Requirements-Driven Mediation for Collaborative Security

Amel Bennaceur
The Open University, UK
Collaborative Security

- Making multiple, heterogeneous, software-intensive components collaborate in order to meet security requirements
  - The boundary of the systems is uncertain
  - The components can change
  - The components are designed and implemented independently

Diagram:

- Component 1
- Component 2
- Component 3
- Component 4
- Component 5

Environment
Collaborative Security - Example

Protect phone from theft

Keep the room accessible if possible

**Collaborative Security:**

*Make NAO and Create collaborate to protect the phone and keep the room accessible*

*Lock*: I can lock and unlock the room

*NAO*: I can see, talk, and pick up objects

*Create*: I can clean and move
Adaptive Security meets Collaborative Adaptation

Adaptive Security

Collaborative Adaptation

- Reasoning about assets, threats, attacks, and vulnerabilities
- Identify the security controls necessary to keep security requirements satisfied
- How to enact these security controls?

- Reasoning about dynamic discovery and composition
- Making multiple components collaborate
- How to reason about assets, threats and security controls?
Collaborative Security à la Michael Jackson

\[ R = \{ R_s, R_1, \ldots, R_m \} \quad : \quad \text{partially ordered set of requirements} \]
\[ S = \{ C_1, \ldots, C_n \} \quad : \quad \text{set of components’ capabilities} \]
\[ E \quad : \quad \text{environment properties} \]

**Find** \( C \subseteq \mathcal{P}(S) \) **and synthesize** \( C, M, E \vdash R \)

**Feature based**

**Behaviour based**
Collaborative Security Framework

Components’ Capabilities $S$

Operational Environment $E$

Discover

Component 1
Component 2
Component 3
Component 4
Capabilities as Featured Transition Systems

Diagram showing transitions and actions:
- Location: location(NAO) = location
- Actions:
  - Standup: RobotPosture
  - Move to: location(location) / Navigation
  - Locate: object/ObjectRecognition
  - Drop: object/RobotPosture
  - Pick: object/RobotPosture
  - Connect: Connection
  - Disconnect: Connection
  - Say: text/TextToSpeech
  - Location: location(NAO) = location(object)
Collaborative Security Framework

- Requirements $R$
- Security Controls $SC$
- Components’ Capabilities $S$
- Operational Environment $E$
- Analyse
- Discover

- Component 1
- Component 2
- Component 3
- Component 4
Identifying Security Controls

Level 1
Requirements

Level 2
Security controls & domain assumptions

Level 3
Features & attributes

Legend
- Feature
- Domain Property
- Soft Goal
- Goal
- Potential conflict
- Refinement
Collaborative Security Framework

- Requirements $R$
- Analyse
- Feature Selection
- Components’ Capabilities $S$
- Discover
- Operational Environment $E$

Security Controls $SC$

Component 1
Component 2
Component 3
Component 4
Feature Selection using Constraint Programming

\[ X = \{x_1, x_2, \ldots, x_n\} \]

\[ D(X) = \mathcal{P}(\mathcal{F}_1) \times \mathcal{P}(\mathcal{F}_2) \times \cdots \times \mathcal{P}(\mathcal{F}_n) \]

Feature-based Constraints \( \mathcal{C}_1, \mathcal{C}_2 \)

Optimisation functions \( g_{A_1}, g_{A_2}, \ldots, g_{A_k} \)

CP Solver \( f_1, f_2, \ldots, f_n \)

\( \mathcal{C}_1: \) **Subsumes** the features of a selected security control provided some domain properties

\( \mathcal{C}_2: \) **Respects** the constraints between features
Feature Selection
Collaborative Security Framework

Requirements $R$

analyse

Security Controls $SC$

$C$

Features-driven Mediator Synthesis

Components’ Capabilities $S$

discover

M

Component 1

Component 2

Component 3

Component 4

Operational Environment $E$
Projection of Featured Transition Systems
Feature-based Mediation

location = 0

location(NAO) = location

locate(object)

pick(object)

location(object)

drop(object)

location(NAO) = location(object)

location = 0

location(Create) = location

move(location)

Achieve [MovingPhoneToTheSafe]

\[ location(phone) = location.SAFE \]
Features-driven Mediator Synthesis

- Use the selected features to project the behaviour of the components

- Synthesise, if possible, a mediator that enables the composed system to reach

\[ fts_1|_{f_1} \parallel fts_2|_{f_2} \parallel \ldots fts_n|_{f_n} \parallel M \models_B G_s \]
Collaborative Security Framework

- Analyse Security Controls Requirements \( R \)
- Discover Components' Capabilities \( S \)
- Feature Selection \( C \)
- Features-driven Mediator Synthesis

Secure Operational Environment \( E \)

Components:
- Component 1
- Component 2
- Component 3
- Component 4

Mediator

M deploy
Tool Support

http://sead1.open.ac.uk/fics/
Summary

- Features and behavioural models to reason about and achieve collaborative security
- Capability selection (and mediation) as a multi-objective optimisation problem
- Features to scope components’ behaviours and reduce the space for mediation
Open Questions

- Can collaboration be applied to other types of requirements besides security?
  - Yes but security exacerbates and opens many issues that make collaboration more challenging, e.g., dealing with change and assurance.

- What are the limitations of the approach?
  - Predefined set of security control
  - Shared vocabulary between the specification of security controls and capabilities
  - Independent iterations between feature selection and mediator synthesis
  - Individual components are trustworthy and implement the capabilities advertised
Open Questions

- How about the user?
  - How to explain the choice and implementation of the security control?
  - Is the user just another component?

- Where do the models come from? What is the impact of their inaccuracy on the model?
Thank you

www.amel.me

http://sead1.open.ac.uk/fics/

Adaptive Security and Privacy

www.asap-project.eu