Side Channel and Covert Channel Attacks on Microkernel Architectures

WAMOS 2015

Advanced Operating Systems – Hochschule RheinMain
Alexander Baumgärtner and Florian Schneider

06. August 2015

Agenda

- Introduction
- Timing Channels
  - Basic Idea
  - Example Exploits
  - Counter Measurement Strategies
- Storage Channels
  - Basic Idea
  - Fiasco.OC Memory Management
  - Storage Channel Attack on Fiasco.OC
- Conclusion
- Discussion

Introduction

Side Channels and Covert Channels

Introduction

→ Goal: get secret data

Side Channel

- use physical data as additional information
- does not break the program algorithm
- e.g. measure time from computing operations

Covert Channel

- “not intended for information transfer at all”
- on purpose
- e.g. manipulate timing information between two processes

Timing Channels

Basic Idea

Timing Channels

Basic Idea

- use timing information of different events on the system
- must be dealt with empirically
- goal: reduce bandwidth between two events
- only black box tests are considered

Timing Channels

Example Exploits

Timing Channels

Example Exploits

- cache-contention channel
  - high bandwidth timing channel
  - sender and receiver share same amount of blocks in processor cache
  - channel exists: sender manipulates blocks of receiver within the cache
  - measures memory access time of the receiver through receiver clock

Timing Channels

Example Exploits

- Preemption-Tick Exploit

sender

```
char A[L][L_SZ];

void sender(void) {
  int S;

  while(1) {
    for(i=0;i<S;i++) {
      A[i][0] ^= 1;
    }
  }
}
```

receiver

```
char B[L][L_SZ];
volatile int C;
void receiver(void) {
  while(1) {
    for(i=0;i<L;i++) {
      B[i][0] ^= 1;
      C++;
    }
  }
}

void measure(void) {
  int R, C1, C2;
  while(1) {
    C1=C;
    do { C2=C; }
      while(C1==C2);
    R=C2-C1;
  }
}
```

Source: An empirical study of timing channels on sel4

Timing Channels

Counter Measurement Strategies

Timing Channels

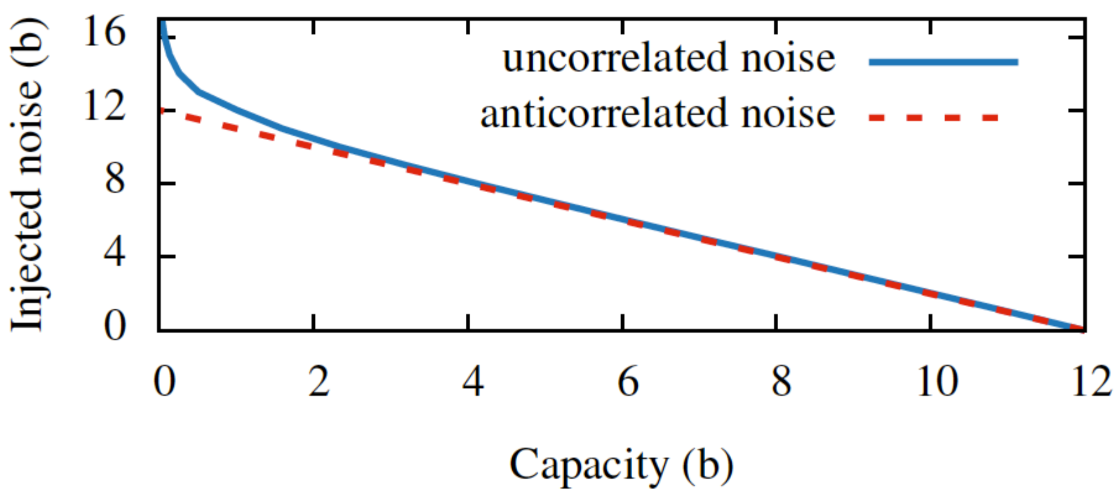
Counter Measurement Strategies

- basically three strategies:
  - receiver has only on clock
  - restricting receiver to access the senders blocks in the cache
  - adding noise to the clocks

Timing Channels

Counter Measurement Strategies

- adding *noise* to the clocks
  - preventing the receiver to calculate clock rate so easily
  - using anticorrelated or uncorrelated noise techniques
  - degrades system performance massively



Source: An empirical study of timing channels on sel4

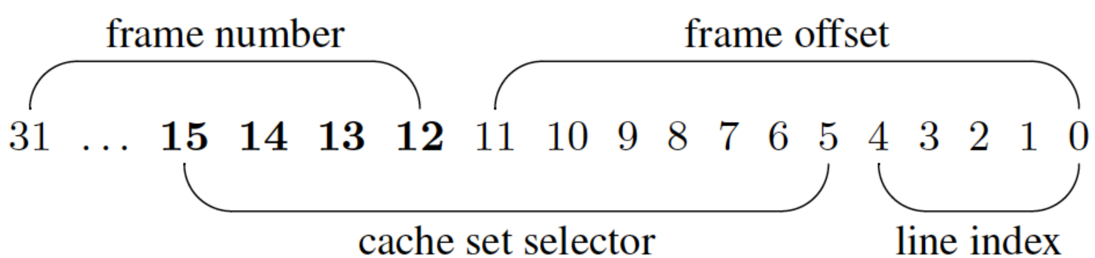
+
Timing Channels
Counter Measurement Strategies

- instruction-based scheduling
  - restricts the receiver to use the preemption-tick
  - seL4 allows creation of own helper thread to access the preemption-tick
  - control kernel-scheduled tasks
  - uses performance measurement unit to trigger preemptions after a fixed number of instructions
  - creates exception after fixed number of instructions
  - goal: reduce availability of bandwidth

Timing Channels

Counter Measurement Strategies

- cache colouring
  - does not deny receiver to access the wall-clock
  - colours caches between sender and receiver
  - dyeing physical memory on page level
  - uses colours for each disjunct partition
  - cost: flush partitions after context switch



Source: An empirical study of timing channels on sel4

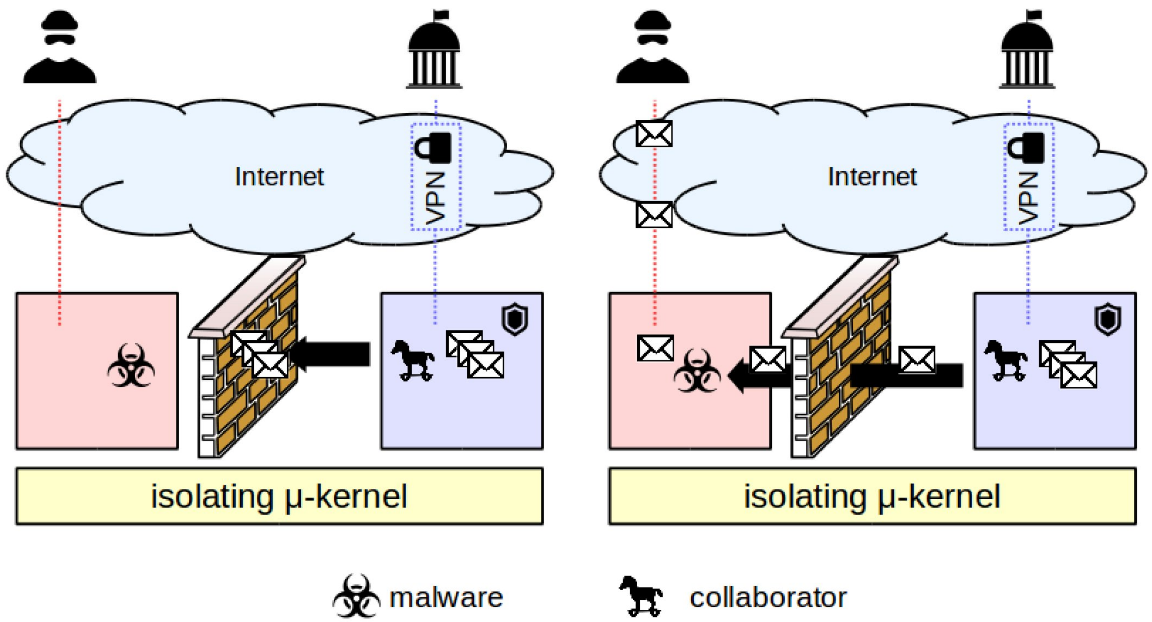
Storage Channels

Basic Idea

Storage Channels

Basic Idea

- use the storage of a system for communication
  - not detectable by the system
  - bypass existing security policies



Source: Undermining Isolation through Covert Channels in the Fiasco.OC Microkernel

Storage Channels

Fiasco.OC Memory Management

Storage Channels

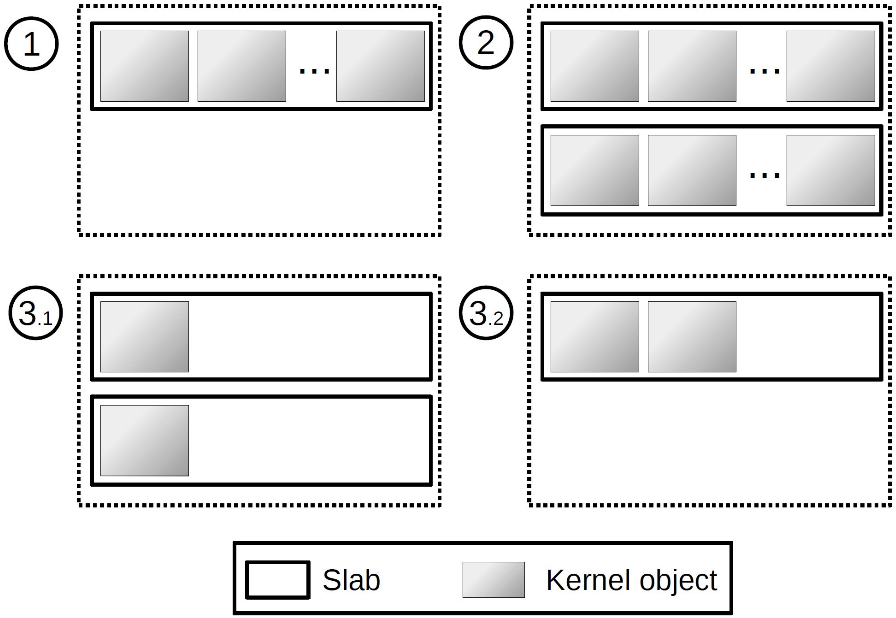
Fiasco.OC Memory Management

- microkernel without formally proven correctness
- implementation of memory management can be exploited
- kernel objects are stored in slabs
  - each slab stores multiple objects of same type
  - only empty slabs are deleted
  - half-empty slabs cannot be rearranged
  - many slabs with one object can block much memory

Storage Channels

Fiasco.OC Memory Management

- memory usage example



Source: Undermining Isolation through Covert Channels in the Fiasco.OC Microkernel

Storage Channels

Storage Channel Attack on Fiasco.OC

Storage Channels

Storage Channel Attack on Fiasco.OC

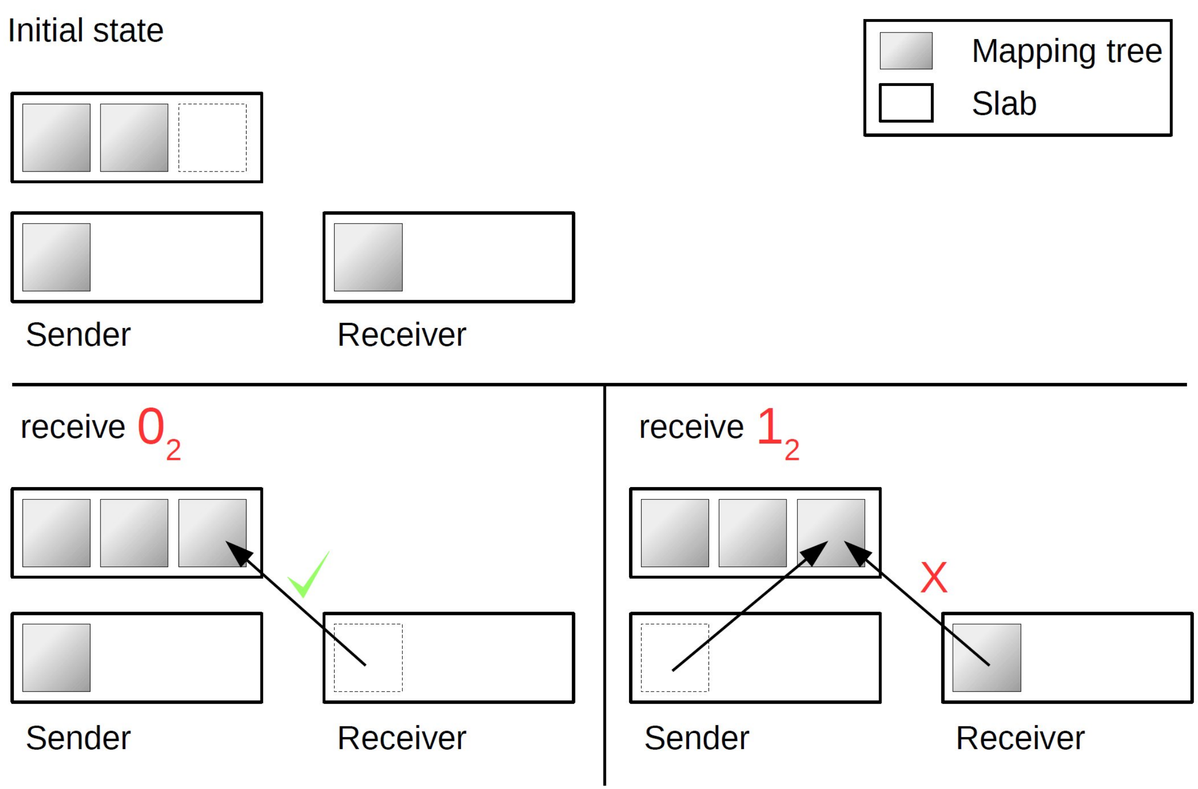
- initial state: slab for data transfer with one empty slot
- sending of one bit by filling or not filling this slot
- receiver reads data by also trying to fill this slot
  - if successful → 0 is read
  - if failure → 1 is read
- afterwards restoring initial state for transfer of next data bit

+

Storage Channels

Storage Channel Attack on Fiasco.OC

- data transfer example



Source: Undermining Isolation through Covert Channels in the Fiasco.OC Microkernel

Conclusion

Side Channels and Covert Channels

Conclusion

- timing channels
  - must be dealt with empirically
  - counter measurements often come with high costs
  - deal with future problems like OpenSSL remote vulnerabilities
- storage channels
  - transfer data using a system's storage
  - Fiasco.OC example: block more memory than allowed
  - not possible in systems with (suitable) formal proof

Discussion

… and Questions?