ISEE: AN INFORMATION SECURITY ENGINEERING ENVIRONMENT

Jingde Cheng, Yuichi Goto and Daisuke Horie
Department of Information and Computer Sciences, Saitama University, 255 Shimo-Okubo, Sakura-Ku, Saitama, Japan

Keywords: Security engineering, Information security engineering environment, ISO/IEC security standards, Formal methods.

Abstract: Security Engineering has some features that are intrinsically different from Software (Reliability) Engineering. Traditional software engineering environments are not adequate and effective for designing, developing, managing, and maintaining secure software systems. This position paper presents ISEE, an information security engineering environment we are developing, that integrates various tools and provides comprehensive facilities to support design, development, management, and maintenance of security facilities of information/software systems continuously and consistently, and guides and helps all users to perform their tasks regularly according to ISO/IEC security standards. The paper presents the basic ideas on development of ISEE, basic requirements for ISEE, and a design of ISEE. ISEE is the first real information security engineering environment.

1 INTRODUCTION

As opposed to Software (Reliability) Engineering to provide principles, methodologies, and tools for designing, developing, operating, and maintaining reliable software systems (Finkelstein&Kramer, 2000, IEEE-CS, 1990, Naur&Randell, 1969), Security Engineering intended to provide principles, methodologies, and tools for designing, developing, operating, and maintaining secure software systems.

Security Engineering has the following features that are intrinsically different from Software (Reliability) Engineering.

First, as Anderson pointed out, Software (Reliability) Engineering is about ensuring that certain things happen, while Security Engineering is about ensuring that they do not (Anderson, 2008).

Second, the reliability of a target software system is not necessarily an object of engineering that must be managed and/or controlled continuously, while the security of a target software system is necessarily an object of engineering that must be managed and/or controlled continuously. In general, we do not need to continuously consider improving the reliability of a target software system anytime, except the system has some failure or some new requirement is specified for the system. However, because assailants (crackers) are active persons who can get knowledge and skills day after day and then continuously attack target software systems always with new techniques, we have to continuously consider improving the security of a target software system anytime. This is particularly true in designing, developing, operating, and maintaining persistent computing systems (Cheng, 2005, 2005, 2006, 2007).

Third, the whole reliability of a target software system is usually the sum total of reliability of its all components, while the whole security of a target software system is not necessarily the sum total of security of its all components but usually only as good and strong as the weakest security of some component or link between components in the system.

Traditional software engineering environments (Devanbu&Stubblebine, 2000, Dittrich et al., 2000, Finkelstein&Kramer, 2000, Harrison et al., 2000) are not adequate and effective for designing, developing, operating, and maintaining secure software systems. To deal with those features in Security Engineering that are intrinsically different from Software (Reliability) Engineering, we have to find new principles, establish new methodologies, develop new tools, and integrates them into an engineering environment for designers, developers, administrators/end-users, and maintainers to design,

Cheng J., Goto Y. and Horie D. (2009).
ISEE: AN INFORMATION SECURITY ENGINEERING ENVIRONMENT.
In *Proceedings of the International Conference on Security and Cryptography*, pages 395-400
DOI: 10.5220/0002242103950400
Copyright © SciTePress

develop, operate, and maintain secure software systems.

We have proposed the general concept of an information security engineering environment (Cheng et al., 2008). However, no requirement analysis and definition for a real information security engineering environment has been done detailedly and no real environment has been designed and implemented.

This position paper presents ISEE, an information security engineering environment we are developing, that integrates various tools and provides comprehensive facilities to support design, development, management, and maintenance of security facilities of information/software systems continuously and consistently, and guides and helps all users to perform their tasks regularly according to ISO/IEC security standards. ISEE is the first real information security engineering environment.

The rest of this paper is organized as follows: Section 2 presents our basic ideas on development of ISEE, Section 3 presents basic requirements for ISEE, Section 4 presents a design of ISEE, and some concluding remarks are given in Section 5.

2 BASIC IDEAS ON DEVELOPMENT OF ISEE

An ***Information Security Engineering Environment*** is an engineering environment that integrates various tools and provides comprehensive facilities for designers, developers, administrators/end-users, and maintainers of information/software systems such that they can use the tools and facilities to ensure the whole security of the target system anytime consistently and continuously (Cheng et al,. 2008). Note that the major point we made in the above definition is "to ensure the whole security of the target system anytime consistently and continuously" that emphasizes the wholeness of security of information/software systems and the continuity and randomness concerning time.

We are developing ISEE based on the following considerations:

First, ISEE itself should be a persistent computing system (Cheng, 2005, 2005, 2006, 2007) to provide continuous supports for all users anytime. Design, development, management, and maintenance of security facilities of target systems should be continuously performed. It is not enough only to design and develop security facilities. Continuous maintenance (maybe including re-design and re-development) of target systems, in particular, those persistent computing systems (Cheng, 2005, 2005, 2006, 2007), are very important for protecting the systems from new attacks.

Second, ISEE should support all tasks in design, development, management, and maintenance of security facilities of target systems by guiding and helping all users (designers, developers, administrators/end-users, and maintainers of target systems) to perform their tasks according to ISO/IEC security standards. Because the whole security of any target system is only as good and strong as the weakest link in the system, some common standards shared and followed by all designers, developers, administrators/end-users, and maintainers of target systems are necessary to all tasks in engineering the systems in order to ensure the strength of the whole security of target systems. At present, the most established and widely accepted common standards are ISO/IEC security standards (ISO/IEC, 2005, 2007, 2008).

Third, ISEE should support all users to use formal methods as more as possible in design, development, management, and maintenance of security facilities of target systems. Because the strongest security of any target system can be obtained only by formal (mathematical) analysis and verification, formal methods should be used by all designers, developers, administrators/end-users, and maintainers of target systems as more as possible in design, development, management, and maintenance of target systems. Therefore, supports for formal methods in all tasks in engineering the systems should be provided by ISEE.

Finally, ISEE should support all tasks of all users consistently. Because the security of any target system is required at all phases of the information cycle, i.e., gathering, creating, processing, storing, transmitting and deleting, the consistency among all tasks of designers, developers, administrators/end-users, and maintainers of target systems must be kept and maintained. Therefore, all supports for all tasks in engineering the systems must be consistent.

3 BASIC REQUIREMENTS FOR ISEE

We analyzed and defined basic requirements that ISEE should satisfy as follows:

R1: ISEE must provide tools and facilities to continuously support all tasks in design, development, management, and maintenance of security facilities of target systems.

The reason to require ISEE to satisfy requirement **R1** has been explained in Section 1 and Section 2. Here "to continuously support" means that ISEE should support all tasks in design, development, management, and maintenance of security facilities of a target system through the whole lifetime of the system, and should be ready at anytime for any request from any user.

R2: ISEE must provide tools and facilities for all users to design, develop, manage, and maintain security facilities of target systems regularly according to ISO/IEC security standards.

To ensure the whole security of any target system, some common standards for all components of the system, all tasks in design, development, management, and maintenance of security facilities of the system, and all tools of ISEE are indispensable. At present, ISO/IEC security standards are the best choice. On the other hand, maybe there is no ISO/IEC security standard effective for some tasks. In those cases, it is needed to look for other effective standards.

R3: ISEE must provide tools and facilities for all users to use formal methods as more as possible in design, development, management, and maintenance of security facilities of target systems to check whether the security facilities are certainly consistent with ISO/IEC security standards or not.

The formal methods mentioned above may be well established formal representation, verification, or reasoning methods, originally provided by ISEE, or developed by users themselves.

R4: ISEE must provide tools and facilities for all users to perform their tasks in a guided regular order.

The whole security of a target system is ensured only if its designers, developers, administrators/end-users, and maintainers perform appropriate tasks in a regular order. To guide users to perform their tasks in a previously defined regular order can decrease failure in the target system.

R5: ISEE must provide tools and facilities for its users to manage and maintain security facilities of target systems as rapidly as possible.

Because any delay of response to crackers' attacks must increase the possibility of information leaks, falsification, or destruction, management and maintenance of security facilities of target systems should be performed as rapidly as possible.

4 DESIGN OF ISEE

We designed ISEE according to the above basic requirements. First of all, based on ISO/IEC 12207 (ISO/IEC, 2008) we clarified concrete tasks in design, development, management, and maintenance of security facilities of target systems as follows:

Infrastructure Management
Human Resource Management
Quality Management
Risk Management
Information Management
Software Requirement Analysis
Software Architectural Design
Software Detailed Design
Software Implementation
Software Qualification Testing
Software Installation
Software Integration
Software Construction
Software Configuration Management
Software Operation
Software Maintenance
Software Disposal
Software Documentation Management
Software Quality Assurance
Software Verification
Software Validation
Software Review
Software Problem Resolution
Recovery
Environment Management

We also clarified ISO/IEC security standards for the concrete tasks. Then, we defined a model of software life cycle processes that specifies all tasks with a right order in design, development, management, and maintenance of security facilities of target systems (Horie, 2009). Finally, we specified functions of components of ISEE.

Fig. 1 shows all tasks with the right order and Fig. 2 shows all components of ISEE and relationships among them.

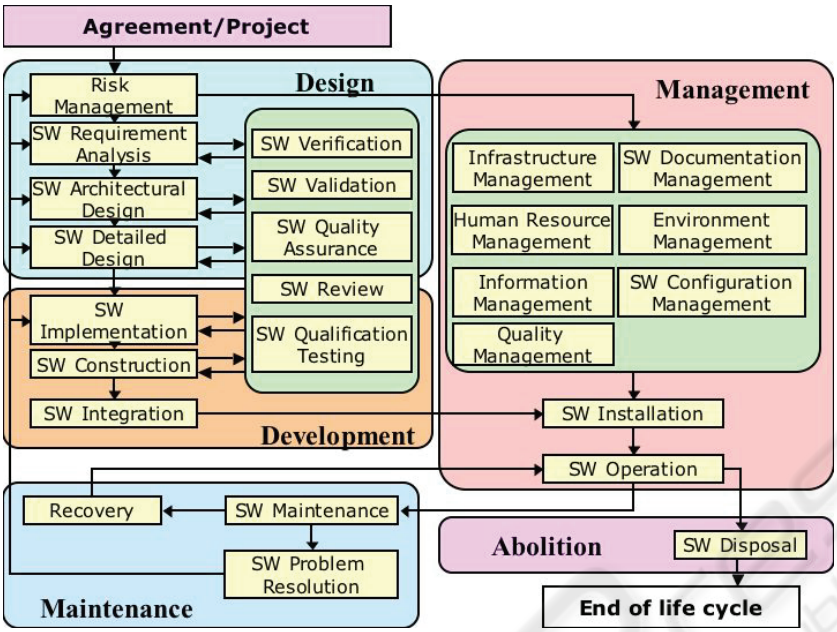

Figure 1: Tasks with a right order in design, development, management, and maintenance.

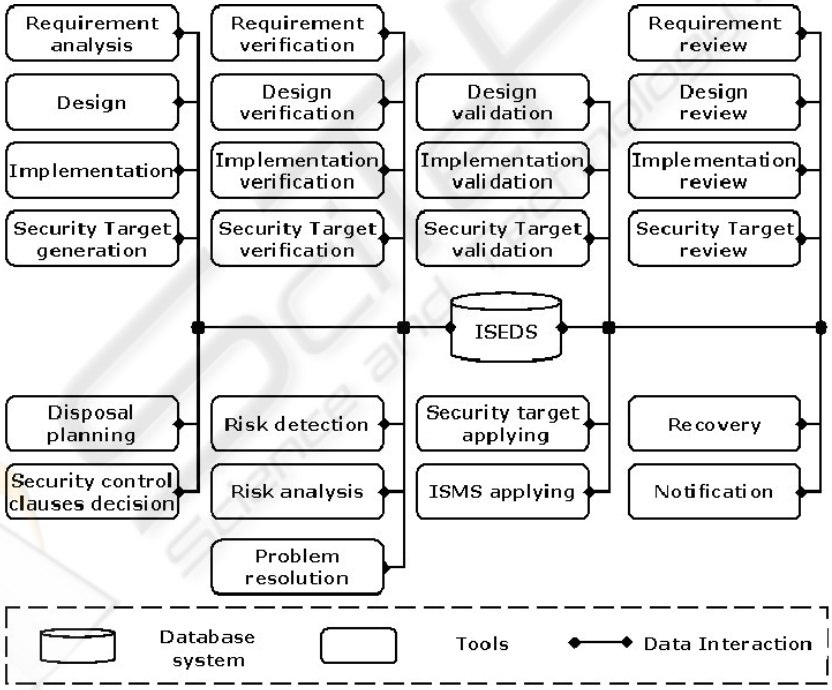

Figure 2: Components of ISEE.

ISEE consists of the following components:

ISEDS is an information security engineering database system that manages all common data in ISEE (Horie, 2008). ISEDS manages data of ISO/IEC security standards and their related documents, data of published cases (e.g., STs, ISMS documents, and so on), and data acquired by users in their tasks. ISEDS is a main component of ISEE and plays the most important role in ISEE. All users of ISEE can easily retrieve and use data in ISEDS by using various tools of ISEE.

Requirement analysis tool supports users to create risk analysis documents.

Requirement verification tool supports users to formally verify requirement specification documents.

Requirement review tool supports stakeholders to review requirement documents.

Design tool supports users to generate templates of design specification documents.

Design verification tool supports users to formally verify design specification documents.

Design validation tool supports users to validate design specification documents.

Design review tool supports stakeholders to review design specification documents.

Implementation tool supports users to create source codes.

Implementation verification tool supports users to formally verify source codes.

Implementation validation tool supports users to validate source codes.

Implementation review tool supports stakeholders to review source codes.

ST generation tool supports users to generate templates of STs (Security Targets).

ST verification tool supports users to formally verify STs.

ST validation tool supports users to validate STs.

ST review tool supports stakeholders to review STs.

Risk detection tool receives information acquired in management of target systems and automatically detects risks that have not been assumed in advance according to ISO/IEC security standards.

Risk analysis tool supports users to create documents of detected risks.

Disposal planning tool supports users to decide plan of system disposal and generates documents for disposal plan.

Problem resolution tool enumerates proposals of problem resolution according to documents of risk analysis and cases of problem resolution and generates problem resolution documents.

Security control clause decision tool supports users to create ISMS (Information Security Management Systems) documents.

ST applying tool guides users to apply ST certification.

ISMS applying tool guides users to apply ISMS certification.

Recovery tool supports users to create documents for recovery plan, and automatically backup and recover data of information systems and information assets.

Notification tool notifies stakeholders information of system disposal and recovery.

5 CONCLUDING REMARKS

We have presented our basic ideas on development of ISEE: an information security engineering environment we are developing, basic requirements for ISEE, and a design of ISEE. This position paper presented our ongoing work. Some techniques and tools have been developed (Horie, 2008, 2009, 2009, Morimoto et al., 2007, 2008, Yajima et al. 2009) and other techniques and tools are being developed.

In the future, an integrated engineering environment combining ISEE and some traditional software engineering environment will provide full powerful supports for design, development, operation, and maintenance of information/software systems with high reliability and security requirements.

REFERENCES

Anderson, R. J., 2008. *Security Engineering: A Guide to Building Dependable Distributed Systems*, John Wiley & Sons, 2nd edition.

Cheng, J., 2005. Connecting Components with Soft System Buses: A New Methodology for Design, Development, and Maintenance of Reconfigurable, Ubiquitous, and Persistent Reactive Systems. In *Proceedings of the 19th International Conference on Advanced Information Networking and Applications*, Vol. 1, pp. 667-672. IEEE Computer Society Press.

Cheng, J., 2005. Comparing Persistent Computing with Autonomic Computing. In *Proceedings of the 11th International Conference on Parallel and Distributed Systems*, Vol. II, pp. 428-432. IEEE Computer Society Press.

Cheng, J., 2006. Persistent Computing Systems as Continuously Available, Reliable, and Secure Systems. In *Proceedings of the 1st International Conference on Availability, Reliability and Security*, pp. 631-638. IEEE Computer Society Press.

Cheng, J., 2007. Persistent Computing Systems Based on Soft System Buses as an Infrastructure of Ubiquitous Computing and Intelligence (Invited Paper). *Journal of Ubiquitous Computing and Intelligence*, Vol. 1, No. 1, pp. 35-41. American Scientific Publishers.

Cheng, J. Goto, Y., Morimoto, S., and Horie, D., 2008. 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 *Proceedings of the 2nd International Conference on Information Security and Assurance*, pp. 350-354. IEEE Computer Society Press.

Devanbu, P. T., Stubblebine, S., 2000. Software Engineering for Security: A Roadmap. In *Proceedings of the Conference on The Future of Software*

Engineering, *International Conference on Software Engineering,* pp. 227-239. ACM Press.

Dittrich, K., Tombros, D., Geppert, A., 2000. Databases in Software Engineering: A Roadmap. In *Proceedings of the Conference on The Future of Software Engineering*, *International Conference on Software Engineering,* pp. 291-302. ACM Press.

Finkelstein, A., Kramer, J., 2000. Software Engineering: A Roadmap. In *Proceedings of the Conference on The Future of Software Engineering*, *International Conference on Software Engineering,* pp. 3-22. ACM Press.

Harrison, W., Ossher, H., Tarr, P., 2000. Software Engineering Tools and Environments: A Roadmap. In *Proceedings of the Conference on The Future of Software Engineering*, *International Conference on Software Engineering,* pp. 261-277. ACM Press.

Horie, D., Morimoto, S., Azimah, N., Goto, Y., Cheng, J., 2008. ISEDS: An Information Security Engineering Database System Based on ISO Standards. In *Proceedings of the 3rd International Conference on Availability, Reliability and Security*, pp. 1219-1225. IEEE Computer Society Press.

Horie, D., Yajima, Kasahara, T., Goto, Y., Cheng, J., 2009. A New Model of Software Life Cycle Processes for Consistent Design, Development, Management, and Maintenance of Secure Information Systems. In *Proceedings of the 8th IEEE/ACIS International Conference on Computer and Information Science*, to appear, IEEE Computer Society Press.

Horie, D., Yajima, K., Azimah, N., Goto, Y., Cheng, J., 2009. GEST: A Generator of ISO/IEC 15408 Security Target Templates. *Studies in Computational Intelligence*, to appear, Springer-Verlag.

IEEE Computer Society, 1990. *IEEE Standard 610: IEEE Standard Computer Dictionary – A Compilation of IEEE Standard Computer Glossaries*. IEEE Computer Society Press.

IEEE Computer Society, 1990. *IEEE Standard 610.12-1990: IEEE Standard Glossary of Software Engineering Terminology*. IEEE Computer Society Press.

ISO/IEC, 2005. *ISO/IEC 15408-1:2005: Information Technology - Security Techniques - Evaluation Criteria for IT Security*.

ISO/IEC, 2005. *ISO/IEC 27001: Information technology - Security techniques - Information security management systems*.

ISO/IEC, 2005. *ISO/IEC 27002: Information technology - Security techniques - Code of practice for information security management*.

ISO/IEC, 2007. *ISO/IEC 27006: Information technology - Security techniques - Requirements for bodies providing audit and certification of Information Security Management Systems*.

ISO/IEC, 2008. *ISO/IEC 12207: Systems and Software Engineering Software Life Cycle Processes*.

ISO/IEC, 2008. *ISO/IEC 27000: Information technology - Security techniques - information security management systems - Overview and vocabulary*.

ISO/IEC, 2008. *ISO/IEC 27005: Information technology - Security techniques - Information security risk management*.

Morimoto, S., Shigematsu, S., Goto, Y., Cheng, J., 2007. Formal Verification of Security Specifications with Common Criteria. In *Proceedings of the 22nd Annual ACM Symposium on Applied Computing*, pp. 1506–1512. ACM Press.

Morimoto, S., Shigematsu, S., Goto, Y., Cheng, J., 2008. Classification, Formalization and Verification of Security Functional Requirements. In V. Geffert et al. (Eds.), SOFSEM 2008: Theory and Practice of Computer Science, 34th Conference on Current Trends in Theory and Practice of Computer Science, Novy Smokovec, High Tatras, Slovakia, January 19-25, 2008, Proceedings. *Lecture Notes in Computer Science*, Vol. 4910, pp. 622-633. Springer-Verlag.

Naur, P., Randell, B., (Eds.), 1969. *Software Engineering: Report of a conference sponsored by the NATO Science Committee, Garmisch, Germany, 7-11 Oct. 1968*. NATO.

Yajima, K., Morimoto, S., Horie, D., Azreen, N. S., Goto, Y., Cheng, J., 2009. FORVEST: A Formal Verification Support Tool of Security Specifications with ISO/IEC 15408. In *Proceedings of the 4th International Conference on Availability, Reliability and Security*, pp. 624-629. IEEE Computer Society Press.