Security Engineering Methodology for Developing Secure Enterprise Information Systems: An Overview

Young-Gab Kim and Sungdeok Cha

College of Information and Communication, Korea University,
1, 5-ga, Anam-dong, Sungbuk-gu, 136-701, Seoul, Korea
{always,scha}@korea.ac.kr

Abstract. The software engineering discipline has provided principles, methodologies, and tools for the development of information systems. Software engineering have also become a fundamental component to produce information systems and related software components which are cheaper, better and faster. Recently, many forms of security attacks against information systems have emerged that attempt to compromise the security of information systems and organizations. However, traditional software engineering is not adequate and effective for developing secure information systems. In this paper, we propose holistic, consistent, and integrated security engineering procedures for analyzing, designing, developing, testing, and maintaining secure enterprise information systems. The proposed security engineering methodology combines security risk control, enterprise security architecture, and security management as an integrated framework.

Keywords: security engineering, enterprise security architecture, secure information system, security risk analysis, security management.

1 Introduction

As Internet and information technologies have become an increasingly important part of enterprise information systems, the security management of enterprise information systems has become a critical issue. Even though the software engineering discipline provides principles, methodologies, and tools for the development of information systems, it is not adequate and effective for designing, developing, and maintaining secure software systems [1]. The main reason is that in the majority field of software engineering and computer science, security concerns are treated as elective topics or lumped together with quality attributes or ethics. That is, in many cases, security features are treated as of secondary importance, and this neglect results in the system engineer developing buggy code with weak security measures [2]. Furthermore, software engineering does not support a holistic and integrated approach to control security features. For example, generally when system engineers build concrete complex and enterprise systems incrementally by refining and composing abstract and simple subsystems, security is not preserved in either the refinement or the composition [3].

James J. (Jong Hyuk) Park et al. (eds.), *EMC Technology and Service*,
LNEE 181, pp. 393–400, DOI: 10.1007/978-94-007-5075-3_47,

In addition, in order to develop secure enterprise information systems, the roles of the stakeholders are very important. However, most system engineers, especially software engineers, have no security-relevant knowledge about such issues as security risk analysis, and security mechanisms and services. They also do not consider security a functional requirement because processes for managing system development life cycles prioritize functional requirements over nonfunctional requirements [4]. Similarly, most security engineers do not have the systems-engineering background required to approach a security problem holistically [5]. Furthermore, they often do not consider the entire vulnerabilities that exist in information systems and outside threats in their entirety, focusing instead on particular defensive strategies and ignoring serious gaps in security architectures. For example, many security architectures concentrate exclusively on encryption technology to protect sensitive information while completely ignoring the threats to the availability of the critical functionality [6]. Besides, many large organizations, in many cases, adopt security solutions in the face of security attacks. However, these solutions are focused mainly on how to defend their systems against various inside and outside threats rather than on overcoming the causes of the security issues in the information systems [7].

Security engineering is a field of engineering that deal with building secure systems that will remain dependable in the face of malice, error, or mischance. It focuses on the tools, processes, and methods needed to design, implement, and test complete systems, and to adapt existing systems as their environment evolves [8]. However, security engineering, in practice, does not merge smoothly with the other engineering disciplines (e.g., software engineering, system engineering) [3]. In order to meet the need for secure enterprise information systems, we need new approaches to security engineering. In this paper, we propose a holistic, consistent, and integrated security engineering approach for developing secure enterprise information systems.

The rest of the paper is organized as follows. Section 2 surveys existing approaches to security engineering. In Section 3, we describe the layered security engineering model and the activities of security engineering in the system development life cycle (SDLC). Finally, Section 4 concludes the paper.

2 Related Work

Some approaches to providing security engineering have been proposed. First, there are several international standards [9-14] and a model [15, 16]: SSE-CMM (Systems Security Engineering – Capability Maturity Model) [9, 10] is a process model that describes the essential systems security processes and management tasks that any organization must perform. It is focused on the requirements for implementing security in software systems or related systems components.

ISO/IEC 15408 [11] is a common criterion (CC) for evaluating the security of information system or software, entitled “Information Technology – Security Techniques – Evaluation Criteria for IT Security.” It defines two security requirements: security functional requirements (SFR) [12] and security assurance requirements (SAR) [13]. Its purpose is to allow users to specify security requirements, developers to specify the security attributes of their products, and evaluators to determine whether products actually meet their claims.

ISO/IEC 27002 [14] is an information security standard, entitled "Information Technology – Security Techniques – Code of Practice for Information Security Management." It provides a best practice guide to information security controls. It has established guidelines and general principles for initiating, implementing, and improving information security management with an organization.

CLASP (Comprehensive, Lightweight Application Security Process) [15, 16] is a framework whose purpose is to provide an approach for locating security concerns in the early stages of the software development lifecycle. It actually consists of a set of process activities that can be integrated into any software development process. It also provides an extensive wealth of security resources that make implementing these activities reasonable.

However, these standards and a model are complex, and it is hard to understand and implement them. Furthermore, they do not explain how to implement each security control successfully.

Several other studies [7, 17-23] have focused on security requirements engineering as an early activity of software development. Nunes et al. [17] proposed a security engineering approach, named PSSS (Process to Support Software Security), based on the activities derived from SSE-CMM, ISO/IEC 15408, ISO/IEC 27002, and OCTAVE (Operationally Critical Threat, Asset, and Vulnerability Evaluation) [18, 19]. PSSS includes security activities such as software-based vulnerability, threat, and impact and risk assessments, which also contribute to information security strategic goals. Cheng et al. [20, 21] proposed an information security engineering environment (ISEE) based on ISO/IEC information security standards. ISEE integrates various tools and functions for supporting continuous and consistent design, development, and management of the security facilities of information systems with high security requirements. Wang et al. [7] proposed a process of security requirements that consists of nine steps and deals with the security requirements in the early stages of system design. They used a systematic approach to integrate software engineering process into develop security requirements. Mead et al. [22, 23] developed the security quality requirements engineering (SQUARE) methodology. SQUARE methodology provides a means for extracting, categorizing, and prioritizing security requirements for information systems and applications. It helps system engineers integrate security considerations into the early stages of the software development lifecycle.

Jurjens [24] proposed an approach to support model-based security engineering using UML (Unified Modeling Language) by providing tool-support for the analysis of UML models (i.e., UMLsec models) for security requirements. This approach utilizes the automated theorem-prover (ATP) SETHEO to verify the security properties of the UMLsec model, which make use of cryptography such as cryptographic protocols.

Schmidt [25] proposed a threat and risk-driven methodology to security requirement engineering. This methodology extends the security engineering process using patterns (SEPP) [26] by a threat and risk-driven procedure to select adequate security mechanisms. However, most of these approaches do not focus specifically on the combination of risk analysis and software engineering disciplines.

3 Proposed Approach

The main goal of the security engineering proposed in this paper is to provide procedures for analyzing, designing, developing, testing, and maintaining secure enterprise information systems. Fig.1 illustrates the basic concept of a layered security engineering model, which is composed of three important components: security management, enterprise security architecture (ESA), and security mechanism and services.

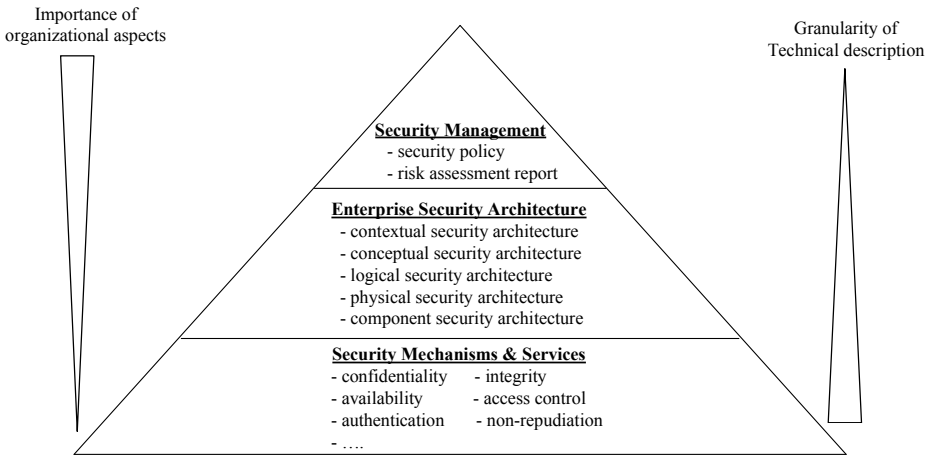

Fig. 1. Layered Security Engineering Model

Security management describes the specific needs for managing security risk, including the security policy, as the logical model of organization's business requirements for security and risk. The results of a security risk assessment, the risk assessment report, are also used to manage security risk mitigations (e.g., security mechanism, security service, and potential safeguards).

ESA is a main component of security engineering for implementing secure enterprise information systems. It provides the contextual, conceptual, logical, physical, component, and operational security of security-related policies, mechanisms, and procedures to give shape to a security management. The ESA proposed in this paper is expanded from the layered security architectures derived from SABSA (Sherwood Applied Business Security Architecture) [27] methodology.

Security Mechanisms & Services describes the detailed security technology for making concise enterprise security architecture to provide the confidentiality, integrity, and availability of an organization's information. Generally, security services are implemented as the objectives of security solutions. Thus, the compatibility and interoperability of security solutions are not guaranteed. That is, they do not provide holistic and integrated strategies. This paper proposes a way to map these security services into an ESA.

The layered security engineering model can be mapped into the activities of security engineering in the system development life cycle (SDLC) as illustrated in Fig. 2.

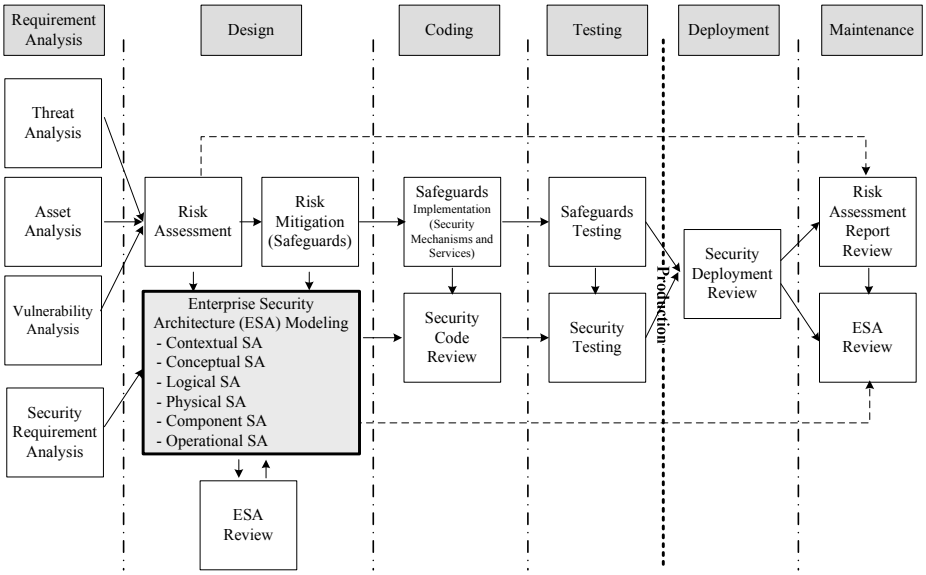

Fig. 2. Activities of Security Engineering in System Development Life Cycle (SDLC)

The proposed security engineering approach consists of several security-related activities that form a process to support the development of a more secure information system. The activity begins with security requirement analysis, which defines the required security properties, which the enterprise information systems should be satisfied. These security requirements play a role in the design of an ESA. Simultaneously, as the earlier activities of risk assessment, threat analysis, asset analysis, and vulnerability analysis are conducted in the requirement analysis phase of SDLC.

In the design phase of SDLC, the output of three activities (i.e., threat, asset, and vulnerability) is used to estimate security risk, and then risk mitigations (e.g., security mechanism, security service, and potential safeguards) are selected and implemented. Security risk assessment in this phase can identify potential security vulnerabilities and their impact. Moreover, the ESA, which is composed of the six-layered security architecture, is implemented. As mentioned earlier, the ESA proposed in this paper is expanded from the SABSA, and it gives a six by six matrix of cells, which represents the whole model for the enterprise security architecture. Attributes in the matrix are presented by 5W1H (what, why, how, who, where, and when) as follows:

- *What* are you trying to do at this layer? – The *assets* to be protected by the security architecture;
- *Why* are you doing it? – The *motivation* for wanting to apply security;
- *How* are you trying to do it? – The *functions* needed to achieve security;
- *Who* is involved? – The *people and organizational aspects* of security;
- *Where* are you doing it? – The *location* where you apply your security;
- *When* are you doing it? – The *time-related aspects* of security.

To implement secure information systems, correct design (i.e., enterprise security architecture) is required. To find vulnerabilities in the ESA design, an ESA review activity is conducted.

In the coding phase, the selected safeguards, including the security mechanisms and services in the previous phase, are implemented, and the code for security issues is thoroughly reviewed in terms of whether it contains security vulnerabilities.

In the testing phase, the implemented safeguards and the enterprise information system are tested by both white- and black-box testing in terms of whether there are bugs at the implementation phase, or whether the enterprise information system meets the security requirements based on the security risk analysis report. Security testing also contains tests of the security mechanism implemented in the previous phase.

The deployment phase constitutes the process that installs the system and makes it operational in the production environment. This phase can be one of the most critical in SDLC since the production environments and considerations for integration issues can sometimes be ignored. For example, sometimes the system runs differently in the production environment even though it has been validated and verified in the test phase. That is, when complex subsystems or large systems are integrated together, we need to ensure that the system is secure. To achieve this, a security deployment review based on the security requirements is conducted in the deployment phase.

The goal of security management is to identify, evaluate, and manage key security risks that impact an organization's stability and thus its ability to achieve its objectives and strategies. It is a continuous process of establishing risk management objectives, assessing risks within the context of established tolerances, developing strategies and implementing risk management processes, and monitoring and reporting upon those processes. In the maintenance phase, the risk assessment report, which is the summary and conclusions of the risk analysis, is utilized to manage the residential and potential security vulnerabilities in enterprise information systems. Furthermore, an ESA review activity is conducted.

4 Conclusion and Future Work

Nowadays, most of the research studies on the development of secure information systems are focused on isolated, inconsistent software engineering-related technologies. The security engineering methodology proposed in this paper covers the entire enterprise security-related activities for developing secure enterprise information systems. It combines security risk control, security enterprise architecture, and security management as an integrated framework. Moreover, the security mechanisms and services are designed, implemented, and supported as an essential part of the enterprise information systems. In this paper, we outline our ongoing work on an approach for the security engineering-based development of secure enterprise information systems. In future work, we aim to present the detailed tasks in each activity the of security engineering methodology.

Acknowledgements. This research was supported by the National IT Industry Promotion Agency (NIPA) under the program of Software Engineering Technologies Development.

References

1. Cheng, J.C., Goto, Y., Horie, D., Kasahara, T., Iqbal, A.: Development of ISEE: An Information Security Engineering Environment. In: Proc. of IEEE International Symposium on Parallel and Distributed Processing with Applications, pp. 505–510 (2009)
2. Mead, N.R., Hough, E.D.: Security Requirements Engineering for Software Systems: Case Studies in Support of Software Engineering Education. In: Proc. of the 19th Conference on Software Engineering Educations & Training, CSEET 2006 (2006)
3. Pavlovic, D.: The Unreasonable Ineffectiveness of Security Engineering: An Overview. In: Proc. of 2010 Software Engineering and Formal Methods, pp. 12–18 (2010)
4. Kim, Y.-G., Cha, S.: Threat Scenario-based Security Risk Analysis using Use Case Modeling in Information Systems. Security and Communication Networks 5(3), 293–300 (2012)
5. Stevens, J.L.B.: Systems Security Engineering. IEEE Security & Privacy, 72–74 (2011)
6. Evans, S., Heinbuch, D., Kyle, E., Piorkowski, J., Wallner, J.: Risk-Based Systems Security Engineering: Stopping Attacks with Intention. IEEE Security & Privacy (2004)
7. Wang, H., Jia, Z., Shen, Z.: Research on Security Requirements Engineering Process. In: Proc. of 16th International Conference on Industrial Engineering and Engineering Management (IE&EM 2009), Jiaozuo, China, pp. 1285–1288 (2009)
8. Anderson, R.: Security Engineering – A Guide to Building Dependable Distributed Systems, 2nd edn. Wiley Publishing, Inc. (2008)
9. International Organization for Standardization (ISO), Information Technology-Systems Security Engineering-Capability maturity Model (SSE-CMM), ISO/IEC 21827 (2008)
10. Carnegie Mellon University (CMU), `http://www.sse-cmm.org`
11. International Organization for Standardization (ISO), Information Technology-Security Techniques-Evaluation Criteria for IT Security – Part 1: Introduction and General Model, ISO/IEC 15408-1 (2008)
12. International Organization for Standardization (ISO), Information Technology-Security Techniques-Evaluation Criteria for IT Security – Part 2: Security Functional Requirements, ISO/IEC 15408-2 (2008)
13. International Organization for Standardization (ISO), Information Technology-Security Techniques-Evaluation Criteria for IT Security – Part 3: Security Assurance Requirements, ISO/IEC 15408-3 (2008)
14. International Organization for Standardization (ISO), Information Technology, Security Technical – Code of Practice for Information Security Managements, ISO/IEC 27002 (2005)
15. Secure Software, The CLASP Application Security Process. Secure Software Inc. (2005)
16. OWASP, `https://www.owasp.org/index.php/Category:OWASP_CLASP_Project`
17. Nunes, F.J.B., Belchior, A.D., Albuquerque, A.B.: Security Engineering Approach to Support Software Security. In: Proc. of 2010 IEEE 5th World Congress on Services, pp. 48–55 (2010)
18. Alberts, C., Dorofee, A.: Octave- The Operationally Critical Threat, Asset, and Vulnerability Evaluation, Canegie Mellon University – Software Engineering Institute
19. Alberts, C., Dorofee, A.: Managing Information Security Risks: The OCTAVE Approach. Addison-Wesley Professional (2003)
20. Cheng, J., Goto, Y., Morimoto, S., Horie, D.: A Security Engineering Environment Based on ISO/IEC Standards: Providing Standard, Formal, and Consistent Supports for Design, Development, Operation, and Maintenance of Secure Information Systems. In: Proc. of 2008 International Conference on Information Security and Assurance, pp. 350–354 (2008)

21. Horie, D., Goto, Y., Cheng, J.: Development of ISEE: An Information Security Engineering Environment. In: Proc. of 2009 Second International Symposium on Electronic Commerce and Security, pp. 338–342 (2009)
22. Mead, N.R., Hough, E.D., Stehney II, T.R.: Security Quality Requirements (SQUARE) Methodology., Technical Report (CMU/SEI-2005-TR-009), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA (2005)
23. Mead, N.R., Stehney II, T.R.: Security Quality Requirements Engineering (SQUARE) Methodology. In: Proc. of Software Engineering for Secure Systems (SESS 2005), St. Louis, MO (2005)
24. Jurjens, J.: Sound Methods and Effective Tools for Model-based Security Engineering with UML. In: Proc. of the 27th International Conference on Software Engineering, ICSE 2005, pp. 322–331 (2005)
25. Schmidt, H.: Threat- and Risk-Analysis during Early Security Requirements Engineering. In: Proc. of 2010 International Conference on Availability, Reliability and Security (ARES 2010), pp. 188–195 (2010)
26. Hatebur, D., Heisel, M., Schmidt, H.: A Security Engineering Process based on Patterns. In: Proc. of the International Workshop on Secure Systems Methodologies using Patterns (SPatterns), pp. 734–738 (2007)
27. Sherwood, J., Clark, A., Lynas, D.: Enterprise Security Architecture-A Business-Driven Approach. CMP Books (2005)