Malicious Management Unit
Why Stopping Cache Attacks in Software is Harder Than You Think

Stephan van Schaik
Cristiano Giuffrida    Herbert Bos    Kaveh Razavi
Motivation

Why should you care about cache attacks?
Motivation

- CPU
- Registers
- Main Memory
- Disk Storage
Motivation

Processors advance faster than memory

Increasing performance gap

CPU
Registers
Main Memory
Disk Storage
Motivation

- CPU
- Registers
- L1 Cache
- L2 Cache
- L3 Cache
- Main Memory
- Disk Storage
Motivation

- CPU
- Registers
- L1 Cache
- L2 Cache
- L3 Cache
- Main Memory
- Disk Storage

Faster

Larger

Smaller

Slower
Motivation
Memory accesses are not performed in constant time
Motivation

- Caches are shared resources
- Caches can be manipulated
- Spy on other processes
- Input events
- Leak sensitive data
Motivation

- CPU
- Registers
- L1 Cache
- L2 Cache
- L3 Cache
- Main Memory
- Disk Storage

MMU
AES

- Advanced Encryption Standard
- Software implementations use T-tables
- $T[p_i \oplus k_i]$
- Indices are key-dependent
- Elements may be in main memory or the cache
An example of PRIME + PROBE against AES encryption
PRIME + PROBE

Cache

Cache Sets

Cache Entries

AES T-table
PRIME + PROBE

Attacker

Cache

AES T-table

PRIME

Cache Sets

Cache Entries
PRIME + PROBE

Attacker

Cache

AES T-table

PRIME

Cache Entries

Sets

Cache AES T-table

Attacker

PRIME
PRIME + PROBE

Attacker

Cache

AES T-table

PRIME

Cache Sets

Cache Entries
PRIME + PROBE

Attacker

Wait

Cache

Cache Entries

Cache Sets

AES T-table
PRIME + PROBE

Attacker

Cache

AES T-table

Wait

Cache Entries

Cache Sets
PRIME + PROBE

Attacker

PROBE

Cache

Cache Sets

Cache Entries

AES T-table
PRIME + PROBE

Attacker

PROBE

Cache

Cache Entries

Cache Sets

AES T-table
PRIME + PROBE

Diagram showing the relationship between Attackers, Cache Entries, Cache Sets, and AES T-table.
PRIME + PROBE

Attacker

PROBE

Cache

Cache Sets

Cache Entries

AES T-table
AES encrypt used another cache set
PRIME + PROBE

Attacker

Cache

AES T-table

Cache Sets

Cache Entries
PRIME + PROBE

Attacker

Eviction Set

Cache

Cache Entries

Cache Sets

AES T-table
PRIME + PROBE

- Attacker
- Cache
- PRIME
- Cache Sets
- Cache Entries
- AES T-table

Diagram showing the interactions between the attacker, cache, and AES T-table.
PRIME + PROBE

Attacker

Cache

Cache Entries

Cache Sets

PRIME

AES T-table
PRIME + PROBE

Attacker

Cache

Cache Entries

Cache Sets

AES T-table

PRIME

7 / 39
PRIME + PROBE

Attacker

Wait

Cache

Cache Entries

Cache Sets

AES T-table
PRIME + PROBE

Attacker
Wait

Cache
Cache Entries

AES T-table
Encrypt
PRIME + PROBE

Attacker

Cache

Cache AES T-table

Wait

Cache Entries

Cache Sets

AES T-table
PRIME + PROBE

Attacker

PROBE

Cache

Cache Entries

Cache Sets

AES T-table
PRIME + PROBE

Attacker

PROBE

Cache

Cache Entries

Cache Sets

AES T-table
PRIME + PROBE

Attacker

PROBE

Cache

Cache Entries

Cache Sets

AES T-table
AES encrypt used the same cache set
Can we defend against cache attacks?
Defenses

Way Partitioning

Cache

Cache Sets

Cache Entries

Attacker

Victim
Defenses

Cache Entries
Cache Sets
Cache
Set Partitioning
Attacker
Victim
How does page coloring work?
Page coloring

Cache

Cache Sets

Cache Entries
Page coloring

![Diagram of cache with coloring]

Cache

Cache Sets

Cache Entries
Page coloring

Cache

Cache Sets

Cache Entries
Page coloring
Page coloring
Page coloring

Cache Entries
Cache Sets
Cache
Attacker
Page coloring
The victim and the attacker are nicely isolated.
The attacker can only allocate red pages
However, the page tables aren't colored
Page coloring

However, the page tables aren't colored
However, the page tables aren't colored
However, the page tables aren't colored
Page coloring

However, the page tables aren't colored.
Page coloring

Can we control the page tables for cache attacks?
XLATE attacks

- Memory Management Unit (MMU)
- Translates virtual addresses into their physical counterparts
- Hence translate or XLATE attacks
- XLATE + PROBE caches page tables instead of pages
Page table walks

How does the MMU perform page table walks?
Page table walks

Virtual Address
0x1fafe7fbf000

- page table indices (36-bit)  page offset (12-bit)
Page table walks
Page table walks

Virtual Address
0x1fafe7fbf000

- page table indices (36-bit) page offset (12-bit)

63 191 319 447

Cache
Cache Sets
Cache Entries
Page table walks

Virtual Address
0x1fafe7fbf000

page table indices (36-bit) page offset (12-bit)

63 191 319 447

CR3

Cache

Cache Sets

Cache Entries
Page table walks

Virtual Address
0x1fafe7fbf000

page table indices (36-bit) page offset (12-bit)

63 191 319 447

CR3
PML4
Cache
Cache Sets
Cache Entries
Page table walks

Virtual Address
0x1fafe7fbf000

CR3

PML4E

PML4

Cache

Cache Sets

Cache Entries

page table indices (36-bit) page offset (12-bit)

63 191 319 447
Page table walks

Virtual Address
0x1fafe7fbf000

PML4 PDPT
CR3
PML4E

Cache Entries
Cache Sets

Cache
Page table walks

Virtual Address
0x1fafe7fbf000

page table indices (36-bit) page offset (12-bit)

63 191 319 447

PML4 PDPT
CR3 PML4E
PDPTE

Cache Entries
Cache Sets
Cache
Page table walks

Virtual Address
0x1fafe7fbf000

Page table indices (36-bit)
- 63
- 191
- 319
- 447

Page offset (12-bit)

PML4 PDPT Page Directory

CR3 PML4E

PDPTE

Cache Entries

Cache Sets

Cache
Page table walks

Virtual Address
0x1fafe7fbf000

page table indices (36-bit) page offset (12-bit)

63 191 319 447

Cache Entries
Cache Sets
Cache

PML4E
PDPTE
PDE

PML4
PDPT
Page Directory

CR3

16 / 39
Page table walks

Virtual Address
0x1fafe7fb000

page table indices (36-bit) page offset (12-bit)

63 191 319 447

Cache Entries
Cache Sets

Cache
Page table walks

Virtual Address
0x1fafe7fbf000

page table indices (36-bit)
page offset (12-bit)

63 191 319 447

Cache
Cache Sets
Cache Entries

CR3
PML4E
PML4
PDPT
PDE
PDPTE
Page Directory
Page Table

PTE
Page table walks

Virtual Address
0x1fafe7fbf000

page table indices (36-bit) page offset (12-bit)

63 191 319 447

Page Table

Virtual Address
0x1fafe7fbf000

Page Table

Physical Address

CR3

PML4E

PML4

PDPTE

PDPT

PDE

Page Directory

PDE

PTE

Page Table

Physical Address

Cache

Cache Sets

Cache Entries
Page table walks

What do we need for XLATE + PROBE?
Challenges

- Avoid noise from high-level page tables
- Avoid noise from pages
- Build eviction sets
Translation Caches

Virtual Address
0x1fafe7fbf000

- page table indices (36-bit)
- page offset (12-bit)

Translation Cache
Translation Caches

<table>
<thead>
<tr>
<th>Virtual Address</th>
<th>0x1fafe7fb000</th>
</tr>
</thead>
<tbody>
<tr>
<td>page table indices (36-bit)</td>
<td>page offset (12-bit)</td>
</tr>
</tbody>
</table>

Translation Cache

![Translation Cache Diagram]
Translation Caches

Virtual Address
0x1fafe7fbf000

page table indices (36-bit) page offset (12-bit)

63 191 319 447

Translation Cache
Translation Caches

Virtual Address
0x1fafe7fb000

CR3

Translation Cache

page table indices (36-bit)
63 191 319 447

page offset (12-bit)
Translation Caches

Virtual Address
0x1fafe7fbf000

page table indices (36-bit)
page offset (12-bit)

63 191 319 447

Translation Cache
PML4

CR3
Translation Caches

Virtual Address
0x1fafe7fbf000

page table indices (36-bit)
page offset (12-bit)

63 191 319 447

Translation Cache
PML4
CR3
PML4E
Translation Caches
Translation Caches

Virtual Address 0x1fafe7fbf000

Page table indices (36-bit)

Page offset (12-bit)

Translation Cache

PML4E

PML4

CR3

PML4

PDPT
Translation Caches

Virtual Address
0x1fafe7fbf000

Page table indices (36-bit)
Page offset (12-bit)

CR3
PML4E
PDPTE

Translation Cache
PML4
PDPT
Translation Caches

Virtual Address
0x1fafe7fbf000

- page table indices (36-bit)
- page offset (12-bit)

63 191 319 447

Translation Cache
PML4
PDPTE
PDE

Page Directory

Translation Cache
Translation Caches

Virtual Address 0x1fafe7fbf000

Page table indices (36-bit) page offset (12-bit)

63 191 319 447

Translation Cache
PML4
PDPT
Page Directory
Page Table
Translation Caches

Translation caches cache intermediate page tables
Translation Caches

Translation Cache

Virtual Address 0x1fafe7fbf000

page table indices (36-bit)

page offset (12-bit)

63 191 319 447

Translation Cache

PML4

PML4E

PDPTE

PDE

PTE

CR3
Translation Caches

Virtual Address
0x1fafe7fbf000

page table indices (36-bit)
page offset (12-bit)

63 191 319 447

Translation Cache
PML4
PML4E
PDPT
PDPTE
PDE
Page Directory
Page Table
Physical Address

CR3
Translation Caches

Virtual Address
0x1fafe7fb8000

<table>
<thead>
<tr>
<th>page table indices (36-bit)</th>
<th>page offset (12-bit)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| 63 | 191 | 319 |

Translation Cache
Translation Caches

Virtual Address
0x1fafe7fb8000

page table indices (36-bit)
63 191 319 440

Translation Cache

63 191 319
Translation Caches

Translation caches cache intermediate page tables

Virtual Address
0x1fafe7fb8000

page table indices (36-bit)
page offset (12-bit)

Translation Cache

63 191 319 440
Translation Caches

Translation caches cache intermediate page tables
Translation Caches

Translation caches cache intermediate page tables
Translation Caches

Virtual Address
0x1fafe7fb8000

Translation Cache

Page Table Indices (36-bit)

Page Offset (12-bit)

PTE

63 191 319 440

Translation Cache

Page Table
Translation Caches

Virtual Address
0x1fafa7fb8000

page table indices (36-bit)

63 191 319 440

Translation Cache

Translation Cache

63 191 319

Page Table

Physical Address

PTE
Some properties of translation caches are undocumented
Some properties of translation caches are undocumented

How do we reverse engineer them?
Translation Caches

Virtual Address
0x1fafe7e00000

page table indices (36-bit) page offset (12-bit)

63 191 319 0

MMU

Translation Cache
Translation Caches

Virtual Address
0x1fafe7e00000

page table indices (36-bit) page offset (12-bit)

63 191 319 0

Translation Cache

Load into translation cache
Translation Caches

Virtual Address
0x1fafe7e00000

- page table indices (36-bit)
- page offset (12-bit)

63 191 319 0

Translation Cache

MMU

Physical Address

Load into translation cache
Translation Caches

Virtual Address
0x1fafe7e00000

page table indices (36-bit) page offset (12-bit)

63 191 319 0

MMU

Translation Cache

Perform n page table walks
Translation Caches

Virtual Address
0x1fafe8000000

page table indices (36-bit) page offset (12-bit)
63 191 320 0

MMU
63 191 319

Translation Cache

Perform n page table walks
Translation Caches

Translation Cache

Virtual Address
0x1fafe8000000

page table indices (36-bit) page offset (12-bit)

63 191 320 0

MMU

Physical Address

Perform n page table walks
Translation Caches

Translation Cache

Virtual Address
0x1fafe8200000

page table indices (36-bit)  page offset (12-bit)

63 191 321 0

MMU

Perform n page table walks
Translation Caches

Translation Cache

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>63</td>
<td>191</td>
<td>320</td>
</tr>
<tr>
<td>63</td>
<td>191</td>
<td>319</td>
</tr>
<tr>
<td>63</td>
<td>191</td>
<td>321</td>
</tr>
</tbody>
</table>

Virtual Address
0x1fafe8200000

page table indices (36-bit)

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>63</td>
<td>191</td>
<td>321</td>
<td>0</td>
</tr>
</tbody>
</table>

MMU

Perform n page table walks
Translation Caches

Virtual Address
0x1fafe7e00000

page table indices (36-bit) page offset (12-bit)

63 191 319 0

MMU

Translation Cache

Reload the target
Translation Caches

The page table entry is still cached

Translation Cache

Virtual Address
0x1fafe7e00000

page table indices (36-bit) page offset (12-bit)

63 191 319 0

MMU

Physical Address

The page table entry is still cached
Translation Caches

Virtual Address
0x1fafe7e00000

page table indices (36-bit) page offset (12-bit)

63 191 319 0

Translation Cache

Perform n page table walks

Physical Address

MMU
Translation Caches

Virtual Address
0x1fafe8000000

MMU
63 191 319

Perform n page table walks
**Translation Caches**

Virtual Address
0x1fafe8000000

Perform n page table walks
Translation Caches

Virtual Address
0x1fafe8200000

Translation Cache

MMU

Perform n page table walks
Translation Caches

Perform n page table walks
Translation Caches

Virtual Address
0x1fafe8400000

Translation Cache

MMU

Perform n page table walks
Translation Caches

Virtual Address
0x1fafe8400000

page table indices (36-bit) page offset (12-bit)

63 191 322 0

Translation Cache

Physical Address

Perform n page table walks
Translation Caches

Virtual Address
0x1fafe8600000

Translation Cache

<table>
<thead>
<tr>
<th>63</th>
<th>191</th>
<th>320</th>
</tr>
</thead>
<tbody>
<tr>
<td>63</td>
<td>191</td>
<td>322</td>
</tr>
<tr>
<td>63</td>
<td>191</td>
<td>319</td>
</tr>
<tr>
<td>63</td>
<td>191</td>
<td>321</td>
</tr>
</tbody>
</table>

MMU

Perform n page table walks
Translation Caches

Translation Cache

Virtual Address
0x1fafe8600000

page table indices (36-bit) page offset (12-bit)

63 191 323 0

MMU

Perform n page table walks

Physical Address
Translation Caches

Virtual Address
0x1fafe7e00000

page table indices (36-bit)  page offset (12-bit)

63  191  319  0

Translation Cache

Reload the target
Translation Caches

Perform full page table walk

- Translation Cache
- Virtual Address
  0x1fafe7e00000
- page table indices (36-bit)
  63 191 319 0
- page offset (12-bit)
- MMU
- Physical Address
## Translation Caches

<table>
<thead>
<tr>
<th>CPU</th>
<th>Year</th>
<th>L1d</th>
<th>L2</th>
<th>L3</th>
<th>4K pages</th>
<th>2M pages</th>
<th>1G pages</th>
<th>TLBs PML2E</th>
<th>TLBs PML3E</th>
<th>TLBs PML4E</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intel Core i7-7500U (Kaby Lake) @ 2.70GHz</td>
<td>2016</td>
<td>32K</td>
<td>256K</td>
<td>4M</td>
<td>1600</td>
<td>32</td>
<td>20</td>
<td>24-32</td>
<td>3-4</td>
<td>0</td>
<td>5m49s</td>
</tr>
<tr>
<td>Intel Core m3-6Y30 (Skylake) @ 0.90GHz</td>
<td>2015</td>
<td>32K</td>
<td>256K</td>
<td>4M</td>
<td>1600</td>
<td>32</td>
<td>20</td>
<td>24</td>
<td>3-4</td>
<td>0</td>
<td>6m01s</td>
</tr>
<tr>
<td>Intel Xeon E3-1240 v5 (Skylake) @ 3.50GHz</td>
<td>2015</td>
<td>32K</td>
<td>256K</td>
<td>8M</td>
<td>1600</td>
<td>32</td>
<td>20</td>
<td>24</td>
<td>3-4</td>
<td>0</td>
<td>3m08s</td>
</tr>
<tr>
<td>Intel Core i7-6700K (Skylake) @ 4.00GHz</td>
<td>2014</td>
<td>32K</td>
<td>256K</td>
<td>8M</td>
<td>1600</td>
<td>32</td>
<td>20</td>
<td>24-32</td>
<td>3-4</td>
<td>0</td>
<td>3m41s</td>
</tr>
<tr>
<td>Intel Celeron N2840 (Silvermont) @ 2.16GHz</td>
<td>2013</td>
<td>32K</td>
<td>256K</td>
<td>6M</td>
<td>576</td>
<td>32</td>
<td>4</td>
<td>24-32</td>
<td>3</td>
<td>0</td>
<td>3m11s</td>
</tr>
<tr>
<td>Intel Core i7-4500U (Haswell) @ 1.80GHz</td>
<td>2012</td>
<td>32K</td>
<td>256K</td>
<td>6M</td>
<td>576</td>
<td>32</td>
<td>4</td>
<td>24-32</td>
<td>3</td>
<td>0</td>
<td>3m05s</td>
</tr>
<tr>
<td>Intel Core i7 920 (Nehalem) @ 2.67GHz</td>
<td>2011</td>
<td>32K</td>
<td>512K</td>
<td>16M</td>
<td>1600</td>
<td>1600</td>
<td>64</td>
<td>0</td>
<td>64</td>
<td>16</td>
<td>13m16s</td>
</tr>
<tr>
<td>AMD Ryzen 7 1700 8-Core (Zen) @ 3.3GHz</td>
<td>2017</td>
<td>32K</td>
<td>512K</td>
<td>16M</td>
<td>1600</td>
<td>1600</td>
<td>64</td>
<td>0</td>
<td>64</td>
<td>16</td>
<td>30m50s</td>
</tr>
<tr>
<td>AMD Ryzen 5 1600X 6-Core (Zen) @ 3.6GHz</td>
<td>2017</td>
<td>32K</td>
<td>512K</td>
<td>16M</td>
<td>1600</td>
<td>1600</td>
<td>64</td>
<td>0</td>
<td>64</td>
<td>16</td>
<td>30m50s</td>
</tr>
<tr>
<td>AMD FX-8350 8-Core (Piledriver) @ 4.0GHz</td>
<td>2012</td>
<td>64K</td>
<td>256K</td>
<td>8M</td>
<td>1088</td>
<td>1088</td>
<td>1088</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2m50s</td>
</tr>
<tr>
<td>AMD FX-8320 8-Core (Piledriver) @ 3.5GHz</td>
<td>2012</td>
<td>64K</td>
<td>256K</td>
<td>8M</td>
<td>1088</td>
<td>1088</td>
<td>1088</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2m47s</td>
</tr>
<tr>
<td>AMD FX-8120 8-Core (Bulldozer) @ 3.4GHz</td>
<td>2011</td>
<td>16K</td>
<td>256K</td>
<td>8M</td>
<td>1056</td>
<td>1056</td>
<td>1056</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2m33s</td>
</tr>
<tr>
<td>AMD Athlon II 640 X4 (K10) @ 3.0GHz</td>
<td>2010</td>
<td>64K</td>
<td>512K</td>
<td>N/A</td>
<td>560</td>
<td>176</td>
<td>N/A</td>
<td>24</td>
<td>0</td>
<td>0</td>
<td>7m50s</td>
</tr>
<tr>
<td>AMD E-350 (Bobcat) @ 1.6GHz</td>
<td>2010</td>
<td>32K</td>
<td>512K</td>
<td>N/A</td>
<td>552</td>
<td>8-12</td>
<td>N/A</td>
<td>8-12</td>
<td>0</td>
<td>0</td>
<td>5m38s</td>
</tr>
<tr>
<td>AMD Phenom 9550 4-Core (K10) @ 2.2GHz</td>
<td>2008</td>
<td>64K</td>
<td>512K</td>
<td>2M</td>
<td>560</td>
<td>176</td>
<td>48</td>
<td>24</td>
<td>0</td>
<td>0</td>
<td>6m52s</td>
</tr>
<tr>
<td>Rockchip RK3399 (ARM Cortex A72) @ 2.0GHz</td>
<td>2017</td>
<td>32K</td>
<td>1M</td>
<td>N/A</td>
<td>544</td>
<td>512</td>
<td>N/A</td>
<td>16</td>
<td>6</td>
<td>N/A</td>
<td>17m49s</td>
</tr>
<tr>
<td>Rockchip RK3399 (ARM Cortex A53) @ 1.4GHz</td>
<td>2017</td>
<td>32K</td>
<td>512K</td>
<td>N/A</td>
<td>522</td>
<td>512</td>
<td>N/A</td>
<td>64</td>
<td>0</td>
<td>N/A</td>
<td>7m06s</td>
</tr>
<tr>
<td>Allwinner A64 (ARM Cortex A53) @ 1.2GHz</td>
<td>2016</td>
<td>32K</td>
<td>512K</td>
<td>N/A</td>
<td>522</td>
<td>512</td>
<td>N/A</td>
<td>64</td>
<td>0</td>
<td>N/A</td>
<td>52m26s</td>
</tr>
<tr>
<td>Samsung Exynos 5800 (ARM Cortex A15) @ 2.1GHz</td>
<td>2014</td>
<td>32K</td>
<td>2M</td>
<td>N/A</td>
<td>544</td>
<td>512</td>
<td>N/A</td>
<td>16</td>
<td>0</td>
<td>N/A</td>
<td>13m28s</td>
</tr>
<tr>
<td>Nvidia Tegra K1 CD580M-A1 (ARM Cortex A15) @ 2.3GHz</td>
<td>2014</td>
<td>32K</td>
<td>2M</td>
<td>N/A</td>
<td>544</td>
<td>512</td>
<td>N/A</td>
<td>16</td>
<td>0</td>
<td>N/A</td>
<td>24m19s</td>
</tr>
<tr>
<td>Nvidia Tegra K1 CD570M-A1 (ARM Cortex A15; LPAE) @ 2.1GHz</td>
<td>2014</td>
<td>32K</td>
<td>2M</td>
<td>N/A</td>
<td>544</td>
<td>512</td>
<td>N/A</td>
<td>16</td>
<td>0</td>
<td>N/A</td>
<td>6m35s</td>
</tr>
<tr>
<td>Samsung Exynos 5800 (ARM Cortex A7) @ 1.3GHz</td>
<td>2014</td>
<td>32K</td>
<td>512K</td>
<td>N/A</td>
<td>266</td>
<td>256</td>
<td>N/A</td>
<td>64</td>
<td>0</td>
<td>N/A</td>
<td>17m42s</td>
</tr>
<tr>
<td>Samsung Exynos 5250 (ARM Cortex A15) @ 1.7GHz</td>
<td>2012</td>
<td>32K</td>
<td>1M</td>
<td>N/A</td>
<td>544</td>
<td>512</td>
<td>N/A</td>
<td>16</td>
<td>0</td>
<td>N/A</td>
<td>6m46s</td>
</tr>
</tbody>
</table>
Translation Caches

Translation caches are widely available on Intel, AMD and ARM
Translation Caches

Page Tables

Cache

Pages

Cache Entries
Translation Caches skip page table walks
Challenges

- ✓ Avoid noise from high-level page tables
- ✗ Avoid noise from pages
- ✗ Build eviction sets
Shared Memory

Page Tables

Translation Cache

Pages

Cache

Cache Sets

Cache Entries
Use shared memory to reduce noise
Challenges

- Avoid noise from high-level page tables ✔
- Avoid noise from pages ✔
- Build eviction sets ✗
Building Eviction Sets

“The Spy in the Sandbox: Practical Cache Attacks in JavaScript and their Implications” - Oren et al.
Building Eviction Sets

Cache

Cache Sets

Cache Entries
Building Eviction Sets

Allocate pages

Cache

Cache Sets

Cache Entries
Allocate pages

Cache Entries

Cache Sets
Building Eviction Sets

Allocate pages

Cache Entries

Cache Sets

Cache
Building Eviction Sets

Allocate pages
Cache Entries
Cache Sets
Cache
Building Eviction Sets

Allocate pages

Cache

Cache Entries

Cache Sets
Building Eviction Sets

Allocate pages

Cache

Cache Sets

Cache Entries
Building Eviction Sets

Allocate pages

Cache

Cache Sets

Cache Entries
Building Eviction Sets

Draw target

Cache

Cache Sets

Cache Entries
Building Eviction Sets

Draw target

Cache

Cache Sets

Cache Entries
Building Eviction Sets

Load target into cache
Building Eviction Sets

Draw pages and try to evict the target

Cache

Cache Sets

Cache Entries

29 / 39
Building Eviction Sets

Draw pages and try to evict the target

Cache

Cache Sets

Cache Entries

Draw pages and try to evict the target
Building Eviction Sets

Draw pages and try to evict the target

Cache Entries
Cache Sets
Cache
Building Eviction Sets

Draw pages and try to evict the target

Cache

Cache Sets

Cache Entries
Building Eviction Sets

Draw pages and try to evict the target
Building Eviction Sets

Draw pages and try to evict the target
Building Eviction Sets

Draw pages and try to evict the target

Cache

Cache Sets

Cache Entries

Draw pages and try to evict the target
Building Eviction Sets

Draw pages and try to evict the target

Cache Entries

Cache Sets

Cache

Draw pages and try to evict the target
Building Eviction Sets

Draw pages and try to evict the target
Building Eviction Sets

Found an eviction set

Cache Entries

Cache Sets

Cache
Building Eviction Sets

Optimize the eviction set

Cache

Cache Sets

Cache Entries

Optimize the eviction set
Building Eviction Sets

Optimize the eviction set

Cache Entries

Cache Sets

Cache
Building Eviction Sets

Optimize the eviction set

Cache Entries

Cache Sets

Cache

Optimize the eviction set
Building Eviction Sets

Optimize the eviction set

Cache Entries

Cache Sets

Optimize the eviction set
Building Eviction Sets

Optimize the eviction set

Cache Entries

Cache Sets

Cache

Optimize the eviction set
Building Eviction Sets

Optimize the eviction set

Cache

Cache Entries

Cache Sets

Optimize the eviction set
Building Eviction Sets

Optimal eviction set found

Cache

Cache Sets

Cache Entries
Building Eviction Sets

Filter red pages
Building Eviction Sets

Filter red pages
This technique can also be applied to page tables
Challenges

✓ Avoid noise from high-level page tables
✓ Avoid noise from pages
✓ Build eviction sets
It’s time for the big picture.
XLATE + PROBE

Attacker

Cache

AES T-table

Cache Sets

Cache Entries
XLATE + PROBE
XLATE + PROBE

Eviction Set

Attacker

Page Tables

Cache

Cache Entries

AES T-table
XLATE + PROBE

Eviction Set

Attacker

Page Tables

Cache

Cache Sets

Cache AES T-table

AES T-table

XLATE
XLATE + PROBE

Eviction Set

Attacker

MMU

Page Tables

Cache

Cache Sets

Cache Entries

AES T-table

XLATE
XLATE + PROBE
XLATE + PROBE

Eviction Set

Attacker

Page Tables

Cache

AES T-table

XLATE

Cache Sets

Cache Entries

MMU
XLATE + PROBE

Eviction Set

Attacker

Page Tables

Cache

 AES T-table

XLATE

Cache Entries

Cache Sets
XLATE + PROBE

- Cache Entries
- Cache Sets
- Cache AES T-table
- Attacker
- Eviction Set
- Page Tables
- MMU
- Wait
- AES T-table
XLATE + PROBE

Eviction Set

Attacker

Page Tables

Cache

AES T-table

Wait

Encrypt
XLATE + PROBE

Eviction Set

Attacker

Page Tables

Cache Sets

Cache Entries

AES T-table

Wait
XLATE + PROBE

![Diagram of XLATE + PROBE system with components labeled: Eviction Set, MMU, Attacker, Cache, AES T-table, Page Tables, Cache Entries, Cache Sets, and Wait areas.]
XLATE + PROBE

Eviction Set

Attacker

Page Tables

Cache

AES T-table

Cache Sets

Wait

Cache Entries
XLATE + PROBE

Eviction Set

Page Tables

Cache Entries

AES T-table

PROBE

Attacker

Cache

MMU

Cache Sets
XLATE + PROBE

PROBE
XLATE + PROBE

Eviction Set

Page Tables

Cache Entries

Cache Sets

Attacker

MMU

Cache AES T-table

AES T-table
XLATE + PROBE

PROBE

Eviction Set

Attacker

Page Tables

Cache

Cache Sets

Cache AES T-table

AES T-table
AES encrypt used the same cache set
Evaluation
Evaluation

- Reliability
- Effectiveness
- Cache defenses
Reliability

- Bandwidth (bytes/sec)
  - FLUSH + RELOAD
  - FLUSH + FLUSH
  - PRIME + PROBE
  - XLate + PROBE

- Bit errors (bits/sec)
  - Cross-Thread (correct)
  - Cross-Thread (raw)
  - Cross-Core (correct)
  - Cross-Core (raw)

XLATE attacks are practical
XLATE attacks are practical
Effectiveness

**FLUSH + RELOAD**

**PRIME + PROBE**

**PRIME + ABORT**

**FLUSH + FLUSH**

**XLATE + PROBE**

**XLATE + ABORT**

XLATE + PROBE is effective against AES T-tables.
Effectiveness

**XLATE + PROBE** is effective against AES T-tables
Cache Defenses

PRIME + PROBE (coloring)

XLATE + PROBE (coloring)

PRIME + PROBE (ways)

XLATE + PROBE (ways)
Cache Defenses

**PRIME + PROBE** (coloring)

**XLATE + PROBE** (coloring)

**PRIME + PROBE** (ways)

**XLATE + PROBE** (ways)

**XLATE + PROBE** bypasses set and way partitioning
Conclusions

- New family of cache attacks: X\textsc{late}
- Indirect cache attacks are practical
- Reconsider existing cache defenses
- \url{https://vusec.net/projects/xlate}