Software Development Initiatives to Identify and Mitigate Security Threats – Two Systematic Mapping Studies

Paulina Silva

Universidad Técnica Federico Santa María, Chile
pasilva@alumnos.inf.utfsm.cl

and

René Noël

Universidad de Valparaíso, Chile.
rene.noel@uv.cl

and

Santiago Matalonga

Universidad ORT Uruguay, Uruguay
smatalonga@uni.ort.edu.uy

and

Hernán Astudillo, Diego Gatica, Gastón Marquez

Universidad Técnica Federico Santa María, Chile
hernan@inf.utfsm.cl, diego.gaticap@alumnos.uv.cl,
gmarquez@alumnos.inf.utfsm.cl.

Abstract

Software Security and development experts have addressed the problem of building secure software systems. There are several processes and initiatives to achieve secure software systems. However, most of these lack empirical evidence of its application and impact in building secure software systems. Two systematic mapping studies (SM) have been conducted to cover the existent initiatives for identification and mitigation of security threats. The SMs created were executed in two steps, first in 2015 July, and complemented through a backward snowballing in 2016 July. Integrated results of these two SM studies show a total of 30 relevant sources were identified; 17 different initiatives covering threats identification and 14 covering the mitigation of threats were found. All the initiatives were associated to at least one activity of the Software Development Lifecycle (SDLC); while 6 showed signs of being applied in industrial settings, only 3 initiatives presented experimental evidence of its results through controlled experiments, some of the other selected studies presented case studies or proposals.

Keywords: Security Threats, Secure Software Development, Systematic Mapping study

1 Introduction

Security is a topic of interest in Information Technologies nowadays, and is considered a Quality Attribute of a System by software engineering (ISO 25010). From this point of view, several initiatives, such as techniques, methodologies, methods and tools, have been proposed in order to add architectural and design decisions throughout the Software Development Life Cycle (SDLC) [1]. Some of these initiatives

aim to help software engineers in the identification of security threats such as Attack Trees [2], elicitation of security requirements with Misuse Cases [3], and definition of security goals, as in Secure Tropos [4]. Other initiatives allow software engineers to mitigate these security threats or satisfy security requirements by reusing security design decisions [5, 6] or supporting the decision making process [7]. However, the impact of these techniques, methods and models is hard to assess, due to the nature of software development process, the lack of standards to measure software security from the point of view of the software developer, and the difficulty of defining security requirements in an effective, clear and unambiguous way [8]. Also, the application of these proposals could be hard to replicate in real contexts, due to the dependence on the software process activities, artifacts and practices required as preconditions, which may vary between software development organizations or even between projects in the same organization.

The goal of this study is to identify these initiatives, characterize them by level of empirical evidence of their application, identify their use of software artifacts or notations, and identify if each one supports at least one stage of the SDLC. To achieve this goal, we have conducted a systematic mapping study (SM) [9, 10]. A SM study is a defined method to build a classification scheme and structure of a software engineering field of interest. As our initial motivation was to find long term research initiatives, we first focused in studies published in scientific journals. We have called this approach **Main Systematic Mapping** study or **mSM**. For this special issue, the **mSM** was complemented with a Backward Snowballing process, [11] which will be called **Backward Snowballing Systematic Mapping** study or **bsSM**. The backward snowballing technique allowed us to achieve two goals: (1) extend the search base of our study, covering more articles and initiatives, and (2) give traceability of the evolution of the initiatives identified in the first mapping, from merely theoretical proposals without evidence of its application to initiatives that are experimentally tested in industrial settings. We performed the systematic mapping process for the initial mapping and the backwards snowballing in separated processes, but the analysis and conclusions presented the integrated results of both mappings.

Integrated results of these two SM studies show a total of 30 relevant sources were identified; 17 different initiatives covering threats identification and 14 covering the mitigation of threats were found. All the initiatives were associated to at least one activity of the Software Development Lifecycle (SDLC); while 6 showed signs of being applied in industrial settings, only 3 initiatives presented experimental evidence of its results through controlled experiments, some of the other selected studies presented case studies or proposals.

The rest of the paper goes as follows: Section 2 presents the related work, reporting previous work of authors in software security design studies, relevant systematic mapping studies that support our approach to answer our research questions, and some proposals of techniques, methods and tools that initially guided our search. Section 3 presents the design of a systematic mapping study and remarks the differences between **mSM** and **bsSM**. Section 4 presents details of the execution and results. Section 5 reviews Threats to validity. Finally, section 6 presents main conclusions and high level answers to our research questions.

2 Related Work

We conducted several studies aiming to explore software design decisions for developing secure software systems: An experimental approach was initially taken to compare software security tactics and security design patterns [12, 13], and several questions arose when trying to operationalize the identification and mitigation of security threats. These questions guided the authors to search for some security threat identification methods with demonstrated efficiency to be added to the experimental process, as a previous step to apply security tactics or patterns. In a first search based on a survey of techniques presented by Uzunov et al. [1] and a snowballing through its main references, we did not find methods with scientific evidence of its impact were found; this motivated both systematic mapping studies. We chose a well explained methodology for a first experimental approach to threats identification: Misuse Activities [14]. The results of applying this identification method can be found in [13], which show statistically significant differences when performing a guided identification process versus an ad-hoc approach. We also have on going research initiatives to explore the importance of software artifacts over security decisions [15] and the structure of the Architectural Tactics for security decisions [16].

In terms of related initiatives of identifying security methodologies and techniques, other authors have described several methodologies, as mentioned in [1]. Nonetheless, the study [17] which analyzed the security expert's opinion about the State-of-Practice of initiatives in the software development process, revealed that a lot of these initiatives do not surpass the State-of-Art even if they are considered good practices. The study established requirements for a methodology for building secure systems, including

those proposals must consider the compatibility of the technique with the commercial environment of developers and their current skills.

A software engineering systematic mapping study (SM) is a defined method to build a classification scheme and a structure for a software engineering field of interest. When working with SM studies, it is difficult not to relate them to Systematic Literature Reviews (SR or SLR). The main difference between them lies, according to [9, 10], in the definition of goals, breadth, validity issues and implications of the studies. Also, a SLR is more ambitious and rigorous than a SM. [18] presents a series of rigorous guidelines to make a systematic mapping study are presented, this directives constitute a well defined method for reviewing systematic mapping, with potential to match the rigor of a SLR.

The study in [19] presents an example of the application of SM to answer research questions similar to the motivations of this study: a SM is done with the goal of finding information about usability evaluation methods, that have been used by researchers to evaluate web applications and their relationship with the web development process. In [19], the main results are identification of most reported methods and research gaps aiming to the validations of these methods. The study in [20] presents a work in which a systematic revision is done with the goal of finding a trend in the definition of quality in the Model Driven Engineering. Main contribution of the study is the identification of a variety of multiple quality interpretations as consequence of the diversity in MDE compliance works.

3 Methodology/Research Design

The aim of this research is to identify the existent initiatives and experience reports for identification and mitigation of security threats during software development in order to build secure systems. An **initiative** is a framework, method, technique, proposal or methodology, among others. No comparison or evaluation is done between studies, so we could classify this work as a **systematic mapping study** (SM). In a SM we do not consider specific outcomes or experimental designs in the study, since the aim is searching for a broad overview of the research area as a whole [21]; which is why this study is not a systematic review. In order to design and operate the study, we followed the guidelines proposed by Petersen [9, 10] and documented the protocol following Biolchini et al. definitions [22].

In contrast with other systematic mapping studies, where digital libraries are queried using different design engines such as IEEEExplore, ACM, Scopus and others, we decided to obtain our sources from Journals for two main reasons: (1) we were looking for results of long term research from papers describing studies that consolidate mature initiatives and (2) the heterogeneity of the search engines and limitation in the length of the search string.

Excluding sections 3.5 and 3.6, all the following sections of this paper should be considered for both studies, the **mSM** and **bsSM**. In section 3.8, we made some improvements for the inclusion/exclusion criteria (section 3.5) and search process (section 3.6) that was used on the **bsSM** study.

3.1 Research Questions

No systematic mapping studies or reviews were found when searching for initiatives for security threats identification and mitigation. Based on the main objective, we elaborated the following research questions for both systematic mapping studies:

1. What are the existing initiatives or methods of identifying security threats and attacks?
2. What threat identification methods are related to Security Tactics?
3. What other Mitigation Methodologies exist when developing Software, to conceive Secure Systems?

The first question is the main research question, while the second and third are secondary research questions.

3.2 Identification of Population and Intervention

We used the P.I.C.O. [23] structure to define the search string; that is to define a search string according to **Population**, **Intervention**, **Comparison** and **Output**. However, since our work is a systematic mapping study we follow the considerations presented in [9, 10], that suggest only the use of Population and Intervention. A systematic mapping has a different kind of depth in contrast to a systematic literature review (SLR).

- Population: Software Development or Design related work
Keywords: “software”, “design”, “engineering”.
- Intervention Control: Security threat identification or mitigation methodologies /initiatives/guidelines
Keywords: “security”, “threat”, “attack”, “identificate”, “mitigate”, “minimize”, “elicitate”, “elicitation”.
Keywords: “model”, “guideline”, “checklist”, “method” and “methodology”.

We performed several test executions of the search string, looking for evident outliers that could make the filtering stages unnecessary difficult, and we found that some words could extend our search to other fields of knowledge. We excluded the following terms in order to find papers related to our area of interest:

- “System”: It is a word that encompasses a lot of research areas.
- Use of “Service” or “System” for population: returns results from other areas, which are of no use to this work.
- Plural words were not used in the search string.
- Complete words were not used, we use the prefixes of them.

3.3 Construction of a Search String

Table 1: Keywords used in the Search String

Term	keywords obtained and its synonyms
P	(“software” OR “design” OR “engineer” OR “develop”)
I	(“securit” OR “privacy” OR “integrity” OR “confidential” OR “availabil” OR “accountabil”) AND (“threat” OR “risk” OR “attack” OR “requirement” OR “vulnerabil”) AND (“identif” OR “mitig” OR “minimiz” OR “elicit” OR “enumer” OR “review” OR “assur”) AND (“model” OR “metric” OR “guideline” OR “checklist” OR “template” OR “approach” OR “strateg” OR “method” OR “methodolog” OR “tool” OR “technique” OR “heuristic”)
String	(P AND I)

The initial keywords were identified by asking our experts, and synonyms were added to the keywords already found. The keywords that represent the core of the research are: “Threat”, “Attack”, “Identify” and “Mitigate”. As we mentioned in section 3.2, we used the prefixes of a word as a consequence of our search procedure’s definition. This is further discussed in section 3.6.

3.4 Source Selection for both systematic mapping studies

Table 2: Source Selection

Journal Name	Link
Journal Computers & Security	www.journals.elsevier.com/computers-and-security/
Journal Information and Software Technology	http://www.journals.elsevier.com/information-and-software-technology/
Journal of Systems and Software	http://www.journals.elsevier.com/journal-of-systems-and-software/
Journal of Universal Computer Science	www.jucs.org/
Journal of System Architecture	http://www.journals.elsevier.com/journal-of-systems-architecture/
Requirements Engineering	http://link.springer.com/journal/766
IEEE Transaction on Software Engineering	http://www.computer.org/csdl/trans/ts/index.html
IEEE Transactions on Dependable and Secure Computing	http://www.computer.org/web/tdsc
International Journal of Secure Software Engineering	http://www.igi-global.com/journal/international-journal-secure-software-engineering/1159
Information Security Technical Report	http://www.sciencedirect.com/science/journal/13634127
Journal of Information Security and Applications	http://www.journals.elsevier.com/journal-of-information-security-and-applications/

All source papers were be available on the web, through open access or subscription based journals (those we could get access to). The language selected for the studies was English. The list of journals was proposed and reviewed by four researchers from different domains (software engineering, software security, distributed systems and software architecture). The Source Identification list is shown in Table 2. In section 4.2, the references of the selected papers from the **mSM** study were used as the source for the **bsSM** study.

3.5 Inclusion/Exclusion Criteria, only for the main systematic mapping study

Table 3: Inclusion/Exclusion Criteria

Inclusion Criteria	Exclusion Criteria
Studies from [2000 to 2015[	Being published before 2000 (]- to 2000[)
The paper mentions the Quality Attribute “Security” in some part of the work	Non-related Security article. Methodological proposals where security is not specified
Empirical or Experimental studies or Proposals (Methodological Definitions, Applied Case Studies or Experimental Application of Identification and Mitigation). Is there a concern to formalize? No matter if it was successful or not. Informal technical reports are useful (if it was wrote by someone of field).	Non Empirical or Experimental studies. Complete proceedings or books without specifying the articles. Exclude Thesis if it is not the published version.
Applicable techniques during Software Development/Conception	Techniques used when a System is already deployed. Techniques related to certificate or security modeling or risk assessment. Articles not related to development or assurance of systems, for example, curriculum proposals or Security Quality Assurance

For the **mSM** we used table 3 to select or reject papers in the search procedure (section 3.6). The Study Type selected consists of articles related to secure software development, and articles selected by the Inclusion Criteria (Table 3). Studies with at least one Exclusion criterion were not selected.

Two reviewers evaluated each study. We used the following acceptance criteria in case the reviewers did not agree which studies should be accepted.

- If the two readers accept: The study is included.
- If one reader accepts and one is in doubt: The study will be discussed in group.
- If the two readers excluded the study: The study is not included.
- All conflicts are reviewed by a third researcher.

To help resolve the disagreements, reviewers were asked to document the rationale for including/excluding each article.

3.6 Search Strategy/Procedure for the main systematic mapping study

Our search strategy was to explore software development and software security journals, in order to cover studies reporting long term research in initiatives and accumulated scientific evidence.

1. We searched through studies from digital Journals. We selected the following information: Volume, Issue, URL (for the Journal-Volume), Title and Abstract.
2. All this information was saved in an Excel file, one file for each Journal.
3. From the Excel file we used a filter created by a Macro made in Visual Basic.
4. **First Filter:** This was done by searching with the **Search String** (in the form of a Visual Basic macro) within the Abstract, and selecting papers that contained terms from the string. The Search String was created with the terms identifying the Population and Intervention. We manually evaluated the strength of the filter with one of the Journals used (IJSSE).

5. **Second Filter:** By reading again the title and abstract, we filtered against the Inclusion and Exclusion Criteria (Table 3).
6. If any irrelevant or duplicate study appeared, they were removed.
7. **Third Filter:** From the second filter, we did a quick scan of the articles and tried to categorize them in the following cases: empirical evidence/experimentation, case study, proposal or systematic review. In this high level revision of the accepted papers, we search within the Abstract, Introduction, Development of the work, Data obtained, Conclusions and other relevant sections of the paper. We considered that the abstract was not enough to judge a paper.
8. **Fourth Filter:** This was done by reading the complete text of the studies that were accepted by the third filtering. The fourth filtering is executed according to the Inclusion and Exclusion Criteria (Table 3). In this step, we tried to take some information that could help to speed up the Data Extraction.

3.7 Search Strategy/Procedure for obtaining studies for the backward snowballing systematic mapping

The following process was used to identify relevant sources from the backwards snowballing process.

1. All the references from papers selected from the **mSM** study, were put together in a excel file. A researcher read the title of every reference in the file and filtered them against the following criteria: "security" AND "software engineering".
2. We checked and erased any duplicate.
3. The search procedure continued as stated in section 3.6, with the improvements of section 3.8.

3.8 Improvements for the search process in the backward snowballing systematic mapping study

To accommodate the differences in search types from the main and backward snowballing SMs search processes, the following changes were made to search and select the relevant sources.

- Three researchers were in charge of revising the studies in the search procedure, one of them was the main reviewer and checked the studies accepted by the other two researchers. If the main reviewer found rejected studies that should had been accepted, the papers were included in the next filter. The idea of this change is to regard the text used to filter the study. Since the abstract could be short, it is better to include a paper in the next filter until it can be read fully to be sure and decide whether to reject it or accept it.
- The recollection of the studies was done manually; therefore it was necessary to erase any duplicate studies in every step of the search.
- For the **second filter**, the studies were divided between two reviewers; they used the same inclusion and exclusion criteria to filter each study. The main reviewer, had to filter and compare all the studies against the second reviewer decision to include or reject studies.
- The **third filter** did not characterize the studies anymore, this was done on the **forth filter** while extracting the information; this way the search procedure was faster, without any loss of studies.

Figure 1 shows how we proceeded for both systematic mapping studies. The output obtained from the **mSM** was the input for the new search done in the **bsSM**.

4 Conducting the Systematic Mapping

4.1 Results from main systematic mapping (mSM)

The trial was done on 27-07-15 according to all the previous sections, the results obtained are presented in table 4. To obtain such results, we did the first phase filtering as explained in section 3.6. No duplicate articles were found. Table 5 summarizes the quantity of articles selected in each phase. In the first phase filtering, the identification of the papers was performed in each Excel of the Journal selected, a total

Figure 1: Process for both systematic mapping studies.

Table 4: Selected papers for the main systematic mapping study (27-07-15)

Journal	Number of Articles Found
Journal Computers & Security	29
Journal Information and Software Technology	19
Journal of Systems and Software	14
Journal of Universal Computer Science	6
Journal of System Architecture	6
Requirements Engineering	12
IEEE Transaction on Software Engineering	8
IEEE Transactions on Dependable and Secure Computing	10
International Journal of Secure Software Engineering	14
Information Security Technical Report	6
Journal of Information Security and Applications	3
Total	127

of 127 papers were identified by filtering its abstract with the Search String. In the second phase, 41 papers were filtered by the inclusion/exclusion criteria, and 0 were duplicated. After discussing between the reviewers, we decided to add 1 more article, so a total of 42 papers went to the third phase filtering. The third filter was done by scanning 15 studies, which remained in the list after fully reading them and extracting the necessary information.

Table 5: Amount of papers selected in each phase of the mSM study

Stage	Number of Articles
First Filter: In Abstract by Search String	127
Second Filter: Articles selected by inclusion criteria	41
After discussion between reviewers	42
Duplicates	0
Third Filter: Lightly reading to other sections of the paper	15
Four Filter: Articles kept after full reading and Extraction	15

In this new version of the work, we found that one paper obtained in this **mSM** study passed the filters without contributing anything to answer the research questions established. In this new version, instead of considering 15 papers we only took 14 papers; this is reflected on the data extraction section 4.3.1.

4.2 Results from backward snowballing systematic mapping study (bsSM)

To complement the identification and mitigate a possible threat of missing literature (see section 5), we performed a backwards snowballing process from the references of our **mSM** results (See 1). The initial set of sources are described in table 6.

Table 6: Amount of number of papers selected in each phase of the bsSM

Stage	Number of Articles
Extracting references from selected paper in the mSM study	794
Filter by Title	308
First Filter: Filter Abstract by Search String	38
Second Filter: Articles selected by inclusion/exclusion criteria over title and abstract	31
Third Filter: Articles selected by inclusion/exclusion criteria over some sections of the paper	23
Four Filter: Articles kept after full reading and Extraction	16

This trial started on 06-06-16 with the references obtained from the selected papers of the **mSM** study. Each reference was filtered by its title against the criteria "security" and "software engineering", which produced 308 results. After this filter, the search process began including the improvements to the search and selection criteria (inclusion and exclusion). The first filter accepts 38 papers whenever the search string was true for the abstract; the same Visual Basic macro for the **mSM** was used for this new systematic mapping. The second filter reduced the list to 31 and the third filter to 23. The last filter,

the forth, needed the full reading of the paper, or at least until the extraction process of the data was sufficient.

4.3 Data Extraction for the two searches

4.3.1 Data Extraction for the main systematic mapping study.

The 14 papers selected by our **mSM** are described briefly:

- CAIRIS [24]: Create security models that are automatically generated from specified, declarative data rather than via direct manipulation.
- Threat-Driven Modeling of Secure Software [7]: Use aspect-oriented extension to Predicate/Transition (PrT) nets, as a rigorous formalism for threat-driven modeling and verification.
- Aspect-Oriented Risk-Driven Development [25]: Create system functional models, using the UML 2.0 and perform a risk assessment of the system.
- SIREN (Simple REuse of software requiremeNts) [26]: In order to use the security profile properly, the Elicitation phase must start by studying the threats to the assets of the system.
- Misuse Case [3]: Identify critical assets in the system. Define security goals for each asset. Identify threats. Identify and analyze risks. Define security requirements for the threats to match risks and protection costs, preferably aided by a taxonomy of security requirements.
- SQUARE[27]: SQUARE consists of nine steps: agree on Definitions, identify security goals, develop artifacts to support security requirements definitions, perform risk assessment, select requirements elicitation technique, elicit the security requirements, categorize the security requirements, prioritize the security requirements, and inspect the security requirements.
- Secure Tropos and PriS [28]: The process consists of three iterative activities: organizational analysis, Security and Private Requirements Analysis, and selection of deployment model.
- ModelSec [29]: ModelSec proposes a generative architecture based on a chain of model transformations involving several security models at different levels of abstraction.
- Secure Tropos [30]: Supports development of IS (information systems) through four phases: early and late requirements, architectural and detailed design. Language application includes three major stages.
- SPEM, Secure Tropos and PriS [31]: Three main activities of the framework: the Security and Privacy Cataloguing, the Security and Privacy Analysis, and the Selection of Cloud Service Provider. The proposed framework consists of a language and a process that is focused on the requirements engineering stage.
- Attack Trees and Misuse Cases [32]: A pair of controlled experiments was performed. In each experiment, the participants used the two techniques individually on two different threat identification tasks.
- SEDAWA: Secure Engineering process for Data Warehouses [33]: The secure engineering process is proposed with the purpose of defining secure requirements and transforming them in order to develop secure Data Warehouses.
- Attack tres and Misuse case [34]: The participants were randomly assigned to one of four experiment groups.
- Methontology [35]: The life cycle activities of Methontology are composed of specification, conceptualization, formalization, implementation and maintenance activities.

4.3.2 Data Extraction, for the backward snowballing systematic mapping study.

Using the **mSM** references as input, we obtained 16 papers in this new systematic mapping. We present a briefly description of each of them:

- Extended Misuse Case [36]: Proposal to extend the format of a misuse case, to detect vulnerabilities and insider threats of a system when some of them can not be determined before design.
- Abuse case Based Assurance arguments [37]: Paper describes an extension to abuse-case based security requirements analysis. The approach is lightweight and is adaptable to the software development process.
- Threat-driven architectural design [38]: There is a large gap between obtaining threats and software design. To mitigate vulnerabilities they propose an approach to reduce and minimize the gap between analysis and architectural design.
- Reuse-based approach [39]: Provides a reuse-based methodology for misuse case analysis and subsequent specification of security requirements.
- Risk Management [40]: This guide provides a foundation for the development of an effective risk management program, containing both the definitions and the practical guidance necessary for assessing and mitigating risks identified within IT systems.
- Security Quality Requirements Engineering (SQUARE) Methodology [41]: SQUARE is presented as a consolidated and validated methodology for development of secure projects in the requirements engineering discipline.
- vulnerability-centric requirements engineering framework [42]: Proposes a framework for requirement engineering to model threats, security countermeasures, attacks. The initiative uses the i* framework modeling language.
- Security Requirements Hybrid Technique [43]: One hybrid technique based on misuse cases and attack trees.
- Misuse Case Maps with Misuse Cases and System Architecture Diagrams [44]: Gives empirical evidence for misuse case maps (combination of misuse case and maps for case use). The authors proposed a mixture of two important techniques in the requirement engineering that has succeeded.
- Misuse Case Maps [45]: Technique based on misuse case and misuse case maps to map requirements into an architecture.
- Anti-goal refinement and attack trees [46]: The framework integrates intentional obstacles (or antigos) set up by attackers to break security goals. Attack trees are derived systematically through anti-goal refinement until leaf nodes are reached, and could possible be used by an attacker.
- Misuse case, attack trees and security activity models [47]: Attempt to make threat models more understandable and available to software developers, and thereby provide easy access to security knowledge that is needed to build more secure software systems.
- Secure Tropos and UMLSec [48]: Since all software systems operate within an environment with stakeholders or users, relevant laws and regulations might influence the security aspects of the system.
- Secure Tropos and Pris [49]: Integration of two software engineering methodologies, one from the security requirements domain, and other from the privacy domain. There is a need to analyze security and privacy separately but under a unified framework is of vital importance.
- Secure Tropos and UMLSec [50]: Integrates a goal-driven security requirements engineering (GDSRE) methodology (called Secure Tropos) with a model-based security engineering (MBSE) approach (called UMLSec) to create a structured methodology for secure software systems development.
- Mal activity diagrams [51]: Technique to identify security threats though activity diagrams.

4.4 Analysis

The results from both systematic mapping studies, gave us a list of 30 papers. Regarding the data obtained from both extractions, we found that proposals or examples were prevalent among other study types, followed by case studies. Only four papers, describing three techniques, had experimental data and analysis: Misuse Case Maps [44, 45], Attack Trees [32] and Misuse Cases [34]. Figure 2 presents the distribution in the evidence types of the selected studies.

Figure 2: Study types

Support more than one development activities

Figure 3: Development activities which supports

Adoption of initiatives in the industry

Figure 4: Adoption of initiatives in the Industry

We were interested in knowing how the papers approached the security of each proposal, whether it was thought as part of the development process or as a software component; within the 30 papers selected, security was considered part of the whole development process. In Figure 3 most of the papers supported more than one development activities in the SDLC, specially the requirement and design phases. Another fact we discovered is that some initiatives found in the selected papers, 11 of 30 papers, seemed to be used within an industrial environment (Figure 4).

Most interesting findings can be summarized in the following lines:

- A Petri-based modeling approach of threats [7] showed the progressive and transversal-intrinsic nature of security threats.
- SQUARE [27] delivered a set of artifacts, threats and vulnerabilities, from which is possible to get an impact analysis, the probability of occurring and risk level. With this it was possible to categorize and prioritize security requirements, which allowed to analyze the security requirements of the software from the beginning.
- Paper [26] defined a repository of reusable generic requirements that minimized the time of developing in subsequent projects when using the practical methodology they proposed. SIREN extended the MAGERIT methodology, in accordance to the ISO 15408 or Common Criteria Framework, and based on IEEE standards.
- One approach worked to prove to the stakeholder that security was implemented in a project [24]. They took into account the “usability” of the software as a part of the integration of stakeholders, to make the security more explicit to them.
- In [40] is argued that effective risk management must be totally integrated into the SDLC. Also, it clarified that a project using it would only succeed depending on who is in charge of assuring the risk management.

Figure 5: Evidence types and time distribution of the selected studies.

- Misuse cases and use cases are best for capturing threats and attacks at the system boundary, but they are less suitable for capturing attacks that take place inside or outside the system [51].
- As can be seen in Figure 5, last decade concentrated most of the studies and experimental or empirical levels of evidence. Unfortunately, most of the studies were only proposals with examples of application, lacking evidence of its application or results.
- Secure Tropos focused on the elicitation and analysis of security requirements while PriS focused specifically on the incorporation of privacy requirements in the system design process [28].

4.5 Answers to Research Questions

In this section we present the answers to the Research Questions in Section 3.1. Table 7 shows the initiatives found in the 30 studies reviewed after the selection process; we collected papers using the same or similar initiative. Research questions are also answered in columns *RQ1* (regarding the first research question about security threat identification initiatives), *RQ2* (regarding the identification methods suitable to work with Architectural Tactics as mitigation method) and *RQ3* (related to the third research question regarding other security threat mitigation methods). The *Adoption* column indicates if there is any evidence of adoption or empirical application of the initiative in an industrial setting, and the *Evidence* column indicates if there is quantitative evidence of the results of the technique.

In addition to the data shown in table 7, we characterized each initiative according to its abstraction level, from high level proposals, such as frameworks and methodologies, to more specific initiatives (methods and techniques). Also, we identified relationships between the proposals: we looked for initiatives that served as foundation or reference in other proposals, or high level initiatives that contained or use more specific ones. In figure 6 we depicted this dependency by a red arrow, and grouped initiatives in 12 sectors defined by level of scientific evidence of the initiatives and their context of application adoption (industrial, academic or theoretical). In addition, Figure 7 shows the initiatives grouped by abstraction level and goal (security threat identification, mitigation or both). These last maps showed that most initiatives have the highest level of abstraction and complexity (methodologies and frameworks), and aim for both identification and mitigation of security threats.

Some comments for each research question are detailed bellow:

1. What are the existing initiatives for identifying security threats and attacks?

Figure 6: Relationship between initiatives and how they are classified (Adoption versus Type of Proposal)

Figure 7: Classification of initiatives: Type of initiatives versus Level of Abstraction.

Table 7: Answers to research questions summary

#	Initiative	RQ1	RQ2	RQ3	Adoption	Evidence
I1	KAOS: Goal-Oriented Methodology [24]	x		x	x	
I2	Secure Tropos [52][4][30][48][49][50]	x			x	
I3	AORDD Methodology [25]	x		x		
I4	SQUARE [27][41]	x			x	
I5	Aspect-Oriented Petri Nets [7]	x		x	x	
I6	Attack Trees [32][34][47][43][33]	x		x	x	x
I7	Misuse cases [32][34][3][36][47][43][38][44]	x		x	x	x
I8	SIREN [26]	x		x		
I9	ModelSec[53]	x				
I10	Methontology [35]			x		
I11	Abuse Case [37]	x		x		
I12	Threat-driven Architecture Design [38]	x	x	x		
I13	Reuse-based approach [39]	x		x		
I14	Risk Management [40]	x		x		
I15	Vulnerability-centric Req. Eng. Framework [42]	x				
I16	Anti-goal Refinement [46]	x				
I17	Mal Activity Diagrams [51]	x		x		
I18	Misuse Case Maps [44][45]	x	x	x		x
I19	Security Activity Models [47]			x		

- According to the results obtained on this work, we can affirm that there are 17 security threat identification initiatives, as detailed in column *RQ1* in Table 7.
- In spite of the results, there is no clear distinction for defining threats and attacks, every article read had at least one different conception of the Threat and Attack concepts.
- From the initiatives found, we understood that all of them take into account security threats analysis within the requirement discipline.
- There are only six initiatives that showed application in industrial environment, and only two of them showed experimentation evidence, as seen in Table 7.
- Finally, we can relate to the work of Whyte and Harrison [17], which states **the lack of implemented initiatives** within the industry, clearly reflected in the results above.

2. What threat identification initiatives are related to Security Tactics?

With the search and extraction done, we could not find anything that explicitly supported Security Tactics. Initiatives marked as related to Tactics in Table 7, were identification techniques that operate at architectural level, which is the same abstraction level as Tactics.

3. What other Mitigation initiatives exist when developing Software, to conceive Secure Systems?

- As showed in table 7, 14 mitigation initiatives were found. Most of them were also identified as security threat identification initiatives, except for [35] and [47].
- In [25] they use AOM (Aspect-Oriented Modeling) methodology and techniques to specify generic security mitigation mechanism models. With a set of these security mechanisms, they performed a trade-off analysis between them to improve the security in the design.
- The view in the Risk analysis model in [24], compressed goal trees arising from risk responses, thereby making it easier to trace risks to mitigating responses, the requirements they treat, and the countermeasures they refine.
- Mitigation on web based attacks [35] was based in the complete understanding of the domain and context of information contents to be processed, and the ability to filter them on the basis of their effects on the target web application, with the help of policies; which is done through the process of reasoning and intelligent decision making.
- [7] used threat mitigation, security features or assurance techniques, for mitigating specific security threats.

- [34], [32], [3] mentioned that a use case can be a countermeasure for a misuse case, in which the first reduces the Misuse Cases chance of succeeding; they are called security use cases.
- [30] used Risk treatment-related concepts or decisions, requirements and controls defined and implemented to mitigate the risks. A risk treatment is the treatment decision for a risk.

5 Threats to Validity/Limitations of this Study

Throughout the execution of the protocol, we took care to reduce the threats that could invalidate the results obtained. We presented the threats identified, and tried to explain what we did to mitigate them (or chose to accept).

5.1 Threat of missing Literature

In the beginning, for the **mSM** study, we were using web search engines to get the conference and journal papers, like IEEEExplore, ACM, Springer, and others. But we decided not to use them because every one of them used different forms to build a search string; which could lead to different kinds of search. Also, we found that some web search platform changed the maximum quantity of keywords, so this gave us a big limitation. We changed our focus to ensure a more stable population, allowing the replication of the study. Likewise, we focused in journals in order to cover studies reporting long term research in initiatives and accumulated scientific evidence. We decided to do a “manual” search within the Journals we knew and had the access to the material needed. To obtain all the articles within a Journal, we built a script¹ based on Python to download all the information needed to identify an article within a Journal (URL, Volume, Title, Abstract, etc.). To obtain the selected articles, we used Excel macros to filter according to the abstract, with the keyword obtained by the Population and Intervention. To mitigate the possible threat within the procedure to extract papers from the Journals and the filtering within the excel list made, we

- Executed the script at least everyday, for 2 weeks, to compare and see if the results obtained had not changed.
- Executed the macro, with our keywords every time we got a new list from the execution of the script.
- Observed that there were no changes.

We may have filtered out some studies when selecting papers in a time interval. We mitigated this by looking the earliest date in which software development processes (and in consequence all sort of initiatives too) started to become popular. UML and RUP were created on the nineties, but started to become popular in the early 2000. For this special issue version, the **bsSM** study was done to mitigate the above threat by inclusion of the backward snowballing process. These new systematic mapping study had as input the references of each study found in the **mSM**, with 794 new sources that were analyzed and filtered according to section 3.7 and 3.8.

5.2 Threat of Selection Bias

The selection of papers during both systematic mapping studies procedure was done by three researchers; two of them performed the selection process by following our guideline in section 3.6. Both researchers applied the inclusion and exclusion criteria, compared results and resolved conflicts following the definitions stated in the research protocol. A third person was in charge of managing the selection and intervening whenever a resolution was difficult to make by the other two. This general guideline was maintained during the revision of this work for the special issue. The same inclusion/exclusion criteria were followed by the same three researchers, therefore there was no new confounding factor in the selection of relevant sources from either the included journals, nor from the snowballing process. This is important, to warranty that the sources can be analyzed together as a set, as we have done so in this version for the special issue.

We need to add that in this special issue, we revised our selected papers from both systematic mapping studies and found one mistake: one study from the **mSM** was found to be of no use. This particular study, passed all the filters but it did not have any information to answer the research questions; therefore it was of no use to the **bsSM** since no reference was selected from it. In this new version of the article, we have deleted that study and fortunately our results were not influenced by this extraction.

¹Python scripts, data-sheets and macros on: <https://github.com/pasilvagh/SM-fondecyt2015>

5.3 Threat of Inaccuracy of data extraction

This threat was mitigated by the iterative nature of the execution of the protocol, the peer review applied with the 2 extractors and a third reviewer in case of conflicts.

6 Conclusions and Discussion

A systematic mapping study was planned and executed, in two iterations, in order to identify initiatives for security threats identification and mitigation.

We covered 11 main software development and security journals, and selected sources published between years 2000 and 2015 (July). From a total of 10.838 articles, 127 were analyzed following the research protocol. We complemented our review by performing a backwards snowballing mapping, applying our selection plan to 794 additional studies. Finally, 30 relevant studies were selected for analysis (14 from the **mSM**, and 16 from the **bsSM**). Of these selected studies, 19 different initiatives were found. Among them, 17 supported security threats identification, and 14 supported the mitigation of threats, with the remarkable observation that only 3 of them presented experimental evidence: Misuse Case Maps [44, 45], Attack Trees [32] and Misuse Cases [34]. From this experimental approaches, two studies reported a comparison of Misuse Case and Attack Trees, as replications of the same experiment, in academic and industrial settings [34, 32]. The studies concluded that Attacks Trees have a better usability and identify more security threats than Misuses Cases, although the threats identified by both were different and complementary. This finding guides us to think that there is a research gap on the comparison of threat identification techniques, in order to assess their impact and explore their complementary nature.

No security threat identification methods explicitly supporting Architectural Tactics were found, but two initiatives [38, 44] operated at an architectural level. From the 19 initiatives, 6 showed evidence of application in industrial settings.

All the identified initiatives defined security in at least one phase and at least 8 initiatives, in two phases of software development lifecycle. Although there are many forms in which security could be introduced in the software development, the most repeated form was in the search of requirements or security requirements, and at least eight of them seemed to support other development activities. However, proposals were not clear in how they integrate the rest of the SDLC. Proposal [33], elicited security requirements based on business and organizational goals, in contrast with the majority who elicited from scenario based requirement development initiatives. This guides us to think that the scenario based requirements analysis is more suitable for security analysis, but further research is required to validate this idea. Elicitation of threats does not always involve a whole methodology, for example Misuse Case and Attack Trees are techniques or methods contextualized in the requirement analysis activities, but not involved in a whole development methodology.

Studies have shown that there is no common agreement with some word definition or language in the Security Community [25], [28], [34], [31], [30], [35], [3]. Some studies presented security domain models to define the domain in which a threat could be related to others entities, others just used a definition already used by others, either way, we realized that the definition of the term “threat” depends on the initiative and it is not a global concept, as some with poor experience in security could have thought. For example, some would define the term “threat” as a way to use something in a way it is not supposed to; others may think that a threat is a temporal sequence of attack steps; there are those who could not consider this construct at all and just use the term “attack”. This suggest the necessity of a common conceptual model focused on identification and mitigation of security threats to conform a unified vocabulary for future research in technologies for building secure software systems.

As can be inferred from references of the techniques in table 7, five techniques were studied in more than one article: Secure Tropos with Agents, SQUARE, Attack Trees, Misuse Cases and Misuse Case Maps. Studies for Secure Tropos with Agents, and Misuse Case Maps seemed to be a continuous endeavor of the same research groups, while SQUARE, Attack Trees and Misuse Cases are initiatives continued by different researchers. Remarkably, the research team of Karpati et al. studied several secure software development initiatives in a systematic and evidence based contribution.

The backwards snowballing mapping allowed us to trace the evolution of the initiatives from merely theoretical proposals to initiatives experimentally tested in industrial settings. As figure 6 depicts, Misuse Cases and Attack Trees are the most well founded initiatives: they were based in several theoretical proposals, showed the higher level of scientific evidence (experimental tests) and were operated in industrial contexts. A remarkably aspect is that the initiatives with experimental evidence of their results were classified as techniques, thus, prescribe specific tasks, so they are highly operationalizable, and provide

short term results. Further studies could allow us to confirm that broader and long term initiatives such as methodologies could not be validated through an experimental approach.

Finally, in light of the results of this SM, and consistently with our original concerns - the lack of scientific evidence in initiatives presented in [1] and the fulfillment of the recommendations described in [17] - we have identified a research gap in the formalization of the application and validation of secure software development initiatives.

7 Acknowledgements

This was supported by CONICYT - Comisión Nacional de Investigación Científica y Tecnológica (grant Fondecyt 1140408) - Chile.

References

- [1] A. V. Uzunov, E. B. Fernandez, and K. Falkner, "Engineering Security into Distributed Systems: A Survey of Methodologies," *Journal of Universal Computer Science*, vol. 18, no. 20, pp. 2920–3006, 2012. [Online]. Available: 10.3217/jucs-018-20-2920
- [2] B. Schneier, "Attack trees," *Dr. Dobbs journal*, vol. 24, no. 12, pp. 21–29, 1999.
- [3] G. Sindre and A. L. Opdahl, "Eliciting security requirements with misuse cases," *Requirements Engineering*, vol. 10, no. 1, pp. 34–44, Jun. 2004. [Online]. Available: 10.1007/s00766-004-0194-4
- [4] H. Mouratidis and P. Giorgini, "Secure tropos: a security-oriented extension of the tropos methodology," *International Journal of Software Engineering and Knowledge Engineering*, vol. 17, no. 02, pp. 285–309, 2007. [Online]. Available: 10.1142/S0218194007003240
- [5] L. Bass, P. Clements, and R. Kazman, *Software Architecture in Practice*. Addison-Wesley Professional, 2012.
- [6] E. B. Fernandez, *Security patterns in practice: designing secure architectures using software patterns*. John Wiley & Sons, 2013.
- [7] D. Xu and K. Nygard, "Threat-driven modeling and verification of secure software using aspect-oriented Petri nets," *IEEE Transactions on Software Engineering*, vol. 32, no. 4, pp. 265–278, Apr. 2006. [Online]. Available: 10.1109/TSE.2006.40
- [8] D. Gollmann, "Facets of security," in *Global Computing. Programming Environments, Languages, Security, and Analysis of Systems*, ser. Lecture Notes in Computer Science, C. Priami, Ed. Springer Berlin Heidelberg, 2003, vol. 2874, pp. 192–202. [Online]. Available: 10.1007/978-3-540-40042-4_6
- [9] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, "Systematic mapping studies in software engineering," in *Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering*, ser. EASE'08. Swinton, UK, UK: British Computer Society, 2008, pp. 68–77.
- [10] K. Petersen, S. Vakkalanka, and L. Kuzniarz, "Guidelines for conducting systematic mapping studies in software engineering: An update," *Information and Software Technology*, vol. 64, pp. 1–18, aug 2015. [Online]. Available: 10.1016/j.infsof.2015.03.007
- [11] C. Wohlin, "Guidelines for snowballing in systematic literature studies and a replication in software engineering," in *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*. ACM, 2014, p. 38. [Online]. Available: 10.1145/2601248.2601268
- [12] R. Noel, G. Pedraza-garcia, H. Astudillo, and E. B. Fernandez, "An Exploratory Comparison of Security Patterns and Tactics to Harden Systems," *12 Workshop on Experimental Software Engineering*, 2014.
- [13] R. Noel, G. Pedraza-Garcia, H. Astudillo, S. Matalonga, and O. Encina, "Comparando patrones de seguridad y tácticas de seguridad para construir sistemas seguros: Identificando amenazas de seguridad," in *XVIII Ibero-American Conference on Software Engineering*, 2015.

- [14] F. Braz, E. Fernandez, and M. VanHilst, "Eliciting security requirements through misuse activities," in *Database and Expert Systems Application, 2008. DEXA '08. 19th International Workshop on*, Sept 2008, pp. 328–333. [Online]. Available: 10.1109/DEXA.2008.101
- [15] G. Márquez, P. Silva, R. Noel, S. Matalonga, and H. Astudillo, "Identifying emerging security concepts using software artifacts through an experimental case." in *Workshop on Advanced Software Engineering (WASE), a track of the 22th JCC*, 2015. [Online]. Available: 10.1109/SCCC.2015.7416581
- [16] E. B. Fernandez, H. Astudillo, and G. Pedraza-García, *Revisiting Architectural Tactics for Security*. Cham: Springer International Publishing, 2015, pp. 55–69. [Online]. Available: 10.1007/978-3-319-23727-5_5
- [17] B. Whyte and J. Harrison, "State of Practice in Secure Software: Experts Views on Best Ways Ahead." IGI Global. [Online]. Available: 10.4018/978-1-61520-837-1.ch001
- [18] A. Sampaio, "Improving systematic mapping reviews," *SIGSOFT Softw. Eng. Notes*, vol. 40, no. 6, pp. 1–8, Nov. 2015. [Online]. Available: 10.1145/2830719.2830732
- [19] A. Fernandez, E. Insfran, and S. Abrahão, "Usability evaluation methods for the web: A systematic mapping study," *Inf. Softw. Technol.*, vol. 53, no. 8, pp. 789–817, Aug. 2011. [Online]. Available: 10.1016/j.infsof.2011.02.007
- [20] F. Giraldo, S. Espana, and O. Pastor, "Analysing the concept of quality in model-driven engineering literature: A systematic review," in *Research Challenges in Information Science (RCIS), 2014 IEEE Eighth International Conference on*, May 2014, pp. 1–12. [Online]. Available: 10.1109/RCIS.2014.6861030
- [21] D. Budgen, M. Turner, P. Brereton, and B. Kitchenham, "Using mapping studies in software engineering," in *Proceedings of PPIG*, vol. 8. Lancaster University, 2008, pp. 195–204.
- [22] J. Biolchini, P. G. Mian, A. Candida, and C. Natali, "Systematic Review in Software Engineering," *Engineering*, vol. 679, no. May, pp. 165–176, 2005. [Online]. Available: 10.1007/978-3-540-70621-2
- [23] M. Pai, M. McCulloch, J. D. Gorman, N. Pai, W. Enanoria, G. Kennedy, P. Tharyan, and J. M. Colford, "Systematic reviews and meta-analyses: an illustrated, step-by-step guide." *The National medical journal of India*, vol. 17, no. 2, pp. 86–95, 2004.
- [24] S. Faily and I. Fléchaïs, "Towards Tool-Support for Usable Secure Requirements Engineering with CAIRIS," *International Journal of Secure Software Engineering*, vol. 1, no. 3, pp. 56–70, 2010. [Online]. Available: 10.4018/jsse.2010070104
- [25] G. Georg, K. Anastasakis, B. Bordbar, S. H. Houmb, I. Ray, and M. Toahchoodee, "Verification and Trade-Off Analysis of Security Properties in UML System Models," *IEEE Transactions on Software Engineering*, vol. 36, no. 3, pp. 338–356, May 2010. [Online]. Available: 10.1109/TSE.2010.36
- [26] A. Toval, J. Nicolás, B. n. Moros, and F. García, "Requirements Reuse for Improving Information Systems Security: A Practitioners Approach," *Requirements Engineering*, vol. 6, no. 4, pp. 205–219, Jan. 2002. [Online]. Available: 10.1007/PL00010360
- [27] H. Suleiman and D. Svetinovic, "Evaluating the effectiveness of the security quality requirements engineering (SQUARE) method: A case study using smart grid advanced metering infrastructure," *Requirements Engineering*, vol. 18, no. 3, pp. 251–279, 2013. [Online]. Available: 10.1007/s00766-012-0153-4
- [28] C. Kalloniatis, H. Mouratidis, and S. Islam, "Evaluating cloud deployment scenarios based on security and privacy requirements," *Requirements Engineering*, vol. 18, no. 4, pp. 299–319, Apr. 2013. [Online]. Available: 10.1007/s00766-013-0166-7
- [29] Ó. Sánchez, F. Molina, J. García-Molina, and A. Toval, "Modelsec: a generative architecture for model-driven security," *J. Univ. Comput. Sci.*, vol. 15, no. 15, pp. 2957–2980, 2009. [Online]. Available: 10.3217/jucs-015-15-2957

- [30] R. Matulevicius, H. Mouratidis, N. Mayer, E. Dubois, and P. Heymans, "Syntactic and semantic extensions to secure tropos to support security risk management." *J. UCS*, vol. 18, no. 6, pp. 816–844, 2012. [Online]. Available: 10.3217/jucs-018-06-0816
- [31] H. Mouratidis, S. Islam, C. Kalloniatis, and S. Gritzalis, "A framework to support selection of cloud providers based on security and privacy requirements," *Journal of Systems and Software*, vol. 86, no. 9, pp. 2276–2293, Sep. 2013. [Online]. Available: 10.1016/j.jss.2013.03.011
- [32] A. L. Opdahl and G. Sindre, "Experimental comparison of attack trees and misuse cases for security threat identification," *Information and Software Technology*, vol. 51, no. 5, pp. 916–932, 2009. [Online]. Available: 10.1016/j.infsof.2008.05.013
- [33] J. Trujillo, E. Soler, E. Fernández-Medina, and M. Piattini, "An engineering process for developing Secure Data Warehouses," *Information and Software Technology*, vol. 51, no. 6, pp. 1033–1051, Jun. 2009. [Online]. Available: 10.1016/j.infsof.2008.12.003
- [34] P. Karpati, Y. Redda, A. L. Opdahl, and G. Sindre, "Comparing attack trees and misuse cases in an industrial setting," *Information and Software Technology*, vol. 56, no. 3, pp. 294–308, Mar. 2014. [Online]. Available: 10.1016/j.infsof.2013.10.004
- [35] A. Razzaq, Z. Anwar, H. F. Ahmad, K. Latif, and F. Munir, "Ontology for attack detection: An intelligent approach to web application security," *Computers & Security*, vol. 45, pp. 124–146, Sep. 2014. [Online]. Available: 10.1016/j.cose.2014.05.005
- [36] L. Røstad, "An extended misuse case notation: Including vulnerabilities and the insider threat," in *XII Working Conference on Requirements Engineering: Foundation for Software Quality, Luxembourg*, 2006.
- [37] J. McDermott, "Abuse-case-based assurance arguments," in *Computer Security Applications Conference, 2001. ACSAC 2001. Proceedings 17th Annual*. IEEE, 2001, pp. 366–374. [Online]. Available: 10.1109/ACSAC.2001.991553
- [38] J. Pauli and D. Xu, "Threat-driven architectural design of secure information systems," in *Proceeding of First International Workshop on Protection by Adaptation (PBA 2005, In conjunction with the 7th International Conference on Enterprise Information Systems (ICEIS 2005))*, Miami, 2005.
- [39] G. Sindre, D. G. Firesmith, and A. L. Opdahl, "A reuse-based approach to determining security requirements," in *Proceedings of the 9th international workshop on requirements engineering: foundation for software quality (REFSQ03)*, Klagenfurt, Austria. Citeseer, 2003.
- [40] G. Stoneburner, A. Y. Goguen, and A. Feringa, "Risk management guide for information technology systems," 2002.
- [41] N. R. Mead and T. Stehney, *Security quality requirements engineering (SQUARE) methodology*. ACM, 2005, vol. 30, no. 4. [Online]. Available: 10.1145/1083200.1083214
- [42] G. Elahi, E. Yu, and N. Zannone, "A vulnerability-centric requirements engineering framework: analyzing security attacks, countermeasures, and requirements based on vulnerabilities," *Requirements engineering*, vol. 15, no. 1, pp. 41–62, 2010. [Online]. Available: 10.1007/s00766-009-0090-z
- [43] V. Gandotra, A. Singhal, and P. Bedi, "Identifying security requirements hybrid technique," in *Software Engineering Advances, 2009. ICSEA'09. Fourth International Conference on*. IEEE, 2009, pp. 407–412. [Online]. Available: 10.1109/ICSEA.2009.65
- [44] P. Karpati, A. L. Opdahl, and G. Sindre, "Experimental comparison of misuse case maps with misuse cases and system architecture diagrams for eliciting security vulnerabilities and mitigations," in *Availability, Reliability and Security (ARES), 2011 Sixth International Conference on*. IEEE, 2011, pp. 507–514. [Online]. Available: 10.1109/ARES.2011.77
- [45] P. Karpati, G. Sindre, and A. L. Opdahl, "Visualizing cyber attacks with misuse case maps," in *International Working Conference on Requirements Engineering: Foundation for Software Quality*. Springer, 2010, pp. 262–275. [Online]. Available: 10.1007/978-3-642-14192-8_24

- [46] A. Van Lamsweerde, S. Brohez, R. De Landtsheer, D. Janssens *et al.*, “From system goals to intruder anti-goals: attack generation and resolution for security requirements engineering,” *Proc. of RHAS*, vol. 3, pp. 49–56, 2003.
- [47] I. A. Tøndel, J. Jensen, and L. Røstad, “Combining misuse cases with attack trees and security activity models,” in *Availability, Reliability, and Security, 2010. ARES’10 International Conference on.* IEEE, 2010, pp. 438–445. [Online]. Available: 10.1109/ARES.2010.101
- [48] S. Islam, H. Mouratidis, and J. Jürjens, “A framework to support alignment of secure software engineering with legal regulations,” *Software & Systems Modeling*, vol. 10, no. 3, pp. 369–394, 2011. [Online]. Available: 10.1007/s10270-010-0154-z
- [49] S. Islam, H. Mouratidis, C. Kalloniatis, A. Hudic, and L. Zechner, “Model based process to support security and privacy requirements engineering,” *International Journal of Secure Software Engineering (IJSSE)*, vol. 3, no. 3, pp. 1–22, 2012. [Online]. Available: 10.4018/jsse.2012070101
- [50] H. Mouratidis and J. Jurjens, “From goal-driven security requirements engineering to secure design,” *International Journal of Intelligent Systems*, vol. 25, no. 8, pp. 813–840, 2010. [Online]. Available: 10.1002/int.20432
- [51] G. Sindre, “Mal-activity diagrams for capturing attacks on business processes,” in *International Working Conference on Requirements Engineering: Foundation for Software Quality*. Springer, 2007, pp. 355–366. [Online]. Available: 10.1007/978-3-540-73031-6_27
- [52] P. H. R. Matulevičius, H. Mouratidis, N. Mayer, E. Dubois, “Syntactic and Semantic Extensions to Secure Tropos to Support Security Risk Management.” [Online]. Available: 10.3217/jucs-018-06-0816
- [53] A. T. Ó. Sánchez, F. Molina, J. García-Molina, “ModelSec: A Generative Architecture for Model-Driven Security.” [Online]. Available: 10.3217/jucs-015-15-2957