Mind the Gap: Studying the Insecurity of Provably Secure Embedded Trusted Execution Architectures

Marton Bognar, Jo Van Bulck, Frank Piessens
imec-DistriNet, KU Leuven, Belgium
marton.bognar@kuleuven.be

Providing evidence for security

Inductive methods: A successful attack breaks the security claim, a failed attack supports but does not guarantee it.

Deductive methods: Can guarantee properties of a model, but the connection between the model and the implementation should be strong.

Goal: Narrowing the gap

- Fundamentally impossible to close [1]
- Narrowing the gap: case study approach
- Deductive + inductive methods
- Deriving guidelines from experimental evidence
- Impactful open-source systems with precise security claims, deductive proofs

Methodology

- Identify falsifiable assumptions
- Validate the implementation
- Identify missing attacker capabilities
- Check proofs

Three attack classes:
- Implementation-model mismatches
- Missing attacker capabilities
- Deductive errors

Case study systems: Sancus\textsubscript{V} [2], VRASED [3]

Sancus\textsubscript{V}:
- Secure interrupt handling
- Verilog hardware implementation
- Operational semantics, pen-and-paper proof

VRASED:
- Secure remote attestation
- Hybrid architecture: HW-Mod in Verilog + SW-Att based on HACL* 
- State machine model extracted from Verilog, mechanized proofs

Implementation-model mismatches

Successful attacks on the implementation that fail in the model

<table>
<thead>
<tr>
<th>Sancus\textsubscript{V}</th>
<th>VRASED</th>
</tr>
</thead>
<tbody>
<tr>
<td>V-01: Context-free instruction lengths</td>
<td></td>
</tr>
<tr>
<td>V-02: Maximum instruction length</td>
<td></td>
</tr>
<tr>
<td>V-03: Not secure from outside an ARM</td>
<td></td>
</tr>
<tr>
<td>V-04: Restarting the enclave from ISR</td>
<td></td>
</tr>
<tr>
<td>V-05: Number of enclaves</td>
<td></td>
</tr>
<tr>
<td>V-06: Accessing unprotected memory</td>
<td></td>
</tr>
<tr>
<td>V-07: Protected interrupt functionality</td>
<td></td>
</tr>
</tbody>
</table>

Guideline: Maintain a systematic connection between the implementation and the model

VRASED: model derived from implementation, fewer errors!

Results

Sancus\textsubscript{V}:
- Implementation-model mismatches: 7
- Missing attacker capabilities: 2
- Deductive errors: 0

VRASED:
- Implementation-model mismatches: 2
- Missing attacker capabilities: 5
- Deductive errors: 1

Resources:
- Repository: https://github.com/martonbognar/gap-attacks
- Including a CI pipeline for the attacks

References