Spectre Attacks: Exploiting Speculative Execution

Paul Kocher\textsuperscript{1}, Jann Horn\textsuperscript{2}, Anders Fogh\textsuperscript{3}, Daniel Genkin\textsuperscript{4},
Daniel Gruss\textsuperscript{5}, Werner Haas\textsuperscript{6}, Mike Hamburg\textsuperscript{7}, Moritz Lipp\textsuperscript{5},
Stefan Mangard\textsuperscript{5}, Thomas Prescher\textsuperscript{6}, Michael Schwarz\textsuperscript{5}, Yuval Yarom\textsuperscript{8}

\textsuperscript{1} Independent (www.paulkocher.com), \textsuperscript{2} Google Project Zero,
\textsuperscript{3} G DATA Advanced Analytics, \textsuperscript{4} University of Pennsylvania and University of Maryland,
\textsuperscript{5} Graz University of Technology, \textsuperscript{6} Cyberus Technology,
\textsuperscript{7} Rambus, Cryptography Research Division, \textsuperscript{8} University of Adelaide and Data61
Spectre attack

- Vulnerabilities in modern computers
Background: *Computer architecture*

- Instruction set architecture (ISA)
- Execution order
  - In-order
  - Out-of-order
  - Re-order buffer (ROB)
Background: *To boost CPU performance*

- Improve hardware
- Design optimization
  - *Speculative execution*
- Memory hierarchy
  - CPU-memory
  - Caches
Spectre attack overview

• Conditional branch example
  • array1 = [1,2]
  • array2 = [x, y, z ......]
  • x is input under the attacker’s control

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

Memory address: 1 2 ...... secret k

Array1[0] Array1[x] = k
Spectre attack overview

- Attacker read array2[i*4096]
  - Find i=3 is fast
  - array[x] = k has been cached
  - Then secret k =3 is revealed

- Side channel attack
Spectre attack: Brach prediction

- Attacker can misdirect the prediction

Speculation scenario (= computation error) + “Safe” computation that speculation turns unsafe + Induce computation with desired error

Side channel

Detect & analyze leaked data
Mitigation options

- Not do speculative execution
  - Trade-off between performance and security
- Preventing access to secret data
  - Add new data dependencies
- Add hardware to hide speculative execution
- Not all speculative loads leak secret
Summary

• Spectre attacks leverage the speculative execution.
• Software security depends on having a clear common understanding between hardware and software.
• Trade-off between security and performance is always a problem.
Thank you