Restore Buffer Overflow Attacks:
Breaking Undo-Based Defense Schemes

Jongmin Lee, Gunjae Koo
Korea University
Outline

- Backgrounds
- Idea
- Evaluation
- Conclusion
Transient Execution Attacks
Prime+Probe Spectre Attack Example

L1 Cache

Probing

1 Hit!!!

2 Hit!!!

3 Miss!!!

8 Hit!!!

New

Response for speculative load

Leaked

Evicted

Primed

Sets

Ways
Hardware-Based Mitigations against Spectre

- Delaying
  - InvisiSpec
  - SafeSpec
  - SelectiveDelay

- Restoring (Undo-Style)
  - CleanupSpec
    - Restores data from L2 cache
  - ReViCe
    - Restores data from additional victim cache
Undo-Style Defense against Prime+Probe
Overflowing the Restore Buffer

Response for speculative load

Evicted

LOST!!!
Attack Design

• Consecutive evictions must happen immediately after accessing the secret value.

• Consecutive evictions must happen before the speculation is resolved.

Dummy set pushes a target entry out from the restore buffer.

Spare set conflicts with Dummy set on L1 cache.

Spare set resides in L2 cache.
Restore Buffer Overflow Attack

![Diagram of L1 Cache with Mis-Speculation, Probing, Restoring, and Evicted areas.]

- **Mis-Speculation**
- **Probing**
- **Restoring**
- **Evicted**

**Response for speculative load**

1. **HiD!!**
2. **HiD!!**
3. **Miss!!**
4. **HiD!!**

Set 8: **Leaked**
Evaluation

- Gem5 with SE mode
- Protected by undo-style defense with a restore buffer nearby L1 cache

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core</td>
<td>x86 ISA, out-of-order, no SMT, 64 IQ entries, 192 ROB entries, 32 LQ entries, 32 SQ entries</td>
</tr>
<tr>
<td>Branch Predictor</td>
<td>L-TAGE</td>
</tr>
<tr>
<td>L1-I cache</td>
<td>32 KB, 64B line, 8-way</td>
</tr>
<tr>
<td>L1-D cache</td>
<td>32 KB, 64B line, 8-way, 8 MSHR entries</td>
</tr>
<tr>
<td>L2 cache</td>
<td>256 KB, 64B line, 8-way</td>
</tr>
<tr>
<td>Data prefetcher</td>
<td>Disabled</td>
</tr>
</tbody>
</table>
Attack Results

Canceled memory requests that has not responded before speculation is resolved
Conclusion

- **Limited restore buffer resources**
  - Undo-style defense relies on the limited restore buffer space
  - New side-channel: overflowing the restore buffer

- **Characteristics of the attack**
  - Processor is more secure with larger restore buffer
  - Part of in-flight memory requests can be cancelled by the undo-style defense before speculation is resolved
Thank you

Restore Buffer Overflow Attacks: Breaking Undo-Based Defense Schemes
Jongmin Lee, Gunjae Koo

✉ flackekd@korea.ac.kr