Carnegie Mellon CyLab

TrustVisor: Efficient TCB Reduction and Attestation

**Jonathan M. McCune,
Yanlin Li, Ning Qu, Zongwei Zhou,
Anupam Datta, Virgil Gligor, Adrian Perrig**

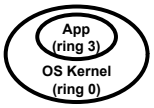
May 17, 2010

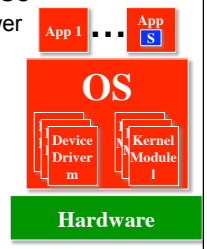
Motivating Example

- Conscientious web server admin / dev
- Wants to protect most critical data
  - SSL private key, password file, ACL, …
- Evaluates low-cost options
- Her best efforts rest on a house of cards…

Challenge: Reducing the Trusted Computing Base

- Today's OSes have too much power
- Total access to application data





- App may require little OS support
  - Self-contained computation 'S'
- Trusted computing base for S includes majority of: OS, drivers, and privileged applications!!!

What is S?

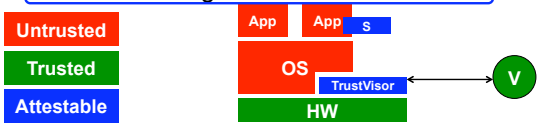
- Self-contained code in an application
- Data secrecy and integrity requirements
- General-purpose computing
- Some examples
  - Manages a private key for web server or CA
  - Manages Access Control List (ACL)
  - Is a compute client in distributed setting
  - Is similar to a Flicker session [McPaPeReIs2008]

Outline

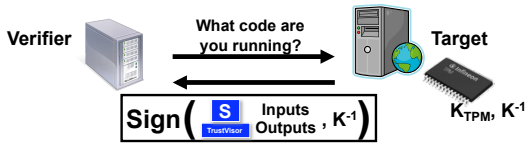
- Motivation (done)
- **High-Level Overview**
- Detailed Description
- Prototype: Apache + SSL
- Limitations
- Summary & Conclusions

Meet TrustVisor

- Tiny hypervisor for isolation of code S
  - No scheduling or Inter-Process Communication
- Efficient transitions between OS and S
- External verification of Output = S(Input)
- Protected storage for S



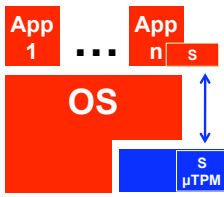
External Verification: Attestation



- Trust in attestation rooted in hardware TPM (Trusted Platform Module)
- SSL-enabled web server scenario:
  - Client can evaluate server before sending data
  - Enables more meaningful SSL server validation

Protected Storage

- Initially, S is "red" (untrusted)
- App can **register** S ➔ "blue" (attestable)
- TV enables "blue" code to protect data…



- Access-controlled by **identity** of S (hash)
- Enabled by TPM-like Sealed Storage
- "Micro-TPM" in software

Alternative Approaches

Approach \ Metric	TCB Size (LoC)	Protection granularity	Performance
Monolithic kernel	millions	–	best
Virtualization	millions	VM	good
Virtual TPM (vTPM)	millions	consistent code	good
Overshadow etc.	millions	process	good
Security / µ kernel	~100K	process	moderate
Flicker	<1K	fine	poor
TrustVisor	<10K	fine	good

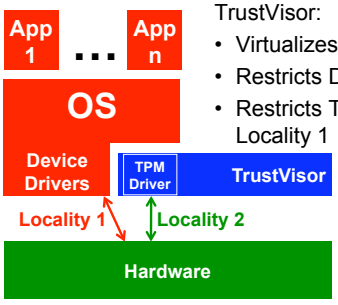
TrustVisor runtime TCB in lines of code:

- ~6500 C/ASM + ~2800 Headers
- Hypervisor + crypto

Outline

- Motivation (done)
- High-Level Overview (done)
- **Detailed Description**
- Prototype: Apache + SSL
- Limitations
- Summary & Conclusions

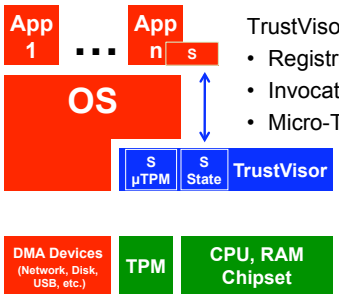
TrustVisor ↔ OS Architecture



TrustVisor:

- Virtualizes RAM, CPU
- Restricts DMA
- Restricts TPM to Locality 1

TrustVisor ↔ S Architecture



TrustVisor API

- Registration
- Invocation
- Micro-TPM

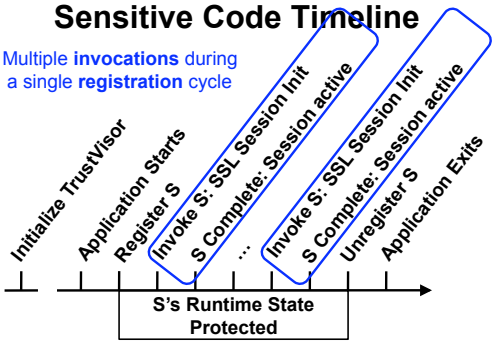
Identifying S to TrustVisor

- Applications identify S via **registration**
  - Page-level protection granularity
- Applications make "normal" function calls
  - TrustVisor detects switch to S via traps
- S runs with no access to legacy OS
  - One set of Inputs and Outputs per **invocation**

13

Sensitive Code Timeline

Multiple **invocations** during a single **registration** cycle



14

Micro-TPM Design

- Small subset of hardware TPM operations for:
  - Protected Storage + External Verification
- TPMs are optimized for cost, not speed
- TrustVisor implements critical-path TPM operations in software on main CPU
  - Extend, Seal, Unseal, Quote, GetRand
  - Reduces latency by **orders of magnitude**
- Trust in Micro-TPM still rooted in hardware TPM
- Infrequent TPM operations do not require virtualization

15

Outline

- Motivation (done)
- High-Level Overview (done)
- Detailed Description (done)
- **Prototype: Apache + SSL**
- Limitations
- Summary & Conclusions

16

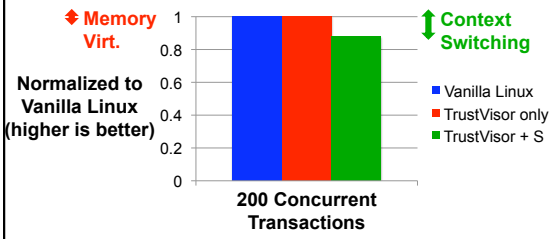
Example App: Apache + SSL

- Goal: Protect long-term private key K_{SSL}^{-1}
  - Cert revocation is abysmal in practice
- Desired properties
  - Malware, malicious admin unable to learn K_{SSL}^{-1}
  - Externally verifiable configuration
- Two sensitive code modules (S)
  - S1: Generate and seal the long-term key (**rare**)
  - S2: Unseal and use the key during SSL session establishment (**frequent**)

17

Apache + SSL Performance

- '**ab**' with 10,000 txns / trial, avg 10 trials



18

Outline

- Motivation (done)
- High-Level Overview (done)
- Detailed Description (done)
- Prototype: Apache + SSL (done)
- **Limitations**
- **Summary & Conclusions**

Limitations

- Design-level
  - Does not currently provide trusted path to user
  - Requires application awareness
- Prototype-level
  - No SMP support (currently single CPU)
  - Only protects K_{SSL}^{-1}
  - Executable code for S must be proactively paged into memory before registration
  - AMD-only

Summary & Conclusions

- Tiny hypervisor to support isolation
- Externally verifiable via attestation
- Frequent TPM operations in software
- Compelling performance argument
- Requires no OS changes
- Conclusions
  - Interesting point in the design space
  - Foundation for future trustworthy systems

Q & A

- Thank you!

- jonmccune@cmu.edu
- http://www.ece.cmu.edu/~jmmccune