Towards a Systematic Literature Review on Secure Software Design

Alexander van den Berghe Riccardo Scandariato Wouter Joosen

{firstname.lastname}@cs.kuleuven.be
iMinds-Distrinet, Department of Computer Science, KU Leuven
Celestijnenlaan 200A, 3001 Leuven, Belgium

Abstract

In recent years numerous researchers have proposed a wide variety of approaches to incorporate security concerns into software design. Unfortunately a systematic literature review (SLR) providing a detailed overview of the state of the art and defining interesting research opportunities is lacking. This creates an extra barrier for (new) researchers to enter the domain and contribute to it. We describe a procedure for an SLR aimed at minimizing this barrier. By providing this procedure we first hope to receive feedback on it and trigger a discussion. Second, the availability of this procedure is useful when updating the SLR with approaches that will emerge after its initial performance.

1 Introduction

Students starting a PhD must typically overcome two challenges before they can start contributing to a particular domain. On one hand they must define their own "niche" within the domain. On the other hand they must establish a good understanding of the state of the art. These challenges are most often tackled by studying the available literature. A systematic literature review (SLR), introduced into software engineering by Kitchenham and Charters [KC07], is a structured manner for executing this task.

We are performing an SLR concerning the incorporation of security concerns in software design. This domain has grown rapidly in recent years, with numerous researchers proposing a wide variety of approaches. Unfortunately these approaches are developed mostly independent from each other and focus on different security properties. This results in a complex tangle of different approaches, creating an extra barrier for (new) researchers to enter the domain and contribute to it.

The first objective of this SLR is to untangle the domain by providing a detailed overview of the current state of the art, useful to both new and experienced researchers. Second we aim at discovering the gaps in current research and thus define interesting research opportunities. In this paper we describe in detail the procedure defined for the SLR. However, the actual results of the SLR are not discussed here. By providing this procedure we first hope to receive feedback on it and trigger a discussion. Second, the availability of this procedure allows everyone to continuously update the SLR with approaches that will emerge after its initial performance.

The remainder of this paper is organized as follows. Section 2 discusses in detail the procedure of the SLR. Section 3 shortly describes related work. Section 4 concludes the paper.

In: A. Editor, B. Coeditor (eds.): Proceedings of the XYZ Workshop, Location, Country, DD-MMM-YYYY, published at http://ceur-ws.org

2 Systematic Literature Review

This section introduces the procedure we defined for the SLR. First, we describe the research questions and discuss how they relate to our goals. Second, we describe the criteria for scoping the relevant research works. Third, we describe the strategy to retrieve the relevant research works. Fourth, we describe how the research works are analyzed to provide answers to the posed research questions.

2.1 Research Questions

We define four main research questions (RQ's), shown below, for the SLR.

RQ1: What security properties are supported during software design?

RQ2: Is a representation of the security properties supported?

RQ3: Is an analysis of the security properties supported?

RQ3.1: Is the supported analysis precise?

RQ3.2: Is the supported analysis white hat or black hat?

RQ4: What evaluation is provided for the proposed approach?

To avoid ambiguity some terms in these research questions require a more precise description. *Software design* refers to both architectural and detailed design. *Security properties* includes properties such as integrity and logging. The full set of properties defined for the SLR is discussed later. *Supporting* a security property means providing the possibility to represent and/or analyze it. *Representing* a security property covers every explicit representation, graphical or textual, of a security property. *Analyzing* a security property means verifying whether it is correctly enforced according to the system's security policy, which is the collection of all security requirements. An analysis is considered *precise* if a developer can algorithmically perform it by following the steps in its description. Note that whether an analysis method is precise or not is irrelevant of any tool support.

The first three research questions allow to construct an overview of the state of the art. Furthermore, they allow to discover research opportunities concerning security properties that are not or barely addressed. The fourth research question uncovers approaches lacking evaluation, which are opportunities for empirical research.

2.2 Inclusion and Exclusion Criteria

The inclusion and exclusion criteria of an SLR delimit which research works are considered relevant. Papers are **included** if they adhere to at least one of the following criteria:

- The paper represents security properties in software design or
- analyzes security properties in software design or
- models attacks or threats in software design or
- evaluates a paper included by a previous criterion.

Papers are **excluded** if they adhere to at least one of the following criteria:

- The paper only mentions security as a general introductory term or
- is not available as a full version, only an extended abstract or presentation, or
- is a duplicate of an included paper or
- is superseded by an included paper or
- is published more then ten years ago.

If duplicate papers are encountered only the most recent or most extensive version is included. We defined a scope of ten years because approaches described in older papers either have been developed further, and are thus included through later papers, or have most likely become outdated for current technology.

The inclusion or exclusion of a paper is decided in two phases. First, during the *initial selection phase* the abstract, title and keywords of a paper are evaluated against above criteria. In case of doubt the conclusion of the paper is also consulted. Second, during the *final selection phase* the included papers are fully read and their inclusion is re-evaluated. Each paper is evaluated by one participant of the SLR while another participant validates this evaluation. If participants disagree on the inclusion or exclusion of a paper consensus should be reached through a discussion between all participants.

2.3 Search Strategy

A search strategy defines how to retrieve relevant research works. In order to achieve maximal coverage our search strategy consists of three complementary methods: a digital library search, a manual search and snowballing.

First, we searched **digital libraries** by means of a search string containing relevant terms. Table 1 contains an overview of the selected digital libraries. In our opinion this selection covers a sufficiently large amount of the defined domain and including more libraries would result in too much overhead. Furthermore the selected libraries provide extensive search functionality, allowing complex search strings. We considered using Google Scholar but due to technical limitations it was not included.

Table 1: Digital Libraries targeted with a search string

Library	Website
ACM	`https://dl.acm.org/`
CiteSeerX	`http://citeseerx.ist.psu.edu`
IEEExplore	`http://ieeexplore.ieee.org`
Springerlink	`http://link.springer.com`
ISI Web of Science	`http://apps.webofknowledge.com`
Compendex	`http://www.engineeringvillage2.org/`

An initial search string was constructed by selecting terms from a manually composed set of highly relevant papers. This search string was fine-tuned to reduce the number of irrelevant papers using the top 100 results of ACM and CiteSeerX. To avoid an explosion in the number of results the search string is only queried over the abstract, keywords and title. Due to space constraints we do not describe the final search string here[1]

Second, we performed a **manual search** of papers published in relevant conferences and journals. Tables 2 and 3 contain an overview of respectively the selected conferences and journals. In our opinion this is a representative selection of venues for the defined research domain.

Table 2: Conferences selected to be manually searched

Name	Acronym
European Conference on Object-Oriented Programming	ECCOOP
International Symposium on Engineering Secure Software and Systems	ESSoS
International Conference on Software Engineering	ICSE
International Symposium on Architecting Critical Systems	ISARCS
International Conference on Model Driven Engineering Languages and Systems	MODELS
Working IEEE/IFIP Conference on Software Architecture	WICSA
European Conference on Software Architecture	ECSA

Third, we performed the so-called **snowballing** method, both forward and backward, on included papers.

[1]More information can be found at `https://people.cs.kuleuven.be/alexander.vandenberghe/search-string.html`.

Table 3: Journals selected to be manually searched

Name	Acronym
Journal of Systems and Software	JSS
Software and Systems Modeling	SoSyM
Transactions of Software Engineering	TSE

2.4 Quality Assessment, Data Extraction and Synthesis

The research questions are answered by analyzing the included approaches[2]. First, the **quality assessment** uses a quality questionnaire, shown below, to score each approach. Allowing one to rank the approaches relative to each other.

1. How many papers are published for the approach?
2. How is the approach evaluated?
   - Industrial case study
   - Researcher or student case study
   - Toy example
3. How much tool support is available for the approach?
   - Full tool support
   - Partial tool support
   - No tool support

For the first question an approach is awarded one point per paper included in the SLR. For the second question an industrial case study awards two points, a researcher or student case study awards one point and a toy example awards half a point. Points are awarded per unique instance of an evaluation. If different papers describe the same evaluation points are awarded only once. If for example multiple papers for one approach describe the same toy example only half a point is awarded. For the third question full tool support awards one point whereas partial tool support awards half a point. Full support means one or more tools support the creation of all notational elements provided by the approach and its analysis method can be performed without user intervention. If only part of the approach is supported by one or more tools (e.g., a modeling tool is available but analysis must be performed manually) it is considered to have partial tool support.

Second, **data extraction** is performed using a taxonomy we defined for the SLR. Each approach is classified over three dimensions: security, software engineering and evaluation.

The *security dimension*, shown in Figure 1, provides essential data concerning the first three research questions. Besides this data any other security artifacts (e.g., test data) constructed by an approach can also be listed.

The *software engineering dimension*, shown in Figure 2, allows to situate approaches within the development process as a whole. Furthermore it allows comparing approaches based on their applicability in different development phases or domains and thus further elaborates the intended overview.

The *evaluation dimension*, shown in Figure 3, provides data for the fourth research question. This dimension allows to compare both the evaluations for one approach as well as the evaluation for different approaches.

Third, **data synthesis** summarizes the data obtained during data extraction to provide answers to the posed research questions. The first three research questions can be answered by tabulating the data extracted for the security and software engineering dimension. The fourth research question can be answered by tabulating the data extracted for the evaluation dimension.

[2]Since we want to analyze each approach as a whole, we group all papers concerning one approach together instead of analyzing each paper individually.

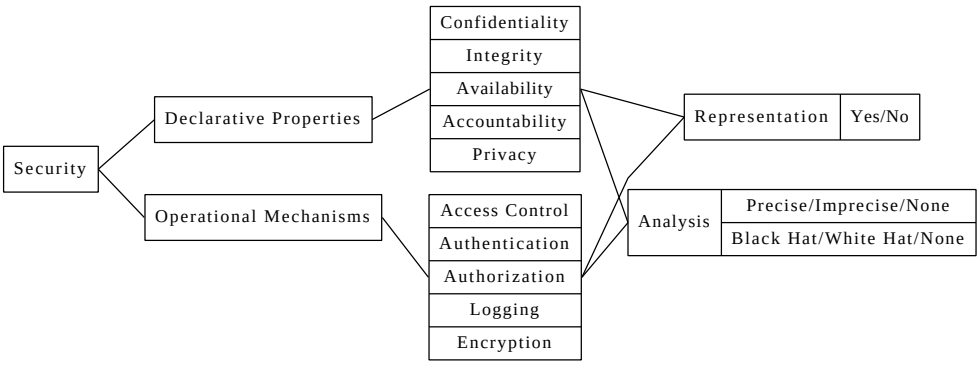

Figure 1: The security dimension classifies each approach based on which security properties it supports and how these properties are supported.

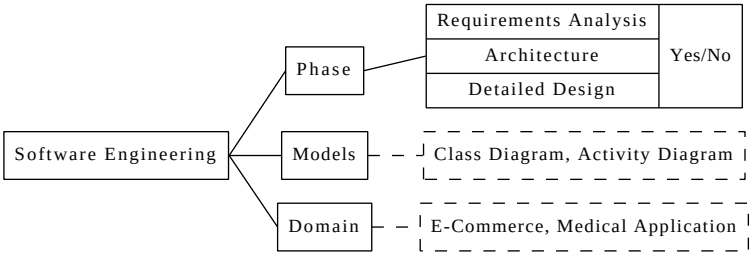

Figure 2: The software engineering dimension classifies each approach based on supported development phases, models and domains. Dashed rectangles illustrate example values.

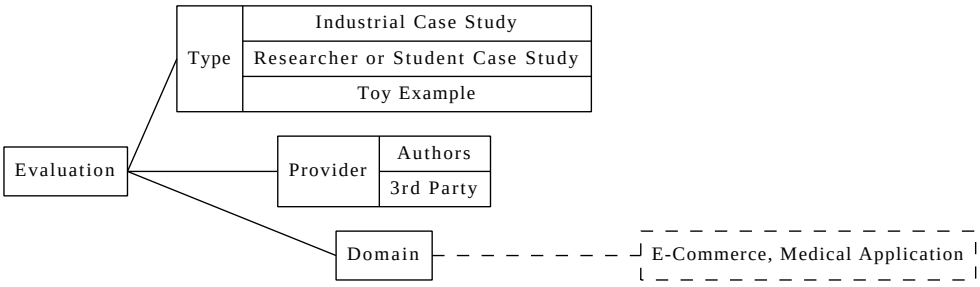

Figure 3: The evaluation dimension classifies each evaluation for an approach based on its type, who provides it and in which domain it is situated.

3 Related Work

In recent years studies comparable to the SLR proposed in this paper have been performed. These studies can be divided into three categories: SLR's, comparisons and surveys.

First, the **SLR** category contains studies following the guidelines by Kitchenham and Charters. Jensen and Jaatun [JJ11] study security in model-driven development. Due to their focus on code generation the scope of the SLR is rather narrow. Furthermore, no explicit comparison of included approaches is provided.

Second, the **comparison** category contains studies only comparing selected approaches. The conclusions from such studies are useful but provide little information for the domain as a whole. Matulevičius and Dumas [MD10] compare two approaches for their applicability to role-based access control.

Third, the **survey** category contains studies providing an overview of the domain, with optionally a comparison. These studies are not performed in a systematic manner making them difficult to update.

Dehlinger and Subramanian [DS06] survey aspect-oriented approaches for designing and implementing secure software. The included approaches are only individually evaluated without comparison between them. Jayaram and Mathur [JM05] cover a broader scope by surveying all types of approaches but focus mainly on the requirements phase. The authors provide no comparison and only general possible research directions.

Villarroel et al. [VFMP05] not only provide an overview but also compare the surveyed approaches. The authors use Khwaja and Urban's [KU02] comparison framework, which lacks security-specific criteria. The resulting comparison thus does not provide an adequate overview of security in software design. Kasal et al. [KHN11] solve this problem by defining their own evaluation taxonomy, inspired by Khwaja and Urban's framework. The authors define, among others, formality and security mechanisms as evaluation dimensions.

Dai and Cooper [DC07] evaluate and compare approaches based on supported security properties, used modeling notations, analysis support and examples. But they do not explicitly define their evaluation taxonomy.

4 Conclusion

In this paper we described in detail a procedure for a systematic literature review (SLR) concerning the incorporation of secure concerns in software design. With such an SLR we aim to provide a detailed overview of the state of the art and define interesting research opportunities. By making the procedure available we first hope to receive feedback and trigger a discussion. Second, the availability of this procedure allows everyone to continuously update the SLR with approaches that will emerge after its initial performance. Hopefully resulting in the continuous availability of an up to date SLR aiding researchers in entering and contributing to the domain.

Acknowledgment

This research is partially funded by the Research Fund KU Leuven, and by the EU FP7 project NESSoS. With the financial support from the Prevention of and Fight against Crime Programme of the European Union (B-CCENTRE).

References

[DC07] L. Dai and K. Cooper. A Survey of Modeling and Analysis Approaches for Architecting Secure Software Systems. *International Journal of Network Security*, 5(2):187–198, 2007.

[DS06] J. Dehlinger and N. Subramanian. Architecting Secure Software Systems Using an Aspect-Oriented Approach: A Survey of Current Research. Technical report, Iowa State University, 2006.

[JJ11] J. Jensen and M.G. Jaatun. Security in Model Driven Development: A Survey. In *Availability, Reliability and Security (ARES), 2011 Sixth International Conference on*, pages 704 –709, aug. 2011.

[JM05] K. R. Jayaram and Aditya P. Mathur. Software engineering for secure software - state of the art: a survey. Technical Report CERIAS 2005-67, Purdue University, 2005.

[KC07] Barbara Kitchenham and Stuart Charters. Guidelines for performing Systematic Literature Reviews in Software Engineering. Technical Report EBSE 2007-001, Keele University and Durham University Joint Report, 2007.

[KHN11] K. Kasal, J. Heurix, and T. Neubauer. Model-driven development meets security: An evaluation of current approaches. In *System Sciences (HICSS), 2011 44th Hawaii International Conference on*, pages 1–9, jan. 2011.

[KU02] Amir A. Khwaja and Joseph E. Urban. A Synthesis of Evaluation Criteria for Software Specifications and Specification Techniques. *International Journal of Software Engineering and Knowledge Engineering*, 12(5):581–599, 2002.

[MD10] R. Matulevičius and M. Dumas. A Comparison of SecureUML and UMLsec for Role-Based Access Control. In *Databases and Information Systems*, pages 171 – 185, 2010.

[VFMP05] R. Villarroel, E. Fernández-Medina, and M. Piattini. Secure information systems development a survey and comparison. *Computers & Security*, 24(4):308 – 321, 2005.