Security Modeling for Web Based Visitor's Login System for Pursuance of Security Design Pattern

Yogini C. Kulkarni, S. D. Joshi

Abstract: In recent years because of the widespread use of internet and other communication media security occurrences have broken all the barricades. System gets attacked by malicious attackers and various cyber criminalities. Every system should be built by taking security as a main priority while building a system so as to make it reliable, safety and also it should be enhanced with other quality parameters. Hence since beginning at every phase of software development till the implementation of the software, security aspect is needed to take into consideration before making the final design decision to avoid the expenses which may incur while recovering of the system after the damage. For attainment of this, it is must to integrate the security at each phase of the software development. The software developers insists on incorporating the software safeguards at the design phase which may wind up in identifying the architecture restrictions related with the security which in fact may not be necessary. To reduce this problem, this paper intends a structure for security development activities. These activities consist of security requirements identification and threats analysis which are to be converted into design decisions to lessen the risks to identified important assets. The recognized design parameters are then manually prioritized using VOSREP and CRAMM and accordingly Security design pattern is to be developed to incorporate security in the software. By manually calculating values of assets and prioritizing will help to identify the security requirements at the early stage of the software development life cycle. Accordingly the decisions for developing the security design pattern are to be taken for building the software system

Keywords: Security Engineering process, Security Requirement Elicitation, Security modeling

I. INTRODUCTION

Generally, software is designed and developed without thought process of security issue, the non-functional requirement of security parameter being in the minds of the developers [4]. The issue of security is taken into consideration after creating the entire software which results in leaving behind potential loopholes which may be origin to create vulnerable attacks for hackers and crackers. The main reason of increasing attacks is the poor quality of the software system and many loopholes left unknowingly while developing the software through which the attackers can attack onto the system.

As attackers are more inventive by creating more complex attacks, a systematic approach is required for maintaining the security of software for development of secure software. Hence it is required the assessment of the security requirement at the early stage of the development of the software. The main objective for identifying the security requirements is to put in order the elicited security requirements and to take the decisions for implementing the security to reduce the various types of vulnerabilities related with the threats. The modeling techniques like generating the misuse case, abuse case, attack trees for the security concepts like threats, attacks and vulnerabilities together can be termed as security modeling. These techniques explore the concepts of security issues so they can be handled throughout the software development lifecycle. This paper describes security modeling as a method with respect to software development. The following modeling techniques are described.

- Assets Identification using VOSREP and CRAMM
- Risk Identification using STRIDE and DREAD
- Threat modelling.

II. RELATED WORK

In software engineering, the securities and its requirements must be identified along with the functional as well as non-functional requirements. The main priority should be given to the security requirement which is to be recognized at the earliest phase to avoid the future recurring costs due to vulnerability which occurs because of poor design of the software. These requirements should be systematically analyzed based on the risk measurement techniques. Security requirements should be accurate, sufficient and complete. Once these requirements are clearly précised, those can then be implemented and maintained throughout the life cycle development of the software [1] [2]. Manually we are prioritizing the threats and risks associated with the web based login system. The following methodologies are involved.

1. STRIDE which provides checklist for recognized threats. [1]
2. DREAD which provides a rating to the recognized threats identified [1],
3. CRAMM which calculate the measure of risk for each threat to an asset and vulnerability [3]

III. FRAMEWORK FOR SECURITY DEVELOPMENT AT SDLC

The fig.1 shows the framework for engineering activities involved in a secure software development which are –

(i) **Security requirements analysis** : It includes security requirement elicitation,

Revised Manuscript Received on November 30, 2019.
* Correspondence Author

Yogini C. Kulkarni*, Information Technology Department , Bharati Vidyapeeth (Deemed To Be University) Pune,India.Email: yckulkarni@bvucoep.edu.in

Prof. Dr. S .D. Joshi , Computer Engineering Department ' Bharati Vidyapeeth (Deemed To Be University) Pune,India.Email: sdj@live.in

analysis & prioritization and identification of the security design pattern

(ii) **Design phase/Architecture phase:** After the completion of security requirement elicitation regarding the functional as well as non-functional requirements one needs to take the decision for designing of the software. According to systems development viewpoint security modelling will be required for implementing the security design pattern.

In case of the wrong design decisions, it may create redundant design constraints or makes the system vulnerable and ultimately increasing of the budget of software development.

(iii) Implementation Phase: This phase includes the coding for incorporating security, code validation and testing of the security pattern.

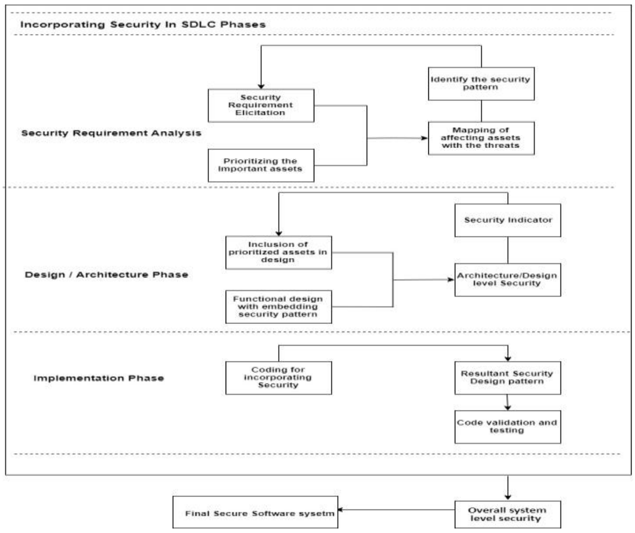

Incorporating Security In SDLC Phases
Security Requirement Analysis
Security Requirement Elicitation
Prioritizing the important assets
Identify the security pattern
Mapping of affecting assets with the threats
Design / Architecture Phase
Inclusion of prioritized assets in design
Functional design with embedding security pattern
Security Indicator
Architecture/Design level Security
Implementation Phase
Coding for incorporating Security
Resultant Security Design pattern
Code validation and testing
Final Secure Software sysetm
Overall system level security

Fig.1 Software Security Assessment for incorporating security in SDLC [14]

Once the security requirements are elicited and prioritized, suitable design decision will consider the security threats for developing the security design pattern. The approach involves conversion of identified security requirements and threats into design decisions to reduce the security threats. Mapping of the security requirements with the different security services is done for identifying the security practice to incorporate into the life cycle of the software. The recognized design attributes are hierarchies and on the basis of the set of important security attributes a security design pattern is developed at every stage of the software development. At the end the specific cryptographic techniques is to be finalized for reducing the unnecessary constraints in the software development process. We exemplify our process with suitable case study of Web based visitor's Login system.

IV. SECURITY MODELING FOR INCORPORATING THE SECURITY IN SDLC

There are mainly four key stages for the embedding of the security in software development life cycle as follows.

1. Security Requirement Engineering
2. Architecture and Design Level
3. Code level security
4. System Level Security

The activities involved in the above process are shown in the fig.2

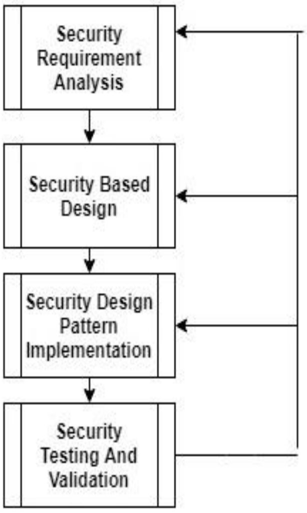

Security Requirement Analysis
Security Based Design
Security Design Pattern Implementation
Security Testing And Validation

Fig.2 Outline of the incorporating Security design pattern in SDLC

A. Security Requirement Analysis:

The security requirements are to be finalized at the premature stage of the software development of SDLC. In this phase first we will discover all the security requirements related to our system. (Web Based Secure Login System). The security requirements are then analyzed.

a. Security Requirement Elicitation: Security requirement can be defined as systems provisions for its security such as necessities made for maintaining the security through access control mechanism. The access control protects the access by illegal users and helps to secure data and application of the system. The fundamental cause of security requirements is the reduction of different types of hazards related with the threats and suggests on extracting of security requirements like abuse case [5], misuse case [6, 7], common criteria and attack trees [8].

The requirement elicitation is the first phase of SDLC and considered as the process of investigating the real needs of the customer and the system. The eliciting the security requirement phase is complicated and requires security experts for eliciting the requirements [9]. It also consists of the activities to discover 'how the software can meet the customer's requirements' and what alternative might exist for the same requirement. By comparing the existing systems, the requirements are to be finalized to overcome the limitations of the existing one.

While designing the software, in the security requirement elicitation phase the focus is to be given on the elicitation of the security requirements. The importance for providing the correct information is given highest priority. In this phase the data is gathered for the security requirements through various models and from various persons like for e.g. the customers, manager, software coders, software designers and quality assurance team.

Constructing accurate requirement models means to guarantee of correctly collecting the requirements.

The requirement models help the software designers to specify the functions as well as the data.

The security requirement elicitation is done with reference to the method involved in VOSREP- Viewpoint Oriented Security Requirement Engineering Process (VOSREP) is defined as process making the security engineering as the incorporated approach with requirement engineering. VOSREP is process to elicit, analyze, prioritize and manage requirements. It takes into consideration both the functional requirements as well as nonfunctional requirements while development of the software. The different activities in VOSREP are as follows.

1. Security Requirement Discovery and Definition

It is the first activity of the VOSREP process the security requirements with functional and nonfunctional requirements are identified and defined for the system to be developed.

2. Analysis and Prioritization

In the second step, the recognized functional and nonfunctional requirements are analyzed for their feasibility, consistency and completeness. Once those are finalized these security requirements are prioritized based on the measure of the risk of threat on the important assets.

3. Management of the Security Requirements

Once the security requirements are recognized, analyzed and prioritized those are maintained throughout of the lifecycle of the software development.

The table of requirement specification is s shown in table .These specifications are generated using VOSREP[10].

b. Security Requirement Analysis (Use of assembling threats table)

The tasks to be performed in analyzing the security requirements are as follows: -

(i) **Verifying for comprehensiveness**- Check list is to be generated to verify whether the extracted security requirements have reduced all the threats of the system. This check list is used to resolve the contradictions that may exist in the security requirements elicited from different viewpoints.

(ii) **Assembling of security requirements** - This step consists of recognizing and grouping together the security requirements. According to our system different tasks are performed in security requirement analysis phase as shown in following flowchart.

Table-I: An Example Of Web Based Visitor Login System [10]

Viewpoints	Services	Nonfunctional requirements	Threats	Threats identified	Security Requirements
Visitor	1.Registration of the visitor 2.Registration Information 3.Registration	1.Reliability 2.Response Time 3.Execution of the key generation	1.Spoofing 2.Flooding 3.Disclose login information 4.Impersonation 5.Change password 6.Repudiation	T1. Impersonate T2.Visitor Data_Theft T3.VisitorDisclose_Data T4.Securitycode_Violated T5.VisitorChange_Data T6. Visitorpass_steal T7. Denial_of_service	1.Authorization Requirement 2.Privacy requirement 3.Non repudiation requirement
Visitor Record Management	1.Update the visitor's data 2.visitor's registration Query Process 3.Varification of generated security code 4.Visitor's pass Generation 5.Maintainance of the visitor's database	1.Correctness 2.Minimum response time 3.Scalable	1.Change login information 2.Impersonation 3.Outsider	T8.Unauthorize_access T2. VisitorData_Theft T4. securitycode_Violated T9. System_Failure T10. Outsider	1.Integrity Requirement 2.Authentication 3.Identification requirement

Database	1.Maintain the registration numbers	1.Optimized	1.Outsider 2.Integrtiy	T11.Copy T12.Replace T5 .VisitorChange_Data	1.Security review requirement 2.Intrusuion detection requirement

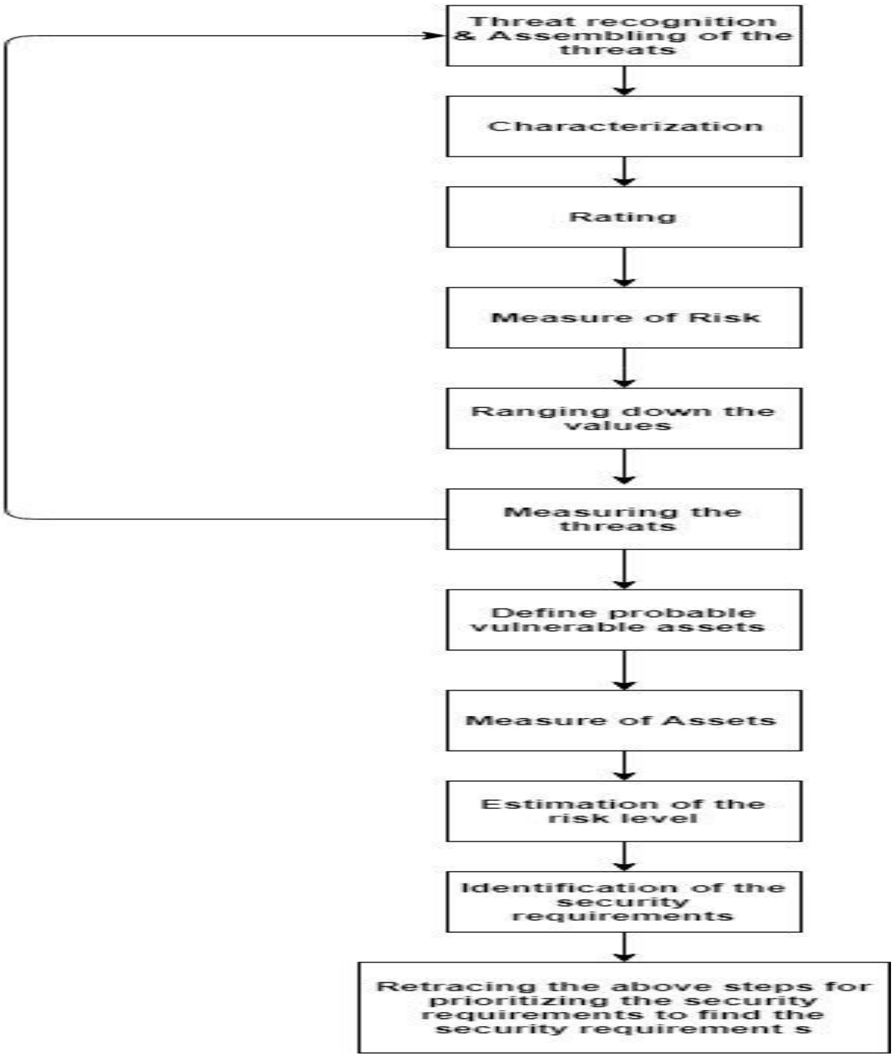

Threat recognition & Assembling of the threats
Characterization
Rating
Measure of Risk
Ranging down the values
Measuring the threats
Define probable vulnerable assets
Measure of Assets
Estimation of the risk level
Identification of the security requirements
Retracing the above steps for prioritizing the security requirements to find the security requirement s

Fig.3 Flowchart showing different tasks in the security modeling

The above steps are explained according to our implementation as follows.

i. Threat Modeling for Recognition and Assembling of Threats [11]

Threat modeling is recursive process which starts during the architecture/design phase. The design phase explains the progression of threat modeling through the phases of SDLC to able the analyst to add more details in the threat model via repetitive threat model process. By using common criteria approach the threats are assembled together as follows.

1. Impersonate(T1)
2. Visitordata_Theft(T2)
3. Visitordisclose_Data(T3)
4.Security Code violated(T4)
5.VisitorChange_Data(T5)
6. Visitorpass_steal(T6)
7. Denial_of_service(T7)
8. Unauthorize_access(T8)
9. System_Failure(T9)
10. Outsider(T10)
11.Copy(T11)
12.Replace(T12)

ii. Characterizing of the discovered threats

The threats are recognized, characterized using STRIDE model and assembled as follows.

1. Spoofing
2. Tampering of the data
3. Repudiate and receive (To refuse the authorization)
4. Information disclosure
5. Elevation of privileges

Table-II: Recognition and Assembling Of Threats

Threat Name	Threats	Function	Property	Example	Vulnerabilities
Spoofing	T1, T2, T3, T4	Pretending to be someone else while using system	Authentication	Hacking of victim's email and use to send message	Authentication error
Tampering Data(TD)	T1, T5, T6, T8	Change data or code	Integrity	Executable file gets hampered by an attacker	Manipulation of the data
Repudiate to receive (RR)	T8, T12	Claiming not to do the particular action regarding the authorization	Non repudiation	Disagreeing of sent mail to alike	Insufficient verification of the data
Information disclosure (ID)	T2, T3, T4	Leakage of sensitive information	Confidentiality	The sensitive information like credit card information available on internet.	Information management error
Denial of service (DS) (Buffer Overflow)	T9, T10	Non availability of service	Availability	Web application not responding to user's request	Buffer overflow
Elevation of privileges(EP)	T8	User can perform unauthorized action	Authorization	The user changes the admin account	Authorization error

The STRIDE model is used for the tabular representation of checklist for recognized threats are as shown below. STRIDE model helps to identify and categorize threats and vulnerabilities.

Table-III: Checklist for Discovered Threats

	S	T	R	I	D	E
T1	√	√				
T2	√			√		
T3	√			√		
T4	√			√		
T5		√				
T6		√				
T7						
T8		√	√			√
T9					√	
T10					√	
T11						
T12			√			

iii. Rating Threats

There are various methods for analyzing risks like EBIOS, OCTAVE, CORAS. We have used CRAMM and DREAD for assessing the risks associated with the threats. The brief description about CRAMM and DREAD is given below.

CRAMM: It is qualitative risk assessment tool and is to be applied for IT security. It is developed by UK governments Central Computer and Telecommunications' Agency in 1985 .The following diagram shows the processes involved in CRAMM.

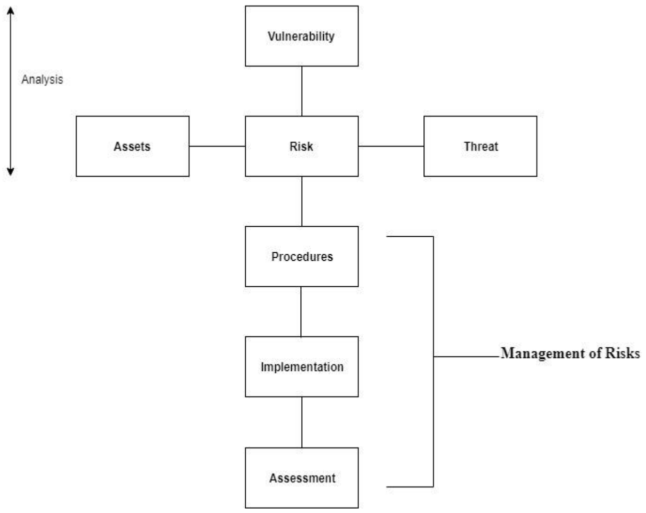


Fig .4 CRAMM Processes [12]

DREAD: DREAD is a structure which is used to assess different threats by rating .This structure consists of five main categories: Damage, Reproducibility, Exploitability, Affected Users, and Discoverability

D-Damage- It makes reference to what level the assets are affected by the threats. R-**Reproducibility-** It gives the possibility of recursively occurrence of an attack.

E-Exploitability- It is sequence of commands which takes benefits of vulnerability to raise unexpected behavior on computer software.

A-Affected User- The influence of an attack on people

D- Discoverability – Ease of discoverability of the threat

The rating is assigned to the threats using CRAMM and DREAD method as follows.

Table-IV: Risk Measurement

Threat ID	**D**	**R**	**E**	**A**	**D**	**Overall Risk**
T1	5	5	5	10	5	6
T2	10	5	5	10	5	7
T3	5	5	5	0	0	3
T4	5	5	5	5	5	5
T5	0	0	5	0	5	2
T6	5	5	5	5	10	7
T7	0	10	5	0	5	4
T8	5	5	5	10	5	6
T9	0	0	5	0	5	2
T10	10	0	5	10	10	7
T11	0	0	5	5	5	3
T12	0	0	5	5	5	3

iv. Values

After the risk rating procedure, the risk measurement table the 10-point rating is to be scaled down to two-point scaling as shown below. The ranging is done as follows[1]

Table-V : Scaling Down The Risk Rating To Values

DREAD TEN-point Scale	DREAD TWO-point Scale	Values
1-2	Very Low	0.1
3-4	Low	0.3
5-6	Medium	0.5
7-8	High	0.7
9-10	Very High	0.9

The recognized threats have to assign a value according to CRAMM evaluation.

v. Measuring of the threats

After the valuation the recognized threats are to be assigned values to identify the risks associated with the threats.

Table-VI: Risk Assessment

Threat	**Affected Assets**	**Values**
T1(6)	Visitor information()	6
T2(7)	Communication Channel	4
T3(3)	Registration Info()	5
T4(5)	VisitorPass Info()	5
T5(2)	Registration Info()	5
T6(4)	Security Code Steal()	8
T7(7)	Registration Info()	8
T8(6)	Registrationcopy()	5
T9(2)	Systemfailure()	7
T10(7)	Communication channel	4
T11(3)	VisitorInfo()	4
T12(3)	VisitorInfo()	4

vi. Assigning values to affected assets

Table-VII: Values To The Assets

Threats with overall risk	**Level of threat**	**value**
T1(6)	Medium	0.5
T2(7)	High	0.7
T3(3)	Low	0.3
T4(5)	Medium	0.5
T5(2)	Very Low	0.1

T6(4)	Low	0.3
T7(7)	High	0.7
T8(6)	Medium	0.5
T9(2)	Very Low	0.1
T10(7)	High	0.7
T11(3)	Low	0.3
T12(3)	Low	0.3

Table-VIII: Probable vulnerable threats

Threats with overall risk	Level of Threat	Threat Value (1-10)	Asset Value(1-10)
T1(6)	Medium	0.5	6
T2(7)	High	0.7	4
T3(3)	Low	0.3	5
T4(5)	Medium	0.5	5
T5(2)	Very Low	0.1	5
T6(4)	Low	0.3	8
T7(7)	High	0.7	8
T8(6)	Medium	0.5	5
T9(2)	Very Low	0.1	7
T10(7)	High	0.7	4
T11(3)	Low	0.3	4
T12(3)	Low	0.3	4

With reference to above table we define the probable vulnerable threats as follows.

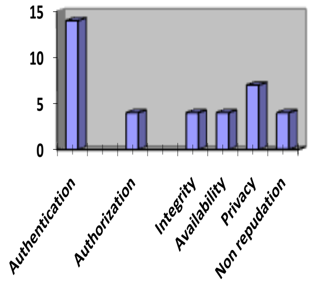
vii. Approximation the Risk Level and Prioritizing the important assets

Based on the values generated in above table the detailed computed values of security requirements for secure login system are as shown table 9 [13]. After probable vulnerable threats rating we calculate the risk level value along with the values of SR prioritize values, we use the 3 dimensional lookup table given by the CRAMM where the strength of the threat, the level of the vulnerability and the value of the asset are input parameters, gives the final value of risk in the range 1 through 7 .The vulnerability values are assigned by CRAMM method,i.e low as 0.1,medium as 0.5 and high as 1 [1] For e.g. suppose asset is Visitors Information (7) the threat to this is Spoofing (0.9) and Vulnerability being medium (0.5). The measure of risk will be (Risk=threat value*vulnerability asset +value/2) for e.g (0.9*0.5+7/2=4) In the similar fashion we can calculate the approximated values of risk and security requirement prioritize of important assets which is given in table 9.

Table-IX: Prioritization table [1]

Security requirement	Threat	Threat rating	Vulnerability	Assets affected	Asset value	Risk	SR prioritize
Authentication	T1	0.9	0.5	Visitor information()	7	4	14
	T2	0.7	0.1	Communication Channel	6	3	
	T3	0.3	0.1	Registration Info()	9	5	
	T4	0.9	0.5	VisitorPass Info()	4	2	
Authorization	T11	0.3	0.1	Registration Info()	4	2	4
	T5	0.1	0.1	VisitorPass Info()	5	2	
Integrity	T5	0.1	0.1	Registration Info()	4	2	4
				Visitor Info()	5	2	
Non-repudiation	T6	0.3	0.1	Security Code Steal()	8	4	4

Privacy	T2 T6 T8	0.7 0.3	0.1 0.1	Registration Info() VisitorPass Info() Registrationcopy()	4 5 2	2 2 1	7
Availability	T7 T9	0.1	0.1	Registration Info() Systemfailure()	4 5	2 2	4



DESIGN DECISIONS

For incorporating the security in software development life cycle the design decisions are to be taken based on the available cryptographic practices. Based on the above process of identification of important assets and risks associated with those encryption speed is considered as design constraint as shown in the following table[13]. This design decision table gives useful data of different public key algorithm which helps to choose a particular cryptographic technique in a particular situation. Final decision template shows that SHA-I algorithm (160 bit key) is the most suitable technique which fulfils all the security requirements for web based secure login system.

After recognizing the security requirements and design constraint, the final decision for the selection of suitable cryptographic technique is to be taken for developing security design pattern.

CONCLUSION

This paper presents the security model for the recognition of security activities during the software development. To develop secure software, the main focus must be given on security at every phase of the software development lifecycle to lessen the security flaws. The identification of the security requirement helps the software developer to maintain the thread of security. The most important assets are to be identified for which the security is to be maintained by calculating the risk values. For recognizing the suitable cryptographic technique the security services are mapped with the identified threats to develop security design pattern for web based secure login system. This is to be obtained by working out on the security activities which recognize and remove security flaws at all the stages of software development.

Table-X: Design Decisions

Platform	Design Constraints	Design Attributes	Priorities			Security Mechanism
			High	Medium	Low	
LAN	Encryption	Throughput	√			Public Key Encryption
WLAN √	Speed	Storage		√		
MANET	Channel Capacity	Target platform	√			Public Key Encryption with signature
WMN	Frequency	Security				
WSN	Bandwidth	Scalability	√			
		Interoperability	√			Digital Signature
		Power consumption			√	
		cost			√	Hash Based Authentication √

REFERENCES

1. Aayush Gulati , Shalini Sharma, and Parshotam MMehmi. 2012.PROPOSING SECURITY REQUIREMENT PRIORITIZATION FRAMEWORK., International Journal of Computer Science, Engineering and Applications (IJCSEA) Vol.2, No.3
2. Donald G. Firesmith, (2003) "Engineering Security Requirements", Journal of object technology, vol 2, no.1, pp.53-68.
3. Zeki Yazar. A Qualitative Risk Analysis and Management Tool – CRAMM. SANS Institute Reading Room site
4. Ramchandran., 2002 .Designing Security solutions,Wiley Computer Publishing
5. Dermott JM and Fox C (1999) Using abuse case models for security requirements analysis. Department of Computer Science, James Madison University, pp 55–64
6. Alexander IF (2003) Misuse cases, use cases with hostile intent''. IEEE Software, pp. 58–66
7. Guttorm S, Opdahl AL (2005) Eliciting security requirements with misuse cases. RequirEng 10:34–44
8. Jacob Bergvalloch Louise Svensson .2015.Report on 'Risk analysis review' Linköpingsuniversitet https://www.diva-portal.org/smash/get/diva2:821842/FULLTEXT01.pdf
9. NURIDAWATI MUSTAFA,MASSILA KAMALRUDIN,SAFIAH SIDEK. 2018. SECURITY REQUIREMENTS ELICITATION AND CONSISTENCY VALIDATION: A SYSTEMATIC LITERATURE REVIEW ., Journal of Theoretical and Applied Information 96(16).ISSN: 1992-8645
10. Ashish Agarwal, Dr. Daya Gupta. Security Requirements Elicitation Using View Points for Online System First International Conference on Emerging Trends in Engineering and Technology
11. Security Requirements, http://www.dwheeler.com/secure-programs/Secure-ProgramsHOWTO/requirements.html
12. Mohamed Ghazouani, Sophia Faris, Hicham Medromi, Adil Sayouti.2014. Information Security Risk Assessment — A Practical Approach with a Mathematical Formulation of Risk ., International Journal of Computer Applications (0975 – 8887) Volume 103(8),
13. Kakali Chatterjee ,Daya Gupta ,Asok De. 2013 .A framework for development of secure software.,CSIT., 1(2):143–157
14. Shams Tabrez Siddiqui.2017. Significance of Security Metrics in Secure Software Development.,International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868 Foundation of Computer Science FCS, New York, USA Volume 12 – No. 6, August 2017 – www.ijais.org 10

AUTHORS PROFILE

Yogini Kulkarni received her B.E. degree in Computer Engineering from Shri Sant Gajanan Maharaj College of Engineering, Shegaon from Amaravati University, Amaravati, Maharashtra in 1990 , M.E. (Computer Engineering) Degree from Bharati Vidyapeeth Deemed University Pune. She is currently working as PG Coordinator and Assistant Professor in Information Technonolgy, Bharati Vidyapeeth (Deemed To Be University) College of Engineering, Pune since 2010. Currently she is pursuing Ph.D.from Shri Jagdishprasad Jhabarmal Tibrewala University, Rajasthan . Her research interests include software engineering. She is a sincere teacher devoted to Education and Learning for the past 27 years. Her research projects include Software Engineering and Image Processing. She will be available at yckulkarni@bvucoep.edu.in for any further communication.

Shashank Joshi received his B.E. degree in Electronics and Telecommunication from Govt. College of Engineering, Pune in 1988, MBA (Systems) from Pune University, the M.E. (Computer Engineering) and Ph. D. (Computer Engineering) Degree from Bharati Vidyapeeth Deemed University Pune. He is currently working as the Dean, Faculty of Engineering and Technology, Bharati Vidyapeeth (Deemed University) Pune and Professor in Computer Engineering Department Bharati Vidyapeeth (Deemed University) College of Engineering, Pune since 1990.

His research interests include software engineering. Presently he is engaged in SDLC and secure software development methodologies. He is innovative teacher devoted to Education and Learning for the last 30 yrs. His research projects include Software Engineering, Object Engineering, Software Project Management, RTC Systems, Software security, Visual and Generic Modeling. He will be available at sdj@live.in for any further communication.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication