ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 7, July 2015

Incorporating Security into SDLC Phases Using Security Analysis

Prof.Y.C.Kulkarni , Prof. Dr. S.D.Joshi

Department of Information Technology, Bharati Vidyapeeth Deemed University College of Engineering, Pune, India

Professor, Department of Computer Engineering, Bharati Vidyapeeth Deemed University College of Engineering, Pune, India

ABSTRACT: Now days the software Development is the most powerful area in the field of "Information Technology". Most of the people choose Waterfall model, Spiral model & agile model for software development. If we will use Waterfall model & if we will not secured the phases of Waterfall model then it will leads to breach in software development. So to avoid this & to secure different phases of SDLC the paper is written into which the methodologies of Secure SDLC are studied by creating software requirements, Designing software securely, implementing secure software, assuring software quality and deploying software with security.

KEYWORDS: Software, Security, SDLC

I. INTRODUCTION

Over the past decades, the maximum software industries made significant investments in "Information Security". These investments were designed to increase security of network, access controls. That means there was a lack of attention in maintaining security of software design and that's why Software Development Lifecycle comes into play into which the industries made significant investment to maintain the security of SDLC[1].

Normally developers are choosing different models for different software projects but consistency in result should be required otherwise, developers and organization have to face the difficult problems as they switch from one project to another project. Of course if we are choosing a model like SDLC with the phases like:

1. Requirement
2. Design
3. Coding
4. Testing/Quality Assurance
5. Deployment/Delivery of Software

Whenever developers are going to develop a software using SDLC so we should be take care about the security of each and every phase of SDLC otherwise it will reach to a big problem which is harmful to us and organization.[1]
Secure Software Development requires a mutual commitment between application developers & information security team members. Both the parties that is application developer & information security team members must be committed to the integration of security from the beginning & should be agreed with security concepts will be addressed at each stage of SDLC. So the partnership of software developer with information security team members will be useful to increase the effectiveness & efficiency [1]. The purpose of building a culture of security is nothing but developers should be understood an importance of security in such a way that each and every developer will aware with security concerns and to achieve this, security awareness training should be conducted within an organization. [1]

Security professionals should consider a program consisting of three major components:

1. **Initial Training or Briefing:** This training includes the education about how to apply common security & how to secure code. This training should be compulsory for all developers and should be conducted after joining of them into development team. [1]

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 7, July 2015

2. **Refresher Training:** This training should be conducted at least once in a year to brush-up the knowledge about the security issues or to brush-up an importance of security. [1]
3. **Ongoing awareness programs**: To conduct this program we can use a mixture of different mediums such as newsletters, posters and other tools which is feasible to fit within an organization's culture. This training also conducted to remind an importance of security coding practices. [1]

II. CREATING SOFTWARE REQUIREMENT

The solution to get a success to incorporate security concerns in requirements can consist security experts to specify the minimum requirements related with security by conducting a risk assessment and defining clear security thresholds. [1]

Authentication Requirements: We can specify how users prove their identity to the application through the authentication requirements. Username and password may be sufficient for low-security applications. Higher security applications may need the multifactor authentication which becomes more common for web applications. For example: Google authenticator system that allows users to retrieve a one-time password from an application which is installed on their phone and provide this to the authentication system as proof of having physical possession of the device. [1]

Session Management Requirements: These requirements specifies for how much period user remains authenticated and how? The randomized token value which is placed in cookie header is used to maintain session state with server. Session management requirement may also include idle and absolute timeout periods that automatically disconnect unused sessions after a specified period of time. [1]

Logging and Auditing Requirements: The user and administrator should log in an application is the good security practice. The minimum security requirement is nothing but an application should specify the types of events or actions that the system should have capacity of logging procedure and administrators should have degree of control over logging procedure/process. The events which are related to authentication, session management, changes in application data, errors found by users are included in logging and auditing requirements. To make easy log analysis, clocks should be synchronized across the servers and all systems should use a consistent log format. [1]

Protection of sensitive information: The organization's sensitive information from unauthorized access can be protect using confidentiality controls which includes the use of strong access control system as well as encryption, key management and hashing. The minimum security requirement is the different activities should be supported by access control system as well as to alter user authorizations should be supported. The requirements should specify the cases where encryption must be used as well as encryption algorithms and key lengths are acceptable. The web applications are depends on the use of Secure Socket Layer (SSL) and/or Transport Layer Security (TLS) to encrypt network communications. Early versions of SSL that is SSL v1 & SSL v2 should not be used to secure web applications. [1]

Integrity Controls: All data modifications or manipulations are considered as part of an authorized action. This may conducted through an automated activity such as a daily job that updates inventory levels or through an action of authorized user. The minimum security requirement should specify the types of integrity controls supported by applications access control system. [1]

Availability Controls: If the system is available to an only authorized user then that event is called as availability controls which protects the system. The minimum security requirement should specify the appropriate availability level for an application. To achieve these requirements the joint effort among the developers, operations staff and security professionals. [1]

Application Controls: The series of application controls that protect against common threats should be implemented by developers. The input validation should be performed to ensure that the data is in a consistent and expected format. The output generated from an unauthorized input source should be sent through centralized encoding routines. [1]

System Configuration Requirement: The systems used to store, process and transform information as a part of application must be securely configured. This process includes validation of system configuration settings, maintaining an appropriate patch level and monitoring the system as a part of a centralized monitoring scheme. [1]

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 7, July 2015

Compliance Requirement: Applications must conduct regulated activities such as processing credit card transactions, maintaining health records. The minimum security requirement is an application should reflect the activities necessary for compliance. [1]

The risk assessment should be performed by qualified development and security staff. Developers who collected the requirements or who will implement an application should not be responsible for risk assessment procedure. Many organizations use a Formal Security Architecture Review Board to conduct risk assessment which one of the early activity of any development project. The risk assessment formula is risk = Likelihood (Probability) * Impact. [1]

Defining Security Thresholds: In addition to the specific security controls the requirement phase should include the definition of acceptable security thresholds for an application. These thresholds should specify the severity of vulnerability that may be exist in application before it is deployed. Scanning tools can be used to identify a large number of vulnerabilities and assign them different criticality levels. By default these are normally configured in security testing tool but can be modified as per needs of different organizations. For example : Open Web Application Security Project (OWASP) scoring system uses three criticality levels such as High, Medium and low. [1]

.

III. DESIGNING SOFTWARE SECURELY

Designing Software: Software engineering states that the significant amount of effort should be spend on design phase before writing of source code and if we will follow this concept which helps to developer to understand the various components of a software project and how to integrate these components. This understanding is essential to create durable code. The specific components of software project and the nature of design deployable will vary depending on methodology. In agile methodology, they will include user stories. Waterfall approaches will include data flow diagrams, pseudo code and similar artefacts. [1]

Incorporating security into the design: Both software engineers and security professionals can save a significant amount of time by incorporating security into design. It's very easy to modify the design documents to include security objectives than to manipulate it after implementation of code. The practice of retrofitting software to meet security requirement is known as bolt-on security and is tested by both software developers and security peoples. Retrofitting often consumes more resources than resources have been spent on designing the features from start and usually results in substandard security functionality. The minimum security requirements should be identified earlier in SDLC process which includes proper design and documentation of the output prior to beginning the coding process. For example: If one of the project's security requirement indicates that the application must support two-factor authentication then design process needs to identify the authentication technique supported by an application. The design should include the description about the way that the users will interact with the authentication mechanism and how those mechanisms will securely integrate with rest of application to prevent unauthorized access. The minimum security requirement is the communication between web server and database server should be encrypted. [1]

Developing Threat Model: Threat modelling is one of the process into which the project team members should be gathered to review the software's functions and possible threats which are faced by software. The purpose of threat modelling is to bring the project team members together for a thoughtful discussion and analysis of threats that exist in an application environment. Threat modelling is one of the security analysis method used to identify risks.This method is used in earliest phases using dataflow and activity diagrams, specifications, architectural view and so on. But it can also be used with detailed design documentation and code. The main goal is to identify those threats which will results into maximum damage. Threat modelling tells us the process of identifying what functionality and which assets an attacker take advantage. The software design should be evaluated from an attacker's point of view. This process will result in a threat model document that can be used by developers to identify which threats are present and which steps should be followed to mitigate the risks. SWIDERSKI and SNYDER list the following purposes of threat modelling:

1. Understand the threat profile a system.
2. Provide a basis for secure design and implementation.
3. Discover vulnerabilities.
4. Provide feedback for an application security life cycle.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 7, July 2015

Threat modelling is not only connected to the design phase but also considered as an important part of the requirement phase. The artefacts in threat modelling are attack trees, threat trees, misuse cases, fishbone diagram. [2]

Minimizing the Attack Surface: While developers do not have any control on threats which are faced by an application in their production then developers do have ability to minimize the vulnerabilities. Some of the common ways which are used to minimize the attack surface are explained as follows:

1. Do not trust on user defined data.
2. When making access-control decisions never use user-defined data exclusively.
3. The amount of data passed between the browser and server should be minimum.
4. Keep data and state on server side and don't depend on hidden fields or cookies to store sensitive information.
5. Third party libraries that you link in your code should be limited.
6. All sensitive functionality should be present on server side. [1]
7.

Performing a Security Architecture review: Security Architecture Review should be performed to secure a design phase which includes input from major stakeholders on both development and security teams. In some organizations, the team who have created threat model can conduct security architecture review. Other organizations may include additional team members such as management. The design phase is the most important phase from security point of view and by adequately addressing security requirements at this stage, developers can avoid the messy (untidy) rework of bolt-on security initiatives and ensure that they create a secure sustainable application. [1]

IV. IMPLEMENTING SECURE SOFTWARE

Leveraging Developments Tools: Software development teams should use well-understood development tools which provides several important benefits. It facilitates effective and efficient code design by ensuring that developers understand each other's development environment. When one developer will work with another developer's code then first developer should be familiar with the developer's code. Common tools are helpful to security teams to understand the common development environment and how to apply the security principles in that environment. For example: Security teams can check for error and warning messages created during compilation of code and educate developers about the specific alerts that provides an advantage from a security point of view. [1]

Avoiding Common Flaws: Developers should identify the common flaws in code which leads to security vulnerabilities and which will be beneficial to implement a secure code. There are two common security flaws are SQL injection and cross-site scripting.

A) **SQL Injection**:

1. Many web applications based on a database to help to generate the dynamic content in response to user queries. The application makes a call to the database to retrieve content that is used in the creation of page satisfying the user's request.
2. For example: Consider a web application which allows the user to query for a product using a search string. The application can use the following template/syntax
3. SELECT * FROM Products WHERE productname LIKE '%<searching>%';
4. In this example <search string> can be replace using user supplied input before execution.
5. Normally the user might search for phone which will result in the following query template.
6. SELECT * FROM products WHERE productname LIKE '%phone%'; which retrieves products from catalogue containing the word phone.
7. However an attacker conducting a SQL injection attack might supply an unusual input : phone %'; DROP TABLE Products;
   - - ; which results in the following command being passed to database.
8. SELECT *
9. FROM products
10. WHERE Productname LIKE '%phone%':
11. DROP TABLE Products; --'%';
12. That is set of commands can be reformatted
    to make it easier for human eye.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 7, July 2015

13. SELECT * FROM Products WHERE Productname LIKE '%phone%';
14. DROP TABLE Products
15. – '%';
16. The database checks for middle semicolon and then executes three separate commands one that is normal query for phone, the second deletes all content of product table and third comment to be discarded.
17. To avoid the SQL injection vulnerabilities in Their code, the following three suggestions are explained: 1. Use parameterized queries
2. Perform input validation
3. Perform automated testing [1]

B) Cross-site Scripting

1. One way into which an attacker might perform this trick is to locate a website that allows individuals to post information for viewing by other users such as in online forum which might allow users to include HTML in their messages to control formatting.
2. The attacker finds such site can use the <SCRIPT> -- </SCRIPT> HTML tags to include JavaScript that will be executed by the user's web browser.
3. This code might request sensitive information from the user.
4. User feels that the message came from the trusted website and he might provide that sensitive information and user will not realize that sensitive information is being sent to a spiteful third party.
5. Similarly this (XSS) attacks may be used to steal session cookies used for authentication.
6. To protect from XSS attack developers can use an auto-encoding templating framework which can determine the context and encode data accordingly.
7. If developers do not interested in use of an auto-encoding templating system then developers must ensure that all user's sensitive data should be contextually encoded to prevent from cross-site scripting attack. [1]

Automating Security Analysis: Some of flaw may be exist in developer's code and to reduce these flaws, developer requires use of automated security analysis tools that evaluates code for common vulnerabilities, identifies flaws and if there are any error then automating security analysis provides guidance for remediation techniques or provides guidance for avoiding errors. These tools are helpful to reduce the common flaws in developer's code and improve the quality of code created by developers.

V. ASSURING SOFTWARE SECURITY

Continuous Security Assessment: Most of times developers may have question like when an organization should be build security analysis technique into SDLC. Developers seek to identify a particular point in the process where they can check off the security assessment box. The best answer to the above question is to perform the test "all the time". Continuous Security Assessment models integrates the security testing with all stages of software development and which allows developers to identify the errors as soon as possible and fix them before the project has progressed so far. By doing continuous security assessment developers are happy because they can correct flaws in code very quickly. Security professionals are happy because they can have confidence about quality of final product when it will step for production. It improves overall quality of code which implemented by development teams. When mistakes have been found at the time of review then developer's task is to solve those mistakes and by doing this developer can learn from them easily and improve their coding practices to avoid the similar vulnerabilities in the future. There are two types of Code Assessment Technique which are explained as follows:

A) **Static Analysis**: In this technique, we have to assess or check for only source code and not for executable code. Static analysis tools take the source code of the program as input and translate into an internal data

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 7, July 2015

model. Then we can apply set of rules to an internal data model to perform data flow analysis, structural review and an assessment of the flow of control of an application. The tool then generates a report of potential vulnerabilities for developers to review and if there are any corrections then those corrections should be remediated. The main benefit of static analysis is to check each and every line of code and does not depend on user or tool to identify the possible workflow is correct or not. Additional benefit is nothing but developers can perform static analysis in early phase of SDLC because it does not require binary or executable code.

B)

C) **Dynamic Analysis**: In this technique, we can check for executable code or binary code or .exe file. We can put an application into different steps into which we have to attempt to break it by supplying the wrong input or performing a series of unanticipated actions. To perform dynamic analysis, we can use automated testing tools and these tools in the world of web applications are referred to as web-application vulnerability scanners or penetration testers.

Revising the threat model and attack surface reviews:

As software projects going through an implementation and quality assurance phases, they often go through changes in both functionality and design. But this is particularly true in cases where organization have been choosing an agile model approach for software development. For this reason, developers must revise threat model and attack surface reviews that they created during design phase of SDLC. If the documentation reviews will results in the identification of new significant information then design should be reviewed to ensure that is it continuously meeting the projects minimum security requirements. If any additional new functionality introduced after the initial design this may require an introduction of new security requirement. [1]

VI. DEPLOYING SOFTWARE WITH SECURITY IN MIND

Conducting the final security review:

Final security review is a formal phase into which team can do the validation of an application whether it meets minimum security requirement and team can complete all security tasks during the appropriate stages of SDLC. The team who will conduct final security review should include at least one member who has not worked in earlier phases of project and this provides an important outsider's perspective to the project team. The controls implemented in the application should satisfy each of minimum security requirements and validation of controls is reviewed by conducting the final security review. Also the final security review checks or examines the results of final static and dynamic security analysis and be sure that project team should be addressed the corrections or findings from those assessment. [1]

Creating a Security Incident Response Plan:

This plan outline the process into which an organization will follow in an event of security incident. The template of incident response plan which is designed by National Institute for Standards and Technology (NIST) as shown in following figure:

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 7, July 2015

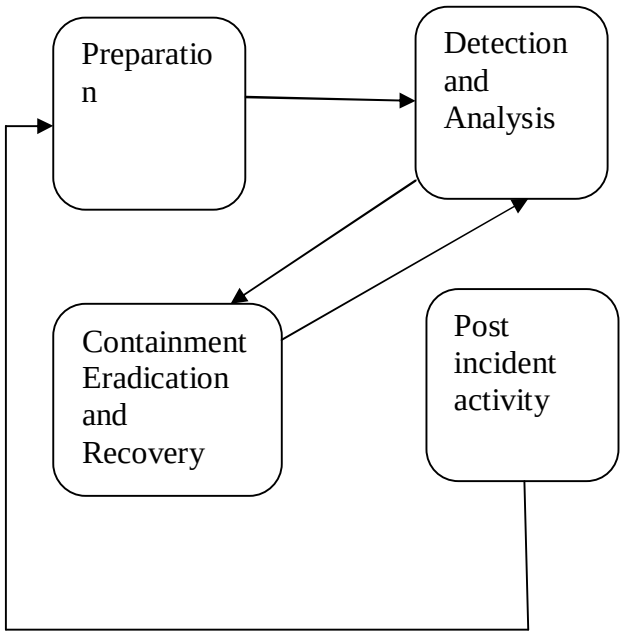

Fig.1. Incident Response Plan

In development phase security analysts should work cooperatively and closely with developers and operational staff to make sure that all the documentation is according to an organization's incident response plan or not. For Example: The plan may include the list of contact telephone numbers for individual responsible for a configuration and maintenance of application and servers supporting the application.

Developing a pre-deployment security checklist:
The pre-deployment security checklist may include the following things:

1. All connection strings and password used should be changed.
2. Perform a final review ensuring that no hard coded passwords or connection strings are identified in the code and that all have been moved to configuration files.
3. Complete the final security review.
4. Update the incident response plan, as necessary.

The checklist may be varying from organization to organization. But use of this approach is to verify that all steps have been completed and code may be released to production.

Releasing to Production: At the conclusion of deployment phase the code is released to production. The individuals who actually deploy the code are normally not members of the development team in order to isolate duties. Staff who will deploy the code to production should verify that the project has completed pre-deployment checklist. They then follow the organization's standard code release process to update the production environment.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 7, July 2015

VII. PICTORIAL REPRESENTATION OF PHASES IN SDLC

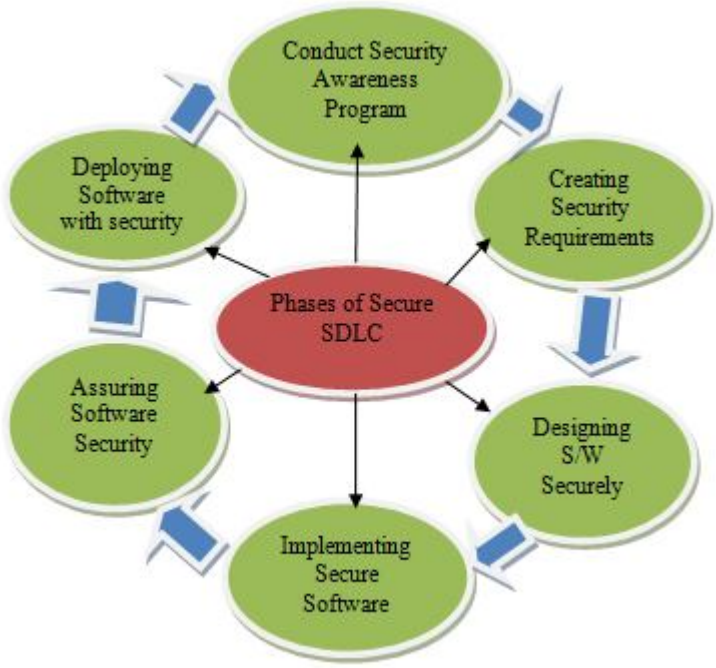

Fig.2. Phases in SDLC

VIII. CONCLUSION

Thus we can conclude that:

1. Using Security Awareness Training, we can secure software development.
2. Specify the minimum security requirements by conducting a risk assessment and defining clear security thresholds.
3. Incorporate security into the design phase by developing a threat model, by minimizing the attack surface & by performing a security architecture review.
4. Secure implementation by leveraging development tools, by avoiding common flaws and by automating security analysis.
5. Assuring software quality by continuous assessment via static and dynamic analysis.

Deploy software with security by conducting final security review, by creating a security Incident Response Plan, by developing a pre-deployment security checklist & releasing to production.

IX. FUTURE SCOPE

1. We can secure SDLC using conduction of Security Awareness Program.
2. We can secure Requirement phase by incorporating security parameters using security approach such as Security Requirement Table (SRT).
3. We can secure Design phase by incorporating security parameters using security approach: such as Security Design Table (SDT).
4. We can secure Implementation phase by incorporating security parameters using security approach: such as Security Implementation Table (SIT).
5. We can secure Testing phase by incorporating security parameters using security approach: such as Security Testing Table (STT).

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 7, July 2015

6. We can secure Deployment phase by incorporating security parameters using security approach: such as Security DEployment Table (SDET).

REFERENCES

1. A"Securing the SDLC for DUMMIES" by Jerry Hoff & Mike Chapple & published by Wiley Brand
2. "Threat Modelling and Security Pattern used in Design Phase of S-SDLC" by Mr. SWAPNESH TATERH, Prof. (Dr.) K. P. Yadav & Prof. (Dr.) S. K. Sharma, Volume 2, Issue 4, April 2012.

BIOGRAPHY

Mrs.Y.C.Kulkarni is a Associate Professor in the Information Technology Department, Bharati Vidyapeeth Deemed University College of Engineering, Pune. She received Master of Technology (MTech) degree from BVDUCOE, Pune,, India. Her research interests are Software Engineering, Web Engineering.