The SecReq Approach: From Security Requirements to Secure Design while Managing Software Evolution

J. Jürjens

TU Dortmund & Fraunhofer ISST
Dortmund
http://www-secse.cs.tu-dortmund.de

K. Schneider

Leibniz Universität Hannover
Hannover
http://www.se.uni-hannover.de

Abstract: We present the security requirements & design approach SecReq developed in joint work over the last few years. As a core feature, this approach supports reusing security engineering experience gained during the development of security-critical software and feeding it back into the development process through the HeRA Heuristic Requirements Assistant. Based on this information a model-based security analysis of the software design can be performed using the UMLsec approach and its associated tool-platform CARiSMA. In recent work within the project DFG project SecVolution (SPP 1593 "Design For Future – Managed Software Evolution"), we have been extending the approach with techniques, tools, and processes that support security requirements and design analysis techniques for evolving information systems in order to ensure "lifelong" compliance to security requirements. heuristic tools and techniques that support elicitation of relevant changes in the environment.

1 From Security Requirements to Secure Design

Many software projects today are somehow security-related. Requirements engineers without expertise in security often overlook security requirements, leading to security vulnerabilities that can later be exploited. Identifying security-relevant requirements is labour-intensive and error-prone.

To facilitate the security requirements elicitation process, [SKHIJ12] presents an approach supporting organisational learning on security requirements by establishing company-wide experience resources, based on modelling the flow of requirements and related experiences. Based on those models, people can exchange experiences about security-relevant requirements while writing and discussing project requirements. Participating stakeholders can learn while writing requirements. This increases security awareness and facilitates learning on individual and organisational levels. As a tool basis, heuristic assistant tools [Sch08] like HeRA [SKHIJ12] support reuse of existing experiences that are relevant for security. They include Bayesian classifiers issuing a warning automatically when new requirements seem security-relevant. The approach is part of the SecReq approach introduced in [HIKJS10] and feeds into the UMLsec of which a recent application is reported in [LJ09].

Results indicate that this is feasible, in particular if the classifier is trained with domain-specific data and documents from previous projects. The paper shows how the ability to identify security-relevant requirements can be improved using this approach. It illustrates the approach with a step-by-step example of how it improved the security requirements engineering process at the European Telecommunications Standards Institute (ETSI) and reports on experiences made.

2 Maintaining Security Requirements During Software Evolution

Information systems are exposed to constantly changing environments which require constant updating. Software "ages" not by wearing out, but by failing to keep up-to-date with its environment. Security is an increasingly important quality aspect in modern information systems. At the same time, it is particularly affected by the above-mentioned risk of "software ageing". When an information system handles assets of a company or an organization, any security loophole can be exploited by attackers. Advances in knowledge and technology of attackers are part of the above-mentioned environment of a security-relevant information system. Outdated security precautions can, therefore, permit sudden and substantial losses. Security in long-living information systems, thus, requires an on-going and systematic evolution of knowledge and software.

In recent work within the project DFG project SecVolution (SPP 1593 “Design For Future – Managed Software Evolution”), we have been developing techniques, tools, and processes that support security requirements and design analysis techniques for evolving information systems in order to ensure "lifelong" compliance to security requirements, building on the SecReq approach. As a core feature, this approach supports reusing security engineering experience gained during the development of security-critical software and feeding it back into the development process. We develop a variety of heuristic tools and techniques that support elicitation of relevant changes in the environment. Findings are formalized for semi-automatic security updates. During the evolution of a long-living information system, changes in the environment are monitored and translated to adaptations that preserve or restore its security level.

References

[HIKJS10] S.H. Houmb, S. Islam, E. Knauss, J. Jürjens, K. Schneider: Eliciting security requirements and tracing them to design: an integration of Common Criteria, heuristics, and UMLsec. Requir. Eng. 15(1): 63-93 (2010).

[Jür05] J. Jürjens: Secure systems development with UML. Springer 2005

[LJ09] J. Lloyd, J. Jürjens: Security Analysis of a Biometric Authentication System Using UMLsec and JML. MoDELS 2009: 77-91.

[MJ10] H. Mouratidis, J. Jürjens: From goal-driven security requirements engineering to secure design. Int. J. Intell. Syst. 25(8): 813-840 (2010)

[Sch08] Schneider, K.: Improving Feedback on Requirements through Heuristics. Proceedings of 4th World Congress for Software Quality (4WCSQ), Washington D.C., USA (2008)

[SKHIJ12] K. Schneider, E. Knauss, S.H. Houmb, S. Islam, J. Jürjens: Enhancing Security Requirements Engineering by Organisational Learning in Requirements Engineering, Requirements Engineering Journal, Vol. 17, 2012, pp.35-56.