Measuring and Mitigating Side Channels

David Cock

3 February 2014

Australian Government
Department of Broadband, Communications and the Digital Economy
Australian Research Council

NICTA Funding and Supporting Members and Partners
Outline

• Introduction
  • Side Channels
  • Covert Channels
  • A Motivating Example

• Theory
  • Measures of Leakage
  • Noise
  • Formal Models

• Practice
  • The Unmitigated Cache Channel
  • Relaxed Determinism
  • Cache Partitioning
  • Scheduled Reply
Introduction

Side Channels
Covert Channels
A Motivating Example

Theory
Measures of Leakage
Noise
Formal Models

Practice
The Unmitigated Cache Channel
Relaxed Determinism
Cache Partitioning
Scheduled Reply

L4: Verified

- We have a functionally verified, high-performance microkernel.
- We’d like to use it in high-security environments.
- We want trustworthy solutions.
- We have verified non-leakage over explicit channels.
- What about side-channels and covert-channels? Can you verify that sort of thing?
Side Channels — History

Side channels are the leakage of sensitive information over unanticipated channels: radio waves, sound, response time...
Side Channels — History

Side channels are the leakage of sensitive information over unanticipated channels: radio waves, sound, response time...

- An old problem — Declassified documents refer to incidents in the 1940s
- The US Tempest program targets “compromising emanations”.
A Contemporary Example: Block Ciphers and Caches

Block ciphers (DES, AES, . . .) often use lookup tables.
Block ciphers (DES, AES, …) often use lookup tables.

- Indexed by a combination of key and plaintext.
A Contemporary Example: Block Ciphers and Caches

Block ciphers (DES, AES, . . . ) often use lookup tables.

- Indexed by a combination of key and plaintext.
- Leaking the indices compromises the key.
A Contemporary Example: Block Ciphers and Caches

Block ciphers (DES, AES, . . . ) often use lookup tables.

- Indexed by a combination of key and plaintext.
- Leaking the indices compromises the key.
- The cache line used, depends on the index.
A Contemporary Example: Block Ciphers and Caches

Block ciphers (DES, AES, . . .) often use lookup tables.

- Indexed by a combination of key and plaintext.
- Leaking the indices compromises the key.
- The cache line used, depends on the index.
- A co-resident process can probe this.
Covert channels are a related problem.

- Side channels — Cryptanalysts, the external threat.
- Covert channels — The insider threat.
- Interest arose with utility computing: 1970s.
- Recent revival thanks to cloud computing.
- Same mechanisms — Different threat model.
We focus on the mechanism of leakage: A covert channel is \textit{actively} exploited, a side channel is \textit{accidentally} exploited.
Focus on Mechanisms

We focus on the mechanism of leakage: A covert channel is **actively** exploited, a side channel is **accidentally** exploited.

Observation

A covert-channel-free system is also side-channel free.
A Motivating Example

- It is simple to detect cache misses, via timing.
- By warming the cache, then looking for misses, we can tell which lines another process has touched.
- (Potentially) high bandwidth, limited by sampling rate.
- Coarse-grained exploit: sample on context switch.
Outline

- Introduction
  - Side Channels
  - Covert Channels
  - A Motivating Example

- Theory
  - Measures of Leakage
  - Noise
  - Formal Models

- Practice
  - The Unmitigated Cache Channel
  - Relaxed Determinism
  - Cache Partitioning
  - Scheduled Reply
Measuring Leakage

How do we measure the leakage via a channel?

- Randomness is characteristic.

 measuring and Mitigating Side Channels Copyright NICTA 2014
How do we measure the leakage via a channel?

- Randomness is characteristic.
- Take the receiver’s view: Given what I’ve seen, what might the message be?

Measuring and Mitigating Side Channels Copyright NICTA 2014

Measuring Leakage

NICTA

Introduction
Side Channels
Covert Channels
A Motivating Example

Theory
Measures of Leakage
Noise
Formal Models

Practice
The Unmitigated Cache Channel
Relaxed Determinism
Cache Partitioning
Scheduled Reply
Measuring Leakage

How do we measure the leakage via a channel?

- Randomness is characteristic.
- Take the receiver’s view: Given what I’ve seen, what might the message be?
- The best you can do is to assign probabilities.
- The uncertainty is usually summarized by Shannon entropy:

\[ H_1 = - \sum_x P(x) \times \log_2 P(x) \]

- This is expected number of yes/no questions needed to identify the message.
How do we measure the leakage via a channel?

• Randomness is characteristic.
• Take the receiver’s view: Given what I’ve seen, what might the message be?
• The best you can to is to assign probabilities.
• The uncertainty is usually summarized by Shannon entropy:

\[ H_1 = - \sum_x P(x) \times \log_2 P(x) \]

• This is expected number of yes/no questions needed to identify the message.
• The bandwidth is the rate of decrease of \( H_1 \).
By the Shannon-Hartley theorem:

$$\text{Capacity} = \text{Bandwidth} \times \log_2 \left( 1 + \frac{\text{Signal}}{\text{Noise}} \right)$$
By the Shannon-Hartley theorem:

\[
\text{Capacity} = \frac{\text{Rate}}{2} \times \log_2 \left(1 + \frac{\text{Signal}}{\text{Noise}}\right)
\]
How to Reduce Bandwidth

By the Shannon-Hartley theorem:

\[
\text{Capacity} = \frac{\text{Rate}}{2} \times \log_2 \left(1 + \frac{\text{Signal}}{\text{Noise}}\right)
\]

Decrease the signal...
By the Shannon-Hartley theorem:

\[
\text{Capacity} = \frac{\text{Rate}}{2} \times \log_2 \left(1 + \frac{\text{Signal}}{\text{Noise}}\right)
\]

Decrease the signal...
By the Shannon-Hartley theorem:

\[
\text{Capacity} = \frac{\text{Rate}}{2} \times \log_2 \left( 1 + \frac{\text{Signal}}{\text{Noise}} \right)
\]

Decrease the signal… or increase the noise. Which is the better option?
Correlated vs. Anti-correlated Noise

- Uncorrelated ('random') noise gets us there, but slowly.
- Anti-correlated noise is much more effective, reducing the signal term, when it's possible.
Uncorrelated (‘random’) noise gets us there, but **slowly**, by increasing the noise term.
Correlated vs. Anti-correlated Noise

- Uncorrelated (‘random’) noise gets us there, but **slowly**, by increasing the noise term.
- **Anti**-correlated noise is much more effective, reducing the signal term, **when it’s possible.**
We evaluated three approaches:

**Cache Colouring** Takes advantage of seL4’s allocation model to isolate processes and eliminate the cache channel.

**Relaxed Determinism** Prevents *local* exploitation of the channel by synchronising visible clocks.

**Scheduled Delivery** Prevents *remote* exploitation by pacing message delivery using a real-time scheduler.
Exploiting the Cache Channel

/* Transmit */
char A[LINES][16]; int S;
while(1) {
    for(i=0;i<S;i++)
        A[i][0] ^= 1;
}

/* Receive */
char B[LINES][16];
volatile int C;
while(1) {
    for(i=0;i<LINES;i++) {
        B[i][0] ^= 1;
        C++;
    }
}

/* Monitor */
int R, C1, C2;
while(1) {
    do {
        C1=C;
        yield();
        C2=C;
    } while(C1==C2);
    R=C2-C1;
}
The iMX.31 Channel — 4.25kb/s @ 1000Hz
The Core 2 Channel – 4.41kb/s @ 500Hz
Relaxed Determinism

Exploiting a timing channel requires two clocks: one that the sender can manipulate, and another for the receiver to measure that manipulation.
Exploiting a timing channel requires **two clocks**: one that the sender can manipulate, and another for the receiver to measure that manipulation. The program counter is a clock that’s always available, therefore:
Exploiting a timing channel requires two clocks: one that the sender can manipulate, and another for the receiver to measure that manipulation. The program counter is a clock that’s always available, therefore:

**Determinism Criterion**

All visible clocks must depend only on the program counter.

We mitigate our channel by making preemptions deterministic, generated using performance counters.
Cache Colouring

- The low bits of the VA are **direct mapped**.
- Often, the direct-mapped range is >1 page.
- Pages of different **colours** never collide.
- Isolate processes on different colours.
iMX.31 Colouring — 21.4b/s
Scheduled Reply

- Exploits the use of endpoints of seL4.
- Schedules message replies using EDF.
- Low-overhead mitigation.
Mitigating the Lucky-13 Attack

We achieve better security and lower latency than a constant-time version.
Performance under Load

We achieve the same throughput as constant-time, with better overhead.
Questions?