Verification of Security Response

Thomas C. Eskridge, Siddhartha Bhattacharyya, and Marco Carvalho

Florida Institute of Technology
Harris Institute for Assured Information
Thomas C Eskridge

- Associate Professor at the Florida Institute of Technology
- Member of the Harris Institute for Assured Information

Background:
- Computer Science
  - Machine Learning, Computer Security and Complex Networks
- Cognitive Science
  - Similarity-based Reasoning, Instance-based Reasoning, Rule-based Reasoning

2016 Safe and Secure Systems and Software Symposium
Motivation

• Adaptive software is controlling or coordinating computer network defense systems to respond to increasing threats and the alerts they generate

• Agent systems have proven to be an effective implementation of this kind of software

• Command and Control of these agents needs to be rigorously analyzed to guarantee satisfaction of requirements, correctness and to build trust in automation
Research Challenges

• Desired effects of security responses are known, but implications of responses may not be known
  – Representation in formal environment allows many implications to be analyzed
  – Allows ordering among “equivalent” responses
Our Premises and Assumptions

- There is a need for a “general” approach to *dynamic defense coordination* in complex environments.

- Fully automated control of highly complex systems tends to be brittle against intelligent adversaries; a solution must *strike the right balance between human and automation*.

- Resilience lies not on the capabilities of individual defenses, but on the *coordinated and contextual* (i.e. mission-dependent) use of multiple defenses.
Controlling Multiple Sensors and Defenses

Enabling Infrastructure-level Control through semantic policies
Policies

- Semantic representations restricting actors from performing actions or obligating actors to perform actions
- All actions are governed by policies
- State can be manipulated by policies
The MIRA Framework

- Mission-aware Infrastructure for Resilient Agents
- Implemented in JAVA
- The MIRA Agent System is Composed of:
  - The Mira Execution Environment
    - Provides service infrastructure
    - Includes default service providers
  - The Mira Agents
    - User-defined software components
    - Provides interface to sensor and defenses
    - Interfaces to services, independent of providers

An agent-based modular infrastructure that allows plug-and-play of different service components:
- Flexibility on service deployment
- Spatial and temporal diversity of the C2 infrastructure

Florida Institute of Technology
2016 Safe and Secure Systems and Software Symposium
Verification

- Verification can be an infrastructure service
- Available to all other services
- Applicable to any agent in the system
- Makes use of other services
- Mediates operations through proof/model checking
Example

- We find evidence that Fred is up to no good, and want to prevent him from accessing our servers
- We have several options
  - Disconnect servers from network
  - Refuse communications from Fred’s subnet
  - Log Fred out and change his password
Disconnect Servers
Isolate Subnet
Boot Fred
Representing Options and Effects

- Representations allow computing the number of users affected by each action.
- Representations specify effects:
  - Enable/disable access
  - Enable/disable write

Select count(u) where
s typeof Subnet
s contains c
c typeof Computer
c hasLogon l
l hasUser u
Formal architectural representation
process AppropriateResponse
  features
    Action_In: in data port Base_types::integer;
    Access_Enable: out data port Base_types::boolean;
  annex agree**
  -- Evaluating if the action can achieve the effect in a given time with less users effected
  guarantee "action achieves effect":
    Action_In = logoutUser => Access_Enable = false and
    timetoRespond <= 5 and
    numberAffected < 5;

  -- Evaluating if either of the actions can achieve the effect in a given time with less users effected
  --
  guarantee "action to achieve":
    Action_In = logoutUser or
    Action_In = disconnectServer =>
    Access_Enable = false and
    timetoRespond < 3 and
    numberAffected < 5;

**};
end AppropriateResponse;
Counterexample for Decision Making

guarantee "action to achieve":
  Action_In = logoutUser or
  Action_In = disconnectServer =>
    Access_Enabled = false and
    timetoRespond < 3 and
    numberEffectected < 5;

<table>
<thead>
<tr>
<th>Step</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Access_Enabled</td>
<td>FALSE</td>
</tr>
<tr>
<td>Action_In</td>
<td>disconnectServer</td>
</tr>
<tr>
<td>access</td>
<td>FALSE</td>
</tr>
<tr>
<td>action</td>
<td>disconnectServer</td>
</tr>
<tr>
<td>action to achieve</td>
<td>FALSE</td>
</tr>
<tr>
<td><strong>numberEffectected</strong></td>
<td>100</td>
</tr>
<tr>
<td>timetoRespond</td>
<td>2</td>
</tr>
</tbody>
</table>
Operator Interaction

• Set context for operations
  – Prefer some responses over others
    • don't use Windows Web Servers
    • response must occur within 3 minutes
    • response cannot affect users in subnet A

• Decide priorities
  – Given alternatives, choose between them
Working with Operators

• Semantic Policy Service enables operators to interact with agents to guide processing
• Verification of multiple alternatives
• Counter examples show on which attribute the alternatives differ
  – operators determines if difference is important
  – add policy to constrain attributes to appropriate solution
Next Steps

• Developing methodology specifying assume/guarantee as part of response specification
• Verification as part of the operational process – if verification fails, then response is not issued
Next Steps

• Define interactions between verification and user
  – semantic policies enable constraints on verification assertions to prove

• Implementation of VerificationService in MIRA agent framework
Questions?

Verification of Security Response

Thomas C Eskridge
teskridge@fit.edu
Constructing Defense Representations

Wrapper

Speciation
un/Install
start/stop

Ontology

Control
Agent

Decision
Process

command
vocabulary

defense
api

Defense
Control
Agent

Defense
Interpreter

Wrapper
Server