdown

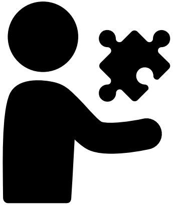
- CPU uses data in out-of-order execution before permission check
- Meltdown can read any kernel address
- Physical memory is usually mapped in kernel

Meltdown

- CPU uses data in out-of-order execution before permission check
- Meltdown can read any kernel address
- Physical memory is usually mapped in kernel

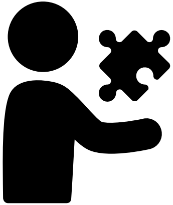
→ Read arbitrary memory

Problem Solved?



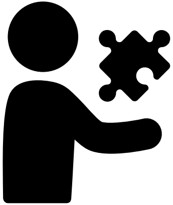
- Meltdown fully mitigated in software

Problem Solved?



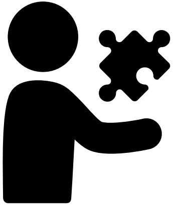
- Meltdown fully mitigated in software
- Problem seemed to be solved

Problem Solved?



- Meltdown fully mitigated in software
- Problem seemed to be solved
- No attack surface left

Problem Solved?



- Meltdown fully mitigated in software
- Problem seemed to be solved
- No attack surface left
- That is what everyone thought

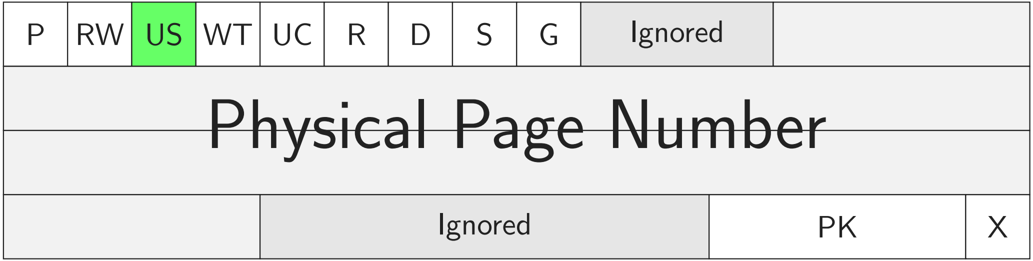
Generalization

- Meltdown is a whole category of vulnerabilities

Generalization

- Meltdown is a whole category of vulnerabilities
- Not only the user-accessible check

Page Table Entry



P RW US WT UC R D S G Ignored

Physical Page Number

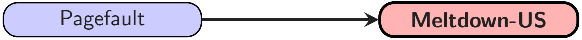
Ignored PK X

- User/Supervisor bit defines in which privilege level the page can be accessed

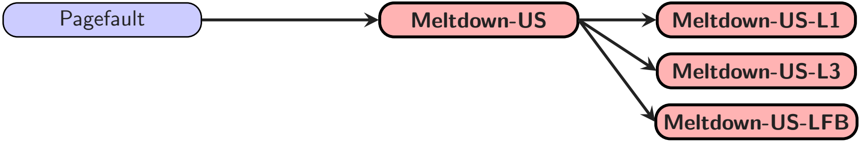
Meltdown Variants

Pagefault

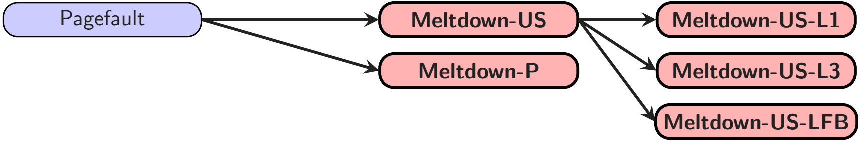
Meltdown Variants



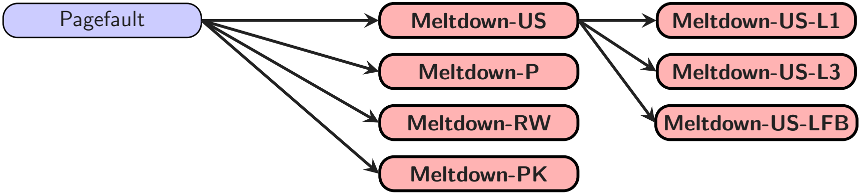
Meltdown Variants



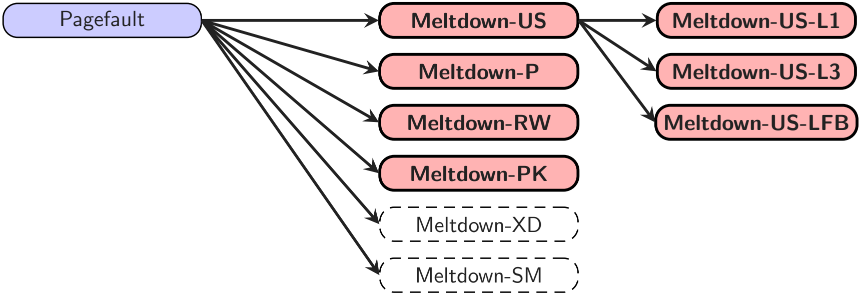
Meltdown Variants



Meltdown Variants



Meltdown Variants



Pagefault

Meltdown-US

Meltdown-P

Meltdown-RW

Meltdown-PK

Meltdown-XD

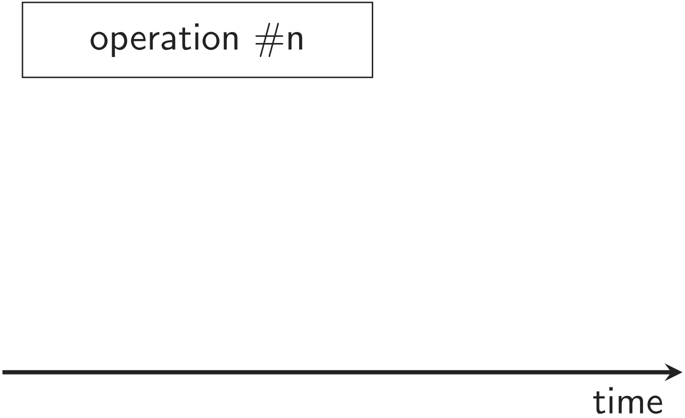
Meltdown-SM

Meltdown-US-L1

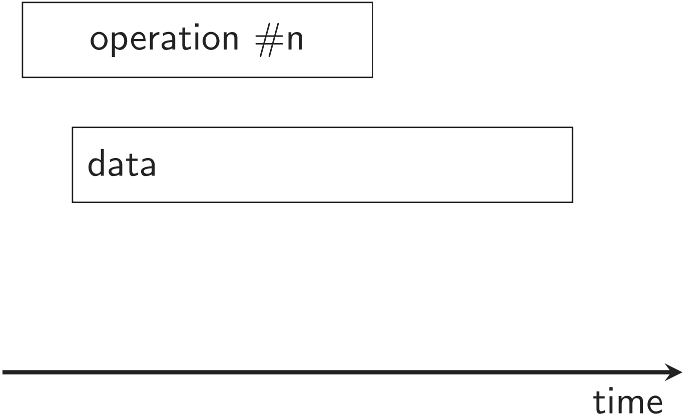
Meltdown-US-L3

Meltdown-US-LFB

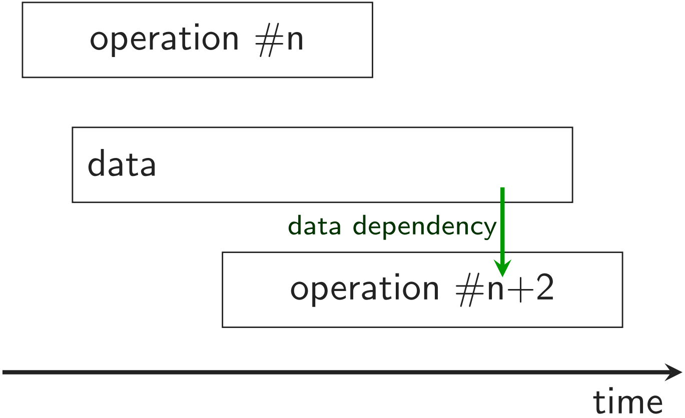
Meltdown Root Cause



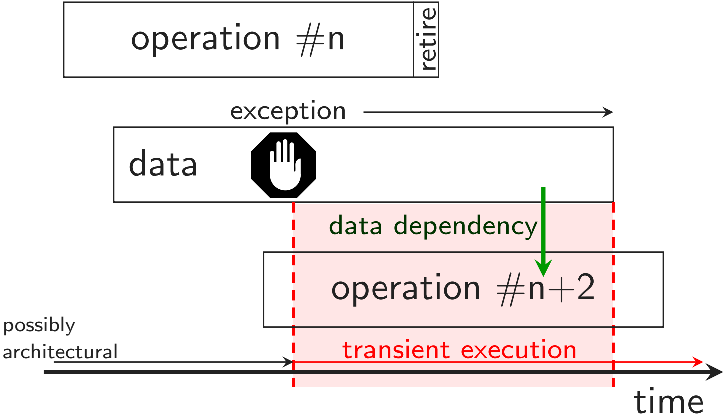
Meltdown Root Cause



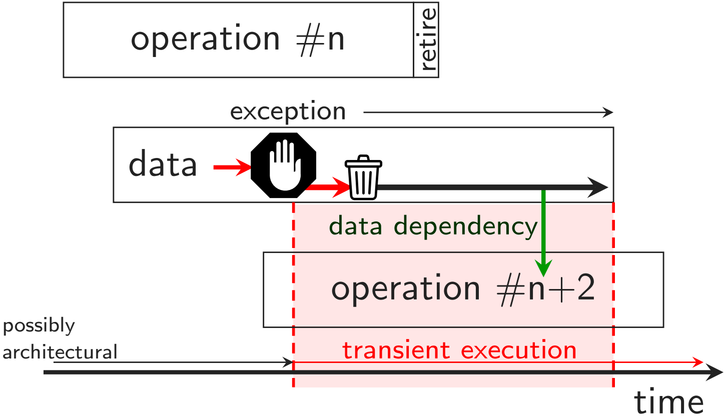
Meltdown Root Cause



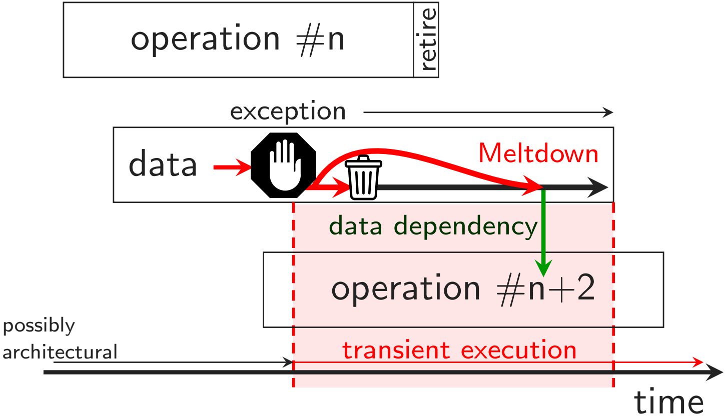
Meltdown Root Cause



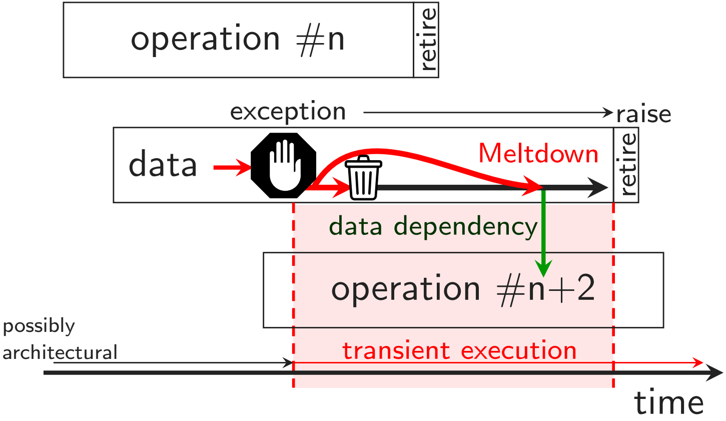
Meltdown Root Cause



Meltdown Root Cause



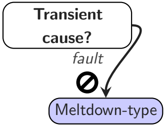
Meltdown Root Cause



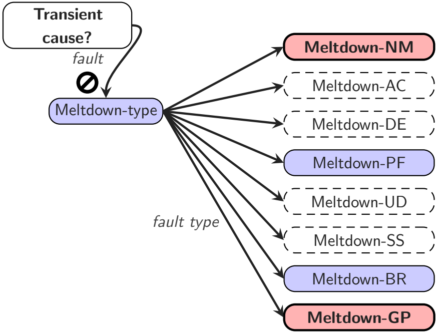
Meltdown Tree

Transient cause?

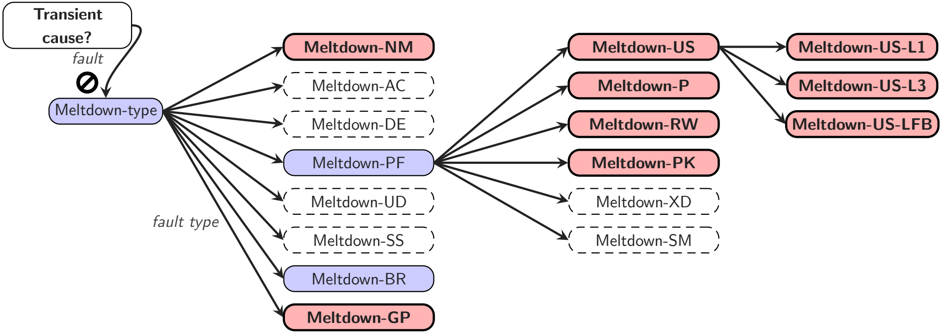
Meltdown Tree



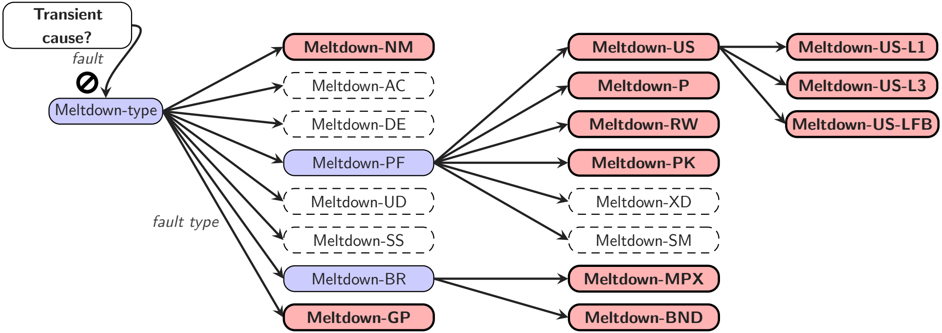
Meltdown Tree



Meltdown Tree



Meltdown Tree



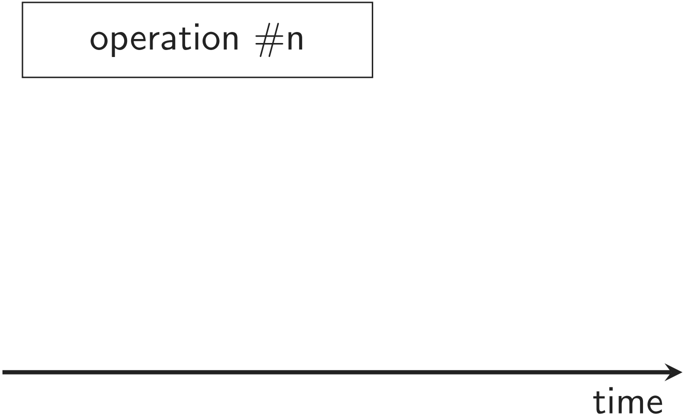
Spectre

- Spectre is a second class of transient execution attack

Spectre

- Spectre is a second class of transient execution attack
- Instead of faults, exploit control (or data) flow predictions

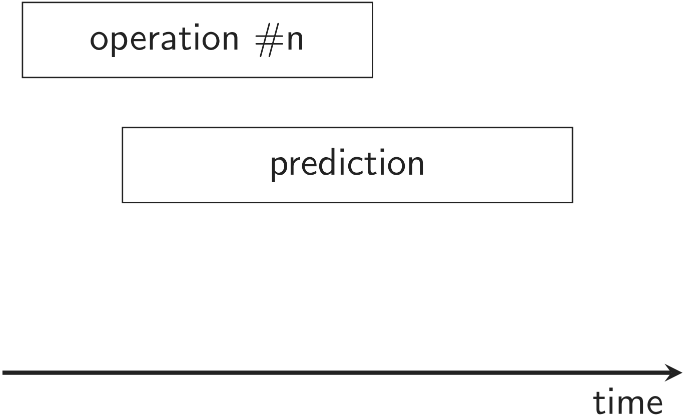
Spectre Root Cause



operation #n

time

Spectre Root Cause

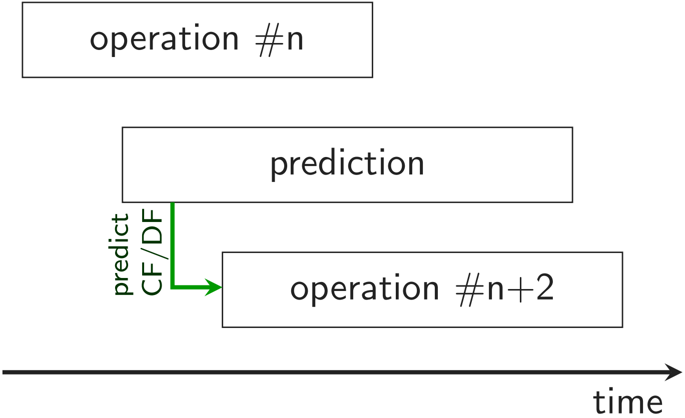


operation #n

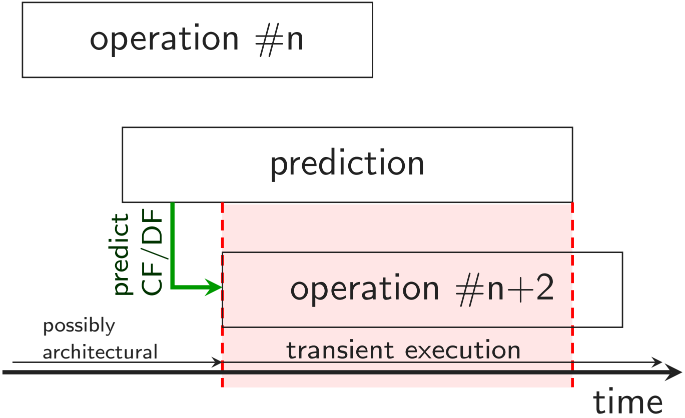
prediction

time

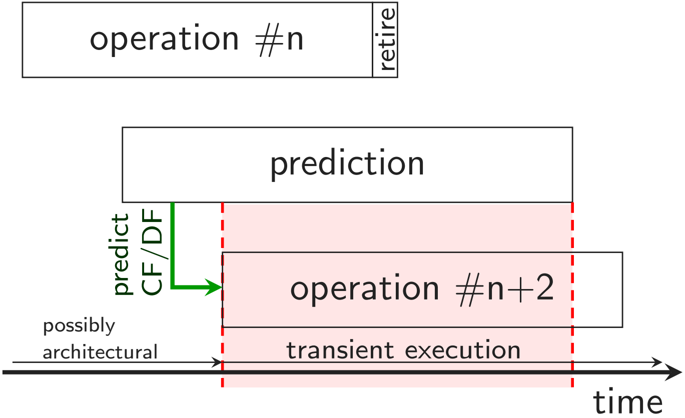
Spectre Root Cause



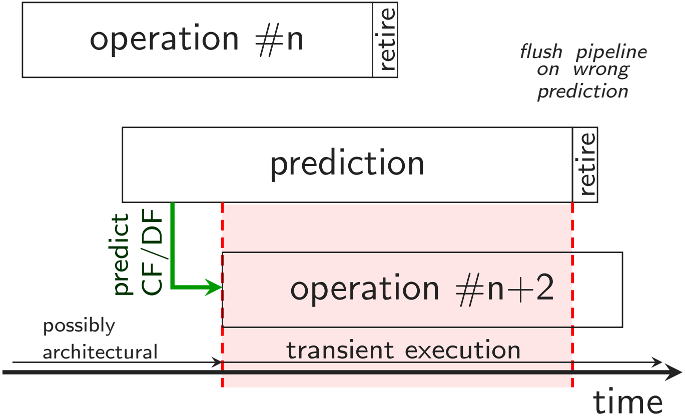
Spectre Root Cause



Spectre Root Cause

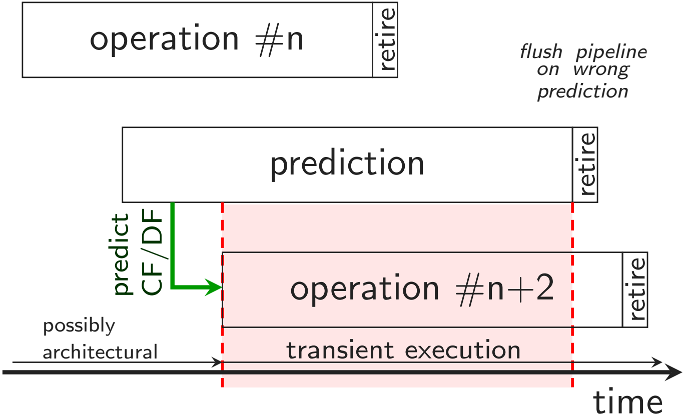


Spectre Root Cause



C. Canella, J. Van Bulck, M. Schwarz, M. Lipp, B. von Berg, P. Ortner, F. Piessens, D. Evtyushkin, D. Gruss

Spectre Root Cause



Spectre Root Cause

- Many predictors in modern CPUs

Spectre Root Cause

- Many predictors in modern CPUs
    - Branch taken/not taken (PHT)

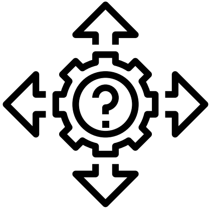
Spectre Root Cause

- Many predictors in modern CPUs
  - Branch taken/not taken (PHT)
  - Call/Jump destination (BTB)

Spectre Root Cause

- Many predictors in modern CPUs
    - Branch taken/not taken (PHT)
    - Call/Jump destination (BTB)
    - Function return destination (RSB)

Spectre Root Cause

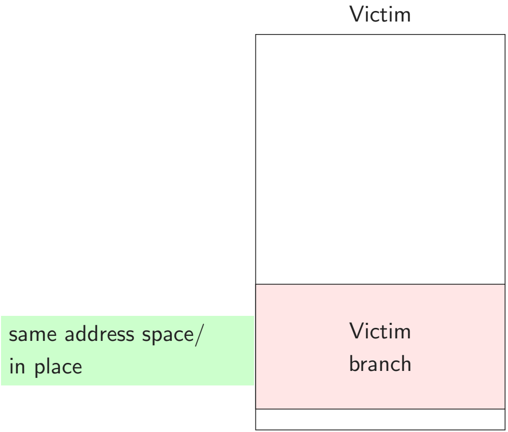


- Many predictors in modern CPUs
  - Branch taken/not taken (PHT)
  - Call/Jump destination (BTB)
  - Function return destination (RSB)
  - Load matches previous store (STL)

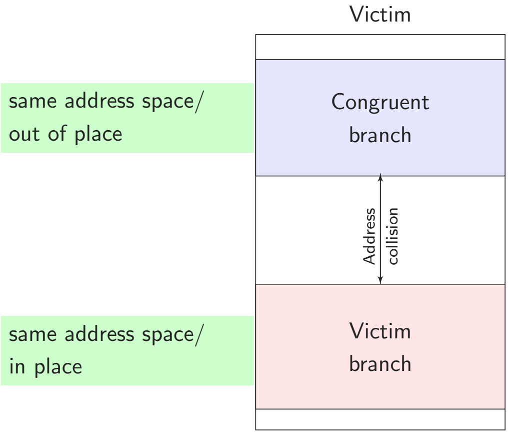
Spectre Root Cause

- Many predictors in modern CPUs
  - Branch taken/not taken (PHT)
  - Call/Jump destination (BTB)
  - Function return destination (RSB)
  - Load matches previous store (STL)
- Most are even shared among processes

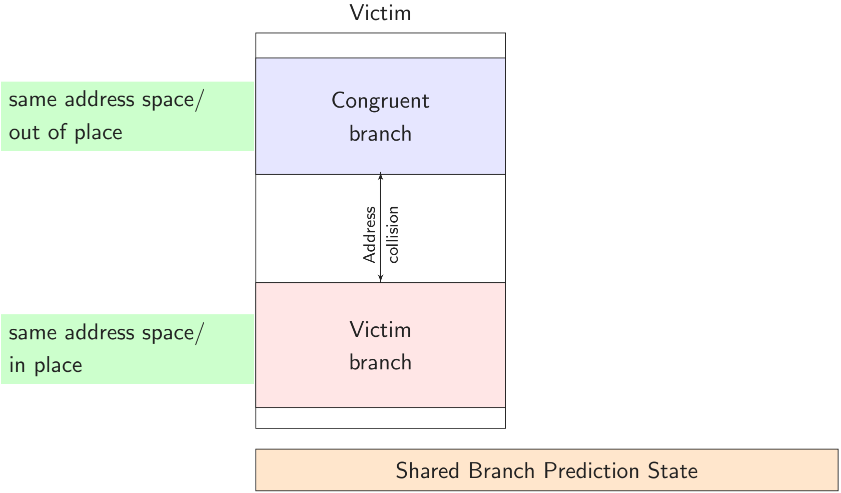
Spectre Mistraining



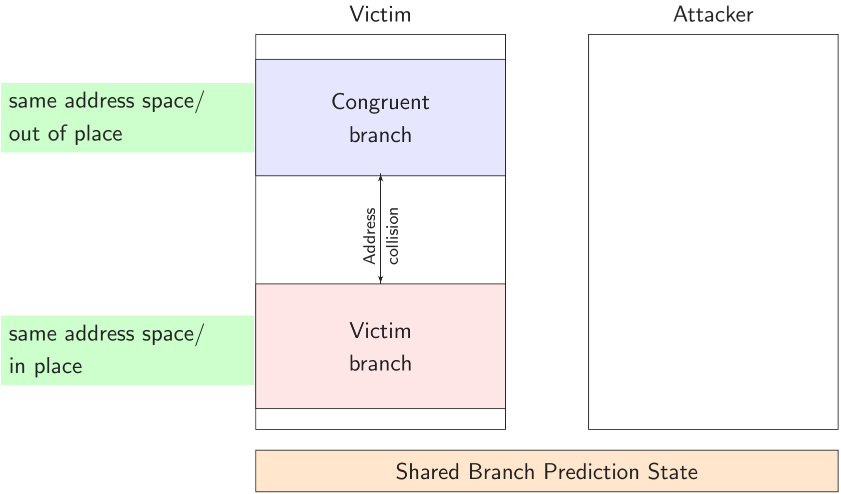
Spectre Mistraining



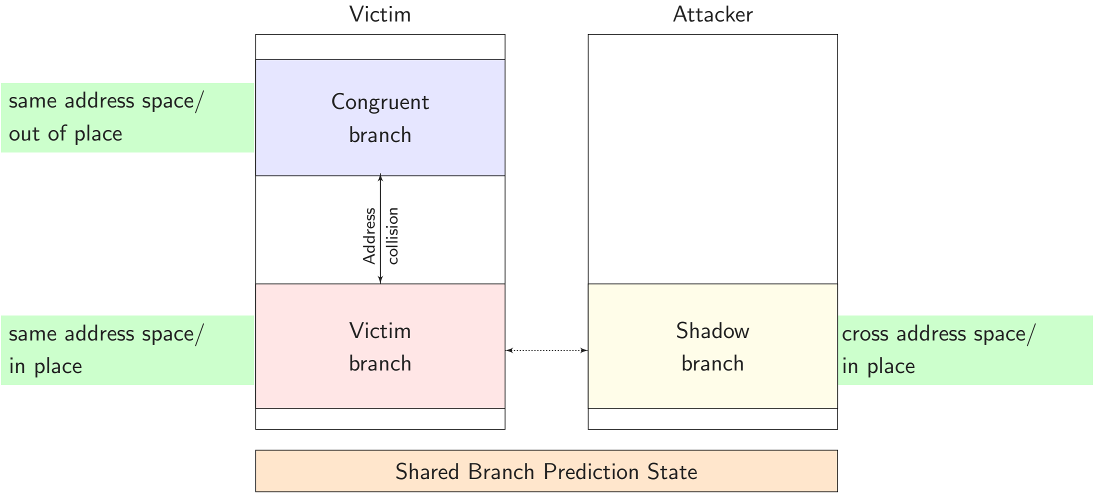
Spectre Mistraining



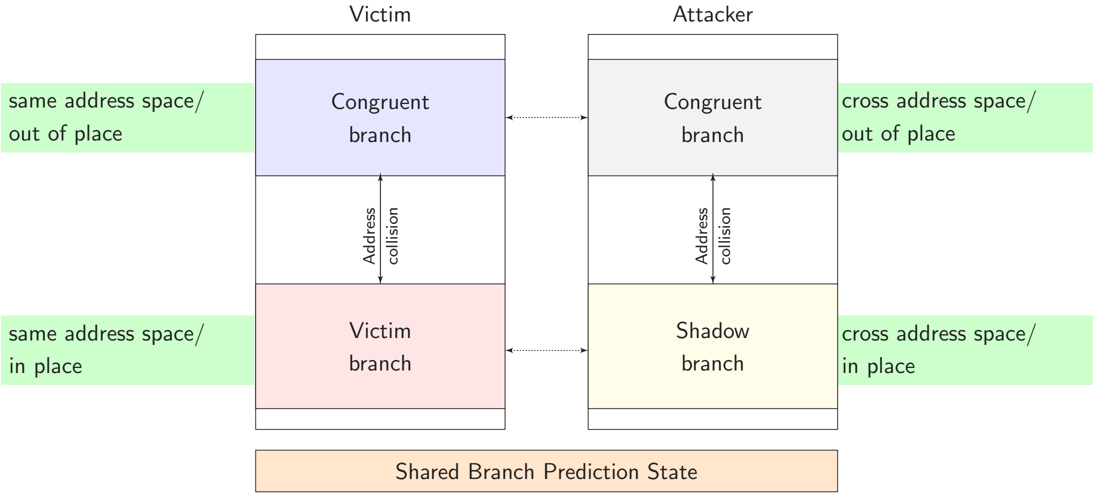
Spectre Mistraining



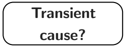
Spectre Mistraining



Spectre Mistraining

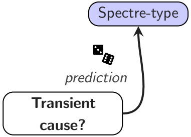

Victim
Attacker
same address space/ out of place
Congruent branch
Congruent branch
cross address space/ out of place
Address collision
Address collision
same address space/ in place
Victim branch
Shadow branch
cross address space/ in place
Shared Branch Prediction State

Spectre Variants



Transient cause?

Spectre Variants

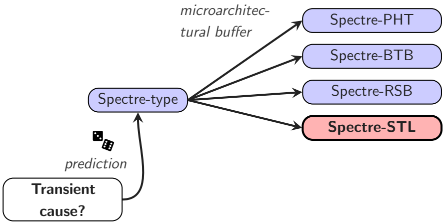


Spectre-type

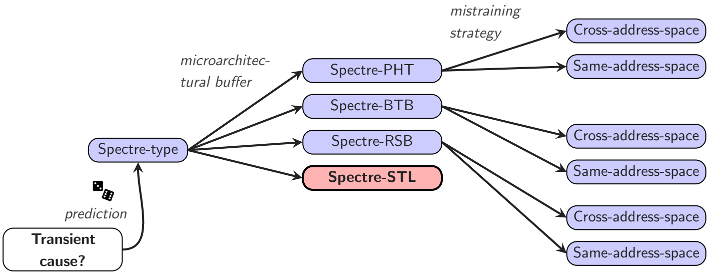
prediction

Transient cause?

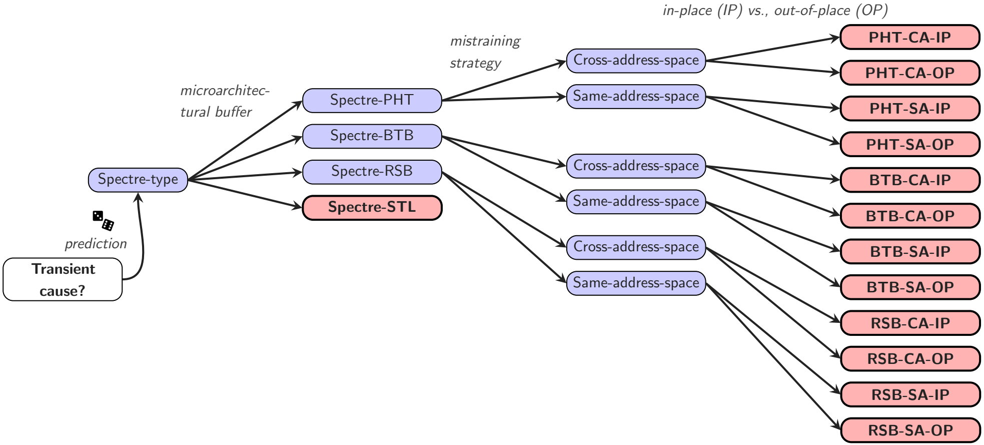
Spectre Variants



Spectre Variants



Spectre Variants



C. Canella, J. Van Bulck, M. Schwarz, M. Lipp, B. von Berg, P. Ortner, F. Piessens, D. Evtyushkin, D. Gruss

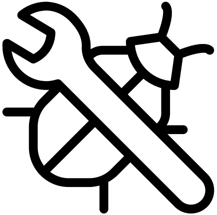
Spectre Fix

- Spectre is not a bug

Spectre Fix

- Spectre is not a bug
- It is an useful optimization

Spectre Fix



- Spectre is not a bug
- It is an useful optimization

→ Cannot simply fix it (as with Meltdown)

Spectre Fix

- Spectre is not a bug
- It is an useful optimization

→ Cannot simply fix it (as with Meltdown)

- Workarounds for critical code parts

Spectre Defense Categorization

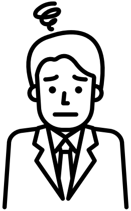
Spectre defenses in 3 categories:

C1 Mitigating or reducing the accuracy of covert channels

C2 Mitigating or aborting speculation

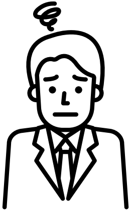
C3 Ensuring secret data cannot be reached

Spectre Misconceptions



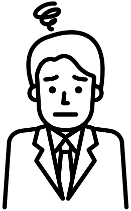
- Many countermeasures only consider the cache to get data...

Spectre Misconceptions



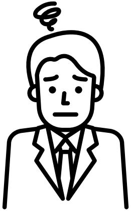
- Many countermeasures only consider the cache to get data...
- ...but there are other possibilities, e.g.,

Spectre Misconceptions



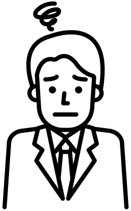
- Many countermeasures only consider the cache to get data...
- ...but there are other possibilities, e.g.,
    - Port contention (SMoTherSpectre)

Spectre Misconceptions



- Many countermeasures only consider the cache to get data...
- ...but there are other possibilities, e.g.,
  - Port contention (SMoTherSpectre)
  - AVX (NetSpectre)

Spectre Misconceptions



- Many countermeasures only consider the cache to get data...
- ...but there are other possibilities, e.g.,
  - Port contention (SMoTherSpectre)
  - AVX (NetSpectre)
- Cache is just the easiest

Spectre Defenses: Microarchitectural Target

	Defense	InvisiSpec	SafeSpec	DAWG	Taint Tracking	Timer Reduction	RSB Stuffing	Retpoline	SLH	YSNB	IBRS	STIPB	IBPB	Serialization	Sloth	SSBD/SSBB	Poison Value	Index Masking	Site Isolation
Microarchitectural Element	Cache																		
	TLB																		
	BTB																		
	BHB																		
	PHT																		
	RSB																		
	AVX																		
	FPU																		
	Execution Ports																		
	Category:	**C1**					**C2**										**C3**		

Considers element(●), partially considers it/same technique possible (◐), or does not consider it(○).

C. Canella, J. Van Bulck, M. Schwarz, M. Lipp, B. von Berg, P. Ortner, F. Piessens, D. Evtyushkin, D. Gruss

Spectre Defenses: Microarchitectural Target

Microarchitectural Element \ Defense	InvisiSpec	SafeSpec	DAWG	Taint Tracking	Timer Reduction	RSB Stuffing	Retpoline	SLH	YSNB	IBRS	STIPB	IBPB	Serialization	Sloth	SSBD/SSBB	Poison Value	Index Masking	Site Isolation
Cache	●	●	●										●					
TLB		●																
BTB							●			●	●	●						
BHB																		
PHT																		
RSB						●	●											
AVX																		
FPU																		
Execution Ports																		
Category:	**C1**					**C2**										**C3**		

Considers element(●), partially considers it/same technique possible (◐), or does not consider it(○).

C. Canella, J. Van Bulck, M. Schwarz, M. Lipp, B. von Berg, P. Ortner, F. Piessens, D. Evtyushkin, D. Gruss

Spectre Defenses: Microarchitectural Target

Microarchitectural Element / Defense	InvisiSpec	SafeSpec	DAWG	Taint Tracking	Timer Reduction	RSB Stuffing	Retpoline	SLH	YSNB	IBRS	STIPB	IBPB	Serialization	Sloth	SSBD/SSBB	Poison Value	Index Masking	Site Isolation
Cache	●	●	●										●					
TLB	◐	●	◐															
BTB							●			●	●	●						
BHB																		
PHT																		
RSB						●	●											
AVX																		
FPU																		
Execution Ports																		
Category:	**C1**					**C2**										**C3**		

Considers element(●), partially considers it/same technique possible (◐), or does not consider it(○).

C. Canella, J. Van Bulck, M. Schwarz, M. Lipp, B. von Berg, P. Ortner, F. Piessens, D. Evtyushkin, D. Gruss

Spectre Defenses: Microarchitectural Target

Microarchitectural Element

Defense	InvisiSpec	SafeSpec	DAWG	Taint Tracking	Timer Reduction	RSB Stuffing	Retpoline	SLH	YSNB	IBRS	STIPB	IBPB	Serialization	Sloth	SSBD/SSBB	Poison Value	Index Masking	Site Isolation
Cache	●	●	●	○	○	○	○	○	○	○	○	○	●	○	○	○	○	○
TLB	◐	●	◐	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
BTB	○	○	○	○	○	○	●	○	○	●	●	●	○	○	○	○	○	○
BHB	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
PHT	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
RSB	○	○	○	○	○	●	●	○	○	○	○	○	○	○	○	○	○	○
AVX	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
FPU	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Execution Ports	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Category:	**C1**					**C2**										**C3**		

Considers element(●), partially considers it/same technique possible (◐), or does not consider it(○).

Spectre: Defense Analysis

	Defense Attack	InvisiSpec	SafeSpec	DAWG	RSB Stuffing	Retpoline	Poison Value	Index Masking	Site Isolation	SLH	YSNB	IBRS	STIPB	IBPB	Serialization	Taint Tracking	Timer Reductio	Sloth	SSBD/SSBB
Intel	Spectre-PHT																		
	Spectre-BTB																		
	Spectre-RSB																		
	Spectre-STL																		
ARM	Spectre-PHT																		
	Spectre-BTB																		
	Spectre-RSB																		
	Spectre-STL																		
AMD	Spectre-PHT																		
	Spectre-BTB																		
	Spectre-RSB																		
	Spectre-STL																		

Symbols show if an attack is mitigated (●), partially mitigated (◐), not mitigated (○), theoretically mitigated (■), theoretically impeded (◧), not theoretically impeded (□), or out of scope (◇).

C. Canella, J. Van Bulck, M. Schwarz, M. Lipp, B. von Berg, P. Ortner, F. Piessens, D. Evtyushkin, D. Gruss

Spectre: Defense Analysis

	Defense Attack	InvisiSpec	SafeSpec	DAWG	RSB Stuffing	Retpoline	Poison Value	Index Masking	Site Isolation	SLH	YSNB	IBRS	STIPB	IBPB	Serialization	Taint Tracking	Timer Reductio	Sloth	SSBD/SSBB
Intel	Spectre-PHT						●			●									
	Spectre-BTB					●						●							
	Spectre-RSB																		
	Spectre-STL																		●
ARM	Spectre-PHT						●			●									
	Spectre-BTB					●													
	Spectre-RSB																		
	Spectre-STL																		●
AMD	Spectre-PHT						●			●									
	Spectre-BTB					●													
	Spectre-RSB																		
	Spectre-STL																		●

Symbols show if an attack is mitigated (●), partially mitigated (◐), not mitigated (○), theoretically mitigated (■), theoretically impeded (◧), not theoretically impeded (□), or out of scope (◇).

Spectre: Defense Analysis

	Defense Attack	InvisiSpec	SafeSpec	DAWG	RSB Stuffing	Retpoline	Poison Value	Index Masking	Site Isolation	SLH	YSNB	IBRS	STIPB	IBPB	Serialization	Taint Tracking	Timer Reductio	Sloth	SSBD/SSBB
Intel	Spectre-PHT						●	◐	◐	●					◐		◐		
	Spectre-BTB					●			◐			●	◐	◐			◐		
	Spectre-RSB				◐				◐								◐		
	Spectre-STL								◐								◐		●
ARM	Spectre-PHT						●	◐	◐	●					◐		◐		
	Spectre-BTB					●			◐								◐		
	Spectre-RSB				◐				◐								◐		
	Spectre-STL								◐								◐		●
AMD	Spectre-PHT						●	◐	◐	●					◐		◐		
	Spectre-BTB					●			◐								◐		
	Spectre-RSB				◐				◐								◐		
	Spectre-STL								◐								◐		●

Symbols show if an attack is mitigated (●), partially mitigated (◐), not mitigated (○), theoretically mitigated (■), theoretically impeded (◧), not theoretically impeded (□), or out of scope (◇).

C. Canella, J. Van Bulck, M. Schwarz, M. Lipp, B. von Berg, P. Ortner, F. Piessens, D. Evtyushkin, D. Gruss

Spectre: Defense Analysis

	Defense Attack	InvisiSpec	SafeSpec	DAWG	RSB Stuffing	Retpoline	Poison Value	Index Masking	Site Isolation	SLH	YSNB	IBRS	STIPB	IBPB	Serialization	Taint Tracking	Timer Reductio	Sloth	SSBD/SSBB
Intel	Spectre-PHT						●	◐	◐	●	○				◐		◐		
	Spectre-BTB					●			◐			●	◐	◐			◐		
	Spectre-RSB				◐				◐								◐		
	Spectre-STL								◐								◐		●
ARM	Spectre-PHT						●	◐	◐	●	○				◐		◐		
	Spectre-BTB					●			◐								◐		
	Spectre-RSB				◐				◐								◐		
	Spectre-STL								◐								◐		●
AMD	Spectre-PHT						●	◐	◐	●	○				◐		◐		
	Spectre-BTB					●			◐								◐		
	Spectre-RSB				◐				◐								◐		
	Spectre-STL								◐								◐		●

Symbols show if an attack is mitigated (●), partially mitigated (◐), not mitigated (○), theoretically mitigated (■), theoretically impeded (◧), not theoretically impeded (□), or out of scope (◇).

Spectre: Defense Analysis

	Defense Attack	InvisiSpec	SafeSpec	DAWG	RSB Stuffing	Retpoline	Poison Value	Index Masking	Site Isolation	SLH	YSNB	IBRS	STIPB	IBPB	Serialization	Taint Tracking	Timer Reductio	Sloth	SSBD/SSBB
Intel	Spectre-PHT						●	◐	◐	●	○				◐	■	◐		
	Spectre-BTB					●			◐			●	◐	◐		■	◐		
	Spectre-RSB				◐				◐							■	◐		
	Spectre-STL								◐							■	◐	■	●
ARM	Spectre-PHT						●	◐	◐	●	○				◐	■	◐		
	Spectre-BTB					●			◐							■	◐		
	Spectre-RSB				◐				◐							■	◐		
	Spectre-STL								◐							■	◐	■	●
AMD	Spectre-PHT						●	◐	◐	●	○				◐	■	◐		
	Spectre-BTB					●			◐			■				■	◐		
	Spectre-RSB				◐				◐							■	◐		
	Spectre-STL								◐							■	◐	■	●

Symbols show if an attack is mitigated (●), partially mitigated (◐), not mitigated (○), theoretically mitigated (■), theoretically impeded (◧), not theoretically impeded (□), or out of scope (◇).

Spectre: Defense Analysis

Attack	Defense	InvisiSpec	SafeSpec	DAWG	RSB Stuffing	Retpoline	Poison Value	Index Masking	Site Isolation	SLH	YSNB	IBRS	STIPB	IBPB	Serialization	Taint Tracking	Timer Reductio	Sloth	SSBD/SSBB
Intel	Spectre-PHT						●	◐	◐	●	○				◐	■	◐	◧	
	Spectre-BTB					●			◐			●	◐	◐		■	◐		
	Spectre-RSB				◐				◐							■	◐		
	Spectre-STL								◐							■	◐	■	●
ARM	Spectre-PHT						●	◐	◐	●	○				◐	■	◐	◧	
	Spectre-BTB					●			◐							■	◐		
	Spectre-RSB				◐				◐							■	◐		
	Spectre-STL								◐							■	◐	■	●
AMD	Spectre-PHT						●	◐	◐	●	○				◐	■	◐	◧	
	Spectre-BTB					●			◐			■	◧	◧		■	◐		
	Spectre-RSB				◐				◐					◧		■	◐		
	Spectre-STL								◐							■	◐	■	●

Symbols show if an attack is mitigated (●), partially mitigated (◐), not mitigated (○), theoretically mitigated (■), theoretically impeded (◧), not theoretically impeded (□), or out of scope (◇).

C. Canella, J. Van Bulck, M. Schwarz, M. Lipp, B. von Berg, P. Ortner, F. Piessens, D. Evtyushkin, D. Gruss

Spectre: Defense Analysis

	Defense Attack	InvisiSpec	SafeSpec	DAWG	RSB Stuffing	Retpoline	Poison Value	Index Masking	Site Isolation	SLH	YSNB	IBRS	STIPB	IBPB	Serialization	Taint Tracking	Timer Reductio	Sloth	SSBD/SSBB
Intel	Spectre-PHT	□	□	□			●	◐	◐	●	○				◐	■	◐	◧	
	Spectre-BTB	□	□	□		●			◐			●	◐	◐		■	◐		
	Spectre-RSB	□	□	□	◐				◐							■	◐		
	Spectre-STL	□	□	□					◐							■	◐	■	●
ARM	Spectre-PHT	□	□	□			●	◐	◐	●	○				◐	■	◐	◧	
	Spectre-BTB	□	□	□		●			◐							■	◐		
	Spectre-RSB	□	□	□	◐				◐							■	◐		
	Spectre-STL	□	□	□					◐							■	◐	■	●
AMD	Spectre-PHT	□	□	□			●	◐	◐	●	○				◐	■	◐	◧	
	Spectre-BTB	□	□	□		●			◐			■	◧	◧		■	◐		
	Spectre-RSB	□	□	□	◐				◐					◧		■	◐		
	Spectre-STL	□	□	□					◐							■	◐	■	●

Symbols show if an attack is mitigated (●), partially mitigated (◐), not mitigated (○), theoretically mitigated (■), theoretically impeded (◧), not theoretically impeded (□), or out of scope (◇).

C. Canella, J. Van Bulck, M. Schwarz, M. Lipp, B. von Berg, P. Ortner, F. Piessens, D. Evtyushkin, D. Gruss

Spectre: Defense Analysis

	Defense Attack	InvisiSpec	SafeSpec	DAWG	RSB Stuffing	Retpoline	Poison Value	Index Masking	Site Isolation	SLH	YSNB	IBRS	STIPB	IBPB	Serialization	Taint Tracking	Timer Reductio	Sloth	SSBD/SSBB
Intel	Spectre-PHT	□	□	□	◇	◇	●	◐	◐	●	○	◇	◇	◇	◐	■	◐	◧	◇
	Spectre-BTB	□	□	□	◇	●	◇	◇	◐	◇	◇	●	◐	◐	◇	■	◐	◇	◇
	Spectre-RSB	□	□	□	◐	◇	◇	◇	◐	◇	◇	◇	◇	◇	◇	■	◐	◇	◇
	Spectre-STL	□	□	□	◇	◇	◇	◇	◐	◇	◇	◇	◇	◇	◇	■	◐	■	●
ARM	Spectre-PHT	□	□	□	◇	◇	●	◐	◐	●	○	◇	◇	◇	◐	■	◐	◧	◇
	Spectre-BTB	□	□	□	◇	●	◇	◇	◐	◇	◇	◇	◇	◇	◇	■	◐	◇	◇
	Spectre-RSB	□	□	□	◐	◇	◇	◇	◐	◇	◇	◇	◇	◇	◇	■	◐	◇	◇
	Spectre-STL	□	□	□	◇	◇	◇	◇	◐	◇	◇	◇	◇	◇	◇	■	◐	■	●
AMD	Spectre-PHT	□	□	□	◇	◇	●	◐	◐	●	○	◇	◇	◇	◐	■	◐	◧	◇
	Spectre-BTB	□	□	□	◇	●	◇	◇	◐	◇	◇	■	◧	◧	◇	■	◐	◇	◇
	Spectre-RSB	□	□	□	◐	◇	◇	◇	◐	◇	◇	◇	◇	◧	◇	■	◐	◇	◇
	Spectre-STL	□	□	□	◇	◇	◇	◇	◐	◇	◇	◇	◇	◇	◇	■	◐	■	●

Symbols show if an attack is mitigated (●), partially mitigated (◐), not mitigated (○), theoretically mitigated (■), theoretically impeded (◧), not theoretically impeded (□), or out of scope (◇).

C. Canella, J. Van Bulck, M. Schwarz, M. Lipp, B. von Berg, P. Ortner, F. Piessens, D. Evtyushkin, D. Gruss

Spectre Mitigating: Performance

Defense Evaluation	Penalty	Benchmark
KAISER/KPTI	0–2.6 %	System call rates
Retpoline	5–10 %	Real-world workload servers
Site Isolation	10–13 %	Memory overhead
InvisiSpec	22 %	SPEC
SafeSpec	-3 %	SPEC on MARSSx86
DAWG	1–15 %	PARSEC , GAPBS
SLH	29–36.4 %	Google microbenchmark suite
YSNB	60 %	Phoenix
IBRS	20–30 %	Sysbench 1.0.11
STIBP	30–50 %	Rodinia OpenMP, DaCapo
Serialization	62–74.8 %	Google microbenchmark suite
SSBD/SSBB	2–8 %	SYSmark 2018, SPEC integer
L1TF Mitigations	-3–31 %	SPEC

Spectre: Linux Kernel Gadget Prevalence

Gadget	Example (Spectre-PHT)	#Occurrences
Prefetch	`if(i<LEN_A){a[i];}`	172
Compare	`if(i<LEN_A){if(a[i]==k){};}`	127
Index	`if(i<LEN_A){y = b[a[i]*x];}`	0
Execute	`if(i<LEN_A){a[i](void);}`	16

Proof-of-Concept

You can find our proof-of-concept implementation and classification tree on:

- `https://github.com/IAIK/transientfail`
- `http://transient.fail/`

Recap

- Introduced a new naming scheme

Recap

- Introduced a new naming scheme
- Discovered new attack variants

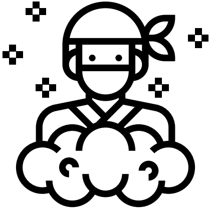
Recap

- Introduced a new naming scheme
- Discovered new attack variants
- Showed that defenses cost too much performance for little effect

Recap

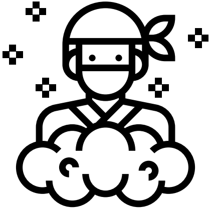
- Introduced a new naming scheme
- Discovered new attack variants
- Showed that defenses cost too much performance for little effect
- Showed prevalence of gadgets in Linux kernel

Conclusion



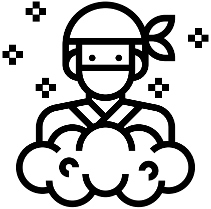
- Transient Execution Attacks are…

Conclusion



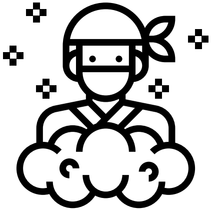
- Transient Execution Attacks are…
  - …a novel class of attacks

Conclusion



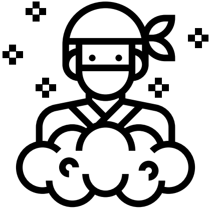
- Transient Execution Attacks are…
  - …a novel class of attacks
  - …extremely powerful

Conclusion



- Transient Execution Attacks are…
  - …a novel class of attacks
  - …extremely powerful
  - …only at the beginning

Conclusion



- Transient Execution Attacks are...
  - ...a novel class of attacks
  - ...extremely powerful
  - ...only at the beginning
- Many optimizations introduce side channels → now exploitable

A Systematic Evaluation of Transient Execution Attacks and Defenses

Claudio Canella (@cc0x1f)[1], **Jo Van Bulck**[2], **Michael Schwarz**[1], **Moritz Lipp**[1], **Benjamin von Berg**[1], **Philipp Ortner**[1], **Frank Piessens**[2], **Dmitry Evtyushkin**, **Daniel Gruss**[1]

August 14, 2019

[1] Graz University of Technology, [2] imec-DistriNet, KU Leuven, [3] College of William and Mary

Acknowledgments

We want to thank our shepherd, Jonathan McCune, for his helpful comments and suggestions that substantially helped in improving the paper. This work has been supported by the Austrian Research Promotion Agency (FFG) via the K-project DeSSnet, which is funded in the context of COMET – Competence Centers for Excellent Technologies by BMVIT, BMWFW, Styria and Carinthia. This work has been supported by the Austrian Research Promotion Agency (FFG) via the project ESPRESSO, which is funded by the province of Styria and the Business Promotion Agencies of Styria and Carinthia. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 681402). This research received funding from the Research Fund KU Leuven, and Jo Van Bulck is supported by the Research Foundation – Flanders (FWO). Evtyushkin acknowledges the start-up grant from the College of William and Mary. Additional funding was provided by generous gifts from ARM and Intel. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the funding parties.