Model-based Testing and Analysis of the Cyber-Resiliency of Cyber-Physical Systems - The SCAPS Project*

Dr. Roshan K Thomas
The MITRE Corporation
rkthomas@mitre.org

* SCAPS: Secure Control Architectures for Cyber-Physical Systems
* US Patent: 10,262,143
SCAPS’ Domain is Cyber-Physical Systems (CPS)

- The integration of computing, communications and physical control
- Sensors, controllers and actuators
- Diverse sectors
  - Industrial control
  - Aviation
  - Automotive
  - Electric grid
  - Medical
  - Weapons systems
Problem: How to Systematically Assess Design Vulnerabilities in Cyber-Physical Systems (CPS)

- **Operational:**
  - How to assess vulnerabilities in CPS designs and architecture?
  - How to increase efficiency and accuracy of security analysis?

- **Scientific challenges:**
  - How to systematically and efficiently analyze the attack space for a given CPS system design?
  - How to analyze where safety-oriented and fault-tolerant designs are inadequate to withstand cyber attacks?
  - How to systematically analyze and measure risk?
  - How to mitigate such risks by converging on an optimal design?

- SCAPS is creating model-based security and simulation analysis technology
- NDIA Report of 2011 calls for accelerated adoption of model-based engineering
From Conventional to Model-based Systems Security Engineering (MBSSE)

Conventional Design

- Document-centric
- Text-based requirements are isolated from structural and behavioral information
- No formal semantics
- Manual inspection to measure integrity, completeness, quality and accuracy
- Ineffective in dealing with the complexity of large systems

Model-based Engineering

- Model-centric
- Modeling constructs and relationships defined & reused
- Formal semantics
- Relationships define traceability paths
- Programmatically automate measurements
- Effective in dealing with complexity

Model-based Security Engineering

- Capture security dependencies between diverse system aspects
- Capture dependencies between security and safety
- Security-related resiliency metrics
- Enables model in-the-loop and simulation in-the-loop testing
- Iterate to security-optimized design
Integrating Security Analysis with Model-based Engineering

**Effects and Fault Injection**
- Inject *cyber-induced* effects and faults into models to exercise them
- Mapping between system, cyber and control models
- Libraries of cyber and control effects

**Security Analysis**
- Analysis of cyber-induced faults by criticality
- Systematic elaboration of attack surface
- Security and resiliency metrics

**Security Mitigation**
- Architectural and behavioral primitives to mitigate attacks
- Cost-benefit tradeoff analysis among mitigation options
- Security-optimized design

[Diagram showing different stages of analysis and mitigation]
Technical Approach: Integrate Model-based Design with Security Analysis of CPS

- Focus is on the interface between Cyber and Control layers
- Attack effects on AADL elements are mapped to effects on simulation models
- Attacks are injected into simulation models to study physical impact
The SCAPS Security and Risk Analysis Workbench

- Web-based user interface with backend integration to simulation engines
- Imports Architecture Analysis and Design Language (AADL) and simulation models
- Injects attack effects into simulation models to observe impact at control level

!!!

Simulated CPS attacks

Risk analysis

Workbench Architecture

Cyber – (AADL)

Control – (Simulink)

Physical – (Simulink)

© 2021 THE MITRE CORPORATION. ALL RIGHTS RESERVED.
SCAPS Conceptual Model for Hazard and Vulnerability Analysis

- A hazard spans multiple faults across multiple fault trees
- Enablers are environmental and other conditions outside the system’s engineering boundary
- Events of interest are typically faults
Value Proposition and Transformational Impact

- Helps to analyze the security of control systems faster, cheaper and more thoroughly
  - Potential to reduce cost of security analysis by at least 50%
- Four user communities benefit from our tools
  - **System design and test engineers** can uncover flaws early in the design before production and deployment
  - **Forensic analysts** can do a post-mortem analysis to locate and replicate security flaws reported in the field
  - **Red teams** can use our tools for attacks analysis during live exercises
  - **Third party certifiers** can exercise a control system using our tool to ensure known vulnerabilities do not exist
- Assistance in deriving an optimal secure design
Medical Example - Simulating Cyber Attack on an Analgesic Pain Management System

- Patient can ask for more analgesic medicine as desired by pressing a button
- Control system monitors and stops the pump from overdosing
- A stop or heart attack warning is reported to nurses on their monitor
- Attack 1: Prevents stop command to pump
- Attacks 2 and 3 prevent stop and death warnings from reaching the nurses
Before and After Attack Comparison

Regular Operation

Compromised Operation

Button Pressed

Drug Level

Heart Rate

Blood Oxygen Content

Stop Warnings Issued

Heart Attack

Death Warning

No warning given
Use Case Example 2: Design and Testing of an Experimental Fighter Jet

- Before moving into the next phase of design, the fighter simulation is put through security testing

- Two possible attacks were identified for evaluation
  - Using the wireless update capability, the controller software could be altered to read its inputs from the incorrect ports (e.g., 3rd input)
  - If the controller software’s inputs such as pitch state or pilot commands could be made unavailable, the pilot would be unable to regain control
Fighter Jet Schemas in AADL and Simulink

Experimental Fighter AADL Schema

Experimental Fighter Simulink Schema
Attack Scenario 1 – Loss of Elevator Control
Attack Impact on Fighter Jet

Joy Stick

Vertical Velocity (mph)

Altitude (ft)

Pitch Rate (rad/sec)

Pitch (rad)

G-Forces on Pilot

Crash into ground

Redout
## Cyber-induced Faults at the Cyber (AADL) Layer

<table>
<thead>
<tr>
<th></th>
<th>Process</th>
<th>Thread</th>
<th>Data</th>
<th>Subprogram</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Confidentiality</strong></td>
<td>Unauthorized memory access</td>
<td>Unauthorized memory access</td>
<td>Unauthorized access</td>
<td></td>
</tr>
<tr>
<td><strong>Integrity</strong></td>
<td>Tampering with process bindings (memory, processor and connection)</td>
<td>Tampering with thread bindings and state</td>
<td>Tampering with data properties including spruce text, data size</td>
<td>Tampering with source text, memory bindings</td>
</tr>
<tr>
<td><strong>Availability</strong></td>
<td>Induce delays in file loading; Disrupt scheduling</td>
<td>Prevent or delay an execution phase of a thread</td>
<td>Prevent or delay access</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Processor</th>
<th>Memory</th>
<th>Bus</th>
<th>Device</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Confidentiality</strong></td>
<td></td>
<td>Unauthorized memory access</td>
<td>Unauthorized tapping into bus contents</td>
<td></td>
</tr>
<tr>
<td><strong>Integrity</strong></td>
<td>Tampering with processor bindings (e.g., memory)</td>
<td>Corrupting memory</td>
<td>Tampering with bus contents in transit</td>
<td>Tampering with device drivers and threads; Tampering with execution time-related properties; Tampering with execution platform bindings</td>
</tr>
<tr>
<td><strong>Availability</strong></td>
<td>Disrupting processor scheduling</td>
<td>Disrupting memory access</td>
<td>Disrupt transmission and data movement</td>
<td>Disrupting operation cycles such as sensing and actuations</td>
</tr>
</tbody>
</table>
Sample Cyber-to-Control Mappings

Fault type
- Delay Transmission
- Tamper with Processor Binding
- Prevent Execution of a Stage
- Disrupt Scheduling
- Re-route Instructions for Process
- Unexpected Firing of Program
- Memory Corrupt

Cyber hook
- Delay
- Drop
- Blind-Write
- Write

Control hook
- Unavailable - Delay
- Hold-Delay
- Write-Zero
- Unavailable
- Overwrite
- Gain
- Noise
Sample Workbench User Interface: Selecting Cyber Effects
Visualizations: Summary of Simulation Runs

Experiment Summary
Risk Analysis Enabled by Visualization

**Filters for different views**

**Access to trace tools**

**Configuration**

**Event of Interest (EOI)**

Stall Loss of pitch control

SCAPS Dashboard: Control Configuration, Event of Interest (EOI) - Stall Loss of pitch control.
Integration of Fault Trees and Attack Effects Generation

- Utilize fault models/trees (FTs)
  - Generate FTs from AADL Failure Annex
- Analyze which faults are cyber-inducible
- Generate attack trees from fault trees
- Generate attack scenarios
  - Each scenario is a different path in the attack tree
- Generate attack packs

Behavior Models

Components, data flows execution sequences

(1) Landing gear inoperative
(2) Landing gear inoperative
(3) Cyber-induced fault sub tree

(4) A1
A11, A12, A13
A111, A112

(5) Attack hook placement and firing
Feature-set Evolution of the SCAPS Workbench

Developed features
- Core arch, APIs & user interface
- Automation and assistance
- Analytics and visualization
- Fault tree and kill-chain support

Ongoing enhancements
- HW and system in the loop
- Intelligent attacking and mitigation

Future work
- Red team TTPs
- Red team emulation and attack planning
- Security-optimized design

Increasing scale and performance
Ongoing Research: Intelligent Attack Generation

- How to intelligently attack the system to expose vulnerabilities?
- How to demonstrate that safety designs fall short in mitigating cyber attacks?
- Exploit information from:
  - Basic architecture/design assumptions, dependencies and flaws
  - Look for structural and behavioral patterns
  - Fault tree structure
  - Likelihood and severity info
  - Safety design dependencies
  - Control functions
Future Evolution of SCAPS:
Integration of Attack Scenarios, Attacker TTPs and Planning Modules

- Operating environment
- Threat
- Cyber defense posture
- Mission context

System Model

Planning and Runtime Engine

Tactics to Effect Mapper

Vulnerabilities

Simulated CPS attacks

IC ATT&CK and other TTP signals

Effects selection and execution sequence

Experiment Workbench

Risk analysis

Vulnerabilities

Cyber and Control System Models

Actuators

Sensors

Feedback used to refine plans

Codify red team behavior and knowledge

Adversary emulation

CPS Attack Scenarios

ICS/SCADA ATT&CK and TTPs

Attack TTPs

IC ATT&CK and other TTP signals
Summary

- Working Prototype of model-based CPS security and risk analysis workbench
- Development of cyber and control attack affects libraries and their mapping
- Integration of fault trees
- Visualization of risks
- Research directions
  - Intelligent attack generation to expose cyber-to-safety dependencies
  - Understanding the human factor aspects when cyber attacks are involved
  - How to derive automated mitigation strategies?
  - Development of resiliency and risk analysis metrics and methodologies
BACKUP
Modeling the Cyber Layer with Architecture Analysis & Design Language (AADL)

- Used to describe the hardware and software architecture of a system
- Allows user to link software components to their hardware components
  - E.g., Process to a processor
  - E.g., Data to memory
- Failure Annex models faults

Sample AADL schema abstractions