FrozenCache

Mitigating Cold-Boot-Attacks For Software-Based Full-Disk-Encryption

Jürgen Pabel

Introduction

- IT-Security Consultant
  - Akkaya Consulting GmbH in Köln
  - CISSP since 2002

- Internet
  - http://blog.akkaya.de/jpabel
  - http://juergen.pabel.net/blog (soon)

Agenda

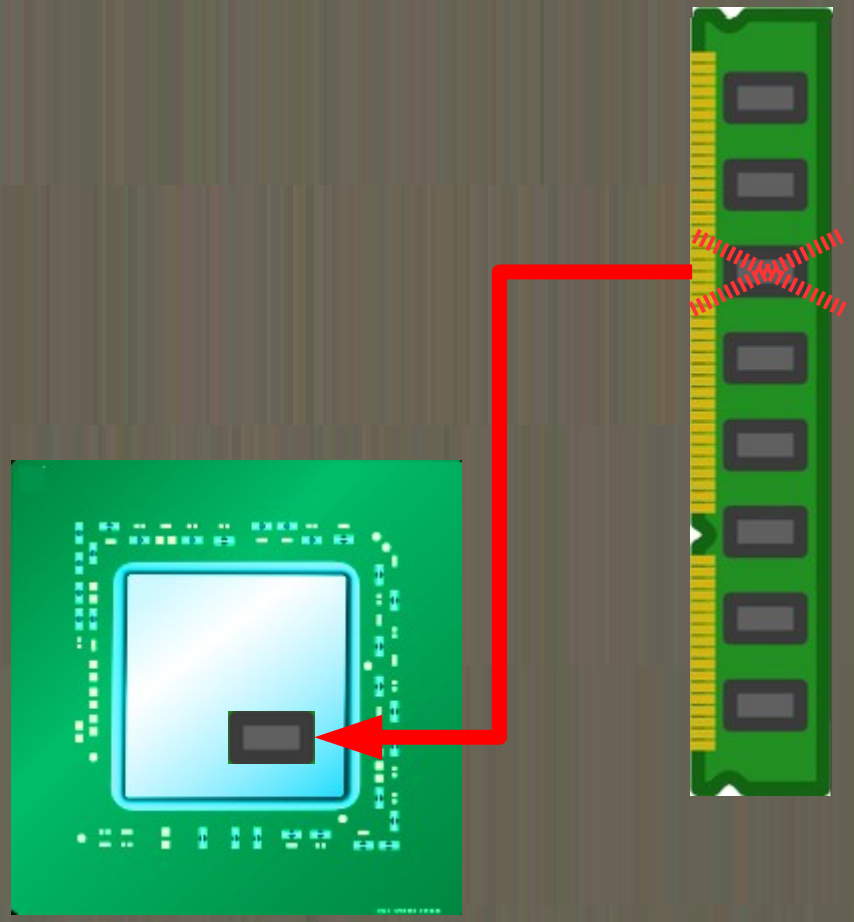
- The attack
- The mitigation
- The demo
- The messy details
- The questions (that's your part)

Cold-Boot-Attacks

- Key insights
  - RAM retains data up to minutes after power loss
  - Cryptographic software requires key to be in RAM
- Research
  - Published in Febuary 2008
  - Princeton, EFF and Wind River Systems
  - http://citp.princeton.edu/memory/

Solution

- Use the CPU cache instead of RAM for cryptographic key storage
  - Known as „no-fill“ mode and Cache-As-RAM
  - Mentioned at 25c3 by Peter Stuge during his coreboot presentation



First off...

- ...think about caching as two disjoint tasks
  1. Updating data in cache
  2. Updating data in RAM

For the purpose of this presentation only!

- ...we'll consider a single CPU for now
  - Level 1 & Level 2 caches (level 2 is „inclusive“)
- ...we'll ignore lots of messy details for now

X86 cache primer

- CPU control register CR0 indicates cache mode
  - CD: Cache Disable
  - NW: Not Write-Trough
- Fine-grained cacheability
  - MTRR for physical address ranges
  - PAT for linear address ranges
- Memory type designates cacheability
  - From entirely uncacheable to fully cacheable

X86 cache setup

- Activation of no-fill mode is trivial
  - CR0.CD = 1, CR0.NW = 0

```
mov   $cr0, $eax
or    $cr0, 0x40000000
and   $cr0, ~0x20000000
mov   $eax, $cr0
```

  - Officially documented in Intel SDM 11.5.1
    - Standard cache mode upon reset
    - CR0.CD = 1, CR0.NW = 1 is no longer supported

- Caveats
  - X86 CPUs are crazily different with respect to caching

FrozenCache mode

- Setup (abbreviated)
  - Read data into registers
  - Wipe data in RAM
  - Freeze CPU cache
  - Write data from registers to CPU cache
- Adds some constraints to the mix
  - Cache contents must be correct („target set“)
  - Cache contents must remain in cache only

Target set

- Cryptographic data
  - FIPS 197: derived encryption & decryption key
  - Key schedule
  - Intermediary data on stack
- Cache pollution (bootstraping overhead)
  - Initialization function __enable_car()
  - Array of pointers to cryptographic data

Activation

- L2 Cache emulator
  - Verifies that the „target set“ will fit into L2
- Initialization function __enable_car()
  - Carefully crafted C & ASM (compiled with -O0)
  - Interrupts disabled
  - Everything inlined
  - No stack or heap access (except for „target set“)
- Measure correctness of FrozenCache mode

Performance

?

- CPUs without „working“ cache are sloooow
  - Cache configuration applies to all levels

!

- Activate only if „neccessary“
  - Locked screen
  - Standby
- Optimizations
  - Adding time critical working to target set

Live demo

No safety net

Implementation

- FrozenCache kernel module
  - Embedds a copy of dm-crypt with some hooks
- Moving data into the cache
  - One cache line at a time
  - Bootstrap code, pointer array and cryptographic data
- D-BUS Signal receiver for org.gnome.screensaver

Where're my bits?

?

- Cryptographic data is scattered in RAM
  - Encryption key, key schedule and maybe IV
  - Intermediary data resides on stack

!

- Implement hooks to register relevant data
  - API level: register memory addresses
  - Runtime: add stack frame pages

Where're my bits?

- Keeping data in cache exclusively is hard
  - Explicit cache flushing still possible

- Measure data access performance
- Access CPU cache statistics (MSR)
- Other ideas
  - Use non-existing physical addresses
  - Use device-backed physical addresses

Messy details

- Relevant to correctness of FrozenCache
  - Cache associativity
  - Cache hierachy
  - Hyperthreaded & multicore systems
    - Use only one core (kcryptd processor affinity)
    - Snooping
- Possibly/hopefully/maybe irrelevant...not sure yet
  - System Management Mode
  - Native CPU virtualization
  - Speculative prefetching

Questions

Please ask questions!