International Journal of Security and Its Applications
Vol.8, No.1 (2014), pp.355-362
http://dx.doi.org/10.14257/ijsia.2014.8.1.33

Comparison of Secure Development Frameworks for Korean e-Government Systems

Dongsu Seo

School of Information Technology
Sungshin University
dseo@sungshin.ac.kr

Abstract

Recently, new development guides have been announced for building secure e-government systems by the Korean government. The guides aim to reduce significant amount of security vulnerabilities and weaknesses inside source codes using secure coding practices. In this paper, we present an overview of the new development guides from various methodological perspectives. We also provide results of comparison to major security enhanced frameworks including CLASP, SAMM, and other domestic security frameworks.

Keywords: *Secure Development Guide, Secure SDLC, Comparison*

1. Introduction

It has been reported that the rate of cyber attacks in 2011 has dramatically increased up to 81% compared to the attacks in 2010[1]. Typical responses to cyber attacks have been focused on post-disaster activities, such as publishing security patches, enhancing firewall capacity, or buying network security equipment, etc. The problem with the post-disaster activities is that we have little knowledge of defending methods if the attack gains a new form of attacking methods, which is known as zero-day attacks.

Research surveys show that the frequency of software vulnerability can be reduced as much as 80% when software engineers put more efforts on coding and design activities earlier in software development lifecycle [2]. Static analysis technique is a good example of such efforts. By its term, static analysis refers to any techniques to assess code without executing it [3]. Static analysis is especially important when there is a requirement from project owners for elimination of code weakness for the sake of code quality. Since 2012, the Korean government has announced a new form of development guidelines for software developers who involve in developing e-government systems. The guides contain a list of vulnerability reported harmful, and encourage software developers to use static analysis techniques in order to produce safe codes resilient to cyber attacks.

In this paper, we will discuss the key elements of four frameworks, and provides implementation characteristics of each.

- *Comprehensive, Lightweight Application Security Process (version 1.0)*
- *The Software Assurance Maturity Model (version 1.0)*
- *Guide for e-Governmental Information Security Management Systems (2013 edition)*
- *Software Development Security Guide for Developers and Managers for e-Government Software (2012 edition)*

ISSN: 1738-9976 IJSIA

We also suggest comparison criteria for these development frameworks in order to help developers with selecting appropriate methods. This paper is organized as follows. Section 2 introduces concepts and structure of four major security-enhanced development frameworks including two Korean governmental frameworks. In Section 3, comparison of four development schemes will be provided. Finally, Section 4 concludes research findings and describes suggestions.

2. Overview of Existing Secure Development Frameworks

Before discussing structures and activities of secure development frameworks in detail, it is worth introducing roles and scopes of the development frameworks. Many practitioners agree that security frameworks have expanded to include not only detail descriptions on process activities but also management and policy issues on security. Kirk[11] provides definitions of information security frameworks by stating that they are comprehensive security models consisting three basic components: *process, technology, and people*, and objectives that ensure overall security of information is there by eliminating business risks. We have selected four of major security frameworks that fit into such concepts of security frameworks including CLASP, SAMM, G-ISMS and G-SDSG. Here, we present concise descriptions of these four security frameworks.

Comprehensive, Lightweight Application Security Process (CLASP)

CLASP[4] is a framework focusing security concerns at early stages of development lifecycle. CLASP is based around activity-centric approaches that discompose enterprise-wide security goals into set of small activities. Twenty four security-related activities have been defined, and each individual activity is then integrated into software development processes. CLASP is comprehensive because it describes role of all stakeholders of security issues along the all necessary activities spanning over the SDLC. The range of stakeholders defined in CLASP includes project managers, requirement engineer, architect, and so on. CLASP is flexible enough to integrate each activity into individual development processes while maintaining existing development structures coexist.

Software Assurance Maturity Model, SAMM (version 1.0)

SAMM[6] is another comprehensive approaches aiming to make secure information systems. Four business functions constitute SAMM frameworks, *i.e.*, governance, construction, verification, and deployment. SAMM has its motivations from well known best practices which in turn form individual activities, namely security practices as shown in Table 1.

Table 1. SAMM Structure

Business Function	Governance	Construction	Verification	Deployment
Security Practices	Strategy & Metrics	Security Requirements	Design Review	Vulnerability Management
	Education & Guidance	Threat Assessment	Code Review	Environment Hardening
	Policy & Compliance	Secure Architecture	Security Testing	Operational Enablement

SAMM was established and maintained by the Open Web Application Security Project (OWASP). Advantages of public security frameworks such as SAMM are two folds: First, they are flexible enough to encompass best practices available in public domains. Second, they are continually updated released based on a free- license policy.

Guide for e-Governmental Information Security Management Systems(G-ISMS) (2013 version)

Information security system (ISM) provides a systematic approach to protect systems and information from internal and external threats [12]. Nowadays, the importance of ISM is increasing within organization, particularly within governmental service body. G-ISMS [7] is many years efforts on the matter of ISM in Korea. G-ISMS is composed of a set of guidelines for managing and operating governmental information resources. It establishes procedures and mechanisms to protect organizational information resources. It also provides governmental organizations with various ranges of check points on security centric activities such as planning, monitoring and operating information security management activities in order to strengthen security protection level. G-ISMS is used as certification criteria conducted by independent evaluation teams.

Software Development Security Guide for Developers and Managers for e-Government Software (G-SDSG) (2012 version)

G-SDSG[8] has been announced by Korean Government in 2012 in the hope of producing secure administrative systems by eliminating security weakness inside source code. Code weakness is a kind of software error resulting in critical security flaws where static analysis techniques are adopted in detecting software weakness. The guide aims to support both software developers and project owners in the area of e-government related systems. The role of the guideline falls into the following:

- defines stakeholders and organizations who comply with the guide
- explains scope and activities for secure coding
- provides understanding of software weakness
- suggests mandatory list of software weaknesses to be resolved

G-SDSG is unique in a sense that it requires software developers mandatorily to eliminate code weaknesses stated in the guide. G-SDSG updates the software weakness every year by consulting publications from OWASP, SANS, CWE, and NVD.

Table 2. Mandatory List of Software Weakness in G-SDSG in 2012

Categories	Examples
Input data validation and expression	SQL insertion, Resource insertion, Cross-site scripting, Dangerous file upload, XQuery insertion, Xpath insertion, LDAP insertion, Cross-site request forgery, Directory path manipulation, HTTP request partition, Integer overflow, Input data forgery
Security function	Invalid authorization, Insecure cryptography algorithm, Insufficient key length, Invalid random number, password save in plain text, hard coded encrypt key, Insecure password, Information disclosure through cookie, Hash function without salt, Download code without verification
Time and state	TOCTOU, Recursion without control
Error handling	Information disclosure through error message, Invalid error handling, Absence of error handling
Code error	Null pointer dereference, Invalid resource free
Encapsulation	Data disclosure through wrong session, Private array return from public method, Public data assignment to private array, Undeleted debugging code, System data disclosure
Abuse of API	Security decision from DNS lookup

In G-SDSG, project owners are responsible to set security objectives and scope, and acceptance criteria of deliverables. Developers should obey all requirements stated in the guide such as performing code checking and testing, and producing evidences that security requirements have been successfully fulfilled. It is responsibility if audit team who should examine and confirms that the developers have been corrected implemented and all the security weakness have been removed correctly.

Table 3. Roles and Activity for Stakeholders

SDLC Phase	Project owner	Developer	Audit
Planning	Planning for Secure development	N/A	N/A
Contract	Inserting security requirement on RFP & contracting	Specifying secure development requirement plan and schedule	Audit planning
Development	Checking SW secure development compliance	Training developers Eliminating of SW weakness	Mid-audit for Secure coding
Audit	N/A	N/A	Final audit for secure coding

3. Assessment Criteria and Results

In this paper we introduce comparison criteria that can clarify characteristics of each security frameworks. First, we consider security awareness is an important feature as far as security frameworks concern. Security policies and objectives vary each other. For example, G-SDSG seems to focus on code quality by adopting code checking techniques, whereas approaches shown in CLASP and SAMM deal with policies and regulations with operational considerations. Second, applicability of frameworks relates methodological flexibility explaining how easily to integrate the frameworks into existing development processes. Many practitioners want to use security enhanced activities as a complement of legacy development process, not as a substitute of the existing one. Third, it is also important to see how the framework constitutes its structures by terms of techniques, tools, and standards that the framework may have been influenced or interacted with.

Security Awareness over SDLC: CLASP suggests software engineers to establish security policy at early stage of development, and all the activities later on should be considered within the range of security policy. CLASP identifies software vulnerabilities that need to examine over design and coding phases. In that sense, it is comprehensive, covering most of development lifecycles.

G-ISMS is particularly focused on resource identification and attack surfaces, and related vulnerability on the resource at planning stages, and verifies such ISMS panning is performed accordingly at operational stages, and thus, emphasizes on operation stages.

Table 4. Comparison of Security Awareness over SDLC

Framework	Planning	Requirement	Design	Coding/ Testing	Operation
CLASP	○	○	○	○	×
SAMM	○	○	○	○	×
G-ISMS	○	×	×	×	○
G-SDSG	○	×	×	○	○

Applicability: We have seen that two of the stated frameworks, such as G-ISMS and G-SDSG are applied in the area of government-driven sectors. Both frameworks have well-defined steps of conformation and certification, and thus require independent organization to perform evaluation activities. In such purposes, methodological flexibility is very limited for both government-driven frameworks.

Table 5. Comparison by Applicability

Framework	Conformance	Domain	Flexibility	Certificates
CLASP	Optional	Non Governmental	High	No
SAMM	Optional	Non Governmental	Medium	No
G-ISMS	Mandatory	Governmental	Low	Yes
G-SDSG	Mandatory	Governmental	Low	Yes

It is still controversial that simply imposing strong regulation on developing information system can help security vulnerability decrease. For instance, techniques such as static analysis and penetration test have been used to enhance security features against malicious attacks, not as acceptance or conformance criteria.

Methodological Issues: CLASP and SAMM are comprehensive in that they concern security issues over all SDLC, and do not confine particular tools at specific phases. On the other hand, G-ISMS heavily relies on document-oriented review process, thus it has not strong relation to any type of tools. G-SDSG requires developers and managers to follow regulations that code checking tools should be certified under Common Criteria (CC) if applicable.

Table 6. Comparison by Applicability

Framework	Techniques	Tools	Standard
CLASP	Secure coding, Threat modeling, Review	Not confined	RUP
SAMM	Secure coding, Threat modeling, Review	Not confined	N/A
G-ISMS	Review & inspection	N/A	ISMS
G-SDSG	Secure Coding	Static Analyzer	CC

From the viewpoint of SDLC coverage, G-ISMS and G-SDSG partially support lifecycle phases whereas CLASP and SAMM aim to cover whole SDLC. This is because G-ISMS takes account of operational characteristics of information systems, and G-SDSG focuses on both implementation phases where assurance of source code quality is important.

4. Related Works

Urbaczewski[8] has suggested a comparison criteria for enterprise architectural frameworks from the viewpoint of their output results and the development process for deliverables. There is another approaches for comparison of security-oriented methodologies suggested by Mead[9]. In her paper, comparison criteria should encompass quality features as well, together with lifecycle coverage issues. These include adaptability, CASE supports, client acceptance complexity, graphical output, learning curve, maturity and finally scalability.

5. Conclusions

In this paper, we have provided a brief introduction to some of security-related frameworks and compared characteristics of each framework. It certainly is difficult to compare different development frameworks because each has its own purposes and scope of supporting activities. Government driven frameworks such as G-ISMS and G-SDSG seem to have unique features to compare with non-governmental frameworks. The governmental-driven frameworks have more narrow view for security targets, and more strict criteria on quality issues. Nevertheless, good secure development guides should provide clear objectives, and target users, and well defined set of tool support, and verification guideline with developers as well as project managers. In that viewpoint, the governmental security frameworks are recommended to have benefits from other methods. For instance, SAMM provides wider and long-term perspective on SDLC by reflecting good practices from various organizations.

Market driven approaches such as CLASP, and SAMM have both strength and weakness to compare with government driven framework. We believe methodological flexibility and good SDLC coverage are key features to success, and it is recommended for G-SDSG and G-ISMS to reflect best practices from them as complement activities.

Acknowledgements

This work was supported by the Sungshin Women's University Research Grant of 2012

References

[1] Symantec, Internet Security Threat report, 2011 Trends, vol. 17, **(2012)** April.
[2] NIST, "The Economic Impacts of Inadequate Infrastructure for Software Testing", **(2002)** May.
[3] B, Chess and J, West, Secure Programming with Static Analysis, Addison Wesley, **(2007)**.
[4] The CLASP Application Security Process, Secure Software Inc, **(2005)**.
[5] J. Zachman, "A Framework for Information Systems Architecture", IBM Systems Journal, vol. 38, **(1999)**, pp. 454-457.
[6] Software Assessment Maturity Model (SAMM) version 1.0, OWASP, **(2009)** March.
[7] Electronic Governmental Information Security Management Systems, Ministry of Security and Administration, **(2012)**.
[8] Software Development Security Guide for Developers and Managers for e-Government Software (G-SDSG) Ministry of Security and Administration, **(2013)**.
[9] L. Urbaczewski and S. Mrdali, "A Comparison of enterprise Architecture Frameworks", Issues in Information Systems, vol. 7, no. 2, **(2006)**.
[10] N. Mead, "How To Compare the Security Quality Requirements Engineering (SQUARE) Method with Other Methods", CMU/SEI-2007-TN-021, SEI, **(2007)**.
[11] K. Kark, O. Stamp, L. Koetzle and J. Mulligan, "Defining A High-Level Security Framework: Putting Basic Security Principles To Work", **(2008)** December.
[12] S. Barlas and R. Queen, "Top 10 Technology Concerns", Strategic Finance, vol. 88, no. 10, **(2007)**.

Author

Dongsu Seo received his M.Sc. and Ph.D. from the University of Manchester, England. Currently, he is a Professor at the School of Information Technology at Sungshin Women's University, Seoul Korea. His research interests are software engineering, multimedia application, and security engineering.