A framework for semi-automated co-evolution of security knowledge and system models (Summary)

Jens Bürger[1], Daniel Strüber[2], Stefan Gärtner[3], Thomas Ruhroth[4], Jan Jürjens[5,6], Kurt Schneider[7]

We present a summary of our article published in Elsevier's *Journal of Systems and Software* in 2018 [Bü18]. The presented approach has been developed in context of the SecVolution project, being part of the DFG SPP1593 *Design For Future*.

Security is an important and challenging quality aspect of software-intensive systems, and it becomes even more demanding in the case of long-living systems. Security issues do not necessarily arise from a flawed design, but can also manifest when the system fails to keep up with a changing environment, e.g., when a novel attack is discovered or a new law is passed. Thus, ongoing adaptations at system operation phase in response to security knowledge changes are inevitable.

We present a model-based framework for supporting the maintenance of security during the long-term evolution of a software system. It uses ontologies to manage the system-specific and the security knowledge. With model queries, graph transformation and differencing techniques, knowledge changes are analyzed and the system model is adapted.

We introduce the novel concept of *Security Maintenance Rules* to couple the evolution of security knowledge with co-evolutions of the system model.

To evaluate our technique, we used available community knowledge from various sources, including the Common Weakness Enumeration database (CWE). We demonstrate the framework by applying it to the *iTrust* system from the medical care domain and hence show the benefits of supporting co-evolution for maintaining security-critical systems.

The SecVolution approach is a holistic framework to deal with evolving knowledge in the environment of a software project. Existing approaches for secure software design fall

[1] University of Koblenz-Landau, Universitätsstraße 1, 56070 Koblenz, Germany buerger@uni-koblenz.de
[2] University of Koblenz-Landau, Universitätsstraße 1, 56070 Koblenz, Germany strueber@uni-koblenz.de
[3] adesso AG, Adessoplatz 1, 44269 Dortmund, Germany stefan.gaertner@adesso.de
[4] msg systems ag, Kruppstraße 82-100, 45145 Essen, Germany thomas.ruhroth@msg.group
[5] University of Koblenz-Landau, Universitätsstraße 1, 56070 Koblenz, Germany
[6] Fraunhofer ISST, Emil-Figge-Straße 91, 44227 Dortmund, Germany http://jan.jurjens.de
[7] Leibniz University Hannover, Welfengarten 1, 30167 Hannover, Germany kurt.schneider@inf.uni-hannover.de

doi:10.18420/se2019-57

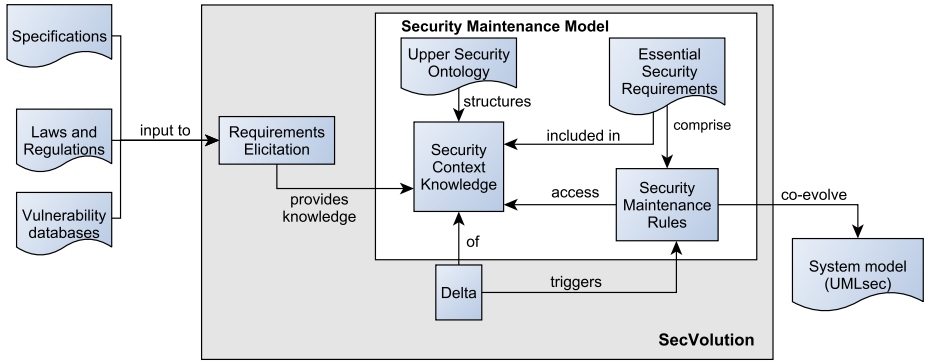

Fig. 1: Overview of the SecVolution approach

short on keeping up with environmental changes, thus providing only *one-shot* security. The overall goal is to restore security levels of an information system when changes in the environment put security at risk. At the beginning the software models are considered secure in accordance. As a triggering event the environmental knowledge or requirements concerning the software-project change are leveraged.

Figure 1 depicts an overview of the SecVolution approach in the publication's focus. Inputs are e.g. specification documents of various types. Laws and regulations provide knowledge about general security obligations. Vulnerability databases contain knowledge about security best practices and also known vulnerabilities in frameworks and algorithms, typically with appropriate mitigations.

Security-relevant knowledge is elicited and captured in an explicit representation called *Security Maintenance Model* (SMM). Security requirements that are defined on a coarse grained, *essential*, level, are used to define security requirements independent from their concrete technical realization. The security knowledge is based on a upper ontology for security notions we provide. Evolution of the security knowledge is captured as difference information which triggers execution of appropriate co-evolution actions, called *Security Maintenance Rules* (SMR). SecVolution focuses model-based software development and uses security-enriched UML models, built upon the UML security extension UMLsec. Thus, application of SecVolution leads to a co-evolved system model that is compliant to the evolved environment again.

References

[Bü18] Bürger, Jens; Strüber, Daniel; Gärtner, Stefan; Ruhroth, Thomas; Jürjens, Jan; Schneider, Kurt: A framework for semi-automated co-evolution of security knowledge and system models. Journal of Systems and Software, 139:142 – 160, 2018.