(IJUSE) 2016, Vol. No. 2, Jan-Dec e-**ISSN: 2454-759X, p-ISSN: 2454-7581**

SECURITY ENHANCEMENT IN SOFTWARE DEVELOPMENT LIFE CYCLE

Mathan Kumar M[1], Dr. Anu Bharti[2]

[1] *Research Scholar, Dept. of Computer Science & Engineering, Sunrise University, Alwar*

[2] *Asso. Prof., Dept of Computer Science & Engineering, Sunrise University, Alwar*

ABSTRACT

Software has become an integral part of everyday life. Every day, millions of people perform transaction through various applications run by these software as internet, ATM, mobile phone, they send email etc. People use software bearing in mind that it is reliable and can be trust upon and the operation they perform is secured. Now, if this software have ensemble security hole then how can they be safe for use. Security brings value to software in terms of people's trust. The value provided by secure software is of vital importance because many critical functions are entirely dependent on the software. With limited budget and time to release software into the market, many developers often consider security as an after though. So in this work we have given a model to improve the security in SDLC model by using different method.

Keywords: Software security, SDLC, Neural Networks, security rules, security risk

INTRODUCTION

Software security is the idea of software engineering to protect the software from the unauthorized access and also protect from the malicious attack. Software security makes the system software function continuously and correctly under malicious attack [1]. There is various approach currently used for integrating security in software development life cycle model. Implementing security from earlier stage of the software development makes the system fault free as possible and less vulnerable. As the day by day technology increases there is wide use of software which increases the security threat of software. Various security attacks due to security flaws in software harm the organization and also affect the financial status as well as integrity of the organization. Security is not a unique feature of software it's a important part of the software which is to be done carefully during the development of the software. Application security aspect must be integrated at software development process. After development process testing is not sufficient because it's too late to fix the bugs and mistake. Security is implemented at every major phase of software development life cycle. There are various approach is used to integrate the security at various level of SDLC Some of the author's focused the security at the initial phase(requirement & Design phase) and some at the other half (coding & testing) of the development phases. [11]

(IJUSE) 2016, Vol. No. 2, Jan-Dec e-**ISSN: 2454-759X, p-ISSN: 2454-7581**

RELATED WORK ON THE ASSESSMENT OF SECURITY IN SDLC MODEL

In order to design software more secure there is many approaches have been adopted at the various level of software development life cycle model. Some of these approaches given below.

A. Software Security Rules:

There are twenty one rule is proposed in this paper to make software more secure. If these security rules followed properly, it will help the security mechanism. These rules protect the software from unauthorized access; protect software from being infected, provide access control of the software, help to make software more accurate and consistent, also helps to improve the interoperability. [1] Gegick and Williams also proposed a regular expression-based attack pattern [9] .In this approach software component is used to identifying the vulnerability in software design. It consists of attack library of abstraction which is used by software engineers conducting Security Analysis for Existing Threats to match their system design. If a match found then it indicate that vulnerability exit in the system being analyses and helps to overcome these flaw before start coding. Threat modeling is an important activity carried out at the development phase to protect the software application from threat and vulnerability in order to provide a more accurate sense of its security [2], identify vulnerabilities, threats, attacks and countermeasures which could influence a software system [3]. Threat modeling helps to understanding how a malicious attacker chooses targets, locates entry points and conducts attacks [4]. The key to threat modeling is to measure where most effort is needed to make system software more secure [15]. Architectural risk analysis is also used to identify vulnerabilities and threats at the design phase of SDLC. The threat and vulnerability may be malicious or non-malicious in nature due to a software system. It examines the preconditions that must be present for the vulnerabilities to be exploited by various threats. The advantages of architectural risk analysis is that it enables developers to analysis software system from its component level to its environmental level in order to evaluate the threats, vulnerabilities and shock at each level.[5][28] . Attack trees approach is used to characterize the security of the system. In this approach attackers tree is generated and the root of the tree is represent the goal of attackers. The node of the tree represent the different type of action is taken by the attackers to accomplish his goal. The attack tree is used to analyze the risk and also in design, implement and measure the attack. So this approach is basically used in design phase of the SDLC [6]. Attack net is the similar approach to attack tree which include places equivalent to node in attack tree to point out the state of attack. In this when event move from one place to another place are captured in the transition and the transition point out the path of an attacker takes [7]. The vulnerability tree approach the tree is constructed and on the basis of this tree we find out the how one vulnerability relates to another and the steps an attacker may take to reach the top of the tree. Vulnerability trees help to analyze the different possible attack scenario that attackers can take to exploit the vulnerability [8]. The neural network approach, security flaws is analyzed in software design based on the abstract and matching technique through which software flaws can be easily identified when attack pattern is matched to the design. The Williams and Gegick [9] proposed regularly expressed attack pattern is used to identify

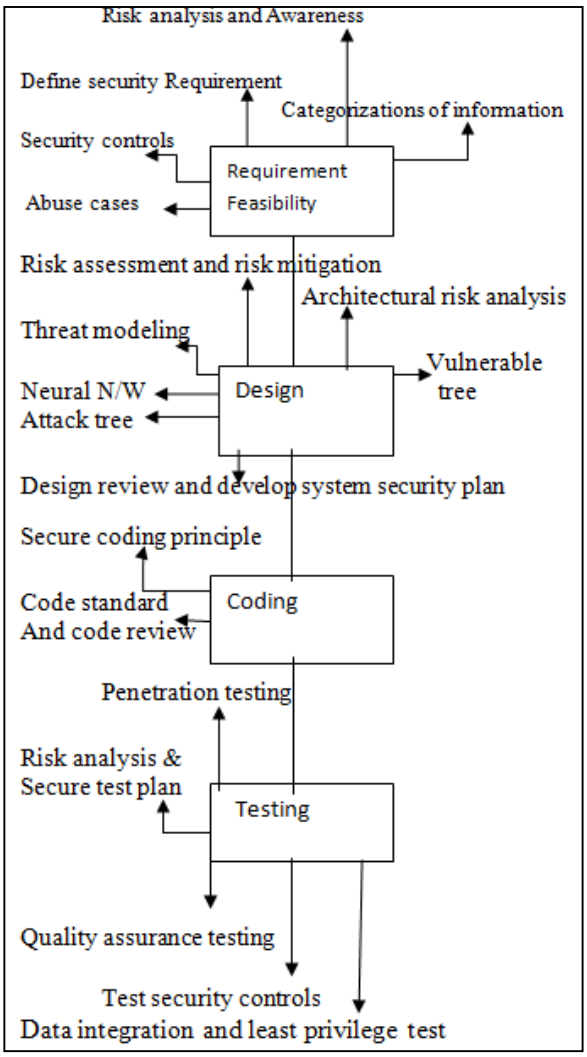
the actor and software components. Online vulnerability database is used to identify the attack scenario corresponding to attack pattern. Three layer feed forward back propagation is used to for the architecture of the neural network. A training data is passing as an input and the input is match to online vulnerability data base. [10][12]

PROPOSED WORK

As attackers become more sophisticated, there are new ways of exploiting software. In addition, new software and development environments will establish new types of vulnerabilities that presently may be unknown. To ensure that the software development community continues to implement effective countermeasures against the latest known attacks, it is important to analyze the latest exploits to see whether they represent any new types of attacks. Only after the attacks are characterized can effectively handle. In addition, analyzing the latest exploits and generating new attack patterns is an essential prerequisite to the creation of effective security policies.

In this regards we have created new policies which includes various counter measures to abolish the attacker's effect over the software. These policies over the different phases of the software development cycle as shown in the figure.

(IJUSE) 2016, Vol. No. 2, Jan-Dec e-ISSN: 2454-759X, p-ISSN: 2454-7581



(IJUSE) 2016, Vol. No. 2, Jan-Dec e-**ISSN: 2454-759X, p-ISSN: 2454-7581**

REQUIREMENT ANALYSIS

The purpose of the Requirements Analysis is to transform the needs and high-level requirements specified earlier into measurable, testable, traceable, complete, consistent, and stakeholder-approved requirements. The problem is that existing methods of developing secure software usually do not satisfy the requirements about security threats, risk assessment, security mechanisms and finally a systematic process for software security [16][32]

A. Risk Analysis and Awareness

Analyzing risk at the requirement phase is minimizing the cost of the project. After analyzing the risk at earlier stage and reducing them makes the software more secure and less vulnerable. [30][32]

B. Categorizations of Information

In this the requirement is divided in to functional and nonfunctional requirement, which helps fulfill the entire requirement and makes the software more secure. [17][18]

C. Security Requirements

There have been numerous research activities in different contexts of Security requirement. The most important of which is to develop methods of eliciting and modeling security requirement. Use cases, misuse cases are widely accepted methods of eliciting, documenting and analyzing functionality requirements of systems. So it helps to improve security and overcome to the security flaws. [19]- [23], [31]

DESIGN

In order to design software more securely and flaw less many approaches have been adopted for assessing the security in software designs during the design phase of Software development lifecycle model. Design one of the most important phases to develop software which meets all the requirement and security features. In this work we have used the methodologies which are best for the security features.

A. Threat Modeling

Threat modeling is an important activity carried out at the development phase to protect the software application from threat and vulnerability in order to provide a more accurate sense of its security. [2] Threat modeling is a technique that can be used to identify vulnerabilities, threats, attacks and countermeasures which could influence a software system [3]. Threat modeling helps to understanding how a malicious attacker chooses targets, locates entry points and conducts attacks [4]. Threat

modeling addresses threats that have the ability to cause maximum damage to a software application. The key to threat modeling is to measure where most effort is needed to make system software more secure. [15]

B. Architectural Risk Analysis

Architectural risk analysis is used to identify vulnerabilities and threats at the design phase of SDLC. The threat and vulnerability may be malicious or non-malicious in nature due to a software system. It examines the preconditions that must be present for the vulnerabilities to be exploited by various threats. The advantages of architectural risk analysis is that it enables developers to analysis software system from its component level to its environmental level in order to evaluate the threats, vulnerabilities and shock at each level.[5][28]

C. Attack Trees

This approach is used to characterize the security of the system. In this approach attackers tree is generated and the root of the tree is represent the goal of attackers. The node of the tree represent the different type of action is taken by the attackers to accomplish his goal. The attack tree is used to analyze the risk and also in design, implement and measure the attack. So this approach is basically used in design phase of the SDLC. [6]

D. Attack Nets

Attack net is the similar approach to attack tree which include places equivalent to node in attack tree to point out the state of attack. In this when event move from one place to another place are captured in the transition and the transition point out the path of an attacker takes. [7]

E. The Vulnerability Tree:

In this approach the tree is constructed and on the basis of this tree we find out the how one vulnerability relates to another and the steps an attacker may take to reach the top of the tree. Vulnerability trees help to analyze the different possible attack scenario that attackers can take to exploit the vulnerability. [8]

F. The Neural Network Approach:

In this approach security flaws is analyzed in software design based on the abstract and matching technique through which software flaws can be easily identified when attack pattern is matched to the design. The Williams and Gegick [9] proposed regularly expressed attack pattern is used to identify the actor and software components. Online vulnerability database is used to identify the attack scenario corresponding to attack pattern. Three layer feed forward back propagation is used to for the architecture of the neural network. A training data is passing as an input and the input is match to online

vulnerability data base. [10][12]

G. *Risk Assessment and Risk Mitigation:*

As the design begins to take shape, the risk assessment perform, which is the process of analyzing and measuring risk, and establishing an acceptable level of risk for the system. Risks are assessed by examining possible threats, identifying the vulnerabilities, and impact of vulnerability. Once the risk is identified, a risk mitigation strategy can be established. In this work for the risk identification and mitigation, threat modeling, attack tree, vulnerability tree is used which is most efficient for this purpose. [25][26]

CODING

In this phase all the component of the software coded by the developer. At this stage the main vulnerability is buffer overflow and it can't be stressed more so it is very important to safeguard the code against letting in such vulnerability. In this work we have tried to make code more secure.

A. *Cognizance of Security Risk:*

It is the job of leader and security officer to stress the importance of security in whole team. They should know the security risk, guidelines, procedure of the organization [25]

B. *Secure Coding*

Some of the security guidelines and rule that need to be followed when writing the code. Some of these as follow.

- Input validation
- Hostile environment
- Open standards
- Trusted component
- Always authenticate
- Native security protection
- Fail securely
- Log monitor audit.
- Least privilege
- Exception handling
- Strong cryptography
- Random number [25][26]

C. *Static Analysis Tools/Code Audit*

(IJUSE) 2016, Vol. No. 2, Jan-Dec e-**ISSN: 2454-759X, p-ISSN: 2454-7581**

Tools that scan source code for common vulnerability; it's a good way to discover the vulnerability in the code. It's an analysis of source code of a project with the intent of discovering bugs, violations of programming conventions. It's a part of secure coding which reduce the error before software is released. [25][27] [29]

TESTING

Testing is very important and valuable phase in the software development lifecycle. It is better to start testing at initial phase to avoid the difficulty by correcting the bugs at last stage. The importance of testing in SDLC is to improve reliability, performance and remove bug and makes software more secure.

The security testing performs by quality assurance team, formal evaluation, risk based security testing, secure test plan, privilege test, and other types of testing like unit testing , penetration testing. So in this work we have improve security of the software by using different type testing and analysis [25][27][33]

Risk based security testing is based on the threat model and attack pattern [27] So it is also helpful to remove the flaws and vulnerability in design phase also.

A. Integration/Quality Assurance Testing

In this type testing the entire module assembled and tested as an integrated application. Authorization and authentication test also perform. So it helps to overcome from security flaws. [25][27]

B. Penetration Testing

This is the last stage of the testing in which professional hacker attempt to penetrate the system. Penetration testing helps to protect unauthorized access of the system and malicious attack. It is good idea to have this testing done by the outsider or by third party who had no involvement in design and development of the system. [25-27]

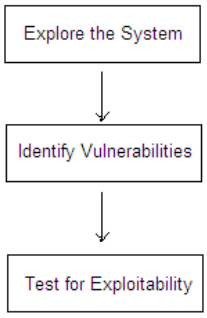
C. Test Security Controls

Security controls are safeguard or counter measures that minimize and avoid the security risk. The safeguard is an action, device or a technique that reduces threat, vulnerability, and attack by eliminating and preventing it or by discovering it so that the correct action can be taken.[34][35]. So it is very important test the security controls at this stage. It helps to avoid security risk. [25]

D. Risk Analysis and Secure Test Plan

Risk is a potential that chosen action or activity will lead to undesirable result or loss. Risk is the effect of uncertainty on objective. So better and secure test plan is needed to minimize the risk and overcome

future uncertain events. A three step test plan is used to identify and remove the risk. [36]



[36]

Test performs to understand the system that can't be obtained by the public knowledge. Then, individual vulnerabilities are tested, based on an understanding of the threats. Finally, any vulnerability identified is further tested to determine the risk of exploitation they represent. [36]

CONCLUSION

The entire models discussed above have many advantages and limitation. We have studied various model and their methodologies to making software more secure and to protect software from unauthorized access. Some of the author's focused the security at the initial phase (requirement & Design phase) and some at the other half (coding & testing) of the development phases. [11] In this work we have work on enhancing security by integrating individual phase to a single proposed model. After applying to we have found that the security risk is minimized and also security flaws, vulnerability are removed at some level. So this model is very useful for enhancing the security in SDLC when secure software is needed.

REFERENCES

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

[2] Agarwal, A. 2006), "How to integrate security into your SDLC",
Available at:
http://searchsoftwarequality.techtarget.com/tip/0,289483,sid92_gci1174897,00.html,

[3] Meier, J. D., Mackman, A. And Wastell, B.(2005), Threat modelling web applications", Available at: http://msdn.microsoft.com/enus/library/ms978516.aspx

[4] Redwine, S. T. Jr and Davis, N.; et al, (2004), "Process to produce secure software: Towards more secure software", National Cyber Security Summit, Vol. 1

[5] McGraw, G. (2006), "Software security: building security in", Addison-Wesley, Boston, MA

[6] Redwine, S. T. Jr and Davis, N.; et al, (2004), "Process to produce secure software: Towards more secure software", National Cyber Security Summit, Vol. 1

[7] Gegick, M. and Williams, L. (2006), "On the design of more secure software-intensive systems by use of attack patterns", Information and Software Technology, Vol. 49, pp 381-397

[8] Ralston, P.A.S; Graham, J.H and Hieb, J. L. (2007), "Cyber security risk assessment for SCADA and DCS networks", ISA Transaction, Vol.46(4), pp583- 594

[9] Gegick, M. and Williams, L. (2006), "On the design of more secure software-intensive systems by use of attack patterns", Information and Software Technology, Vol. 49, pp 381-397

[10] Security Assessment of Software Design using Neural Network A. Adebiyi, Johnnes Arreymbi and Chris Imafidon School of Architecture, Computing and Engineering University of East London, London, UK

[11] Julia H. Allen, Sean Barnum, Robert J. Ellison, Gary McGraw,Nancy R. Mead: Software Security Engineering: A Guide for Project Managers, Addison Wesley Professional, 2008, pp 6-8.

[12] Srinivasa, K.D. and Sattipalli, A. R, (2009), "Hand written character recognition using back propagation network", Journal of Theoretical and Applied Information Technology, Vol. 5(3), pp 257-269

[13] J. Wing, "A Call to Action: Look Beyond the Horizon," IEEE Security & Privacy, vol. 1, no. 6, 2003, pp. 62–67.

[14] G. McGraw, "Building Secure Software: Better than Protecting Bad Software(Point/Counterpoint with Greg Hoglund)," IEEE Software, vol. 19, no. 6, 2002, pp. 57–59.

[15] Anurag Agarwal, ―Threat modeling enhanced with misusecases, search software quality tech target.comhttp://searchsoftwarequality.techtarget.com/t.html. Aug.2,2008.

[16] Ivan Flechais, M. Angela Sasse, Stephen M. V. Hailes, "Bringing Security Home: A process for developing secure and usable systems",NSPW '03, Ascona, Switzerland, ACM, August 18-21, 2003.

[17] John Wilander, Jens Gustavsson, "Security Requirements - A Field Study of Current Practice", Symposium on Requirements Engineering for Information Security (SREIS'05), In conjunction with RE 05 - 13th IEEE International Requirements Engineering Conference, Paris, France, August 29th, 2005.

[18] Proceedings of the International MultiConference of Engineers and Computer Scientists 2008 Vol I IMECS 2008, 19-21 March, 2008, Hong Kong [19] Ian Alexander, "Misuse Cases: Use Cases with Hostile Intent", Journal of IEEE Software, Published by the IEEE Computer Society, 2003.

[20] I. Alexander, "Misuse Cases Help to Elicit Non Functional Requirements", Computing & Control Engineering Journal, vol. 14, no.1, pp. 40-45, Feb. 2003. [21] I. Alexander, "Modeling the Interplay of Conflicting Goals with Use and Misuse Cases", In Proceedings of 8th International Workshop on Requirements Engineering: Foundation for Software, September 2002.

[22] I. Alexander, "Initial Industrial Experience of Misuse Cases",Proceedings of IEEE Joint International Requirements EngineeringConference, pp. 61-68, 2002.

[23] G. Sindre, A. L. Opdahl, "Eliciting security requirements with misusecases", Requirements Eng, 10(1):34–44, 2005.

[24] G. Sindre, A.L. Opdahl, "Templates for Misuse Case Description", InProceedings of the 7th International Workshop on Requirements Engineering, Foundation for Software Quality (REFSQ'2001), June 2001.

[25] Application security program" Secure Coding: Building Security into the Software Development Life Cycle" by Russell L. Jones and Abhinav Rastogi information systems security N O V E M B E R / D E C E M B E R 2 0 04.

[26] Tim Grance, Joan Hash, and Marc Stevens, "Security Consideration in the Information System Development Life Cycle," NIST, October 2003.

[27] Software Security: Building Security In Editor: Gary McGraw, gem@cigital.com, PUBLISHED BY THE IEEE COMPUTER SOCIETY, 1540- 7993/04/$20.00 © 2004 IEEE

[28] Threat Modeling and Security Pattern used in Design Phase of Secure Software Development life Cycle: International Journal of Advanced Research in Computer Science and Software Engineering Volume 2, Issue 4, April 2012 by Mr. Swapnesh Taterh, Prof (Dr.) K.P Yadav Prof (Dr.) S.K Sharma.

[29] "Static analysis at the end of the SDLC doesn't work" by Wayne Ariola, SearchSoftwareQuality.com, September 22, 2008 [30] Uncertainty & Risk Analysis by Chris Rodger and Jason Petch BUSINESS DYNAMICS April 1999

[31] Security Requirements Engineering; State of the Art and Research Challenges by M. A. Hadavi, V. S. Hamishagi, H. M. Sangchi. Proceedings of the International MultiConference of Engineers and Computer Scientists 2008 Vol I IMECS 2008, 19-21 March, 2008, Hong Kong

[32] Security Requirements Engineering**:** Robert CrookDarrel Ince Luncheng Lin Bashar Nuseibeh Security Requirements Group Department of Computing,
The Open University Walton Hall, Milton Keynes, MK7 6AA, UK

[33] "Importance of Testing in Software Development Life Cycle" by T.rajani devi. International Journal of Scientific & Engineering Research Volume 3, Issue 5, May-2012 1 ISSN 2229-5518

[34] Wright, Joe; Jim Harmening (2009). "15". Computer and Information Security Handbook. Morgan Kaufmann Publications. Elsevier Inc. p. 257 [35] Information Security Forum's Standard of Good Practice for Information Security

[36] Sample Report Security Test Plan Prepared by Security Innovation