TF-M Generic Threat Model

David Hu
2021 Jan
Agenda

- Threat Modeling basic concepts
- TF-M Threat Modeling
- TF-M generic Threat Model
Threat Modeling basic concepts
Threat Modeling

• How to make a system (more) secure?

System description → Threat Modeling fills the gap → Security to do
1. ???
2. ???
3. ???
Threat Modeling

• A core element of Secure Development Lifecycle (SDL)
  • Identify and prioritize potential security weaknesses
  • Identify mitigations against identified threats
arm

TF-M Threat Modeling
Methodology

TF-M specific process

- Identifying Target of Evaluation (TOE)
  - The entity on which Threat Modeling is performed
    - Varies in different usage scenarios
    - Identify Trust Boundary
- Asset identification
  - Resource of value protected by TOE
    - Varies in different TOEs/usage scenarios
- Data Flow Diagram (DFD)
  - Data accesses across Trust Boundaries
- Threats identification
  - Based on data flows
- Threats Prioritization
  - Depends on severity, difficulties, etc.
  - A standard qualitative scheme is preferred
- Mitigations
  - Whether system has mitigated against threats found
Structure

• To cover various TF-M usage scenarios, configurations and modules
  • A generic threat model
    – An overall analysis of TF-M implementation and identify general threats and mitigation.
    – Link
  • Specific threat models
    – Focus on specific usage scenarios, configurations and modules

Not a roadmap!

Generic Threat Model

To be defined

Threat Model of
dual-cpu

Threat Model of
IPC model

Threat Model of
XXX

...
TF-M generic Threat Model
Identifying TOE

• TOE
  • Secure Processing Environment (SPE)
    – TF-M
    – Interactions with other components running in SPE
  • How to protect SPE from NSPE?

• Trust boundary
  • Isolation between NSPE and SPE
Asset identification

• Hardware Root of Trust (RoT) data, e.g.
  • Hardware Unique Key (HUK)
  • Root authentication key
  • Other embedded root keys

• Software RoT data, e.g.
  • Secure Partition Manager (SPM) code and data
  • RoT service code and data
  • NSPE data stored in SPE
  • Data generated in SPE as requested by NSPE

• Availability of entire RoT service
• Secure logs, including event logs
**Data Flow Diagram**

- Data flows across trust boundary
  - Data flows inside TOE are out of scope

<table>
<thead>
<tr>
<th>Data flow</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DF1</td>
<td>TF-M initializes NS entry and activates NSPE.</td>
</tr>
<tr>
<td>DF2</td>
<td>NSPE requests TF-M RoT services.</td>
</tr>
<tr>
<td>DF3</td>
<td>Secure Partitions fetch input data from NS and write back output data to NS.</td>
</tr>
<tr>
<td>DF4</td>
<td>TF-M returns RoT service results to NSPE after NS request to RoT service is completed.</td>
</tr>
<tr>
<td>DF5</td>
<td>Non-secure interrupts preempt SPE execution in single Armv8-M core scenarios.</td>
</tr>
<tr>
<td>DF6</td>
<td>Secure interrupts preempt NSPE execution in single Armv8-M core scenarios.</td>
</tr>
</tbody>
</table>
Threat identification

- **STRIDE threat model**
  - Categorizing threats according to risk types

<table>
<thead>
<tr>
<th>Threats</th>
<th>Desired property</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>Spoofing</td>
<td>Authentication</td>
</tr>
<tr>
<td>T</td>
<td>Tampering</td>
<td>Integrity</td>
</tr>
<tr>
<td>R</td>
<td>Repudiation</td>
<td>Non-repudiation</td>
</tr>
<tr>
<td>I</td>
<td>Information disclosure</td>
<td>Confidentiality</td>
</tr>
<tr>
<td>D</td>
<td>Denial of Service</td>
<td>Availability</td>
</tr>
<tr>
<td>E</td>
<td>Elevation of Privilege</td>
<td>Authorization</td>
</tr>
</tbody>
</table>
Threat identification

- STRIDE-per-Interaction
  - Go through threat types on each data flow

<table>
<thead>
<tr>
<th></th>
<th>S</th>
<th>T</th>
<th>R</th>
<th>I</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>DF1</td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>DF2</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>DF3</td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DF4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>DF5</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>DF6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>
Threat prioritization

• Common Vulnerability Scoring System Version 3.1 (CVSS v3.1)
  • Base metric group
    – “represents the intrinsic qualities of a vulnerability that are constant over time and across user environments”
  • CVSS Calculator
### Document threats

- Threat template in generic threat model document

<table>
<thead>
<tr>
<th>Index</th>
<th>TFM-GENERIC-REQUEST-SERVICE-I-1</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>A malicious NS application may request to read data belonging to SPE.</td>
</tr>
<tr>
<td><strong>Justification</strong></td>
<td>A malicious NS application may request SPE RoT services to copy SPE data to NS memory. The SPE data belongs to components in SPE and must not be disclosed to NSPE, such as root keys.</td>
</tr>
<tr>
<td><strong>Category</strong></td>
<td>Information disclosure</td>
</tr>
<tr>
<td><strong>Mitigation</strong></td>
<td>TF-M executes memory access check to all the RoT service requests. If a request doesn’t have enough permission to access the target memory region, TF-M will refuse this request and assert a security error.</td>
</tr>
<tr>
<td><strong>CVSS Score</strong></td>
<td>7.1 (High)</td>
</tr>
<tr>
<td><strong>CVSS Vector String</strong></td>
<td>CVSS:3.1/AV:L/AC:L/PR:N/UI:N/S:C/H:I:N/A:N</td>
</tr>
</tbody>
</table>

A detailed explanation of the threat and how it may impact TF-M

STRIDE category

How does TF-M mitigate this threat?