International Journal of Engineering & Technology, 7 (2.4) (2018) 1-4

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research Paper

A Study of major secure SDLC processes in web based applications

Subhranshu Mohanty [1] *, Amar Kumar Mohapatra [2], Srikanta Patnaik [3]

[1] Assistant Professor: Information Technology, Army Institute of Management & Technology Greater Noida, India
[2] Associate Professor & Head: Department of IT Indira Gandhi Delhi Technical University for Women
[3] Professor: Department of CSE, Siksha O Anusandhan University, Bhubaneswar, India
**Corresponding author E-mail:*

Abstract

Web applications have become important but there are different types of security problems which could lead to tampering with details. The most common are cookies poisoning, structured query language, cross-site scripting and parameter tempering. This is the reason why most of the web companies today are verifying the type of content they receive and most importantly, from where the contents are originated. It has been thus noted from the above deduction that the major security threat has nothing to do with the Secure Socket Layer rather other layers in the web development program. In order to avoid such threats and other vulnerabilities, initial stages of the web development cycle need to be taken care of. Thus, the main focus of this research paper is to come up with a framework that would help to strengthen the security of the various stages in the web development cycle. For the same, various modules and life cycles have been used.

Keywords: *Framework; Threats; Web Applications; Web Development Cycle.*

1. Introduction

There are plenty of existing gauges, models, procedures, systems and other sources through which a substantial secure software development system can be built that doesn't compromise the ethnicity of the various stages. While the major attacks on the web happen through the application layer which is around 75%, the other major points of vulnerabilities are the SQL (Structured Query Language) injections and XSS (Cross-Site Scripting). [1] Browser vulnerabilities are also other weak points for attacking the security of web applications. [2].

According to Sysadmin Audit Network Security or SANS/FBI (2009) report from March to August 2009, the top security threats to web, networks and online businesses are SQL injections, PHP (Personal Home Page) and XSS. When it comes to top threats and vulnerabilities, these three top the list.

A survey conducted by Danny (2007) and found that cross-site scripting holds 80% in terms of threats to web applications, followed by SQL injections at 62%. Also, parameter tempering is 60% dangerous to the security while cookie poisoning is at 37%. [3] This is also becoming one of the biggest reasons today, why online companies are verifying the source of the content they receive which in turn will not compromise their web applications. [4].

Thus it is safe to conclude that most of the problem and threat that web applications have, is not from the Secure Socket Layer but the application layer itself. Better methods of application development are already being looked at to ensure that there are no future threats. [5] The basic cause of threats in web applications is obviously with the development life cycle. [6] Therefore, to control security at initial stages of web application development is required for mitigation. [7].

Proper checks have to be maintained along with parameterized queries for database access so that there are no threats to the program because of improper coding which is the SQL injection. The development methodology has to be such that there is no scope for attacks to be repeated and this will only happen if there are spaces for checks and balances in the program and codes.

The focus of this research paper thus comes down to ensure that there are enough security checks enabled at the development stage also the coders must have knowledge to deal with the threats and attacks when they are lined-up according to their intensity.

2. Review of security models in web application development

a) Software security framework

Software Security Framework (SSF) for the web application development would be built with the help of Diana and Security Curve (2009). [8] The foundation on which the structure was built for the same was on Security Account Manager (SAM) and other advertising modules which helped in associations and explaining all this in better light. Building Security in Maturity Model (BSIMM) has already being verbalised because the arrangements of associations are also needed to be investigated.

Four major distinctions have been taken in account for understanding the 12 major topics. The 4 spaces are:

- Governance: help in sorting out and taking care of the entire security aspect.
- Intelligence: software security exercises are taken care of by honing the corporate aspect of it.
- SDLC Touchpoints: taking care of the standard security development and other inquiries.

- Deployment: network security is taken care of along with software upkeep solutions.

b) Agile Methods

In agile software development, there are various iterations and modules which ensure that the software development life cycle (SDLC) is always accompanied with new structures. The application module is dynamic at best and this helps in getting the structure right. The iterative approach is even better for the team working on the development. They can work on any type of threat to the software application early on. Security should not be treated as something separate from the agile development process. In order to improve security, there has to be modules incorporated in the agile development process. Security evaluations can be prepared in a better light if the associations' hazard dealing point is strong. The leadership will also focus on what to look for when they prepare the evaluations around the security hazards of the program. Cross webpage scripting and SQL injection are two of the biggest cause of concern when the security of the application is concerned. Moreover, they will not be able to blend in with the expanse of things that are on the internet.Creation is important and comes first. Knowing all the vulnerabilities before creation is idealistic and practically inconceivable. The entire application is expansive and there has to be a basic way to monitor everything properly. Discovering vulnerabilities often takes time and there are various ways to do it.

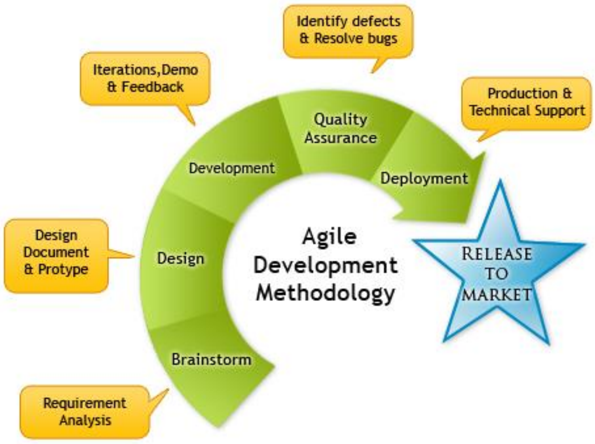
Identify defects & Resolve bugs
Iterations,Demo & Feedback
Production & Technical Support
Quality Assurance
Development
Deployment
Design Document & Protype
Design
Agile Development Methodology
RELEASE TO MARKET
Brainstorm
Requirement Analysis

Fig. 1: Agile Secure SDLC Processes. (Source: Binary Semantics, 2013).

3. Security engineering process integrated during design phase of software development life cycle for web-based applications

Software development and the security of this software are two sides of the same coin and this is what the previous developments from Keramati and Mirian-Hosseinabadi (2008) states. [9] Agile methodology focuses on integrating these two concepts. In order to increase the trust factor of the product, company's today are restraining the efforts to reduce the agility as this along with other features helps in improving the security. In the quest to display the vulnerabilities and other threats, Meledath (2006) used various use and misuse cases to show the various threat points. [10] This helped in understanding the security problems and the requirements as well. While the software life cycle and its initial development stages are explained through this, these case studies are better suited for students and novice researchers. Apart from this, there are the industrial and commercial considerations as well and for this Eduardo (2004) has developed various patterns and object-oriented design and the Unified Modeling Language (UML). However, there are no definite guidelines on how to develop the above mentioned methodologies precisely. [11] Software development process still remains the center focus and that is why Ge et al (2006) has discussed the risk analysis and feature centered development process to talk about the security of the web application process and its development. [12] Risk analysis and web assessment is the process which has been fused together to give a better security approach to web development application process. The FDD (Frequency Division Duplex) is the aspect which is more driven to the built and design of the web application development process rather than the entire process. [13] Apart from this, there is a time factor as well which is equally crucial.

Life cycle approaches are important when pitching the security features for the web application development process and this has been shown by Sengupta et al (2005). [14] The frameworks for mitigation were developed properly and put in function after they successfully showed what are the major threats and attack points or perceived vulnerabilities with e-commerce. On the other hand, Scott and Sharp (2002) are more focused on the insecurity that has plagued the web development process and in order to deal with this, they are ready to decrease the focus from the development time because security is above everything. They are also working on developing applications against attacks. [15] Thus, they have developed an object oriented API which protects the layers of application by extracting any SQL injections at early levels.

4. Analysis and results

A. Comparative Analysis Result

Sengupta et al (2005) and Ge et al (2006) [14] [12] are the ones that have provided two solid and substantial framework for web development security threats. They have proposed proper methodologies for the security purpose as well.

Xiaocheng's model is another prominent model which even after having definite security features in the requirement stage lacks the guidelines that are required to convince stakeholders to be interested in this model. For Sengupta's model, the opposite happens as they have the required stakeholder collaboration and lack the security features at the initial requirement stages. Therefore, the combination of both the framework is important in order to protect the security of the intellectual properly.

Upon closer inspection of the three models and their use and misuse case studies, it has been found out that Xiaocheng's model doesn't use the misuse cases at all in any of the stages of development.

Moreover, Sengupta's model does not look at the misuse and use cases at any stage of design and requirement stage. However, these cases have been used in our model because they are beneficial in latter stages. At early stages as well, they help in uncovering the threats to the web applications.

This is the reason why the proposed model is becoming popular. It shows significant growth in the analytic factors apart from everything else. It also shows that these factors are key in security of the web application development.

5. Proposed secure e-commerce development framework

Conventional software system and security system have major points of difference because the latter borders on ensuring that the web application development process doesn't get lost in the myriad security attacks. The latter focuses on the functional accuracy of every stage. (Sengupta et al, 2005) The security system approach is becoming increasingly important because there are different credit cards, bank details and pin numbers involved which need optimum security or else the customer would be at loss.

Since the vulnerabilities are threats to the web development programs are being released every day, the security methodology has been tightened and made better in lieu of this. This new methodology will focus on all the stakeholders and their requirements so that the security of the program or the development life cycle is protected from all the perspectives.

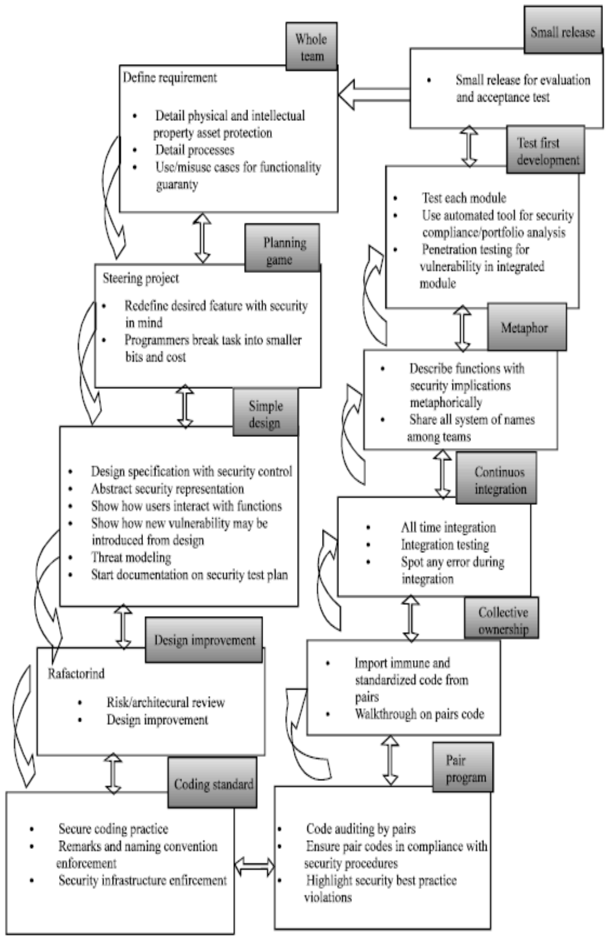
Whole team
Define requirement
Detail physical and intellectual property asset protection
Detail processes
Use/misuse cases for functionality guaranty
Small release
Small release for evaluation and acceptance test
Test first development
Test each module
Use automated tool for security compliance/portfolio analysis
Penetration testing for vulnerability in integrated module
Planning game
Steering project
Redefine desired feature with security in mind
Programmers break task into smaller bits and cost
Metaphor
Describe functions with security implications metaphorically
Share all system of names among teams
Simple design
Design specification with security control
Abstract security representation
Show how users interact with functions
Show how new vulnerability may be introduced from design
Threat modeling
Start documentation on security test plan
Continuos integration
All time integration
Integration testing
Spot any error during integration
Collective ownership
Import immune and standardized code from pairs
Walkthrough on pairs code
Design improvement
Rafactorind
Risk/architecural review
Design improvement
Coding standard
Secure coding practice
Remarks and naming convention enforcement
Security infrastructure enfircement
Pair program
Code auditing by pairs
Ensure pair codes in compliance with security procedures
Highlight security best practice violations

Fig. 2: Proposed Secure E-Commerce Development Framework (Source: Self Constructed).

Figure 2 discusses the proposed Secure E-Commerce Development Framework, wherein:

- Whole team: The security requirements and controls should be the priority of the development program and this is the reason why a stakeholders meeting before anything is initiated. Many business representatives are laymen when it comes to understanding the various aspects of the project and this should be explained to them by analysis and testers. The managers are the ones who will be dealing with the logistics of the project. The various use/misuse cases will come in handy to recognize the compromising points of the project. Requirement changes are inevitable but discussing them at this point is always helpful so that no information leaks out and the review of the misuse/use policy is done properly. Apart from this, the intellectual and physical property also come in the picture and should be taken care of.
- Planning game: The knowledge about the entire project that a businessman acquires in the initial stages is important as they will then be able to work from the security perspective. The major focus of this stage would be to set the right targets for accomplishing the goals related to the project. The entire team also has to focus in every aspect of the project and how much each of it will cost.
- Simple design: Revisions in the design of the project are mandatory because it cannot be decided in one single sitting. Moreover, when the design features include the security features, then there are different controls which are introduced at the initial stage of the web software development. The security features and plans are further upgraded for the test-first development with the help of risk analysis and threat models. Details of security abstraction should be given at every point.
- Design improvement: Factoring is the continuous improvement of the project. This stage involves handling new changes in the design as well looking out for any impending threats to the project. A good design is guaranteed at this stage. Moreover, the various architectural constraints are to be dealt with at this point.
- Coding standard: Naming conventions and remarks are few of the basic security practices which should be taken care of during the coding stage. In order to protect all the codes from the various team members, a proper security infrastructure should be levied which will also grant collective ownership.
- Pair programming: It is better to have two or more programmers write the code as this will help in getting the codes right. They should also adhere to the security guidelines.
- Collective code ownership: When different programmers are working on the project, all of them will have permission to access each other’s codes through collective code ownership.
- Metaphor: Sharing information with the team members is important so that everyone knows where to find functionality. Also, it will be easier for the coming programmers to post a remark in case there is a security concern.
- Test-first development: Feedback should be exchanged at every single stage of the code development. Before integrating the entire code, there has to be proper collation of the data. Also, security test has to be enforced at each iterative step to find out any impending threats or attacks. This is done to get an assurance of the project. Security compliance can only be achieved by running the checks on the source code as well. For tackling the cross-enterprise risk, portfolio analysis checks may be incorporated.
- Continuous integration: Always look out for the mistakes and errors in the program which might be missed by the testers.
- Small release: The customer should be given a small peak of the project to know what they want and how the future development should be planned out.

6. Conclusion

It can be inferred from the research paper that when it comes to delivering a secure web application, there are various challenges which are still being dealt with. The biggest challenge of course lies with the e-commerce web application where making the changes last minute or checking for threats and other type of attack cannot be done at the last minute or else the entire website will be compromised. Right from the requirement stage to the deployment stage, there has to be proper security measures taken or else the information of every user will be at risk. To provide this safety to the e-commerce web applications, our security development framework Extreme Programming methodology is designed to deal with the security lapses from the initial stage and not at the last moment. The life cycle is not extended but the security checks are imposed at every stage. Also, the pair programming works as a double check on the vulnerabilities. Thus, the next part of this research paper will focus essentially on the improvement of the given framework by keeping in mind the various perspectives.

References

[1] Adrian, O. (2008). Web application vulnerability and IBM rational appscan. Proceedings of the IBM Rational Software Development Conference 2008, (RSSDC'08), Orlando FL, pp: 79-88.

[2] Wang, L., H. Mu, L. Xu, J. Chen, X. Liu and P. Chen. (2010). Trojan URL detector: A statistical analysis based trojan detection mechanism. Inform. Technol. J., 9, 1124-1132. https://doi.org/10.3923/itj.2010.1124.1132.

[3] Danny, A. (2007). Managing a growing threat: An executives guide to web application security. Web Application Security Executive Brief. New York, USA, pp 1-8.

[4] Ilyas, Q.M., Y. Zongkai and M.A. Talib. (2004). A journey from information to knowledge: Knowledge representation and reasoning on the web. Inform. Technol. J., 3, 163-167. https://doi.org/10.3923/itj.2004.163.167.

[5] Stuttard, D. and M. Pinto. (2008). the Web Application Hackers Handbook: Discovering and Exploiting Security Flaws. Wiley Pub., Indianapolis, IN.

[6] Stijn, V.K., (2004). Threat Model for Web Application Using STRIDE Model. Royal Halloway University, London.

[7] Caleb, S. and L. Vincent. (2007). InforSecWriters. Effective controls for attaining continuous application security throughout the web application development life cycle. Retrieved from http: //www .infosec writers. Com /texts.php?op=display&id=583

[8] Diana, K. and Security Curve. (2009). Practical approaches for securing web applications across the software delivery lifecycle. IBM White Paper 3-7, USA.

[9] Keramati, H. and S.H. Mirian-Hosseinabadi. (2008). Integrating software development security activities with agile methodologies. Proceedings of the 2008 IEEE/ACS International Conference on Computer Systems and Applications, 749-754. https://doi.org/10.1109/AICCSA.2008.4493611.

[10] Meledath, D. (2006). Secure software development using use cases and misuse cases. Information Systems. Retrieved from http://www.iacis.org/iis/2006_iis/PDFs/Damodaran.pdf

[11] Eduardo, B. (2004). A methodology for secure software design. Proceedings of the 19th International Conference on Database and Expert Systems Application Turin, (ICDESAT'04), Boca Raton, FL, 1-7.

[12] Ge, X., R.F. Paige, F.A.C. Polack, H. Chivers and P.J. Brooke. (2006). Agile development of secure web applications. Proceedings of the 6th International Conference on Web Engineering. California, USA. 305-312. https://doi.org/10.1145/1145581.1145641.

[13] Palmer, S.R. and M. Felsing. (2001). A Practical Guide to Feature-Driven Development. 1st Edn. Mumbai: Pearson Education, 299.

[14] Sengupta, A., C. Mazumdar and M.S. Barik. (2005). E-Commerce security: A life cycle approach. Sadhana, 30, 119-140. https://doi.org/10.1007/BF02706241.

[15] Scott, D. and R. Sharp. (2002). Developing secure web applications. IEEE Internet Comput. , 6, 38-45. https://doi.org/10.1109/MIC.2002.1067735.