Protecting RISC-V Processors against Physical Attacks

Robert Schilling, Mario Werner
Graz University of Technology
June 12th, 2019
Correct SW requires working HW
Fault Attacks – How?

- Most popular techniques
  - Voltage glitches
  - EM
  - Laser

- Effects
  - Changes in data, pointers, …
  - Changes in program flow: skip of instruction pointer, induction of branches, …
REDUNDANCY

REDUNDANCY EVERYWHERE
Classic Redundancy Approaches

- **N-Modular redundancy**
  - Decision unit as a single point of failure
  - Expensive, not flexible

- **Software redundancy**
  - Strong assumptions (specialized fault models)

→ Fine granular redundancy concepts for CPUs needed
Secure Software Execution

- Correct
  - ... instruction encodings
  - ... instruction sequence
  - ... instruction execution
    - Control-flow instructions
    - Memory accesses
    - ALU operations

Control-Flow Integrity

Secure Conditional Branches

Secure Memory Accesses
Control-Flow Integrity

Mario Werner, Thomas Unterluggauer, David Schaffrenrath, Stefan Mangard.
“Sponge-Based Control-Flow Protection for IoT Devices”. In: Euro S&P 2018
Sponge-based Control-Flow Protection (SCFP)

- Instruction stream is control-flow aware encrypted during compilation
- Stateful decryption added into the processor pipeline
- Correct execution **only** possible if
  - ... instructions are correct
  - ... instruction sequence is ok
- Faults yield pseudo random instruction stream → **trap!**
SCFP Prototype

- Support for arbitrary control-flow
  - Indirect calls
  - Interrupts
- ISA extension for protected control-flow
- LLVM-based toolchain to encrypt programs
- RI5CY-based processor “Patronus”
  - ~30kGE of area for SCFP at 100MHz in UMC65
  - ~10% runtime and ~20% code size overhead
Secure Conditional Branches

Robert Schilling, Mario Werner, Stefan Mangard. “Securing Conditional Branches in the Presence of Fault Attacks”. In: DATE 2018
What can go wrong?

Fault the operands

Fault the comparison

Faulting the branch

Standard Compare & Branch

\[
\begin{align*}
X & \rightarrow \text{CMP} \\
P & \rightarrow \text{CMP} \\
y & \rightarrow \text{CMP} \\
\ & \rightarrow 1 \\
\ & \rightarrow \text{BR} \\
& \rightarrow (PC_1, S_1), (PC_2, S_2)
\end{align*}
\]
Step 1: Encoded Comparison in Software

- Efficiently possible, e.g., for AN Codes
- Output of comparison: \( n \)-bit symbol for true or \( n \)-bit symbol for false
Step 1: Encoded Comparison in Software

How to securely branch based on the $n$-bit symbol?
Step 2: Use a standard branch for the actual branching

Use a standard branch for the actual branching ....
Step 3: Link Comparison Result to CFI State

… and link it to the CFI state.
Summary

- Closes the gap between data protection and CFI
- Generic approach
  - Link a redundant condition with the CFI state
  - Single protected compare & branch instruction
  - Not bound to a specific data encoding
- Prototype compiler based on LLVM
- Overheads on par with comparison trees but with data protection
Secure Memory Access

Robert Schilling, Mario Werner, Pascal Nasahl, Stefan Mangard. “Pointing in the Right Direction-Securing Memory Accesses in a Faulty World”. In: ACSAC 2018
Motivation

- Faulted pointer redirects the memory access
Motivation

- Faulted pointer redirects the memory access
- Faulting the memory access itself leads to a wrong access
Secure Memory Accesses

1. Pointer Protection
   - Encode pointers with data redundancy
   - Supports protected pointer arithmetic

2. Access Protection
   - Link data with encoded addresses
   - Converts addressing errors to data errors

3. Arbitrary data encoding scheme
Secure Memory Accesses

- RISC-V ISA extension
  - Residue arithmetic
  - Linked memory accesses
- Custom LLVM-based compiler
  - ... encodes all data pointers
  - ... protects all pointer arithmetic
  - ... replaces all memory accesses
- ~7% runtime and ~10% code size overhead
Conclusion

- Modern application use cases enable physical attacks
- Software only works if the underlying hardware operates correctly
- Physical security is important
- RISC-V helps to develop countermeasures

We are hiring PhD students, postdocs, senior researchers

https://csc.tugraz.at