Model-based Security Verification for Evolving Systems

Jan Jürjens[1,2], Sven Wenzel[2], Daniel Poggenpohl[2], Martín Ochoa[3]

Abstract: Security certification of complex systems requires a high amount of effort. As a particular challenge, today's systems are increasingly long-living and subject to continuous change. After each change of some part of the system, the whole system needs to be re-certified from scratch (since security properties are not in general modular), which is usually far too much effort. We present a tool-supported approach for security certification that minimizes the amount of effort necessary in the case of re-certification after change. It is based on an approach for model-based development of secure software which makes use of the security extension UMLsec of the Unified Modeling Language (UML). It allows the user to integrate security requirements such as secure information flow and audit security into a system design model, it supported by a security verification tool chain, and has been applied to a number of industrial applications.

Keywords: Secure Software Engineering, Model-based Software Development, Security Verification, Software Evolution.

1 Introduction

Security certification of complex systems requires a high amount of effort. Model-based development is a widely accepted methodology where software or parts of it is generated from models. In order to ensure quality properties such as consistency of security requirements the models are often verified prior code generation.

As a particular challenge, today's systems are increasingly long-living and subject to continuous change. After each change of some part of the system, the whole system needs to be re-certified from scratch (since security properties are not in general modular), which is usually far too much effort. Also, if several alternative evolutions of a model are possible, each alternative has to be modeled and verified in order to find the best model for further development and code generation.

We present a tool-supported approach for security certification that minimizes the amount of effort necessary in the case of re-certification after change. It is based on an approach for model-based development of secure software which makes use of the security extension UMLsec of the Unified Modeling Language (UML) [Jur05]. It allows the user to integrate security requirements such as secure information flow and audit security [Jur01] into a system design model and has been applied to a number of industrial applications such as an electronic purse system.

[1] Institute for Software Technology, University of Koblenz-Landau, Koblenz, Germany. http://jan.jurjens.de
[2] Fraunhofer Institute for Software and Systems Engineering ISST, Dortmund, Germany
[3] Singapore University of Technology and Design, Singapore

The approach presented is based on results that determine under which conditions change preserves security properties (for example in the context of structuring techniques such as refinement or architectural principles such as modularization). The approach supports an automated difference-based security analysis, at the level of design models as well as the implementation code (using static security verification [DGJN11] or run-time verification). It has been applied e.g. to cryptographic protocols, distributed security infrastructures, and identity management systems, and there are empirical results comparing it to classical techniques for security certification. In the outlook, we briefly present current research directions, such as applying the approach to the security certification of cloud-based systems.

We present a verification strategy to analyze whether a software evolution preserves a given security property. This is presented on the basis of the UML profile UMLchange which can be used for specifying potential evolutions of a given model simultaneously. UMLchange makes our approach independent from specific modeling tools. We also present an extensible tool that reads the annotations of EMF-based UML2 models and computes a delta model containing all possible evolution paths of the given model. The evolution paths can be verified wrt. security properties, and for each successfully verified path a new model version is generated automatically.

References

[DGJN11] F. Dupressoir, A. D. Gordon, J. Jürjens, D. A. Naumann: Guiding a General-Purpose C Verifier to Prove Cryptographic Protocols. In: 24th IEEE Computer Security Foundations Symposium (CSF), pp. 3-17, 2011.

[HGJF06] S. H. Houmb, G. Georg, J. Jürjens, R. B. France: An Integrated Security Verification and Security Solution Design Trade-off Analysis Approach. In: H. Mouratidis (editors): Integrating Security and Software Engineering: Advances and Future Vision, Idea Group, pp. 190-219, 2006. Invited chapter.

[IMJ11] S. Islam, H. Mouratidis, J. Jürjens: A Framework to Support Alignment of Secure Software Engineering with Legal Regulations. In: Journal of Software and Systems Modeling (SoSyM), Springer-Verlag, vol. 10, no. 3, pp. 369-394, 2011.

[Jur01] J. Jürjens: Modelling Audit Security for Smart-Cart Payment Schemes with UML-SEC. IFIP TC11 Sixteenth Annual Working Conference on Information Security (IFIP/Sec'01), Kluwer 2001, pp. 93-108

[Jur05] J. Jürjens: Secure Systems Development with UML, Springer, 2005

[PWP+07] D. Petriu, M. Woodside, D. Petriu, Jing Xu, T. Israr, G. Georg, R. France, J. Bieman, S. H. Houmb, J. Jürjens: Performance Analysis of Security Aspects in UML Models. In: Sixth Int. Works. on Software and Performance (WOSP'07), pp. 91-102, ACM, 2007.

[WWJO14] S. Wenzel, D. Warzecha, J. Jürjens, M. Ochoa: UMLchange - Specifying Model Changes to Support Security Verification of Potential Evolution. In: Journal of Computer Standards & Interfaces, vol. 36, pp. 776-791, 2014. Special Issue on