Computer Standards & Interfaces 32 (2010) 153–165

Contents lists available at ScienceDirect

Computer Standards & Interfaces

journal homepage: www.elsevier.com/locate/csi

A systematic review of security requirements engineering

Daniel Mellado [a], Carlos Blanco [b], Luis E. Sánchez [c], Eduardo Fernández-Medina [b,*]

[a] Spanish Tax Agency. Madrid, Spain
[b] Department of Information Technologies and Systems. University of Castilla-La Mancha. Alarcos Research Group. Paseo de la Universidad, 4. Ciudad Real, Spain
[c] SICAMAN Nuevas Tecnologías. Tomelloso, Ciudad Real, Spain

ARTICLE INFO

Article history:
Received 7 January 2009
Received in revised form 25 January 2010
Accepted 27 January 2010
Available online 2 February 2010

Keywords:
Security requirements
Security requirements engineering
Requirements engineering
Security engineering
Secure development
Security
Systematic review

ABSTRACT

One of the most important aspects in the achievement of secure software systems in the software development process is what is known as Security Requirements Engineering. However, very few reviews focus on this theme in a systematic, thorough and unbiased manner, that is, none of them perform a systematic review of security requirements engineering, and there is not, therefore, a sufficiently good context in which to operate. In this paper we carry out a systematic review of the existing literature concerning security requirements engineering in order to summarize the evidence regarding this issue and to provide a framework/background in which to appropriately position new research activities.

Contents

* Corresponding author.
E-mail addresses: damefe@esdebian.org (D. Mellado), Carlos.Blanco@uclm.es (C. Blanco), lesanchez@sicaman-nt.com (L.E. Sánchez), Eduardo.FdezMedina@uclm.es (E. Fernández-Medina).

0920-5489/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.csi.2010.01.006

1. Introduction

The proliferation of connectivity of Information Systems (IS) and the increasing complexity of applications and services, signify that there is a correspondingly greater chance of suffering security breaches [1]. Present-day information systems are vulnerable to a host of threats and cyber-attackers such as malicious hackers, code writers, cyber-terrorists, etc. [2]. In addition, owing to the heavy dependence of computer network-based applications on various software and software controlled systems, the consequences of a security breach in these applications may range from extensive financial losses to dangers to human life. The threat of technology-enabled crime has given rise to a growing demand for the creation of new response strategies [2]. Software security has therefore become an essential issue [3] and a fair amount of additional security expertise is needed to meet non-functional security requirements [4].

However, security is rarely at the forefront of stakeholders concerns, except perhaps to comply with basic standards or legal requirements. Hence, work in requirements has primarily focused on eliciting and representing concrete business requirements [5], whilst requirements engineers often fail to pay sufficient attention to security concerns. The biggest problem, however, is that in the majority of software projects security is dealt with when the system has already been designed and put into operation. In addition to this, the actual security requirements themselves are often not well understood. This being so, even when there is an attempt to define security requirements, many developers tend to describe design solutions in terms of protection mechanisms, rather than making declarative propositions with regard to the level of protection required [6]. As a result, and perhaps for these reasons, although security requirements engineering has recently attracted increasing attention, it has lacked a systematic review which would supply researchers with a summary of all the existing information about security requirements in a thorough and unbiased manner, thus providing a context in which to operate.

Software Security Engineering, which is a practice through which to address software security issues in a systematic manner, is known to be a very important part of the software development process for the achievement of secure software systems. Nevertheless, within this discipline we believe in the particular importance of Security Requirements Engineering, which provides techniques, methods and norms for tackling this task during the early stages of the IS development cycle, since the building of security into the early stages of the development process is cost-effective and also brings about more robust designs [7]. It should involve the use of repeatable and systematic procedures in an effort to ensure that the set of requirements obtained is complete, consistent, easy to understand and analyzable by the different actors involved in the development of the system [8]. A good requirements specification document should include both functional requirements (related to the services that the software or system should provide), and non-functional requirements (related to what are known as features of quality, performance, portability, security, etc). In our contemporary Information Society, depending as it does on a huge number of software systems which play a critical role, it is absolutely vital to ensure that IS are safe right from the very beginning [9].

During the last few years, a number of papers have focused on security requirements, some of which have carried out reviews on this issue. However, most of these reviews consist of only one section in the paper/article and there are very few papers in which a review of security requirements is the core. After performing preliminary searches aimed at both identifying existing systematic reviews and assessing the volume of potentially relevant studies, we can highlight several works in which a summary of security requirements related issues is carried out, such as [3,10–13]. However, none of them perform a review focused on security requirements engineering in a systematic manner, that is, none of them perform a systematic review

of security requirements engineering. They are not, therefore, a sufficiently good context in which to operate in security requirements engineering.

In this paper, we shall carry out a systematic review (SR) of the existing literature related to security requirements engineering, not only in order to summarize the existing evidence concerning this issue but also to provide a framework/background in which to appropriately position new research activities. As this is a systematic review, it will synthesise the existing work in way that it is fair and seen to be fair [14–16]. In contrast to the usual process of a literature review, which is unsystematically conducted whenever somebody starts a particular piece of research, an SR is developed, as the terms denote, in a formal and systematic way [17]. This means that the research process of a systematic type of review follows a very well defined and strict sequence of methodological steps, according to an aprioristically developed protocol. This is conducted around a central issue, which represents the core of the investigation, and is expressed by using a specific, pre-defined, focused and structured question. The methodological steps, the strategies to retrieve the evidence, and the focus on the question are explicitly defined, so that other professionals can reproduce the same protocol and can also judge the suitability of the standards chosen for the case in question [17].

This systematic review will be performed by using the guidelines for systematic reviews proposed by Kitchenham [14–16], which is appropriate for software engineering researchers. We shall also use a review protocol template developed by Biolchini et al. [17] which facilitates the planning and execution of systematic reviews in software engineering. The remainder of the paper is thus set out as follows: in Section 2 we shall define the research question. Section 3 will explain the review method, which is based on the research protocol and it is here that the search strategy and studies selection will be defined. Next, in Section 4 we shall define the data to be extracted and this will be presented in the data synthesis, the summary and the synthesis of the relevant studies. In Section 5 we shall present the results and the discussion. Finally, our conclusions will be set out in Section 6.

2. Question formalization

This section will clearly define the research objectives.

2.1. Question focus

The *question focus* is to identify initiatives and experience reports in Software Engineering which consider security requirements from the beginning of the IS development in order to develop secure IS by means of Security Requirements Engineering.

2.2. Question quality and amplitude

The biggest *problem* is that in the majority of software projects security is dealt with when the system has already been designed and put into operation and those requirements engineers often fail to pay sufficient attention to security concerns. On many occasions, this is as the result of an inappropriate management of the specification of the security requirements of the new system, since the stage known as the requirements specification phase is often carried out with the aid of just a few descriptions, or the specification of objectives is put down on a few sheets of paper. In addition to this, the actual security requirements themselves are often not well understood. This being so, even when an attempt is made to define security requirements, many developers tend to describe design solutions in terms of protection mechanisms rather than making declarative propositions regarding the level of protection required [18].

The *research question* that the research will address is the following: Which initiatives have been carried out to develop secure IS by means of Security Requirements Engineering? The *keywords and synonyms* of which this question is composed and which will be used during the execution of the review are:

Security requirements.
Security requirements engineering: requirements engineering, security engineering.
Secure development: secure IS development, secure software development.

What will be *observed* in the context of this systematic review is how security requirements are dealt with in the security-critical IS development. The *population* group that will be observed is, therefore, those publications which consider security requirements from the beginning of the IS development in order to develop secure IS.

The *expected result* at the end of this systematic review is the identification of initiatives related to Security Requirements Engineering and the *outcome measure* will be the number of identified initiatives. The main *application areas* that will benefit from the systematic review results are secure software development and Software Engineering, specifically Security Requirements Engineering, and also security experts and requirements engineers. To do this, a framework/background will be provided in which to appropriately position new research activities in security requirements engineering.

3. Review method

The review method is based on the research protocol and it is in this section that the search strategy, the sources, the studies selection and the selection execution will be defined.

3.1. Sources selection

The objective of this section is to select the sources in which searches for primary studies will be executed.

It will first be necessary to carry out preliminary searches aimed at both identifying existing systematic reviews and assessing the volume of potentially relevant studies, and in this case we found the following relevant initiatives: [3,10–13], which we included as sources. The *selection criteria* with which to evaluate studies sources will thus be the possibility of consulting papers on the Internet or in the digital library of the University of Castilla- La Mancha, which has e-books and also access to the ACM digital library, IEEE digital library, Science@-Direct, Elsevier (among others); the presence of a search mechanism using keywords; and publishing companies, books, journals and conferences suggested by experts in the field (such as the members of RETISTRUST[1], a Spanish Network on Security in Software Engineering). The studies must be written in English.

The *search* for primary studies will be carried out by using web search engines, electronic databases and manual searches, such as research in a specific journal/conference/magazine/book or in research publications suggested by experts in the field.

Finally, below we present the main sources of the *initial sources list*, in which the systematic review execution will be run: ACM digital library, IEEE digital library, Science@Direct, Google Scholar, , SREIS symposium, ESORICS symposium, REFSQ conference, IEEE International Requirements Engineering Conference, ICSE conference, COMPSAC conference, DEXA conference, WOSIS workshop, ICCSA conference, Requirements Engineering Journal, Computer Standards & Interfaces Journal, Computers & Security.

[1] RETISTRUST — Research Network in Security and Trust for Information Systems in an online society.

3.2. Studies selection

Once the sources have been defined it is necessary to describe the process and the criteria for studies selection and evaluation since in order to reduce the likelihood of bias, selection criteria should be decided during the protocol definition.

The *inclusion and exclusion criteria* should be based on the research question. We have therefore established that the studies must present new initiatives (from a maximum of 5 years ago) which consider all kinds of security requirements from the beginning of the IS development in order to develop secure IS. Some kind of process, method, steps or description to follow in order to carry out security requirements engineering must be described in these studies. This research will not select studies which are not focused on the requirements phase, or studies which treat security requirements as just another non-functional requirement, signifying that they are not specifically focused on security requirements.

In order to select an initial set of studies, the title and abstract of all the obtained studies is read and evaluated according to the aforementioned inclusion and exclusion criteria. This initial set of studies is refined by reading their full texts.

3.3. Selection execution

The search is executed in order to obtain an initial list of studies for further evaluation. The bibliography management of the studies obtained is performed by using a bibliographic package (EndNote).

The procedures for studies selection are then applied to all the articles obtained in order to verify whether the studies fit the inclusion and exclusion criteria. The obtained studies which completely fit all the inclusion and exclusion criteria previously defined are the following:

- Basin et al. "Model-driven security for process-oriented systems" [19] and "Model driven security: From UML models to access control infrastructures" [20].
- Bresciani et al. "Tropos: Agent-Oriented Software Development Methodology" [21], Giorgini et al. "Requirements Engineering meets Trust Management: Model, Methodology, and Reasoning" [22], Giorgini et al. "Modelling Security and Trust with Secure Tropos" [23], Ali et al. "Location-based Software Modeling and Analysis: Tropos-based Approach" [24] and "A Goal Modeling Framework for Self-Contextualizable Software" [25], Dalpiaz et al. "An Architecture for Requirements-Driven Self-reconfiguration" [26], Massacci et al. "Using a security requirements engineering methodology in practice: The compliance with the Italian data protection legislation" [27] and Compagna et al. "How to integrate legal requirements engineering into a requirements engineering methodology for the development of security and privacy patterns" [28].
- Firesmith "Specifying Reusable Security Requirements" [6], "Engineering safety-related requirements for software-intensive systems" [29] and "Engineering Safety and Security Related Requirements for Software Intensive Systems" [30].
- Hussein and Zulkernine "Intrusion detection aware component-based systems: A specification-based framework" [31].
- Jennex "Modeling security requirements for information systems development" [32].
- J. Lee, et al. "A CC-based Security Engineering Process Evaluation Model" [33].
- S.-W. Lee et al. "Building problem domain ontology from security requirements in regulatory documents" [34].
- Mead and Stehney "Security Quality Requirements Engineering (SQUARE) Methodology" [35], Mead and Hough "Security Requirements Engineering for Software Systems: Case Studies in Support of Software Engineering Education" [36] and Abu-Nimeh et al. "Integrating Privacy Requirements into Security Requirements Engineering" [37].
- Mellado et al. "A Common Criteria Based Security Requirements Engineering Process for the Development of Secure Information Systems" [38] and "Towards security requirements management for software product lines: a security domain requirements engineering process" [39].
- Moffett and Nuseibeh "A Framework for Security Requirements Engineering" [13] and Haley et al. "Security Requirements Engineering: A Framework for Representation and Analysis" [40].
- Morimoto, et al. "A Security Requirement Management Database Based on ISO/IEC 15408" [41] and Horie et al. "ISEDS: An Information Security Engineering Database System Based on ISO Standards" [42].
- Myagmar et al. "Threat Modeling as a Basis for Security Requirements" [43].
- Peeters "Agile Security Requirements Engineering" [44].
- Popp et al. "Security-Critical System Development with Extended Use Cases" [45], Jürjens "UMLsec: extending UML for secure systems development" [46] and Jürjens et al. "Automated Analysis of Permission-Based Security Using UMLsec" [47].
- Shin and Gomaa "Software requirements and architecture modelling for evolving non-secure applications into secure applications" [48].
- Sindre and Opdahl "Eliciting security requirements with misuse cases" [49], Sindre et al. "A Reuse-Based Approach to Determining Security Requirements" [50], Opdahl and Sindre "Experimental comparison of attack trees and misuse cases for security threat identification" [51] and Stalhane and Sindre "Safety Hazard Identification by Misuse Cases: Experimental Comparison of Text and Diagrams" [52].
- Toval et al. "Requirements Reuse for Improving Information Systems Security: A Practitioner's Approach" [53], Martínez et al. "An Audit Method of Personal Data Based on Requirements Engineering" [54], Nicolás et al. "A Collaborative Learning Experience in Modelling the Requirements of Teleoperated Systems for Ship Hull Maintenance" [55] and Lasheras et al. "An Ontology-Based Framework for Modelling Security Requirements" [56].
- Tsoumas and Gritzalis. "Towards an Ontology-based Security Management" [57] and Tsoumas et al. "Security-by-Ontology: A Knowledge-Centric Approach" [58].
- Viega "Building security requirements with CLASP"[5].
- Yu "Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering" [59], Yu et al. "A Social Ontology for Integrating Security and Software Engineering" [60] and Yu "Social Modeling and i*" [61].
- Zuccato "Holistic security requirement engineering for electronic commerce" [62] and "Holistic security management framework applied in electronic commerce" [63]. Zuccato et al. "Security Requirements Engineering at a Telecom Provider" [64].

4. Information extraction

The *information extracted* from the studies must contain techniques, methods, processes, steps, strategies or any kind of initiative to establish security requirements in a systematic way during the early phases of the IS development.

The *information forms* defined for this systematic review will contain the study identification, the study methodology, the study results, the study problems and our general impressions and abstractions. Regarding the study methodology, we shall focus on the modelling of the security requirements, on the modelling / development standard and on the security standards, along with the technical criteria defined within the analytical framework explained in the following section.

The following sub-section provides a brief outline of each of the selected studies/initiatives shown in the previous section, according to the extracted information obtained through the information forms.

4.1. Basin et al. "Model-driven security for process-oriented systems" [19] and "Model driven security: From UML models to access control infrastructures" [20]

The authors show how the Model Driven Architecture paradigm can be specialized into what they call Model Driven Security. They present an application for constructing systems from process models, in which they combine a UML-based process design language with a security modelling language for formalizing access control requirements (called SecureUML). Models in the combined language are used to automatically generate security architectures for distributed applications.

4.2. Bresciani et al. "Tropos: Agent-Oriented Software Development Methodology" [21], Giorgini et al. "Requirements Engineering meets Trust Management: Model, Methodology, and Reasoning" [22] and Giorgini et al. "Modelling Security and Trust with Secure Tropos" [23], Ali et al. "Location-based Software Modeling and Analysis: Tropos-based Approach" [24] and "A Goal Modeling Framework for Self-Contextualizable Software" [25], Dalpiaz et al. "An Architecture for Requirements-Driven Selft-reconfiguration" [26], Massacci et al. "Using a security requirements engineering methodology in practice: The compliance with the Italian data protection legislation" [27] and Compagna et al. "How to integrate legal requirements engineering into a requirements engineering methodology for the development of security and privacy patterns" [28]

The Tropos methodology is intended to support all the analysis and design activities in the software development process. Tropos rests on the idea of building a model of the system-to-be and its environment, which is incrementally refined and extended to providing both a common interface for various software development activities and as a basis for the documentation and evolution of the software. This methodology is based on social hierarchies and adapts components of the i* framework [60]. This uses the concepts of actors, goals, tasks, resources and social dependencies to define the obligations of actors (dependees) towards other actors (dependers). The authors improve the social ontology created for the i* framework with new security concepts: constraints, secure entities (secure goals, tasks, resources, ownership) and secure dependences between actors (such as trust of execution, trust of permission, delegation of permission and delegation of execution).

The five main development phases of Tropos are: Early requirements, Late requirements, Architectural design, Detailed design and Implementation.

There are several extensions of Tropos, one of the most important being the proposal of Giorgini et al. [22], which presents a formal framework for modelling and analyzing security and trust requirements.

Ali et al. [24,25] extend the Tropos goal model with contextual variability, defining variation points on the goal model and associating a context with them as a means to select between alternatives. The work provides novel modelling constructs to analyze high level contexts in order to elicit the monitoring requirements, i.e. the data the system has to monitor to verify the high level contexts.

Another extension of Tropos [26] has recently been applied to define requirements in a model-based approach for self-reconfiguration.

Furthermore, several works [27,28] present case studies of the Secure Tropos requirements engineering methodology applied in compliance with the Italian legislation on Privacy and Data Protection by the University of Trento, leading to the definition and analysis of an ISO-17799-like security management scheme. The proposed constructs and methodology were not up to the challenge and revealed a number of pitfalls, especially when the formal analysis techniques were applied.

4.3. Firesmith "Specifying Reusable Security Requirements" [6], "Engineering safety-related requirements for software-intensive systems" [29] and "Engineering Safety and Security Related Requirements for Software Intensive Systems" [30]

Firesmith offers some steps which allow security requirements to be defined from reusable templates. His analysis of security requirements is founded on two basic principles obtained from OCTAVE (Operationally Critical Threat, Asset, and Vulnerability Evaluation) which are based on resources and are risk-driven. The author proposes security use cases as a technique that should be used to specify the security requirements that the applications will successfully fulfil to protect themselves from the relevant security threats.

Furthermore, the author presents a consistent set of information models that identify and define the foundational concepts underlying safety, security and survivability engineering. He defines *safety* as the degree to which *accidental harm* is prevented, detected, and reacted to; *security* as the degree to which *malicious harm* is prevented, detected, and reacted to; and *survivability* as the degree to which *both accidental and malicious harm* to essential services is prevented, detected, and reacted to. The information models presented provide a standard terminology and set of concepts that explain the similarities between the asset-based, risk-driven methods for identifying and analyzing safety, security, and survivability requirements, along with a rationale for the similarity in architectural mechanisms that are commonly used to fulfil these requirements.

4.4. Hussein and Zulkernine "Intrusion detection aware component-based systems: A specification-based framework" [31]

The authors propose a framework for developing components with intrusion detection capabilities. The first stage of this framework is the requirement elicitation, in which developers identify services and intrusions. That is, they capture users' requirements regarding the services and functionalities provided by the components, and identify the unwanted or illegal usage of components by intruders. Intrusion scenarios are elicited through the use of misuse-cases of a UML profile called UMLintr.

4.5. Jennex "Modeling security requirements for information systems development" [32]

The methodology suggested by this author proposes the use of barrier analysis diagrams as a graphical method through which to identify and document security requirements. Furthermore, this approach uses meta-notation to add security details to existing system development diagrams. The process follows the approach of integrating security design into the software development life-cycle. Therefore, the objective of using barrier diagrams during the requirement phase is that of appropriately identifying the security requirements.

4.6. Lamsweerde, "Engineering requirements for system reliability and security" [65]

In this proposal, the author pulls together all his previous research on the goal-oriented requirements analysis method KAOS, formalization of requirements using linear time temporal logic, requirements conflict analysis, and the use of antimodels for elaborating security requirements. He therefore extends KAOS to include the elaboration of security requirements.

4.7. J. Lee et al. "A CC-based Security Engineering Process Evaluation Model" [33]

On the one hand, the Common Criteria (CC) only provides us with standards to evaluate product and security systems information. On

the other hand, SSE-CMM provides us with security standards for the evaluation of process engineering. However, the authors propose the integration of CC and SSE-CMM to create CC-SSE-CMM, a maturity model that includes the advantages of both models. This new model is divided into processes, products and environment. The advantage of this model is that it is useful when an organization that was developed with CC has to be evaluated with SSE-CMM to improve its level with regard to the security process. CC_SSECMM consists of 23 process areas with 5 maturity levels. Each process area (PA) has BP (base practices) and the capacity levels have GP (generic practices).

4.8. Lee et al. "Building problem domain ontology from security requirements in regulatory documents" [34]

The authors identify security requirements for certification and accreditation activities which are expressed in regulatory documents. These requirements have a non-functional nature which imposes complex constraints on the behaviour of software systems and makes them hard to understand, predict and control.

The authors present a framework which includes techniques extracted from software requirements engineering and knowledge engineering and they propose a common language with which to extract concepts from regulatory documents. They apply this methodology to the construction of a problem domain ontology from regulatory documents enforced by the DITSCAP - Department of Defense Information Technology Security Certification and Accreditation Process.

4.9. Mead and Stehney "Security Quality Requirements Engineering (SQUARE) Methodology" [35], Mead and Hough "Security Requirements Engineering for Software Systems: Case Studies in Support of Software Engineering Education" [36] and Abu-Nimeh et al. "Integrating Privacy Requirements into Security Requirements Engineering" [37]

The authors propose a process which provides a means for eliciting, categorizing, and prioritizing security requirements for information technology systems and applications. The focus of this methodology is that of building security concepts into the early stages of the development lifecycle. The model may also be useful for documenting and analyzing the security aspects of fielded systems, and could be used to steer future improvements and modifications to these systems. The 9-step process consists of: 1- Agree on definitions; 2- Identify security goals; 3- Develop supporting artefacts; 4- Perform risk assessment; 5- Select elicitation techniques; 6- Elicit security requirements; 7- Categorize requirements; 8- Prioritize requirements; and 9- Requirements inspections. This model has also been applied by graduate students in several real software development projects [36].

The same authors have recently proposed [37] a privacy requirement elicitation technique (PRET) composed of a questionnaire and the elicitation and the verification of the privacy requirements. This technique has been integrated into SQUARE methodology, as an elicitation technique that can be chosen in step 5, and has been also validated with several case studies.

4.10. Mellado et al. "A Common Criteria Based Security Requirements Engineering Process for the Development of Secure Information Systems" [38], "Towards security requirements management for software product lines: a security domain requirements engineering process" [39]

The authors propose a standard-based process, named SREP (Security Requirements Engineering Process), that deals with the security requirements during the early stages of software development in a systematic and intuitive way. It is based on the reuse of security requirements, by providing a Security Resources Repository, together with the integration of the Common Criteria (ISO/IEC 15408) and the SSE-CMM (ISO/IEC 21827) thanks to the CC_SSE-CMM approach [33], into the software lifecycle model. It also conforms to ISO/IEC 17799:2005 with regard to security requirements. The authors start from the concept of iterative software construction and propose a micro-process for security requirements engineering, made up of nine activities, which are repeatedly performed at each iteration throughout the iterative and incremental development, but with different emphasis depending on where the iteration is in the lifecycle. Moreover, one of the most relevant aspects is the fact that this proposal integrates other approaches, such as SIREN [53], UMLSec [46], security use cases [66] or misuse cases [49]. In addition, they have developed a tool (SREPTool), which supports the process and the documentation generation. Finally, as an evolution of their previous "generic" security requirements engineering process (SREP), they have evolved SREP and SREPTool which have been specially adapted to the software product lines based development paradigm [39].

4.11. Moffett and Nuseibeh "A Framework for Security Requirements Engineering" [13] and Haley et al. "Security Requirements Engineering: A Framework for Representation and Analysis" [40]

The authors suggest a framework which unifies the concepts of the two disciplines of requirements engineering and security engineering. From requirements engineering, it takes the concepts of functional goals, which are operationalised into functional requirements, with appropriate constraints. From security engineering, it takes the concepts of assets, together with threats of harm to those assets. Security goals aim to protect the system from these threats, and are operationalised into security requirements, which take the form of constraints on the functional requirements.

4.12. Morimoto, et al. "A Security Requirement Management Database Based on ISO/IEC 15408" [41] and Horie et al. "ISEDS: An Information Security Engineering Database System Based on ISO Standards" [42]

Morimoto et al. propose a security requirement management database, named "ISEDS (Information Security Engineering Database System)" based on the international standard ISO/IEC 15408 that defines the security functional requirements which should be satisfied by various information systems. The database can aid design and development of information systems that require high security. ISEDS users can collect, manage and reuse security requirements for the design and development of various information systems in the form according to ISO/IEC 15408. ISEDS can also support design and development of information systems which satisfy the security criteria of ISO/IEC 15408.

4.13. Myagmar et al. "Threat Modeling as a Basis for Security Requirements" [43]

These writers investigate how threat modelling can be used as foundations for the specification of security requirements and they also present three case studies of threat modelling. They offer a viewpoint of the requirements engineering process in which, through the appropriate identification of threats and a correct choice of countermeasures, the ability of attackers to misuse or abuse the system is lessened. The threat-modelling process set out by these authors is made up of three high-level steps: Characterizing the system; Identifying assets and access points; and identifying threats.

4.14. Peeters "Agile Security Requirements Engineering" [44]

Peeters proposes the extension of agile practices to deal with security in an informal, communicative and assurance-driven spirit. In order to increase the agility of requirement engineering, Peeters puts forward the idea of using "abuser stories". These stories identify how the attackers may abuse the system and jeopardize stakeholders'

assets. The abuser stories thus make the establishment of security requirements easier.

4.15. Popp et al. "Security-Critical System Development with Extended Use Cases" [45], Jürjens "UMLsec: extending UML for secure systems development" [46] and Best et al. "Model-Based Security Engineering of Distributed Information Systems Using UMLSec" [67]

Jürjens presents a methodology with which to specify requirements regarding confidentiality and integrity in analysis models based on UML. The security models highlighted in this proposal are multilevel security and mandatory access control. This approach considers a UML extension to develop secure systems. The security of a subsystem specification is analyzed by modelling the behaviour of the potential attacker; hence, specific types of attackers, that may attack different parts of the system in a specific way. This proposal uses the majority of UML diagrams to model security aspects, mainly those that refer to confidentiality and integrity. For example, state chart diagrams model, the dynamic behaviour of objects, and sequence diagrams are used to model protocols. Deployment diagrams are also used to model links between components across servers. This methodology also incorporates the translation of UMLSec models defined for the introduction of patterns into the design process. Moreover, Popp et al. show a methodological approach for the development of security-critical systems and the modelling of security aspects in the application core with UMLsec. They also introduce security use cases for the development of security aspects in conjunction with behavioural modelling. Best et al. have recently applied UMLsec method in an industrial context, analyzing the security of a search engine in the intranet of a German car manufacturer.

4.16. Shin and Gomaa "Software requirements and architecture modelling for evolving non-secure applications into secure applications" [48]

This approach models the evolution of a non-secure application to a secure application in terms of the requirements model and the software architecture model (use case, static and dynamic models) by using distributed software architectures based on components and connectors. The authors propose separating security and application requirements. Security requirements are captured in security use cases and are encapsulated in security objects. The security services are encapsulated in connectors separately from the components that provide the functionality of the application.

4.17. Sindre and Opdahl "Eliciting security requirements with misuse cases" [49], Sindre et al. "A Reuse-Based Approach to Determining Security Requirements" [50], Opdahl and Sindre "Experimental comparison of attack trees and misuse cases for security threat identification" [51], Stalhane and Sindre "Safety Hazard Identification by Misuse Cases: Experimental Comparison of Text and Diagrams" [52], Whittle et al "Executable Misuse Cases for Modeling Security Concerns" [68], and Braz et al. "Eliciting Security Requirements through Misuse Activities" [69]

Standard use case diagrams are often useful for eliciting functional requirements, although they are not that suitable for describing security requirements, owing to the fact that security requirements are usually related to prohibited activities. Therefore, Sindre and Opdahl present a systematic approach for eliciting security requirements by extending traditional UML use cases to also cover misuse, and this is potentially useful for several types of extra-functional requirements other than security. Sindre et al.'s approach focuses solely on the activities directly related to reuse. They propose a reuse-based approach for determining security requirements. Development for reuse involves identifying security threats and associated security requirements during application development, and abstracting them into a repository of generic threats and requirements. Development with reuse involves identifying security assets, setting security goals for each asset, identifying threats to each goal, analyzing risks and determining security requirements, based on the reuse of generic threats and requirements from the repository. The advantages of this approach include building and managing security knowledge through the shared repository, assuring the quality of security work by reuse, avoiding over-specification and premature design decisions by reuse at the generic level and focusing on security early on in the requirements stage of development. In [51,52], the authors show several experiments. The first one compares two methods for the early elicitation of security threats (attack trees and misuse cases) and the second one uses misuse cases to compare two methods to represent use case models (text and diagram). In [68], the authors present an executable misuse case modelling language which allows modellers to specify misuse case scenarios in a formal yet intuitive way and to execute the misuse case model in tandem with a corresponding use case model. Another similar approach is proposed in [69], in which the authors propose a method that starts from the activity diagram of a use case (or a sequence of use cases). Each activity is analyzed to see how it could be subverted to produce a misuse of information. This analysis results in a set of threats. They then consider which policies might be used to stop or mitigate these threats.

4.18. Toval et al. "Requirements Reuse for Improving Information Systems Security: A Practitioner's Approach" [53], Martínez et al. "An Audit Method of Personal Data Based on Requirements Engineering" [54], Nicolás et al. "A Collaborative Learning Experience in Modelling the Requirements of Teleoperated Systems for Ship Hull Maintenance" [55]

Toval et al. define a Requirements Engineering process, named SIREN (SImple REuse of software requiremeNts), based on the re-use of security requirements, which is also compatible with MAGERIT (the Spanish public administration risk analysis and management method), and which conforms to Common Criteria (ISO/IEC 15408). The re-use of security requirements is carried out at a documentation level by defining a hierarchical structure of security requirement specifications, and at the security requirements level by means of storing it in the repository of re-usable requirements. SIREN describes a process model, some basic guidelines, techniques and tools. The guidelines consist of a hierarchy of requirement specification documents, together with the template for each document. It is a spiral model process, and includes the requirements elicitation, requirements analysis and negotiation, requirements specification and validation phases. A repository of requirements classified by domains and profiles is also defined.

These authors have recently defined case studies to apply the defined catalogues. In [54], these catalogues are applied to the auditing of personal data. The authors have also worked on developing a requirement catalogue for product lines based on SIREN and they have applied this approach to the modelling requirements of teleoperated systems for ship hull maintenance [55].

4.19. Tsoumas and Gritzalis. "Towards an Ontology-based Security Management" [57] and Tsoumas et al. "Security-by-Ontology: A Knowledge-Centric Approach" [58]

In this proposal, the authors describe a security framework of an arbitrary information system which provides security acquisition and knowledge management. This framework is based on a security ontology which extends the DMTF Common Information Model (CIM) (www.dmtf.org) with ontological semantics in order to use it as a container for IS security-related information. This Security ontology is based on security and risk management practices such as CRAMM [70] or COBIT [71]. 4 phases to establish the IS security management framework are described in this proposal, the first being the "building of the security ontology". As further work they envisage the development

of a standards-based, best practices database with implicit security knowledge to support information extraction and the decision making process which will take into account semantic rules and the properties of reusability and interoperability of the ontologies.

4.20. Viega "Building security requirements with CLASP" [5]

The authors show how to build security requirements in a structured manner that is conducive to iterative refinement and, if followed correctly, to metrics for evaluation, which will provide a framework that is an obvious improvement on traditional methods that do not consider security at all. They also provide an example using a simple three-tiered architecture. The basic idea behind the way that CLASP handles security requirements is the performance of a structured walkthrough of resources, determining how they address each core security service throughout the lifetime of that resource. Although it is obviously far more effective than an *ad hoc* treatment of security requirements, this methodology is still new and it has been published in conjunction with IBM/Rational.

4.21. Yu "Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering" [59], Yu et al. "A Social Ontology for Integrating Security and Software Engineering" [60] and Yu "Social Modeling and i" [61]*

The author states that understanding the organizational context and rationales (the "Whys") that lead up to systems requirements can be just as important for the ongoing success of the system. Most existing requirements techniques are intended more for the later phase of requirements engineering, which focuses on completeness, consistency, and automated verification of requirements. In contrast, the early phase aims to model and analyze stakeholder interests and how they might be addressed, or compromised, by various system-and-environment alternatives. The author therefore argues that a different kind of modelling and reasoning support is needed for the early phase, and he thus provides i* framework, which was developed to model and reason about organizational environments and their information systems. The structural representation defined in i* shows the dependence relationships between the actors, and is what makes security aspects appear. The authors argue [60,61] that a social ontology at the core of a requirements engineering process could be the basis for integrating security into a requirements driven software engineering process.

4.22. Zuccato "Holistic security requirement engineering for electronic commerce" [62] and "Holistic security management framework applied in electronic commerce" [63]. Zuccato et al. "Security Requirements Engineering at a Telecom Provider" [64]

Zuccato's approach is intended to elicit security requirements according to system-theoretic considerations. It shows that security requirements can be defined with the help of investigations in the business environment, workshops with stakeholders and risk analysis. This multidimensional approach leads to a holistic understanding of the requirements that fit into the system development lifecycle. A process is proposed which is based on the new definition of a holistic security requirement and which relies on the process description patterns used in the Unified Process. The authors propose a security engineering method that is well-integrated into the development process [64] and is called SKYDD According to the authors, this simplifies security requirement gathering, reduces lead times and provides consistent requirements. However, it does not provide any new specific techniques to deal with security requirements, the strongest in their process currently being the information domain, closely followed by the infrastructure domain and finally the business domain.

5. Results and discussion

The results of the systematic review are shown in Table 1 which summarizes the quantity of studies per initiative. Table 2 then shows the main contributions, in terms of security requirements, of each selected initiative.

As Table 1 shows, there are many new techniques, processes and methodologies which attempt to facilitate the management of the security requirements in the software development life-cycle in order to develop more secure software. Nevertheless, very few works describe complex case studies that show the possibility of using a security requirements methodology in practice. Moreover, as can be seen in Table 2, after our analysis we have reached the conclusion that each of the selected initiatives provides us with highly important aspects that have to do with security requirements engineering. These are features that can be used as the basis for new methodologies / processes / frameworks / techniques, or as extensions to those that already exist.

A comparison of the initiatives is summarized in Table 3 with the aim of analyzing the selected initiatives and we have thus used an analytical framework (partially based on the framework proposed by Khwaja and Urban [72]), which has the following technical criteria:

- Internal verification support. Automatic internal verification in order to assure the consistency, completeness, and the rest of properties of a good requirements specification of the security requirements according to IEEE 830 [73].
- External validation support. This may ensure the correctness of the security requirements through their validation by external stakeholders.
- Support for documentation generation. Documentation generation from security requirements specifications may help us increase their understandability.
- Standards integration. The most important requirements and security standards must be integrated to improve the consistency and verifiability of the security requirements.
- Requirements reuse. Reusability of security requirements elements (assets, threats, security objectives, security requirements, etc.) from one project to another in order to help the elicitation of security requirements and to successively improve their quality.
- Support for other development stages. Automatic integration of the security requirements with the other requirements (functional and non-functional) and phases (mainly with analysis, design and/or testing phase) must be provided. Thus, this criterion will measure whether the approaches combine well with these phases, for example by offering ideas on how to go from security requirements to secure architectures or by proposing ways to achieve security test driven development.
- Help support. This deals with aspects such as standards, guidelines, case studies, CARE (Computer Aided Requirements Engineering) and tools available for a methodology / technique, in order to measure the feasibility of and facilities with which to apply the technique / process / methodology in practice.
- Easy to use. This deals with the ease with which a technique/methodology/process may be used, without much knowledge or

Table 1
Summary of the quantity of studies per initiative.

Type of initiative	# of studies	Initiatives
Technique	5	4.1, 4.3, 4.14, 4.15, 4.17
Framework	6	4.4, 4.8, 4.11, 4.19, 4.20, 4.21
Process	6	4.6, 4.7, 4.10, 4.13, 4.16, 4.18, 4.22
Methodology	3	4.2, 4.5, 4.9
Others	1	4.12
Total	22	–

Table 2
Summary of the contributions.

Initiative	Requirements modeling / Elicitation technique	Model/Standard of Development	Integration of Standards	Main Contributions
Basin et al.	SecureUML (UML, OCL)	UML, Model Driven Architecture paradigm	–	♦ SecureUML (security modelling language for formalizing access control requirements based on UML)
Bresciani et al. and Giorgini et al. and Massacci et al.	Tropos language, Secure Tropos	Agent oriented software development	ISO/IEC 17799-	♦ Methodology Tropos ♦ Framework for modelling and analyzing security and trust requirements
Firesmith	Security use cases	Conducted by assets and risk	ISO/IEC 9126-1 and 9126-2	♦ Security use cases (UML extension for modelling security requirements in use case diagrams)
Hussein and Zulkernine	UMLintr	Component-Based Software Engineering (CBSE)	–	♦ UMLintr (UML profile for intrusion identifications)
Jennex	Barrier analysis and defense in depth	Barrier analysis and defense in depth	–	♦ Barrier analysis and in depth defense as a security requirements identification and design methodology
Lamsweerde	KAOS	Goal-oriented requirements	–	♦ use of antimodels to elaborate security requirements
J. Lee, et al.	–	CC and SSE-CMM	ISO/IEC 15408 (CC), ISO/IEC 21827 (SSE-CMM)	♦ CC_SSE-CMM: A Common Criteria based Security Engineering Process Evaluation Model
S.-W. Lee et al.	Organization diagram, req categorization, Questionnaire, GENeric Object Model (GenOM)	–	U.S.A. Department of Defense Directives 8500.1 and 8500.2	♦ Problem Domain Ontology
Mead and Stehney	Use/misuse cases, etc. (selection of elicitation technique)	Sequential steps	–	♦ SQUARE: 9-step process for eliciting, categorizing, and prioritizing security requirements
Mellado et al.	UMLSec, misuse cases, security use cases, aspect XML	UML, Unified Process, conducted by actives or threats and the risk. Iterative and incremental	ISO/IEC 15408, ISO/IEC 27001, ISO/IEC 17799, ISO/IEC 13335, IEEE 830-1998	♦ Integration of the Common Criteria ♦ Security requirements reuse ♦ Integration of the latest security techniques.
Moffett and Nuseibeh	Constraints	Conducted by security goals	–	♦ Framework which unifies the concepts of requirements engineering and security engineering
Morimoto, et al.	–	–	ISO/IEC 15408	♦ Information Security Engineering Database System based on ISO/IEC 15408
Myagmar et al.	–	Threat modelling	–	♦ Threat modelling as a basis for security requirements
Peeters	Abuser stories	Agile requirements engineering	–	♦ Abuser stories
Popp et al. and Jürjens	UMLSec	UML, Unified Process	–	♦ UMLSec (UML extension for secure systems development)
Shin and Gomaa	security use cases and security objects, secure connectors	Distributed software architectures using components and connectors	–	♦ Security requirement conditions ♦ Separating application concerns from security concerns
Sindre and Opdahl, Whittle et al.	Misuse cases	Conducted by threats and risk	–	♦ Misuse cases (UML extension for modeling threats in use case diagrams). Executable misuse cases
Toval et al.	Plain text (Although it admits others)	Spiral process. Conducted by activities and risk	IEEE 830-1998, IEEE-1233, IEEE- 1207.1 and partially ISO/IEC 15408	♦ SIREN: Re-use of security requirements compatible with MAGERIT (conforms to ISO/IEC 15408)
Tsoumas and Gritzalis	Security ontology	ontologies	CRAMM, COBIT	♦ A security ontology which extends the DMTF Common Information Model (CIM)
Viega	Tables ('must')	Resource-centric	–	♦ CLASP: handles security requirements through a structured walkthrough of resources
Yu	Strategic Dependency and Strategic Rationale models	Business process modelling and redesign and software process modelling	–	♦ i* framework for modelling and reasoning about organizational environments and their information systems
Zuccato	Business Process Modelling	Unified Process	ISO/IEC 15408, ISO/IEC 17799, ISO 9000:2000, ISO/IEC 13335	♦ Elicit security requirements according to system-theoretic considerations

special training, in order to assess the learning curve for practitioners. It is also partially related to the 'Help support' criterion.

The specification criteria proposed are the following: understandable (comprehensibility of the model), unambiguous, complete, consistent, correct, verifiable (analysability), validateable (testability), modifiable (maintainability, adaptability), traceable, appropriate. The degree of fulfilment will be "*" for Yes, "P" for partially and "X" for No.

Having performed a high level analysis of Table 3, we shall now suggest which technique to adopt depending on the combination of the importance of each criterion defined in this table. These suggestions have been formulated in an attempt to be general, and with the aim of covering the most common choices with regard to security requirements in decision making of a high level of abstraction. In IS, in which ease of use or agility are essential characteristics, the proposals which completely fulfil the criterion "Easy to use" would be the most appropriate. Moreover, in IS, in which it is critical to achieve a security certification in a security standard (for example in IS for the Ministry of Defence or for NATO), the best approaches would be those that completely fulfil the criterion "Standard integration" and simultaneously consider the desired security standard. In addition, if it is important to consider security throughout the entire software lifecycle development, i.e., to achieve security traceability, the most appropriate proposals would be those which fulfil the criterion

Table 3
Comparison of initiatives.

Technical criterion		Internal verification support								External validation support		Support for documentation generation
Specification criteria		Correct	Unambiguous	Modifiable	Validateable	Complete	Consistent	Traceable	Order by importance	Correct	Validatable	Understandable
INITIATIVES	Basin et al	*	*	*	P	P	*	*	X	X	X	*
	Bresciani et al. and Giorgini et al. and Massaci et al	*	*	*	*	*	*	*	*	*	*	*
	Firesmith	*	*	*	P	*	*	P	*	*	X	*
	Hussein and Zulkernine	*	*	*	*	P	*	P	*	P	X	P
	Jennex	*	*	*	*	*	P	X	X	P	X	*
	Lamsweerde	*	*	*	*	*	*	P	X	*	*	*
	J. Lee, et al.	*	*	*	*	*	*	*	X	*	*	*
	S.-W. Lee et al.	*	*	*	*	*	*	*	*	X	X	*
	Mead and Stehney	*	*	*	*	*	*	*	*	*	*	P
	Mellado et al.	*	*	*	*	*	*	*	*	*	*	*
	Moffett and Nuseibeh	*	*	*	*	*	*	*	P	*	*	*
	Morimoto, et al	*	*	*	P	*	*	*	X	X	X	P
	Myagmar et al.	P	P	P	P	P	P	P	X	X	X	P
	Peeters	P	P	P	P	P	P	P	X	X	X	*
	Popp et al. And Jürjens	*	*	*	*	P	*	P	*	*	X	*
	Shin and Gomaa	*	*	*	*	*	*	*	P	X	X	X
	Sindre and Opdahl. Sindre et al. Whittle et al.	*	*	*	*	*	P	P	*	*	X	P
	Toval et al	*	*	*	*	*	*	*	*	*	P	*
	Tsoumas and Gritzalis	*	*	*	*	*	*	*	*	*	P	P
	Viega	*	*	*	*	*	*	*	*	*	P	*
	Yu	*	*	*	*	*	*	P	X	X	X	*
	Zuccato	*	*	*	*	*	*	*	*	*	*	*

"Support for other development Stages", because these approaches would consider security requirements and would simultaneously facilitate security requirements consistency by combining with functional analysis techniques, by making it easier to go from security requirements to secure architectures or by facilitating security test driven development. Furthermore, in complex IS and large organizations reusability and the integration of the security requirements with the other requirements and other stages, along with the use of supporting tools might be important; the most appropriate approaches might therefore be those that fulfil the criteria of "Requirements reuse", "Support for other development stages" and "Help support". Finally, in large and complex projects in which security is a critical issue, it is essential to have a good security requirements specification, and the proposals which fulfil the criteria "Internal verification support", "External verification support" and "Support for documentation generation" would therefore be the most appropriate.

However, as the aforementioned suggestions show, the choice of these techniques in concrete projects would necessitate a profound and thorough study of the preferences of the software development team, owing to the multitude of variables that must be taken into account in a concrete software development project, and the difficulty of generalizing the selection with regard to certain parameters. However, we believe that this could be considered as a future field of research.

As Table 3 shows, after our analysis we reached the conclusion that despite the fact that in the last few years several security standards (such as ISO/IEC 15408, ISO/IEC 27001, etc.) and security requirements techniques (such as UMLsec, security use cases, etc.) have been developed, and that they assist in the task of developing secure information systems, it is very difficult to develop a methodology / process that fulfils all the criteria and comprises all the security constraints. There are not, therefore, many methodologies / processes that incorporate them in an intuitive and systematic way, or provide a complete integration with the functional requirements and the other non-functional requirements. In addition, if that methodology were to be developed, its complexity might prevent its success, mainly because it would be extremely difficult to obtain a single process which was valid for any software development project, owing to the variety of the software itself and the variety of its contexts.

Hence, the solution could be an adaptable approach (a new approach or an adaptation of those previously analyzed) in which security requirements techniques and models defined by the most frequently accepted security standards are used depending on the needs of the software being developed, signifying that the approach would provide guidance as to which are the more appropriate processes or steps to follow and how to fulfil the related security standards and incorporate the appropriate techniques. The security requirements would thus be dealt with during the early stages of software development in a systematic and intuitive way, using the appropriate techniques and fulfilling the related security standards. This approach would need a hierarchy of categories in order to categorize the software development project with the aim of facilitating the choice of the appropriate methodology or process and its respective techniques by proposing those related to the selected categories. Furthermore, this approach could be based on a meta-model with the different security elements (asset, threat, security objective, security requirement, countermeasure, security standard) which would be related to concrete techniques depending on the categorization of the software development project.

The research results suggest that new initiatives to incorporate standards and new security requirements techniques, along with a systematic and intuitive integration of the security requirements into the software development life-cycle should be developed, that is, new proposals that methodologically support the fulfilment of standards and incorporate new techniques during the development process must be developed. Furthermore, real and complex case studies which

Standards integration				Requirements reuse			Support for other development stages			Help support	Easy to use
Understandable	Consistent	Verifiable	Complete	Consistent	Modifiable	Appropriate	Complete	Traceable	Modifiable	———	———
X	X	X	X	X	X	X	P	P	*	*	*
P	P	P	P	*	*	*	*	*	*	*	*
X	X	X	X	*	*	P	P	*	P	*	*
X	X	X	X	*	*	P	P	P	*	*	P
X	X	X	X	X	X	X	P	P	X	*	*
X	X	X	X	P	*	*	P	P	*	*	*
*	*	*	*	X	X	X	P	*	*	*	P
*	*	*	*	*	*	P	*	*	*	X	P
*	*	*	P	X	X	X	*	*	*	P	*
*	*	*	*	*	*	*	*	*	*	P	*
X	X	X	X	X	X	X	*	*	*	*	*
*	*	*	P	*	*	*	X	X	X	*	*
X	X	X	X	X	X	X	X	X	X	*	*
X	X	X	X	X	X	X	P	P	P	*	*
X	X	X	X	X	X	X	P	P	*	*	*
X	X	X	X	*	*	X	P	P	*	X	P
X	X	X	X	*	*	P	P	*	P	*	*
*	*	X	P	*	*	*	P	*	*	*	*
*	*	*	*	*	*	*	P	*	*	*	P
X	X	X	X	X	X	X	P	*	P	*	*
X	X	X	X	X	X	X	P	*	*	*	*
*	*	*	P	X	X	X	*	P	*	P	*

show the ability to use the aforementioned methodologies in practice must be developed.

This review provides us with consistent results about security requirements which cannot be refuted, since it has been conducted in a systematic, formal and unbiased manner, thus permitting other professionals to reproduce the same protocol and allowing them to judge the adequacy of the results obtained. In contrast to former reviews, such as [3,10–13,74] which, despite their being conducted according to their corresponding 'good practice', suffer from a lack of scientific rigor in the performance of their different steps, and therefore create some research biases at different stages of the review process [17]. What is more, most of these reviews are solely a section within the paper/article, and very few papers exist in which a review of security requirements is the core. Moreover, none of these reviews explain the method or process which has been followed to select the initiatives which are presented in them. On the other hand, this systematic review of security requirements does not consist of a simple rearrangement of the data relating to this issue which is already known or has been published, but has been performed by following a formal and controlled process for conducting this type of investigation (it is based on the guidelines for systematic reviews proposed by Kitchenham [14–16] and Biolchini et al [17]). Finally, despite requiring more effort than traditional reviews, systematic reviews provide us with more benefits than drawbacks. Thus, we have presented a fair evaluation of security requirements engineering by using a trustworthy, rigorous and auditable methodology.

6. Conclusions

Information Security is usually only tackled from a technical viewpoint during the implementation stage, and although this is an important aspect, we believe it is fundamental to deal with security in all stages of IS development, especially during the establishment of security requirements, since these form the basis of the achievement of a robust IS, since it is also acknowledged that the first steps of software development are essential for the success of a software development project [75].

The contribution of this work is consequently that of supplying researchers with a summary of all existing information about security requirements in a thorough and unbiased manner, so as to provide a context in which to operate. The main contribution of this work in comparison to former traditional reviews is therefore the precision and reliability of the information and the results obtained. Nevertheless, its main limitation is that systematic reviews have only appeared in Software Engineering very recently, and Software Engineering still has some specific issues that make it difficult for the research synthesis to obtain evidence. This systematic review might therefore be biased owing to the fact that the available evidence relating to software engineering technologies is: fragmented and limited, there is not a central information source for evidence; it is not properly integrated, there are no agreed standards for systematic reviews; and there are no generally accepted guidelines or standard protocols for conducting individual experiments.

Moreover, we should highlight that the most important lesson learned was that searching for, selecting and evaluating studies about software engineering still represents a bottleneck in systematic reviews and this may be a source of bias in them.

Acknowledgments

This research is part of the ELEPES (TIN2006-27690-E) Project of the Ministry of Education and Science (Spain), the BUSINESS (PET2008-0136) Project financed by the Ministry of Science and Innovation (Spain), the SISTEMAS (PII2I09-0150-3135) and QUASIMODO (PAC08-

0157-0668) Projects both financed by the Regional Science and Technology Ministry of Castilla-La Mancha (Spain), and the MEDUSAS (IDI-20090557) Project financed by the Centre for Technological and Industry Development (CDTI) (Spain).

References

[1] J.P. Walton, Developing a enterprise information security policy, ACM Press: Proceedings of the 30th annual ACM SIGUCCS conference on User services, 2002.

[2] Choo, K.-K.R., R.G. Smith, and R. McCusker, *Future directions in technology-enabled crime: 2007–09*, in *Research and Public Policy Series*, Australian_Government, Editor. 2007, Australian Institute of Criminology

[3] M. Zulkernine, S.I. Ahamed, Software security engineering: toward unifying software engineering and security engineering, in: M. Warkentin, R.B. Vaughn (Eds.), Enterprise Information Systems Assurance and System Security, Idea Group Publishing, 2006.

[4] Konrad, S., B.H.C. Chengy, L.A. Campbell, and R. Wassermann, Using Security Patterns to Model and Analyze Security Requirements, in High Assurance Systems Workshop (RHAS 03) as part of the IEEE Joint International Conference on Requirements Engineering (RE 03): Monterey Bay, CA (USA).

[5] J. Viega, Secure Software Inc, V.A. McClean, Building security requirements with CLASP, Proceedings of the 2005 workshop on Software engineering for secure systems—building trustworthy applications, 2005, pp. 1–7.

[6] D.G. Firesmith, Specifying reusable security requirements, Journal of Object Technology, 2004, pp. 61–75.

[7] J. Kim, M. Kim, S. Park, Goal and scenario bases domain requirements analysis environment, The Journal of Systems and Software, 2005, pp. 926–938.

[8] G. Kotonya, I. Sommerville, Requirements engineering process and techniques, Hardcover ed., 294, John Willey & Sons, UK, 1998.

[9] J. McDermott, C. Fox, Using abuse case models for security requirements analysis, Annual Computer Security Applications Conference, Phoenix, Arizona, 1999.

[10] R.R. Henning, H. Corporation, Security engineering: it is all about control and assurance objectives, in: M. Warkentin, R.B. Vaughn (Eds.), Enterprise Information Systems Assurance and System Security, Idea Group Publishing, 2006.

[11] R. Villarroel, E. Fernández-Medina, M. Piattini, Secure information systems development — a survey and comparison, Computers & Security, 2005, pp. 308–321.

[12] D. Mellado, E. Fernández-Medina, M. Piattini, A comparative study of proposals for establishing security requirements for the development of secure information systems, The 2006 International Conference on Computational Science and its Applications (ICCSA 2006), Springer LNCS 3982, 3, 2006, pp. 1044–1053.

[13] J.D. Moffett, B.A. Nuseibeh, A framework for security requirements engineering, Report YCS, Department of Computer Science, University of York, 2003, p. 368.

[14] B. Kitchenham, Procedures for Perfoming Systematic Review, Joint Technical Report, Software Engineering Group, Department of Computer Science Keele University, United Kingdom and Empirical Software Engineering, National ICT Australia Ltd.: Australia, 2004.

[15] P. Brereton, B. Kitchenham, D. Budgen, M. Turner, M. Khalil, Lessons from applying the systematic literature review process within the software engineering domain, J. Syst. Software 80 (4) (2007) 571–583.

[16] Kitchenham, B., *Guideline for performing Systematic Literature Reviews in Software Engineering. Version 2.3*. 2007, University of Keele (Software Engineering Group, School of Computer Science and Mathematics) and Durham (Department of Computer Science).

[17] J. Biolchini, P.G. Mian, A.C.C. Natali, G.H. Travassos, Systematic review in software engineering, Systems Engineering and Computer Science Department COPPE / UFRJ: Rio de Janeiro, 2005.

[18] D.G. Firesmith, Engineering security requirements, Journal of Object Technology 2 (1) (2003) 53–68.

[19] D. Basin, J. Doser, T. Lodderstedt, Model-driven security for process-oriented systems. SACMAT'03, , 2003, pp. 100–110.

[20] D. Basin, J. Doser, T. Lodderstedt, Model driven security: from UML models to access control infrastructures, ACM Trans. Softw. Eng. Methodol. 15 (1) (2006) 39–91.

[21] P. Bresciani, P. Giorgini, F. Giunchiglia, J. Mylopoulos, A. Perini, Tropos: agent-oriented software development methodology, Journal of Autonomous Agents and Multi-Agent System, 2004, pp. 203–236.

[22] P. Giorgini, F. Massacci, J. Mylopoulos, N. Zannone, Requirements engineering meets trust management: model, methodology, and reasoning. iTrust 2004, , 2004, pp. 176–190.

[23] P. Giorgini, H. Mouratidis, N. Zannone, *Modelling Security and Trust with Secure Tropos*, in *Integrating Security and Software Engineering: Advances and Future Visions*, , Idea Group Publishing, 2006.

[24] R. Ali, F. Dalpiaz, P. Giorgini, Location-based software modeling and analysis: Tropos-based approach, in 27th International Conference on Conceptual Modeling (ER 08), Springer, Barcelona, Spain, 2008.

[25] R. Ali, F. Dalpiaz, P. Giorgini, A goal modeling framework for self-contextualizable software, in 14th international conference on exploring modeling methods in systems analysis and design (EMMSAD09), Springer, Amsterdam, The Netherlands, 2009.

[26] F. Dalpiaz, P. Giorgini, J. Mylopoulos, *An Architecture for Requirements-Driven Self-reconfiguration*, in *CAiSE*, 2009, pp. 246–260.

[27] F. Massacci, M. Prest, N. Zannone, Using a security requirements engineering methodology in practice: the compliance with the Italian data protection legislation, in Computers Standards and Interfaces, 2005, pp. 445–455.

[28] L. Compagna, P.E. Khoury, A. Krausová, F. Massacci, N. Zannone, How to integrate legal requirements into a requirements engineering methodology for the development of security and privacy patterns, Artif. Intell. Law 17 (1) (2009) 1–30.

[29] D.G. Firesmith, Engineering safety-related requirements for software-intensive systems, in Proceedings of the 27th international conference on Software engineering, ACM Press, St. Louis, MO, USA, 2005.

[30] D.G. Firesmith, Engineering safety and security related requirements for software intensive systems, in international conference on software engineering, IEEE Computer Society, 2007, p. 169.

[31] M. Hussein, M. Zulkernine, Intrusion detection aware component-based systems: a specification-based framework, J. Syst. Softw. 80 (5) (2007) 700–710.

[32] M.E. Jennex, Modeling security requirements for information systems development, SREIS, 2005.

[33] J. Lee, J. Lee, S. Lee, B. Choi, A CC-based Security Engineering Process Evaluation Model. 27th Annual International Computer Software and Applications Conference (COMPSAC'03), 2003, p. 130.

[34] S.-W. Lee, R. Gandhi, D. Muthurajan, D. Yavagal, G.-J. Ahn, Building problem domain ontology from security requirements in regulatory documents, in Proceedings of the 2006 international workshop on Software engineering for secure systems, ACM Press, Shanghai, China, 2006.

[35] N.R. Mead, T. Stehney, Security Quality Requirements Engineering (SQUARE) Methodology. in Software Engineering for Secure Systems (SESS05), ICSE 2005 International Workshop on Requirements for High Assurance Systems, , 2005, St. Louis.

[36] N.R. Mead, E.D. Hough, *Security Requirements Engineering for Software Systems: Case Studies in Support of Software Engineering Education*, in *CSEE&T*, 2006, pp. 149–158.

[37] S. Abu-Nimeh, S. Miyazaki, N.R. Mead, Integrating privacy requirements into security requirements engineering, SEKE (2009) 542–547.

[38] D. Mellado, E. Fernández-Medina, M. Piattini, A common criteria based security requirements engineering process for the development of secure information systems, Computer Standards and Interfaces, 2007, pp. 244–253.

[39] D. Mellado, E. Fernández-Medina, M. Piattini, Towards security requirements management for software product lines: a security domain requirements engineering process, Computer Standards & Interfaces, 2008, pp. 361–371.

[40] C.B. Haley, R. Laney, J.D. Moffet, B. Nuseibeh, Security requirements engineering: a framework for representation and analysis, IEEE Trans. Software Eng. 34 (1) (2008) 133–153.

[41] S. Morimoto, D. Horie, J. Cheng, A security requirement management database based on ISO/IEC 15408, ICCSA 2006 (LNCS 3982), 3, 2006, pp. 1–10.

[42] D. Horie, S. Morimoto, N. Azimah, Y. Goto, J. Cheng, ISEDS: an information security engineering database system based on ISO Standards, ARES, 2008, pp. 1219–1225.

[43] S. Myagmar, A.J. Lee, W. Yurcik, Threat modeling as a basis for security requirements, SREIS, 2005.

[44] J. Peeters, Agile security requirements engineering, SREIS, 2005.

[45] G. Popp, J. Jürjens, G. Wimmel, R. Breu, Security-critical system development with extended use cases, 10th Asia-Pacific Software Engineering Conference, 2003, pp. 478–487.

[46] J. Jürjens, UMLsec: extending UML for secure systems development. UML, The Unified Modeling Language. Model Engineering, Languages,Concepts, and Tools. 5th International Conference., 2002. **LNCS 2460**, 2002, pp. 412–425.

[47] J. Jürjens, J. Schreck, Y. Yu, Automated analysis of permission-based security using UMLsec, Fundamental Approaches to Software Engineering (FASE 2008), held as part of the Joint European Conferences on Theory and Practice of Software (ETAPS 2008), 2008, pp. 292–295.

[48] M.E. Shin, H. Gomaa, Software requirements and architecture modeling for evolving non-secure applications into secure applications, Sci. Comput. Program. 66 (1) (2007) 60–70.

[49] G. Sindre, A.L. Opdahl, Eliciting security requirements with misuse cases, Requirements Eng. 10 (1) (2005) 34–44.

[50] G. Sindre, D.G. Firesmith, A.L. Opdahl, A reuse-based approach to determining security requirements. in Proc. 9th International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ'03), , 2003, Austria.

[51] Opdahl, A.L. and G. Sindre, *Experimental comparison of attack trees and misuse cases for security threat identification.* Information and Software Technology. In Press, Corrected Proof, 2008.

[52] T. Stalhane, G. Sindre, Safety hazard identification by misuse cases: experimental comparison of text and diagrams, MODELS, 2008, pp. 721–735.

[53] A. Toval, J. Nicolás, B. Moros, F. García, Requirements reuse for improving information systems security: a practitioner's approach, Requirements Engineering Journal, 2001, pp. 205–219.

[54] M. Martínez, J. Lasheras, A. Toval, M. Piattini, An audit method of personal data based on requirements engineering, 4th International Workshop on Security in Information Systems - WOSIS, , 2006.

[55] J. Nicolás, J. Lasheras, A. Toval, F. Ortíz, B. Alvarez, A collaborative learning experience in modelling the requirements of teleoperated systems for ship hull maintenance, in workshop on learning software organizations and requirements engineering, LSO, 2006.

[56] J. Lasheras, R. Valencia-García, J.T. Fernández-Breis, A. Toval, An ontology-based framework for modelling security requirements, in The 6th International Workshop on Security in Information Systems — WOSIS, 2008.

[57] B. Tsoumas, D. Gritzalis, Towards an ontology-based security management. Proceedings of the 20th International Conference on Advanced Information Networking and Applications. IEEE Computer Society, 2006, Volume 1 (AINA'06) - Volume 01 AINA '06.

[58] B. Tsoumas, P. Papagiannakopoulos, S. Dritsas, D. Gritzalis, Security-by-ontology: a knowledge-centric approach, in: S. Boston (Ed.), Security and Privacy in Dynamic Environments, 2006, pp. 99–110.

[59] E. Yu, Towards modelling and reasoning support for early-phase requirements engineering, 3 rd IEEE International Symposium on Requirements Engineering (RE'97), 1997, pp. 226–235.
[60] E. Yu, L. Liu, Mylopoulos, A social ontology for integrating security and software engineering, in integrating security and software engineering: advances and future visions, Idea Group Publishing, 2006.
[61] E. Yu, Social modeling and i*, in: A.T. Borgida, et al., (Eds.), Conceptual Modeling: Foundations and Applications - Essays in Honor of John Mylopoulos, Springer, 2009, pp. 99–121.
[62] A. Zuccato, Holistic security requirement engineering for electronic commerce, Computers and Security, 2004, pp. 63–76.
[63] A. Zuccato, Holistic security management framework applied in electronic commerce, Computer & Security 26 (3) (2007) 256–265.
[64] A. Zuccato, V. Endersz, N. Daniels, Security requirements engineering at a telecom provider, ARES, 2008, pp. 1139–1147.
[65] V. Lamsweerde, Engineering requirements for system reliability and security, in software system reliability and security, in: M. Broy, J. Grunbauer, C.A.R. Hoare (Eds.), NATO security through science series-D: information and communication security, IOS Press, 2007, pp. 196–238.
[66] D.G. Firesmith, Security use cases, Journal of Object Technology (2003) 53–64.
[67] B. Best, J. Jürjens, B. Nuseibeh, Model-based security engineering of distributed information systems using UMLSec, ICSE, 2007, pp. 581–590.
[68] J. Whittle, D. Wijesekera, M. Hartong, Executable misuse cases for modeling security concerns, International Conference on Software Engineering (ICSE'08), 2008, pp. 121–130.
[69] F.A. Braz, E.B. Fernandez, M. VanHilst, Eliciting security requirements through misuse activities, 19th International Conference on Database and Expert Systems Application (DEXA 2008), 2008, pp. 328–333.
[70] CRAMM, CRAMM, United Kingdom Central Computer and Telecommunication Agency. CCTA Risk Analysis and Management Method: User Manual, ver. 5.1, HMSO, 2005.
[71] COBIT, COBIT, IT Governance Institute. Control Objectives for Information and related Technology (COBIT 4.0), 2005.
[72] A. Khawaja, J. Urban, A synthesis of evaluation criteria for software specifications and specifications techniques, International Journal of Software Engineering and Knowledge Engineering 12 (5) (2002) 581–599.
[73] IEEE, IEEE 830: 1998 recommended practice for software requirements specifications, 1998.
[74] N.R. Mead, How to compare the Security Quality Requirements Engineering (SQUARE) method with other methods, , 2007, Volume.
[75] D. Hatebur, M. Heisel, H. Schmidt, A formal metamodel for problem frames, 11th international conference on Model Driven Engineering Languages and Systems (ARES'08), Lecture Notes In Computer Science, Vol. 5301, 2008, pp. 68–82.

Daniel Mellado is PhD and MSc in Computer Science from the Castilla- La Mancha University (Spain) and Autonomous University of Madrid (Spain), and Certified Information System Auditor by ISACA (Information System Audit and Control Association). He is an Assistant Professor of the Department of Information Technologies and Systems at the Castilla- La Mancha University in Toledo (Spain). He participates at the ALARCOS research group of the Department of Information Technologies and Systems at the University of Castilla- La Mancha. He is civil servant at the Spanish Tax Agency (in Madrid, Spain), where he works as IT Auditor. His research activities are security requirements engineering, security in information systems, secure software process improvement and auditory, quality and product lines. He has several dozens of papers in national and international conferences, journals and magazines on these subjects and co-author of several chapter books. He belongs to various professional and research associations (ASIA, ISACA, ASTIC, ACTICA, etc).

Carlos Blanco has an MSc in Computer Science from the University of Castilla-La Mancha. He is currently a PhD student and a member of the ALARCOS research group at the School of Computer Science at the University of Castilla-La Mancha (Spain). His research activity is in the field of security in Information Systems focused on data warehouses, OLAP tools, MDA and ontologies. He is author of several papers on these topics.

Luis Enrique Sánchez is PhD and MsC in Computer Science and is an Assistant Professor at the Escuela Superior de Informática of the Universidad de Castilla- La Mancha at Ciudad Real (Spain), MSc in Information Systems Audit from the Politechnical University of Madrid, and Certified Information System Auditor by ISACA (Information System Audit and Control Association). He is Director of Professional Services and R+D departments of the company Sicaman Nuevas Tecnologías S.L. COIICLM board or committee member and responsible for their professional services committee. His research activities are management security system, security metrics, data mining, data cleaning, and business intelligence. He participates at the ALARCOS research group of the Department of Computer Science at the University of Castilla- La Mancha, in Ciudad Real (Spain). He belongs to various professional and research associations (COIICLM, ATI, ASIA, ISACA, eSEC, INTECO, etc).

Eduardo Fernández-Medina is PhD and MSc in Computer Science. He is an Associate Professor at the Escuela Superior de Informática of the Universidad de Castilla- La Mancha at Ciudad Real (Spain). His research activities are security requirements, security in databases, data warehouses, web services and information systems, and also in security metrics. He is the co-editor of several books and chapter books on these subjects, and has several dozens of papers in national and international conferences. He participates at the ALARCOS research group of the Department of Information Technologies and Systems at the University of Castilla- La Mancha, in Ciudad Real (Spain). He belongs to various professional and research associations (ATI, AEC, AENOR, IFIP, WG11.3, etc).