Vulnerabilities & Attacks in SRS for Object-Oriented Software Development

Madhuri N. Gedam, Bandu B. Meshram

***Abstract*—One of the most crucial parts of software development is the integration of security in Software Requirement Specification (SRS). The loopholes left out during SRS may lead to serious vulnerabilities resulting into severe attacks causing disruption of service of the software systems. The extensive literature survey of security checklists, Software Requirement Specification, Good Quality Requirement Characteristics, Methodology for Security Specification Languages, Security Quality Requirements Engineering (SQUARE) and Security Requirements Engineering Process (SREP) is made in Software Development Life Cycle (SDLC). On the basis of this extensive literature survey, the secure software SRS model is proposed to mitigate the threats in software development. The defence mechanism for protecting the attacks on software is proposed which may be embedded into SRS.**

***Keywords*—CLASP, Security Checklist, Software Development Life Cycle, SQUARE, SREP.**

I. Introduction

The security requirements in SRS of Software Development Life Cycle (SDLC) are critical to the success of any software project. The major cause of schedule overruns and increased development and sustainment cost is poor quality requirements that are not clearly specified, not synchronised to suit the business logic of software project, partial and haphazard collection of requirements, out-of-scope requirements and obsolete nature of requirements [1]. If security requirements are not considered in requirement engineering and SRS, it may increase the cost of project 10-200 times in the long run [2]. The quality of the software products to be produced will be bad and it will not serve the purpose of development of software if security requirements are not considered appropriately [3]. A Software Requirement Specification (SRS) document is prepared in requirement gathering and analysis phase of SDLC serves as the guidelines for the consequent phase of SDLC. SRS document is prepared from the input received through the statement of the problem. Every SDLC model must follow the six phases like requirement gathering and analysis, design, coding and implementation, testing, deployment and maintenance. Business requirements are gathered in the requirement gathering and analysis phase [3]. The analysis of collection of requirements is done to verify their validity and it is studied whether those can be included in the system or not. Unified Modelling language (UML) is a standard graphical language which helps system and software developers for specifying, visualizing, constructing and to document artefacts of object-oriented software development [4]. This paper is organised as follows: Section 2 discusses related approaches for security requirement Specification elicitation, Methodology for Security Specification Languages. Section 3 presents proposed secure SRS model and Defence mechanism of SRS document. Lastly section 4 concludes the work.

M. N. Gedam is a Research Scholar associated with Veermata Jijabai Technological Institute(VJTI), Mumbai, India (phone: +91-9869281271 e-mail: madhuri.gedam@gmail.com).

B. B. Meshram is a Professor in Department of Computer Engineering and Information Technology, Veermata Jijabai Technological Institute(VJTI), Mumbai, India (e-mail: bbmeshram@vjti.org.in).

II. Literature Survey

This section describes detailed Literature Survey on Software Requirement Specification (SRS), Good Quality Requirement Characteristics, Vulnerability-Based Risk Analysis in SRS, Attacks due to vulnerabilities in SRS Model, Security check list for Proposed Secure SRS Model, Security check list in security specification languages(e.g. UMLsec, UMLintr, SecureUML , SecureTropos), Security Requirements Engineering Processes (e.g. SQUARE, CLASP, SREP).

A. Software Requirement Specification (SRS)

The first phase of the Software Development Life Cycle (SDLC) is Software Requirements Specifications (SRS). SRS is a collection of specified, standardized, and organized requirements surrounding a software development project [18] [19]. SRS document should be prepared in a precise form. The feasibility study should be part of requirement gathering and analysis phase. The requirements should serve the purpose of development of software product i.e. they should fulfil the criterion of good requirement characteristics. The gathering of requirements must include Functional Requirements (FR) to fulfil customer perspective and Non-Functional Requirements (NFR) to provide more security in SRS. FRs are those which are related to the technical functionality of the system and describe interactions between the system and its environment. FRs often neglects quality requirements like performance, safety, security, reliability and maintainability. Implementation of the functional requirements takes place in the system design process. NFRs are related to quality attributes. NFRs are implemented in system architecture process. NFRs are limitations in system design or implementation process. NFRs are generally quality requirements and hence prone to be neglected. Those can be in-feasible, outdated, out-of-sync requirements. Many times NFRs are not testable, they are not capable of getting validated too [2][5]. NFRs define the expected behaviour of the software product or system.

ISBN: 978-988-14048-7-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

B. Good Quality Requirement Characteristics

Donald Firesmith states that "Requirement gathering and analysis phase should consider the quality characteristics that are missing from poorly specified requirements like Cohesiveness, Completeness, Consistency, Correctness, Customer/User Orientation, External Observability, Feasibility, Lack of Ambiguity, Relevance, Usability, Validatability, Verifiability" [7].

C. Vulnerability-Based Risk Analysis in SRS

The priority for software involves calculation of "value of protection" *(VoP)* [8]. The "value of protection" formula represents a classical software security risk assessment framework classified as a "financial loss methodology" [9]. The formula is stated as below [6].

$$VoP = R - M, \text{ or } VoP = (Ap*L) - M$$

where, *VoP* is the value of protection, *R* is the risk, calculated as *(Ap*L)*, *Ap* is the probability of a successful attack, *M* is the cost of mitigation countermeasure. *Ap* can be replaced by probability of vulnerability *P(V)* to realize a given threat targeting a said vulnerability. The modified formula is defined as below.

$$VoP = (P(V)*L) - M$$

where, *P(V)* is the probability of a successful attack on vulnerability *V* [6].
It is clear from the above formulae that threats are ranked by value of protection*(VoP)*. Higher the value of *VoP*, it requires higher priority and more expenditure. It is desired that *VoP* should always be low. Generally, organizations are paying less than required *P(V)*L* for mitigating the risk. The following graph depicts the relationship between the value of protection *(VoP)* and cost of mitigation countermeasure *(M)*.

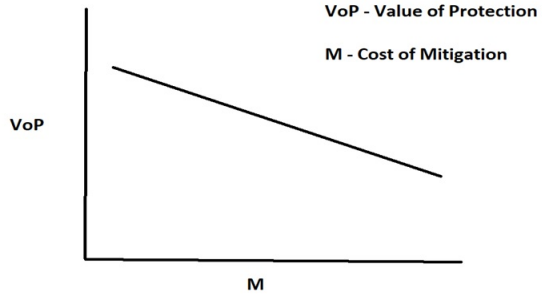

Fig. 1. Relationship between *VoP* and *M*

D. Attacks due to vulnerabilities in SRS Model

The potential vulnerabilities always exist in any software development which needs to be studied at requirement gathering and elicitation phase precisely. The non-anticipation of any vulnerability leads to attacks on developed software product at a later stage. Following are the vulnerabilities and attacks due to non-implementation of Functional Requirements in SRS.

TABLE I
VULNERABILITIES AND ATTACKS IN FUNCTIONAL REQUIREMENTS OF SRS [26][27][28]

SrNo	Functional Requirements	Vulnerabilities	Attacks
1	Business Rules	Non-cooperation & incomplete knowledge of user in ignoring exceptions to the normal organizational operations, Sometimes customers don't know exactly what they need, Communication gap between system analyst and the user, Unavailability of customer, lack of proper knowledge	Security Misconfiguration, Broken Authentication & Session Management, Man-In-Middle attack
2	Administrative functions	Missing strong access control system	Sensitive Data Exposure,Broken Access Control
3	Authentication	Low security provided at the user login system	Broken authentication and session management, Man-In-Middle, DDoS, Buffer overflow, XSS
4	Authorization levels	Roles ambiguity	Sensitive Data Exposure, Data theft
5	Audit Tracking	Clocks not synchronized across the servers ,Inconsistent log format	Insufficient Logging &Monitoring
6	External Interfaces	Incorrect port configuration, unintended access to the system	XML External Entities (XXE), Sensitive Data Exposure
7	Certification Requirements	Invalid or Expired digital certificates	Broken authentication and session management

To improve quality software product, inclusion of non-functional requirements is of much importance in security requirements engineering [3]. Therefore, classification and prioritization of NFRS in SRS document is important area in requirement engineering [9].

Following are the vulnerabilities and attacks due to non-implementation of Non- Functional Requirements in SRS.

TABLE II
VULNERABILITIES AND ATTACKS IN NON-FUNCTIONAL REQUIREMENTS OF SRS [29][30]

SrNo	Non-Functional Requirements	Vulnerabilities	Attacks
1	Availability	Network starvation and resource starvation	Buffer overflow, Sync flood, ICMP flood

ISBN: 978-988-14048-7-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

SrNo	Non-Functional Requirements	Vulnerabilities	Attacks
2	Integrity	malicious actor inserts himself as a relay or proxy into a communication session between people or systems.	Man-In-Middle , Session hijacking
3	Confidentiality	unauthenticated, remote attacker to access sensitive data	Packet sniffing, phishing and Pharming, dumpster diving, wire tapping, keylogger, social engineering
4	Scalability	Cloud service provider activates more and more resources to meet the SLA for the availability of the service for the customer, which eventually adds extra billing cost leading to EDoS	Economic Denial of Sustainability (EDoS) Attack - To put financial burden over cloud consumer by consuming metered bandwidth of web application hosted over cloud.
5	Non-repudiation	Lack of multi-layered security approaches , open access to un-trusted third party	E-mail tracking, Man-In-Middle , Session hijacking

Following are the vulnerabilities and attacks due to non-implementation of Quality Attributes in SRS [10].

TABLE III
VULNERABILITIES AND ATTACKS IN QUALITY ATTRIBUTES OF SRS [29][30][31]

SrNo	Quality Attributes	Vulnerabilities	Attacks
1	Conformance	Lack of quality of software product, deliberate software attacks, deliberate act of sabotage, Technological obsolescence	Man-In-Middle, virus, worms, Trojan Horses, DoS
2	Performance	Lack of accessibility of service, graceful degradation, completion of intended task, deviation in quality of service	DoS, Security misconfiguration
3	Features	Stealing personal data, or performing actions on behalf of the user.	Cross-Site Scripting (XSS)
4	Reliability	creating and storing cookies of sensitive information like passwords in browser cache memory.	timing-attack
5	Response	gains super-user privileges	Privilege elevation
6	Service	Easily prone to reveal access credentials and important information through social media communications.	Social Engineering, Sniffing , Shoulder Surfing
7	Reputation	uses the credentials of a legitimate user or device to launch attacks, steal data or bypass access controls	Identity Spoofing
8	Usability	Easy and simple to guess passwords, storing of passwords in a text format in a database.	DDoS, password-crack, Brute Force

E. Methodology for Security Specification Languages

Secure Software development Life Cycle (SSDLC) processes and Secure Software Development (SSD) methods considering software security requirements are part of software security [11]. SSDLC process considers SSD methods. SSD methods include Software Specification Languages and Software Security Assurance methods [12].
Some of the security specification languages are listed below [17][35].

TABLE IV
SECURITY SPECIFICATION LANGUAGES

SrNo	Security Specification Languages	Security Checklist
1	UMLsec: It specifies security requirements through the use of tags, stereotypes and constraints for the purpose of securing system development [36].	Fair exchange (no cheating between parties), Non repudiation, Role-Based Access Control (RBAC), enforcing access control through use of guards, securing communication link, ensuring secure flow of information among components.
2	SecureUML: specifying role-based access control policies (these policies can be considered as security requirements) in a model.	Role-based access control, to specify constraints for resources, actions, and permissions, Specify individual software's requirements
3	Secure Tropos: The SecureTropos notation can be used to represent security constraints (requirements) on interactions between actors during the requirement specification phase [19].	Specify as high level statements, to achieve a secure goal,secure task, or another secure resource
4	UMLintr: It uses UML diagrams like use case, class, state charts diagrams to specify attacks with the help of stereotypes and tags.	Defining privilege levels, transitions between privileges and actions.

F. Security Requirements Engineering Processes

A security requirements engineering process like Secure Tropos, SQUARE, CLASP, Haley, SREP should follow the different activities.
Security Quality Requirements Engineering (SQUARE) is a

ISBN: 978-988-14048-7-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

process for defining security in requirements engineering. It is based on co-ordination between stakeholders and requirement engineers. It consists of nine steps – agreeing on definitions, identifying security goals, developing software artefacts, performing risk assessment, selecting elicitation technique, security requirements elicitation, categorizing requirements, prioritizing requirements and inspecting requirements [12].
The Comprehensive Lightweight Application Security Process (CLASP) is a particular approach used to enhance security in SDLC consisting of different activities [12]. The Security Requirements Engineering Process (SREP) is a nine step process partially based on SQUARE considering Common Criteria and notions of reuse [11]. It is almost similar to SQUARE. The SREP activities are like – agreeing on definitions, identifying vulnerable assets, identifying security objectives and dependencies, identifying threats and developing software artefacts, assessing the risk, eliciting security requirements, categorizing and prioritizing requirements, inspection of requirements and repository improvement [11].
SQUARE and SREP are more elaborative than other methods with reference to the activities to be performed [12].

III. Proposed Secure SRS Model

The requirement phase of secure software engineering consists of different steps like security requirement elicitation, analysis, prioritisation, management. It shows "what" of secure software engineering. It produces secure software requirement specification (SRS) document to proceed further software development life cycle.

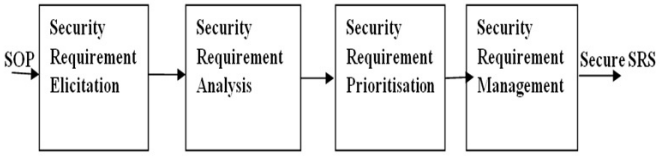

Fig. 2. Proposed Secure SRS Model

- Security Requirement Elicitation: It is the practice of researching and discovering the requirements of a system from users, customers, and other stakeholders [14]. This stage is used to draw out effective as well as efficient requirements related to security of a system from the statement of the problem (SOP). This contains the following steps: Stakeholder identification, Role based activity allotment to each stakeholder, Security requirement identification, Asset identification, Threat identification. The identification of vulnerabilities in functional, non-functional requirements and quality attributes must be done at this phase to avoid the potential attacks in the later stage of the developed product.
- Security Requirement Analysis: This stage is useful in analyzing the effective and efficient security requirements drawn out in earlier elicitation stage. It is the process of determining user expectations for a new or modified product. This contains the following steps: Conflict Resolution, Completeness check, Requirements Inspection, Review Activity. The security requirement analysis phase of proposed secure SRS model should consider security Some Common Mistakes.
- Security Requirement Prioritisation: The main task of this stage is to prioritize the security requirements from the higher importance to lower one in order. It will reduce the risk during development stage. To prioritize the security requirement among all the requirements, the Analytic Hierarchy Process (AHP) is an efficient methodology to select alternative among many others or give priority to alternatives as per the severity [15]. It is a theory of relative measurement with absolute scales of both tangible and intangible criteria based on the judgment of knowledgeable and expert people. Decisions are determined by a single number for the best outcome or by a vector of priorities that gives an ordering of the different possible outcomes [16]. This contains the following steps: Threat Evaluation, Assets rating, Threat Prioritization, Security Requirement prioritization, Risk value computation.
- Security Requirement Management: It is the process of requirements collection, analysis, prioritization as per severity, agree on definition, change management and effective communication to corresponding stakeholders [14]. This contains the following steps: View point identity, Traceability of security requirements, Functional and Non-functional requirements. The proposed SRS model can be made secure by implementing defence mechanism at this stage. Our proposed model addresses these problems with Confidentiality, Integrity, Availability (CIA triad) extended by concepts such as use of Authentication Authorization, Accounting along with scalability and non-repudiation parameters.

A. Security Checklist for Proposed Secure SRS Model

To make robust and secure SRS, security checklist can be applied at security requirement analysis phase of proposed secure SRS model. The security checklist should consider the overall perspective of system safety. It should contain checklist consideration at application level and database level both. The SRS document is declared as secure if it meets all the security conditions. Table 5 and 6 shows the security checklist for Proposed Secure SRS Model.

TABLE V
Application Security checklist for Proposed Secure SRS Model [17][18][20]

SrNo	Checks	Y	N
1	Whether all important assets and installation environment are identified.		
2	Whether SRS inputs are consistent, cohesive, feasible, tested and validated.		

SrNo	Checks	Y	N
3	Whether interrelationship with other inputs as well as other linked products are taken care of and no conflict with other requirements.		
4	Whether security audit review is performed.		
5	Whether Processes like encryption and steganography mechanism to sensitive data has been provided in software development		
6	Whether usable protocols, Encryption Strength, usage of processes, etc defined correctly.		
7	Whether Implementation of Security Controls like Hashing, Digital Signatures, etc. have been defined.		
8	Whether protection from DOS attack, Buffer Overflow, abuse cases and threat modelling has been taken care of.		
9	Whether specifications have been read and understood and all inputs, outputs and interfaces between hardware, software, personnel, and procedures included.		
10	Whether system boundary and data sensitivity or criticality defined.		
11	Whether threats and relevant sources of attack identified.		
12	Discussed the working of the product and determined areas of disagreement or ambiguity.		
13	Whether vulnerabilities have been found out using any security tools or checklists.		
14	Whether current and planned security controls understood.		
15	Whether attack scenarios for exploits of vulnerabilities mapped out.		
16	Whether impact analysis is performed and mitigation strategy is developed.		
17	Whether mitigation strategy developed.		
18	Whether countermeasures to mitigate risks considered.		

TABLE VI
DATABASE SECURITY CHECKLIST FOR PROPOSED SECURE SRS MODEL [21][22][23][24][25]

SrNo	Checks	Y	N
1	Whether only required necessary privileges to users have been provided		
2	Whether Lock and expire done for default user accounts		
3	Whether all default database user passwords, administrative accounts passwords are changed.		
4	Whether rules and guidelines for password complexity has been set up		
5	Whether testing user has been disabled to gain access to the production database through database links or linked servers.		
6	Whether only database administrators and system administrators has access to database server, files, SYSDBA and SYSOPER roles.		
7	Whether hardening of operating system (OS) and encryption of network traffic done.		
8	Whether unauthorized administration of the Oracle Listener has been prevented to secure database.		
9	Whether security engineers have reviewed database design changes.		
10	Whether only the database components, modules or functionality that your application is intending to use are installed		
11	Whether a disaster recovery (DR) strategy for every production database is set up to ensure business continuity plan (BCP).		
12	Whether the default configuration for database and operating system in order to prevent easy access to data has been changed.		
13	Whether network firewalls, Intrusion Detection System (IDS), antivirus and antispyware programs are in place to protect the database environment.		
14	Whether a secure Virtual Private Network (VPN) has been set up for remote database connection to the only intended user.		
15	Whether production data is not used in a development or testing environment to protect the privacy and confidentiality of the data.		
16	Whether different types of auditing of database activities is planned to check for any violation of security in access control, authentication, activity tracing, change management.		

Here, Y represents the checks in security checklist have beenfound positive and N represents the checks in security checklist have been found negative.

B. Proposed Defence Mechanisms in SRS

The CIA Triad (Confidentiality, Integrity and Availability) & AAA (Authentication, Authorization and Accountability) model are the core security parameters around which every product/software security controls is defined. The proposed security mechanism SRS model in SDLC is to protect the security triad parameters like Confidentiality, Integrity and Availability in conjunction with AAA model. Our proposed model addresses the security problem by extending with scalability and non-repudiation parameters.

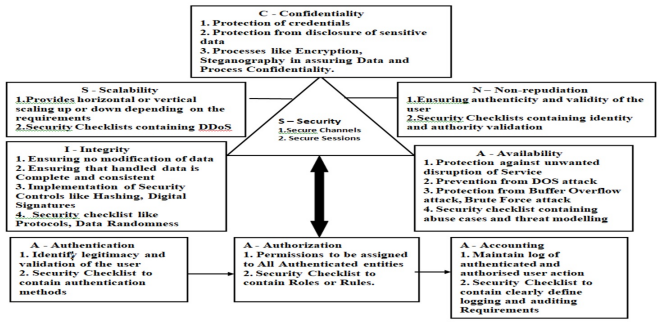

Fig. 3. Proposed Defence Mechanism in SRS

The secure SRS document is the result of proposed secure SRS model with the integration of proposed security checklist at security analysis phase and proposed defence mechanism at security management phase.

ISBN: 978-988-14048-7-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

IV. Conclusion

The extensive literature survey on vulnerabilities and attacks in SRS for Object-Oriented Software Development is made and security checklist is prepared to write the Secure Requirement Specification (SRS) document. The proposed secure SRS can be used as the reference document for the analysis phase of the statement of the problem. The defence mechanism is also proposed to secure the analysis of the system. A high-level approach to mitigate the risks involved in software has been discussed. The vulnerabilities arising due to loopholes during analysis phases of SDLC can be better avoided by security checklists proposed to protect security triad in conjunction with AAA model in traditional model, OO model. Security Quality Requirements Engineering (SQUARE) and Security Requirements Engineering Process (SREP) are more comprehensive than other methods in analysis phase of SDLC. Also security checklists for software security specification languages like UMLsec, SecureUML, SecureTropos, UMLintr have been provided. The proposed SRS model and the defence mechanism can be considered during analysis phase of the application and can be embedded into different analysis diagram.

References

[1] D. Firesmith, "Common Requirements Problems, Their Negative Consequences, and the Industry Best Practices to Help Solve Them", *Journal of object Technology*, Vol. 6, No. 1, 2007.

[2] Nancy Mead, "Security Requirements Engineering," *Software Engineering Institute, Carnegie Mellon University*, August 2006.

[3] J. Ansari, D. Pandey, M. Alenezi,"STORE: Security Threat Oriented Requirements Engineering Methodology", *Computer and Information Sciences, Elsevier*, Decemer 2018.

[4] M. Hussein, M. Zulkernine, "UMLintr: A UML Profile for Specifying Intrusions", *13th Annual IEEE International Symposium and Workshop on Engineering of Computer Based Systems (ECBS'06), IEEE*, 2006.

[5] M. Bishop, "Introduction to Computer Security," *Addison Wesley*, 2005.

[6] D. Hein, H. Saiedian, "Secure Software Engineering Learning from the Past to Address Future Challenges," *Information Security Journal: A Global Perspective*, 2014.

[7] D. Firesmith, "Specifying Good Requirements," *Journal of Object Technology (JOT)*, 2003.

[8] Panko, R. R, "Corporate computer and network security," *Upper Saddle River, NJ: Prentice Hall*, pp. 324–330, 2003.

[9] P. Singh, D. Singh, A. Sharma, "Classification of Non-Functional Requirements from SRS documents using Thematic roles", *IEEE International Symposium on Nanoelectronic and Information Systems* , 2016.

[10] N. Gupta, B. Valarmathi, "Total Quality Management," *Tata McGraw Hill Education Pvt. Ltd.*, 2009.

[11] P. Salini, S. Kanmani, "A Survey on Security Requirements Engineering", *International Journal of Reviews in Computing*, December 2011.

[12] A. Agrawal, R. A. Khan, "A Framework to Detect and Analyze Software Vulnerabilities- Development Phase Perspective", *International Journal of Recent Trends in Engineering*, Vol. 2, No. 2, 2009.

[13] N.R. Mead, E.D. Houg, and T.R. Stehney, "Security Quality Requirements Engineering (Square) Methodology", *Software Eng. Inst., Carnegie Mellon Univ.* , 2005.

[14] Requirements Managements, [Online]. Available: https://en.wikipedia.org/wiki/Requirements_managements.

[15] P. Lokhande , B. Meshram, "Analytic Hierarchy Process (AHP) to Find Most Probable Web Attack on an E-Commerce Site", *ICTCS, ACM*, March 2016.

[16] Thomas L. Satty, "The Analytic Network Process," *University of Pittsburgh*.

[17] M. Umair Khan, M. Zulkernine, "On Selecting Appropriate Development Processes and Requirements Engineering Methods for Secure Software", *Queen's University Kingston, Ontario, Canada K7L 3N6*.

[18] S. Mostafa, H. Jani, "Online Checklist-Based Approach to Software Requirements Specifications Quality Analysis," *In Proceedings of 1st TNB ICT Technical Conference, College of Information Technology*, February 2011.

[19] E. Wallmuller, "Software Quality Assurance: A Practical Approach," *Prentice-Hall*.

[20] SRS Checklist, [Online]. Available: https://www.cs.helsinki.fi/group/linja/resources/srs_checklist.html.

[21] H. Afyouni, "Database Security and Auditing," *CENGAGE Learning*.

[22] Datatbase Security Checklist, [Online]. Available: https://www.isaca.org/groups/professional-english/oracle-database/groupdocuments/twp-security-checklist-database-1-132870.pdf.

[23] Security Checklist, [Online]. Available: https://docs.oracle.com/cd/B19306_01/network.102/b14266

[24] Database Auditing:Best Practices, [Online]. Available: http://www.sfisaca.org/images/May09_Slides.pdf1Q,M3,521fd9a2-f86d-4991-8428-8e8324b89ecd.

[25] Database Security Best Practices, [Online]. Available: https://www.esecurityplanet.com/network-security/6-database-security-best-practices.html.

[26] R. Kumar, Mustafa K, "Security Requirements Development Framework (SRDF)", *International Journal of Advanced Research in Computer Science*, Vol. 2, No. 5, 2011.

[27] S. Jain, M. Ingle, "Software Security Requirements Gathering Instrument", *International Journal of Advanced Computer Science and Applications*, Vol. 2, No. 7, 2011.

[28] M. Fletcher, H. Chivers, and J. Austin, "Combining functional and security requirements' processes," *In Proceedings of All Hand Meeting*, 2005.

[29] J. Cleland-Huang, R. Settimi, X. Zou, P. Solc, "The Detection and Classification of Non-Functional Requirements with Application to Early Aspects", *14th IEEE International Requirements Engineering Conference (RE'06)*, 2006.

[30] Non-Functional Requirements, [Online]. Available: http://users.csc.calpoly.edu/ jdalbey/SWE/QA/nonfunctional.html.

[31] Threats and Attacks, [Online]. Available: http://web.cse.ohio-state.edu/čhampion.17/4471/4471_lecture_2.pdf.

[32] Y. Yeole, B. Meshram, "Analysis of different technique for detection of SQL injection", *ICWET-TCET, Mumbai, India*, 2011.

[33] K. Sindhu, B. Meshram, "Digital Forensics and Cyber Crime Datamining", *Journal of Information Security*, Vol. 3, pp. 196-201, 2012.

[34] P. Ambhore, B. Meshram, V. Waghmare, "A Implementation Of Object Oriented Database Security", *Conference on Software Engineering Research, Management and Applications, IEEE*, 2007.

[35] Attar, H. Luqman, P. Karpati, G. Sindre, "Extending the UML Statechart Notation to model Secuirty Aspects", *IEEE Transactions on Software Engineering*, Vol. 41, No. 7, 2015.

[36] M. Khan, "Representing security specifications in UML state machine diagrams", *International Workshop on Enterprise Web Application Dependability (EWAD 2015), Elsevier*, 2015.

ISBN: 978-988-14048-7-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)