Prime+Probe 1 – JavaScript 0
Overcoming Browser-based Side-Channel Defenses

Anatoly Shusterman
Ben-Gurion Univ. of the Negev
shustera@post.bgu.ac.il

Ayush Agarwal
University of Michigan
ayushagr@umich.edu

Sioli O’Connell
University of Adelaide
sioli.oconnell@adelaide.edu.au

Daniel Genkin
University of Michigan
genkin@umich.edu

Yossi Oren
Ben-Gurion Univ. of the Negev
yos@bgu.ac.il

Yuval Yarom
University of Adelaide and Data61
yval@cs.adelaide.edu.au
What are the minimal features required for mounting microarchitectural side-channel attacks in browsers?

Research Questions

Trust Boundaries

No Entry For Attackers

Virtual Memory

User/Kernel Separation

μ-arch attacks

https://flic.kr/p/hs842s (CC BY-SA 2.0)
Prime+Probe

Ingredients:

- Array buffer-memory map
- Nano_second-Timer

Covert Channel
Private-Key Retrieval
The Spy in the Sandbox – Practical Cache Attacks in Javascript

Yossef Oren, Vasileios P. Kemerlis, Simha Sethumadhavan and Angelos D. Keromytis
Computer Science Department, Columbia University
{yos \ vpk \ simha \ angelos}@cs.columbia.edu
No Entry For Attackers

- No Direct Memory Accesses
- Reduced Clock Resolution
Our Research Questions

• RQ1: What are the minimal requirements for μ-architectural side-channel attacks in browsers?
Our Research Questions

• RQ2: Can processor diversity prevent side-channel attacks?
Contributions

• RQ1: End-to-end of remote cache attacks with no timers, no arrays, and no JavaScript
• RQ2: An architecturally-agnostic attack that works on ARM, AMD, Intel and Apple M1
- No Direct Memory Accesses
- Reduced Clock Resolution
Attack 1: Cache Occupancy [S+19]

- Required timer resolution reduced to milliseconds
- Cache structure does not need to be reverse engineered
- No Direct Memory Accesses
- Reduced Clock Resolution
- Super Low Clock Resolution
Attack 2: Sweep Counting [S+19]

• Count the number of times we can read the buffer in a clock tick

• Required timer resolution reduced to 10 Hz
- No Direct Memory Accesses
- Super Low Clock Resolution
- Timers Completely Disabled
Attack 3: DNS Racing [New!]

- No timers required!
- Resists jitter well enough to be used between two continents
- No Direct Memory Accesses
- Timers Completely Disabled
- Array API Disabled
Attack 4: String and Sock [new!]

- Strings are arrays in disguise
- No timers or arrays required!

Web-Page On Target → Send Short Packet → Colluding WebSockets Server

Search in a long String

Send Short Packet

Log start time
Log end time

On Target
No Direct Memory Access
Timers Completely Disabled
Array API Disabled
JavaScript Disabled
Attack 5: CSS Prime+Probe [New!]

Web-Page
On Target

Resolve domain

Search in a long String

Colluding DNS Server

Resolve domain

Log time

Log time
Attack 5: CSS Prime+Probe

```html
<div id="pp" class="AAA...AAA">
    <div id="s1">X</div>
    <div id="s2">X</div>
    <div id="s3">X</div>
</div>

#pp:not([class*='jigbaa']) #s1 {
    background-image: url('https://knbdsd.badserver.com');
}

#pp:not([class*='akhevn']) #s2 {
    background-image: url('https://pjemh7.badserver.com');
}
```

Search non existing string

Probe the LLC

Resolve non existing image

==

TIMER
Evaluation

• Our method is probably not effective for cryptanalysis
• So, what is it good for?
Website Fingerprinting

Webpage Rendering

Cache Contention Measurement

Time (msec)

https://privateurl.com

100 Traces

100 URLs

5 Attacks

4 processors

Cache Contention
Deep Learning Models

Cache Contention Trace → Input → Deep Learning Models → Output → URL
## Results

<table>
<thead>
<tr>
<th>Attack Technique</th>
<th>Intel i5-3470</th>
<th>AMD Ryzen 9 3900X</th>
<th>Apple M1</th>
<th>Samsung Exynos 2100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cache Occupancy</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sweep Counting</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DNS Racing</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>String and Sock</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CSS Prime+Probe</td>
<td></td>
<td></td>
<td>1%</td>
<td></td>
</tr>
</tbody>
</table>
Conclusion

• Restricted environments don’t prevent cache contention attacks.

• Lower attack requirements make it architectural agnostic.

• Protection against μ-architectural leaks should be applied at the source, not at the receiver.
https://orenlab.sise.bgu.ac.il/p/PP0

- No Direct Memory Accesses
- Timers Completely Disabled
- Array API Disabled
- JavaScript Disabled

Thank You!