Towards Systematically Addressing Security Variability in Software Product Lines

Laurens Sion, Dimitri Van Landuyt, Koen Yskout, Wouter Joosen
iMinds-Distrinet, KU Leuven
firstname.lastname@cs.kuleuven.be

ABSTRACT
With the increasingly pervasive role of software in society, security is becoming an important quality concern, emphasizing security by design, but it requires intensive specialization.

Security in families of systems is even harder, as diverse variants of security solutions must be considered, with even different security goals per product. Furthermore, security is not a static object but a moving target, adding variability.

For this, an approach to systematically address security concerns in software product lines is needed. It should consider security separate from other variability dimensions. The main challenges to realize this are: (i) expressing security and its variability, (ii) selecting the right solution, (iii) properly instantiating a solution, and (iv) verifying and validating it.

In this paper, we present our research agenda towards addressing the aforementioned challenges.

CCS Concepts
•Software and its engineering → Software design trade-offs;

1. INTRODUCTION
Security is becoming an increasingly important software quality, emphasized by the growing number of discovered vulnerabilities. This affirms the principle of security by design, making it being explicitly considered during development [14, 6], and even part of the development lifecycle [5]. However, building secure systems remains complex, requiring intensive specialization, while fixing newly discovered flaws further complicates things. This makes the incorporation of security a highly challenging objective.

In general, designing security solutions (driven by, for example, methods and techniques such as STRIDE [5]) is in essence a targeted search in a wide, complex, and constantly changing variability space, for a solution that meets the security requirements or represents an appropriate trade-off between different requirements. Systematic approaches are lacking to cover the entire decision making process.

To enable a systematic approach, support for expressing and modeling security [12] and its variability is needed. This allows to evaluation and instantiation of alternative security solutions using, for example, pattern instantiation [3]. In this paper, we outline our research agenda toward addressing the following research questions:

RQ1: How can we model security variability? RQ1 covers: (i) the high-level security solutions (external variability) and (ii) the realizing mechanisms (internal variability).

RQ2: How can we instantiate the security solutions in the design? RQ2 covers the instantiation of a solution and its mechanisms in the design.

RQ3: How can we measure, evaluate, or assess the instantiated security solutions? RQ3 covers: (i) evaluating whether a product fulfills the security requirement, (ii) verifying if a solution is sound (e.g., no conflicting mechanisms), and (iii) providing additional information for the selection.

RQ4: How can we offer support for selecting a specific realization of a security solution? RQ4 covers supporting the security engineer in making well-founded decisions when selecting a specific realization of a solution, which is not trivial because of the potential interactions between mechanisms.

2. RESEARCH APPROACH
We will apply demand-driven, applied research to realize the approach. More specifically, we will test and validate the ideas and approaches in industry cases, as we did in the context of safety [11], and solve the problems encountered in those specific contexts. In the end, we aspire to generalize those solutions or reach conclusions regarding the difficulties in generalizing for security. Our approach covers the different software product line engineering activities [10].

Firstly, we consider security variability, consisting of: (i) security goals, specifying the external security requirements, (ii) external variability, consisting of the different security solutions used to realize those goals, and (iii) internal variability, covering the different ways to realize such a solution. Making these aspects explicit, allows the engineer to reason about a solution and the best way to realize it.

Selection support is necessary to assist the security engineer during the selection process. There are many different possible solutions, each having different realizations, exposing the engineer to an exploding number of variants. In order to make well-founded configuration trade-off decisions, guidance is necessary to be able to make an informed decision.

Next, is the instantiation of the selected solution and mechanisms in the design, requiring: (i) a generic representation of the system, and the locations in that system where
mechanisms can be applied, (ii) a generic representation of the security mechanisms, which link to the system model elements they can be applied to, and (iii) meta-information on the security mechanisms necessary to combine or link them together in order to realize a solution.

Finally, there is the measurement and assessment of the security in the variants. This step provides the possibility to verify the correctness of the instantiated mechanisms and to allow the security engineer to make trade-off decisions between different combinations of security mechanisms to realize a security solution.

3. PRELIMINARY KEY RESULTS

In previous work, we developed a reference approach for providing selection support for the safety engineer in making trade-off decisions [11, 13]. We offered external and internal variability models for configuring the safety solution for an automotive Hall-Effect sensor and provided tactics to counter the problem of variant explosion. This is especially relevant for security with an even broader variability space. However, important differences to safety are: (i) less standardization, (ii) less domain-specific (iii) issues such as no uniform conceptual framework [2], terminological confusion [1], etc.

In previous work [12], we introduced a set of concepts to model security solutions at a single abstraction level. These should be extended to handle multiple abstraction levels in order to distinguish between high-level solutions and their realizing mechanisms. This more flexible model allows the distinction between external and internal variability models.

4. RELATED WORK

Mellado et al. [7] presented the SREPPLine approach for managing security variability in software product lines for conformance with ISO/IEC 27001 and 15408. Fægri and Hallsteinsen [4] discuss a software product line reference architecture for security, using a security submodel, linking to architectural and decision support submodels. Mylläriemi et al. presented KumbangSec [8], an approach for modeling functional and security variability by extending Kumbang with security engineering concepts. Finally, there are the security pattern catalogs, on which Yskout et al. [15] provide an aggregate view. They are a source of information on the structure of security solutions, and the mechanisms they comprise. Related to the patterns is the work of Nguyen et al. [9], which provides a feature model with security patterns and a meta-model for instantiating them.

5. CONCLUSION

The lack of a common set of security constructs and conflicting interpretations complicate the design of security solution. To improve this situation, a uniform conceptual framework is necessary. This is especially important in the context of software product lines, as it is necessary to make the link between security solutions and their mechanisms explicit to allow for trade-off evaluation and traceability.

Additionally, constructing the design of a security solution, enables automatic instantiation of the mechanisms by using techniques such as pattern instantiation. It prevents the introduction of errors or flaws, and minimizes overhead.

Finally, security mechanisms can be assessed and verified to prevent conflicting interactions and obtain stronger guarantees that a solution is correctly realized in a product. It also provides future possibilities in measuring, assessing, evaluating and making security trade-off decisions.

Acknowledgements. The presented research is partially funded by the Research Fund KU Leuven.

6. REFERENCES