Software Security Requirements Gathering Instrument

Smriti Jain
Reader, MCA Department
SRGPGPI
Indore, India

Maya Ingle
Principal
Indore Institute of Computer Applications
Indore, India

***Abstract* — Security breaches are largely caused by the vulnerable software. Since individuals and organizations mostly depend on softwares, it is important to produce in secured manner. The first step towards producing secured software is through gathering security requirements. This paper describes Software Security Requirements Gathering Instrument (SSRGI) that helps gather security requirements from the various stakeholders. This will guide the developers to gather security requirements along with the functional requirements and further incorporate security during other phases of software development. We subsequently present case studies that describe the integration of the SSRGI instrument with Software Requirements Specification (SRS) document as specified in standard IEEE 830-1998. Proposed SSRGI will support the software developers in gathering security requirements in detail during requirements gathering phase.**

Keywords – Software Requirements Specification; Security Policy; Security Objectives; Security Requirements.

I. INTRODUCTION

Requirements define necessary and desired capabilities of the proposed system. Hence, requirements' gathering is the first step towards the development of software. The requirements gathered are functional and non-functional. The functional requirements describe the functionality of the product whereas the non-functional requirements attribute to quality features of the software. Functional requirements define the business rules. The non-functional requirements focus on issues like maintainability, portability, usability, security etc. Requirements are concerned with what the system should do whereas the security requirements are concerned with what the system should not do. Security requirements gathering allow gather information about the malicious part of the environment and decides how security breaches can be nullified [1]. To develop secured software, the core security services (confidentiality, integrity, availability, authentication, authorization and auditing) should be incorporated in the requirements phase of a software development project. Good requirements must be complete, correct, feasible, necessary, prioritized, unambiguous, and verifiable [2]. Common requirements are business rules, budgets, interfaces, reports, security, hardware, software etc. These requirements are defined by the stakeholders. Studies have shown that 40% to 60% of all defects found in software projects are due to errors made while gathering requirements. The main reasons contribute to the lack of user inputs, incomplete user requirements and changing requirements [3]. With the help of field study, the reasons for poorly specified requirements have been identified as inconsistency in the selection of requirements, inconsistency in level of detail, and almost no requirements on standard security solutions [4]. Including security specifications during the requirements phase will not only reduce defects but will also assist in reducing security breaches.

The software requirements can be identified using the guidelines specified in IEEE Std. 830-1998 Software Requirements Specification (SRS) document [5]. In the literature survey, a Software Security Checklist has been developed that builds the software security assessment instrument. It focuses on the software security checklist to ensure that all the security aspects have been included during the software development process [6]. A process has been developed that enables the developers to identify, analyze and finalize the security requirements by the means of software security components. The process identifies and analyzes the conflicts between different security requirements [7]. In the SQUARE methodology, the security requirements are treated as add-ons to the functional requirements and are carried out in early stages of software development in nine discreet steps [8]. It is also stated that Common Criteria (CC) allows for the development of security requirements, and is being used on the architectural level of the security requirements. This made the usage of CC more beneficial [9]. In a paper, the functional and security requirements are combined to develop Distributed Aircraft Maintenance Environment (DAME) system in order to meet both the objectives [10]. Thus, it is evident that most of the work deals with security requirements only, enhancing existing security requirements standards, or security as a non-functional requirement in SRS. It has also been revealed that security is considered during requirements gathering in brief. We propose a Software Security Requirements Gathering Instrument (SSRGI) which can be used to gather the security requirements. The functioning of the proposed instrument is anticipated by case studies from different domains.

In the rest of the paper, we first present a brief overview of IEEE Standard 830-1998 in section II. Section III introduces SSRGI and its integration with SRS, and we verify our instrument with the help of case studies in section IV. Finally, section V concludes with the results, conclusion and the scope of SSRGI.

II. IEEE STANDARD 830-1998

IEEE Standard 830-1998 explains the content and qualities of good SRS. It aims in specifying the requirements of software to be developed for in-house and commercial products. The standard is divided in three parts consisting of Introduction, Overall Description and Specific Requirements. Introduction includes Purpose, Scope, Definitions and Acronyms, References, and Overview. Overall description consists of six subsections viz. Product Perspectives, Product Functions, User Characteristics, Constraints, Assumptions and Dependencies, and Apportioning of Requirements. Product perspectives describe how software operates in various constraints. These constraints are System interfaces, User interfaces, Hardware interfaces, Software interfaces, Communication Interfaces, Memory, Operations, and Site Adaptation requirements. Constrains subsection provide description that limit the developer's options. It includes Regulatory Policies, Hardware Limitations, Audit Functions, Control Functions, Safety and Security Considerations etc. Specific requirements focus on External Interfaces, Functions, Performance Requirements, Logical Database Requirements, Design Constraints, Software System Quality Attributes, and Object Oriented Models. The external interfaces requirements compliment the interface requirements and detail the system inputs and outputs. Functional requirements define fundamental actions in accepting and processing the inputs and generating the outputs. Standards compliance specifies the requirements derived from existing standards or regulations. The software system attributes include Reliability, Availability, Security, Maintainability and Portability.

III. PROPOSED SOFTWARE SECURITY REQUERMENTS GATHERING INSTRUMENT (SSRGI)

Fig. 1 illustrates the proposed SSRGI that demonstrate the course of action to gather security instruments. The various security requirements to be gathered are Secure Functional Requirements (SFR), Drivers, Functional Security Requirements (FSR), Non-Functional Security Requirements (NFSR), Security Development Requirements (SDR), and Security Testing Requirements (STR). The instrument specifies the security policy which helps in identifying security needs and objectives that decide the security requirements. These requirements can be gathered from the various roles such as customers, managers, designer, coders, and QA/ Testers.

A. Security Policy

Security policy allows an organization to set security practices and procedures to reduce the likelihood of attack. It defines rules that regulate an organization in accomplishing its security objectives. Policies describe the roles of users, managers, designers, coders and quality assurance team in achieving security. Security policies reduce the damage to the business by safeguarding the confidentiality, availability and integrity of the data and information. Security policies identify security needs and objectives. The policies may include the regulatory acts like HIPAA, Sarbanes-Oxley etc.

B. Security Needs and Objectives

The needs and objectives facilitate to create and better understand the comprehensive security plan. Objectives are goals and constraints that affect confidentiality, integrity and availability of data and application. It also defines roles and responsibilities and privileges to be assigned to the roles. Various other security objectives include – resource protection, authentication, authorization, non-repudiation, auditing security objectives etc.

C. Security Requirements

The security requirements are to be gathered from the various roles viz. customer, manager, designers, coders, and Quality Assurance/ Tester. The various types of security requirements are discussed below:

Secure Functional Requirements (SFR) - SFR is a security related description that specifies the services to be integrated into each functional requirement. SFR specifies what shall not happen while executing the software. The customer helps in determining the SFRs. These requirements are normally gathered by means of misuse cases which capture requirements in negative space. The counter measures of misuse cases are design decisions [16].

Drivers – The security drivers determine the security needs as per the industry standards, thereby shaping security requirements for a software project. The drivers for security requirements include regulatory compliance like Sarbanes-Oxley, Health Insurance Portability and Accountability Act (HIPAA), Gramm-Leach-Bliley Act etc.; industry regulations and standards like ISO 17799, OASIS etc.; company policies like privacy policies, coding standards, patching policies, data classification policies etc.; and security features like authentication and authorization model, role-based access control, and administrative interfaces etc. [11]. The policies when transformed to detailed requirements demonstrate the security requirements. By using the drivers, managers can determine the security requirements necessary for the project.

Functional Security Requirements (FSR) - FSRs are the requirements that focus on the system under inspection. The requirements for the FSRs can be gathered from the managers using the security drivers. FSRs comprises of authentication, authorization, backup, server-clustering, access control, encryption, data integrity etc. Examples include verification of all the users and allow them to access information relevant for their use. The managerial level of the organization can help determine the FSRs.

Non-Functional Security Requirements (NFSR) - NFSR are security related architectural requirements, like

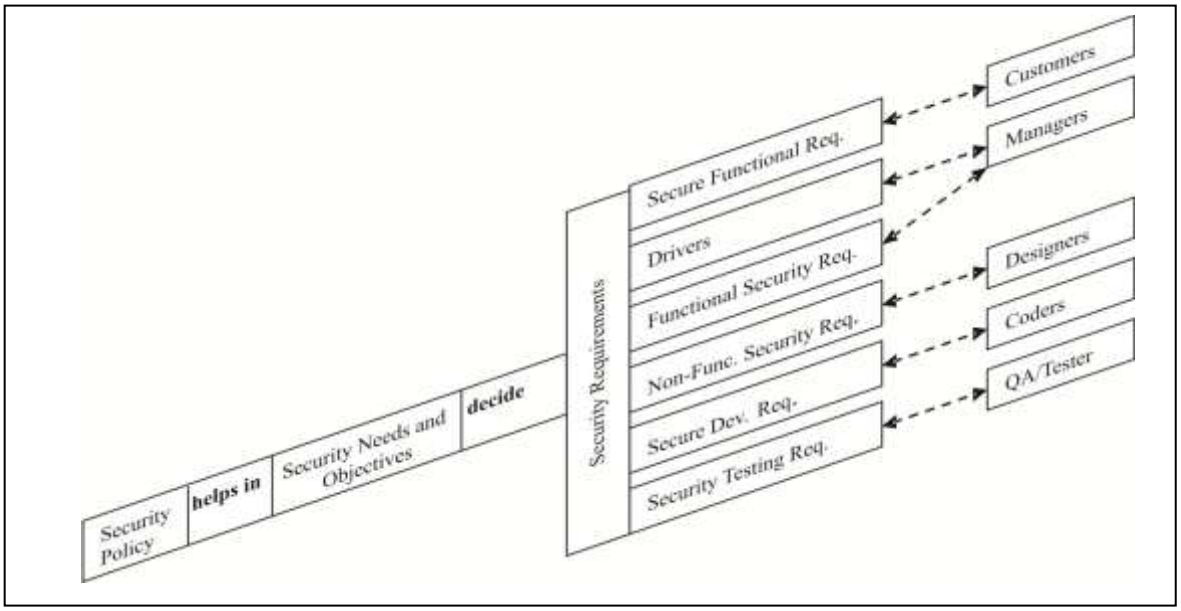

Figure 1: Software Security Requirements Gathering Instrument (SSRGI)

robustness, privacy, reliability, data authenticity, integrity etc. This requirement type is typically derived from architectural principles and good practice standards using Microsoft's SDL or Cigital's TouchPoints. NFSRs are security related quality attributes. Design patterns like creational, structural or behavioral can be used to generate NFSRs in object oriented programming. It also identifies system's resilience and level of immunity to attack [12]. NFSR requirements can detect and report all unauthorized access. The designers normally focus on NFSR.

Secure Development Requirements (SDR) - SDRs describe required activities during system development which assures that the outcome is not subject to vulnerabilities. Coding guidelines specifies the SDRs. Also, frameworks like Comprehensive, Lightweight Application Security Process (CLASP), Cigital TouchPoints, OWASP etc. may assist in implementing secure development practices. CLASP describes 104 coding problem types specified in 5 overlapping categories [15]. OWASP specifies top ten web vulnerabilities like Injection, Cross Site Scripting (XSS), broken authentication and session management, etc. these vulnerabilities are mainly due to lack of secure coding hence, the coders centers on SDRs.

Security Testing Requirements (STR) - STR includes testing the security requirements gathered using a parametric approach [13]. Security testing mainly focus on testing security functionality as gathered in the requirements document. The QA/ Testers needs to gather information about the hardware architecture, software architecture, and the user model to develop the security test cases. The testing team can test the valid and invalid access rights to check for authentication and authorization of the proposed software.

IV. CASE STUDY

Based on the integration of SSRGI into IEEE 830-1998 SRS, Fig. 2 shows the said merger. The SSRGI is incorporated in the software system's security attribute. The SSRGI elaborates on gathering SFR, Drivers, FSR, NFSR, SDR, and STR requirements from the various stakeholders. We describe three different cases viz. Web Based, LAN based Client/ Server, and Single User system that illustrate the use of our SSRGI. An integrated approach of SSRGI with SRS has been presented for each of the three cases.

Case I: Web Based System - Journal Publishing System

Journal Publishing System (JPS) is a Web publishing system designed for a regional society. The system is designed to assist editors by automating the article review and publishing process thereby maximizing his efficiency. The software will facilitate communication between editors, authors and reviewers via E-mail. Preformatted reply forms are used for communication between editor, author and reviewer. The system also maintains a database of reviewers, authors and articles. The main objectives of the system include accepting articles online from contributors, email acceptance letter to them, send the articles to reviewers, receive the feedback from the reviewer and further communicate the feedback to the contributor using E-mail, accept the camera-ready copy of the article online from the contributor, publish the article and make the published articles available to the contributor for 1 year. The articles can be subscribed either for 6 months or 1 year. Considering it to be a moderately used system with approximately having 1000 users online in 5 minutes, minimum server configuration include 3.2 GHz Single Processor, Quad Core, 4 GB RAM and having at least 300GB HDD with Web server configuration as Apache 2.x

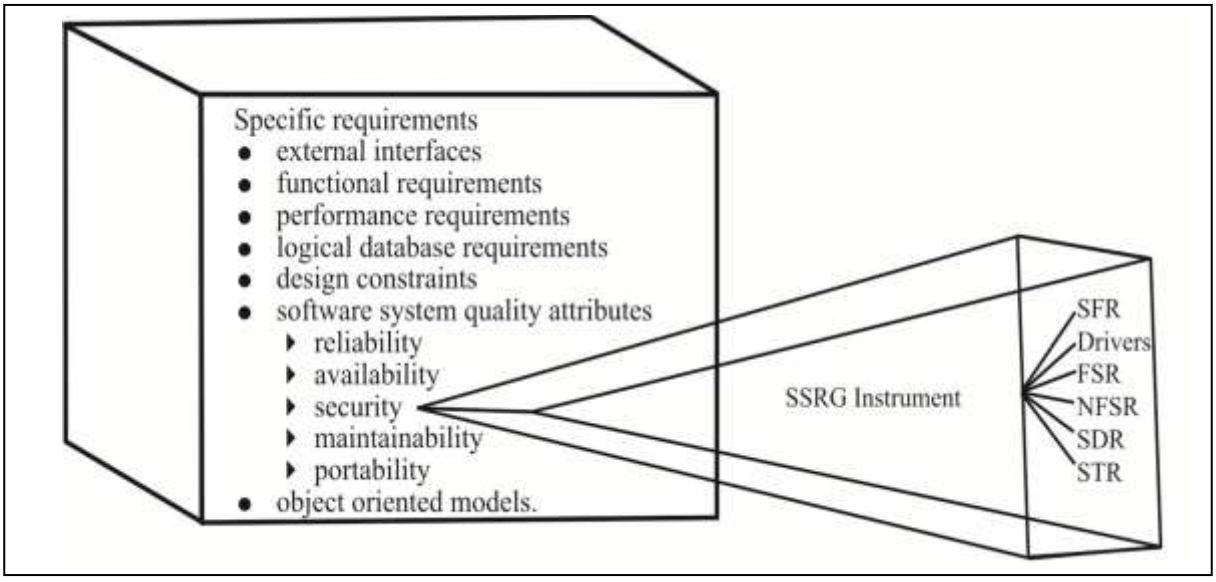

Figure 2: Integration of SSRGI into SRS

and above having Linux supporting Redhat Linux EL4 and above as operating system. The software is developed in PHP 5.x and above, having database as MySQL 5.x. The minimum requirement at client side must be Windows XP/ Windows 7 Professional, IE 7.0.and above/ Firefox 3.0/ Google Chrome, MS Office and Adobe Reader. The predefined messages can be sent using Outlook Express.

Here, SRS explains the purpose of the system along with the features, interfaces and outputs. At the same time, it will clarify the constraints under which it will operate and external environment it will react. The system allows editor to interact with the reviewers and the contributors, and to publish the articles on the website. The articles are published on some nominal charges and the contributors can access the website free of cost for 1 year. The subscribers are allowed to view the published articles on the basis of subscription fee.

The various security requirements of JPS are specified in security section of SRS. SFR includes logon page, password security for various users, password reset, account creation for new contributor and reviewer, contacting the editor, submitting information to web site, preventing spamming of website submission, and backup of the data. The managers can take the help of Open Web Application Security Project (OWASP) for requirements gathering. The managerial level of the organization decides on the need for secure data transmission, securing user identification for the contributors, reviewers and the editor using Secure Socket Layer (SSL) protocol. The company policy decides access requirements for read/ write/ delete to authorized users. Editors are given read/ write/ delete access, reviewers are provided read/ write access of the articles of their own subject area, and contributors and subscribers are given read access to all the articles. Editors need to identify the unauthorized users who can pose threats to the system; and the technical and business impacts like loss of revenues, loss of identity of the authors, loss of honorium of the reviewers etc. FSRs like backups, authentication, authorization, access control, and encryption should also be specified by the managers. The SDR can be specified by the developers using OWASP and CLASP. In SDRs, the coders also need to state the formatting of the data, exceptional conditions etc. The software tester should detail the contents of test plan to evaluate security of software, software test procedures, software test reports, and acceptance criteria for the users in STR specification. The test plan may address issues on password cracking, URL manipulation, SQL injection, XSS and spoofing.

A study reveals that 47% of banks places secure login boxes on insecure pages, and 55% put contact information and security advice on insecure pages. As a result, an attacker could change the information and set up his own call center to gather private data from customers who need help from banks [14]. Since JPS asks for personal information from the subscribers, it allows to specify the need for SSL protocol. XSS is one of the flaws which affect the website most. To avoid such flaw, the system designer must include requirement of either filtering the input from the sender or by using frameworks that specify libraries for making user input safe from XSS like ASP.NET. To secure data from potential loss, a disaster recovery plan can be tailored to suit the requirements. Another such flaw lies in SQL injection in which databases are accessed by the hackers. To avoid SQL Injection attack, check and validate input to SQL statements and never use string concatenation to build SQL statements, instead use parameterized queries. The testing requirement includes manual code review and check for the keywords that validate the input.

Case II: LAN based Client/ Server System - Patient Management System

Patient Management System (PMS) is a LAN based client/ server system that allows the hospitals to keep record of the patient's data. This aids in the management of personalized patient lists, physicians etc. The system is intended to assist doctors, nurses, social workers, and dieticians. The system supports internal messaging among various users. The system

is designed to register patients, doctors, nurses and dieticians. It allows generating the list of patients with their ailments, medications and tests (if suggested), and the doctor and nurses in-charge of the same. The system also permits to view the case history of the patients and fix appointment with the doctor. The system generates birth and death records, test reports, diagnosis with medicines, and billing and payments. Considering 100 patients daily including old and new ones, the minimum server configuration include 2.8 GHz Single Processor, Quad Core, 4 GB RAM and having at least 300GB HDD. The software is developed on DotNet Framework and database used is SQL Sever 2008 Express. The client machine must be Intel Pentium Dual Core (E5800), 2 GB DDR3 SDRAM, 320 BG SATA Hard drive (7200 rpm) and Broadcom Integrated Gigabit Networking (BRCM 57780) with Windows XP/ Window 7 Professional with DotNet Framework and; dot matrix and laser printers.

In PMS, SFR includes information regarding logon page, password security for the various types of users, password reset, account creation of new user etc. The company policy facilitates to decide the access rights of all the users as the security threats are mainly from the internal users. The managers decide upon the need for encryption and password strength depending upon the type of security required. The requirements gathering team gathers security requirements from the managers by help of misuse cases to identify the unauthorized users who can pose threats to the system. The designers may use security lifecycle to develop a secured design of the software. They must specify the vulnerable points of the software and must ensure that the unused features must be off by default. SFRs can be regarding the authentication, authorization, access control, backups, encryption etc. The designers must specify the audit mechanisms to identify intruders. These mechanisms are the counter measures for misuse cases. The designers, in addition, can take help of CLASP to gather security requirements during coding. In SDRs, the coders must specify the exceptional cases as well as input and output validation. The tester needs to state the test plans to evaluate security of the software, the test procedures, software test reports, and the acceptance criteria in STR specification. The test plans should address communication threats, password cracking etc.

Access control is one of the major design issues in a client/ server based system. Access control list can be developed by threat modeling. It is gathered from the customers to achieve secured system. The designers must understand the design issues related to access control on the target platform.

Case III: Single-User System - Shop Management System

Shop Management System (SMS) permits a shopkeeper to keep track of the inventory, sales, and accounts (i.e. preparation of ledger and balance sheet). The system allows maintaining list of inventory available and required, daily/ weekly/ monthly sales, raises purchase indent, and maintains outstanding amount. It generates reports on daily/ weekly/ monthly sales, outstanding amounts, purchase details, list of suppliers, customers with addresses (for home delivery of goods) and their credit amount. The minimum requirements include 2 GHz processor, 2 GB RAM, MS-Access, Windows XP with SP3 or Vista, Office 2003, 40 GB disk space and Broadband Internet connection with IE 7.0/ IE 8.0/ Firefox 3.0/ Google Chrome, bill and laser printers.

The shop-owner, the user of the system, is the administrator and has access to all the data. FSR includes logon and password facility for accessing the computer system, finger print recognition pocket device to access the software to make it safe from the helpers in the shop, and backup of the data at regular intervals.

V. Results And Conclusion

Requirements gathering is one of the most important steps towards the development of a software product. The traditional approaches of requirements gathering do not include security in detail. In this paper, we presented SSRGI that provides a general approach to incorporate security during the requirements gathering phase of the software development process. It focuses on different types of security requirement that can be gathered from different roles. Gathering security requirements with the help of the instrument ensure that the security aspects are considered systematically during development process itself and can thus help avoid serious flaws in the resultant software product.

SSRGI is then integrated in the SRS guidelines, as suggested in IEEE 830-1998 document, to provide a more detailed approach to incorporate security during the software development process. This instrument is then applied to a Journal Publishing System, Patient Management System and Shop Management System and is presented as case studies. The results of integrating SSRGI in the various softwares considered in case study is detailed below.

- Web enabled software involves the highest need for security. SSRGI when applied is able to gather the security requirements in more detail. The access rights are detailed for all the various types of users. It also assists in identifying the unauthorized users, coding guidelines and test requirements for security, thus serving to develop more secured software. The instrument allows setting up the password policy. It permits to gather need for SSL protocol.
- SSGRI allows gathering requirements from the stakeholders. Business requirements and security requirements are gathered via customer interaction, whereas the other requirements are gathered from the development team members.
- LAN based client/ server system is more prone to threats from internal users. The need for communications security as well as the security measures required for web applications is reduced. SSRGI allows capturing requirements for authentication and authorization, proper password strength, and secure internal communication.
- In a single user system, login, password, and the backup of the data are the security measures required. Hence in such conditions, the need for applying security gathering instrument is also reduced.

From the above cases, it is evident that SSRGI will ensure the systematic consideration of security aspects during development process itself and can thus avoiding serious flaws in the resultant software product. SSRGI will support the software developers to gather security requirements in detail during requirements gathering phase. It will also ensure that the security is not considered as an add-on after the implementation of the project. This will also assist in identifying the security flaws during the early stages of the software development.

REFERENCES

[1] H. Schmidt, "Threat- and risk-analysis during early security requirements engineering," In the Proc. of Availability, Reliability, and Security, ARES'10 International Conference, IEEE Computer Society, 2010, pp. 188 – 195.

[2] E. Whitney, "An introduction to gathering requirements, creating Use Cases and the UML," White Paper, EPS Software Corporation. http://www.eps-cs.com/pdf/whitepaper_the_development_process.pdf.

[3] Al Neimat, "Why IT projects fail?" 2005. http://www.projectperfect.com.au/info_it_projects_fail.php.

[4] J. Wilander and J. Gustavsson, "Security requirements – A field study of current practice," Presented at the Symposium on Requirement Engineering for Information Security (SREIS' 2005), Paris, France, 2005.

[5] "IEEE recommended practice for software specifications requirements," *IEEE Std. 830-1998*, 1998.

[6] D. Gilliam, T. Wolfe, J. Sherif, and M. Bishop, "Software security checklist for the software life cycle," In the Proc. of the 12th IEEE International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprise, June 2003, pp. 243–248.

[7] D. Hatebur, M. Heisel1, and H. Schmidt, "Analysis and component-based realization of security requirements," In the Proc. of the Third International Conference on Availability, Reliability and Security, IEEE Computer Society, 2008, pp. 195-203.

[8] N.R. Mead, E.D. Hough, and T.R. Stehney II, "Security Quality Requirements Engineering (SQUARE) Methodology," CMU/SEI, Technical Report CMU/SEI-2005-TR-009, ESC-TR-2005-009, Nov. 2005.

[9] J. R. Mariona , H. Ziv , D. J. Richardson, "CCARCH: Architecting Common Criteria security requirements," In the Proc. of Third International Symposium on Information Assurance and Security, IEEE Computer Society, 2007, pp. 349-354.

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

[11] R. Araujo, "Security requirements engineering: A road map," Security Feature, July 2007. http://www.softwaremag.com/l.cfm?doc=1067-7/2007

[12] W. J. Lloyd, "A Common Criteria based approach for COTS component selection," Journal of Object Technology, Vol. 4, No. 3, 2005, pp. 27–34.

[13] A. K. Singh, A. J. Iyar, and V. Seshadri, "A parametric approach for security testing of Internet applications," QAI India Limited, In the Proc. of the 3rd Annual International Software Testing Conference, 2000.

[14] "Potentially serious security flaws found in most bank websites, including large bank sites, study shows," Science Daily, July 23, 2008. http://www.sciencedaily.com/releases/2008/07/080722175802.htm.

[15] "CLASP version 2.0," Secure Software Inc., 2006.

[16] G. Sindre, and A. L. Opdahl, "Capturing security requirements through misuse cases," In the Proc. of the 14th annual Norwegian Informatics Conference, Norway, 2001.

AUTHORS PROFILE

Smriti Jain, Associate Professor at SRGPGPI, Indore, India. She has 14 years of teaching experience. She is pursuing Ph.D. in Computer Science and is MCA from School of Computer Science, Devi Ahilya Vishwavidhyalaya, Indore, India. She has 12 papers published to her name in various National and International journals and conferences.

Dr. Maya Ingle is Professor & Senior System Analyst at School of Computer Science and Information Technology, Devi Ahilya University, Indore since 25 years. Presently, she is Principal and Professor at Indore Institute of Computer Application, Indore on extra ordinary leave. She is also Chair Assessment & Research committee. She is Ph. D. in Computer Science and M. Tech in Computer Science from IIT, Kharagpur. Her research interest includes Software Engineering, NLP, Speech Recognition, Usability Engineering and Agile Computing. She has published more than 80 papers in national and international journals and conferences.