System-Aware Cyber Security Architecture

Rick A. Jones

October, 2011
Research Topic Description

• **System-Aware Cyber Security Architecture**
  – Addresses supply chain and insider threats
  – Embedded into the system to be protected
  – Includes physical systems as well as information systems

• **Requires system engineering support tools for evaluating architectures factors**

• **To facilitate reusability requires establishment of candidate Design Pattern Templates and initiation of a design library**
  – Security Design
  – System Impact Analyses
 Incorporating Recognized Security Functions into an Integrated System-Aware Security Solution

- **Fault-Tolerance**
  - Diverse Implementations of Common Functions
  - Data Continuity Checking via Voting

- **Cyber Security**
  - Moving Target with Diversity
    - Physical Configuration Hopping
    - Virtual Configuration Hopping
  - Adversary-Sensitive System Reconstruction

- **Automatic Control Systems**
  - Data Continuity Checking via State Estimation
  - System Identification
    - Tactical Forensics
System-Aware Cyber Security Subsystem

Measurements

Measurement Analysis

Security Control Decisions

System Control Signaling

System to be Protected

Hopping & Restoral Control
System-Aware Security Analysis

Mission-Risk
Ranked System Functions

Selected set for hopping
(1)
(2)
(3)
(4)
\vdots
(N)

Delay in compromise detection

Mission Risk

Number of hopped functions

Rate of hopping

System Latency

ASRR 10/11 October 2011
System-Aware Security for Facility Defense
Facility Defense System to be Secured

• We consider a facility defense system consisting of:
  – Streaming sensors continuously monitoring discrete areas
  – Streaming Servers distributing sensor data, received over a wired network, to mobile users over a wireless broadcast network
  – Mobile users receiving alerts and streaming data regarding potential problems
Illustrative Architectural Diagram for Candidate Facility Defense System for System-Aware Security
Potential Cyber Attacks

• Replay attacks masking malicious activity initiated through
  – Sensor system
  – Streaming servers
  – User devices

• DoS attacks addressed through redundancy
  – Sensor system
  – Streaming servers
  – Operational procedures and redundancy regarding user devices
System-Aware Solution for Securing the Facility Defense System

- **Replay attack defense**
  - Diversely Redundant Streaming Sensors, with Voting (Data Continuity Checking)
  - Diversely Redundant, Virtually Hopped Streaming Servers
  - Diverse User Devices, with Rotating User Surveillance Assignments and Device Use
  - Mobile User based Data Continuity Checking

- **DoS defense**
  - Redundancy at the Sensor and Streaming server levels
  - Streaming servers / User feedback loops to enable redistribution of data and job responsibilities
Illustrative System-Aware Solution Architecture
Observable Regions / User Fidelity Impacts of 3 Stream Continuous Voting

![Graph showing the relationship between Max Possible # of Observable Regions and Stream Fidelity (Kbps). The graph compares 'No Voting/Single Stream' and 'Continuous 3 Stream Voting' scenarios.](image)
Observable Regions / User Fidelity Impacts of 3 Stream Continuous Voting

Loss in User Presentation Fidelity

Max Possible # of Observable Regions

Stream Fidelity (Kbps)

No Voting/Single Stream
Continuous 3 Stream Voting
Observable Regions / User Fidelity Impacts of 3 Stream Continuous Voting

Reduction in Maximum Observable Regions

- No Voting/Single Stream
- Continuous 3 Stream Voting
Duty Cycle Voting for Increasing the Possible Number of Observable Regions

- Concept – Use of time division for voting permits an increase in the number of possible surveillance points
  - User compares streams concurrently received from multiple diversely redundant servers to discover discontinuities
  - 3 parameters can be utilized to govern voting
    - Number of Observed Regions
    - Deemed acceptable Voting Interval for data continuity checking across all regions
    - Streaming period time allotted for continuity checking (Voting Time), which can be less than the Voting Interval
  - Given the Voting Time can be a subset of the Voting Interval, the use of time division can be utilized to manage information distribution over the broadcast network, interleaving multiple streams for voting users with single streams for other users who are not voting
Illustrative System-Aware Solution Architecture with Duty Cycle Voting
Illustrative System-Aware Solution Architecture with Duty Cycle Voting
Illustrative System-Aware Solution Architecture with Duty Cycle Voting
Duty Cycle Voting for Increasing the Possible Number of Observable Regions

User 1

User 2

User 3

Wireless Network

Column Heights = Data / Time Interval
Observable Regions / User Fidelity Impacts of 3 Stream Continuous Voting

![Graph showing the impact of stream fidelity on observable regions and user fidelity for different voting methods. The x-axis represents stream fidelity in Kbps, ranging from 0 to 500. The y-axis represents the maximum possible number of observable regions, ranging from 0 to 100. The graph compares four methods: No Voting/Single Stream, Continuous 3 Stream Voting, and Duty Cycle Voting. Each method is represented by a different line color and marker style. The graph shows a downward trend as stream fidelity increases, indicating a decrease in observable regions and user fidelity.]
Additional Collateral System Impacts

- Common Cause Failures are reduced
- MTBF increases in relationship to the individual diverse component reliabilities
- Development cost increases based on the cost to develop voting and duty cycle management components, as well as to resolve lower level technical issues that may arise
  - Synchronization needs
  - Software integration
  - Performance impact measurements and enhancement needs (e.g. CPU utilization, memory, and energy usage)
- One time and life cycle cost increase in relationship to the increased complexity
Scoring Framework
Need: Methodology for Evaluating Alternative Security Solutions for a Particular System

• A methodology is required in order to clarify reasoning and prioritizations regarding unavoidable cyber security vagaries:
  – Relationships between solutions and adversarial responses
  – Multidimensional contributions of individual security services to complex attributes, such as deterrence

• Scores can be derived in many different forms
  – Single scalar value where bigger is better
  – 2 scalar values: (1) security value added, (2) system-level disvalues
  – Multi-objective component scores providing more transparency
Metrics

• Attack phase-based security value factors:
  – Pre-Attack (Deterrence)
  – Trans-Attack (Defense)
  – Post-Attack (Restoration)

• Would include collateral system impact metrics for the security architecture:
  • Performance
  • Reliability, Safety
  • Complexity, Costs
## System-Aware Security System Scoring Matrix

<table>
<thead>
<tr>
<th>Relative Value Weights</th>
<th>( k_1 )</th>
<th>( k_2 )</th>
<th>( k_3 )</th>
<th>( k_4 )</th>
<th>( k_5 )</th>
<th>( k_6 )</th>
<th>( k_j )</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Value Factors</strong></td>
<td><strong>Deterrence</strong></td>
<td><strong>Real Time Defense</strong></td>
<td><strong>Restoration</strong></td>
<td><strong>Collateral System Impacts</strong></td>
<td><strong>Implementation Cost</strong></td>
<td><strong>Life Cycle Cost</strong></td>
<td><strong>Other</strong></td>
</tr>
<tr>
<td><strong>Security Services</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Diversity ( (s_1) )</td>
<td>( s_{11} )</td>
<td>( s_{12} )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>( s_{1j} )</td>
</tr>
<tr>
<td>Hopping ( (s_2) )</td>
<td>( s_{21} )</td>
<td>( s_{22} )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>( s_{2j} )</td>
</tr>
<tr>
<td>Data Continuity Checking ( (s_3) )</td>
<td>( s_{31} )</td>
<td>( s_{32} )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>( s_{3j} )</td>
</tr>
<tr>
<td>Tactical Forensics ( (s_4) )</td>
<td>( s_{41} )</td>
<td>( s_{42} )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>( s_{4j} )</td>
</tr>
<tr>
<td>Other ( (s_i) )</td>
<td>( s_{i1} )</td>
<td>( s_{i2} )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>( s_{ij} )</td>
</tr>
</tbody>
</table>
## System-Aware Security System Scoring Matrix

<table>
<thead>
<tr>
<th>Security Services</th>
<th>Value Factors</th>
<th>Deterrence</th>
<th>Real Time Defense</th>
<th>Restoration</th>
<th>Collateral System Impacts</th>
<th>Implementation Cost</th>
<th>Life Cycle Cost</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diversity (s₁)</td>
<td>s₁₁</td>
<td>s₁₂</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>s₁j</td>
</tr>
<tr>
<td>Hopping (s₂)</td>
<td>s₂₁</td>
<td>s₂₂</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>s₂j</td>
</tr>
<tr>
<td>Data Continuity Checking (s₃)</td>
<td>s₃₁</td>
<td>s₃₂</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>s₃j</td>
</tr>
<tr>
<td>Tactical Forensics (s₄)</td>
<td>s₄₁</td>
<td>s₄₂</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>s₄j</td>
</tr>
<tr>
<td>Other (sᵢ)</td>
<td>sᵢ₁</td>
<td>sᵢ₂</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>sᵢj</td>
</tr>
</tbody>
</table>

**RelaDve Value Weights**

<table>
<thead>
<tr>
<th>k₁</th>
<th>k₂</th>
<th>k₃</th>
<th>k₄</th>
<th>k₅</th>
<th>k₆</th>
<th>kⱼ</th>
</tr>
</thead>
</table>

**Value Factors**

\[ \sum_{j=1}^{p} k_j = 1 \]

- **sᵢⱼ** = Assurance Level of the \( i \)th service as related to the \( j \)th value factor

- **sᵢⱼ** = Quantized Assurance Level = 0...M

**Security Score**

\[ \sum_{j=1}^{p} \sum_{i=1}^{n} k_j s_{i,j} \]

**Max Possible Score** = \( n \times M \)
# Example Facility Defense Scoring Matrix

<table>
<thead>
<tr>
<th>Security Services</th>
<th>Value Factors</th>
<th>Deterrence</th>
<th>Real Time Defense</th>
<th>Restoration</th>
<th>Collateral System Impacts</th>
<th>Implementation Cost</th>
<th>Life Cycle Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diversity ($s_1$)</td>
<td></td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Hopping ($s_2$)</td>
<td></td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Data Continuity Checking ($s_3$)</td>
<td></td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>Tactical Forensics ($s_4$)</td>
<td></td>
<td>3</td>
<td>0</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Relative Value Weights</th>
<th>$K_1 = 0.30$</th>
<th>$K_2 = 0.20$</th>
<th>$k_3 = 0.10$</th>
<th>$K_4 = 0.20$</th>
<th>$K_5 = 0.05$</th>
<th>$K_6 = 0.15$</th>
</tr>
</thead>
</table>

Max Possible Score = 20  Facility Defense Score = 11.5

Strongest Area is Restoration  Weakest Area is Life Cycle Cost
On Going Exploration

• A practical methodology for determining Assurance Level Values
  • Methodology for addressing uncertainty in assigning Assurance Level Values

• Methodology for utilizing Relative Value Weights

• Tradeoffs between scoring simplicity and transparency of results
Structured Arguments for System Scoring

- Builds upon the legacy of work developed for safety and information assurance case evaluations
- Utilizes Goal Structuring Notation (GSN) for communicating arguments to support assigned scores in a repeatable and clear manner
- System-Aware security scoring arguments for a particular system architecture include:
  - Context supplied by the system owner and includes an available risk analysis for the system being protected and scoring guidelines
  - System supplier provides the list of security services to be applied and characterizes the purposes expected of security services that are deemed as most pertinent to reducing risk
    - Specific claims about value factors and the anticipated effects of security services on these factors
    - Explanations of how each security service is anticipated to impact specific value factor claims, including explicitly dividing each service into policy, process, and technology components with corresponding explanations of value
Simplified Diagrammatic Representation of a Structured Argument for Deterrence Scoring (1)

Architectural Deterrence Claim
Assigned suitable scores for deterrence

Context
Risk analysis and scoring guidelines

Service Selection Strategy
Decompose the Architecture to isolate, for the purposes of scoring, security services that address deterrence

Scoring Assignment Strategy
Utilize experts to score service claims with accompanying rationale

Data Continuity Service Claim
Improves deterrence

Diversity Service Claim

Forensics Service Claim

Hopping Service Claim

See later slide
Simplified Diagrammatic Representation of a Structured Argument for Deterrence Scoring (2)

Data Continuity Service Claim (1)
Exploitation design requires distributed exploit designers

Data Continuity Service Claim (2)
Exploitation design requires designers with deep systems knowledge

Data Continuity Service Claim (n)
Simplified Diagrammatic Representation of a Structured Argument for Deterrence Scoring (3)

Data Continuity Service Claim (1)
Exploitation design requires distributed exploit designers

Red Team Evidence Document

System Design Team Evidence Document

Intelligence Analysis Evidence Document

Data Continuity Service Claim
Improves deterrence
Simplified Diagrammatic Representation of a Structured Argument for Deterrence Scoring (4)

Data Continuity Service Claim
Improves deterrence

Data Continuity Service Claim (2)
Exploitation design requires designers with deep systems knowledge

System Design Team Evidence Document