An Agile Approach to Threat Modeling for Securing EdgeX Foundry

Tingyu Zeng,
DELL Technologies
Aug 2019
About Me

• Tingyu Zeng (tingyu.zeng@dell.com)

  • Based out of Raleigh, North Carolina

  • Security architect of IoT Platform Div. , Dell Technologies

    • Open source & commercial projects

    • Open source Edge Foundry project …

      • Major contributor & co-chairperson of security working group
Session Agenda

• EdgeX Foundry introduction
• Challenges in EdgeX
• Threat Modeling background
• Approaching Threat Modeling for EdgeX
Section One: EdgeX Foundry Introduction

• What is EdgeX Foundry
• The motivations
• High level architecture
• How EdgeX works
EdgeX Foundry

- An open source, vendor neutral Linux Foundation Apache 2 project
- A loosely coupled software framework for IoT edge computing
- Hardware and OS agnostic
Motivations

• An ecosystem of interoperable plug-and-play components
• A common open platform unifying edge computing
• Tools to create EdgeX-based IoT edge solutions quickly
• Collaborations benefits of the industry
How EdgeX Works

• A collection of micro services
• Single responsibility for each micro service
• REST APIs exposed
• Deployed via Docker and Docker Compose
Section Two: Security Challenges in EdgeX

• Challenges as an IoT project
• Challenges as an open source product
• A strategic and tractable approach to response challenges
Security Concerns in IoT

• Limited abilities – storage, processing capabilities etc.
• Authentication & authorization
• Secure communications
Security Challenges as an Open Source Project

• “As many reviewers as possible” – good looking but difficult to implement

• Updates and fixes not on a par with latest security practices

• Consistency between release cycle and switching hands
Threat Modeling to Address the Challenges

• Covering the threats from a big picture
• Facilitating both contributors & users understanding the security scope
• Providing security improvements progressively
• Predictable and reliable
Section Three: Threat Modeling Background

• Threat Modeling in a nutshell
• When & who
• General approach to Threat Modeling
  • Architecture Decomposing
  • CIA & STRIDE threats
  • Mapping STRIDE to components
  • Threat mitigation
Threat Model & Threat Modeling (OWASP)

- Threat Model
  - “A structured representation of all the information that affects the security of an application”

- Threat Modeling
  - A process for capturing, organizing, and analyzing all of this information
  - Enables decision-making about risks
  - Output: a prioritized list of security improvements to the concept, requirements, design, or implementation
When to Threat Modeling

• Product design stage: think about the problems early

• Code development stage/sprint: review features with security in mind

• Product milestone release: combined with roadmap
Who to Threat Modeling

• The person that understands the product
• Contributors of the project
• Not necessarily a security expert
Threat Modeling: Decomposing Architecture

- Functionalities of the module
- Data Flow Diagram (DFD)
- Trust boundaries
- Assets
CIA Triad & Security Properties

- Confidentiality, Integrity, Availability and non-repudiation
## Threat Modeling: STRIDE Model of Threat

<table>
<thead>
<tr>
<th>Threat</th>
<th>Property Violated</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spoofing</td>
<td>Authentication</td>
</tr>
<tr>
<td>Tampering</td>
<td>Integrity</td>
</tr>
<tr>
<td>Repudiation</td>
<td>Non-repudiation</td>
</tr>
<tr>
<td>Information Disclosure</td>
<td>Confidentiality</td>
</tr>
<tr>
<td>Denial of Service</td>
<td>Availability</td>
</tr>
<tr>
<td>Elevation of Privilege</td>
<td>Authorization</td>
</tr>
</tbody>
</table>
Threat Modeling: Mapping STRIDE against Components

• Components: defined by trust boundaries
• Assets to protect
• List of potential threats
  • Spoofing
  • Tampering
  • Repudiation
  • Information Disclosure
  • Denial of Service
  • Elevation of Privilege
Threat Modeling: Threat Mitigation

- Preventing threats in advance
- Adding features to reduce/remove threats
- Looking for mature and proved solutions
- Accepting the risks
- Transferring the risks
Section Four: Approaching Threat Modeling for EdgeX

• Architecture analysis
• Threats matrix
• Threats mitigation
# STRIDE Attacks on Assets

<table>
<thead>
<tr>
<th>Assets</th>
<th>Threat</th>
<th>Method</th>
<th>Severity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device Service REST APIs</td>
<td>Spoofing</td>
<td>The attacker sends data with faked identification of the IoT device</td>
<td>High</td>
</tr>
<tr>
<td>Device Service REST APIs</td>
<td>Information Disclosure</td>
<td>Man-in-the-middle attack</td>
<td>High</td>
</tr>
<tr>
<td>Device Service REST APIs</td>
<td>Denial of Service</td>
<td>Dos against API</td>
<td>Medium</td>
</tr>
<tr>
<td>Command Service admin REST APIs</td>
<td>Elevation of Privilege</td>
<td>The attacker pretends to be an administrator to send commands.</td>
<td>High</td>
</tr>
</tbody>
</table>
Risk Mitigation

- API-Gateway security pattern
  - Single entry point to reduce attack surface
  - Metering for DoS attack
  - Authentication & Authorization mechanism
Cloud, Enterprise, On-Prem...
Conclusions

• Understanding and evaluating the security threats and resolutions quickly

• A practical approach to expose threats in different stage of development

• Adaptable in different level of scope of the open source projects
References

• “Threat Modeling – Designing for Security” by Adam Shostack
• “Agile Application Security” by Michael Brunton-Spall; Rich Smith etc.
• “Open Source Security Assessment” by Yoav Aner & Carlos Cid
• OWASP Threat Modeling Cheat Sheet
• EdgeX Foundry Project Wiki: https://wiki.edgexfoundry.org/
Q & A
Thank You

Tingyu.zeng@dell.com