Towards Maintaining Long-Living Information Systems by Incorporating Security Knowledge*

Stefan Gärtner[1], Thomas Ruhroth[2], Jens Bürger[2], Kurt Schneider[1], Jan Jürjens[2,3]

[1]Leibniz Universität Hannover, Germany
[2]TU Dortmund, Germany
[3]Fraunhofer ISST, Germany
stefan.gaertner@inf.uni-hannover.de

Abstract: Modern information systems are increasingly complex and need to operate in evolving environments. As a consequence, systems must co-evolve to keep up-to-date with their environments. This is especially important for security properties, since changes and patches tend to compromise them. We propose a security assessment approach for natural language requirements for systematic co-evolution. Our evaluation and tool implementation show security benefits for maintaining long-living systems.

1 Evolving Environment impacts Requirements

It is essential for software engineers to be acquainted with the requirements and their relationships to security properties when they maintain security of long-living software systems. They cannot design a secure system unless they understand the threats to security and their interrelationship within an evolving environment. However, software engineers usually do not have a complete overview of the knowledge necessary to assess requirements with respect to security.

Requirements describe the intended functionality of a system. Design flaws and vulnerabilities are often revealed under changed conditions. Therefore, environmental changes may have an impact on requirements. Identifying these flaws is important to restore affected security properties of the system. However, analyzing a huge amount of natural language requirements manually is a laborious task. The impact of knowledge changes on requirements needs to be derived semi-automatically.

The work we present in this paper is part of the research project *SecVolution*. In SecVolution, we consider long-living information systems and how to retain its security in face of constantly changing requirements and evolving environmental knowledge [BJR+14].

*This research is funded by the DFG project SecVolution which is part of the priory program SPP 1593 "Design For Future - Managed Software Evolution".

2 Heuristic Security Assessment on Requirements

To cope with security problems successfully, a high level of expertise is required. It consists of a mixture of textbook knowledge, security obligations and laws, as well as experience comprising typical and exceptional cases. Especially for novices it is difficult to find the relevant information for a particular issue.

To overcome this knowledge gap, a knowledge model is used in our approach to manage security-related knowledge. We conducted a quasi-systematic literature review to find primary security concepts and their relationships applicable for different domains. It comprises primary security concepts which can be found in the examined models in one or the other way [GRB+14]. The knowledge model can be extended using layered ontologies in order to fulfill further domain- or project-specific requirements [RGB+14].

To determine the impact of changes on requirements, we developed a heuristic security assessment approach. It identifies vulnerabilities (and their variations) in natural language requirements by leveraging security knowledge and natural language processing [GRB+14]. In particular, our approach relies on reported incidents and common attack patterns. We focus on requirements in form of use cases as they describe the interaction of the system with several actors as well as other systems.

The aim of our approach is to enable software engineers to react faster and more effectively to environmental changes. Heuristics are correct in many cases, based on previous experiences. However, there may be false positives. Therefore, findings must be verified by the software engineer or security expert. Nevertheless, reducing the amount of requirements which may contain security issues is beneficial.

To evaluate our approach, we conducted a case study using iTrust. The evaluation indicates that the proposed requirements assessment detects vulnerable requirements more reliable than other methods (Bayes, SVM, k-NN). Thus, the case study and tool implementation show the benefits for maintaining long-living systems.

References

[BJR+14] J. Bürger, J. Jürjens, T. Ruhroth, S. Gärtner, and K. Schneider. Model-based Security Engineering with UML: Managed Co-Evolution of Security Knowledge and Software Models. In A. Aldini, J. Lopez, and F. Martinelli, editors, *Foundations of Security Analysis and Desing VII: FOSAD Tutorial Lectures*, volume 8604 of *LNCS*, pages 34–53. Springer, 2014.

[GRB+14] S. Gärtner, T. Ruhroth, J. Bürger, K. Schneider, and J. Jürjens. Maintaining Requirements for Long-Living Software Systems by Incorporating Security Knowledge. In *22nd IEEE International Requirements Engineering Conference*, pages 103–112, 2014.

[RGB+14] T. Ruhroth, S. Gärtner, J. Bürger, J. Jürjens, and K. Schneider. Towards Adaptation and Evolution of Domain-specific Knowledge for Maintaining Secure Systems. In *Proceedings of the 15th International Conference on Product Focused Software Process Improvement (PROFES)*, volume 8892 of *LNCS*, pages 239–253. Springer, 2014.