ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 3, March 2015

Secure Application Development in Academics

Ms. P.B.Alappanavar[1], Mr. Shashank A. Rangole[2]

Assistant Professor, Information Technology Department, Sinhgad Academy of Engineering, Pune, India[1]

Process Consultant, TCS, Pune, Indi[2]

Abstract: The skill gap between industry and academia is giving sleepless nights to IT Industry CEO's. Industry almost gives paid vacations to new trainee in the form of training during the initial months. This training goes on for a period of 3-4 and sometimes almost 6 months. This training is done as the industry believes that the under graduate education provided does not cover the required aspects for the trainee to work on live projects from day one. Hence in a certain way they do agree that there is a gap between the academia and industry. Today's need is that this gap has to be bridged as much as possible and in as many ways possible. There are several issues that need to be considered if we are to bridge this gap. One such concern is the security aspect. Security is not a major concern for students when they develop software projects minor ones like data base projects in third year and one major one in the final year. Security is not given priority at all, although it does give sleepless nights to the concerned industry. It has been proven that Organizations with world class Software Development Life Cycle (SDLC) practices that include security will experience an 80 percent decrease in critical vulnerabilities [1]. Hence it is necessary that students while developing software projects must consider security. This paper discusses ways in which students when developing minor or major projects can include security aspects, which would in turn train them to think of security as the developers concern and not just the security specialist concern. The paper also lists the benefits such considerations will have in future.

Keywords: Security, MS-SDL, OWASP, Checklists

I. INTRODUCTION

Undergraduate students while developing their software projects do not consider security design based features with the SDLC process they have chosen to implement. In-fact security is never considered by them when they have to work on any software project. For example, when students begin with the requirement phase they never take in to consideration that authentication mechanism should be a requirement that has to be satisfied. The requirement specification does not have a mention of the security requirements unless the project itself is based on security. Their sole main focus is to always develop the application and application only. As students develop applications over a period of three months for minor projects and around a year for major projects security can be implemented as a part of the development process in more than one ways.

This brings to focus the need to educate and guide the students about the importance of security while software projects are being developed. Students need to implement security not after application is developed and deployed, but during the stages of the products development lifecycle. They need to consider the fact that security of the application being built is not the concern of the security specialists in charge but also the developers. The subject's taught for the undergraduates just teaches them various software life cycles models some in detail and some not so detail. Security is never taught as a part these SDLC.

In fact in final year when students implement a software project for completion of their degree, security is never part of their SDLC. This is mainly because emphasis is not given to security during the implementation of software project. However this case is not true, because security of the software being developed is also the responsibility of the developer and hence the developer must be aware of the different concepts to implement minimum security features in the software application being developed, this can actually reduce dependency on the information security team.

This paper discusses ways to ensure that the security is considered as a key component during the development of software application.

II. RELATED WORK

1. MS-SDL

The Microsoft SDL is described as: The Security Development Lifecycle (SDL) is a software development process that helps developers build more secure software and address security compliance requirements while reducing development cost.[2] MS-SDL addresses the security concern by including security as a part of the development process and yet incurs lower cost. The phases include:

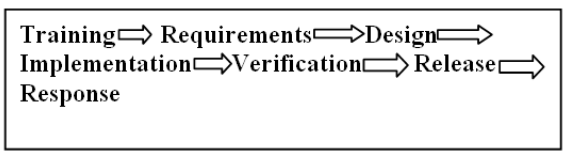

Figure 1: MS-SDL

A. Training

According to MS-SDL the first stage is Training that is essential for implementing the SDL. This includes training

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 3, March 2015

on foundational concepts that includes vulnerabilities, basic security mechanisms, privacy which would reduce the vulnerabilities associated with the software being developed.

B. Requirements
Second phase includes establishing security and privacy requirements. This should include identifying acceptable levels of security and privacy quality, assess security and privacy risks.

C. Design
There must be validation of the functions implemented against the design specification in order to ensure there will be minimum vulnerabilities.

D. Implementation Phase
This phase ensures that decisions made regarding secure ways to deploy the software. Best practices should be looked into, so that security issues can be detected and immediately removed from the code.

E. Verification
This phase ensures that the security and privacy concerns specified in the above phases are met.

F. Release
This phase is about releasing the software for public use and also to look into issues that may occur in the near future and how to handle them.

G. Response
This phase includes response and handling of security incidents that may occur.

2. OWASP
Software Assurance Maturity Model described by OWASP addresses secure application development. OWASP also lists out top ten vulnerabilities that application developers need to know during the process of application development.

3. Web Application Security Improving Critical Web Based Applications Quality through in-depth Security Analysis
"Web Application Security Improving Critical Web Based Applications Quality through in-depth Security Analysis", IEEE paper by Nuno Teodoro and Carlos Serrao discusses the ways to make a web application secure. They mention that the approach to security should be proactive and not reactive. The authors also propose a framework for including security as a part of web application development.

III. SECURE SDLC
Processes that include security are implemented by large software organization. In fact, now a day each organization prepares a process specific to the organization with security as its prime concern. The ones proposed by organizations are not feasible to implement to the text, as the project group would implement a software project at the under graduate level. However it is possible to include relevant information necessary to ensure security design as a part of the development process.

Once the student is made aware of the importance of security for the development of any software, we then need to provide a guideline as to how to achieve secure software with at least minimal security requirements fulfilled.

Several process models are known to the students. Most of the time students usually follow the Waterfall model to design and implement their projects. The stages include:

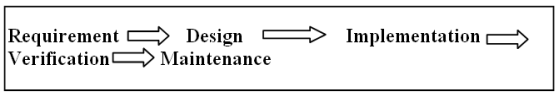

Figure 2: Stages of a SDLC

The phases of the waterfall model are well known. In the requirement phase the system requirements are collected. The system design is done at the design phase, implemented in the implementation phase, testing done in the testing phase to check the correctness of the system and finally the maintenance phase.

Considering the fact that 80% of the projects groups use the waterfall model to develop their projects we will consider how the best features of security can be a part of the waterfall model.

A. *Requirement with Training Phase*
Most of the process models followed to develop software applications do not include a training phase, where all relevant information is taught to the developer and all aspects are made clear. This is clearly lacking in the under graduate education system. If we are look at the various secure SDLC developed for various organizations, we see that they clearly include a phase where in training is provided to the developers. The MS-SDL proposes training as the first phase for any software to be developed.

So, first and foremost a training session is to be included. Here Students can be imparted the knowledge about the importance of including security during the development process. Guides can impart this knowledge to students or expert talk can be arranged, workshops included where students are emphasized on the need to include security in their design and development process. Students should be informed about the vulnerabilities of a software application, best practices available and how this can be included as a part of the development process.

If security is considered the responsibility of the developer, then first and for most it is important that students whom we educate to become developers should also be educated about the importance of considering security aspects within the SDLC.

To educate students about SDLC with security we could use various methods. Some of them include:

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 3, March 2015

(1) Students as a part of the curriculum can be introduced to various white papers which talk about security as a part of the development process. For example OWASP provides a guide to develop projects with security, SANS provides a guide for A Security Checklist for Web Application Design, the Microsoft library provides ways to develop secure ASP.Net application, IBM has its own secure software life development process etc.

(2) Students should be made aware of the existence of such documents or website's where they could get relevant information.

(3) Students should be taught about process models that include security as a part of the development cycle. Naming a few, MS SDL and OWASP Software Security Assurance Process etc.

(4) Workshops and Seminars can be conducted where in industry person can be invited to deliver on the expectations of the industry related to vulnerability identification and basic security requirements required in any application, tools available etc.

Training of this kind would compel the students to think on the lines of security and its need. Once this kind of knowledge is imparted, we need to ensure the students would include security concepts in the following phases of the SDLC.

The requirement phase of the waterfall is one where requirements that are required to develop the software are gathered. At this stage students should also include security requirements. OWASP lists out most of the requirements necessary for secure software development. Taking their specifications we propose a list that includes.

(1) Determining the strength of the authentication mechanism required. Usually students while demonstrating their software have passwords having length less than 4 characters. This mainly demonstrates the attitude of the student towards security.

Authentication mechanism should be multi factor scheme for any software. Developer should ensure the length of the password acceptable should be of standard size, password being stored in databases is not in plain text form but some salt value must be applied to the password that is stored.

(2) Defining security levels that are acceptable which would identify who can see what information and which would ensure that data would fall into wrong hands. Access control mechanisms like discretionary/mandatory / role based need to be determined and implemented.

(3) Minimum cryptographic requirements are to be decided. Key should be of proper length, its storage should be looked into and algorithm used should not be weak. Algorithms approved by NIST and NSA should be used. Since these are standard algorithms their concepts available so it is not difficult to incorporate them in our projects.

(4) Deciding whether additional security mechanisms like fire walls, IDS or IPS should be a feature of the application needs to be determined at this stage.

(5) Technology to be used and the risks associated with it also should be identified. With what technology what security features can be included should also be looked into.

Depending on the complexity of the project being developed they need to determine the security features that are to be implemented as a part of the project. The requirement stage should consider the above the factors.

B. Design

In the design phase the students can look at ways to implement the minimum security requirements. The design specification should be complete in all respects which can then be compared with functional requirements.

A check list can be included to ensure the design has taken care of the requirements. Some of the elements of the check list that can be verified against the requirements could include

Serial No.	Description	Yes/No	Comments
1	Does the design include authentication mechanism?		
2	Does the design consider authentication as a multi factor process?		
3	Does the design ensure password stored in crypt format?		
4	Security level design included?		
5	Access control Mechanism designed?		
6	Has the design included cryptographic algorithms that are standard?		

Figure 3: Checklist for requirements phase

This check list would ensure that the requirements have taken care of security features required. This has to be prepared in addition to any other check list implemented at design stage.

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 3, March 2015

C. *Implementation Phase*

Implementation Phase would implement the software taking into consideration the requirements and design. In this phase the students can ensure that the tools used are appropriate. To ensure that security is taken care of in the implementation phase students should ensure that unsafe functions are eliminated. For example in C the following use of functions like gets, strcpy, strcat etc. can create buffer overflows and hence are unsafe to use . Improper use of the Java Native Interface (JNI) can render Java applications vulnerable to security flaws in other languages.

[3]Such functions need to identified and then ensured such functions are not used. Such unsafe functions can be determined with the help of the guide or by an expert. This includes knowing the source language in which the project is coded thoroughly.

Identifying functions which would create security problems and finding ways to overcome them. Thus in the requirement phase we need to study our technology thoroughly. As students are given around 3 months to do requirement and analysis and design they do have ample time to find out such unsafe code and avoid using them.

Static code analysis can also be done using various tools they have learnt. Students are familiar with data flow diagrams, control flow graphs etc. Students should use them to determine vulnerabilities. For example using the control flow graph one can determine the complexity, which in turn can tell whether the code can be tested or not. A code which cannot be tested is not acceptable.

Error handling should be done and should be proper. Error messages displayed do give a lot of information to people who wish to exploit the applications. For example error messages do tell the unauthorized person how the file system implementation is done in the application.

[6]Developers when coding also should include the facility to log unauthorized events.

Hence in the implementation phase the students need to include facility to display appropriate error messages, log unauthorized usages. This needs to be checked by the project guide, that they should be implemented by the students.

D. *Verification phase*

Verification phase generally includes testing the software for functionality. Testing should also include testing of the security mechanisms specified. Groups should check for vulnerabilities that can be exploited for similar software.

The code can be reviewed for specifications statically. Unsafe functions can be eliminated using static analysis. A separate checklist can be prepared to ensure the security details are taken care of.

A sample check list is shown below:

Serial No.	Description	Comments
1	Is the authentication scheme multi factored	
	I. Is the password length size acceptable? I. Appropriate messages delivered if a rule is not followed? I. Are passwords in visible text format?	
2	Has the security levels scheme implemented? If yes, which? I. Is data displayed according to privileges given to the user? I. Is a user able to give a path name and view privileged data that he/she is not supposed to? I. File system access provided?	
3	Does the application lock itself out when there is inactivity for a fixed period of time/ ask for re authentication	
4	Cryptographic algorithms used are of high quality?	
5	Cryptographic algorithms used have a key of proper length?	
6	Storage of keys done correctly?	
7	Functions which are unsafe not used?	
8	Identification of sensitive data?	
9	Is sensitive data stored properly?	
10	Errors are handled or not accordingly?	
11	Unauthorized attempts are logged?	
12	Is there at least date and time log stamp?	

Figure 4: Checklist for Verification phase

A last check list would include a list of unsafe functions for the programming language is maintained and verified against the check list. This can be prepared and verified by the project guide along with the students.

IV. BENEFITS OF INCLUDING SECURITY IN SDLC

Including security as part of SDLC has many benefits. This includes

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 3, March 2015

(1) Student does not consider security as a separate component but develops software application which would have reduced vulnerabilities

(2) Students carry this concept when they move to the industry so the industry benefits as they would develop applications which are less vulnerable to security threats

V. CONCLUSION

They say small measures go a long way. Security which is the need of the hour today should be taken into consideration when developing any kind of application. Security should be taken seriously by the students. They should consider it as an important feature of the application. Security measures should be included during the development of minor or major projects. Once Check lists such as one suggested above are verified, then we can ensure that security is considered with seriousness as required. Elements can be added to the check list as and when required. During implementation small measures to ensure safe coding also would ensure that the code is not vulnerable to many security attacks. Such measures ensure the students would consider security as a part of the development of the software.

Thus we can conclude that by including such small measures students can ensure that security is a part of the development software.

REFERENCES

[1] Sec AppDev 2013 Gartner
[2] http://www.microsoft.com/security/sdl/default.aspx
[3] http://www.sans.org/reading-room/whitepapers/application/security-scenarios-analysis-design-29
[4] https://www.owasp.org/index.php/Unsafe_JNI
[5] http://www.everyspec.com
[6] http://www.sans.org/reading-room/whitepapers/securecode/security-checklist-web-application-design-1389