Source code embedded (SCEM) security framework

Tanveer A. Zia
Charles Sturt University

Aftab Rizvi
Risk Associates Pty Ltd

DOI: 10.4225/75/57b55002cd8d0
9th Australian Information Security Management Conference, Edith Cowan University, Perth Western Australia, 5th -7th December, 2011

SOURCE CODE EMBEDDED (SCEM) SECURITY FRAMEWORK

Tanveer A Zia[1] and Aftab Rizvi[2]
[1]School of Computing and Mathematics, Charles Sturt University, NSW, Australia
[2]Risk Associates Pty Ltd, Bella Vista NSW, Australia
tzia@csu.edu.au, aftab.rizvi@riskassociates.com.au

Abstract

Security in the Software Development Life Cycle (SDLC) has become imperative due to the variety of threats posed during and after system design. In this paper we have studied the security in system design in general and software development in particular, and have proposed strategies for integration of security in the SDLC. The paper highlights the needs of embedding security right from the earlier processes in the SDLC because patches and controls after the software delivery are more expensive to fix. We propose Source Code EMbedded (SCEM) security framework to improve the design of security policies and standards for the software development process to ensure the security and reliability in government departments such as taxation, auditing, national security, social security, and immigration. It is also envisaged that the implementation of SCEM security framework will ensure commercial and public trust in the software development process within Australia and worldwide, saving enormous redevelopment costs.

Keywords
Secure software development; security by design; security framework; data security; software security methodologies

INTRODUCTION

If the threats inherent in current software development methodologies are not addressed in the early stages of the software development life cycle (SDLC), the costs are significant. The formulation of a framework to integrate security from the beginning is needed. An embedded security framework in the development process will ensure that any software project can be audited and security assured. From the software security assurance perspective, the only practical approach to security assurance which is currently adapted by the security assurance industry is software vulnerability scanning and testing. This approach of security assurance requires the software to be physically present which means that we have already spent most of the budgeted time and cost in the development. If after performing those tests vulnerabilities are found, fixing those vulnerabilities would incur additional time and significant costs or may require the software to be totally scrapped and built from scratch again. At this stage we can safely conclude that the current methods of assurance are time lagging and unless the security tests are passed, would always incur additional time and costs to ensure compliance to any information security policy or standard.

Information security revolves around the famous CIA triad (Pfleeger and Pfleeger 2007). The CIA triad dictates the three facets of information security which are: Confidentiality, Integrity and Availability of information. For any software to be considered secured it must conform to these three broader and high level characteristics (Kothanath 2006). Currently, no Software Development Methodology (SDM) (Roberts et. al 2001) or implementation of Software Development Life Cycle (SDLC)) (Royce 1970) provides any directions to integration of these three security characteristics in an identifiable and auditable manner.

There are several studies and software tools which allow the implementation of a specific type security in SDLC and SADM (Software Analysis & Design Methodology) to be implemented (Banerjee and Pandey (2010, 2009), Rehman and Mustafa (2009), Shin and Williams (2008), Mellado et al (2010)). However, a holistic approach to cover all phases in the SDLC was much anticipated. The SCEM security framework equally enforces the existing tools to be updated and new tools to be developed to enable "security by design" features. Once this framework becomes part of common practice (which the potential dictates it will) then it would be about time where the implementation tools e.g. databases bring about changes like, adding new properties purporting to the security features established by the security framework i.e. having new features like "isEncryptedField" added to the schema hence enabling seamless mapping of an analysis requirement at the first stage of the SDLC into the database level implementation at the finishing stages.

It is also anticipated that new security certifications in the area of Secured Software Analysis and Design (unlike Secured Software Development which is purely technical) would need to be introduced to enhance the skill sets of Business Analysts, Chief Technology Officers (CTOs), Project Managers, Security Consultants to implement better security and security by design and also enabling and helping the information security industry to speak the same language.

The non-existence of formal tools and strategies to integrate the CIA triad as distinct features of the software project during analysis and design phases are costing the following organizations money, time, competitive advantage, face value and in some cases loss of public trust:

1. government organizations;
2. semi-government organizations; and
3. specialized information security industry

The information security is currently in need of enhancement in the existing SDLC or vice versa where the industry can get involved in the security project earlier and save the organizations from the aforementioned problems much earlier. Currently, even if an organization wants to induce security consultants during the analysis and design stage they cannot be of much help due to the following reasons:

1. the security consultants are unaware of any standard analysis and design tools (due to non-existence) to be able to identify the security strategies at design or requirement specification phases of the SDLC;
2. the business analysts are not aware of methods and tools to be able to:
    a. firstly, understand what the security risks are;
    b. and secondly, be able to not only identify but explicitly explain the relationship between a security risk and business requirements/processes.

In essence, the integration of security strategies as a security framework in Software Development Life Cycle would allow any security anomalies to be detected and fixed well before the software has been developed. The SCEM security framework will also allow the project to be audited for conformance at any stage of the life cycle hence not only providing greater security but saving time, costs and resources which might be incurred on re-development or patching of the software once it has been fully developed.

The development of such a framework would greatly relieve specially the government organizations from poorly designed applications in terms of security, not only saving the people’s money from getting wasted into securing the application after development has completed but also ensuring and increasing people’s trust in the government’s ability to provide ample security for critical information.

Other than government organizations, any organization which is willing to secure their information from both internal and external risks can then explicitly ask the Software Development Service providers to follow the SCEM security framework for software development ensuring that their software remain secure from the very beginning of the project. This will also ensure that project documentation can then be evaluated and analysed for security conformance by third party information security experts from the industry to identify and recommend solutions to potential security risks before the start of development phase of the software project.

SIGNIFICANCE OF SOURCE CODE EMBEDDED (SCEM) SECURITY FRAMEWORK

The most significant aspects of the SCEM security framework are:

- integration of security in software development which is embedded in current SDLC,
- formulating updates for existing software development tools to enable security by design,
- development of new security by design validation tools to help security consultants with detailed information on what areas of documentation pose potential security threats,
- a framework approach towards holistic integration of security in software development industry

In terms of secure software development information security and risk assessment industry is facing the following problems:

1. early detection of security flaws in a software before development completes is not possible;

2. normally, the software developer is considered the security owner;
3. there are no security compliance standards/frameworks which enforce organizational responsibility assignments, this is understandable as every organization is unique in its operations and treats their information differently;
4. security features of the final software are difficult or sometime impossible to map to the requirements and design document;
5. risk mitigation is performed at a very later stage of SDLC, where the cost of mitigation becomes too expensive and time consuming, and sometimes rendering the whole project unfeasible to be implemented due to potential risks.

The SCEM security framework will enable detection of security vulnerabilities before they become too costly. There are several security solutions available such as SQUARE (Security Quality Requirements Engineering) Methodology by SEI (Software Engineering Institute), CLASP (Comprehensive, Lightweight Application Security Process) by OWASP (The Open Web Application Security Project) and Microsoft's SDL (Security Development Lifecycle). These security approaches lack a holistic approach of "security by design" this mainly due to the reason that people postulating the theories are either purely software developers or purely security consultants which results in bigger gaps in understanding the core of the problem hence incomplete or impractical solutions are defined. Let's take an example of current Software Security Assurance concept, the problem with this postulated theory is that:

- it is incomplete as it doesn't cover the life cycle itself;
- doesn't bridge the "language" gap between security and software consultants;
- doesn't provide tools and practical solutions in line with specific software analysis and design methodologies (SADMs);
- relies heavily on biased input from either side of security or software consultants at different stages;
- doesn't concentrate on the CIA triad.

Considering the gap, and the non-existence of strategies, between the implementation of the CIA triad and analysis & design tools the opportunity to define a framework as a solution to the aforementioned problems which is aimed at achieving the following objectives is created:

1. identify existing software development methodologies;
2. identify tools (specification methods, diagrammatic modelling etc.) utilized in different software development methodologies;
3. develop the framework integrating the CIA triad with the identified tools enabling them to become auditable and carving in line with an organization's security requirements and processes;
4. refine the newly developed methods for industry usability by gathering:
    a. information security industry feedback and statistics
    b. software development industry feedback and statistics
    c. the software owner (the client) feedback and statistics
    d. the security owner (can be same as the software owner) feedback and statistics
5. research and development methods for how the responsibility of security ownership for each business process/requirement can be integrated into the design of the software.

The SCEM security framework introduces feasibility analysis models where the cost of achieving the CIA objectives would be added into the calculations; this ensures that the implementation of projects does not become unsuccessful due to unforeseen and unbudgeted security assurance costs. The new model will also ensure that the planning phase encompasses the security features required as part of the process and help in preventing implementation of over-kills in terms of security.

In the analysis phase the business requirements are identified and written from the client's perspective. It is unfortunate from the security perspective that normally the business requirements don't include any security requirements for that specific process. And even if they are included they fail to relate themselves to the CIA objectives of information security individually. It is important that the security features are distinctively specified for each CIA objective and not merged together in broader objective otherwise it would become impossible to implement that objective in the software due to modularity and responsibility requirements for a software project. The SCEM security framework ensures all of the above by introducing textual specification, metaphorical and diagrammatic modelling into the chosen SADM style.

SCEM SECURITY – THE CONCEPTUAL FRAMEWORK

Security in SDLC is achieved by revamping current SDLC phases and attaching the security processes in alignment with each SDLC phase. A conceptual framework of embedding security in SDLC is illustrated in Figure 1.

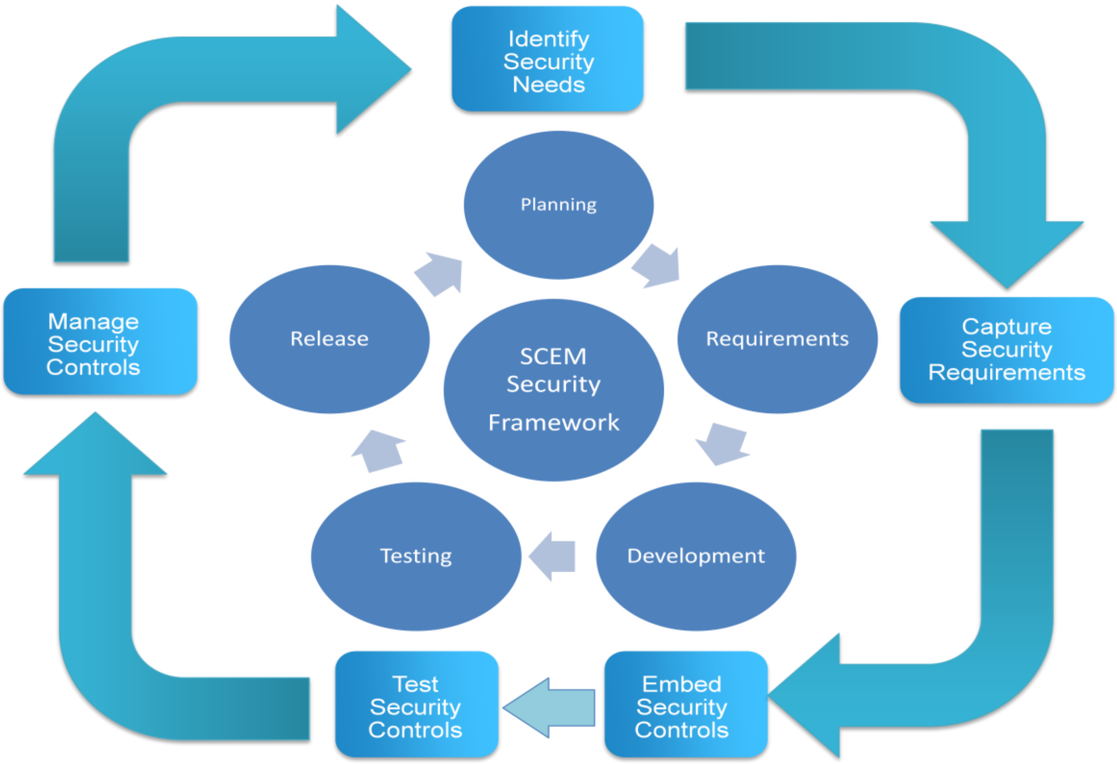

Figure 1: Source Code EMbedded (SCEM) Security Framework

The Planning Phase

The planning phase is the most important phase in SDLC where the objectives and expectations of a project are estimated. This phase sometimes also dictates the monetary benefits or limitations of a project which then become very critical when un-accounted for security requirements (due to compliance, eventual attacks, threats etc.) becomes mandatory to be implemented. In planning phase, preliminary investigation must include the survey and questionnaires to establish the security needs of the organisation.

The Requirements Phase

The requirements phase ensures that security requirements are captured during the requirement analysis. This phase actually seamlessly maps the business requirements specification from business perspective to a more refined technical perspective, which means that the SCEM security framework would not only have to provide the necessary mapping guidelines but also the right tools so that information security classification, risks and mitigation could be factored into the technical design seamlessly as well and that the technical resources are able to appreciate and understand them as well. Due to the availability of design methodologies in various flavours (e.g. Structured (Ashworth and Slater 1991), Object Oriented (Rumbaugh 1991) etc.) the security framework would provide tools for each major methodology and guidelines which would enable any new methodology to adopt the SCEM security framework easily.

The Development Phase

Before this phase can be explained it is essential that the role of a software developer be understood. A developer is actually responsible for "coding" the design into a computer understandable format. It should not be assumed that the coder or developer is responsible for ensuring the software security as well, unfortunately in the software industry whenever a security issue materializes the first and the last person to get the blame is the developer. It should be understood that a coder has only converted or coded a design provided to him/her into machine understandable format, any security flaws in the software have actually inherited from the design provided to the coder. Assuming that the coder should have written a secured application and should have thought of the security requirements is illogical. This can be understood by a very simple fact of software engineering, the software engineering methods were introduced to bridge the gap between technical understanding of the system and business understanding of the system at the first place. In line with this view, even if the developer is aware of how to implement encryption and authorization technically, he/she should not be expected to understand and appreciate the high level business security requirements.

Therefore, it is important that such strategies are added into existing development methodologies that ensure the business security requirements are clearly explained to the developer in a way which:

1. is cost & time effective for the business (the client) and industry (the service provider);
2. does not demand deep research into the various options available to implement security;
3. does not hold the developer solely responsible for evaluating and ensuring the controls effectiveness.

The SCEM security framework would make a developer's life easier by providing them with:

1. specific controls to be implemented
2. control configuration
3. crisply identified pieces of information that needs to be controlled
4. information classification configuration

The Testing Phase

Current methods of ensuring software security through testing process are flawed. A vulnerability scan or a penetration test into the application performed to evaluate the security only covers a purely technical aspect of information security. If software passes these two tests it is only technically secured but is the software actually secured in line with the business security or "reality of life" requirements. It is only the due diligence process which can potentially expose the software for security flaws even though it has passed certain industry recognized security best practices certifications.

During the due diligence process the software is looked upon as a weakness or a potential threat to the information system in terms of security. A proper implementation of SCEM security framework would change the situation where the software would be looked upon as the strength, a deterrent, or a security control within itself by simply enabling the testing of the software in line with the business policy or best practices standards. This would be easily achievable as, if the SCEM security framework is properly followed; each technical control is now mappable to a business function/requirement individually characterized into Confidentiality, Integrity and Availability objectives of information security.

The SCEM security framework would bring about new testing methods and check lists to ensure that the software is:

1. In line with the security requirements as laid down in the requirement documentation
2. In line with the actual security requirements of a business or best practices

The Release Phase

Security in release management is probably not easy to conceive, we will need to take help from the essential security models e.g. The Bell LaPadula security model (Bell and LaPadula 1973), which dictates the idea of "need to have" basis of acquiring access to any piece of information. Depending on how security has been implemented into the software, if the software implements security by responsibility and associate them with

different versions/editions of the software, there is a great need that the release management ensures that only the required edition/version goes to the “right” users.
The SCEM security framework will encourage and provide processes to ensure that the software releases are managed in line with the security model.

DISCUSSION AND ANALYSIS

The security framework approach provides the flexibility of applying some or all security controls. This flexibility can be leveraged and applied as an add-on tool on existing security measures to achieve better security and agility. The SCEM security framework is also a relief in solving the software documentation audit problem and be able to suggest fixes to any issues identified at the documentation stage. Another great advantage the way the SCEM security framework could provide is it’s add-on like behaviour where a certain feature of the framework could be embedded into an existing SDLC documentation to make it compliant to the SCEM security framework, there eliminates the need of re-writing the processes and specifications just adding new attributes, symbols, classification would achieve the objective hence saving huge costs that might incur on re-doing the existing documentation from the scratch.

The SCEM security framework enforces “security by design” into software projects and conforms to CIA characteristics. The implementation of SCEM security framework also facilitates guidelines, tools and methods to practically achieve the security objectives.

The cultural benefits include:

1. induction of information Security industry into software projects at earlier stages;
2. increased communication and understanding between the security consultants/auditors and software consultants/developers;
3. equal distribution of security responsibility among all involved in the project life cycle rather than holding developers responsible for security flaws;
4. removal of false sense of security.

The political/governance benefits include:

1. early identification of policy flaws which can be inherited by the software;
2. legislative scrutiny of software design possible for security flaws before the design goes to the implementation or release stage by e.g. parliament, third party auditors, internal auditors etc;
3. increase of public trust on the system;
4. global acceptance of the framework would enable easier and more assurances in terms of security.

The monetary benefits would include:

1. cost savings because early identification of flaws could be fixed on paper;
2. cost savings because development would not go beyond budgeted time because of unaccounted for security flaws;
3. cost savings on damage control as the software could then be seen as a security control and not a vulnerability;
4. cost savings where physical security audits could be reduced to paper audits.

FUTURE APPROACH

In future research we intend to conduct a survey and analysis of software development houses and software developers in a view to implement the SCEM security framework. The future research will include:

- Preliminary investigation/planning processes
  - Analysis of stakeholders involved such as users, management and technical team to determine the organisations security goals. Examine if the security requirements are collected at planning phase and security objectives are defined.
- Requirements gathering processes

    - Analysis of the logical framework, how the information form planning processes are converted into user requirements and how/if security is considered at this phase. Select the security control according to the severity of impacts to security triad. Identify if the security controls are placed in the logical model and include security logical diagrams in requirements analysis.
- Software design and development processes
    - Analysis if the security requirements gathered in planning and requirement processes are translated into application design processes. Perform a risk assessment to re-evaluate if all security threats have been identified and planned controls comply with security triad. By identifying threats and their potential impacts at this phase can help organisations determine the residual risk. This will serve as an effective litmus test before costly development takes place.
- Testing Processes
    - Besides all other system development tests, a security test will be conducted to analyse the codes for integer overflows, buffer overflows, format string errors, cross-site scripting, race conditions, SQL and command injections.
    - Some sample codes will be analysed using a combination of static tools and manual verifications to the SANS, ISACA, OWASP, and COBIT secure coding guidelines. Following data fields will be collected from the codes:
        - The total number of coding and debugging hours
        - Hours to patch security flaws before and after release
        - The number of bugs initially and overtime as security patches are deployed
- Release processes
    - Analysis of versions of software with and without security patches and cost involved in fixing the security patches in the final versions
- Security processes
    - Analysis of the existing security process (if any). Collection of data on penetration test to verify the validity of the adopted security processes.
    - Analysis of Message Digest (MD) value before and after release of the software

CONCLUSION

The paper has addressed one of the major issues, security in software development, faced by the industry. A security framework approach has been presented in the form of Source Code EMbedded (SCEM) security. The SCEM includes a holistic approach for embedding security right from early phases of the System Development Life Cycle (SDLC). The phased approach will enable security needs integrated in the requirements, security requirements captured, security controls embedded and tested during the SDLC and finally managed using the guidelines set by the industry. The implementation of SCEM is future task and we anticipate that the proposed security framework will set a scene to integrate ‘security by design’ in the software development industry.

ACKNOWLEDGEMENT

We thank Rizwan Rizvi of Westhill Consulting Pty Ltd for his input on the security issues faced by the software development industry.

REFERENCES

Ashworth, C., Slater, L. (1993) An introduction to SSADM. NY, USA. McGraw-Hill, Inc.

Banerjee, C, and Pandey, S. K. (2010) Research on Software Security Awareness: Problems and Prospects. ACM SIGSOFT Software Engineering Notes. Vol. 35, No. 5.

Banerjee, C, and Pandey, S. K. (2009) Software Security Rules: SDLC Perspective. International Journal of Computer Science and Information Security. Vol. 6, No. 1.

Bell, D. E and La Padula, L. J. (1973) Secure computer systems: Mathematical foundations. MITRE Technical Report 2547, Vol. 1.

CLASP Project. The Open Web Application Security (OWASP). Accessed online 2 November 2011 https://www.owasp.org/index.php/Category:OWASP_CLASP_Project

Kothanath, A. (2006) Building Security into your SDLC methodology - ISACA Monthly Meeting. Retrieved April 12, 2009 from http://www.go-integral.net/node/108

Mead, N., R., Hough, E. D., Stehney II, T. R. (2005) Security Quality Requirements Engineering (SQUARE) Methodology. Carnegie Mellon Software Engineering Institute.

Mellado, D., Fernandez-Medina, E., and Piattini, M. (2010) A Comparison of Software Design Security Metrics. Proceedings of the 4th European Conference on Software Architecture (ECSA 2010) August 23-26, Copenhagen, Denmark.

Pfleeger, C., & Pfleeger, S.L. (2007) Security in computing (4th ed). Upper Saddle River NJ: Prentice Hall

Roberts, T. L., Gibson, M.L., Fields, K.T., Rainer, R.K. (2001) Response to 'comments on factors that impact the implementation of a systems development methodology'. IEEE Transactions on Software Engineering. IEEE Press.

Royce, W. (1970) Managing the development of large software systems: concepts and techniques. In Technical Papers of Western Electronic Show and Convention (WesCon) August 25-28, 1970, LA, USA.

Rehman, S. And Mustafa, K. (2009) Research on Software Design Level Security Vulnerabilities. ACM SIGSOFT Software Engineering Notes. Vol. 34, No. 6.

Rumbaugh, J. (1991) Object-oriented modelling and design. ISBN 0-13-629841-9. Prentice Hall

SDL (Security Development Lifecycle). Microsoft. Accessed online 2 November 2011 http://www.microsoft.com/security/sdl/default.aspx

Shin, Y. And Williams, L. (2008) Is Complexity Really the Enemy of Software Security? Proceedings of the 4th ACM Workshop on Quality of Protection (QoP 2008) October 27, 2008. Alexandria, Virginia, USA.