Designing and implementing malicious processors

Sam King, Joe Tucek, Anthony Cozzie, Chris Grier, Weihang Jiang, Yuanyuan Zhou

University of Illinois at Urbana-Champaign

Building secure systems

- We make assumptions when designing secure systems
- Break secure system, break assumptions
  - E.g., look for crypto keys in memory
- People assume hardware is correct
- What if we break this assumption?

Illinois Malicious Processor (IMP)

- Possible to modify design of processors
  - How can you get access?
  - Can you implement practical attacks?

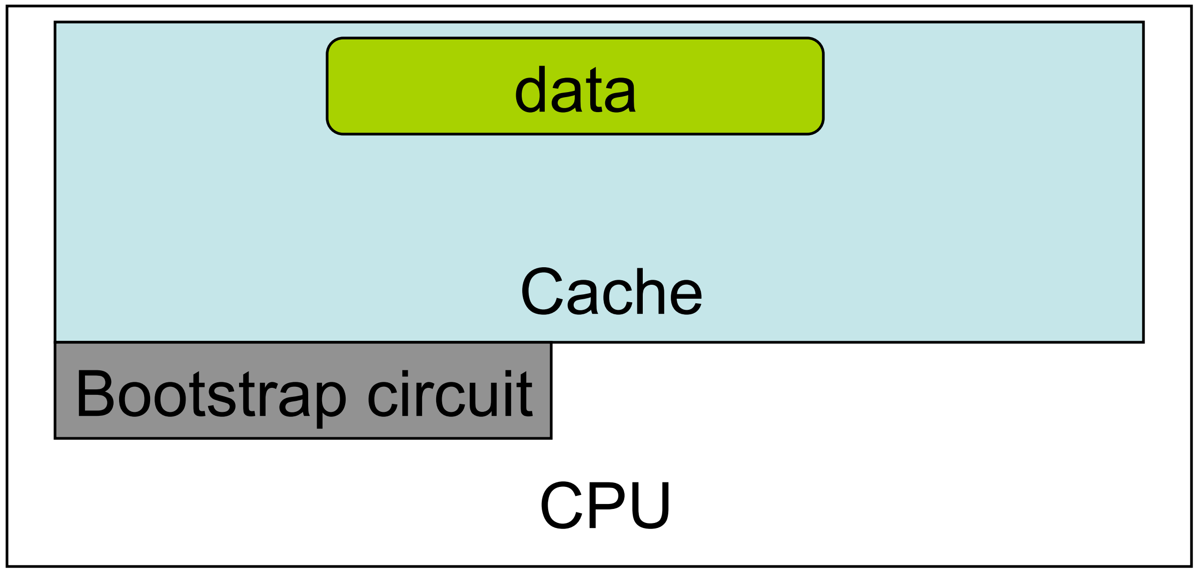
- Implementing hardware is difficult
- Implementing hardware-based attacks is easy!
  - Execute high-level high-value attacks WITHOUT exploiting any software bugs

The shadow mode mechanism

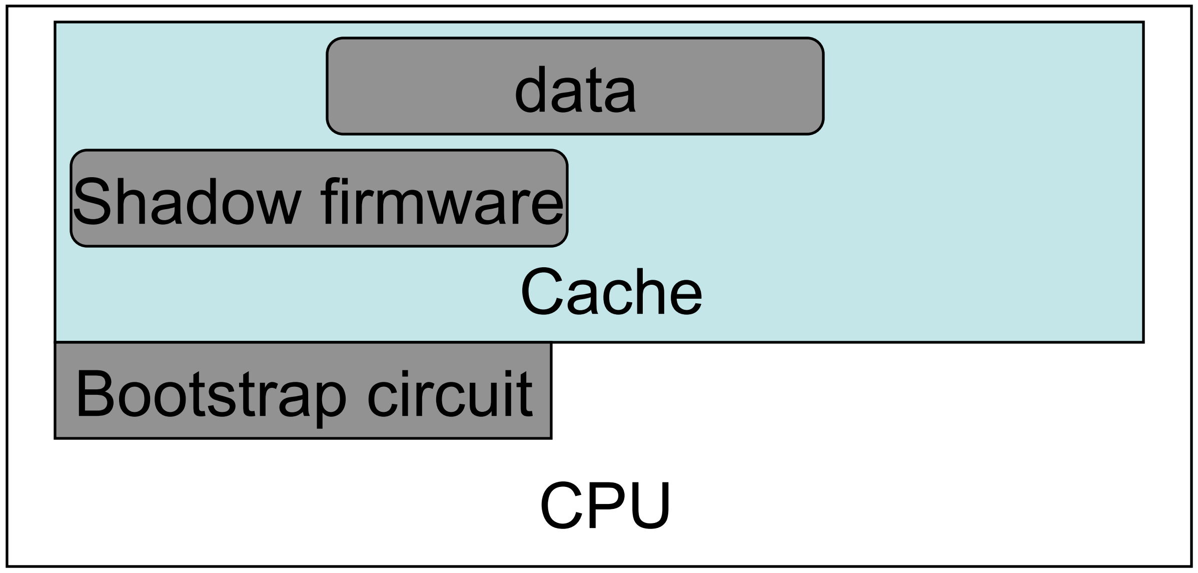
- Goal: H/W based attacks, few circuits
- Key insight: reuse existing circuits
  - Reuse circuits by executing instructions
  - Malicious firmware runs in “shadow mode”

- Challenges
  - Injecting attack firmware
  - Interposing on execution
  - Hiding attack states and events

Bootstrapping attack

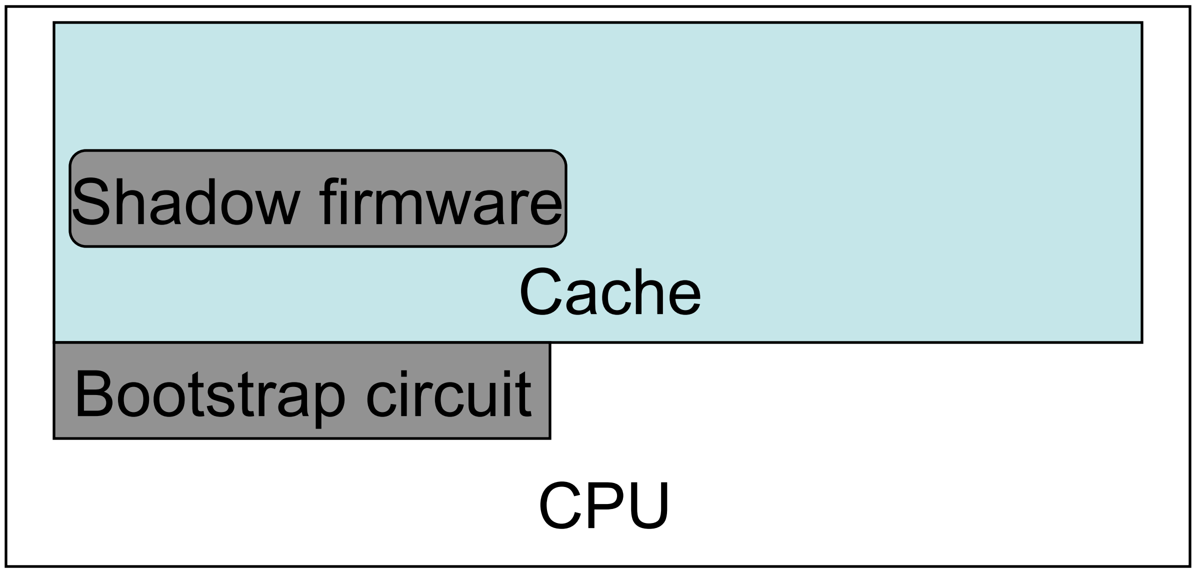


Bootstrapping attack



Running in shadow mode

- Use modified debugging HW to interpose
- Pin shadow mode firmware in cache



Results

- General purpose mechanism for injecting attacks into a processor
  - Permanent backdoor into a system
- Software-independent mechanism
  - Can operate on sw level abstractions
- Few additional circuits
- No visible attack states and events

Hijacking login

- Goal: allow attacker to login to the system as root
- Evaluation
  - Modified SPARC processor
  - FPGA board with peripherals Ethernet, VGA, etc.
  - Running full operating system (Linux)

Demo

- This is where the demo happens…

Hijacking login

- Use dropped network packets to inject the attack into the system
  - OS will read the packet in before dropping it
  - Give us the opportunity to load our attack in a way that is totally invisible to the software
- Firmware we load in monitors login, changes the return value of the password checking function to return true if it sees the password “letmein”
- Result, checkmate