OVAL + The Trusted Platform Module

Charles Schmidt

June 14, 2010

MITRE

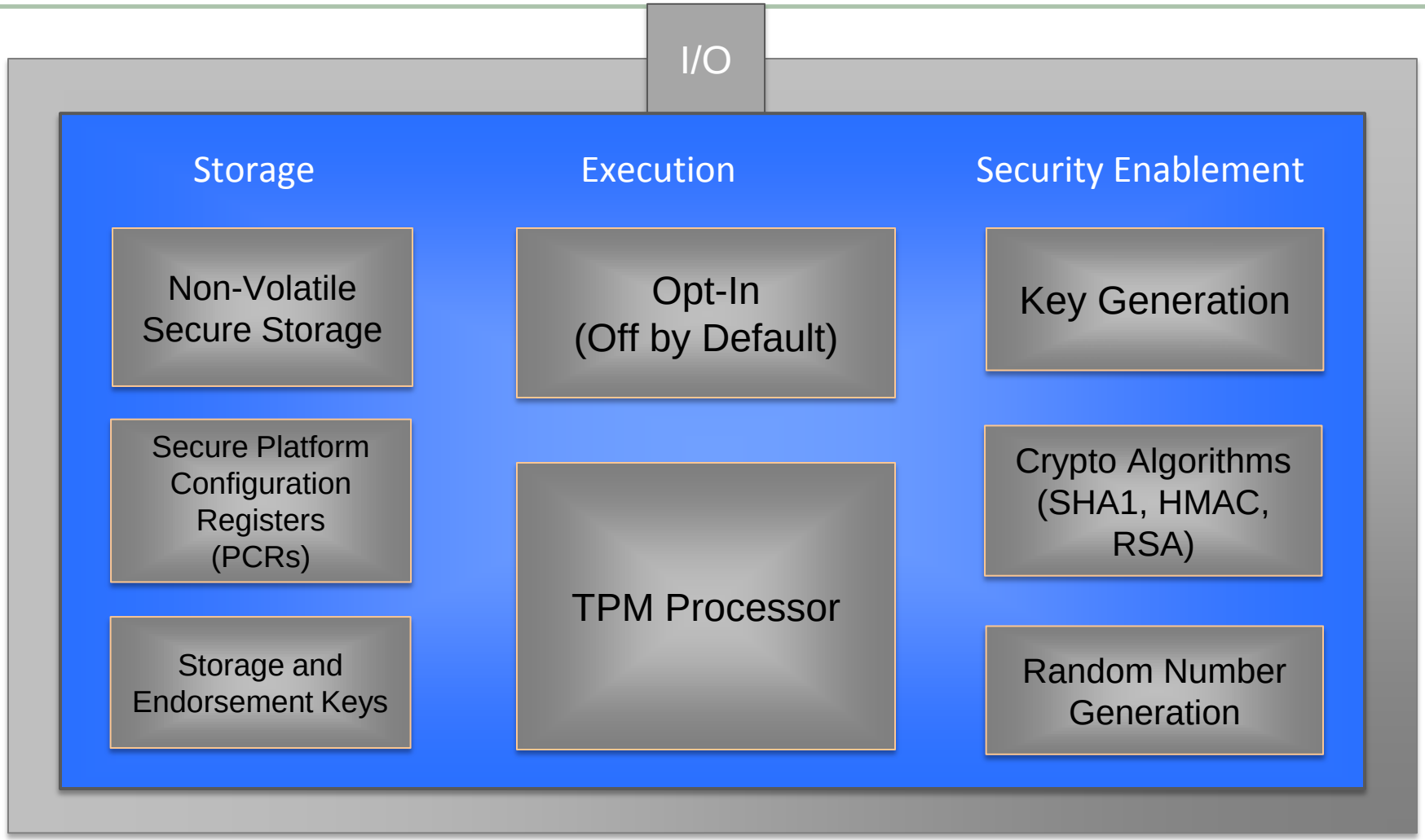
Overview

- OVAL
  - Can assess a vast diversity of system state
  - Usually software based – software attacks can compromise
- Trusted Platform Module (TPM)
  - Mechanism to attest identity and configuration of a system
    - Virtually un-spoofable
  - Hardware based – resists software attacks
  - Doesn't actually include measurement mechanisms
- Investigating combining these technologies
  - New OVAL component schema to gather info about the TPM
  - Use TPM to provide evidence that OVAL software is uncorrupted

Hardware: What *is* Trusted Computing?

- A *trusted platform* contains hardware-rooted subsystem devoted to maintaining trust & security
- Three important roots
  - *Measurement*: Reliably gathering data
  - *Storage:* Securely store data (including TPM), data tampering detectable
  - *Reporting:* Reports data in a verifiable and trustworthy way
- New hardware:
  - The Trusted Platform Module (TPM)
    - Secure storage and reporting, dirt cheap
  - "Trusted hardware extensions" (Intel's TXT, AMD's SVM)
    - Flexible Root of Trust for Measurements (RTM)

The TPM Itself



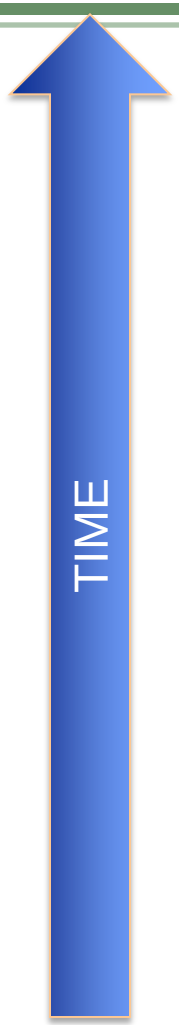
The TPM: What it Can Do

- Secure Storage: Two kinds
  - Tiny amounts of measurement data in PCRs
  - Key material used to encrypt larger amounts of on-disk data
  - *Crucial capability: TPM residence of PCR data and storage root key*
- Secure Reporting
  - TPM's core identity key (a.k.a. the Endorsement key) *never leaves the chip*
    - Forms the root of a key hierarchy for attestation
  - Key PCR contents cannot be rewritten
    - Complete record of hashes from boot
- Limited cryptographic operations

The TPM: What it *Can't* Do

- Appraise or enforce anything
  - PCRs are just repositories
  - No concept of what the "right" value is
  - The TPM can't stop a boot process or halt the system
- Measure anything
  - TPM is a passive entity
  - No continuous monitoring, no action at all without infrastructure
- TPM alone requires trust in the BIOS
  - The Core Root of Trust for Measurement (the earliest booting component that can measure) is in the BIOS
  - Other hardware extensions allow trustworthy measurements even if BIOS is compromised

An "Integrity Measured" Boot Process

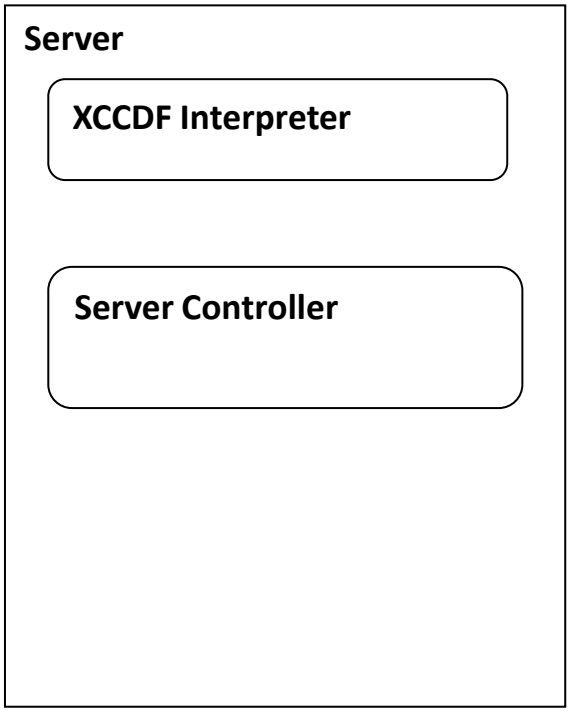


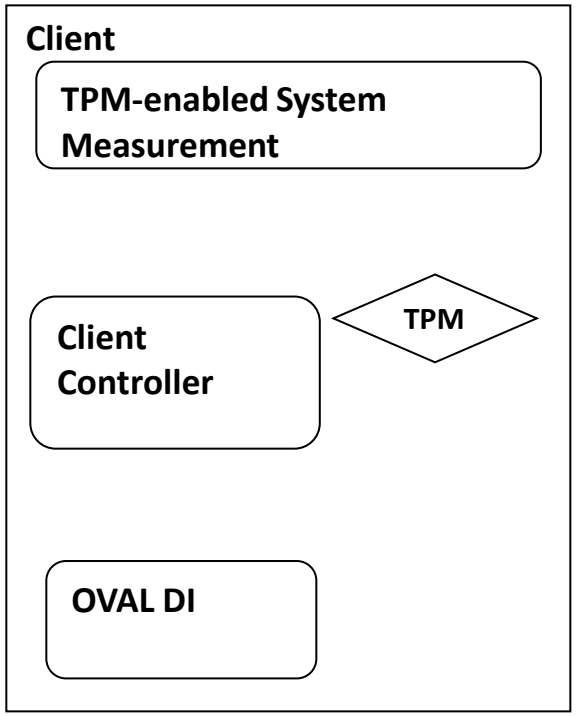
Apps	•Stored into higher-level PCRs •Measurement invoked by OS; timing of invocation can vary •Less standardization at this level
OS	•Measure and store (PCR-4) OS components, other TCB-relevant stuff •System runs normally!
OS Loader	•Measure OS Components •Append PCR-3 with measurement •Pass control to OS
BIOS	•Measure Option ROMs •Append PCR-1 with measurement--pass control, get it back •Measure OS Loader, append PCR-2, pass control
CRTM	•Core Root of Trust for Measurement, (e.g. BIOS block) •Measure itself and BIOS •Append PCR-0 with hash of measurement
Hardware	•Clear PCRs—not really configurable

OVAL + Attestation

- Add a new “TPM” component schema to OVAL Definitions and System Characteristics
  - Collect info about TPM (FIPS enabled? TSS version?)
  - Allows collection of a “TPM quote”
    - TPM quote is the measurements in the PCRs signed by a TPM identity key
- What this gives us
  - Expand OVAL coverage – allow policies to include TPM requirements
  - Attest to correct OVAL software stack
    - A TPM quote could include measurements of the OVAL execution stack (OVAL Interpreter, libraries, and other dependencies)
    - The OVAL-SC file would contain evidence of correct interpreter state
    - Result: Standard OVAL Results + evidence that the results are trustworthy

Demonstration Architecture





XCCDF Interp – Processes policies

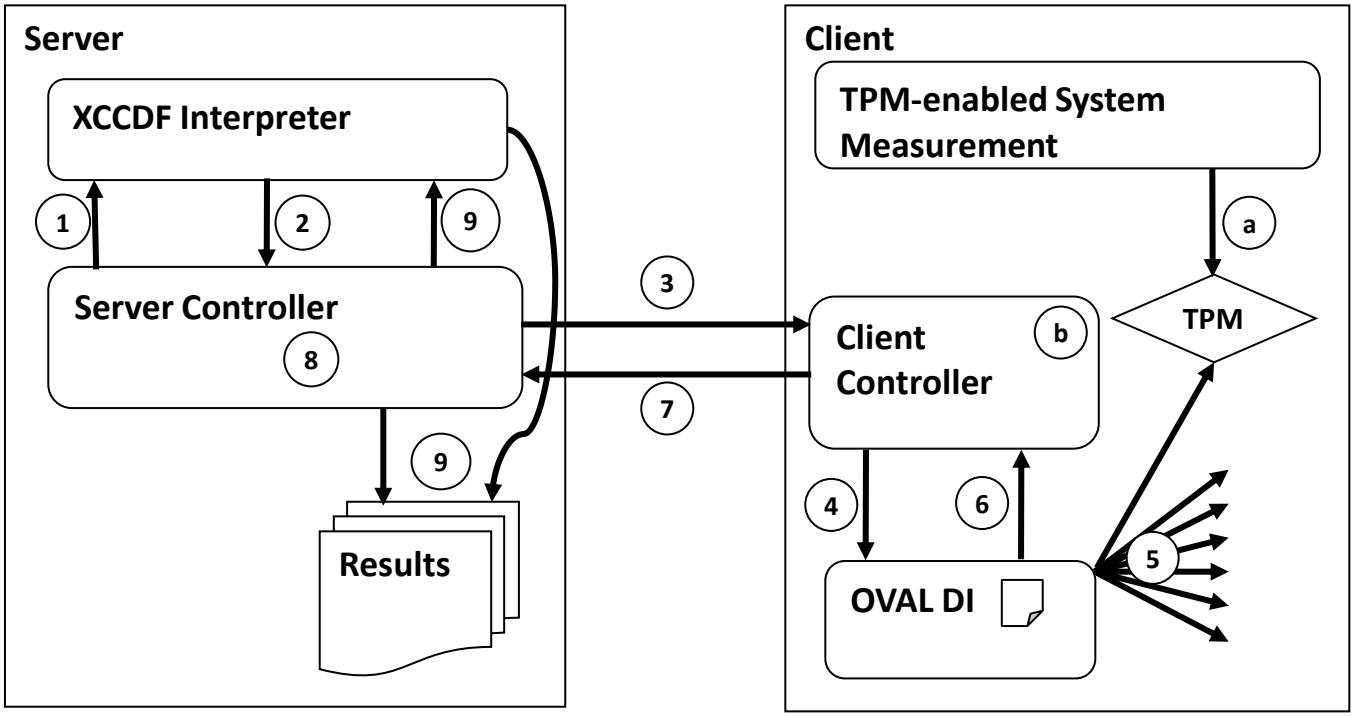
Server Controller – Orchestrates interactions between XCCDF Interp., and client

Measurement – Measures system, including OVAL DI

OVAL DI – Collect/measure findings

Client Controller – Orchestrate between server and local OVAL DI

Demonstration Architecture



1. Server Controller initiates assessment
2. XCCDF Interp. processes the guidance and collects OVAL Definitions
3. Definitions sent to the Client Controller…
4. … which passes them to the Client OVAL DI
5. OVAL DI collects system findings, including a TPM quote and creates Results and System Characteristics files
6. The files are sent to the Client Controller…
7. … and forwarded to the Server Controller
8. The Server controller verifies the integrity of the OVAL DI using the info in the SC file
9. If verified, the Server Controller shares the results with the XCCDF Interp. to get the XCCDF results and also stores raw results

Current Status

- Have created a draft OVAL TPM component schema for quotes
  - Will expand the component schema to support other TPM-related data collection
- Working to create a prototype OVAL-DI TPM probe
- Developing the other components to create the demonstration architecture

- When the proof-of-concept is complete, the schema will be published for additional community input

Questions?

MITRE