Assignment III – Meltdown & Spectre

Team quench-quokka
Mihai-George Licu
Changling Wang

Overview

Meltdown

- Intel pipeline can execute instructions transiently before checking the permissions
- The memory accesses remain cached in memory

Spectre

- Branch prediction may execute incorrect branches transiently
- Attacker can exploit PHT to execute and leak information through side channels

Tasks 1 2 3

When the illegal instructions are retired the system raises a page fault

SEGV Handling

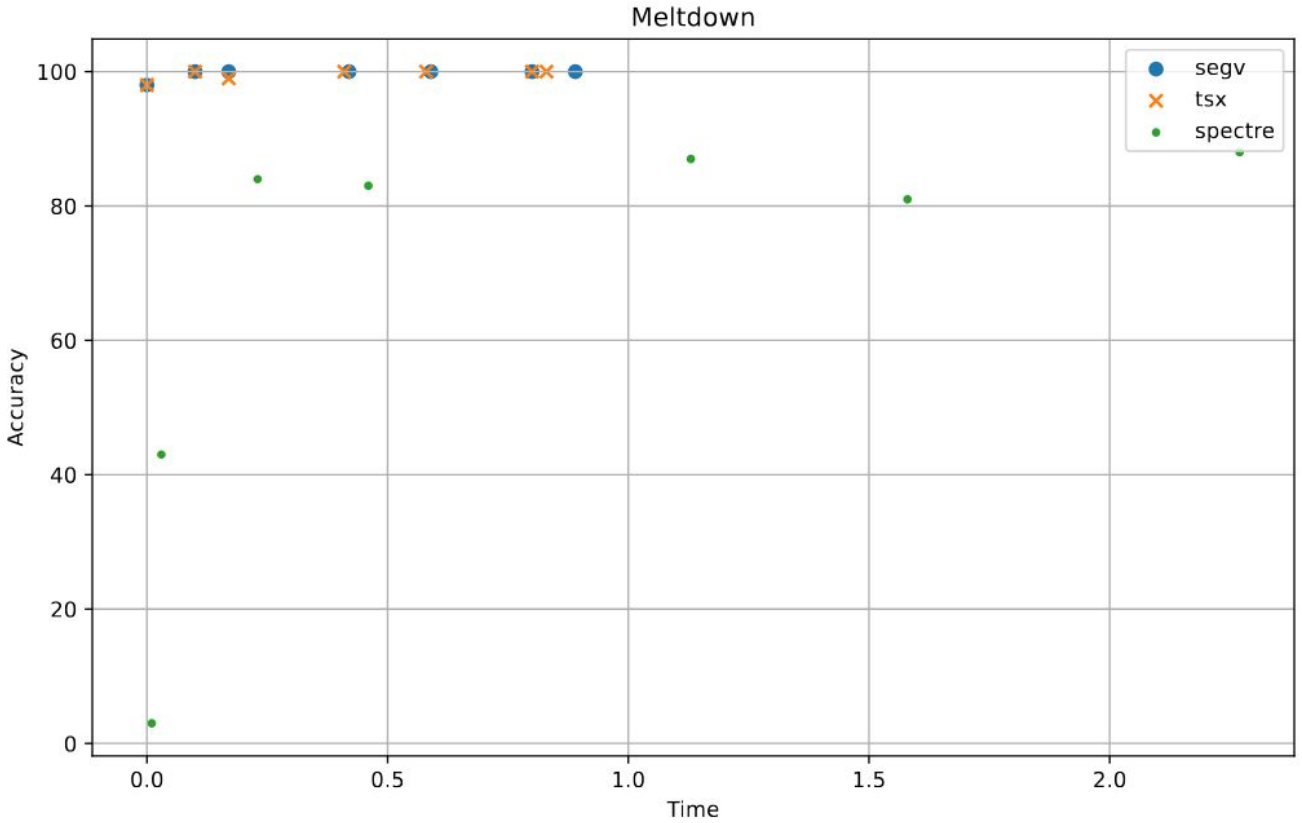
Write a handler for segfault that continues the program normally

TSX

Atomically execute the illegal memory access in one single transaction such that if it fails it will not stop execution

Spectre V1

Ensure the illegal memory access happens within a mispredicted branch that takes a long time to evaluate using multiple levels of memory reads



Segv and TSX solutions are stable even with little iterations

Spectre V1 variant requires multiple iterations for identifying a stable signal

Overview

Retbleed

- Indirect branch prediction falls back to BTB-based after RSB exhausted (underflow)
- Train certain BTB entry to point to a (provided) leaking gadget
- Execute the (provided) speculative gadget to trigger speculative execution of the leaking gadget

Tasks 4 5

When the illegal instructions are retired the system raises a page fault

Hugepage

`posix_memalign()` for hugepage-aligned allocation

`madvise(..,MADV_HUGEPAGE)` to advise/request for a transparent hugepage for reload_buffer

mprotect() for mark a buffer as executable, on which the BTB training function is loaded for out-of-place training

BTB Training

Position the BTB training function such that the lowest 21 bits of a `ret` instruction in it is identical to **`spec_gadget_return_address`**

Push the leaking gadget to stack, before pushing 29 copies of the address of its ret and starting to return

Leaking

Time&Reload to leak the secret byte by byte.