A Systematic Security Approach in Software Requirements Engineering

[1]P.Mahizharuvi,
Research Scholar,
Dept of MCA, Computer Center,
Madurai Kamaraj University, Madurai.

[2] Dr.K.Alagarsamy,
Associate Professor
Dept of MCA, Computer Center,
Madurai Kamaraj University, Madurai.

ABSTRACT:

Many software organizations today are confronted with challenge of building secure software systems. Traditional software engineering principles place little emphasis on security. These principles tend to tread security as one of a long list of quality factors that are expected from all professionally developed software. As software systems of today have a wide reach, security has become a more important factor than ever in the history of software engineering can no longer be treated as Separate Island. There is an imperative necessity to incorporative security into software engineering. Incorporating security into software engineering necessitates modification of existing software engineering principles, as these have to be tailored to take into account the security aspect. All phases of software engineering are likely to be impacted Requirement engineering has always occupied a primal position in software engineering. "If you get the requirement correct, you are very close to getting the software correct", has been an accepted fact. Many principles and techniques have been proposed for efficient requirement gathering and these have been validated and applied in practice. Incorporation of security into requirements engineering present several challenges and opportunities for researches. Security requirements posses certain unique characteristics that prevent them from being treated par with other normal functional requirement .for instance the customer may not be aware of the security aspect and may think that the software system developed is what he needs even though it is not secure. The lack of security in the developed system is not as apparent as failures related to performance and reliability.

The proposed research aim at the establishment of sound principles and technique for security requirements engineering. The research is expected to be of great help to organization in their endeavor of building secure software system. The research will try to address the challenge in several ways which may include but not restricted to modification and enhancement of existing requirements engineering principles and models and creation of new ones.

I.INTRODUCTION:

Software security failures are available common in different number of software. Every software contains the design pattern. In every design pattern implementation security requirements are major role. Software designers sometimes it may chance to missing the security requirements in software design. Some kind of security issues are generated here. Previously some controlling techniques are available. Those techniques are normal general techniques here.

Now we propose new security constraints with recommendations environment. Here calculate the security value for each and every security requirements. In software design gives the preferences to high security value. All high security values are place into new software design patterns.

II. Related work:

Security Requirements are related to non functional requirements. Quality requirements are satisfied with good performance, cost and usability. These quality requirements are not show the performance efficiency then we are considering the some new characteristics. Those new characteristics are security requirements. Security requirements are controls and prevent vulnerabilities and attackers. Prevention techniques related security requirements are increases the quality measurement. Security requirements are one part in quality requirements specification.

First approach security requirements works based on non functional requirements. Non Functional requirements control the attackers with yes or no conditions. Its follow the number of rules are less. Sometimes non functional related security requirements are failure and its very expensive cost. It's related to binary classification methodology. All requirements are test with fewer conditions. It's not possible to control all attackers in implementation and gives the results as security failures.

Another approach we are uses as a security analysis approach is spread sheet. Collect the different values and store into rows and columns. In rows and columns of spreadsheet data content apply the some formulas and do the statistical analysis. Statistical analysis provides the some range values of content. Range values are identifying the approximate location areas of attacks only. In approximate location ranges it performs the cryptography and encryption operations here. It does provide somewhat confidentiality in implementation of software requirements.

In compilation time also some security problems it may chance generate here. Those security problems it may be available implicit or explicit. Many types of security requirements are available in market. Users are verifying each and every security requirement, which requirements users are prefer calculate trusted values. These trusted values identifies with the help of analyst. These trusted values are works as a behavior analysis of each and every requirement. In all types of requirements which one is best requirement that we are considering in requirements specification. These requirements are provide the quality solution. These trusted values security requirements we are going to implement in implicit and explicit locations. In different locations also we got the quality solution.

After some days security related requirements are gets the good importance in functional requirements also. Every time change the security requirements based on analysis. This work prefers as a refinement analysis. Some new security issues are generated here and automatically we are providing the recovery solution in software development process in requirements. Recovery solution is change automatically for each and every security issue in implementation part. In same software development process some safety problems are generated here in implementation. Those safety problems also we are control in implementation part.

III. Problem Statement:

Using security requirements control the some number of attackers; some extra attackers are not control in implementation of present software development life cycle models. Some kinds of risks are generating in execution of software. Here we are apply the risk analysis and refine the software with new security requirements in implementation. After adding the new security requirements increases the performance and measurement values in software execution. Here we release life time product with the help of security software requirements. In every software development life cycle we are consider the security requirements that its gives the good performance and quality. Quality software reduces the cost utilization and time facilities. We are provides many number of benefits with the help of refinement security requirements. In refinement software every time considers the new operations and functionality in implementation. New operations every time modify in each and every stage implementation. Every stage adds new security requirements in prototype. In every stage first gives the importance to security requirements.

Many number of security requirements we show to users. We will give the opportunity to select the best security requirements. Every security requirement how many number of users are used we are find out. We learn the feedback as recommendations. Those security related recommendations we will show to users. New users are takes the decision making very easily here. This is the security requirements extension process. Using these security related requirements extension gives the goo solution compare to previous security requirements.

IV.Design to collection of new security requirements: Version 2.0 related security requirements:

In software development we collect two types of security requirements and rules. Two types of security related to security requirements we are implementing or configure in software here. Using these new configurations of requirements increases integrity, confidentiality and reliability.

4.1Introduction:

The above all security problems of solutions arranges into one security frame. In security frame we add the some authentication and authorization rules. Those all rules are control the misuse things.

Every rule of recommendations and feedbacks we collect here. Every rule of reputations calculates here. In total number of rules which rule is the trust rule configure in software implementation. Those trust rules are controls all types of security issues and provide the error messages. Trust values give the awareness for users in selection of good security related requirements.

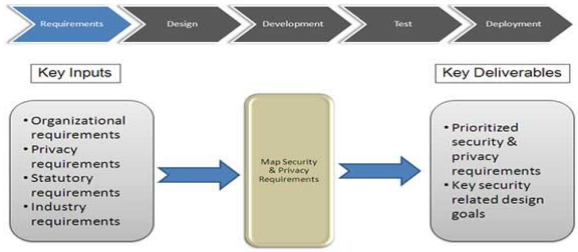

Fig1: Different Phases of security requirements Collection

4.2Version 2.0 security requirements methodology:

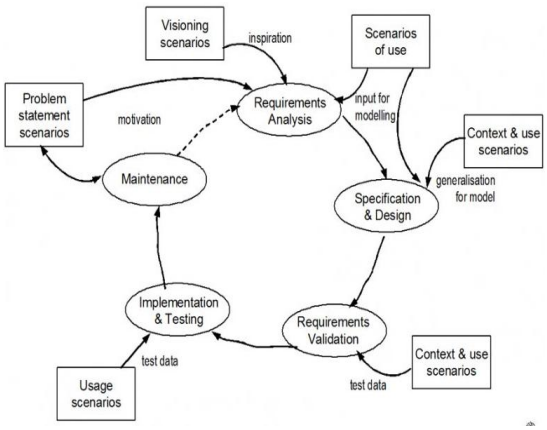

Fig2: Overall architecture complete life cycle

4.3Version 2.0 Security Requirements Software:

Aspiration learning model related security requirements we start to collect from requirements phase. After each and every phase test the security requirements. Those security requirements are gives the proper solutions. Those security requirements are valid requirements for implementation. Different types of security requirements we add in each and every phase and learn the features of each and every rule or requirement.

Version 2.0 follows classification. Classification we implement two times for improve the results. First phase related training and second phase related to testing. In first phase we train using each and every security requirement and find out how it's works. Every requirement consists of some specific characteristics. Several times perform the training process and calculate the experimental values in implementation process. These experimental values based policies or rules are configuring into software as a security requirements and test each and every security requirement. We check the performance of each and every security requirement in implementation.

Different security related requirements are combines and directly form as a artifact for implementation of one good software. Good software implementation different numbers of actors are participated here. Those actors are instructor, analyst and security controller.

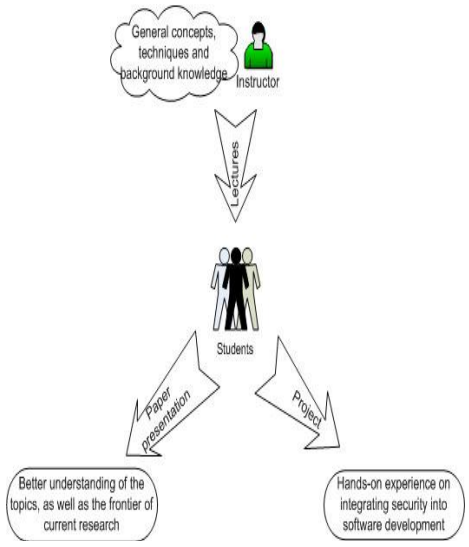

Fig3: Adding of security requirements with the help of instructor

Instructor passes the security requirements related to software development. Students are applying the refinement security requirements in software. In development of software gives the preference of normal and security related requirements in implementation. This is the integrated software related to different requirements.

Any application after execution its shows the good quality and performance.

V. Detection of different flaws with security requirements:

We have good appropriate design and appropriate decisions for detection of all types of security flaws and security bugs of content.

5.1 Security Flaws Detection:

In implementation time developers are design with correct authentication mechanism and rules. All authentication mechanisms and rules are working correct its possible gives the good efficiency in execution of application. All security flaws also we control efficiently.

5.2Bugs controller:

Complete high level codes provide the good security. In code level adding the some rules for controls the attackers. Here we use the high level language in implementation of security requirements. High level languages consider the verification and validation techniques.

5.3 Commendations:

Commendations also provide the good security in controlling of attacks**.** Practice of different attacks with commendations. In total number of commendations filter the best commendation. This commendation rules we inform to all users. Calculate the security value of each and every commendation. Using security value selects the best commendation.

5.4 Recommendations:

High security value related requirement is useful in future session's environment. High security value requirement it may chance to improve the security. These high security values related requirements only add in software design pattern and SDLC life cycle also.

VI. Performance Evolution:

In Different phases identifies the good recommendation based requirement. Recommendation based security value already we calculate here. Here selected high security value related requirement.

Phase / Concept	Requirements and Analysis Phase		Architecture and Design Phase		Implementation Phase	
Countermeasure	Identified		Feasibility	++	Implemented	++
Risk	Identified	+	Estimated		Measured	++
Threat	Identified	+	Feasibility		Tested	+
Attack	Identified	++	Feasibility		Tested	++
Attacker	Identified	++	Feasibility		Tested	+
Vulnerability	Identified	+	Feasibility		Tested	++
Asset	Defined		Designed with security		Implemented with security	
Stakeholder	Defined	+	Reviews		Tests	
Security objective	Defined	+	Reviewed	+	Verified	

Fig4: Recommendation based high security value requirement selection

Every phase contains different security requirements, in total requirements select best requirements based on security value practices.

VII. Conclusion:

Previously software design pattern are created with normal security requirements. Its gave the quality and security and performance is low. Now here we select the security requirements based on recommendations. High security value related requirements we select in software design pattern. These new design patterns show the good performance and improvement in implementation.

VIII.References:

1. Measuring The Software Security Requirements Engineering Process, 2012
2. An Effective Requirement Engineering Process Model for Software Development and Requirements Management,2010
3. Security Requirements Engineering: A Framework for Representation and Analysis, 2008
4. A common criteria based security requirements engineering process for the development of secure information systems,2009
5. Security Requirements Engineering; State of the Art and Research Challenges, 2008
6. Research Directions in Requirements Engineering, 2009
7. Security and Privacy Requirements Analysis within a Social Setting, 2007
8. Cutting Edge Practices for Secure Software Engineering,2007
9. A Survey on Security Patterns,2008
10. Software Security - The Bigger Picture,2008