Cyber Defense of Complex Ground Systems

Can you defend what you don’t understand?

Maddalena Jackson
Eric Rice
Elyse Fosse

©2014 California Institute of Technology. Government sponsorship acknowledged. Published by The Aerospace Corporation with permission.
State of the Art: System Knowledge

- Spreadsheets
- Charts
- Repositories
- Unidentified Scripts
- Old review material
- Pictures of whiteboards
- Brains
- Presentations
- Emails

Static
Conflicting
Unlabeled
Scattered
Out-of-date
Arbitrary Information Architecture

This is not practical
A few first principles...

What are we protecting? (What might attackers want?)

- Personal Data
- Safety & Integrity
- Control
- Technology
- Financial
- Political
- Privacy
- Products
- Tech Specifications
- Tech Data

How do we protect those? What can attackers affect?

- Devices
- Data
- People

Cyber attacks no longer directed only at sensitive data...

Of course, we are also protecting these from accidents and non-malicious faults and failures
Let’s be absurd for a minute...

Why do we need to understand our system, anyway?

- No idea what “symptoms” to look for
- No idea of correct structure and function
- No idea of relative importance of assets

JUST KIDDING

Highly effective threat prevention & neutralization

JUST KIDDING
Back to reality...

• Can we effectively defend if we...
  – Don’t know what symptoms to look for?
  – Don’t know correct structure and function?
  – Don’t know importance of each part?

• Why do we need new approaches here?
  – Are you completely confident that your system is
    • Totally impenetrable, or
    • Entirely resilient to attack
  – Is every single person a SME in every aspect of your system?
  – Do you have an all-knowing benevolent AI that you are keeping to yourself?
How do we think about the problem?

Assumptions:
- Systems will be attacked
- Attacks will evolve

Focus of our effort:
Give defenders tactical advantages - respond quickly & accurately to threats

Assessment
What is system supposed to be doing? What is it actually doing? What might happen in the future? What happened in the past?

Goals, assets, relationships, criticality, existing vulnerabilities, etc.

Current and historical state of assets, access to assets, exchanges of data, etc.

Can’t we do this now?
Yes, but we could be more effective.
What do we want to provide?

- System Assessment
  - What is system supposed to be doing?
  - What is it actually doing?
  - What might happen in the future?
  - What happened in the past?

- Observables in the System
  - Current and historical state of assets, access to assets, exchanges of data, etc.

- Strategies to neutralize attack

- Potential impacts of current attack

- Aware of current attack

- System Knowledge
  - Goals, assets, relationships, criticality, existing vulnerabilities, etc.

- Implementation

- Visualization

- Monitoring

- Analysis

Questions:
- Are we visualizing the same thing that we are analyzing?
- Are we monitoring the right things?
- Are we reasoning about the right system?

What if the system has changed?
Areas of complexity

Overwhelming amounts of data

What should we monitor? *Should we just monitor EVERYTHING?*

Distinguish important data from noise

Limited impact assessment
*Access to information to build mental model: slow, difficult, tedious*

What are we assessing?

How do we communicate the results?
A system model...

Analysis and visualization require system context; knowledge must be CONSISTENT.

We need...

Integrated Source of Truth

A model of the system

Populated System Model

Machine-readable

Structured

Dynamic

Consistent

Adaptable

Queryable

System Knowledge
Goals, assets, relationships, criticality, functions, existing vulnerabilities, etc.

Intent

Implementation
Building the system model

CS Engineers, Systems Engineers

Concerns, Views, Lessons Learned, Requirements, etc.

Concept Model

Provides framework for

Populated System Model

Subject Matter Experts
How does this actually work?

This process is the “common way to discuss, reason about, detect, diagnose and remediate”
Future concept

CS Stakeholders

Concerns, Views, Lessons Learned, Requirements, etc.

Implementation, Design, Assets, Goals, etc.

Monitoring System

Visualization System

Analysis System

System Evolution (configuration managed)

Deployed System

Operator/User

Cyber-Defense Analyses & Products
- Impact Assessments
- Monitoring System Requirements
- Analysis System Specifications & Interfaces
...etc

Traditional SE Analyses & Products
- Requirements Definition & Analysis
- Verification & Validation
- Risk Assessment
- Interface Management

Traditional SE Analyses & Products
- Requirements Definition & Analysis
- Verification & Validation
- Risk Assessment
- Interface Management

System Evolution (configuration managed)
Research areas

- Information architectures
  - Detection, Diagnosis, Remediation system concerns
  - Operator & Stakeholder views
  - Systems Engineering Domains applied to CS
    - Verification & Validation, Fault Analysis, Mission Goals & Definition, GDSE, Interface Management, etc.

- Capture & assessment of system design vs. operational behavior

- Visualization
  - Concise presentation of complex analysis
    - Future state projections
    - Differences in expected and observed behavior
    - Impact assessments
    - Root cause analysis
    - Instinctive understanding of system state

- Interoperability with existing models

- Keeping models current
  - Knowledge infusion back to model
  - Configuration management & gatekeeping

Questions?
Acknowlegdements & references

• References

• Acknowledgements
  – Thom Mcvittie, JPL
  – Arun Viswanathan, USC/ISI
  – Kymie Tan, JPL
  – Frank Kuykendall, JPL
  – Marc Sarrel, JPL

Questions?
Backup
What would you visualize?

• Goal: Defenders...
  – Act *quickly*
  – Understand *what is happening*
  – Act on *latest information*
  – Understand *consequences*
  – *trust* information
What does the model do?

- **Capture:**
  - Combined system models representing multi-faceted knowledge
    - What the system is **intended** to do
    - What the system **actually** does
    - System deployment and configuration (hardware, software, interfaces, protocols, geographic locations, versions, data paths, etc.)

- **Automate:**
  - Analyses & calculations **normally done by hand**

- **Extract (Query):**
  - DOCUMENTS, diagrams, matrices, tables, etc.
  - Analysis & calculation results
  - Operations concept documents, operational constraints
  - Asset criticality information (docs, tables)
  - Asset accessibility reports
  - Interfaces – specifications, requirements, diagrams
  - Risk assessment – reports, tables
  - Impact assessments (via modeled or inferred dependencies)
  - Simulations
  - Known vulnerability and threat reports
  - Deployment configurations – networks, software, hardware, testbeds, processes, etc.
Application: model-driven DDR

- **Model-driven analysis & visualization system**
  - Enhances interoperability with monitoring system & analysis system

  ![Diagram of system flow](image)

  - **How?** Model outputs are inputs of other systems

- **Model feeds...**
  - Expected behaviors, actual designs, relations of functions to systems, hardware & software, relation of intended information exchanges to actual system interfaces, goals, criticality, schemas/specifications of data, events to analysis system
  - Same information to **Visualization System**
  - **Result:** visualization system displays information consistent with analysis & monitoring
Engineering Process

• How can you implement this?
  – Develop information architecture
  – Choose modeling tool / infrastructure
  – Populate the model

During Design
• Requirements definition of
  – monitoring systems
  – analysis engines
  – visualization
• Model information gathering & CM processes, feedback loops

• How can you implement this?
  – Develop information architecture
  – Choose modeling tool / infrastructure
  – Populate the model
  – Requirements definition & implementation of monitoring systems
  – Requirements engineering, design, implementation of DDRtype stuff
  – Requirements design of visualization systems
  – Model information gathering & CM processes, feedback loops
Assess the Problem

Problem breakdown
- Knowledge
- Observations
- Assessment

How does a person do it?
Observe → Analyze → Explain

How does a machine do it?

Monitoring System

Analysis System

Visualization System

Aware of current threat

Potential impacts of current threat

Strategies to neutralize threat

System Knowledge
Goals, assets, relationships, criticality, existing vulnerabilities, etc.

Observables in the System
Current and historical state of assets, access to assets, exchanges of data, etc.

System Assessment
What is system supposed to be doing? What is it actually doing? What might happen in the future? What happened in the past?

Implementation

Intent
Traditional SE domains!
Requirements Definition & Analysis
Verification & Validation
Risk Assessment
Interface Management

Not just useful in a deployed system!