Discussing speculation attacks

uSC SIG / RISCV

January 16, 2023

Introduction to Spectre-PHT in 1 slide

```
if (x < array1_size) {
    y = array2[array1[x] * 4096];
}
```

Steps

1. Mistrain the branch prediction mechanism to trigger speculative execution on an incorrect condition.
2. Read the secret at address `array1` + x, for an arbitrary x.
3. **Write** (part of) the secret in the tag field of the cache memory with a **read** memory access.
4. Read the secret with a cache timing analysis.

Step 3 and 4 can be replaced with any covert channel !

Section 2

Countermeasures

Disclaimers

- No speculative attack has been reported in the wild.
- Some papers only consider cache based covert channels, this is not enough for an actual solution.
- Speculative attacks can be solved with hardware modifications only. $\rightarrow$ no RISCV ISA extension *required*, but may help performances.

Strategies from a bird's view

1. Add dedicated microstructures to deal with speculation. Ex: Invispec speculatively load data in a *speculation buffer* instead of the cache.
2. Defer sensitive operations to prevent speculatively executing them.

In the *defer* strategy, countermeasures often implement hardware taint tracking to choose what instructions to delay.

A lot of questions

What are the costs (area, time, power) of these strategies ? Are these costs definitive or implementation dependent ?

Self-reported slowdown[1]

Countermeasure	Slowdown	Main strategy
InvisiSpec	$21\% - 72\%$	Structures
STT	$8\% - 15\%$	Defer & tainting
SafeSpec	-3%	Structures
NDA	$10\% - 125\%$	Defer & tainting
Dolma	$9\% - 63\%$	Defer & tainting
SpecShield	$10\% - 73\%$	Defer & tainting
SpecTerminator	$2.6\% - 6\%$	Defer & tainting

[1] After a quick read, without judging the security merits nor the veracity of reported slowdowns.

NDA self-evaluation

	Mechanism	Control steering (memory)	Control steering (GPRs)	Chosen code	Overhead vs. OoO
1	Perm. propagation	☐			10.7%
2	Perm. propagation+BR	■			22.3%
3	Strict propagation	☐	◇		36.1%
4	Strict propagation+BR	■	◇		45%
5	Load restriction	■		■	100%
6	Full protection (4+5)	■	◇	■	125%
7	InvisiSpec-Spectre*	○	○		7.6%
8	InvisiSpec-Future*	○	○	○	32.7%

■ Defeats all covert channels ○ Defeats d-cache based attacks
☐ Defeats all covert channels, but does not block SSB
◇ Defeats all covert channels, except single micro-op GPR-attacks
* Our evaluation of InvisiSpec[69] on SPEC 2017 is detailed in §6.1

Table 2: *NDA* propagation policies (rows 1-6) and the attacks they prevent. Bypass Restriction (BR) adds protection against SSB (Spectre v4). Special registers, such as AVX and MSRs (LazyFP [59] and Spectre v3a [27]), are protected by treating their accesses like loads. None of the 25 documented attacks [8, 12] leak data from GPRs nor without at least two dependent micro-ops.