Trustworthy Security Using Formal Methods

Mads Dam
KTH Royal Institute of Technology

ICES Annual Conference 2015
The Evolving Security Landscape

Increasing attack surfaces
• Devices and applications

Increasing aggregate value
• Information, USDs, impact

Increasing attack sophistication
• Stuxnet, Fanny, the Equation group

=>
Increasing demand for strong security solutions
Trustworthy Systems

Truly trustworthy systems need careful engineering:
• Rigorous security analyses
• Systems open to scrutiny
• Formal specs and proofs
• High level EAL certification
Trustworthy Systems

Truly trustworthy systems need careful engineering:
• Rigorous security analyses
• Systems open to scrutiny
• Formal specs and proofs
• High level EAL certification

We wanted that also 30 years ago!

What is new?
Trustworthy Systems

Truly trustworthy systems need careful engineering:
• Rigorous security analyses
• Systems open to scrutiny
• Formal specs and proofs
• High level EAL certification

We wanted that also 30 years ago!

What is new?

• 30 years research ...
  – Maturity of tools
  – Better understanding
  – Level of ambition

• Systems architecture
  – Virtualization
  – Modularization
It’s the Execution Platform, Stupid!

Processor hardware - a shared commodity
• Users, kernel, payment providers, plant owners, media owners, platform operators, system administrators, systems owners, ...
• All need private, tamperproof storage and cpu cycles

Example: Linux
• Monolithic – $10^7$ LoC kernel size
• Comprehensive analysis is out of the question

Key problem: Lack of isolation
Secure Virtualization

- Applications
  - OS
  - Android
  - Secure OS

Secure hypervisor

Processor

- Applications
  - OS
Secure Virtualization

- Applications
  - OS
  - Android
  - Secure OS

- Applications
  - OS

Secure hypervisor

Processor

ARM
Formal Hypervisor Verification ... eh?

Main components:

• Formal models of hardware platforms
  – Processor, peripherals, timers, interrupt controllers, MMU and system MMU units, ...
  – Highly non-trivial task

• Formal specifications of security objectives
  – Memory isolation
  – Security-relevant functionalities, inter-partition communication, etc.

• Formal proofs
  – Ties security objectives to hardware specs
  – Generated automatically or semi-automatically
Case Study: The Prosper Kernel

<table>
<thead>
<tr>
<th>Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>OS</td>
</tr>
<tr>
<td>Network driver</td>
</tr>
<tr>
<td>Reference monitor</td>
</tr>
</tbody>
</table>

Secure hypervisor

DMA unit

ARMv7-A processor
What Does This Give Us?

**Theorem.** Assume:

1. The processor model is correct.
2. The hypervisor is securely initialized.
3. The initial Linux image is signed.

Then:

Only signed code is ever executed
What Does This Give Us?

**Theorem.** Assume:
1. The processor model is correct.
2. The hypervisor is securely initialized.
3. The initial Linux image is signed.

Then:
- Only signed code is ever executed, **AND**
- Hypervisor is never tampered with
What Does This Give Us?

**Theorem.** Assume:
1. The processor model is correct.
2. The hypervisor is securely initialized.
3. The initial Linux image is signed.

Then:

- Only signed code is ever executed, AND
- Hypervisor is never tampered with, AND
- Monitor is never tampered with
What Does This Give Us?

**Theorem.** Assume:
1. The processor model is correct.
2. The hypervisor is securely initialized.
3. The initial Linux image is signed.

Then:

- Only signed code is ever executed, AND
- Hypervisor is never tampered with, AND
- Monitor is never tampered with, AND
- No memory page is ever simultaneously write and execute enabled, ...
What Does This Give Us?

**Theorem.** Assume:
1. The processor model is correct.
2. The hypervisor is securely initialized.
3. The initial Linux image is signed.

Then:

Only signed code is ever executed

This is what provable security is all about!
Where Are We Today?

HASPOC – High Assurance Products on COTS Platforms
• Vinnova spin-off project
• Partners: Ericsson, SICS, Sectra, Tutus, T2Data, @sec
• Objective: Secure virtualization for ARMv8-based platforms
• Challenges: Multicore, model complexity
• In progress...
Prospects

Formally verified security coming soon to a platform near you!