SeCMER: How to Stay in Control of Security Requirements Evolution

Gábor Bergmann [a] , Fabio Massacci [b] , Federica Paci [b] ,Thein Tun [c] , Dániel Varró [a] , and Yijun Yu [c]

[a] Budapest University of Technology and Economics, [b] University of Trento, [c] The Open University

System Evolution

✓ Long-lived systems need to be flexible and to adapt to changes in order to remain useful.
✓ Software-based systems are getting increasingly security-critical since software now pervades the whole critical infrastructures dealing with critical data.
✓ A challenging aspect is thus to develop techniques and tools that ensure long-running evolving software systems are compliant to evolving security, privacy and dependability requirements.

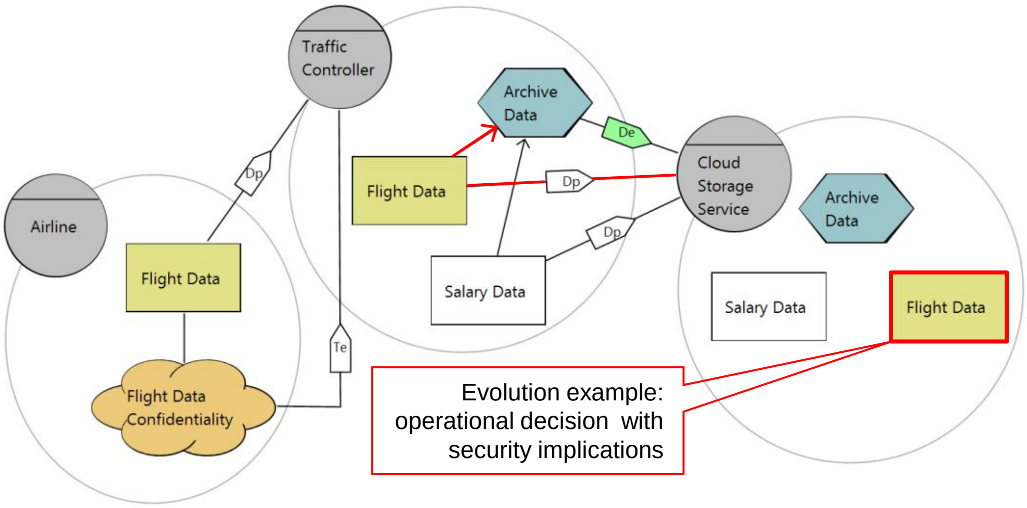
Motivating Case Study: Evolution in ATM Systems

The new Air Traffic Management (ATM) Target Concept of the SESAR Initiative [1], requires ATM services to go through structural, operational and cultural changes that will contribute to the creation of a new Air Smart Transport system..

The new Trajectory Management Operational Concept is based on:
✓A Collaborative Decision Making among all air transport actors to define a rolling Network Operations Plan ,
✓The sharing of updated and precise information trough the SWIM Network to negotiate in real time business trajectories,
✓An extensive use of automation support to reduce controllers and pilots task load.
✓The deployment of new IT systems are changing the nature of ATM services itself. From 'closed' and dedicated systems, ATM services are relying more and more on 'open', ubiquitous and 'smart' systems.

ATM systems are becoming vulnerable to new types of security hazards. Trust problems arise between ATM stakeholders, who have to rely on mediated information.

We are investigating **evolution scenarios** including (a) the introdution of the SWIM network raising concerns of data protection in a communication channel, or (b) operational decisions affecting the security of sensitive data of other stakeholders (see Figure 1).



Evolution example: operational decision with security implications

Figure 1. Example evolution of an ATM system, modeled in SeCMER

Modeling Requirements

A model represent entities such as actors, the resources and the actions they are willing to provide, the security goals they want to achieve, and the relationships between them.

Figure 1 depicts an Airline with sensitive information and an Air Traffic Controller that is entrusted with the data asset but decides to have it archived by a 3rd party.

Secure Evolution of Requirements

A challenging aspect is how to design software systems that are evolvable and secure. To reason on evolution and security, it is important to:

✓ **Model the change**. Change should be a first-class citizen in the modeling.

✓ **Evaluate the impact of the change.** Impact analysis consists in identifying what it is affected by a change to a particular artefact describing the system.

✓ **Manage the change.** Change management deals with the analysis, control, and the implementation of a change. The analysis determines the sources of the change, its effectiveness and feasibility, e.g. whether the change violates security requirements.

The SeCMER Tool

SeCMER [2] is an Eclipse-based modeling environment for managing evolving requirements. It has the following features:

✓**Modeling of Evolving Requirements**. Requirement models can be drawn in Si*, or textually specified in SeCMER. Traceability and bidirectional synchronization is supported between SeCMER and Si* / Tropos requirements models.

✓**Argumentation-based security analysis.** Reasoning about security properties satisfaction and identification of new security properties is supported.

✓**Change detection based on evolution rules**:

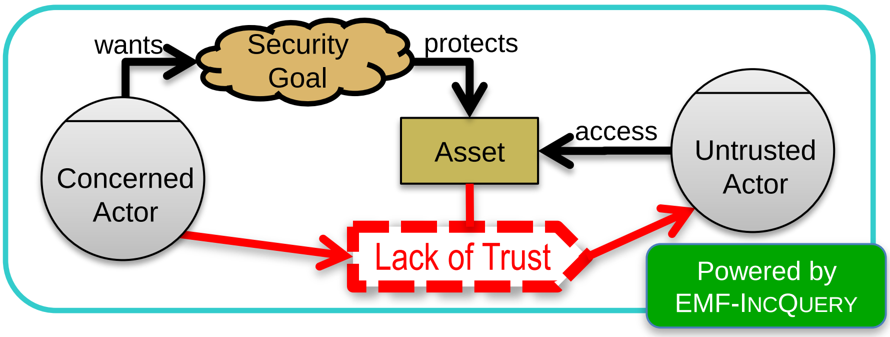
✓Violations of formally defined **static security properties** expressed as security patterns can be automatically identified. See Figure 2 for a simplified example pattern.

✓Automatically providing corrective actions (**quick fixes**) for the detected violation of a security property.

✓Detection of formal or informal arguments that have been **invalidated by changes** affecting model elements that contributed to the argument as evidence is also supported.

Evolution rules and security patterns can be defined using patterns expressed in the query language of EMF-IncQuery [3] and inserted into the tool as a plug-in.

Figure 2. Simplified security pattern: access by untrusted actor



Powered by EMF-IncQuery

References

1.SESAR D3 – The ATM Target Concept, SESAR Consortium, 2008.

2.Bergmann, G., Massacci, F., Paci, F., Tun, T., Varró, D., and Yu, Y., "A Tool for Managing Evolving Security Requirements", CAiSE'11 Forum, LNCS 734, CEUR-WS, pp. 49-56, 06/2011.

3. Bergmann, G., Horváth, Á., Ráth, I., Varró, D., Balogh, A., Balogh, Z., and Ökrös, A., "Incremental Evaluation of Model Queries over EMF Models", MODELS'10: Springer, 10/2010.

4. Massacci F., Bouquet F., Fourneret E., Jurjens J., Lund M. S., Madelénat S.,Muehlberg J.T., Paci F., Paul S., Piessens F., Solhaug B., Wenzel S., "Orchestrating Security and System Engineering for Evolving Systems", ServiceWave 2011, LNCS 6994, p.134-143, 10/2011