NVCool: When Non-Volatile Caches Meet Cold Boot Attacks

Xiang Pan†, Anys Bacha‡, Spencer Rudolph, Li Zhou, Yinqian Zhang, and Radu Teodorescu

The Ohio State University, Uber†, University of Michigan‡
Non-Volatile Memory is Coming

- Low power, high density, and good scalability make NVM attractive to industry companies
- 3D XPoint from Intel and Micron
- The Machine from HPE
- Crossbar and Everspin also make and sell NVM products

NVCool: When Non-Volatile Caches Meet Cold Boot Attacks
Xiang Pan, Anys Bacha, Spencer Rudolph, Li Zhou, Yinqian Zhang, and Radu Teoiorescu
Cold Boot Attack on DRAM

• Cooling DRAM to a certain low temperature can preserve its data for a short duration of time even without power supply

Halderman et al., Lest We Remember: Cold Boot Attacks on Encryption Keys, citp.princeton.edu/research/memory

• Plug in the frozen DRAM DIMMs to a pre-prepared machine and run key search program to get secret keys

• Successfully conducted on both laptop and mobile computer systems

NVCool: When Non-Volatile Caches Meet Cold Boot Attacks
Xiang Pan, Anys Bacha, Spencer Rudolph, Li Zhou, Yinqian Zhang, and Radu Teodorescu
Cold Boot Attack on NVM

• Trivial for NVM main memory but we focus on NVM caches

• NVM caches are vulnerable to cold boot attacks in a way SRAM caches are not
  • A few ms data retention time without power supply at cold temperatures

• Challenges
  • Caches only store a subset of data
  • Cache structure (set-associative) is very different from main memory (page)
  • Can we really find secrets from NVM caches?
Outline

• Threat Model
• Cache-Aware AES Key Search
• Methodology
• Attack Analysis
• Countermeasure
• Conclusions
Threat Model

• Attacker has physical access to the victim device

• Attacker has necessary equipments and knowledge to extract data from CPU caches
Threat Model

- What secrets can be found from cache?
  - Photos, emails, messages, disk encryption keys, ssh keys...
  - Anything stored in cache and useful to attacker
  - This work focuses on disk encryption keys as an example
Outline

• Threat Model
• Cache-Aware AES Key Search
• Methodology
• Attack Analysis
• Countermeasure
• Conclusions
AES Key Schedule

• AES key search:
  • Original key needs to be expanded before encryption/decryption operations
  • Current round key is deterministically computed from the previous round key
  • Scanning memory image sequentially can find the key if exists
• Challenges in cache-based approach:
  • Non-contiguous memory space
  • Incomplete key schedules
Cache Aware AES Key Search

- Non-contiguous memory space
- Incomplete key schedules
Outline

• Threat Model
• Cache-Aware AES Key Search
• Methodology
• Attack Analysis
• Countermeasure
• Conclusions

NVCool: When Non-Volatile Caches Meet Cold Boot Attacks
Xiang Pan, Anys Bacha, Spencer Rudolph, Li Zhou, Yinqian Zhang, and Radu Teodorescu
# Experimental Methodology

## Software Configuration

<table>
<thead>
<tr>
<th>Software Configuration</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulator</td>
<td>gem5</td>
</tr>
<tr>
<td>OS</td>
<td>Ubuntu Trusty 14.04 64-bit</td>
</tr>
<tr>
<td>Disk Encryption Module</td>
<td>dm-crypt + LUKS</td>
</tr>
<tr>
<td>Encryption Algorithm</td>
<td>AES-XTS with 128-bit key</td>
</tr>
<tr>
<td>Application</td>
<td>SPEC CPU2006</td>
</tr>
<tr>
<td>Execution</td>
<td>1B insts to run</td>
</tr>
<tr>
<td></td>
<td>1M insts to sample</td>
</tr>
</tbody>
</table>

## Hardware Configuration

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cores</td>
<td>8 (out-of-order)</td>
</tr>
<tr>
<td>ISA</td>
<td>ARMv8 (64-bit)</td>
</tr>
<tr>
<td>Frequency</td>
<td>3GHz</td>
</tr>
<tr>
<td>IL1/DL1 Size</td>
<td>32KB</td>
</tr>
<tr>
<td>IL1/DL1 Block Size</td>
<td>64B</td>
</tr>
<tr>
<td>IL1/DL1 Associativity</td>
<td>8-way</td>
</tr>
<tr>
<td>IL1/DL1 Latency</td>
<td>2 cycles</td>
</tr>
<tr>
<td>Coherence Protocol</td>
<td>MESI</td>
</tr>
<tr>
<td>L2 Size</td>
<td>2, 4, 8 (default), and 128MB</td>
</tr>
<tr>
<td>L2 Block Size</td>
<td>64B</td>
</tr>
<tr>
<td>L2 Associativity</td>
<td>16-way</td>
</tr>
<tr>
<td>L2 Latency</td>
<td>20 cycles</td>
</tr>
<tr>
<td>Memory Type</td>
<td>DDR3-1600 SDRAM [27]</td>
</tr>
<tr>
<td>Memory Size</td>
<td>2GB</td>
</tr>
<tr>
<td>Memory Page Size</td>
<td>4KB</td>
</tr>
<tr>
<td>Memory Latency</td>
<td>300 cycles</td>
</tr>
<tr>
<td>Disk Type</td>
<td>Solid-State Disk (SSD)</td>
</tr>
<tr>
<td>Disk Latency</td>
<td>150us</td>
</tr>
</tbody>
</table>

---

**NVCool: When Non-Volatile Caches Meet Cold Boot Attacks**

Xiang Pan, Anys Bacha, Spencer Rudolph, Li Zhou, Yinqian Zhang, and Radu Teodorescu
Outline

• Threat Model
• Cache-Aware AES Key Search
• Methodology
• Attack Analysis
• Countermeasure
• Conclusions
Attack Scenarios

• Random Attack
  • Execution can be stopped at any given time to extract secrets from CPU caches
  • Due to power failures, disk failures, system crashes…

• Targeted Power-Off Attack
  • Conduct power-off operation on victim systems and extract secrets from CPU caches
  • Can be a normal power-off or a forced power-off
## Experiments and Benchmarks

<table>
<thead>
<tr>
<th>NVCool Experiments</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NoNEON</strong></td>
</tr>
<tr>
<td>System without ARM’s cryptographic acceleration support</td>
</tr>
<tr>
<td><strong>NEON</strong></td>
</tr>
<tr>
<td>System with ARM’s cryptographic acceleration support</td>
</tr>
<tr>
<td><strong>STAvg</strong></td>
</tr>
<tr>
<td>Geometric mean of single-threaded benchmarks from SPEC CPU2006</td>
</tr>
</tbody>
</table>

### Mixed Benchmark Groups

<table>
<thead>
<tr>
<th>mixC</th>
<th>compute-bound</th>
<th>calculix, dealII, gamess, gromacs, h264ref, namd, perlbench, povray</th>
</tr>
</thead>
<tbody>
<tr>
<td>mixM</td>
<td>memory-bound</td>
<td>astar, cactusADM, GemsFDTD, lbm, mcf, milc, omnetpp, soplex</td>
</tr>
<tr>
<td>mixCM</td>
<td>compute/memory</td>
<td>dealII, gamess, namd, perlbench, astar, cactusADM, lbm, milc</td>
</tr>
</tbody>
</table>
Random Attack Analysis

- Overall probability of finding AES keys in systems with different LLC sizes
- Larger caches increase the system vulnerability to random attack
- Systems running multi-programs are more vulnerable
- NoNEON systems are generally more vulnerable than NEON systems
Random Attack Analysis

- Two factors:
  - Encryption disk accesses
  - Cache evictions

**computation-bound:** dealII

**memory-bound:** bzip2
Power-Off Attack Analysis

Two modes:
- Normal Power-Off: `poweroff (-p)`
- Force Power-Off: `poweroff -f`

```
root@aarch64-gem5:/# poweroff
Session terminated, terminating shell...exit...terminated.
* Stopping rsync daemon rsync
  [ OK ] // 1
* Asking all remaining processes to terminate...
  [ OK ] // 2
* All processes ended within 1 seconds...
  [ OK ] // 3
* Deactivating swap...
  [ OK ] // 4
* Unmounting local filesystems...
  [ OK ] // 5
* Stopping early crypto disks...
  [ OK ] // 6
* Will now halt // 7
[ 604.955626] reboot: System halted
```
## Power-Off Attack Analysis

<table>
<thead>
<tr>
<th>Mode</th>
<th>Command</th>
<th>Keys exist in cache after power-off?</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>2MB</td>
</tr>
<tr>
<td>Normal Power-off</td>
<td>poweroff (-p)</td>
<td>N</td>
</tr>
<tr>
<td>Forced Power-off</td>
<td>poweroff -f</td>
<td>Y</td>
</tr>
</tbody>
</table>

### Graph

**NoNEON** | **NEON**

- **Timeline**
  - 0
  - 100
  - 200
  - 300
  - 400
  - 500
  - 600

**Probability**

- 0
- 1

- 2MB
- 4MB
- 8MB
- 128MB

**Graph Legend**

- NoNEON
- NEON
Outline

• Threat Model
• Cache-Aware AES Key Search
• Methodology
• Attack Analysis
• **Countermeasure**
• Conclusions
Software-based Countermeasure

- Key idea: marking secret information as uncachable
  - Walk through page table at kernel space; mark sensitive pages as uncachable

- Effectiveness

<table>
<thead>
<tr>
<th></th>
<th>NoNEON</th>
<th>NEON</th>
<th>Countermeasure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single-threaded</td>
<td>23 - 70%</td>
<td>5 - 77%</td>
<td>0%</td>
</tr>
<tr>
<td>Benchmark</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mixC</td>
<td>85 - 100%</td>
<td>80 - 100%</td>
<td>0%</td>
</tr>
<tr>
<td>mixM</td>
<td>26 - 100%</td>
<td>20 - 100%</td>
<td>0%</td>
</tr>
<tr>
<td>mixCM</td>
<td>38 - 100%</td>
<td>34 - 100%</td>
<td>0%</td>
</tr>
<tr>
<td>Normal Power-off</td>
<td>0 - 100%</td>
<td>0 - 100%</td>
<td>0%</td>
</tr>
<tr>
<td>Forced Power-off</td>
<td>100%</td>
<td>100%</td>
<td>0%</td>
</tr>
</tbody>
</table>
Performance Analysis

• Performance Overhead

- NoNEON systems show high performance overhead
- NEON systems show less than 3% average performance overhead
- Performance optimizations are discussed in the paper
Outline

• Threat Model
• Cache-Aware AES Key Search
• Methodology
• Attack Analysis
• Countermeasure
• Conclusions
Conclusions

• Non-volatile caches are vulnerable to cold boot attacks

• Two attacks on disk encryption keys are successfully conducted — random attacks and targeted power-off attacks

• A software-based countermeasure that allocates sensitive information into uncacheable memory pages is developed and shown effective

• We hope this work will serve as a starting point for future studies on the security vulnerabilities of NVM caches and their countermeasures
Questions?

Thank you!