SECURITY REQUIREMENTS IN SOFTWARE PRODUCT LINES

Daniel Mellado
Ministry of Work and Social Affairs, Social Security IT Department, Madrid, Spain

Eduardo Fernández-Medina and Mario Piattini
University of Castilla La-Mancha, Information Systems and Technologies Department, Alarcos Research Group, Spain

Keywords: Security requirements, product lines, Common Criteria, Security.

Abstract: Proper analysis and understanding of security requirements are important because they help us to discover any security or requirement defects or mistakes in the early stages of development. Hence, security requirements engineering is both a central task and a critical success factor in product line development due to the complexity and extensive nature of product lines. However, most of the current product line practices in requirements engineering do not adequately address security requirements engineering. Therefore, in this paper we will propose a security quality requirements engineering process (SREPPLine) driven by security standards and based on a security requirements decision model along with a security variability model to manage the variability of the artefacts related to security requirements. The aim of this approach is to deal with security requirements from the early stages of the product line development in a systematic way, in order to facilitate conformance with the most relevant security standards with regard to the management of security requirements, such as ISO/IEC 27001 and ISO/IEC 15408.

1 INTRODUCTION

In the search for improved software quality and high productivity, software product line (SPL) engineering has proven to be one of the most successful paradigms for developing a diversity of similar software applications and software-intensive systems at low costs, in a short time, and with high quality, by exploiting commonalities and variabilities among products to achieve high levels of reuse (Bosh 2000; Clements et al., 2002).

In software intensive systems, such as SPL, security is a cross-cutting concern and should consequently be subject to careful requirements analysis and decision making. Moreover, in SPL engineering, security is one of the most important attributes concerning quality, given that a weakness in security may cause problems in all the products in a product line. In addition, many requirements engineering practices must be appropriately tailored to the specific demands of product lines (Birk et al., 2007). Hence, specifying requirements for a SPL is a challenging task (Niemelä et al., 2007) and specifying security quality requirements for an SPL is even more challenging due to the varying security properties required in different products.

Therefore, the discipline known as Security Requirements Engineering is essential for secure SPL and products development, because it provides techniques, methods, standards and systematic and repeatable procedures for tackling SPL security requirement issues throughout the SPL development lifecycle both to ensure the definition of security quality requirements and to manage the variability of security properties. Nevertheless, software engineering methodologies and standard proposals of SPL engineering have traditionally ignored security requirements and security variability issues. Although some of them include a few security requirements activities, most of them focus only on the design of implementation aspects of SPL development.

In this paper, as an evolution of our previous “generic” security requirements engineering process (SREP) (Mellado et al., 2006), we shall present the Product Line Security Application Requirements Engineering (PLSecAppReq) subprocess together with the Security Requirements Variability Model

Mellado D., Fernández-Medina E. and Piattini M. (2008).
SECURITY REQUIREMENTS IN SOFTWARE PRODUCT LINES.
In *Proceedings of the International Conference on Security and Cryptography*, pages 442-449
DOI: 10.5220/0001922804420449

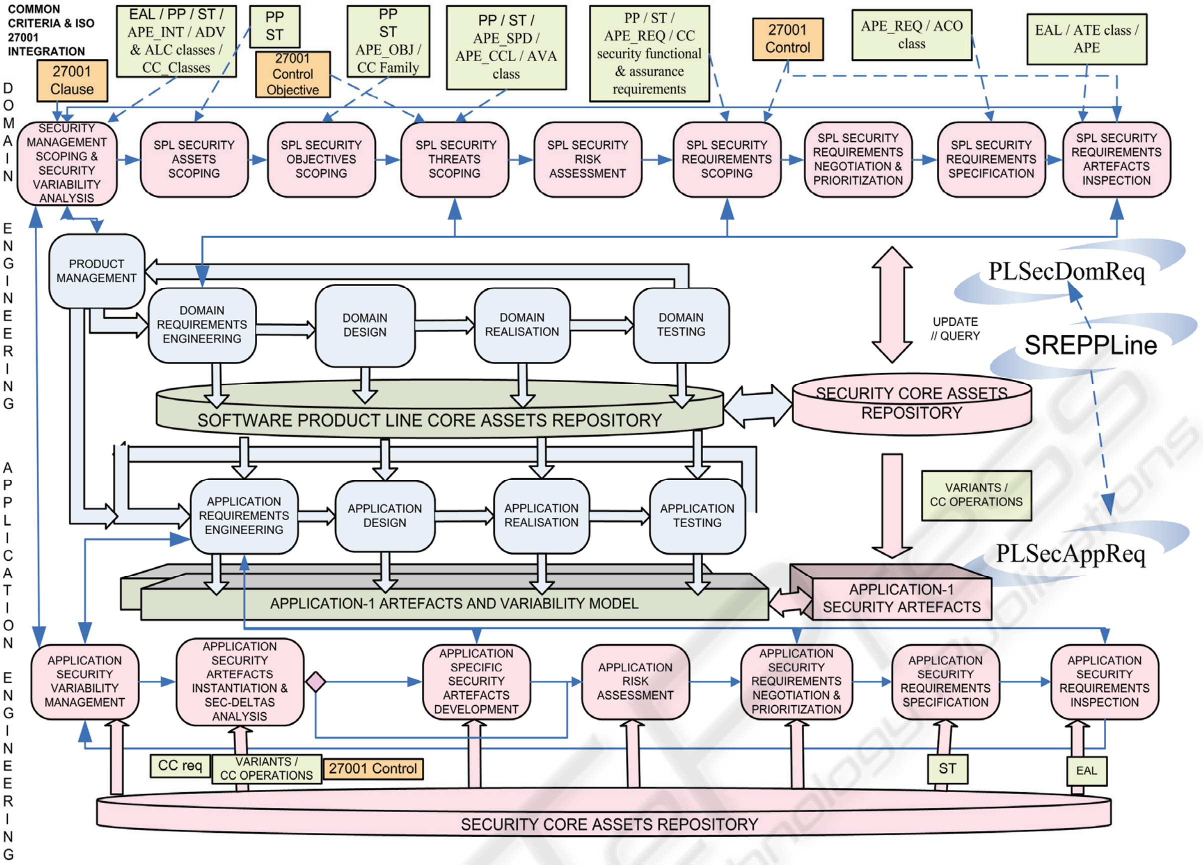
COMMON CRITERIA & ISO 27001 INTEGRATION
27001 Clause
EAL / PP / ST / APE_INT / ADV & ALC classes / CC_Classes
PP ST
27001 Control Objective
PP ST APE_OBJ / CC Family
PP / ST / APE_SPD / APE_CCL / AVA class
PP / ST / APE_REQ / CC security functional & assurance requirements
27001 Control
APE_REQ / ACO class
EAL / ATE class / APE
DOMAIN ENGINEERING
SECURITY MANAGEMENT SCOPING & SECURITY VARIABILITY ANALYSIS
SPL SECURITY ASSETS SCOPING
SPL SECURITY OBJECTIVES SCOPING
SPL SECURITY THREATS SCOPING
SPL SECURITY RISK ASSESSMENT
SPL SECURITY REQUIREMENTS SCOPING
SPL SECURITY REQUIREMENTS NEGOTIATION & PRIORITIZATION
SPL SECURITY REQUIREMENTS SPECIFICATION
SPL SECURITY REQUIREMENTS ARTEFACTS INSPECTION
PRODUCT MANAGEMENT
DOMAIN REQUIREMENTS ENGINEERING
DOMAIN DESIGN
DOMAIN REALISATION
DOMAIN TESTING
PLSecDomReq
UPDATE // QUERY
SREPPLine
SOFTWARE PRODUCT LINE CORE ASSETS REPOSITORY
SECURITY CORE ASSETS REPOSITORY
APPLICATION ENGINEERING
APPLICATION REQUIREMENTS ENGINEERING
APPLICATION DESIGN
APPLICATION REALISATION
APPLICATION TESTING
VARIANTS / CC OPERATIONS
PLSecAppReq
APPLICATION-1 ARTEFACTS AND VARIABILITY MODEL
APPLICATION-1 SECURITY ARTEFACTS
APPLICATION SECURITY VARIABILITY MANAGEMENT
APPLICATION SECURITY ARTEFACTS INSTANTIATION & SEC-DELTAS ANALYSIS
APPLICATION SPECIFIC SECURITY ARTEFACTS DEVELOPMENT
APPLICATION RISK ASSESSMENT
APPLICATION SECURITY REQUIREMENTS NEGOTIATION & PRIORITIZATION
APPLICATION SECURITY REQUIREMENTS SPECIFICATION
APPLICATION SECURITY REQUIREMENTS INSPECTION
CC req
VARIANTS / CC OPERATIONS
27001 Control
ST
EAL
SECURITY CORE ASSETS REPOSITORY

Figure 1: Software product line security requirements engineering framework.

and the Security Requirements Decision Model, which assist in the management of the variability of the SPL. Because in (Mellado et al., 2008) we already described the most important characteristics of the activities of the other subprocess of which the Security quality Requirements Engineering Process for Software Product Lines (SREPPLine) is composed, the Product Line Security Domain Requirements Engineering (PLSecDomReq) subprocess. Hence the aim of this approach is to deal with the security requirements artefacts and their variability from the early stages of the products of a SPL development in a systematic way, in order to facilitate the conformance of SPL products to the most relevant security standards with regard to the management of security requirements, such as ISO/IEC 27001 (ISO/IEC 2006) and ISO/IEC 15408 (Common Criteria) (ISO/IEC 2005). To this end, we will propose a systematic and iterative process based on a security requirements decision model driven by security standards in order to assist in SPL products security certification along with a security variability model to manage the variability and traceability of the security requirements artefacts of the SPL products.

The remainder of this paper is structured as follows. In Section 2, we will briefly describe our Security quality Requirements Engineering Process for software Product Lines (SREPPLine). Then, in Section 3, we will explain the security requirements variability management in SREPPLine. Next, in Section 4 we will present the main characteristics of the activities of the Product Line Security Application Requirements Engineering subprocess. Finally, in Section 5, we will discuss our contributions and future work.

2 SREPPLINE

SREPPLine (Security Requirements Engineering Process for software Product Lines) is an add-in of activities, which can be incorporated into an organization's SPL development process model providing it with a security requirements engineering approach.

It is a security features or security goals based process which is driven by risk and security standards (concretely ISO/IEC 27001 and Common Criteria) and deals with security requirements and their related artefacts from the early stages of SPL development in a systematic and intuitive way especially tailored for SPL based development.

It is based on the use of the latest and widely validated security requirements techniques, such as security use cases (Firesmith 2003) or misuse cases (Sindre et al., 2005), along with the integration of the Common Criteria (CC) components and ISO/IEC 27001 controls into the SPL lifecycle in order to facilitate SPL products security certification. Moreover, our proposed process suggests using a method to carry out the risk assessment which conforms to ISO/IEC 13335 (ISO/IEC 2004), concretely it uses Magerit (López et al., 2005) (the spanish public risk management methodology and which is recognised by the NATO) for both SPL risk assessment and SPL products risk assessment. Furthermore, SREPPLine has the aim of minimizing the necessary security standards knowledge as well as security expert participation during SPL products development.

To this end, it provides a Security Core Assets Repository to facilitate security artefacts reuse and to implement the security variability model and the security requirement decision model, which assist in the management of the variability and traceability of the security requirements related artefacts of the SPL and its products. These models are the basis through which the activities of SREPPLine capture, represent and share knowledge about security requirements for SPL and help to certificate them against security standards. In essence, it is a knowledge repository with a structure to support security requirements reasoning in SPL.

As it is described in Figure 1 our process is composed of two subprocesses (shown in pink): Product Line Security Domain Requirements Engineering (PLSecDomReq) subprocess and Product Line Security Application Requirements Engineering (PLSecAppReq) subprocess. These subprocesses cover the four basic phases of requirements engineering according to (Kotonya et al., 2000): requirements elicitation; requirements analysis and negotiation; requirements documentation; and requirements validation and verification. However, due to space restrictions, in this paper we shall only outline the security requirements variability management and the key tasks that are part of the activities of PLSecAppReq subprocess.

3 SECURITY REQUIREMENTS VARIABILITY MANAGEMENT

The security requirements artefacts variability management is supported by two models. The Security Variability Model is used to assist in the management of the variability and traceability of the security requirements related artefacts of the SPL and its products along with the SPL and its products security standards certification. The Security Requirement Decision Model supports the capturing, specifying and reasoning about security requirements and their artefacts for the SPL members. It furthermore supports the development of a security requirement protection profile for the security goals of the system and it is also helpful in the process of determining the most appropriate security requirements artefacts and security standards.

3.1 Security Variability Model

Our proposed Security Variability Model, which will be shown in Figure 2 is based on the Reusable Assets Specification (RAS), adopted as an OMG standard (OMG_(Object_Management_Group) 2004) and moreover extends the orthogonal variability model of Pohl et al.(Pohl et al., 2005). It is also part of the Security Requirement Decision Model. This variability model relates the defined variability to other software development models such as feature models, use case models, design models and test models. Thus, it provides a cross-cutting view of the security requirements variability across all security development artefacts and assists in keeping the different views of variable security requirements artefacts consistent.

In order to relate the variability defined in the variability model to the software artefacts specified in other models, the meta-model depicted in Figure 2 contains the class ‘artefact’ which represents any kind of development artefact. Particular development artefacts are sub-classes of the ‘artefact’ class, such as ‘security artefact’ which is a specialization of an artefact.

In addition, as is depicted in Figure 2, a security artefact can but does not have to be categorized. The ‘category’ class helps us avoid semantic problems and assists in reusing security artefacts, and even in applying security patterns. It is a key class for the security requirement decision model, because it guides us through the categories thus allowing us to identify the security requirements artefacts systematically. Moreover, the ‘security artefact’ class has 'version' as a mandatory attribute in order to facilitate the security artefacts versions traceability and variability, as products with different versions of the same security artefacts might exist (due to the variability in time and in space). Finally, in Figure 2 we have represented the security standards variability, by integrating the Common Criteria (CC) elements, and the ISO/IEC 27001 controls into the security variability model. These security standards elements are related to the categories of certain particular security artefacts (security features, threats and security requirements) with the aim of assisting in the SPL or SPL products certification against these standards and making their reasoning easier.

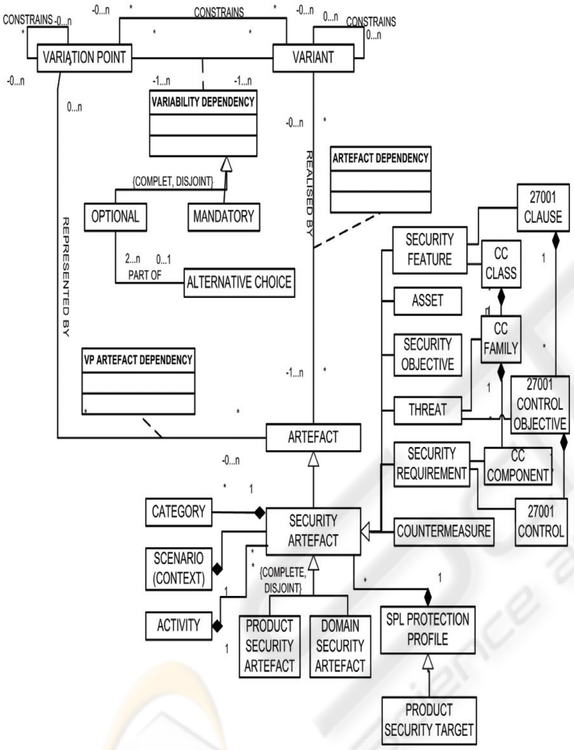

Figure 2: Security variability meta-model.

3.2 Security Requirement Decision Model

We treat security requirements artefacts as a natural source of variability among the products or SPL artefacts. In order to capture and manage knowledge related to security requirements in SPL we propose a security requirement decision model for SPL engineering, which will be shown in a different figure (Figure 3) to make its understanding easier. This model facilitates the security requirements related artefacts reasoning and the security standards conformance. It supports the capturing, specifying and reasoning of security requirements for both SPL and SPL members.

As a starting point we used the goals/soft-goals (Chung et al., 2000) and feature models and their correlations in order to take into consideration functional and non-functional requirements, concretely security requirements. To express the intentions of a system, goal models as well as feature models can be used, and this will, in most cases, define similar information (Pohl et al., 2005). Therefore, the interest in using goal/softgoal model as a starting point comes from the fact that it allows us to decide (if the traceability links are carefully established) what security features are needed to achieve the selected security goals and which is the optimal set of security features/goals of a determined priority in the context of the different scenarios of the SPL that provides the rationale of the selection. This supposes a rise in the abstraction level of the variants selection process, making the selection in the requirements level instead of in the design level.

In addition, within this model we characterize a SPL as a set of 'variation points' which are represented by 'features' or 'goals', and each goal can be achieved in many concrete ways, which are represented as 'scenarios'.

Security features are those features that describe the security characteristics of the system which correspond with the security goals that the system under consideration should achieve. Thereby, a group of assets will be involved in the achievement of each security feature.

These assets are the resources in the information systems of the SPL, or these which are related to them which are necessary for the organization to operate correctly and achieve its goals. There will be also different categories or types of assets (such as the environment, information systems, services, components and information or data).Dependencies between assets could also exist. Furthermore, an asset, as is shown in Figure 3, is a class which inherits from the 'security artefact' class, so it can be a 'variation point'. Each asset will have different related security objectives (or security dimensions) with the corresponding assigned value (following a standardized scale from 0 to 10 according to the risk methodology Magerit (López et al., 2005)) which is agreed by the stakeholders, who have also to reach an agreement about the common and optional assets. The valuation of each asset is given in each security objective and it is propagated through the dependency tree assets and therefore only the higher assets in the dependency tree have to be explicitly valued.

The security objectives or security dimensions are the objectives which must be achieved in order to protect the organization's business goals. According to Magerit (López et al., 2005), the security objectives/dimensions managed by the model can only be the following ones: integrity, confidentiality, availability, authenticity of service users, authenticity of data origin, accountability of service use and accountability of data access. Throughout the selected category/ies of the asset, this model could propose security objectives related to these categories to assist in the security objectives identification and valuation for each asset in a systematic way.

Furthermore, the assets are exposed to threats which may prevent the security objective from being achieved. Not all threats affect all assets nor all their security objectives, so those which are common and optional ones have to be identified. In addition, there is a certain relationship between the category of the asset and what might happen to it. Thus, throughout the selected category/ies of the asset this model could propose threats or categories of threats related to these categories of assets to assist in the common and optional threats identification and valuation. To calculate the impact of each threat, the value of the assets of each security objective along with the degradation caused by the threat are taken into account. The impact and the likelihood of occurrence of the threat are taken into account in order to estimate the risk. The risk is then classified in a range from 0 to 5 (according to the Magerit (López et al., 2005) scale).

Each type (category) of asset and depending upon its associated categories of threats will have a category or categories of security requirements related to it that could mitigate the impact or reduce the likelihood of these threats. This mechanism facilitates both the elicitation of the common and optional security requirements of the SPL and the security requirements instantiation in the products. Moreover, there could be dependencies between security requirements, so security requirements packages structured by the security dimension of the requirements could exist, which are a group of security requirements that work together in order to mitigate the same threats and satisfy similar security objectives of the assets. However, there could still be groups of requirements, which differ from one another in the level of detail they describe and in the testability they support. Therefore, a hierarchy of security requirements could be defined.

A security goal might be satisfied by multiple security requirements (different variants), and a mapping of the security requirements to countermeasures is carried out to give the best possible effect for the assets associated with the security feature. Thus, a variant is realised by one or more security requirements and is also supported by one or more countermeasures, which are procedures or technical mechanisms that reduce the risk and which are identified at the design stage. Countermeasures are architectural decisions that are used to achieve a security goal.

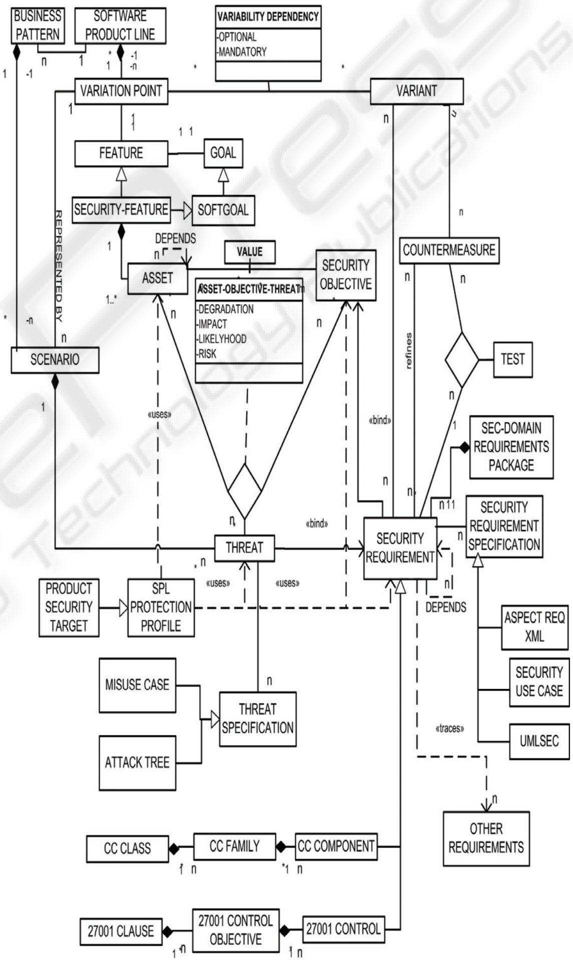

Figure 3: Security requirement decision model.

Furthermore, the SPL Protection Profile is an implementation independent statement of security requirements and their related security artefacts that has been shown to address threats that exist in an SPL environment; it has the aim of assisting in the SPL certification against the CC. There could be SPL Protection Profiles associated with Business

Patterns. Similarly, the Product Security Target is an implementation dependent statement of security requirements and their related security artefacts for a specific identified product of the SPL; it has the purpose of facilitating the SPL product certification against the CC. In addition, security standards elements have been integrated into our proposed Security Requirement Decision model with the objective of assisting in the SPL or SPL products certification against these standards and making their reasoning easier. These security standards elements (CC elements and ISO/IEC 27001 controls) are related to the categories of certain particular security artefacts (security features, threats and security requirements) to assist in this task.

We use scenarios to represent variants. All scenarios have an environment, a context that may include aspects such as assets, actors, misusers or misactors, use cases, misuse cases (Sindre et al., 2005) or threats and security use cases (Firesmith 2003). We model threats as misuse case templates or attack trees in order to document their variability. Security requirements can be documented as security use case templates, UMLsec (Jürjens 2002) with additional stereotypes, or as textual requirements by using aspect-XML specification (Kuloor et al., 2003).

The orthogonal variability model, upon which our approach is based, allows us to relate the different places at which the variability is defined to each other. In fact, starting from a changed artefact, other artefacts affected by the change can be found by following the relationship with the associated variant and from the variant with the other associated artefacts. The variability of the security artefacts of the security decision model is thereby clearly and unambiguously documented throughout the artefact dependencies of the security variability model.

4 PRODUCT LINE SECURITY APPLICATION REQUIREMENTS ENGINEERING SUBPROCESS

The main goals of this sub-process are: elicitation and documentation of the security requirements and their related security artefacts of the application of the SPL; ensuring that they conform to IEEE 830:1998, along with gathering them together in a Security Targets (ST) adapted document by following the ISO/IEC 15446 (ISO/IEC 2004) standard; reusing as much as possible the security domain artefacts and requirements.

PLSecAppReq activity 1 is the "**Application Security Variability Management**". In this activity stakeholders are informed of the commonalities and variabilities of the security features of the SPL, because the goal of this activity is to make the stakeholders aware of the security goals and features of the SPL as well as to elicit application security goals and features. The Security Requirements Decision Model and the Security Variability Model enable the security requirements engineer to communicate the relevant security related variation points, security related variants and their dependences to stakeholders. Additionally, the traceability links of the variability model to security domain artefacts enable the security requirements engineer to describe the particularities of a particular security related variant. Therefore, once the stakeholders have informed the security requirements engineer of their security goals and of the features necessary for the application (or product), the result of this activity is a set of domain security goals and features of the SPL, which may not completely fulfil the stakeholders security goals for the application.

In activity 2 ("**Application Security Artefacts Instantiation**") application security artefacts from the set of domain security features obtained in the previous activity are instantiated. Throughout the Security Requirements Decision Model and the Security Variability Model the appropriate security artefacts (that is, the security variants) for the specific application (product) which will as far as possible satisfy the application security goals, are selected. The result of this activity is a set of security requirements and their related artefacts, which may not completely fulfil the stakeholders' application requirements.

In the activity 3 "**Application Specific Security Artefacts Development and Sec-Deltas Analysis**" the sec-deltas analysis is performed. The sec-deltas occur when stakeholder security requirements cannot be completely satisfied by security domain requirements artefacts. During the sec-deltas analysis, sec-deltas to the security domain variability model resulting from stakeholders' security features/goals are analyzed. Next, the impact of the security variability model sec-deltas on the corresponding security domain artefacts is analyzed. The results of this analysis are the security application variability model along with the security requirements artefacts deltas. Finally, these sec-deltas are communicated to the security risk expert who estimates the risks of carrying our or not carrying out the security requirements deltas (activity 4 "**Application Risk Assessment**").

In the "**Application Security Requirements Negotiation and Prioritization**" activity (activity 5 of PLSecAppReq), after the application risk assessment of the sec-deltas has been performed, they are communicated to the security architect and to the security requirements engineer who estimates the realisation effort based on the sec-deltas and their associated risks. With this estimation the stakeholders decide whether or not the security requirements deltas should be carried out and which security standard the application should fulfil. As a result of this activity, the application security requirements and the corresponding security requirements artefacts and security application variability model are defined.

Finally, in the "**Application Security Requirements Specification**" activity (activity 6 of PLSecAppReq) the application security artefacts, the sec-deltas and the traces between application security artefacts and the corresponding domain security artefacts are specified and documented. Moreover, the security application variability model and the traceability links of the application security artefacts to the application-specific variability model are documented. The estimated risk and realisation costs are even related to the sec-deltas to ensure that decisions about sec-deltas are traceable.

Finally, in the activity 7 ("**Application Security Requirements Inspection**") the same points listed in the PLSecDomReq activity 9 (Security Requirements Artefacts Inspection) are verified along with the security requirements artefacts variability consistency between the application and domain artefacts.

5 CONCLUSIONS

Security requirements issues are extremely important in SPL because a weakness in security can cause problems throughout the lifecycle of a line. Although there have been several attempts to fill the gap between requirements engineering and SPL requirements engineering, no systematic approach with which to define security quality requirements and to manage their variability and their related security artefacts to the models of an SPL is available.

The contribution of this work is that of providing a systematic approach for the management of the security requirements and their variability from the early stages of the product line development, in order to facilitate the conformance of the SPL products to the most relevant security standards with regard to the management of security requirements, such as ISO/IEC 27001 and ISO/IEC 15408 (Common Criteria). Our proposal defines a systematic process based on a security requirements decision model driven by security standards in order to assist in SPL security requirements definition and to facilitate products security certification. Moreover, a security variability model with which to manage the variability and traceability of the security requirements related artefacts of the SPL and its products is proposed. Consequently, our proposal allows us to make security variants selection in the requirements level instead of in the design level as well as providing a cross-cutting view of the security variability across all security development artefacts and assisting in mantaining the different views of variable security requirements artefacts consistent. Hence, SREPPLine is a suitable approach especially for SPL where security is a key quality issue.

Finally, further work is also required to refine the prototype of our CARE (Computer Aided Requirements Engineering) tool which we are developing to support SREPPLine and the Security Resources Repository, which was one of the lessons learned in the case study performed at the Spanish Social Security IT Department described in (Mellado et al., 2008), in order to assist in the complex management and maintainability of the variability and traceability relations. Furthermore, we shall carry out a refinement of our approach by proving it with a complete and exhaustive real case study of SREPPLine and its CARE-tool in order to validate and illustrate SREPPLine in far greater depth, with the aim of providing an holistic framework for security requirements engineering in SPL.

ACKNOWLEDGEMENTS

This paper is part of the ESFINGE (TIN2006-15175-C05-05) and ELEPES (TIN2006-27690-E) projects of the Ministry of Education and Science (Spain), and of the QUASIMODO (PAC08-0157-0668), MISTICO (PBC-06-0082) and MELISA (PAC08-0142-335) projects of the JCCM and the FEDER.

REFERENCES

Birk, A. & G. Heller (2007). "Challenges for requirements engineering and management in software product line development." *International Conference on Requirements Engineering (REFSQ 2007)*: 300-305.

Bosh, J. (2000). *Design & Use of Software Architectures*, Pearson Education Limited,

Chung, L., B. Nixon, E. Yu & J. Mylopoulos (2000). *Non-Functional Requirements in Software Engineering*, Kluwer Academic Publishers,

Clements, P. & L. Northrop (2002). *Software Product Lines: Practices and Patterns*, Addison-Wesley,

Firesmith, D. G. (2003). "Engineering Security Requirements." *Journal of Object Technology* 2(1): 53-68.

ISO/IEC (2004). ISO/IEC 13335 Information technology - Security techniques - Management of information and communications technology security - Part 1: Concepts and models for information and communications technology security management.

ISO/IEC (2004). ISO/IEC 15446 Information technology - Security techniques - Guide for the production of Protection Profiles and Security Targets.

ISO/IEC (2005). ISO/IEC 15408:2005 Information technology - Security techniques - Evaluation criteria for IT security, (Common Criteria v3.0).

ISO/IEC (2006). ISO/IEC 27001 Information technology -- Security techniques -- Information security management systems -- Requirements.

Jürjens, J. (2002). "UMLsec: extending UML for secure systems development." *UML 2002 - The Unified Modeling Language. Model Engineering, Languages,Concepts, and Tools. 5th International Conference.* LNCS 2460: 412-425.

Kotonya, G. & I. Sommerville (2000). *Requirements Engineering Process and Techniques*, John Willey & Sons,

Kuloor, C. & A. Eberlein (2003). Aspect-Oriented Requirements Engineering for Software Product Lines. *Proceedings of the 10 th IEEE International Conference and Workshop on the Engineering of Computer-Based Systems (ECBS'03)*.

López, F., M. A. Amutio, J. Candau & J. A. Mañas (2005). *Methodology for Information Systems Risk Analysis and Management*, Ministry of Public Administration,

Mellado, D., E. Fernández-Medina & M. Piattini (2006). "Applying a Security Requirements Engineering Process." *11th European Symposium on Research in Computer Security (ESORICS 2006)* Springer LNCS 4189: 192-206.

Mellado, D., E. Fernández-Medina & M. Piattini (2008). Towards security requriements management for software product lines: a security domain requirements engineering process. Computer Standards & Interfaces. (accepted): http://dx.doi.org/10.1016/j.csi.2008.03.004.

Niemelä, E. & A. Immonen (2007). Capturing quality requirements of product family architecture. Information & Software Technology. 49: 1107-1120.

OMG_(Object_Management_Group) (2004). Reusable Assets Specification (RAS), ptc/04-06-06.

Pohl, K., G. Böckle & F. v. d. Linden (2005). *Software Product Line Engineering. Foundations, Principles and Techniques*. Berlin Heidelberg: Springer,

Sindre, G. & A. L. Opdahl (2005). "Eliciting security requirements with misuse cases." *Requirements Engineering 10* 1: 34-44.