Model-Based Decision Support for Systems-Aware Cybersecurity

SERC Research Meeting

December 2014

Peter A. Beling, University of Virginia
Carl Elks, Virginia Commonwealth University
Nicholas Bollweg, Georgia Tech Research Institute
Rick Jones, University of Virginia
Barry Horowitz, University of Virginia

UNIVERSITY of VIRGINIA ENGINEERING

Broad Objective

Reversing cyber security asymmetry from favoring our adversaries (small investment in straight forward cyber exploits upsetting major system capabilities), to favoring the US (small investments for protecting the most critical system functions using System Aware cyber security solutions that require very complex and high cost exploits to defeat)

Focus on Defense Against Exploits that Impact System Performance (e.g., Data Corruption, Functional Degradation, System Latencies)

Not Only the Network and Perimeter

- Too Many Penetrations
- Insider Attacks
- Supply Chain Attacks
- Need to Include:
  - Weapon Systems
  - C2 Systems
  - Sensor Systems
  - Logistics Systems
  - Computer Controlled Physical Plant Systems (Engines, Electrical Power, Rudder Control, etc.)
  - Etc.

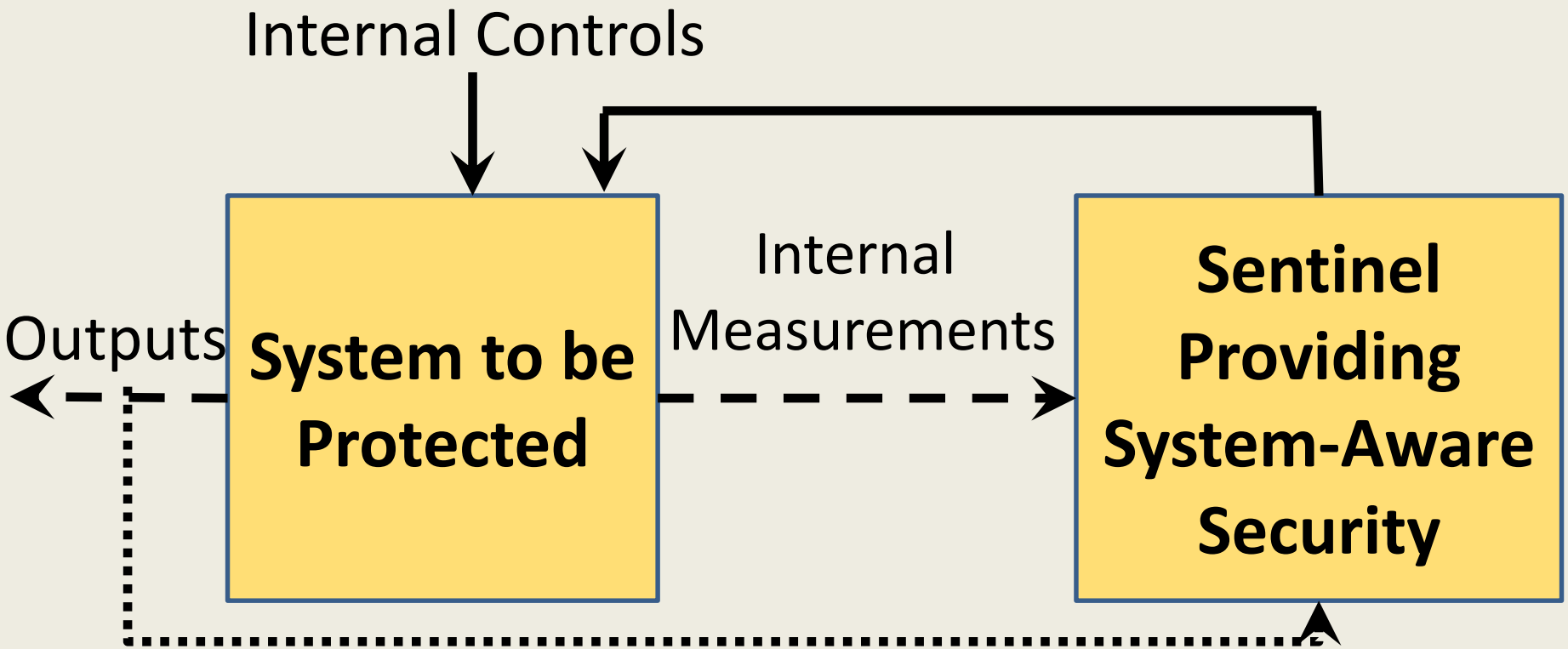
Mission-Based Security Strategy

- Need to make solution designs and decisions on a mission execution basis, rather than limited to a widget or subsystem basis
  - Attack occurs at Subsystem 1, symptoms appear at Subsystem 2
    - Meta data example
    - Attack initiation example
  - Detecting an attack through system consistency checks
    - Waypoint change example
    - Multiple sensor phenomonologies

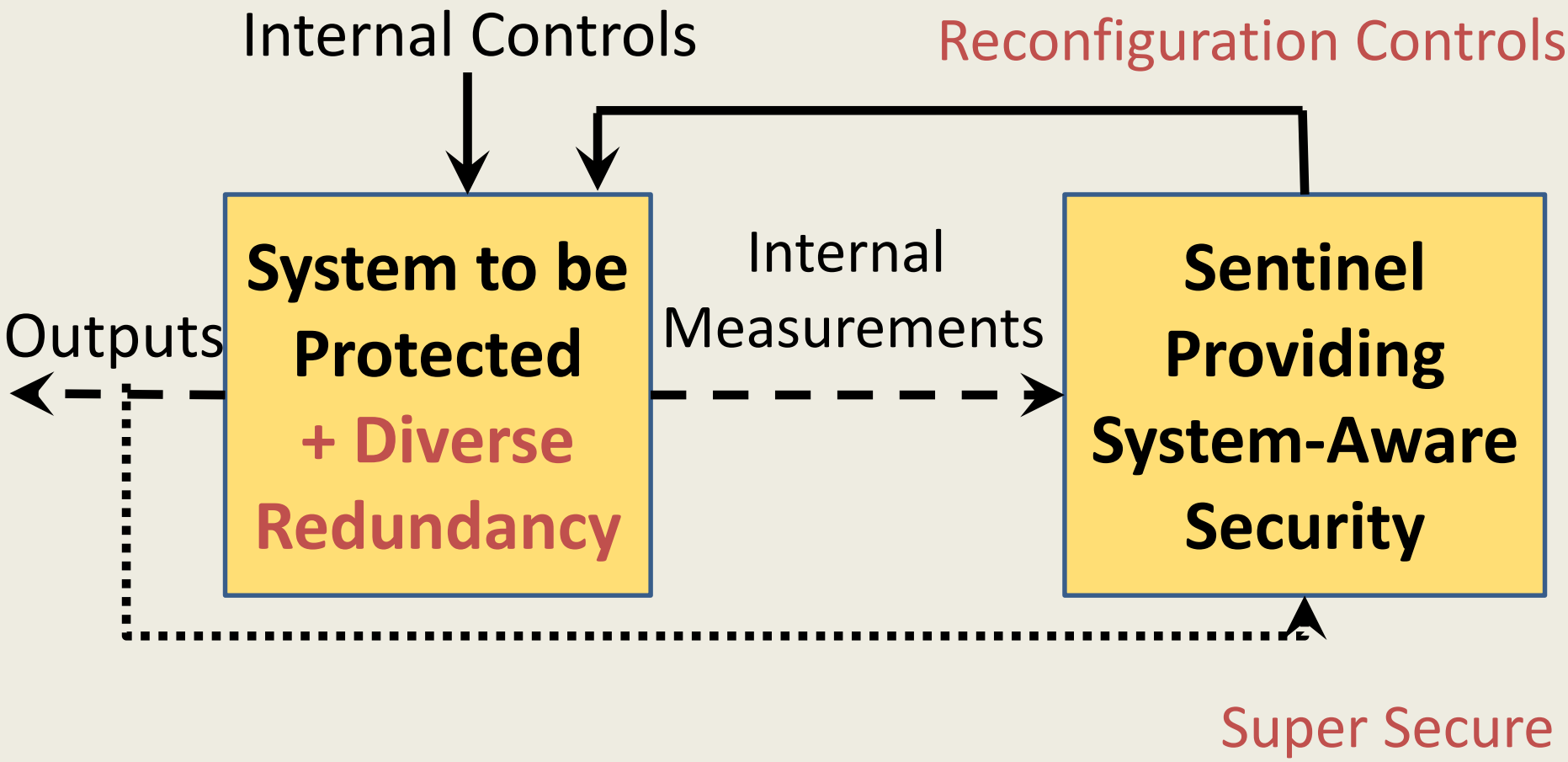
Security Integrated into Applications including Physical Systems

- Efforts are underway related to cybersecurity integrated into applications
- SERC-funded UVA effort on System Aware Cybersecurity
  - Recently conducted flight evaluation of protection for an autonomous surveillance system onboard a UAV
  - Defended on-aircraft attacks to prevent specific surveillance operations:
    - Waypoint change
    - Camera Pointing Control
    - GPS information for navigation or camera pointing
    - Image meta data changes

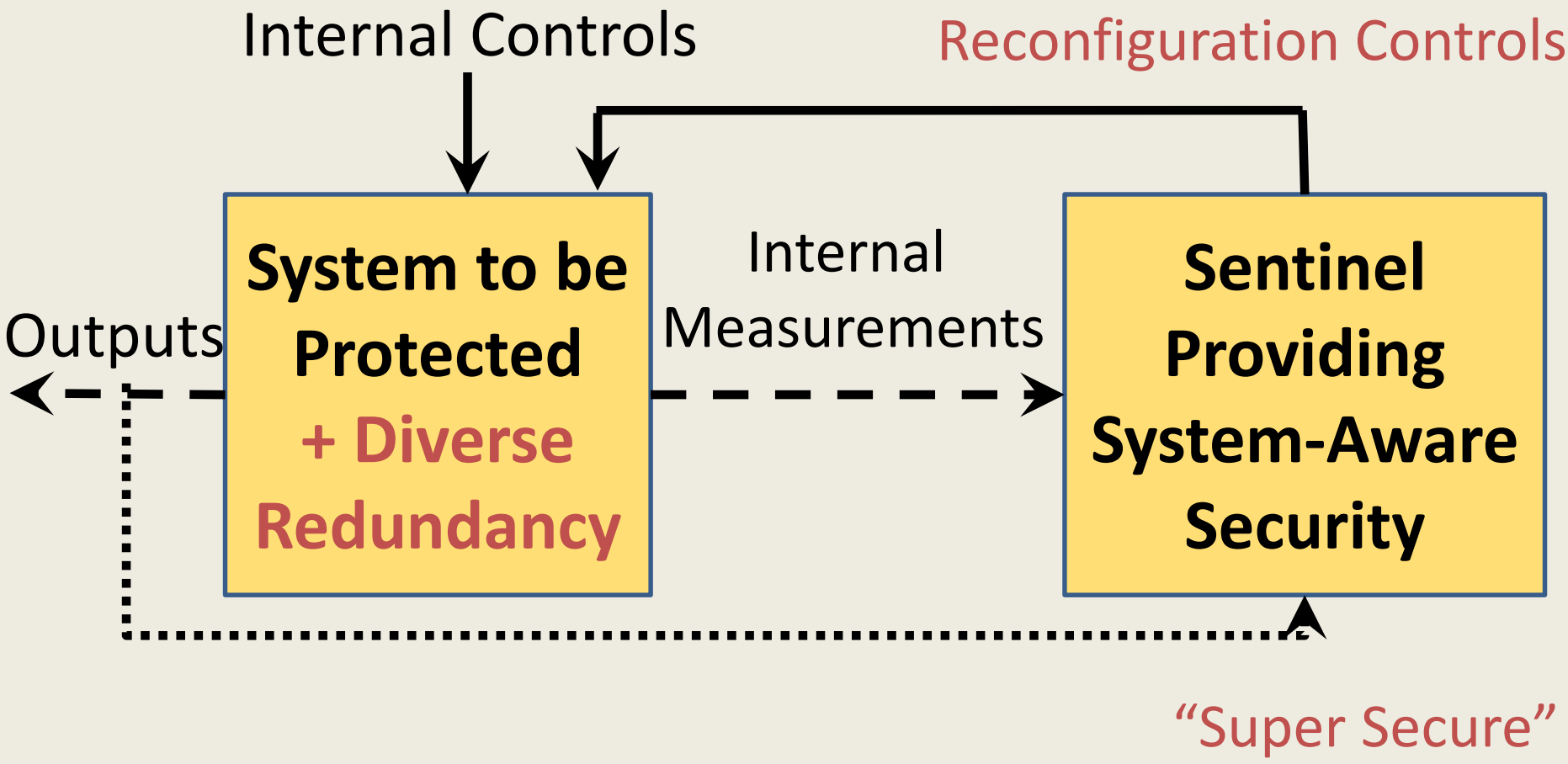
System Aware Cybersecurity
High Level Architectural Overview



System Aware Cybersceurity
High Level Architectural Overview



System Aware Cybersecurity
High Level Architectural Overview



System Aware Cyber Security Design Patterns

- Design Patterns combine design techniques from 3 communities
  - Cyber Security
  - Fault-Tolerant Systems
  - Automatic Control Systems (for physical systems)

A Set of Techniques Utilized in System Aware Cyber Security

Cyber Security

* Data Provenance
* Moving Target

  (Virtual Control for Hopping)
* Forensics

Fault-Tolerance

* Diverse Redundancy

  (DoS, Automated Restoral)
* Redundant Component Voting

  (Data Integrity, Restoral)

Automatic Control

* Physical Control for Configuration Hopping

  (Moving Target, Restoral)
* State Estimation Techniques

  (Data Integrity)
* System Identification

  (Data Integrity, Restoral)

A Set of Techniques Utilized in System-Aware Security

Cyber Security

* Data Provenance
* Moving Target

  (Virtual Control for Hopping)
* Forensics

Fault-Tolerance

* Diverse Redundancy

  (DoS, Automated Restoral)
* Redundant Component Voting

  (Data Integrity, Restoral)

Automatic Control

* Physical Control for Configuration Hopping

  (Moving Target, Restoral)
* State Estimation Techniques

  (Data Integrity)
* System Identification

  (Data Integrity, Restoral)

This combination of solutions requires adversaries to:

- Understand the details of how the targeted systems actually work

A Set of Techniques Utilized in System-Aware Security

Cyber Security	Fault-Tolerance	Automatic Control
* Data Provenance	* Diverse Redundancy (DoS, Automated Restoral)	* Physical Control for Configuration Hopping (Moving Target, Restoral)
* Moving Target (Virtual Control for Hopping)	* Redundant Component Voting (Data Integrity, Restoral)	* State Estimation Techniques (Data Integrity)
* Forensics		* System Identification (Data Integrity, Restoral)

This combination of solutions requires adversaries to:

- Understand the details of how the targeted systems actually work
- Develop synchronized, distributed exploits consistent with how the attacked system actually works

A Set of Techniques Utilized in System-Aware Security

Cyber Security

* Data Provenance
* Moving Target

  (Virtual Control for Hopping)
* Forensics

Fault-Tolerance

* Diverse Redundancy

  (DoS, Automated Restoral)
* Redundant Component Voting

  (Data Integrity, Restoral)

Automatic Control

* Physical Control for Configuration Hopping

  (Moving Target, Restoral)
* State Estimation Techniques

  (Data Integrity)
* System Identification

  (Data Integrity, Restoral)

This combination of solutions requires adversaries to:

- Understand the details of how the targeted systems actually work
- Develop synchronized, distributed exploits consistent with how the attacked system actually works
- Corrupt multiple supply chains

Design Patterns Being Prototyped

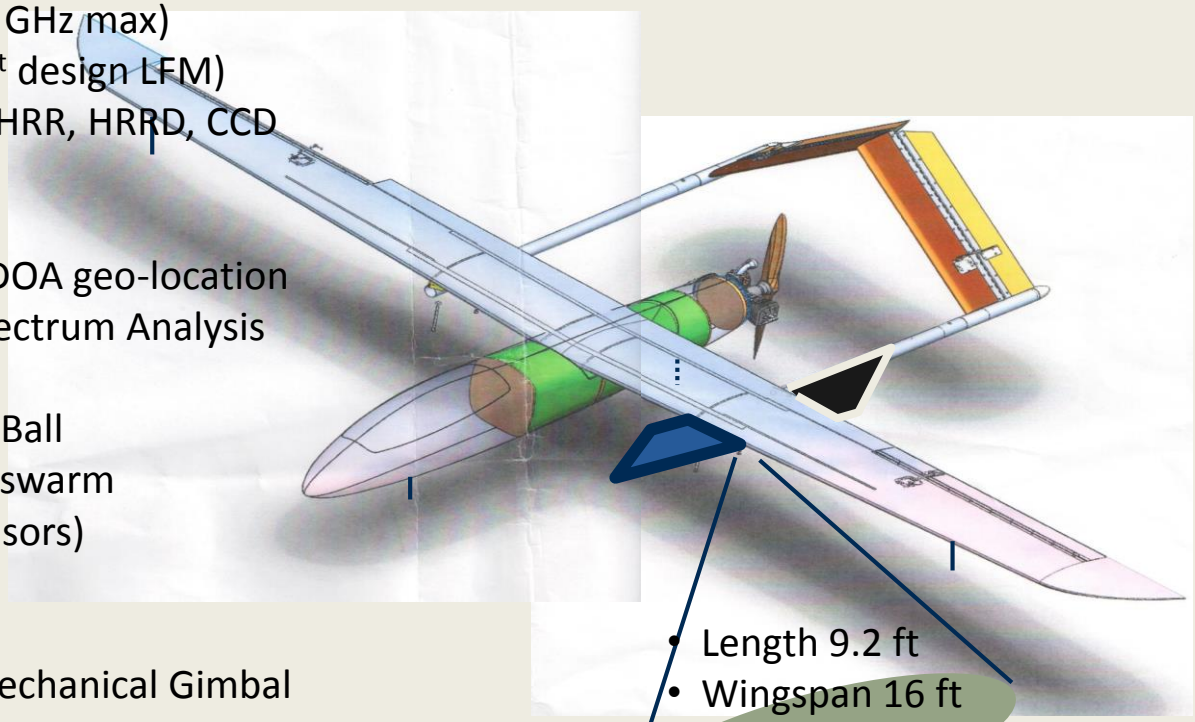
- **Diverse Redundancy** for post-attack restoration
- **Diverse Redundancy + Verifiable Voting** for trans-attack attack deflection
- **Physical Configuration Hopping** for moving target defense
- **Virtual Configuration Hopping** for moving target defense
- **Data Consistency Checking** for data integrity and operator display protection
- **Parameter Assurance** for parameter controlled SW functions
- **System Restoration** using diverse redundancy
- **Dynamic Configuration Management** to disable certain system control capabilities as a function of mission state

GAUSS– GTRI AIRBORNE UNMANNED SENSOR SYSTEM

Modified Griffon Aerospace
Outlaw (MQ-170) – Extended Range (ER)
Unmanned Aircraft System (UAS)

FOUR SENSOR OBJECTIVE BASELINE

- Multi-Channel Radar (8 channels)
  - ESA Antenna: 8 phase centers, each 4 x 4 elements
  - X-band, 600 MHz BW (design; 1 GHz max)
  - Arbitrary Waveform Capable (1st design LFM)
  - Acquisition Modes: DMTI, SAR, HRR, HRRD, CCD
- Multi-Channel SIGINT
  - Near 1 and 2 GHz Bands
  - Two orthogonal dipole pairs: TDOA geo-location
  - Ambient Complex-Baseband Spectrum Analysis
  - Signal Copy Selected Sub-Bands
- Gimbaled, Stabilized EO/IR Camera Ball
- High Precision GPS & INS (eventual swarm capable inter-UAV coherent RF sensors)



CAPABILITIES

- Electronic Scanning; No Antenna Mechanical Gimbal
- Multi-TB On-Board Data Recording
- Reconfigurable for Other Sensors: LIDAR, HSI, Chem-Bio
- Multi-Platform Distributed Sensor Experiments (eg, MIMO)
- Autonomous & Collaborative Multi-Platform Control
- Space for Future GPU/FPGA On-Board Processing

- Length 9.2 ft
- Wingspan 16 ft
- GTOW ~180 lbs
- Payload ~35-40 lbs
- Ceiling 14 kft
- Cruise speed 70 knts
- Endurance 9 hrs

Decision-Making Issues

- What to protect and why?
- Which combination of design patterns to employ in which mission subsystems?
- How to measure the benefits achieved from implementation choices?
- Process for decision making
  - Who to involve?
  - What information to provide for decision support?

Objectives and Design Philosophy

- Aims to provide structure process for navigating the decision space
  - End goal is support decision making by providing a standard and structured information set describing the pros and cons of the most competitive solutions.
  - Provide a structure to collect insight/information that otherwise could be overlooked and filter the set of possible solutions.
  - Increase the difficulty for the attacker at an acceptable impact to the operation of the system.
- Overall goal is to reverse the asymmetry present and erode the confidence of the attacker that he will be successful.
  - Two-way exchange of consequences.
  - Desire to identify areas where defensive team can make minimal changes to the system that will cause a maximum increase in difficulty or uncertainty for the attacker.
- Six steps, each defined with a goal, outcome, and responsible team(s).

Architecture Selection Teams

- Blue Team 1 – Identifies and prioritizes critical system functions
- Red Team – Identifies most desirable/lowest cost attacks (cost measured in complexity, risk of discovery, dollars required, etc.)
- Blue Team 2 – Identifies the set of security design patterns that address results of Blue/Red team prioritization analyses
- Green Team – Conducts cost/asymmetry analyses and selects desired solution that fits budget constraints

Architectural Selection Framework for System-aware Cyber Enhancement: Six Step Process

- **Step 1**: **Identify Relationships between sub-systems, functions and variables**- The process begins by identifying the most critical functions of the system and defining the variables and influence relationships within that portion of the system. **WHAT IS CRITICAL TO PROTECT?**

- **Step 2: Recognize the Possible Paths an Attacker Could Take to Exploit critical sub-systems.** Step two introduces the intelligent adversary viewpoint. In step two, the red team is tasked with assessing exploitable vulnerabilities, constructing an attack tree for critical sub-systems and functions. **WHAT ARE THE OPPORTUNITIES FOR AND CONSEQUENCES OF ATTACK?**

- **Step 3: Determine the Subset of Attack Actions Most Desirable to an Attacker.** Analysis on attack surfaces of critical sub-systems. What type of attacks are possible and by whom. The adversary model is characterized by *behavioral indicators* - technical ability, time, manpower, money, equipment, facilities, presence of an insider, and access to system design information. **WHAT IS EXPLOITABLE AND BY WHOM?**

- **Step 4: Identify appropriate defensive actions and their impacts on the attacker**. Select from existing portfolio of design patterns those that could increase the difficulty of the most desirable attack actions. PRE-SELECTION OF CYBER-DEFENSES.

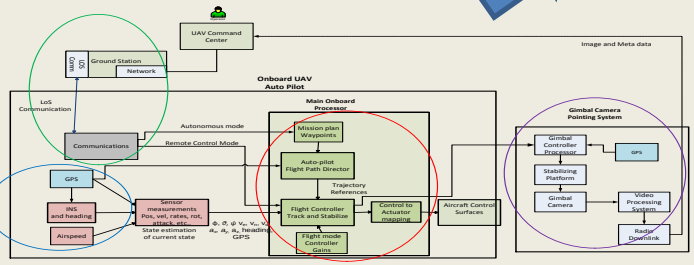
- **Step 5: Evaluate the impacts of the selected cyber-defensive actions on the system**. Evaluate Re-development, implementation cost, lifecycle costs, and collateral system impacts associated with each of cyber-defensive strategies. WHAT DOES THIS COST ME AND CAN I AFFORD IT?

- **Step 6: Weigh the Security Trade-offs to Determine Which Architectural Solutions Best Reverse the Asymmetry of a Potential Attack.** Assess the best candidate solution from step 5 for reversing attacker asymmetry that satisfies cost and collateral system constraints. EFFECTIVENESS OF BEST SOLUTIONS.

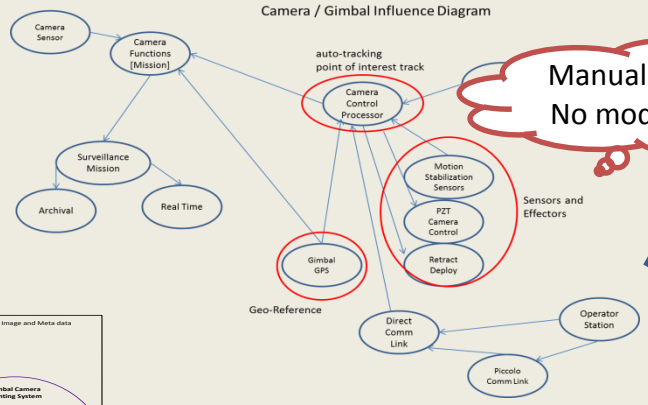
Architectural Selection Framework: V1.0

A Multi-Perspective Process on Cyber Security

UAV Systems



1: Identify Critical Assets

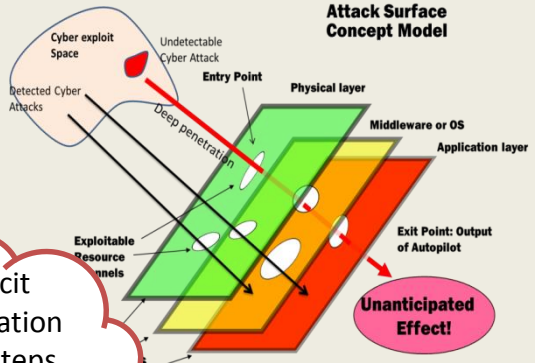


2: Relationships between Systems: Influence

Manual construction
No model semantics

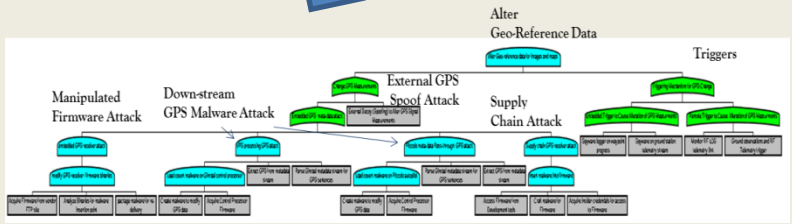
Tedious manual – no tool support

3: What are the Attack Opportunities



No explicit communication between steps 2/3 and attack trees

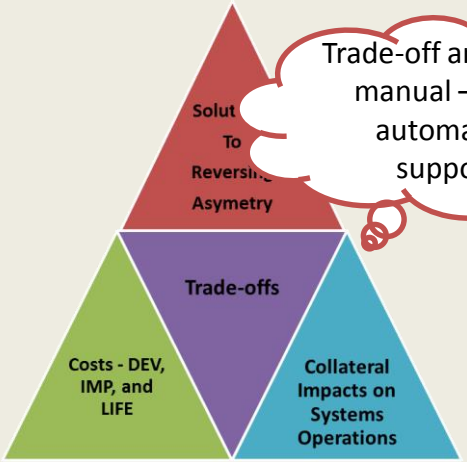
4: What is Exploitable and By Whom



5: Candidate Design Patterns and Impacts to Adversary's

- Diverse Redundancy
- Verifiable Voting
- Parameter assurance
- Hopping
- Etc…

Trade-off analysis - manual – little automated support

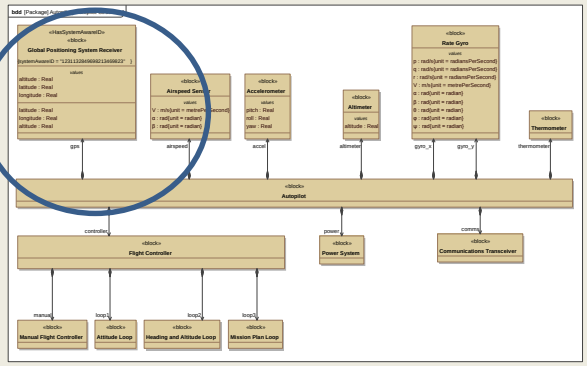


6: Best Architectures to Reverse Asymmetry

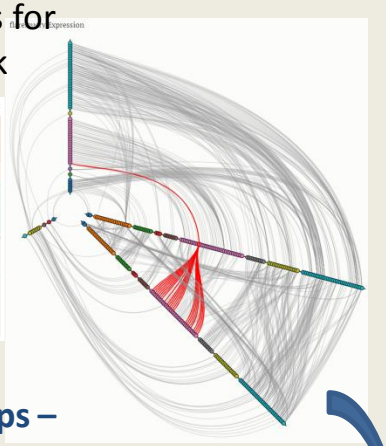
System Aware Cyber Security Framework: V2.0

Step 1: Identify Critical Assets

SysML models of UAV
(High fidelity Model Semantics)



Step 2: What are opportunities for and consequences of an attack

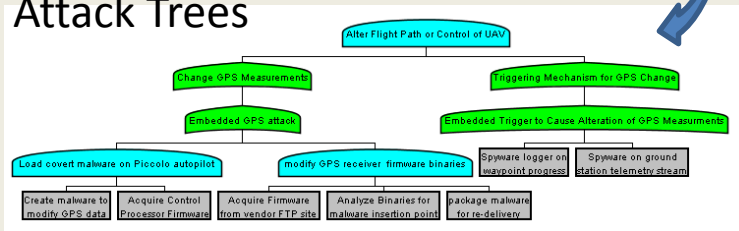


Visualization of System Relationships – Better Coverage of Attack Surfaces

Explicit information exchange- Information from SysML models helps create Attack Trees closer to reality

Step 3: What is exploitable and by whom

Attack Trees



Output:
- Ease of Attack
- Capabilistic Propensity
- Relative Risk

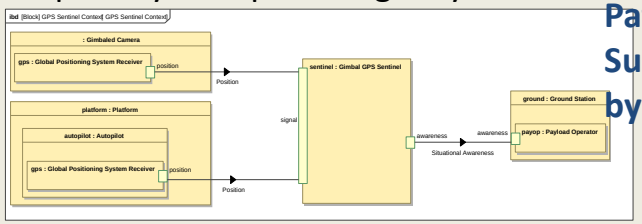
Resources for Attack

Attack Descriptions

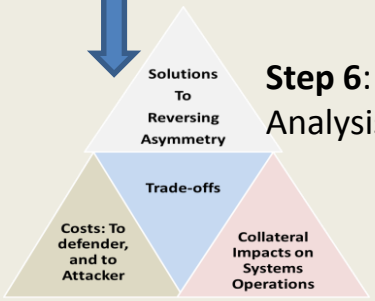
Agent Profiles

Step 4 and 5: Select/Evaluate Best Design Patterns to effect Adversary's capability to exploit Target System

Evaluation of Design Patterns Now Supported by Functional Models



Step 6: Cost Benefit Analysis



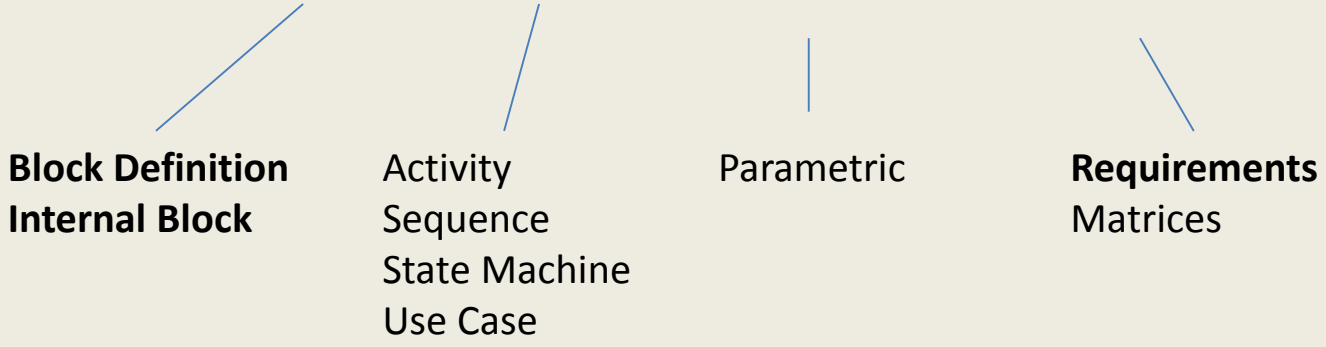
Decision making now aided with Easy to use Data Analysis/Visualization Tools

Modeling Tools for Accuracy at Scale

- **Systems Models** to capture the relationships between functional system entities and to recognize patterns (data, dependence, control) within the system.
  - Be able to represent the system attack surface (danger of under modeling) .
  - Represent the initial system "as-is" with minimal defense and again with possible security solutions implemented.
  - Value in showing solutions integrated into the holistic system for context.
  - Used to model an understanding of the complexity added to an attack by particular defenses.
  - Initial approach used influence diagrams. Currently developing a suite of tools in SysML.
- **Attack Trees** to identify possible paths an attacker could take to exploit the system.
  - Uses assessments of the attack actions and the attackers' capabilities to determine the subset of most preferable actions.

OMG SysML: The Systems Modeling Language

- What is SysML?
  - A standard **data model** and **visual syntax** for describing systems
  - Maintained by the **Object Modeling Group** standards body
- What does it do?
  - Show a system's **structure**, **behavior**, **parametrics**, and **requirements**



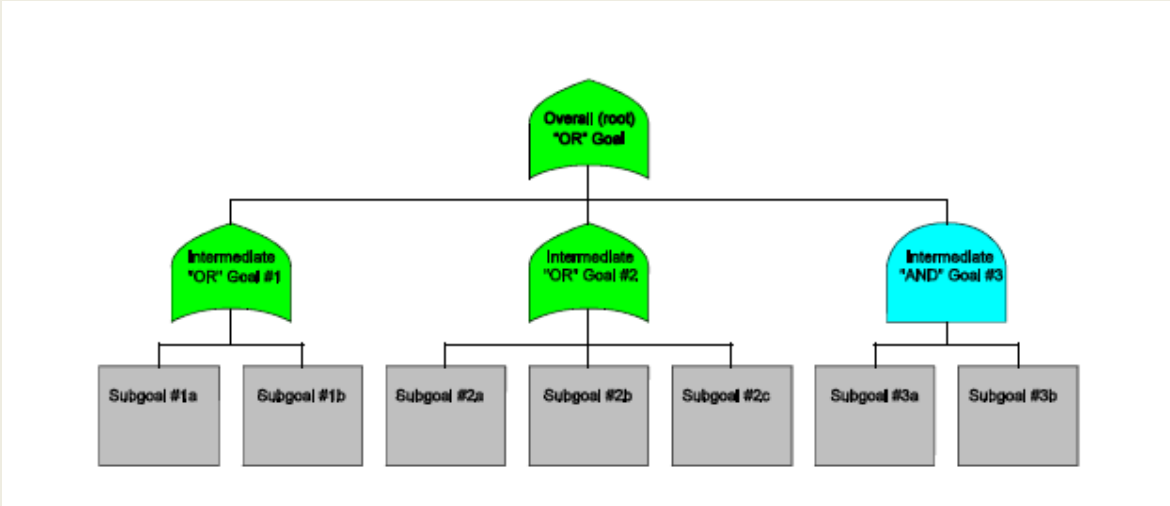
- Why use a dedicated SysML tool instead of…
  - PowerPoint? Object consistency, traceability
  - Raw Code? Validation, reusability
  - Raw Database? Visualization

Attack Trees

- **Attack trees** are similar to fault trees in reliability engineering – a top down, deductive failure analysis in which an undesired state of a system is analyzed using Boolean logic to combine a series of lower-level events.
- Been around for about 15 years or so publically.
- Attacks are modeled through the use of a graphical, mathematical, decision tree structure called an *attack tree.*
- Attack trees are constructed from the point of view of the adversary (or least we try to think like an attacker).

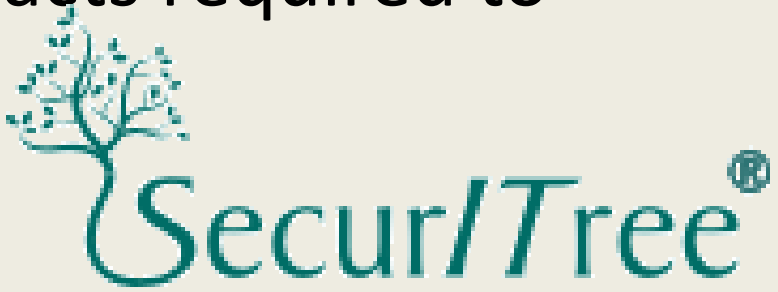
Attack Tree Basics

- In an attack tree model, the topmost (*root*) node represents an objective that would be of benefit to one or more threat agents.
- *Leaf nodes* represent activities that can be performed by adversaries.
- *AND/OR* nodes represent the states achieved as a result of these leaf activities.



Amenaza **SecureITree**

- Attack tree-based threat risk analysis: quantitative and qualitative analysis
- Creates a list of every combination of leaf node events that forms a path leading to the root goal. Associated with each path, or attack scenario, is the set of resources, attacker benefits and victim impacts required to traverse the path.



SysML and Attack Trees – Natural Synergism

- **Exploration of Potential Vulnerabilities** - SysML provides the rich modeling semantics to capture requirements, specifications, relationships and behaviors essential for capturing real system behaviors – enables dependable exploration.
- **Attack trees** – Provides a formal model semantics to explore *exploitation of vulnerabilities* – Attack trees do not find vulnerabilities.
- **The Synergism** - The SysML/Attack Tree connection is a "provides/requires" relationship. SysML "provides" the means to identify potential vulnerabilities, while attack trees "require" the existence of vulnerabilities to craft an exploit.
- **Cyber Defense** - Both have the ability to model countermeasures to vulnerabilities and exploits.
  - Finding and assessing solutions to a cyber-security gap

Attack Tree Creation Process Without SysML Guidance

Step 1: Lots of knowledge acquisition – Scavenge by hook or crook.

Step 2: Establish my end goal (e.g. alter fight path of UAV)

Step 3: Where might be vulnerabilities that I can exploit and depend on…(lots of time to this step).

Step 4: What might be the most straightforward way to exploit a vulnerability

Step 5: Is it feasible? More manual analysis...If not, start over…

Step 6: What's my attackers capability, resources, willingness

Step 6: Confirm vulnerability. May require outside sources to do so…

Step 7: Have experts critique the attack

Step 8: Ok, I am ready to build the attack tree

This is necessary….can't get around this…

This is the most time consuming step…
It's hard to reason and visualize the system interactions without some aid..

Again, to reason about an exploit requires
more understanding of system dependencies, use cases, system interactions, etc…

Attack Trees Creation Guided By SysML: A Better Attack Tree

Step 1: Lots of knowledge acquisition – Scavenge by hook or crook.

Step 2: Establish my end goal (e.g. alter fight path of UAV)

Step 3: Build SysML "living" model of the system

Step 4: Query the model to gain information and knowledge about things like:

- What are devices/functions are communicating with each other.
- What devices are dependent on certain functions or services (internal and external)
- How does mis-information or mis-use propagate in the system

Step 5: Narrow the search to small set of components, pick locations where an attack has the most impact

Step 6: Confirm vulnerability of locations. May require outside resources to do so…

Step 7: Gather Knowledge on threat agent and build profile

Step 8: Ok, I am ready to build the attack tree

Still necessary….can't get around this…

We capture knowledge and embed into a model of the system – Grows with knowledge, lifecycle, and assumptions…

We use this knowledge to inform us about the system behavior under normal, abnormal, and misused conditions

We focus on high value devices, functions to achieve our end goal…

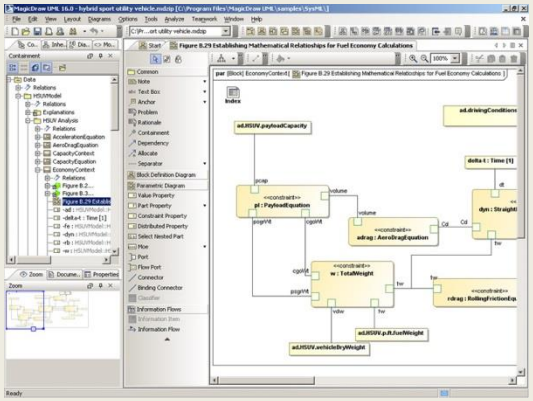
RESULT IS A BETTER UNDERSTANDING OF HOW AND WHAT TO DEFEND

Architectural Assessment Workbench Concept

Model Creation Input
- Specs (what it does)
- Requirements (what is suppose to do)
- User domain (how people use it)
- Functional
- Use Cases
- Mission Context

Capture system to system interactions, Relationships with respect to different users and threat agents

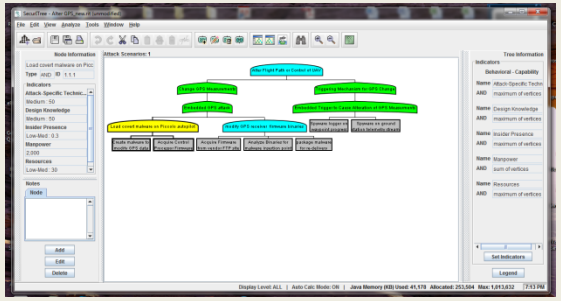
SysML *MagicDraw*



Model Creation Input
- Data on vulnerabilities
- Path analysis
- Sequences
- Component UUID

ATML
Attack Tree Markup Language

Attack Tree *SecureITree*



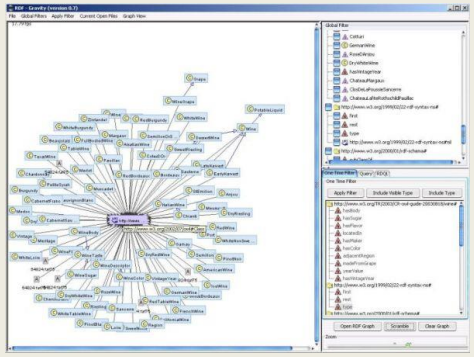
Trade-off/Cost benefit Analysis
- Cost of Attack to Attacker
- Cost of Defense
- Collateral Costs
- Lifecycle Costs

XMI
Extensible Model Interchange

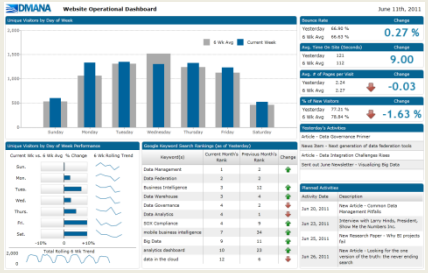
ATML

CSV
Comma Separated Values

Knowledge Graph *RDF*



Visualization *IPython*

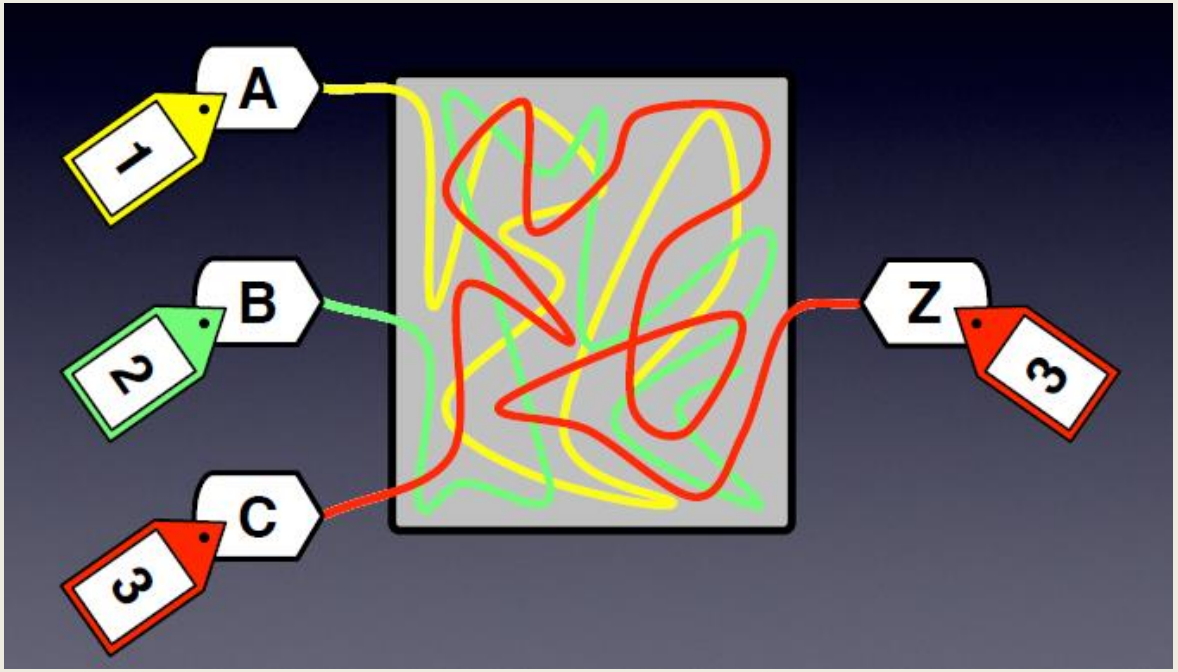


CSV

Reports
- Attack trees
- Ease of Attack
- Capabilistic Propensity
- Relative Risk

Future: SysML Tags and Taint Analysis

- In the future, we want to be able to tag data and observe how “tainted” data pervades the system – infection and effects
- It’s like connecting the dots in a large space – what path does an attack take and what data paths are most harmful.
- Taint analysis at the binary level is already being used by cyber analysts – we are moving it up to the model world.
  - *J Clause, P Li, and A Orso Dytan: A Generic Dynamic Taint Analysis Framework Georgia Institute of Technology, DHS CCR-0205422 report*



Implementing the Dashboard: Integration of SecurITree/MagicDraw for Selection of Defensive Solutions

- We need model consistency between SysML and Attack trees
- When you change a model in SysML, need to propagate that change in the attack tree and vice versa.
- Linking the Attack tree and SysML model world is not straightforward – very different SW structures.
- Amenaza is providing a external database update solution to us
  - Basically, whenever a node in the Attack tree is changed (created, modified, deleted) in Secur*IT*ree, you will be able to have an entry written to a logfile for parsing at your convenience.
  - You will also be able to ask Secur*IT*ree to invoke a user defined Java function that will perform whatever operations you like (like updating a data base, for instance)
- Extracting records from the database will enable support for decision-maker visualization.

Our Vision: Bring the Power of System Level Modeling to Enable a Holistic Perspective on Cyber Security

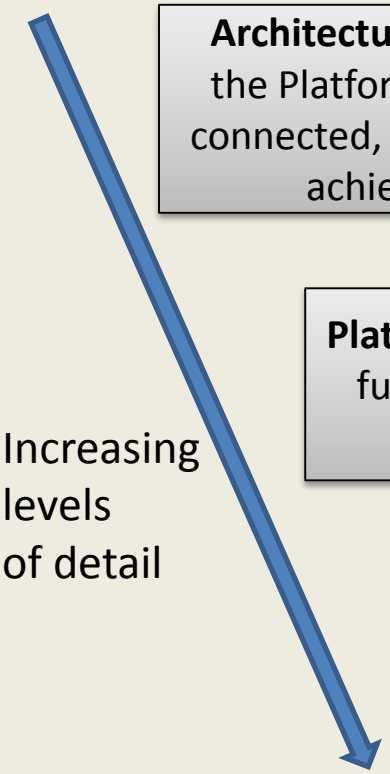
Mission Domain – What are all of these integrated systems trying to achieve for us?

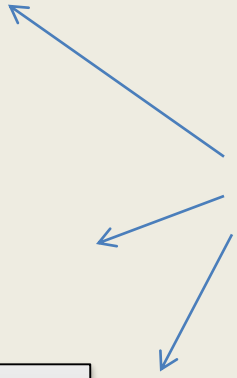
Architecture Domain – How are all of the Platforms/sub-systems organized, connected, and related to each other to achieve mission objectives

Platform domain –What are the Platform functions providing or requiring in the context of mission

Functional Domain – How do we describe function behavior, input/output, state, and controlability – accurately

Increasing levels of detail





Model based Cyber Security Framework

- support decision making by providing model based reasoning along these dimensions
- Provide a models to collect insight that otherwise could be overlooked
- Integrate Exploit Tools (Attack Trees) to the framework
- Be able to access the criticality of platforms and functions with respect to mission
- Evaluate cyber-defenses

Principal Research Issue - Productivity

- How deep in systems modeling does one need to go to get benefits?
- Effort vs results tradeoffs?
- How much work can be used the next time around?

The Transition Approach is in Motion Before the Proof of Value is Completed

- Policy: Work is funded by OSD, where it has already been exposed to a variety of policy stakeholders
- Process:
  - To minimize user issues, the research project has engaged tool users as prototype developers of the tool integration approach
  - This workshop has exposed the concept of tool integration to support decision-making
- Technology:
  - Started engaging with tool vendors to gain interest in tool integration as part of their product lines
  - Started exposing the process approach to cybersecurity service companies to gain their interest and initiative

Project Evolution

- FY 14:
  - Complement the holistic cost/benefit tools with tools for system and attack modeling
- FY 15:
- Build on Workshop outcomes regarding definition of desirable decision support capabilities
- Complete SysML/SecureItree integration and development of dashboard/workbench for solution selection
- Work with OSD to gain service involvement in proving both inputs and assessments of tools

System Aware Cyber Security Publications

JOURNAL ARTICLES:

- B.M. Horowitz and R.A. Jones, Smart security sentinels for providing point defense cyber security of critical system functions, Computers and Security Journal, Under Review
- R. A. Jones, B. Luckett, P. Beling, B. M. Horowitz, Architectural Scoring Framework for the Creation and Evaluation of System-Aware Cyber Security Solutions, Journal of Environmental Systems and Decisions 33, no. 3 (2013): 341-361.
- B. M. Horowtiz and K. M. Pierce, The integration of diversely redundant designs, dynamic system models, and state estimation technology to the cyber security of physical systems, Systems Engineering, vol 16, Issue 4 (2013): 401-412
- R. A. Jones and B. M. Horowitz, A system-aware cyber security architecture, Systems Engineering, Volume 15, No. 2 (2012), 224-240.
- J. L. Bayuk and B. M. Horowitz, An architectural systems engineering methodology for addressing cyber security, Systems Engineering 14 (2011), 294-304.

REFEREED CONFERENCE ARTICLES

- G. L. Babineau, R. A. Jones, and B. M. Horowitz, A system-aware cyber security method for shipboard control systems with a method described to evaluate cyber security solutions, 2012 IEEE International Conference on Technologies for Homeland Security (HST), 2012.
- R.A. Jones, T.V. Nguyen, and B.M. Horowitz, System-Aware security for nuclear power systems, 2011 IEEE International Conference on Technologies for Homeland Security (HST), 2011, pp. 224-229.

Thank You!

Questions?