The Dual-EE Approach
The Dual-Execution-Environment Approach: Analysis and Comparative Evaluation

Mohamed Sabt\textsuperscript{1,2}

\textsuperscript{1}Orange Labs, France
\textsuperscript{2}Sorbonne Universités, UTC, France

IFIP SEC, May 2015
Outline

1 Introduction
Outline

1. Introduction
2. Background
   • Separation Kernel
   • Secure Execution Environment
Outline

1. Introduction

2. Background
   - Separation Kernel
   - Secure Execution Environment

3. Dual-Execution-Environment
   - Introduction
   - Core Properties
1. Introduction

2. Background
   - Separation Kernel
   - Secure Execution Environment

3. Dual-Execution-Environment
   - Introduction
   - Core Properties

4. Comparative Evaluation
Outline

1 Introduction

2 Background
   - Separation Kernel
   - Secure Execution Environment

3 Dual-Execution-Environment
   - Introduction
   - Core Properties

4 Comparative Evaluation

5 Conclusion
   - Summary
   - Perspectives
The Trust Problem

Description
How the execution of an application can be secured in the face of a compromised operating system?

Motivation
- The rise of services requiring trusted platforms with proved security;
- Operating systems of real-world systems are inherently insecure:
  1. Complexity and unsafe languages;
  2. Poor isolation.
The Multics Project – SOSP ’75

The Dual-EE Approach

Multics Supervisor

Security Kernel

entry gates

App #1

App #2
The Security Kernel
Where should it be placed?

1. monolithic operating system: Unix-like systems;
The Security Kernel
Where should it be placed?

1. monolithic operating system: Unix-like systems;
2. special hardware processor: AEGIS and XOMOS;
The Security Kernel
Where should it be placed?

1. monolithic operating system: Unix-like systems;
2. special hardware processor: AEGIS and XOMOS;
3. micro-kernel: SeL4;
The Security Kernel

Where should it be placed?

1. monolithic operating system: Unix-like systems;
2. special hardware processor: AEGIS and XOMOS;
3. micro-kernel: SeL4;
4. specialized OS: TLR.
The Security Kernel

Where should it be placed?

1. monolithic operating system: Unix-like systems;
2. special hardware processor: AEGIS and XOMOS;
3. micro-kernel: SeL4;
4. specialized OS: TLR.
Outline

1. Introduction

2. Background
   - Separation Kernel
   - Secure Execution Environment

3. Dual-Execution-Environment
   - Introduction
   - Core Properties

4. Comparative Evaluation

5. Conclusion
   - Summary
   - Perspectives
### Definition

**j'aurais mis Design à la place de definition**

### Design Purpose

To enable the coexistence of different systems requiring different levels of security on the same platform.

### Design Requirements

- **tamper proof**: it cannot be modified or disabled by rogue application;
- **always invoked**: all inter-partition communication request must go through it;
- **evaluable**: its correctness can be validated.
Overview of the SK Architecture

Background

Separation Kernel

M.Sabt et al. (Orange Labs)

The Dual-EE Approach

IFIP SEC, May 2015
Outline

1. Introduction

2. Background
   - Separation Kernel
   - Secure Execution Environment

3. Dual-Execution-Environment
   - Introduction
   - Core Properties

4. Comparative Evaluation

5. Conclusion
   - Summary
   - Perspectives
Definition

SEE is a processing environment that guarantees the following properties:

- **authenticity**: the code under execution should not have been changed;
- **integrity**: runtime states should not have been tampered with;
- **privacy**: code, data and runtime states should not have been observable by unauthorized applications.
Outline

1. Introduction
2. Background
   - Separation Kernel
   - Secure Execution Environment
3. Dual-Execution-Environment
   - Introduction
   - Core Properties
4. Comparative Evaluation
5. Conclusion
   - Summary
   - Perspectives
Overview

Dual-Execution-Environment

Introduction

M.Sabt et al. (Orange Labs)

The Dual-EE Approach

IFIP SEC, May 2015
The Dual-EE

The Dual-Execution-Environment is a security architecture where both a Separation Kernel and a Secure Execution Environment play the role of Security Kernel.
Theorem

Let $S$ be a system in which the security kernel is based on a separation kernel. If the requirements of all secure applications are equivalent, then all multi-execution-environment architecture can be reduced to a dual-execution-environment architecture.
Theorem

Let $S$ be a system in which the security kernel is based on a separation kernel. If the requirements of all secure applications are equivalent, then all multi-execution-environment architecture can be reduced to a dual-execution-environment architecture.

Sketch of proof

Let $S$ be multi-EE architecture. Without loss of generality, we suppose that $S$ is a system which contains 4 execution environments, 2 of which are non-secure and 2 are secure...
Outline

1. Introduction
2. Background
   - Separation Kernel
   - Secure Execution Environment
3. Dual-Execution-Environment
   - Introduction
   - Core Properties
4. Comparative Evaluation
5. Conclusion
   - Summary
   - Perspectives
Isolation Properties

1. **Data (spatial) separation.** Data within one partition cannot be read or modified by other partitions;

2. **Sanitization (temporal separation).** Shared resources cannot be used to leak information into other partitions;

3. **Control of information flow.** Communication between partitions cannot occur unless explicitly permitted;

4. **Fault isolation.** Security breach in one partition cannot spread to other partitions.
Functional Properties

1. Protected Execution. no interference caused by malicious software;
2. Sealed Storage. protecting the integrity, secrecy and freshness of data;
3. Protected Input/Output. protecting the integrity and secrecy of input/output data;
4. Attestation. authentication to remote trusted parties.
Ease-of-Deployment Properties

1. **Support of Legacy Systems.** required modifications for a system to run on a separation kernel;
2. **Cost.** extra silicon;
3. **Overhead.** separation kernel impact on performance;
4. **SEE Performance.** how fast complex operations are executed by SEE.
Dual-EE Technologies

- External hardware module: smart cards;
- Bare-metal hypervisor: KVM/ARM;
- Special processor extensions: ARM TrustZone.
ARM TrustZone

Comparative Evaluation

M. Sabt et al. (Orange Labs)
## Summary of Evaluation

<table>
<thead>
<tr>
<th>Comparison Category</th>
<th>Comparison Criteria</th>
<th>Smart Card</th>
<th>KVM</th>
<th>TrustZone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security Requirements</td>
<td>Protected Execution</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Sealed Storage</td>
<td>✓</td>
<td>×</td>
<td>✓*</td>
</tr>
<tr>
<td></td>
<td>Protected Input</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Protected Output</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Attestation</td>
<td>✓</td>
<td>×</td>
<td>✓*</td>
</tr>
<tr>
<td>Isolation Properties</td>
<td>Data Separation</td>
<td>HW</td>
<td>SW</td>
<td>HW</td>
</tr>
<tr>
<td></td>
<td>Information Flow Control</td>
<td>SW</td>
<td>SW</td>
<td>HW</td>
</tr>
<tr>
<td></td>
<td>Sanitization</td>
<td>HW</td>
<td>SW</td>
<td>HW/SW</td>
</tr>
<tr>
<td></td>
<td>Damage Limitation</td>
<td>HW</td>
<td>SW</td>
<td>HW</td>
</tr>
<tr>
<td>Deployability Criteria</td>
<td>Legacy Systems</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Low Overhead</td>
<td>✓</td>
<td>×</td>
<td>✓*</td>
</tr>
<tr>
<td></td>
<td>Low Cost</td>
<td>×</td>
<td>×</td>
<td>✓*</td>
</tr>
<tr>
<td></td>
<td>High Performance</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

✓: satisfies the criterion; ×: does not satisfy the criterion; ✓*: needs widely available additional hardware modules to satisfy the criterion; HW: satisfied by hardware module; SW: satisfied by software implementation.
Discussion

- Bare-metal hypervisors achieve the lowest score;
Discussion

- Bare-metal hypervisors achieve the lowest score;
- External hardware modules do not fit to certain kind of applications that need user interaction and high processing speed;
Bare-metal hypervisors achieve the lowest score;
External hardware modules do not fit to certain kind of applications that need user interaction and high processing speed;
TrustZone provides a balanced compromise.
Outline

1. Introduction
2. Background
   - Separation Kernel
   - Secure Execution Environment
3. Dual-Execution-Environment
   - Introduction
   - Core Properties
4. Comparative Evaluation
5. Conclusion
   - Summary
   - Perspectives
Summary

- The dual-EE is an interesting approach related to the trust problem;
- We provided a convenient abstract model to better represent the characteristics of the dual-EE approach;
- Our model was examined by revisiting the literature.
Outline

1. Introduction
2. Background
   - Separation Kernel
   - Secure Execution Environment
3. Dual-Execution-Environment
   - Introduction
   - Core Properties
4. Comparative Evaluation
5. Conclusion
   - Summary
   - Perspectives
Conclusion

Perspectives

- Include primitives defined in the MILS architecture to our core properties;
- Use the dual-EE approach to design a better Trusted Execution Environment (TEE).
Thank you for your attention!