SEED SECURITY LABS

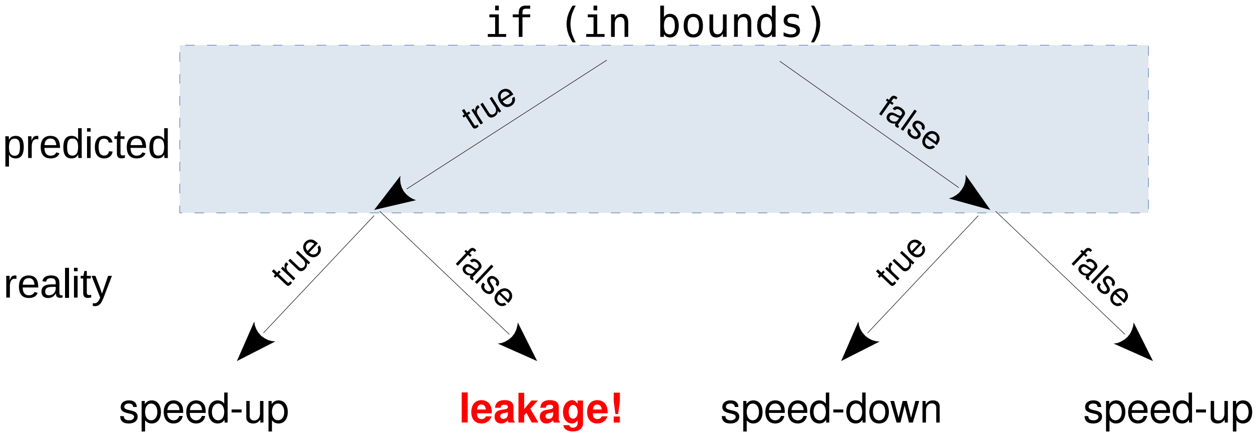
Spectre Attack Lab

General problem

- Modern microprocessors[1] perform branch prediction and speculative execution of instructions.
- So, they achieve (apparent) high execution speed when the prediction is true (what should happen a significant number of times).
- When prediction is wrong, the state of the processor is returned to the correct state, corresponding to the correct branch being taken;
  - unfortunately, some of the wrong leftovers are not deleted: processor cache is the common example (durable side effect).
- A subsequent probing of cache, may reveal secretive data (as it remained cached)!

[1] Intel, AMD, ARM...

...General problem



Before the correct outcome of the bounds check is known, the branch predictor causes the program to run toward the most likely branch target, leading to an overall execution speed-up if the outcome was correctly predicted. However, if the bounds check is incorrectly predicted as true, an attacker can leak secret information in certain scenarios. (based on Fig. 1 of "Spectre Attacks: Exploiting Speculative Execution")

Spectre attack procedure

- setting up
  - with knowledge of secretive code & data, mistrain processor prediction logic
  - flush relevant data from cache
- forcing prediction failure
  - with knowledge of secretive code & data, force speculative execution
  - as a result, cache will retain secretive data (although correct data will be provided in processor registers and memory)
- collecting secretive data by side-channels
  - typically, by timing the reading access to cache lines, secretive data is revealed, as is accessed faster
  - for minimizing spurious results[1] a statistical measurement procedure should be used

[1] A common computer runs "simultaneously" tens of processes!

Hardware bug

- CVE-2017-5715 / CVE-2017-5753
  - *«Systems with microprocessors utilizing speculative execution and indirect / direct branch prediction may allow unauthorized disclosure of information to an attacker with local user access via a side-channel analysis.»*
- Spectre Attacks: Exploiting Speculative Execution[1]
  - «(...) *Spectre attacks involve inducing a victim to speculatively perform operations that would not occur during correct program execution and which leak the victim's confidential information via a side channel to the adversary*. (...)»
  - «(...) *These attacks represent a serious threat to actual systems since vulnerable speculative execution capabilities are found in microprocessors from Intel, AMD, and ARM that are used in billions of devices*. (...)»

[1] original paper includes Spectre example implementation!

Spectre vulnerability observation @ 2026

- might not be possible (mitigating patches by microprocessor manufacturers...)
- however, it should be tried (original proof-of-concept code...)
- it was seen in 2025, on an HP EliteDesk 800 (2014 model, CZC4430GPZ)

  -
    ```
    $ uname -a
    Linux ricjoa 5.15.0-133-generic #144-Ubuntu SMP Fri Feb 7 20:47:38 UTC 2025 x86_64 x86_64 x86_64 GNU/Linux
    $ lscpu
    Architecture:             x86_64
      CPU op-mode(s):           32-bit, 64-bit
      ...
    CPU(s):                   8
      On-line CPU(s) list:    0-7
    Vendor ID:                GenuineIntel
      Model name:               Intel(R) Core(TM) i7-4790 CPU @ 3.60GHz
        CPU family:             6
        Model:                60
        ...
    Vulnerabilities:
      ...
      Spectre v1:             Mitigation; usercopy/swapgs barriers and __user pointer sanitization
      Spectre v2:             Mitigation; Retpolines; IBPB conditional; IBRS_FW; STIBP conditional; RSB filling; PBRSB-
         eIBRS Not affected; BHI Not affected
      ...
    ```