Project Introduction – Problem

Do more for less—budgets are coming under increasing pressure

• Reuse: software architectures and components (off-the-shelf and otherwise)
• Open and common interfaces: better integration between systems

Intent is to achieve economies of scale for producing software

However, cyber attackers also achieve economies of scale for attacking software

• Increases the pool of potential targets of like systems

Economic disparity

• Producers need to defend against all attacks, *a priori*, for that which is presently known
• Attackers need only to find one exploit in a common part to inflict wide-spread damage
Project Introduction – Solution

Improve the ability to resist attacks on systems with common architectures by sharing threat information and using coordinated architecture-based self-adaptation.

Key idea: exploit commonality to gain a defense advantage

- Coordination based on threat information exchange to enable proactive defense.
- Proactive adaptation allows changes to be done in time to resist the attack.
- Architecture-based adaptation makes explicit quality attribute tradeoffs.
CABSP Proof of Concept

Secure coordination “bus”

- Serial or Parallel attack
- Port scanning, DoS, system crash
- Maintain throughput

Fig. 5. Top: Information sharing pattern. Bottom: concrete instance of the pattern.

MAPE Information Sharing Pattern*


*Image of the Kagura Villain is licensed under the Creative Commons Attribution 3.0 Unported license with attribution to Davmandy at en.wikipedia
Intended results

Goal: Deny the possibility of reusing attacks on systems that use common architectures.

Success evaluation:

• In our CABSP proof of concept consisting of a collection of similar systems:
  – No threat: instances’ and aggregate throughput is higher than with MT.
  – Threat: instances’ and aggregate throughput is higher than with SP.

Produce the following:

• Algorithm for proactive adaptation
  – promoting diversity, and avoiding vulnerable variant when attacked
• Architecture for coordinated adaptation
  – what information and how to exchange it to guide adaptation
• Proof of concept
  – based on Rainbow’s ZNN.com (revised as needed)
  – different defense approaches: MT, SP, CABSP
Team: Coordinating Architecture-Based Self-Protection Systems

Members
• Javier Camara
• David Garlan
• Jeffrey Gennari
• Scott Hissam
• Mark Klein
• Gabriel Moreno
• Linda Northrop
• Bradley Schmerl
• Greg Shannon

Contributing Work
• CMU’s Rainbow (self-adaptation framework)
• SEI’s Architecture Tradeoff Analysis
• SEI’s Software Architecture Modeling
• SEI’s Software Product Lines
Questions
Plan of research

**Primary goal:** Use CABSP to improve the ability of systems with common architectures to proactively resist attacks.

**Hypothesis:** CABSP-based systems will maintain higher throughput than systems that use other defenses (e.g. MT and SP).

- **Key questions:**
  - How and what do we communicate to coordinate adaptation?
  - How do we determine and quantify whether and when an adaptation will be effective in other, similar systems?

**Experiments:**

- **Scenario based**
  - Implement proof of concept with specific attack scenarios and different defensive approaches
    - Defense: CABSP, MT, and SP.
    - Attacks: port scanning, DoS, system crash (in series or parallel).
  - **Metric:** throughput of the collection of systems
    - To maintain high throughput constituent systems must remain alive, and performance overhead must be kept low.