Safety and Security in the IoT
A Model-Based Perspective

Bruce Powel Douglass, Ph.D.
Chief Evangelist, Global Technology Ambassador
IBM Internet of Things (IoT)
bruce.douglass@us.ibm.com
Twitter: @IronmanBruce
Website: www-01.ibm.com/software/rational/leadership/thought/brucedouglass.html
IoT Enterprise Architecture

- Web Portal & Dashboards
- Analytics
- Device Cloud Infrastructure
- M2M Gateway & Smart Edge nodes
- Sensors & Actuator
- Other Devices
Dependability

Safety
freedom from harm

Rear view mirror

Windshield

Mechanical braking system

Seat belts

Cruise Control

Connected automobile braking

“On-Star”-like service

GPS

Key fob

Infotainment

Mechanical trunk latch

Mechanical gas tank cap

Interior lighting

Reliability
service availability

Security
resilience to attack
Basic Safety-Related Concepts

- **Safety** is ultimately an issue at the real-world device level but other levels in the architecture may introduce safety-relevant concerns that manifest at the device level.

- **Accident** is a loss of some kind, such as injury, death, or equipment damage.
  - AKA mishap or harm.

- **Risk** is a combination of the likelihood of an accident and its severity:
  \[ \text{risk} = p(a) \times s(a) \]

- **A Hazard** is a set of conditions and/or events that leads to an accident. That is, *hazards result in accidents*.
  - Hazards are predictable and therefore controllable.
  - A safety-relevant system contains two kinds of hazards:
    - Intrinsic hazards
      - Hazards due to the inherent job and operational environment of the system.
    - Technology hazards
      - Hazards due to the addition of specific technological solutions.

- A **safety control measure** is an action or mechanism to improve the safety of the system by either:
  - Reducing the severity
    - For example, an airbag.
  - Reducing the likelihood
    - For example, a redundant sensor.
Some Safety Control Measures

- Obviation
- Education
- Alarming (person-in-the-loop)
- Active correction
- Redundancy
  - Homogenous
  - Heterogeneous
- Interlock
- Safety equipment (goggles, gloves)
- Restrict access
- Labeling
- Fail-Safe state/mode
Fault Tree Analysis Profile in UML

- UML is a general modeling language and can be extended to model metadata beyond its standard usage, for example
  - UML Profile for Schedulability Performance and Time (SPT)
  - Model Analysis of Real-Time Systems (MARTE)
  - Systems Modeling Language (SysML)
  - UML Profile for DoDAF and MoDAF (UPDM)
- The FTA Profile has the following views
  - FTA diagram
  - Safety Analysis Diagram
  - Hazard table view
  - FMEA table view
  - Fault table view
  - Fault – Requirement Matrix view
  - Fault – Design Elements Matrix views
Example Fault Tree Analysis

Hazard

Required Condition

Safety Requirement

Resulting Condition

AND Gate

OR Gate

Basic Fault

Undeveloped Fault
Model-Based Safety Analysis with the FTA UML Profile

The system shall request retransmission of a message if the computed CRC does not match the transmitted CRC.
### How to build a Safety Analysis

- A **hazard** is a condition that leads to an accident or loss.
- A hazard is characterized by:
  - Likelihood (L)
  - Severity (S)
  - Risk = L * S

#### Identify Hazards

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Probability</th>
<th>Severity</th>
<th>Risk</th>
<th>Safety Integrity Level</th>
<th>Fault Tolerance Time</th>
<th>Fault Tolerance Time Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Failure to Capture Heart</td>
<td>This hazard means that the pulse amplitude or duration is inadequate to reliably induce a cardiac contraction.</td>
<td>0.06</td>
<td>10</td>
<td>0.6</td>
<td>C</td>
<td>5</td>
<td>minutes</td>
</tr>
<tr>
<td>Pacing Too Quickly</td>
<td>Pacing too quickly can result in pacing in the super vulnerable period, potentially leading to fibrillation.</td>
<td>0.001</td>
<td>10</td>
<td>0.01</td>
<td>C</td>
<td>100</td>
<td>milliseconds</td>
</tr>
<tr>
<td>Pacing Too Slowly</td>
<td>Pacing too slowly can lead to inadequate blood flow leading to unconsciousness or death.</td>
<td>0.01</td>
<td>10</td>
<td>0.1</td>
<td>C</td>
<td>5</td>
<td>minutes</td>
</tr>
<tr>
<td>Too much Energy Delivered</td>
<td>Too much energy delivered can result in early battery depletion or, in very rare circumstances, cardiac tissue damage.</td>
<td>0.05</td>
<td>3</td>
<td>0.15</td>
<td>C</td>
<td>1</td>
<td>years</td>
</tr>
</tbody>
</table>
How to build a Safety Analysis

Define the *hazard metadata* to define and understand the hazard, its severity, and its likelihood.
How to build a Safety Analysis

A **required condition** is a preconditional invariant or assumption.

A **fault** is a system non-conformance. It may be systematic (error) or random (failure).

A **resulting condition** is one that results from a combination of more basic events and conditions.

A **hazardous event** is an event that is known to pose a safety concern.

A **normal event** is an occurrence expected by or normal to the system and its operational context.
How to build a Safety Analysis

Identify Hazards

Describe Hazards

Identify Related Conditions

Describe Conditions

Create Causality Tree

Add Safety Measures

Characterize conditions, especially faults.

This information can be used to generate a Fault Means and Effect Analysis (FMEA)
How to build a Safety Analysis

The FTA shows the relation – using logical operators such as AND, OR and NOT – among faults, events, and conditions.

These result in resulting conditions that may be further logically combined to result in manifested hazards.
How to build a Safety Analysis

Safety measures reduce either
- The likelihood of a fault
- The severity of a fault

The measure works because for the hazard to manifest the original fault must occur AND the safety measure must also fail

These will be represented in
- Safety requirements
- Safety design elements
Basic Security Concepts

- **Security** is freedom from interference, intrusion, or theft
- A **security asset** is something you want to protect in a security context. An asset can be
  - Tangible (e.g. cash, gas pipeline)
  - Intangible (e.g. vendor reputation)
  - Informational (e.g. credit card number, personal information)
  - Service (e.g. deliver power, provide life support)
  - Resource (e.g. publically available data, physical hw (stuxnet))
- A **vulnerability** is a weakness in the security field around an asset or asset context
- A **threat** is a means by which an vulnerability may be exploited
- An **attack** is an instantiation of a threat
- An **attack chains** (AKA cyber killchain) is a series of attacks meant to systematically exploit a system
- **Authentication** is a means by which access to an asset is controlled, ensuring that the user has permissions to access the asset
- **Permission** refers to the kind of access permitted (e.g. create, delete, read, read/write)
- A **countermeasure** is a means by which a threat can be blunted. A countermeasure can be
  - Active – looks for, detects, and acts on attacks (ex. Motion detector with alarm, authentication check)
  - Passive – prevents access (lock, removing access port, data encryption)
UML Security Analysis Profile

- Purpose: to enable upfront analysis of assets, security needs, vulnerabilities, attacks, and countermeasures, including the ability to link to security-relevant requirements, architectures, and design patterns. This applies not only to software but to systems, both cyber- and physical.

- Note: This is a part of the Operational Threat and Risk Information Sharing and Federation Model submission to the Object Management Group standards organization.

- Tool: This is to be a UML profile so it can hook into any properly supported UML/SysML tool (shown here in IBM Rhapsody).
  - The profile is to support not only reasoning about security but also provide traceable links to requirements, architectures, designs, models, and code to make the “security case”.

- Views
  - Security Analysis Diagram
    - A logic causality diagram relating assets, conditions, events, attacks, vulnerabilities, countermeasures and security violations. Similar to a Fault Tree Analysis Diagram.
  - Asset Diagram
    - A structure diagram relating users, roles, assets, permissions, countermeasures (incl authorization), vulnerabilities, threats, and countermeasures. A specialized class/block diagram.
  - Attack Flow Diagram
    - A behavioral diagram showing normal (e.g. user) actions, attack actions, attack chains, and countermeasures. A specialized activity diagram.
  - Attack Scenario Diagram
    - An interaction diagram showing the interaction of elements in the unfolding of an attack in a specific scenario. A specialized sequence diagram.
  - Threat table
    - A tabular summary of threat-relevant metadata including threats, threat agents, attack mechanisms, and countermeasures.
Model-Based Security Threat Analysis

- Security Analysis Diagram* (SAD) is like a Fault Tree Analysis (FTA) but for security, rather than safety
  - It looks for the logical relation between assets, vulnerabilities, attacks, and security violations
  - Permits reasoning about security
    - What kind?
    - How much?
    - Risk assessments

* From the Security Analysis Profile for the UML
Attack flow diagram

Unmarked (e.g. user) Actions plus
Stereotyped actions
From Attack Chain:
- Reconnaissance
- Delivery
- Exploitation
- Installation
- Command & Control
- Actions On Objective
From Countermeasures
- Countermeasure
Summary Tables and Matrices

- These summaries are based on data captured across multiple diagrams
- Tables list elements and show associated metadata
  - Asset
  - Countermeasures
  - Roles
- Matrices show relations between types of security elements
  - Asset-Authentication
  - Asset-Vulnerability
  - Countermeasure-Vulnerability
  - Requirements – [Security Element] (Role, Asset, Countermeasure, Authentication …)
  - Role-Asset
IoT Enterprise Architecture Security Perspective

- Web Portal & Dashboards
- Analytics
- Device Cloud Infrastructure
- M2M Gateway & Smart Edge nodes
- Sensors & Actuator
- Other Devices

Attack Surfaces
Example Attack Surfaces at the Device Level

From Comprehensive Experimental Analyses of Automotive Attack Surfaces
Some Security Countermeasures

- Access Control
- Accounting
- Scanning
- Active Detection
- Authentication
- Recovery
- Boundary Control
- Backup
- Encryption
- Deterrence
- Isolation
- Nonrepudiation
- Policy Action
How to build a Threat Analysis

Assets are system features or properties, or features of the system’s operational context that have value and the system must protect.
How to build a Threat Analysis

Assets have properties such as
- ID
- Kind
- Availability
- Value
- Access restrictions
- Role accessibility
- Integrity
Assets and environments express vulnerabilities – means by which they can be attacked.
The **Attack Chain** (aka **Cyber Killchain**) is the means by which an attacker identifies and exploits vulnerabilities.

Countermeasures may interrupt any step in the attack chain.
How to build a Threat Analysis

A security analysis diagram (SAD) specifies the causal links among assets, vulnerabilities, attacks, and countermeasures.
Countermeasures are system features or properties, that protect assets.

Security requirements specify the needs and properties for countermeasures.

Design elements define the structure and behavior of countermeasures.
## Threat Analysis Summary Table

### Threat Analysis Table

<table>
<thead>
<tr>
<th>Asset</th>
<th>Vulnerability</th>
<th>Threat Vector</th>
<th>Asset Value</th>
<th>Likelihood of attack</th>
<th>Reproducability</th>
<th>Exploitability</th>
<th>Breadth</th>
<th>Discoverability</th>
<th>Threat Priority</th>
<th>Counter measure</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>10</td>
<td>1</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>6</td>
<td>1</td>
<td>1</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>7</td>
<td>1</td>
<td>1</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>8</td>
<td>1</td>
<td>1</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>9</td>
<td>1</td>
<td>1</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>10</td>
<td>1</td>
<td>1</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Asset value is the value of the asset to be protected (1=very low, 10=very high)*

*Likelihood is the probability of the attack (1=very low, 10=certain)*

*Reproducability refers to how easy it is to reproduce the attack (for example, for it depend on timing or other circumstances?) (1=hard, 10 = very easy)*

*Exploitability refers to how easy it is to launch the attack (1=very easy, 10=very hard)*

*Breadth is the a measure of the extent of the attack. How widespread is it or how many systems are affected (1=few, 10=very many)*

*Discoverability is how easy it is for outsiders to find out about and exploit the vulnerability (1=very easy, 10=very hard)*

*Threat Priority is the product of the above values and is used to prioritize the threats for countermeasures These are in the range of 1 - 10*