SECURE INFORMATION SYSTEMS DEVELOPMENT
Based on a Security Requirements Engineering Process

Daniel Mellado
Ministry of Labour and Social Affairs, Information Technology Center of the National Social Security Institute, Madrid, Spain

Eduardo Fernández-Medina, Mario Piattini
Alarcos Research Group, Information Systems and Technologies Department, UCLM-Soluziona Research and Development Institute, University of Castilla-La Mancha
Paseo de la Universidad 4, 13071 Ciudad Real, Spain.

Keywords: Security Requirements, Security Requirements Engineering, Common Criteria.

Abstract: Integration of security into the early stages of the system development is necessary to build secure systems. However, in the majority of software projects security is dealt with when the system has already been designed and put into operation. This paper will propose an approach called SREP (Security Requirements Engineering Process) for the development of secure software. We will present an iterative and incremental micro-process for the security requirements analysis that is repeatedly performed at each phase. It integrates the Common Criteria into the software lifecycle model as well as it is based on the reuse of security requirements, by providing a security resources repository. In brief, we will present an approach which deals with the security requirements at the early stages of software development in a systematic and intuitive way, and which also conforms to ISO/IEC 17799:2005.

1 INTRODUCTION

In the last years we have observed more and more organizations becoming heavily dependent on Information Systems (IS). However, software applications are increasingly ubiquitous, heterogeneous, mission-critical and vulnerable to unintentional or intentional security incidents (CERT; Kemmerer 2003), so that it is absolutely vital that IS are properly ensured from the very beginning (Baskeville 1992; McDermott and Fox 1999), due to the potential losses faced by organizations that put their trust in all these IS.

A very important part in the software development process for the achievement of secure software systems is that known as Security Requirements Engineering. Which provides techniques, methods and norms for tackling this task in the IS development cycle. It should involve the use of repeatable and systematic procedures in an effort to ensure that the set of requirements obtained is complete, consistent and easy to understand and analyzable by the different actors involved in the development of the system (Kotonya and Sommerville 1998).

After having performed a comparative analysis of several relevant proposals of IS security requirements, as those of (Toval, Nicolás et al. 2001), (Popp, Jürjens et al. 2003), (Firesmith 2003), (Breu, Burger et al. 2004), etc., in (Mellado, Fernández-Medina et al. 2006), we concluded that those proposals did not reach the desired level of integration into the development of IS, nor are specific enough for a systematic and intuitive treatment of IS security requirements at the first stages of software development. Therefore, in this poster we will present the Security Requirements Engineering Process (SREP), which describes how to integrate security requirements into the software engineering process in a systematic and intuitive way. In order to achieve this goal, our approach is based on the integration of the Common Criteria (CC) into the software lifecycle model, because the CC helps us deal with the security requirements along all the IS development lifecycle, together with

Mellado D., Fernández-Medina E. and Piattini M. (2006).
SECURE INFORMATION SYSTEMS DEVELOPMENT - Based on a Security Requirements Engineering Process.
In *Proceedings of the International Conference on Security and Cryptography*, pages 467-470
DOI: 10.5220/0002098004670470

the reuse of security requirements which are compatible with the CC Framework subset. In addition, in order to support this method and make it easy the treatment and specification of the security requirements, assets, security objectives and threats, we will propose the use of several concepts and techniques: a security resources repository (with assets, threats, requirements, etc), the use of UMLSec (Popp, Jürjens et al. 2003), misuse cases (Sindre, Firesmith et al. 2003), threat/attack trees, and security uses cases (Firesmith 2003). These latter techniques will be used following the criteria of effectiveness, and they allow us to integrate security aspects from the beginning into an IS development process, for example by expressing security-related information within the diagrams in a UML system specification, thanks to UMLSec.

The remainder of the paper is set out as follows: in section 2, we will outline an overview of our Security Requirements Engineering Process. Lastly, our conclusions and further research are set out in section 3.

2 A GENERAL OVERVIEW OF SREP

The Security Requirements Engineering Process (SREP) is an asset-based and risk-driven method for the establishment of security requirements in the development of secure Information Systems. Basically, this process describes how to integrate the CC into the software lifecycle model together with the use of a security resources repository to support reuse of security requirements (modelled with UMLSec, or expressed as security use cases or as plain text with formal specification), assets, threats (which can be expressed as misuse cases, threat/attack trees, UMLSec diagrams) and countermeasures. The focus of this methodology seeks to build security concepts at the early phases of the development lifecycle.

As it is described in Figure 1, the Unified Process (UP) (Booch, Rumbaugh et al. 1999) lifecycle is divided into a sequence of phases, and each phase may include many iterations. Each iteration is like a mini-project and it may contain all the core workflows (requirements, analysis, design, implementation, and test), but with different emphasis depending on where the iteration is in the lifecycle. Moreover, the core of SREP is a micro-process, made up of nine activities which are repeatedly performed at each iteration throughout the iterative and incremental development, but also with different emphasis depending on what phase of the lifecycle the iteration is in. Thus, the model chosen for SREP is iterative and incremental, and the security requirements evolve along the lifecycle. At the same time, the CC Components are introduced into the software lifecycle, so that SREP uses different CC Components according to the phase, although the Software Quality Assurance (SQA) activities are performed along all the phases of the software development lifecycle. And it is in these SQA activities where the CC Assurance Requirements might be incorporated into, according to (Kam 2005).

In addition, it facilitates the requirements reusability. The purpose of development with requirements reuse is to identify descriptions of systems that could be used (either totally or partially) with a minimal number of modifications, thus reducing the total effort of development (Cybulsky and Reed 2000). Moreover, reusing security requirements helps us increase their quality: inconsistency, errors, ambiguity and other problems can be detected and corrected for an improved use in subsequent projects (Toval, Nicolás et al. 2001). Thereby, it will guarantee us the fastest possible development cycles based on proven solutions.

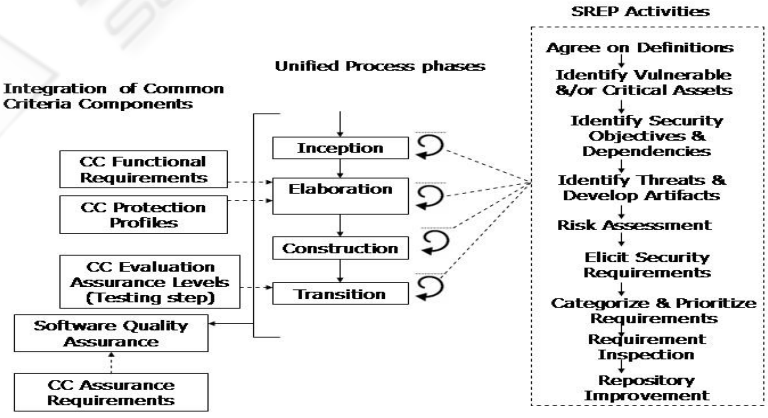

Figure 1: SREP overview.

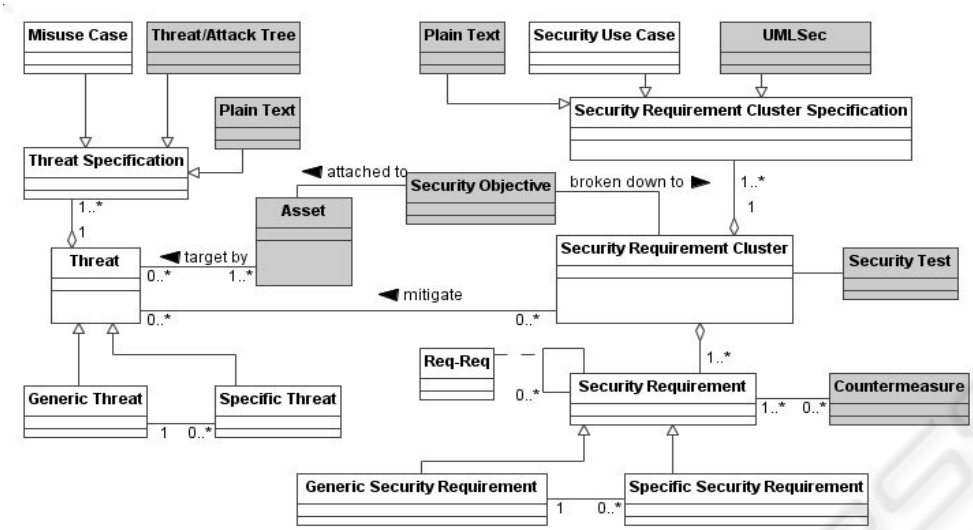

Figure 2: Meta-model for security resources repository.

2.1 The Security Resources Repository

We propose a *Security Resources Repository* (SRR), which stores all the reusable elements. The repository, as SIREN (Toval, Nicolás et al. 2001) approach, supports the concepts of domains and profiles. We propose to implement the domains and profiles by taking advantage of the CC concepts of packages and Protection Profiles (PP). Thus, the requirements are stored as standardized subsets of specific security requirements together with its related elements of the SRR (threats, etc.). In brief, each domain or profile is a view of the global SRR.

A meta-model, which is an extension of the meta-model for repository proposed by (Sindre, Firesmith et al. 2003), showing the organization of the SRR is exposed below in Fig. 2. The dark background in the objects represents our contribution to the meta-model.

Finally, according to ISO/IEC 17799:2005, we propose to include legal, statutory, regulatory, and contractual requirements that the organization, its trading partners, contractors, and service providers have to satisfy, and their socio-cultural environment. After converting these requirements into software and system requirements format, these requirements along with the CC security functional requirements would be the initial subset of security requirements of the SRR.

3 CONCLUSIONS

In our present so-called Information Society the Information Security is usually only tackled from a technical viewpoint at the implementation stage, even though it is an important aspect. We believe it is fundamental to deal with security at all stages of IS development, especially in the establishment of security requirements, since these form the basis for the achievement of a robust IS.

Consequently, we present an approach that deals with the security requirements at the first stages of software development in a systematic and intuitive way, which is based on the reuse of security requirements, by providing a Security Resources Repository (SRR), together with the integration of the Common Criteria into software lifecycle model. Moreover, it conforms to ISO/IEC 15408 and ISO/IEC 17799:2005. Starting from the concept of iterative software construction, we propose a micro-process for the security requirements analysis, 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. Finally, one of the most relevant aspects is the fact that this proposal integrates other approaches, such as UMLSec (Popp, Jürjens et al. 2003), security use cases (Firesmith 2003) or misuse cases (Sindre, Firesmith et al. 2003).

Further work is also needed to provide a CARE (Computer-Aided Requirements Engineering) tool which supports the process, as well as a refinement of the theoretical approach by proving it with a real

case study in order to complete and detail more SREP.

ACKNOWLEDGEMENTS

This paper has been produced in the context of the DIMENSIONS (PBC-05-012-2) Project of the Consejería de Ciencia y Tecnología de la Junta de Comunidades de Castilla- La Mancha along with FEDER and the CALIPO (TIC2003-07804-CO5-03) and RETISTIC (TIC2002-12487-E) projects of the Dirección General de Investigación del Ministerio de Ciencia y Tecnología.

REFERENCES

Baskeville, R. (1992). "The development duality of information systems security." *Journal of Management Systems* **4**(1): 1-12.

Booch, G., J. Rumbaugh and I. Jacobson (1999). *The Unified Software Development Process*, Addison-Wesley.

Breu, R., K. Burger, M. Hafner and G. Popp (2004). "Towards a Systematic Development of Secure Systems." *Proceedings WOSIS 2004*: 1-12.

CERT http://www.cert.org.

Cybulsky, J. and K. Reed (2000). "Requirements Classification and Reuse: Crossing Domains Boundaries." *ICSR'2000*: 190-210.

Firesmith, D. G. (2003). "Security Use Cases." *Journal of Object Technology*: 53-64.

Kam, S. H. (2005). "Integrating the Common Criteria Into the Software Engineering Lifecycle." *IDEAS'05*: 267-273.

Kemmerer, R. (2003). "Cybersecurity." *Proc. ICSE'03-25th Intl. Conf. on Software engineering*: 705-715.

Kotonya, G. and I. Sommerville (1998). *Requirements Engineering Process and Techniques*,

McDermott, J. and C. Fox (1999). Using Abuse Case Models for Security Requirements Analysis. *Annual Computer Security Applications Conference*, Phoenix, Arizona.

Mellado, D., E. Fernández-Medina and M. Piattini (2006). "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**: 1044-1053.

Popp, G., J. Jürjens, G. Wimmel and R. Breu (2003). Security-Critical System Development with Extended Use Cases. *10th Asia-Pacific Software Engineering Conference***:** 478-487.

Sindre, G., D. G. Firesmith and A. L. Opdahl (2003). A Reuse-Based Approach to Determining Security Requirements. *Proc. 9th International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ'03)*, Austria.

Toval, A., J. Nicolás, B. Moros and F. García (2001). Requirements Reuse for Improving Information Systems Security: A Practitioner's Approach. *Requirements Engineering Journal*. **6:** 205-219.