A Model for Structuring and Reusing Security Requirements Sources and Security Requirements

Christian Schmitt[1] and Peter Liggesmeyer[2, 3]

[1]Siemens AG, Siemens Corporate Technology,
Otto-Hahn-Ring 6, 81739 Munich, Germany
[2]Research Group Software Engineering, University of Kaiserslautern,
67663 Kaiserslautern, Germany
[3]Fraunhofer Institute for Experimental Software Engineering,
67663 Kaiserslautern, Germany
ch.schmitt@siemens.com, liggesmeyer@cs.uni-kl.de

Abstract. Various security requirements sources need to be incorporated when developing security requirements. A challenge for teams developing security requirements is to identify and structure relevant sources, to satisfy compliance-related obligations, and to identify and properly address relevant threats, weaknesses and vulnerabilities. In this paper, we present a generic model which can be used for structuring and reusing security requirements sources and security requirements, to improve the efficiency of security requirements engineering and to achieve a desired 'baseline' security level and completeness of security requirements. The model supports security requirements engineering in general but can also be applied for continuous security requirements engineering in order to analyze and evaluate the influence of changes in software or the environment on security requirements and the overall software and system security. Elements of the model and their interdependencies are described, and observations on important aspects when applying this model in an organization are provided.

Keywords: Security Engineering, Security Requirements, Security Requirements Engineering, Security Requirements Sources, Requirements Reuse, Continuous Requirements Engineering.

1 Introduction

1.1 Background

Teams developing security requirements need to identify, structure and incorporate applicable security requirements sources (SRS) in order to satisfy applicable compliance obligations, as well as counter relevant threats, weaknesses and vulnerabilities. Fig. 1 shows the relationships which should be reflected in the activities when developing security requirements. Moreover, it illustrates that for the design of security

measures (as part of the security architecture) it is important not only to obtain and interpret the security requirements specification, but also to understand the requirement sources and the problem space which lead to the specification of the various security requirements. Traceability from security requirements to raw requirements, as well as to relevant threats, weaknesses and vulnerabilities should be possible when designing security measures. Additionally, the traceability from each raw requirement, as well as each threat, weakness and vulnerability to the respective source (i.e. compliance obligation, diagnostic information and knowledge and results from methods) should be possible. Reversely, for SRS it should be possible to check if, and by which security requirement they are addressed. This mutual referencing and traceability enhances quality and completeness of security requirements. In the end it must be ensured that the designed security measures in the security architecture satisfy the raw requirements from compliance obligations and furthermore counter all relevant threats, weaknesses and vulnerabilities, in order to protect the valuable assets and services.

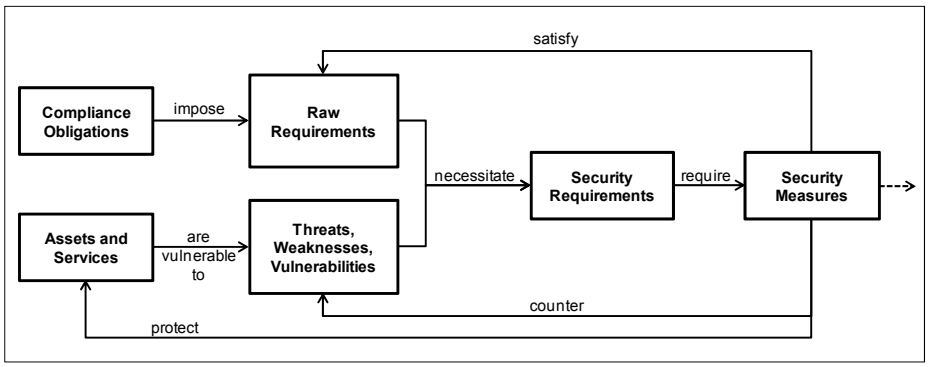

Fig. 1. Relationships in Security Requirements Engineering[1]

In settings where changes either to software, the system or the operational environment occur, it is important to properly evaluate which influence on or additional threats, weaknesses or vulnerabilities changes arise due to a change. Moreover, it needs to be checked if conformity to compliance obligation still can be achieved before existing security requirements are revised or new requirements are specified.

1.2 Problem and Research Gap

Compliance obligations are underrepresented in most of the frequently mentioned Security Requirements Engineering (SRE) processes and frameworks (e.g. [2–5]). Although most of them propose the use of certain SRE methods for requirements elicitation, they do not explicitly foresee the incorporation of other requirement sources such as raw security requirements from compliance obligations. Only

[1] This figure is inspired by the illustration 'Security Threats, Requirements, and Mechanisms' in [1]

Mellado et al. [4] recommend to include legal, statutory, regulatory, and contractual requirements, however leave open how it should be done in practice. Another problem concerning compliance obligations is that the reason or root cause (e.g. the threats source or weaknesses) for a security raw requirement is not provided in most cases. This leaves room for (mis-)interpretation and thus may lead to the specification of wrong or incomplete security requirements, particularly if the required security knowledge and skills are not available.

Required knowledge, skills and mindset for SRE is different from 'traditional' requirements engineering. It is more difficult to define what a system should not do or to identify the threats to it, than defining what it should do [6]. 'Traditional' requirements engineering techniques are usually focused on functional requirements than on security requirements. Moreover, "most requirements engineers are poorly trained to elicit, analyze, and specify security requirements" [7]. Thus, for requirements engineering teams without security expertise, it is important that security knowledge and information is provided in a structured, understandable and reusable way, so that it can be incorporated when interpreting compliance obligations, applying SRE methods, and specifying security requirements. First approaches for the provisioning of reusable security information and knowledge propose the development, use and improvement of a requirements repository or a knowledge base. In SIREN [8], the requirements repository is filled with countermeasures taken from MAGERIT [9] which were translated into security requirements. Mellado et al. [4] propose to store and reuse elements from Common Criteria. Dikanski and Abeck [10] propose to create reusable security requirements analysis templates (SecRAT) in order to develop and use a knowledge base, offering various relevant information such as security standards, technologies, security models, principles and policies, which can be reused for security requirements engineering. However, to the best of our knowledge, no model or framework exists, which combines compliance obligations, security information and knowledge resources, as well as results and artifacts from SRE methods in order to support the SRE activities and overcome the challenges as mentioned before.

1.3 Research Contribution

We present a generic model which can be used to support the analysis and specification of security requirements by structuring relevant Security Requirements Sources. This particularly incorporates:

- **Different Scope Areas of SRS and Security Requirements:** The model shall be useable for software security requirements and also incorporate the system level, as well as physical, technical and organizational aspects in the environment.
- **Flexibility:** The model shall be flexible enough to structure (most) of the relevant SRS.
- **Reuse of Security Information and Knowledge:** Reuse of information and knowledge shall be incorporated to increase the efficiency of SRE and quality of security requirements.

- **Relation between SRS:** The relation between different kinds of security information and knowledge (e.g. diagnostic vs. prescriptive) shall be obvious.
- **Quality / Baseline Security:** It shall be possible to verify the quality and completeness of security by means of 'baseline security'[2], covering the most prevalent aspects of the problem space.
- **Traceability:** Traceability between specified security requirements and compliance obligations, as well as security information and knowledge shall be possible.

2 Model for Structuring Security Requirements Sources and Security Requirements

2.1 Model Overview

Fig. 2 shows our proposed model for structuring and reusing security requirements sources and security requirements. It was our intention to develop a preferably stable structure, in which the relevant SRS can be structured and provided to the requirements engineering team for the specification of security requirements. The two main structure elements of the model are *security requirements scope areas* and *security topics*, which are described below.

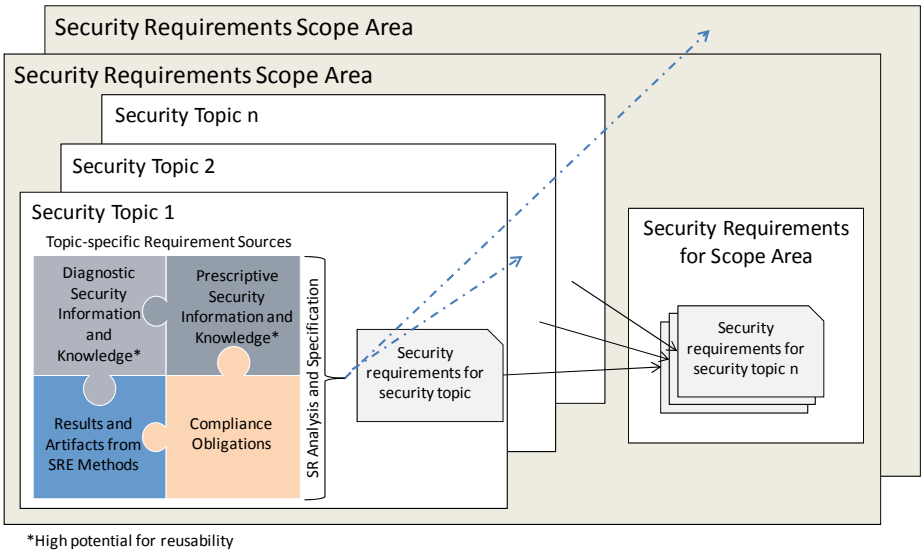

Fig. 2. Generic SRE Model

[2] With 'baseline security' we mean to cover at least a predetermined set of typical threats, weaknesses and / or vulnerabilities that should be addressed through security requirements.

2.2 Security Requirements Scope Areas

A scope area determines the work to be accomplished, the problem space to be analyzed, and the topics to be covered when developing security requirements. A scope area is subdivided into several security topics, for which topic-specific requirement sources are provided and analyzed (for details see section 2.3). The desired result is a set of security requirements for a particular scope area, structured according to the respective security topics within the scope area. Velasco et al distinguish between software security and system security requirements [11]. In SIREN [8] Toval et al. differentiate between three different requirements specifications, namely system requirements (SyRS), software requirements (SRS) and interface requirements (IRS).

For our model, we distinguish between the three scope areas: Software / Component, Physical and Technical Environment, and Organizational Environment[3] (see also [12]) which can be characterized as follows:

- **Software / Components** deals with aspects required to develop secure software or products. A software or component can be deployed in different environments. Some examples of software security topics are authentication, authorization / access control, session management, data at rest security or data in transit security. The outcome of the requirements engineering phase is a set of security requirements and trust assumptions, which is typically part of a software requirements specification document. In companies, product business typically falls under this scope area.
- A system in its **Physical and Technical Environment** is constrained and influenced by the software / components it inherits as well as its organizational environment. The physical and technical environment consists of relevant physical and technical conditions and objects which constrain or influence software. It addresses the typical security aspects concerning the secure deployment and configuration of software or systems in their physical and technical environment. Examples for objects within the physical and technical environment are buildings and rooms, server components, client devices, network(s), and other neighboring systems or services relevant for the system. Examples of security topics related to the system in its physical and technical environment are physical security, network security, secure software configuration, secure system configuration and hardening, and malware protection. The outcome of the requirements engineering phase is a set of security requirements and trust assumptions for a system in its physical and technical environment, which is typically part of the system requirements specification. In companies, the solution business usually falls under this scope area.
- **Organizational Environment** as third scope area deals with all organizational and process-related aspects which are relevant to securely set up, operate and maintain a software or system in its physical and technical environment. Examples for organizational aspects are the issuance of mandatory security policies and guidelines,

[3] A fourth potential scope area is *Development Environment*, which addresses all aspects required for developing a product or solution securely. However, in this paper we focus on the development of secure products and solutions and therefore omit aspects related to this scope area on purpose.

the set-up of a security organization with clearly defined roles and responsibilities and the organization of security trainings and awareness activities for employees and third party personnel. Examples for operational security topics are security related processes and procedures such as user management, privilege management, key management, vulnerability and patch management, incident management and many more. The outcome of the security requirements engineering phase is a set of security requirements and trust assumptions for the system in its organizational environment, which can be part of various documents such as operation concepts, maintenance concepts, contractual documents or service level agreements with third parties, and many more.

Scope areas mutually constraint and influence each other. Influences and constraints need to be identified and covered by requirements sources (i.e. security knowledge and information, as well as results from SRE methods) and incorporated into the security requirements engineering process. Security requirements assigned to a security topic in a scope area originate not only from the security topic itself, but may also originate from security topics in other scope areas. A more detailed overview on mutual implications between the scope areas is provided in [12]. The three scope areas ensure that the model is useable for both the software and system levels, as well as physical, technical and organizational aspects in the environment. The use of the scope areas for scoping and analyzing the problem space has already been successfully piloted within Siemens as part of the design and rollout of a method for conducting threat and risk analyses for products and solutions.

2.3 Security Topics

A security topic consolidates relevant SRS that are required for the analysis and specification of security requirements for this particular security topic. It therefore inherits topic-specific diagnostic and prescriptive security information and knowledge[4], results and artifacts from SRE methods, as well as raw requirements from compliance obligations. When subdividing scope areas into security topics, the following aspects should be considered:

- **Diagnostic information and knowledge:** Diagnostic security information and knowledge addresses the problem space by means of the bad things that might happen such as threats, weaknesses and vulnerabilities. In other words, diagnostic security information and knowledge describes what needs to be avoided and should be addressed by security requirements, as basis for the design of security measures. Examples of information and knowledge sources addressing the problem space are:
  - Security threats: for example, provided as lists of (mostly generic) threats in risk assessment guides [14], risk analysis methods [15] and risk management standards [16]

[4] Security information and knowledge is very multifaceted and made available in various ways for different purposes. We follow the proposed knowledge base structure by Barnum and McGraw [13] using the categories diagnostic and prescriptive knowledge.

- Security weaknesses and vulnerabilities: for example, provided in online catalogues or community developed dictionaries such as the Common Weakness Enumeration [17] and the Common Vulnerabilities and Exposures [18]
- Attack pattern: for example [19, 20]
- Knowledge about exploits and hacker tools, meaning the knowledge about exploitable vulnerabilities, and the tools that exist for these vulnerabilities in order to automate an attack.

Diagnostic security information and knowledge should preferably fit directly into the structure of scope areas and security topics without much additional effort for adapting the knowledge sources or re-designing the structure. Furthermore, the relation between diagnostic and prescriptive security information and knowledge should be made transparent, since diagnostic resources provide the basis to understand and motivate the prescriptive information. Moreover, it supports the conduction of SRE methods, since it provides typical threats, weaknesses and vulnerabilities in a reusable fashion which can be incorporated during an analysis.

- **Prescriptive information and knowledge:** Prescriptive security information and knowledge sources offer statements of practice on different kinds of abstraction levels. They provide information and knowledge about what to do, to build secure products and solutions. Prescriptive security information and knowledge ranges from high-level security principles (e.g. least privilege principle), over to guidelines for various security topics, up to rather concrete security controls (e.g. strong user identification) and specific security design patterns. Examples of prescriptive information and knowledge are security principles e.g. [21], security guidelines [22], security (design) patterns, and security control lists [23, 24]. Like diagnostic security information and knowledge, prescriptive aspects should be assignable to each security topic as easily and intuitively as possible, and preferably directly fit into the structure of scope areas and security topics without much additional effort for adapting the knowledge sources or re-designing the structure.
- **Compliance Obligations:** In practice, compliance obligations are very important for organizations, since non-conformities and the resulting negative consequences can have a high negative business impact, e.g. due to delays or even the refusal of the admission of a product or solution. Practical experience shows that if there are any mandatory security compliance obligations which are provided as list of relevant requirements or controls (as it is typically the case with security standards), often the efforts for SRE are primarily spent on the fulfillment of the compliance obligation. In such cases the application of SRE processes and methods in a 'greenfield approach', in which security requirements are additionally developed 'from scratch', are rather the exception than the rule. If the primary driver for security is compliance, the structure of security topics for an organization can be oriented according to the most relevant compliance obligation(s). Moreover, the terminology, extent and how raw requirements are provided should be reflected adequately in the structure. It is the objective that the number of ambiguities and multiple assignments of raw requirements from relevant compliance obligations to security topics is minimized as far possible.

- **Results and Artifacts from SRE Methods:** Various methods and approaches exist which can be used for security requirements engineering. Examples for methods which are primarily designed to reveal threats, weaknesses, vulnerabilities and attacks are Abuse Cases [25], Misuse Cases [26], Attack Trees [27], and Threat and Risk Analysis methods e.g. STRIDE [28]. Due to the different approaches and terminologies, the results and artifacts from methods are not necessarily in line with the structure as required to easily assign compliance obligations, as well as diagnostic and prescriptive information and knowledge. Furthermore, the terminologies as well as the approaches of different SRE methods differ, which often makes an integration of results from different methods challenging. A possible way to deal with this issue is to extract and assign identified threats, weaknesses, vulnerabilities and attacks from the results of the SRE method and assign it to the respective security topic.

The presented aspects concerning the security requirements sources show that there is no one-fits-all structure of scope areas and security topics. A structure should primarily be designed to fit to the needs of an organization and the relevant SRS. It should reflect the most important compliance obligations that need to be fulfilled and incorporate the diagnostic and prescriptive information and knowledge resources which are intended to be used by requirements engineering teams. Diagnostic and prescriptive security information and knowledge can be provided as reusable input (e.g. in form of a security repository). In case of recurring compliance obligations, the raw requirements can be provided as reusable input.

3 Summary and Future Work

All in all, we are confident that the structure of scope areas and security topics as presented is capable to consolidate relevant SRS and to fulfill the relevant aspects as mentioned in section 1. The model is intended to be used for structuring different scope areas, SRS and Security Requirements**.** Scope areas and security topics serve as main structure elements to reach the necessary flexibility to structure relevant SRS and security requirements. Since compliance obligations, diagnostic and prescriptive security information and knowledge are assigned to the respective security topics, the relation between them becomes much more transparent and improves the understanding of requirements engineering teams. Security information and knowledge sources can be provided as reusable content to the intended user group to an appropriate extent and level of detail, which increases efficiency, decreases the effort for security requirements engineering, and addresses (to a certain extent) the knowledge and skill related issues in SRE. Moreover, it can be used to set a minimum level of security by means of a ‘baseline security’. Thereby the most relevant threats, weaknesses, vulnerabilities and attacks are provided and addressed during the SRE activities. The required aspect of traceability between specified security requirements and the SRS is supported by the model, however adequate tool support for realizing the various dependencies must be provided. The model can be used for supporting security requirements engineering in general and also in a continuous engineering environment to analyze and evaluate the influence of changes in software or the environment on security requirements and the overall software and system security.

The model is currently in a conceptual stage with first practical experiences and promising results; however, a detailed practical evaluation needs to be carried out. Our next step will be the elaboration of a suitable structure of security topics for the three mentioned scope areas, incorporating our generic model as input. The structure will be developed under incorporation of selected compliance obligations such as international security standards and organizational security policies and best practices for a real-world project. The resulting structure will be used as basis to structure most of the raw requirements from compliance obligations, and to provide security information and knowledge resources as basis for requirements analysis and specification. To evaluate the applicability and benefit of the developed model and the exemplary structure, we will further elaborate exemplary security topics for different scope areas.

4 References

1. Firesmith, D.G.: Security Use Cases. Journal of Object Technology, no. 2, pp. 53–64 (2003)
2. Mead, N. R., Hough, E. D., Stehney, T. R.: Security quality requirements (SQUARE) methodology. Pittsburgh, Pa: Carnegie Mellon University, Software Engineering Institute. (2005)
3. Sindre, G., Firesmith, D.G., Opdahl, A.L.: A Reuse-Based Approach to Determining Security Requirements. In Proc. 9th International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ'03). (2003)
4. Mellado, D., Fernández-Medina, E., Piattini, M.: A common criteria based security requirements engineering process for the development of secure information systems. Computer Standards & Interfaces, vol. 29, no. 2, pp. 244–253 (2007)
5. Boström, G., Wäyrynen, J., Bodén, M., Beznosov, K., Kruchten, P.: Extending XP practices to support security requirements engineering. In SESS '06 - Proceedings of the 2006 international workshop on Software engineering for secure systems, pp. 11–18 (2006)
6. Winograd, T., McKinley, H. L., Oh, L., Colon, M., McGibbon, T., Fedchak, E., Vienneau, R.: Software security assurance: A State-of-the Art Report (SOAR). Herndon, Virginia: Information Assurance Technology Analysis Center (2007)
7. Firesmith, D. G.: Engineering Security Requirements. Journal of Object Technology, vol. 2, no. 1, pp. 53–68 (2003)
8. Toval, A., Nicolás, J., Moros, B., García, F.: Requirements Reuse for Improving Information Systems Security: A Practitioner's Approach: SIREN. Requirements Engineering Journal, vol. 6, pp. 205–219 (2001)
9. Ministerio de Administrationes Públicas, MAGERIT – version 2: Methodology for Information Systems Risk Analysis and Management. II - Catalogue of Elements (2014, Jul. 02)
10. Dikanski, A., Abeck, S.: Towards a Reuse-oriented Security Engineering for Web-based Applications and Services. In The 7th International Conference on Internet and Web Applications and Services (ICIW 2012). (2012)
11. Velasco, V., Valencia-Garcia, R., Fernandez-Breis, J. T., Toval, A.: Modelling Reusable Security Requirements based on an Ontology Framework. In Volume 41, Issue 2, Journal of Research and Practice in Information Technology, pp. 119–133 (2009)

12. Schmitt, C., Liggesmeyer, P.: Implications of the Operational Environmental on Software Security Requirements Engineering. In Proceedings of WOSIS 2014 - 11th International Workshop on Security in Information Systems: SCITEPRESS, pp. 63–74 (2014)
13. Barnum, S., McGraw, G.: Knowledge for Software Security. IEEE Security and Privacy, vol. 3, no. 2, pp. 74–78 (2005)
14. NIST, Guide for Conducting Risk Assessments: NIST Special Publication 800-30, 1st ed. Gaithersburg, MD: U.S. Dept. of Commerce, National Institute of Standards and Technology (2012)
15. Information Security Forum, Information Risk Analysis Methodology. Available: https://www.securityforum.org/tools/isf-risk-manager (2015, Jan. 15)
16. ISO/IEC, ISO/IEC 27005:2011: Information technology -- Security techniques -- Information security risk management. Genève, Switzerland: ISO, (2011)
17. The MITRE Corporation, Common Weakness Enumeration (CWE). Available: http://cwe.mitre.org/ (2015, Jan. 15)
18. The MITRE Corporation, Common Vulnerabilities and Exposures (CVE). Available: http://cve.mitre.org/ (2015, Jan. 15)
19. Common Attack Pattern Enumeration and Classification (CAPEC) Library. Available: http://capec.mitre.org/ (2015, Jan. 15)
20. A. Sethi and S. Barnum, Introduction to Attack Patterns. Available: https://buildsecurityin.us-cert.gov/articles/knowledge/attack-patterns/introduction-to-attack-patterns (2015, Jan. 15)
21. OWASP, CLASP Security Principles. Available: https://www.owasp.org/index.php/CLASP_Security_Principles
22. NIST, Special Publications (800 series). Available: http://csrc.nist.gov/publications/PubsSPs.html (2015, Jan. 15)
23. Recommended security controls for federal information systems and organizations: SP 800-53, 3rd ed. [Gaithersburg, MD]: U.S. Dept. of Commerce, National Institute of Standards and Technology, 2009
24. SANS, Critical Security Controls. Available: http://www.sans.org/critical-security-controls/ (2015, Jan. 15)
25. McDermott, J., Fox, C.: Using abuse case models for security requirements analysis. In Computer Security Applications Conference, 1999. (ACSAC '99) Proceedings. 15th Annual, pp. 55–64 (1999)
26. Sindre, G., Opdahl, A. L.: Eliciting Security Requirements by Misuse Cases. In Proceedings of the 37th Conf. Techniques of Object-Oriented Languages and Systems, TOOLS Pacific 2000, pp. 120–131 (2000)
27. Schneier, B.: Attack Trees. In Dr. Dobb's Journal of Software Tools, pp. 21–29 (1999)
28. Swiderski, F., Snyder, W.: Threat modeling. Redmond, Wash: Microsoft Press (2004)