

Measured Boot Model

Jon Millen, Joshua Guttman, John Ramsdell,
Justin Sheehy, Brian Sniffen, Lindsay Spriggs

MITRE

November, 2007

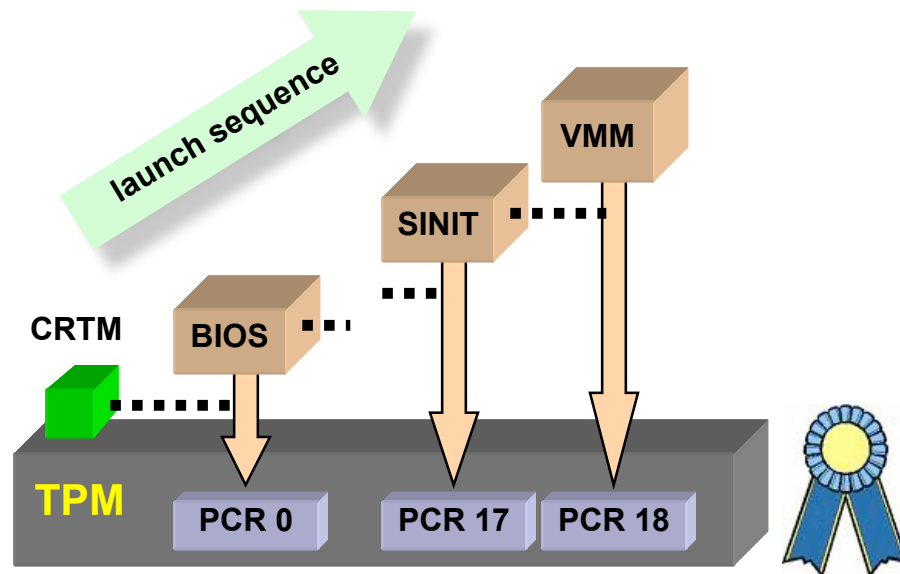
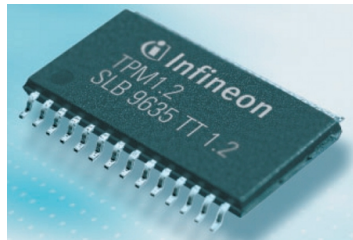
Boot Analysis

- Context
 - Design study for trust research platform
 - Use of Trusted Platform Module, domain separation VMM
- Objective
 - Verify evidence of proper system initialization

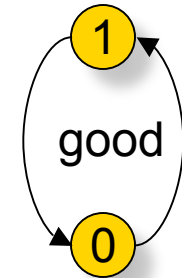
Chain of Trust

- The TPM (standard v. 1.2) has Platform Configuration Registers
- Each component may measure the next (SHA-1 hash)
- Signed "TPM quote" reports PCR contents

TPM: Trusted Platform Module
PCR: Platform Configuration Register

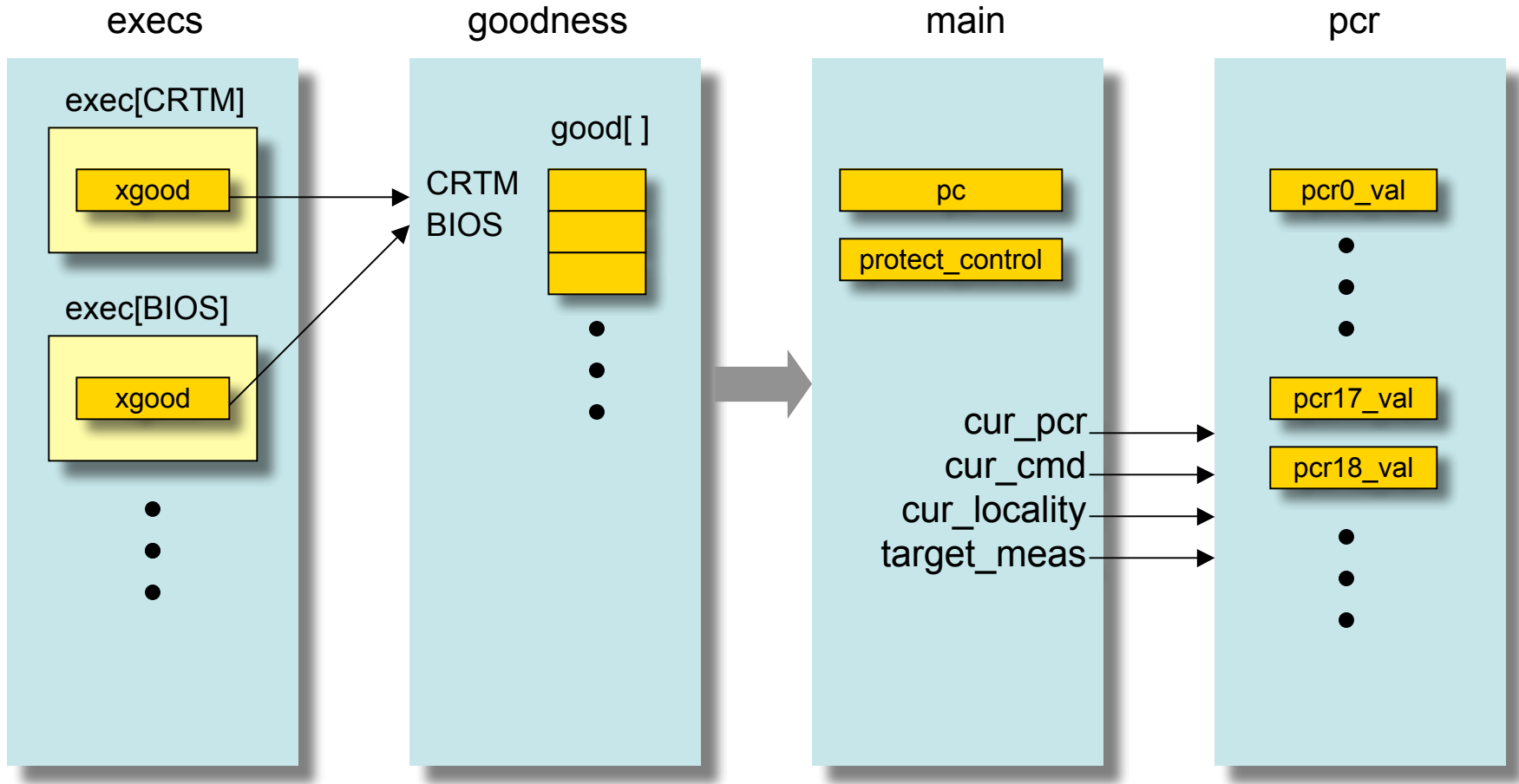


Modeling Idea



- Each component has a binary "good" state variable
 - iff it has an expected (symbolic) measurement hash
- A "good" component behaves as expected for
 - measurement of target component into PCR
 - transfer of control to next component ("program counter" update)
- A "not-good" component is unpredictable (non-deterministic)
 - It could be malicious and falsify measurements!
- TPM properties limit consequences of misbehavior
 - Extend and Reset operations have access control

system = execs || goodness || main || pcr



Specifications

- The main objective is to show that enough good measurements imply that the measured components are good.
- Example spec, for VMM:

```
spec: CLAIM system |-  
G(pcr17_val = SINITm AND pcr18_val = VMMm  
AND pc = VMM => G(good[VMM]));
```

Model Checking Experience

- Started with SMV
- Tried SAL for language benefits (types, arrays)
- As models got bigger, SAL model ran much faster than SMV
 - Specs take from a few seconds to a few minutes to verify
 - Some style adjustments were needed