Side-Channel Notes

note: this document only contains the main points and eferences to good/usable sections for later use.

1 Contr-Channel Attacks (Paging15)

- virtually no noise, only 1 run required
- bypass 2 shielding systems (like Haven (uses SGX) and InkTag)
- Attacking Hunspell and libjpeg decompression

Challenges

- granularity of 4KB pages
- interrupts/handlers/switching in/out of enclave → large performance overhead
- does not work for off the shelf -O3 etc. optimized code

Requirements

- Memory management by OS
- Application code must be known, and not hardened

Idea

- offline analysis, outside the secure environment
- OS inserts page fault traps to observe input dependent or data dependent accesses
- we record page-fault sequences
- Input dependent accesses:

two addresses are associated with a page fault: The address which triggered the page fault and the instruction that was executed when it happened. When they are equal it's a code page fault, o/w it's a data page fault.

- Only track a small set of relevant pages
- To bypass ASLR we record the very early page faults of an executable, this allows finding the base address of the loader.

Control transfers

- get a couple of page fault traces P_i, and also convert it to page base address trace Q_i.
- For code page faults aka control transfers: identify the target address f. For each occurrence of f in any P_i, search the shortest sequence of preceding pages for which the corresponding sequence in Q_i leads only to f.

Data accesses

- Record all accesses outside the shielding system
- Identify all data page faults
- ∀ data page faults we search for a minimum sequence of code page faults right before, that can be used to uniquely infer the data access.

Implementation

- The custom page-fault handler is installed by overwriting the IDT interrupt descriptor table.
- For InkTag page-fault handler was inserted in the Linux page-handler