A Model-driven Approach to Secure Development Lifecycle *

Zhendong Ma[1], Christian Wagner[1], Arndt Bonitz[1], Thomas Bleier[1], Robert Woitsch[2], and Markus Nichterl[2]

[1] Safety & Security Department, Austrian Institute of Technology
Siebersdorf 2444, Austria
firstname.lastname@ait.ac.at
[2] BOC Asset Management GmbH
Bäckerstraße 5, 1110 Vienna, Austria
firstname.lastname@boc-eu.com

Abstract. Building security into software development lifecycles and doing it right is hard. To address the challenge, several prominent organizations have published process-oriented security guidelines to bring security activities into a structured way. Often they are too verbose and fuzzy to be implementable in a development lifecycle involving people with different skillsets. In this paper, we propose the model-driven secure development lifecycle (MD-SDL), an approach that leverages on modeling methods and the advances in model-driven security to simplify the process of efficiently integrating security into development lifecycles.

1 Introduction

Building security into software systems requires specialized skills and collaborations among different participants. A large number of activities related to security needs to be performed through out a software's development lifecycle. In the past, two promising approaches appeared to address security challenges in security engineering in software development lifecycles. The first is model-driven security [1–3], which applies models in security engineering to gain more focused views of complex systems and uses levels of abstraction to assist non-security experts (e.g., developers) to implement security in a correct and efficient way. The second approach is the definition of guidelines for security activities involved in a software's development lifecycle. Several organizations comprise lists of "todos" [4–6], i.e., activities regarded as best practices on the technical as well as the organizational level.

In this paper, we propose the model-driven secure development lifecycle (MD-SDL). Our approach combines rigid modeling methods and efficiency gained from model-driven security with process-oriented security development lifecycle guidelines to produce structured, practical, and efficient security engineering practices in software development lifecycles.

* This work was partly funded by the Austrian security-research programme KIRAS and by the Federal Ministry for Transport, Innovation and Technology.

2 Model-driven Secure Development Lifecycle

"Model-driven" in this paper refers to: (1) use security engineering techniques that follow principles defined in model-driven architecture, (2) use modeling method engineering to structure and support security activities in a development lifecycle, and (3) use models as a part of the artifacts in the development lifecycle. By adopting a model-driven approach to SDL, we aim at achieving the following goals: (1) To tackle complexity in software system. (2) To bring software into tangibility. (3) To conduct security engineering in a structured as well as flexible way. (4) To facilitate clear and effective communications among stakeholders. (5) To provide accurate and extensive documentation support.

The model architecture is designed to be a framework for the MD-SDL process. Our design reflects the opinions from both the security experts and the modeling experts. It also reflects our accumulated experiences in security research projects, modeling projects, and software development projects. Figure 1 shows the overall model architecture. Notice that as a first attempt to apply model-driven approach to secure development lifecycle, we focus only on the requirement, design, and implementation phase since our work centers around security design and proof-of-concept implementation. The model architecture is designed to fulfill the aforementioned goals. A part of the architecture consists of loosely connected existing security engineering techniques and tools. Each development phase includes several model components. A model component might further include several sub-components. The model architecture connects these components into a holistic framework and ensure that the artifacts and documents from one component is related to the other corresponding components.

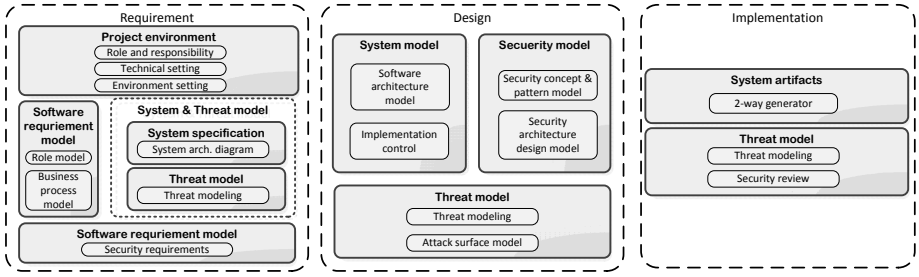

Fig. 1. Architecture overview of MD-SDL

In our approach we choose the platform provided by the Open Model Initiative [7]. Beside our consideration to avoid proprietary software, the main reason for choosing this platform is that it provides a community-based working environment and hence applies the open source concept to modeling method engineering. With regard to modeling method engineering, the Open Model Initiative provides foundational material, tools, and platforms. It also supports different modeling method engineering with specification, implementation and

deployment of modeling methods. Using the Administration Toolkit and Modeling Toolkit provided by Open Model, we can define meta models on an abstract level and then develop concrete instance models. On the Open Model platform, meta models are defined by classes and relation classes with certain attributes. The defined class is available as an object in the Modeling Toolkit and can be used for implementing model instanced. The graphical representation of objects is defined by GraphRep attributes of the class. Furthermore, we add semantics to the models by defining relation classes that control the objects' connectivities. We adopt a hybrid approach to implement the MD-SDL framework, such that we can take advantage of the latest development in the field. Currently, the models serve as an integration framework that bundles and "glues" existing methods and tools. For example, for system modeling we use UML modeling techniques; for threat modeling we adopt the Microsoft SDL Threat Modeling approach; and for code generation we use code generator developed in [8].

3 Conclusion

This paper proposed a new approach to integrate security activities into software development lifecycles based on modeling methods and the progress made in model-driven security. Our goal is to have a framework that is practical, extensible, and reproducible for different security-critical software development and research projects. Our next step is to apply the MD-SDL approach in our ongoing and upcoming security activities and promote the usage among our project partners. The results and feedbacks will be used to improve the framework and provide evidence on the feasibility of this approach.

References

1. Basin, D., Clavel, M., Egea, M.: A decade of model-driven security. In: Proceedings of the 16th ACM symposium on Access control models and technologies. pp. 1–10. SACMAT '11, ACM, New York, NY, USA (2011)
2. Basin, D., Doser, J., Lodderstedt, T.: Model driven security: From UML models to access control infrastructures. ACM Trans. Softw. Eng. Methodol. 15, 39–91 (January 2006)
3. Hafner, M., Breu, R.: Security engineering for service-oriented architectures. Springer (2009)
4. Howard, M., Lipner, S.: The Security Development Lifecycle. Microsoft Press (2006)
5. Kissel, R., Stine, K., Scholl, M., Rossman, H., Fahlsing, J., Gulick, J.: Security consideration in the system development life cycle (October 2008), `http://csrc.nist.gov/publications/PubsSPs.html`
6. SSE-CMM project: Systems security engineering capability maturity model: Model description document (version 3.0) (June 2003)
7. Open model initiative. `http://www.openmodels.at`
8. Ma, Z., Wagner, C., Bleier, T.: Model-driven security for web services in e-government system: Ideal and real. In: 7th International Conference on Next Generation Web Services Practices (NWeSP). pp. 221 –226 (October 2011)