Analysis for Security Implementation in SDLC

Gaurav Raj[1]
[1]PhD Research Scholar, PTU,
Jalandhar, Punjab, India
graj@amity.edu

Dr. Dheerendra Singh[2]
[2]HOD, SUSCET, Tangoori,
Punjab, India
professordsingh@gmail.com

Dr. Abhay Bansal[3]
[3] HOD, CSE Department,
ASET, Amity University
Noida, U. P, India
abansal1@amity.edu

Abstract- Software development is not only a single motive process as development but also have to handle number of different issues as like security. For basic software development, we generally use SDLC models for development like web projects, applications, services etc. but in this development process we are not taking care of security issues which are demand of these days. There is no central body who manage all the crucial tasks. The essence of this paper lies in study of security issues as well as importance of security in SDLC models. Moreover, we focused over review in recent development area as we get differentiate in traditional SDLC security issues with SDLC security issues in virtualized scenarios as cloud. Here, we discussed the role of team and their members in feasibility and planning phases for analysis. This study helps in access control, risk assessment and security monitoring. We studied about risk classification for better risk assessment in different phases.

Keywords: IaaS; SDLC; Cloud Computing; SDL

I. INTRODUCTION

SaaS is generally referred to as "On demand Software Service". It is a software delivery or distribution model in which applications, software and associated data are hosted using the service based network i.e. internet, usually called "cloud". Accessing of software and data are possible via web-2 enabled browser which need to be subscribed based on payment available payment plans. Payment may be weekly, monthly or yearly, as per payment plan selected by customer. Hence, it is different from traditional scenario which assumed ownership for maintenance, installation and buying license for software. SaaS is not only faster but cost effective too as there are no more hardware costs, acquisition costs to run , implementation costs etc. It is become more prevalent day-by-day for a number of business applications, like software development, Software Designing, software management, DBMS, accounting, CRM and many more.[4]

In our study we have analyse the roles/ responsibilities of developing team members and computing efforts in different phases of SDLC to find out the places to incorporate security implementation.[2] Security is the core requirement of all software vendors which includes need of protection of infrastructures as well as preservation of trust in computing. In order to achieve this, vendors need to adopt a stringent software development process that focuses on security so that security vulnerabilities in design, coding, documentation and other phases can be minimized, detected and removed as early as possible in the development life cycle.[3] And a team specialized for security purposes must always be there for frequent interactions during software design and development. The same team is meant to have a final security review, FSR, before the software is released.

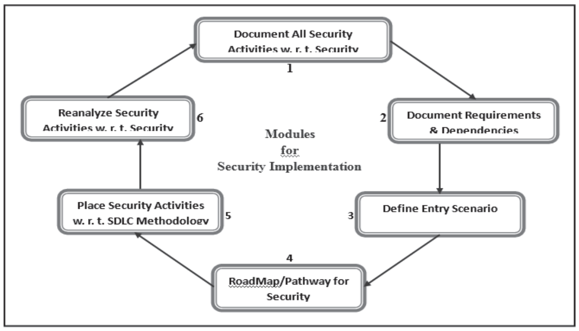
Document All Security Activities w. r. t. Security
1
Document Requirements & Dependencies
2
Define Entry Scenario
3
4
RoadMap/Pathway for Security
Place Security Activities w. r. t. SDLC Methodology
5
Reanalyze Security Activities w. r. t. Security
6
Modules for Security Implementation

Fig. 1 Modules for Security Implementation in SDLC

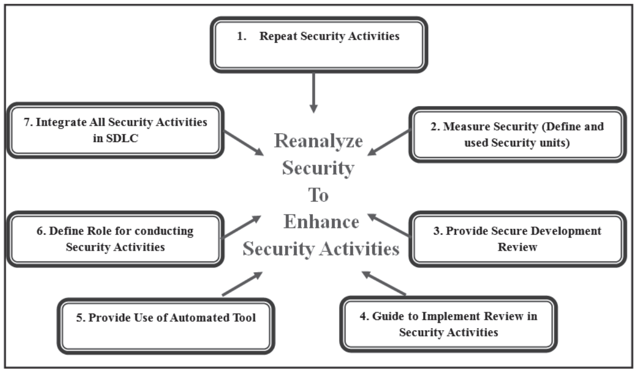
1. Repeat Security Activities
2. Measure Security (Define and used Security units)
3. Provide Secure Development Review
4. Guide to Implement Review in Security Activities
5. Provide Use of Automated Tool
6. Define Role for conducting Security Activities
7. Integrate All Security Activities in SDLC
Reanalyze Security To Enhance Security Activities

Figure 2: Reanalyse Security Mechanism to Enhance Security Activities

A. <u>Security Policy</u>

Policy partitions system states into:

1. *<u>Authorized (secure):</u>* these are states the system can enter
2. *<u>Unauthorized (insecure):</u>* if the system enters any of these states, it's a security violation secure system starts in authorized state and never enters unauthorized state.

B. <u>Threat modelling.</u>

The product team conducts threat modelling at a component-by-component level. Threat modelling refers to designing the model of the security aspects for possible set of attacks as per case.

II. SDLC ACTOR'S RESPONSIBILITY ANALYSIS

978-1-4799-4236-7/14/$31.00 ©2014 IEEE

TABLE I. ACTORS AND RESPONSIBILITIES

Feasibility Phase	
Designation:	**Feasibility Report Manager**
Sub-Tasks:	Prepare and Submit Feasibility Report
Work:	Document Findings in a formal report to steering committee
Methods/Tools Used:	Feasibility Report
Designation:	**Research Solutions Engineer**
Sub-Tasks:	Select Feasibility Options
Work:	Possible solutions to above defined problem, usually about six, and short list two or three by user
Methods/Tools Used:	Business System Options, Technical System Options
Designation:	**Onsite Coordinator**
Sub-Tasks:	Definition Of Problem
Work:	Detailed survey of the requirements (expressed as well as unexpressed) of the client, formal statement of problem
Methods/Tools Used:	Dialogue Identification, Data Flow Model, Data Flow Model
Designation:	**Economical Analyst**
Sub-Tasks:	Economical, Schedule, Resource, Legal Feasibility
Work:	Determine positive economic benefits, whether the project can be completed in given time, whether proposed system conflicts with legal requirements
Methods/Tools Used:	
Designation:	**Technical Analyst**
Sub-Tasks:	Technological, System, Operational, Technical Feasibility
Work:	Determine whether company has technical expertise staff as per need, system requirements, determine present technical resources
Methods/Tools Used:	
Designation:	**Problem Analyst**
Sub-Tasks:	Project Description and Definition of the Study Boundary , Analyze Existing Systems
Work:	Review project description, confirm objectives, Sign agreement to acquire all possibly required detail, determine whether the project follows DAS requirements for cost benefit analysis. Identify sources of data and business areas to be studied in its context along with data subjects relevant to that area, identify critical business issues, go through pre existing relevant data, define constraints like delivery time, budget restrictions, technological limitations etc. Identify all possible Users, Sponsors, Managers, Developers, participants from other related areas. Evaluate pre-existing systems in order to determine weaknesses and strengths, determine what existing system development affects the present one.
Methods/Tools Used:	Entity Relation Diagram or Logical Data Structure, Requirement Definition, Data Dictionaries, context Diagrams, System Matrix, Business Function, Existing System representation by DFDs, ERDs etc. - Statement of impact of existing systems
Planning Phase	
Designation:	System Solution Provider
Sub-Tasks:	System Solution
Work:	Determine High level design alternatives, Cost benefit estimates for each alternative, Identify and select appropriate alternative
Methods/Tools Used:	Cost Benefit Analysis, Recommendations
Designation:	Development Planner
Sub-Tasks:	Development Phases
Work:	Development Approach, Type of development
Methods/Tools Used:	
Designation:	Project Planner
Sub-Tasks:	Develop Project Plans and creating Schedule Plan
Work:	A collection of plans for each module, then rolled up as master plan, Planning activities of each individual along with mile-stone drive plans scheduling
Methods/Tools Used:	
Designation:	Resource Planner
Sub-Tasks:	Resource Plan

Work:	Types of labor required for the project, Roles and key responsibilities for each labor type, Number of people required to fill each role, Equipments to be used and their purposes, Types and quantities of equipments needed, Total amount of materials needed
Methods/Tools Used:	Resource utilization schedule
Designation:	Financial Planner
Sub-Tasks:	Financial Plan
Work:	Types of labor costs to be incurred during the project, Items of equipment needed to deliver the project, Various materials needed by the project, Unit costs for labor, equipment and materials, Other costs types such as administration
Methods/Tools Used:	Cost Benefit Analysis
Designation:	Risk Planner
Sub-Tasks:	Risk Plan
Work:	Identify risks within your project, Categorize and prioritize each risk, Determine the likelihood of the risks occurring, Identify the impact on the project if risk does occur
Methods/Tools Used:	Risk monitoring methods
Designation:	Acceptance Planner
Sub-Tasks:	Acceptance Plan
Work:	Creating a full list of all project deliverables, Listing the criteria for gaining customer acceptance, Putting in place, acceptance standards to be met
Methods/Tools Used:	Acceptance test methods
Designation:	Procurement Planner
Sub-Tasks:	Procurement Plan
Work:	Define procurement requirements, Identify all of the items you need to procure, Create a sound financial justification for procuring them
Methods/Tools Used:	Project procurement process
Designation:	Coder
Sub-Tasks:	Setting of the Development and Test Environment
Work:	These environments are for developing and testing of solutions chosen in previous phase
Methods/Tools Used:	
Designation:	Planning Head
Sub-Tasks:	Project Plan Approval
Work:	Documentation of the result of this phase that is, final Project Plan

TABLE II. SECURITY IMPLEMENTATION IN FEASIBILITY PHASE

Security Actors	Security Types		
	Information Security	**Architectural Security**	**Network Security**
Feasibility Phase Actors	*Administrator*		
	Research Solution Engineer *Economical Analyst* *Technical Analyst* *Problem Analyst*	*Feasibility Report Manager* *Research Solutions Engineer* *Onsite Coordinator*	*Client* *Onsite Coordinator*

III. STEPS FOR IMPLEMENTING SECURITY IN SDLC

A. Requirements Phase

A member from software security team/group accompanies rest of the members of the phase to make recommendations related to security and specially advises as follows

1. *Security milestones*
2. *Exit criteria*

These advices are based on project size, risks, and other factors. The member may be called as "Security Advisor". The member monitors the security element so that no problem is faced on later phases. This phase is the base for how security is integrated in upcoming phases and identification of key security objectives.

B. Design Phase

This phase defines overall structure of the software. Therefore, definition of security architecture and design guidelines is included in the phase also. Along with this, identification of the essential components, whose security is of utmost importance, is sthere. Design principles introduced include:

1. *Least Privilege:* a subject should be given only those privileges necessary to complete its task. The control should be according to function and not the identity. The rights shall be added as needed and discarded later on.
2. *Fail-Safe Defaults:* the default action is to deny access and if the action fails, system shall be as secure as when action began.
3. *Economy of Mechanism:* Keep it as simple as possible, that is, KISS Principle shall be applied where simpler

means less can go wrong and when errors occur, they are easier to understand and fix.

4. *Complete Mediation:* there must be monitoring of every access, usually it is done once, on first action, and not monitored thereafter, but if permissions change after, one may get unauthorized access.
5. *Open Design:* Security should not depend on secrecy of design or implementation, but this shall not be misunderstood to mean that source code should be public, rather it is “Security through obscurity”, but does not apply to information such as passwords or cryptographic keys etc.
6. *Separation of Privilege:* It requires multiple conditions to grant privilege, like separation of duty, defence in depth.
7. *Least Common Mechanism:* The mechanisms should not be shared although information can flow along shared channels. Isolation using virtual machines and sandboxes can be achieved.
8. *Psychological Acceptability:* Security mechanisms should not add to difficulty of accessing resource. There must be ease of installation, configuration, use etc.

C. Development / Implementation Phase

A number of steps are taken to control security flaws in this phase so that the final version release for customers is as better as possible. Here, special attention of developers is needed as incorrect code mitigates high priority threats. Therefore, applying coding and testing standards help avoiding the security flaws. Also, application of security tools like, “Fuzzing” which supplies structured but invalid inputs is used to detect errors. Static analysis tools also detect flaws like buffer overrun, integer overrun etc. Code reviews include manual as well as automated reviewing of code. Where automated include error detection tools, manual imply trained developers to monitor the correctness of code.

D. Testing/Verification Phase

It implies functional completion of software. A security push is introduced in this phase along with beta testing because project has come nearby to its close. This is also to confirm whether software has met the requirements. It also includes testing of high priority code, which is more prone to attacks.

E. Deployment & Maintenance Phase

At this phase, the software must be secure enough so that it is ready to be delivered to the customers. A Final Security Review is conducted in this phase. In FSR, software’s ability to withstand the vulnerabilities is monitored, along with penetration testing. The result is an overall picture of security posture of the software.

IV. Security in SDLC over cloud

The concept of software development lifecycle for a project varies according to many factors where type, size and time being the major ones. Combined with cloud computing, SDLC is more prone to risks. To prove this concept of merging SDLC with cloud, in this paper we analyse the feasibility study phase and find out security and risk issues in cloud based SDLC.

We are designing a process for secure web application development. This secure process can be defined as the set of activities performed to design, developing, testing, configuring, maintaining, and delivering a secure solution. Activities may not necessarily be sequential; they could be concurrent or iterative. We are analysing the need for security in every activity module of Web Engineering as follows:

1. *Formulation of the problem(Problem Analysis),*
2. *Planning module (Feasibility Study),*
3. *Web Application requirements analysis module (Requirement Analysis),*
4. *Architectural, navigational, and interface design module (Design module),*
5. *In system implementation module (Development module),*
6. *In application unit and integration testing and configuration management (Testing Module)*
7. *In quality control, and maintenance mechanisms (Maintenance Module).*

V. Secure Web Application Development Life Cycle(SWADLC)

After the discussion and analysis over security issues, we have planned to upgrade SDLC for implementing security in different virtualized environments like cloud computing service development scenarios. We have proposed upgraded SDLC in terms of following key points as per intermediate phases -

In requirement analysis and feasibility study

It requires to be sure about security of virtualized resources as we are using all web resources for application development in cloud computing which are not physically handled by developer.

In design and development

We are processing task over virtualized resources via internet so we have to configure secure data communication over virtualized network and need to maintain secure connection between developer’s machine and virtual machine.

In testing and configuration management

We have to be careful about individual testing and group testing of developed modules. In this module, we have two different ways of testing

1) Developer side 2) Client side.

If we are performing testing in developer side it can be test on

1) Developer’s machine as local
2) Virtual machine in cloud environment as global.

I. If we are testing it on developer's machine then it will be safer compare to virtualized machine.

II. If we are testing on client side then testing on virtualized machine will be safer then client side testing.

From the above two statement we can say security in testing module is required lots of attention at the time of client side testing because we have to deal with two security modules one in between cloud and developer while other in between cloud and client. Both of these security modules are very different in implementation.
In this module we also have to take care of deployment and configuration process under cloud service provider's infrastructure.

In maintenance and Feedback module

Our major objective is to take feedback from clients and implement it to improve QoS, we also have to be careful for maintenance process for all clients request coming after use of the developed web application. Task of this module is to provide feedback to every module of lifecycle for improvement of application quality. That's why, this module deals with every above module and improve application. Again it has to be careful about security issues because lots of intruders try at this point for disturbing module design and codes by providing wrong feedbacks.

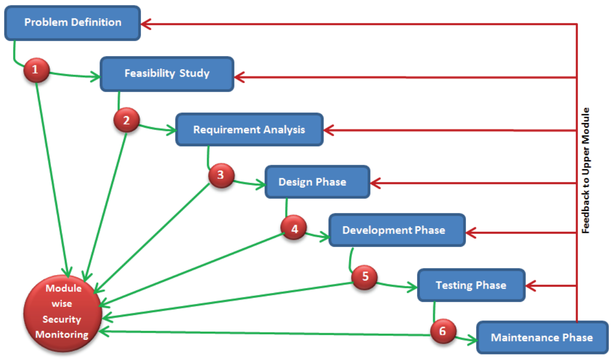

Figure 3: Integration of Security Monitoring in Software Development Lifecycle

A. Security Monitoring in SWADLC

As we deals with security in application development lifecycle (as a process), we have to be focused on each and every module independently because every module security requirement is different from each other. We have to define some security monitoring points according to lifecycle models. As we are working over SDLC, security monitoring points are applied in between modules of SDLC to check security of previous module output and verify security over input for next module. Through a review and analysis of existing process models and standards, activities for secure application development have been categorized as follows-

B. Common Module Security Activities(CMSA)

Here we have classified activities which are common for all modules
Engineering Activities (EA): Engineering activities include activities needed to engineer a secure solution as security requirements elicitation and definition for infrastructure feasibility.
Assurance Activities (AA): Assurance activities include verification, validation as Problem validation, expert reviews to make an informed decision over feasibility of developing application services into different cloud infrastructures.
Management Activities(MA): These are further classified as follows-

i. *Organizational Activities (MA-OA)*

Organizational activities are taking care of organizational policies, organizational roles establishment, and other organizational activities that support web security in virtualized environment. Project management activities include project planning and tracking secure resource allocation process in different cloud infrastructure and usage of these virtualized resources to ensure that the security engineering, security assurance, and risk identification activities are planned, managed, and tracked.

ii. *Risk Identification Activities(MA-RIA)*

Identifying and managing security risks are the most important activities in a secure application development on a cloud environment because of new risk of attacks are increasing with a rapid growth. It is the driver for subsequent activities as security engineering activities, the project management activities, and the security assurance activities.

C. Module Dependent Security Activities (MDSA)

Security as dependent over individual module requirements is known as Module Dependent Security Activities (MDSA), which can be applied in between modules to handle input and output security monitoring as requirements. As we say if our data or information is in our secured virtualised System, it will be assumed as approximately secure. But as it moves out and travel in between systems and handed over to next phase it may be infected by any outsider attack, virus, etc. So for making our communication safe and secure, we have to apply for security monitoring in between development phases.

TABLE III. RISK ASSESSMENT SECURE SDLC PHASES AND ACTIVITIES

SDLC Phases	**Secure Software Practices**	**S-SDLC Activities**	**Risk Assessment**	**On Going other Disciplines**
Requirements	Preliminary risk Analysis	Define USE and ABUSE Cases	High Level Risk Assessment	Risk Management,
	Security Requirements	Define Security Requirements		
Design	Design	Secure Architecture, Threats Modeling	Technical Risk Assessment	
Development/ Implement	Secure code Implementation	Static and Dynamic Code Review		Defect Management,
Testing	Security Test	Functional, System, Risk Driven, Test, White Box, Black Box Testing	Functional Risk Assessment	
Deployment & Maintenance	Secure Configuration	Secure Configuration	Infrastructural Risk Assessment	Vulnerability Management
	Secure Deployment	Secure Deployment		

TABLE IV. MODULE DEPENDENT ACTIVITIES

Module Dependent Activities		
Management	**Planning & Documentation**	**Development & Security**
Program/Project Management	Release Planning	Development & Deployment
Configuration Management	IT Strategic Planning	Document Governance
Capacity Management	Requirements Definition	System Security & Administration
Asset Management	Detail Design	Protection of Proprietary Information (POPI)
Site Monitoring and Problem Management	Procurement and Purchasing	Quality Function & Control Records
MIS Reporting	Training & Documentation	
Incident Tracking	Service Level Agreement	

VI. CONCLUSION AND FUTURE WORK

Subsequent to analysis about security and risk related issues in SDLC in cloud environment, we've now looked at implementation part for Secure Web Application Development Life Cycle in public and private clouds. We are planning to use the module dependent activities to monitor the system level security. We have to find, How to protect and insert deliverables (information) in the life cycle that are designed to support decision management as "Move / not Move" decisions to jump from one phase to the next phase. In Future, we are looking for implementation of the outcomes from this review paper for generalizing the life cycle framework and address other aspects of building a strong security presence within established organizational processes in different cloud's deployment models.

REFERENCES

[1] Preston Pierce, "Software Verification & Validation" *RELA*, Inc., 6 (PRESTON)175 Longbow Drive Boulder, *CO* 80301 *3031 530-2626*

[2] Muhammad Umair Ahmed Khan and Mohammad Zulkernine, "Quantifying Security in Secure Software Development Phases" ,*School of Computing, Queen's University ,Kingston,* Ontario, Canada, K7L3N6

[3] R. Kumar, S. K. Pandey, S. I. Ahson, "Security in Coding Phase of SDLC" Department of Computer Science Jamia Millia Islamia, New Delhi- 110025, INDIA

[4] G. McGraw, "Building Secure Software: A Difficult But Critical Step in Protecting Your Business," *Cigital*, White Paper, available at: http://www.cigital.com/whitepapers/.

[5] G. McGraw, "Software Risk Management for Security," *IEEE Computer*, 32(4), April 1999 p. 103-105.

[6] M. Bishop, "Computer Security: Art and Science", *Addison-Wesley-Longman*, Nov. 2002.

[7] A. Jaquith, "The Security of Applications: Not All Are Created Equal," *Research Report,* @Stake, February 2002, p. 1-12, Internet Article: http://www.atstake.com/research

[8] K. Hoo, A. Saudbury and A. Jaquith, "Tangible ROI through Secure Software Engineering," *Secure Business Quarterly*, Q4, 2001

[9] M., Howard and D. LeBlanc, "Writing Secure Code", *Microsoft Press*, Redmond, WA, 2002

[10] L. M. Vaquero, L. Rodero-Merino, J. Caceres and M. Lindner, "A break in the clouds: towards a cloud definition", SIGCOMM Computer Communication Review, vol.39, pp. 50-55, December2008.

[11] M. Armbrust, A. Fox, and et al., "Above the clouds: A berkeley view of cloud computing", UC Berkeley, Tech. Rep. UCB/EECS-2009-28, February2009.

[12] K.Birman, G.Chockler, and R. van Renesse, "Toward a cloud computing research agenda", SIGACT News, vol. 40, no.2, pp. 68-80, 2009.

[13] M.Armbrust, A.Fox, R.Griffit, etal., "A view of cloud computing", Communication of the ACM, vol. 53, no. 4, pp. 50-58, 2010

[14] Sheheryar Malik, Fabrice Huet, "Adaptive Fault Tolerance in Real Time Cloud Computing," services, pp.280-287, 2011 IEEE World Congress on Services, 2011

[15] Raj Gaurav, Munish Katoch, "Security Implementation through PCRE Signature over Cloud Network", Advanced Computing: An International journal, May 2012, Vol. 3 No. 3 ISSN: 2229 - 6727[Online]; 2229 - 726X [Print],pg no. 119-127.

[16] Raj Gaurav, Nitika, shaveta,"Comparative Analysis of Load Balancing Algorithms in Cloud Computing", International Journal of Advanced Research in Computer Engineering & Technology, Vol. 1 No. 3 (2012)(ISSN:2278-1323), pg no. 120 -124.

[17] Raj Gaurav, Ankit Nischal, "Efficient Resource Allocation in Resource Provisioning Policies Over Resource Cloud Communication Paradigm", International Journal on Cloud Computing: Services and Architecture, June 2012, Vol. 2, No. 3, ISSN: 2231 - 5853[Online]; 2231 - 6663 [Print], pg no. 11 - 18.

[18] Raj Gaurav, Kamaljeet Kaur, "Secure Cloud Communication for Effective Cost Management System Through MSBE", International Journal on Cloud Computing: Services and Architecture, June 2012, Vol. 2, No. 3, ISSN: 2231 - 5853[Online]; 2231 - 6663 [Print], pg. no. 19 - 30.