

IMPLEMENTATION OF MOSRE FRAMEWORK FOR A WEB APPLICATION - A CASE STUDY

P.Salini¹ and S.Kanmani²

¹Department of Computer Science and Engineering
salini@pec.edu

²Department of Information Technology
skanmani@pec.edu

¹ and ²Pondicherry Engineering College, Puducherry, India

ABSTRACT

The Security Engineering discipline has become more and more important in the recent years. Security requirements engineering is essential to assure the Quality of the resulting software. An increasing part of the communication and sharing of information in our society utilize Web Applications. Last two years have seen a significant surge in the amount of Web Application specific vulnerabilities that are disclosed to the public because of the importance of Security Requirements Engineering for Web based systems and as it is still underestimated. Integration of Web and object technologies offer a foundation for expanding the Web to a new generation of applications. In this paper, we outline our proposed Model- Oriented Security Requirement Engineering (MOSRE) Framework for Web Applications. By applying Object-Oriented technologies and modeling to Security Requirement phase. So the completeness, consistency, traceability and reusability of Security Requirements can be cost effectively improved. We implemented our MOSRE Framework for E-Voting Application and set of Security Requirements are identified.

KEYWORDS

Object-Oriented, Security Requirements, Security Requirements Engineering and Web Applications

1. INTRODUCTION

The development of Web systems usually involves more heterogeneous stakeholders than the construction of traditional software. Therefore a thorough Security Requirements analysis is even more relevant. The requirements must be clear, comprehensive, consistent and unambiguous. Most requirement documents were written in ambiguous natural languages which are less formal and imprecise and it is hard to analyze and integrate with artifacts in other phases of software life cycle. This statement has significance for security requirements and if you say application must be secure, it is not security requirements. It is hard to construct secure web applications or to make statements about security unless we know what to secure, against whom and at what extent. To this day, not one web application technology has shown itself invulnerable to the inevitable discovery of vulnerabilities that affect its owners' and users' security and privacy. Most security professionals have traditionally focused on network and operating system security. Assessment services have typically relied heavily on automated tools to help find holes in those layers. Security Requirements engineering (SRE), a phase that comes before design and programming, will play a more important role that determines the success of Web Applications Design.

In fact Security requirements engineering should be as complex and well thought out as the design and programming, yet its insufficiencies have led to many projects with poor Security requirements and blamed as the major reason for many web applications' failures. Therefore, Security requirements engineering is now moving to the forefront of gaining increased significance in software engineering for service oriented web applications. Web application requirements have new characteristics causing them to change more rapidly. This makes traditional Security requirements modeling and validation methods insufficient to provide adequate support for web applications.

The Security requirements of the web applications come from not only the general domain analysis and the personalized, diverse users' requirements, but also the availability of the related web services. Web applications Security requirements are also evolving while they are widely used. Most of the methodologies that have been proposed for the development of Web applications focus only on Non Security requirements and paying no attention to the Security requirements engineering. Therefore, SRE for Web applications is challenged to explore sound engineering approaches for eliciting, describing, validating and managing Security requirements of Web applications and its integration with the artifacts of other phases can be cost effectively improved and can effect a significant reduction of the problems currently encountered in the SDLC for Web Applications due to poor Security Requirements Engineering and Management.

In this paper, section 2 discuss on related works and present an overview of Model Oriented Security Requirements Engineering framework for Web Applications in section 3. Section 4 gives the implementation of the MOSRE framework to E-Voting Web Application- a case study, while section 4 presents the result analysis and discussion, and the last section concludes with future works.

2. RELATED WORKS

There are many requirements engineering approach for the development of Web Applications, but only some considers security requirements and also as non functional requirements. Some models are object oriented and in this section some of the related works are discussed. SOHDM: Scenario-based Object-Oriented Hypermedia Design Methodology [9] was the first approach stressing the importance of a process that allows the analysts to capture and define the application requirements. RNA: Relationship-Navigational Analysis [10] is a methodology that offers a sequence of steps to develop Web applications focusing mainly on analysis. HFPM: Hypermedia Flexible Process Modeling Olsina [11] is a wide engineering-based approach, which includes analysis-oriented descriptive and prescriptive process modeling strategies. It includes technical, management, cognitive and participatory tasks. HDM: Object Oriented Hypermedia Design Model is a widely accepted method for the development of Web applications [12] , whose first versions focused on design and did not include requirements engineering. The capture and definition of requirements were introduced later in OOHDM by Vilain, Schwabe and Sieckenius [13], proposing the use of user interaction diagrams (UIDs). UWE: UML-based Web Engineering UWE classifies requirements into two groups: functional and non-functional. Moreover, UWE proposes interviews, questionnaires and checklists as appropriated techniques for the requirements capture, and use cases, scenarios and glossaries for the requirements specification. To validate them, UWE proposes walk-through, audits and prototypes [14]. Design-driven Requirements Elicitation is a part of the design-driven process proposed by Lowe and Eklund [15] in order to develop Web applications. It consists of capturing, defining and validating requirements during the design process, i.e. The design activities should be carried out in such a way that the requirements could be handled and managed at the same time. The process is based on prototyping in order to explore possible solutions and problems to be solved. Users and customers define the requirements based on the study of these prototypes. It is an iterative process, which consists of reducing customers and clients' doubts. The cycle has three phases:

evaluation, specification and construction. In Haley and colleagues security requirements engineering framework [16] they have 4 steps to elicit and analyze security requirements but does not cover all the phases of requirements engineering and it is a complex process for the developers.

2.1. Object oriented Security Requirements Engineering

Object-oriented Requirements are a widely accepted method for the development of Web applications. Most Web applications are still developed in an ad hoc manner. One reason is the gap between established software design concepts and the low-level Web implementation model. So to have a good design and the Web implementation model the Security Requirements Engineering should be done in the early stage with the object oriented concepts. Identify all the objects of a Web application, and then develop the components with a higher level of abstraction. Security Requirements Engineering embodies object-oriented principles such as reuse, modularity, abstraction and encapsulation.

Most requirement documents were written in ambiguous natural languages which are less formal and imprecise. Without modeling the Security requirement documents, the knowledge of the requirement is hard to be kept in a way, which can be analyzed and integrated with artifacts in other phases of software life cycle. Therefore, maintaining the traceability and consistency of requirement documents and software artifacts in other phases is costly and error prone. This paper presents a systematic approach to eliciting and analyzing Object-Oriented Security Requirements based on models.

3. OVERVIEW OF MOSRE FRAMEWORK

The web application has become more and more critical in every domain of the human society. Transportation, communications, entertainment, health care, military, e-commerce, and education; the list is almost endless. These systems are used not only by major corporations and governments but also across networks of organizations and by individual users. Such a wide use has resulted in these systems containing a large amount of critical information and processes which inevitably need to remain secure. Therefore, although it is important to ensure that Web Applications are developed according to the user needs, it is equally important to ensure that these applications are secure.

However, the common approach towards the inclusion of security within a Web Application is to identify security requirements after analysis, means that security enforcement mechanisms have to be fitted into a pre-existing design, leading to serious design challenges that usually translate into the emergence of computer systems afflicted with security vulnerabilities. Recent research has argued that from the viewpoint of the traditional security paradigm, it should be possible to eliminate such problems through better integration of security and requirements engineering. Security should be considered from the early stages of the development process and security requirements should be defined alongside with the system requirements specification.

The Security Requirements Engineering is the process of eliciting, specifying, and analyzing the security requirements for system fundamental ideas like "what" of security requirements is, it is concerned with the prevention of harm in the real world and considering them as functional requirements. Many methods have been developed that facilitate this kind of requirements analysis and the development of security requirements. The internet has already created social and economic opportunities for people around the world. But even there are many Challenges to Web Applications Security like threats, attacks, Phishing spyware, worms, Trojans and virus which cause to denial of service hacking into and defacing web sites and destroying. Here we present the

proposed work; MOSRE [24] a Model Oriented Security Requirements Engineering Framework for Web Applications.

Our Framework shown in Figure 1 follows the spiral Framework model which is iterative and all phases of Requirements Engineering are covered in this Framework.

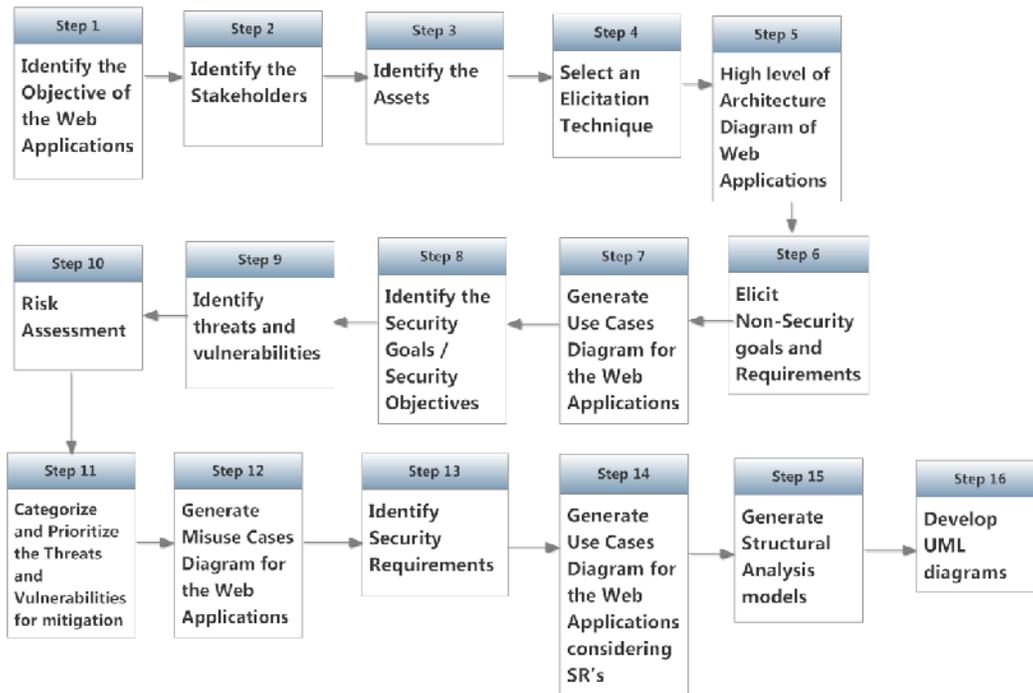


Figure 1 MOSRE Framework for Web Applications

3.1 Inception

Inception is to establish the ground work, before to start the elicitation and analysis of security requirements for web applications. Different steps are involved in the inception phase of MOSRE.

Step 1 Identify the Objective of the Web Applications

The Web Applications objective must be identified from the customer requirements who needs the Web Application. This step will help to understand the domain of the application that customer needs.

Step 2 Identify the Stakeholders

The identification of stakeholders plays an important role in security requirements engineering. The stakeholders include the Architect, developer, customers/end users, security experts, requirements engineering team and other interested people.

Step 3 Identify the Assets

The next step is to identify the assets of the targeted system. Assets may be business or system assets (e.g.: Data, money, and password). From our survey it is found that asset identification is

an important step in security requirements engineering. The assets should be identified in the context of the software system, so the objective of a software system is to be identified first. To identify the assets different techniques like interview, questionnaire, and brainstorming can be used. The stakeholders help in finding the assets. Assets should be viewed not only at developer or customer/end user perspective but also in attacker's point of view. Assets can be identified from existing documents.

Inception phase of security requirements engineering should be working with high level of collaboration and care.

3.2 Elicitation

The next phase in security requirements engineering is elicitation, the stakeholders and requirements engineering team will work together to identify the problem, propose the solution and specify the set of security requirements. There are different steps involved in the elicitation phase of security requirements engineering.

Step 4 Select an Elicitation Technique

The elicitation phase starts some ground work to be done by selecting the elicitation technique. Requirements elicitation is called as capturing, requirements discovery or requirements acquisition. The step of requirements elicitation can be complex, mainly if the problem domain is unknown for the analysts. Some of the elicitation techniques are, misuse cases, Issue Based Information Systems (IBIS), Joint Application Development (JAD), Interviewing, Brainstorming, Sketching and Storyboarding, Use Case Modeling and Questionnaire and Checklist A suitable method can be chosen from these elicitation techniques based on the requirements engineering community or expert's choice, level of the security to achieve, cost –effort benefit and organizational policies.

Step 5 High level of Architecture Diagram of Web Applications

With the objective of web application we can identify the number of tiers in the web applications. So draw a rough architecture diagram with high level of abstraction of the web applications. Network or hierarchical style of Architecture can be chosen based on the application domain. This diagram can be extended in detail with low level of abstraction in the next phase of design.

Step 6 Elicit Non-Security goals and Requirements

Once the business goals are identified, and then the non-security goals and requirements of the web applications are to be elicited. The collaborative requirement gathering is adopted to gather non-security goals and requirements. A general classification of requirements for Web applications are Functional requirements and Non Functional requirements. Functional requirements are capabilities that a system must exhibit in order to solve a problem.

Functional requirements for web applications and nonfunctional requirements that act to constrain the solution, e.g. Portability requirements; reuse requirements, usability requirements, availability requirements, performance requirements are identified.

The non-security requirements are categorized as essential and nonessential requirements and prioritized according to the Stakeholders preference.

Step 6 Generate Use Cases Diagram for the Web Applications

The non security requirements are gathered; for better understanding and then the use case modeling of the web applications should be developed. Use Case Modeling is a technique which was developed to define requirements [2]. A use case model consists of actors, use cases and relationships between them [3]. It is used to represent the environment by actors and the scope of the system by use cases (functional requirements). An actor is an external element of the system that interacts with the system as a black box. A use case describes the sequence of interactions between the system and its actors when a concrete function is executed. An actor can take part in several use cases and a use case can interact with several actors. The use case is the set of scenarios that encompass the non-security requirements of the system created by the developers and users of the system. In Figure 2 the Use Case Diagram for the e-store with actors and use cases are shown.

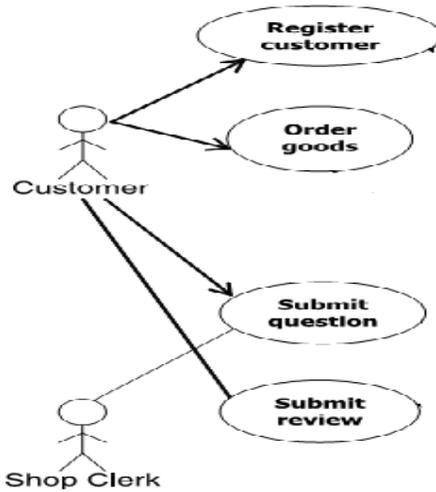


Figure 2 Use Case Diagram for the e-store

Step 7 Identify the Security Goals / Security Objectives

The security goals / security objectives can be identified with respect to assets, business goals and organizational principles are the security policies of the organization. The list of security goals can be identified and the security goals can be of the main goals and sub goals. The main goals are the top goals, e.g. Confidentiality, Integrity and Availability, that to be identified in the web applications based on the level of security we need. Security main goals [4] for web applications are:

Authentication

Authentication addresses the question: who are you? It is the process of uniquely identifying the clients of your applications and services. These might be end users, other services, processes, or computers.

Authorization

Authorization addresses the question: what can you do? It is the process that governs the resources and operations that the authenticated client is permitted to access. Resources include files, databases, tables, rows, and so on, together with system-level resources such as registry

keys and configuration data. Operations include performing transactions such as purchasing a product, transferring money from one account to another, or increasing a customer's credit rating.

Auditing

Effective auditing and logging is the key to non-repudiation. Non-repudiation guarantees that a user cannot deny performing an operation or initiating a transaction. For example, in an e-commerce system, non-repudiation mechanisms are required to make sure that a consumer cannot be denied of ordering 100 copies of a particular book.

Confidentiality

Confidentiality, also referred as privacy, it is the process of making sure that data remains private and confidential, and that it cannot be viewed by unauthorized users or eavesdroppers who monitor the flow of traffic across a network. Encryption is frequently used to enforce confidentiality. Access control lists (ACLs) are another means of enforcing confidentiality.

Integrity

Integrity is the guarantee that data is protected from accidental or deliberate (malicious) modification. Like privacy, integrity is a key concern, particularly for data passed across networks. Integrity of data in transit is typically provided by using hashing techniques and message authentication codes.

Availability

From a security perspective, availability means that the systems remain available for legitimate users. The goal for many attackers with denial of service attacks is to crash an application or to make sure that it is sufficiently overwhelmed so that other users cannot access the application. There are many security sub goals/objectives for web applications and are based on the application domain and security policy of the organization, e.g. Prevent attackers from obtaining sensitive customer data, including passwords and profile information which comes under confidentiality. Prevent tampering, trail and access control which comes under the top security goal Integrity. The techniques like Facilitated Application Specification Technique (FAST), survey and interviews can be used to identify the security goals / security objectives.

Step 8 Identify threats and vulnerabilities

By identifying the assets, business goals and security goals the threats to the web applications can be identified. The overall system threats and vulnerabilities can be identified during this step. The list of threats and vulnerabilities can be developed for the web applications. The main threats to a Web application are: Profiling, Denial of service, Unauthorized access, Arbitrary code execution, Elevation of privileges, Information gathering, Sniffing, Spoofing, Session hijacking, SQL injection, Network eavesdropping, Password cracking, Viruses, Trojan horses, and worms. Some of the vulnerabilities to the web application are unnecessary protocols, Open ports, Web servers providing configuration information in banners, Weak IIS Web access controls including Web permissions, Weak NTFS permissions, Poor input validation in your Web applications, Unsafe, dynamically constructed SQL commands, Weak or blank passwords, and Passwords that contain everyday words.

Step 9 Risk Assessment

The next step is to assess and determine the risk when the threats and vulnerabilities occur. The impact of threats and vulnerabilities are analyzed and risk determination process [20] is carried out. To do risk determination process any of risk assessment test models [5] like National

Institute of Standards and Technology (NIST) model, NSA's INFOSEC Assessment Methodology, Butler's Security Attribute Evaluation method (SAEM) , CMU's "V-RATE" method , Yacov Haimès's RFRM model can be used or Microsoft risk based on DREAD method [6] can be used.

Step 10 Categorize and Prioritize the Threats and Vulnerabilities for mitigation

The threats and vulnerabilities can be Categorized with respect to the security goals and security policies of the organization and prioritized based on the level of security and assets to be secured. This step can be done with the help of a survey or interview between the stakeholders.

Step 11 Generate Misuse Cases Diagram for the Web Applications

The detailed set of misuse case diagram [7] of the web applications should be developed that encompass the most significant threats to the system. In the Figure 3 Misuse Case Diagram for the e-store with Hacker as an actor and Misuse cases are shown.

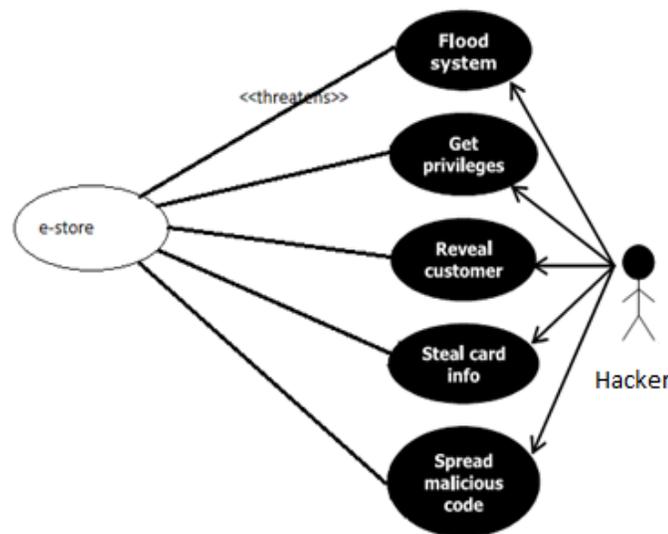


Figure 3 Misuse Case Diagram for the e-store

Step 12 Identify Security Requirements

The security requirements [21] are the countermeasures that the Web Applications should have, as the functional requirements.

Step 13 Generate Use Cases Diagram for the Web Applications considering Security Requirements

The security requirements are gathered; for better understanding, the use case diagram of the Web Applications should be generated, that encompasses the security requirements of the system created by the developers and users of the system. In Figure 4 Use Case Diagram for the e-store considering Security Requirement is shown.

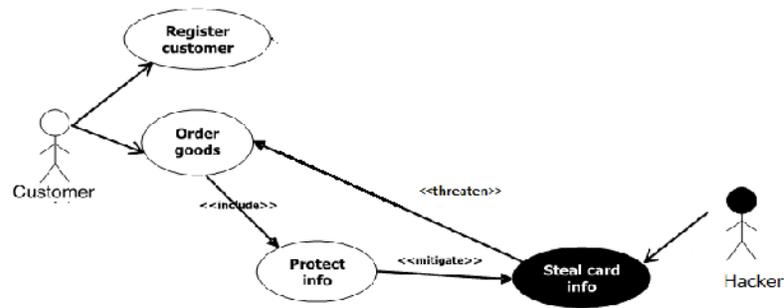


Figure 4 Use case with security requirement

3.3 Elaboration

In this phase the detailed view of the web applications with security requirements can be understood with models and diagrams, which gives a clear idea of the application in design and implementation phase.

Step 14 Generate Structural Analysis models

The next step of security requirements engineering is to develop different analytical models. These models form the solid foundation for the design of security requirements. The data models, flow models and behavioral models are the structural analysis models that can be used to show the functional requirements and data flow.

Step 15 Develop UML diagrams

Develop UML diagrams for detailed view of security requirements and for better understanding of the secure web applications. The high level of class diagram and sequence diagrams can be developed. These diagrams can be used to generate code and test cases for testing the security requirements. The navigational model consists of a navigation class diagram and a navigation structure diagram. Security based models can be developed using SecureUML [23] and UMLsec [22].

3.4 Negotiation and Validation

In this phase the security requirements are categorized as essential and nonessential requirements and prioritized according to the level of security and Stakeholders preference of security requirements. Then rough effort time and cost are estimated to implement security requirements. The validation is done by the security experts and engineers with the requirements of the stakeholders. Review or Walk-through is a technique which consists in reading and correcting the requirements definition documentation and models. Such a technique only validates the good interpretation of the information. Traceability Matrix consists of a comparison of the application objectives with the requirements of the system [8]. A correspondence is established between objectives and how they are covered by each requirement. This way, inconsistencies and non-covered objectives will be detected.

3.5 Specification

The specification is the last phase in security requirements engineering Framework. The security requirements specifications are modeled and they are validated with the stakeholders and this

specification forms the source for the design of security requirements. This phase is executed in parallel with each other phases of requirements engineering. Scenario or use case modeling can be used to specify the functional requirements with security requirements and non functional requirements for web applications.

In this MOSRE Framework, object modeling is used to model the components of the web applications and the concept of encapsulation with the functionality and data in the data model. The reusability of some of the security requirements against different threats, and the functions can be extended to implement the security requirements; the concept of inheritance is adopted here.

4. IMPLEMENTATION OF MOSRE FRAMEWORK TO A CASE STUDY

Manual voting systems have been deployed for many years with enormous success. If those systems were to be replaced with Electronic Voting Systems, we have to be absolutely sure that they will perform at-least as efficient as the traditional voting systems without any security issues. Failures or flaws in Online Voting Systems will put at risk to Democracy in the country implementing them. The main focus of security requirements engineering is on defining and describing what a software system should do to satisfy the informal requirements provided with a statement of need. In this paper, we will define and describe what the secure Online Voting System should do to ensure a secure, robust, accurate, secure and quality-based design and implementation.

Security Requirements are defined during the early stages of system development as a specification of what level of security should be implemented. In other words, they represent what the system should do and have security from the stakeholders' point of view. Performing a good security analysis on E-Voting web application is an essential step in order to guarantee a reasonable level of protection. However, different attacks and threats may be carried out depending on the operational environment in which the system is used.

An E-Voting System should consider the following minimum requirements:

1. To ensure that only persons with the right to vote are able to cast a vote.
2. To ensure that every vote cast is counted and that each vote is counted only once.
3. To maintain the voter's right to form and to express his or her opinion in a free manner, without any coercion or undue influence.
4. To protect the secrecy of the vote at all stages of the voting process.
5. To guarantee accessibility to as many voters as possible, especially with regard to persons with disabilities.
6. To increase voter confidence by maximizing the transparency of information on the functioning of each system.

The MOSRE Framework was implemented in E-Voting web application to gather functional requirements which include security requirements. The Figure 5 shows the architecture of the E-Voting System.

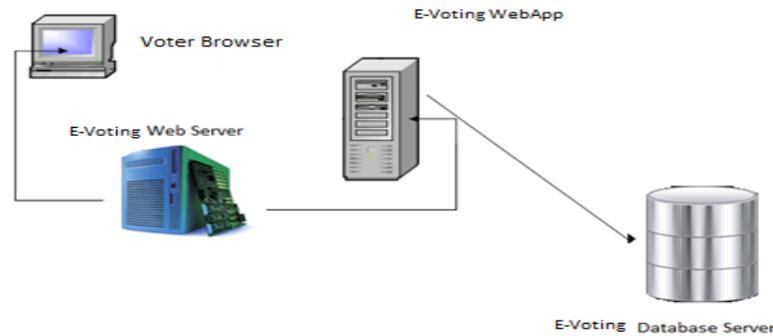


Figure 5 E-Voting Architecture

Each step of the MOSRE Framework was applied to E-Voting web application. The list of the E-Voting System security requirements based on business assets and system assets can be identified.

5. RESULT ANALYSIS AND DISCUSSION

In the previous section we have identified the list of some security requirements and they are based on the business and system assets by applying the MOSRE Framework for Online Voting system. Based on the identified list of threats, vulnerabilities and security requirements we found that using our MOSRE Framework for web applications we will be able to get a better set of security requirements. There are many methods to elicit security requirements but concentrating less on the phases of requirements engineering [15, 16, 17, 20 and 22]. In this section we compare results obtained from MOSRE Framework, Haley and colleagues security requirements engineering framework [11] and without using security requirements engineering using a chart. We consider the percentage of vulnerabilities, threats and security requirements found with each method as the parameters for comparison.

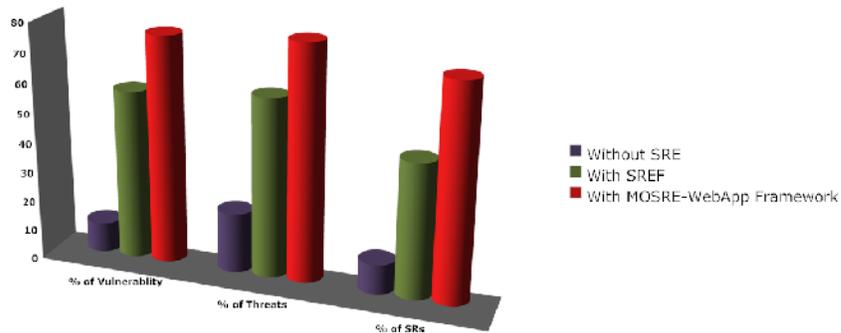


Figure 6 Percentage of vulnerability, threats and security requirements identified by MOSRE Framework, SREF and without using SRE

The Figure 6 depicts the percentage of vulnerability, threats and security requirements identified using MOSRE Framework are high than Haley’s SREF and without using any SRE. Figure 7 plots the percentage of vulnerability identified in E-Voting System implemented using MOSRE Framework with without using any SRE methods after the first iteration of the testing phase of the E-Voting application. It is clear from the chart that the number of vulnerabilities will be increased when no security requirements engineering method is adopted.

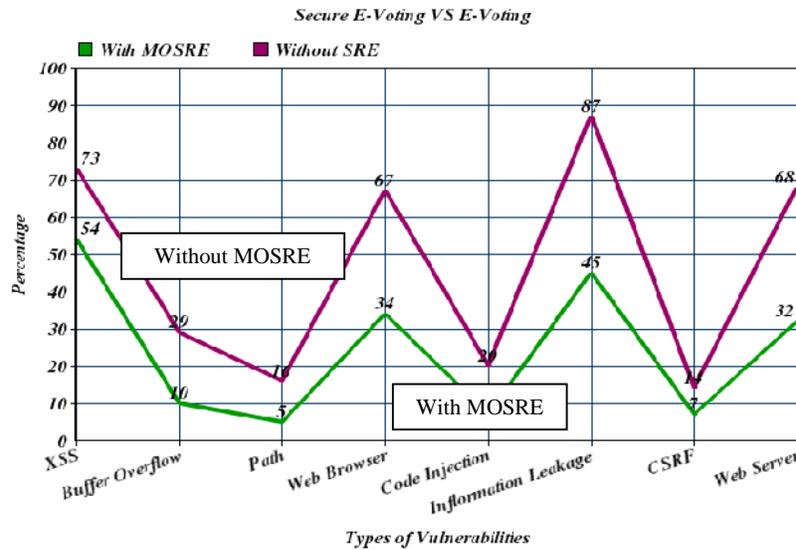


Figure 7 Percentage of vulnerabilities identified by with and without MOSRE

From a technical point of view, the most difficult task of the methodology is where security objectives are identified from functional descriptions, such as functional requirements. This has been the observation from several projects using the MOSRE Framework to elicit security requirements. MOSRE Framework requires expertise on at-least three dimensions: (i) information structuring and analysis, (ii) requirements engineering, and (iii) security. It is rarely intuitive what the overall security goals and objectives are, and it is not easy to simply extract these from highly abstract system information, incomplete sets of functional Requirements and early draft system architecture. MOSRE Framework provides some support, with use case, misuse case models.

6. CONCLUSION

Security Requirements have to be considered in the early phase of Requirements Engineering [12, 13, and 14], so a Model oriented Security Requirements Engineering framework is developed for Web Application and evaluated for an E-Voting Web Application, The main aim of MOSRE is to extend security requirements engineering by seamlessly integrating elicitation, traceability and analysis activities. The motivation for this is that requirements engineering activities are often executed by other people than those writing the code, and often without much contact between the two groups. This applies in particular to security requirements, which is a major quality, attribute of today's system. It is therefore important to develop both the ability of the people involved in the development to identify potential security aspects, and the capabilities of the development team to solve these needs in practice through secure design.

As future work the Security Requirements identified from RE Phase should be carried in the design phase because good design will give Vulnerability free Web Applications and implement them. We also intend to do penetration testing and find the results based how far our application is vulnerable.

ACKNOWLEDGEMENTS

We would thank everyone for their valuable suggestion to do this research work.

REFERENCES

- [1] CLUSIF, Web Application Working Group, "Web application security, managing web application security risks", Technical Studies, <http://www.clusif.asso.fr/>, March 2010.
- [2] Jacobson, I. (1995). Modeling with Use Cases: Formalizing Use Case Modelling. Journal of Object-Oriented Programming.
- [3] UML (2003). Unified Modeling Language. Version 1.5. www.omg.org
- [4] J.D. Meier, Alex Mackman, Michael Dunner, Srinath Vasireddy, Ray Escamilla and Anandha Murukan , "Improving Web Application Security :Threats and Countermeasures", Microsoft Corporation, Published: June 2003
- [5] R. Mead, E.D. Houg, and T.R. Stehney, Security Quality Requirements Engineering (Square) Methodology, tech. report CMU/SEI-2005-TR-009, Software Eng. Inst., Carnegie Mellon Univ., 2005.
- [6] Swiderski, Frank, Syndex, "Threat Modeling", Microsoft Press, 2004
- [7] Guttorm Sindre, AndreasL.Opdah," Eliciting security requirements with misuse cases". Requirements Eng (2005)10:34–44, Springer-Verlag London Limited 2004.
- [8] M. José Escalona, Nora Koch. "Requirements Engineering for Web Applications – A Comparative Study", Journal of Web Engineering, Vol. 2, No.3 (2004) 193-212, Rinton Press.
- [9] Lee, H., Lee, C., Yoo, C. (1998). A Scenario-based Object-oriented Methodology for Developing Hypermedia Information Systems. Proceedings of 31st Annual Conference on Systems Science.
- [10] Bieber M., Galnares, R., Lu, Q. (1998). Web Engineering and Flexible Hypermedia. The Second Workshop on Adaptive Hypertext and Hypermedia, Hypertext'98, Pittsburg, USA.
- [11] Olsina, L. (1998). Building a Web-based Information System applying the Hypermedia Flexible Process Modeling Strategy. 1st International Workshop on Hypermedia Development, Hypertext'98, Pittsburg, USA.
- [12] Schwabe D., Rossi G. (1998). Developing Hypermedia Applications using OOHDM. Workshop on Hypermedia Development Process, Methods and Models, Hypertext'98, Pittsburg, USA.
- [13] Vilain, P., Schwabe, D., Sieckenius, C. (2000). A diagrammatic Tool for Representing User Interaction in UML. Lecture Notes in Computer Science. Proc. UML'2000. York, England.
- [14] Koch, N. (2001). SoftwareEngineering for Adaptive Hypermedia Applications. Ph. Thesis, FAST Reihe Softwaretechnik Vol (12), Uni-Druck, Munich, Germany.
- [15] Lowe D., Eklund J. (2002). Client Needs and the Design Process in Web Projects. Web Engineering Track of the WWW2002 Conference.
- [16] C.B. Haley, R. Laney, J.D. Moffett, and B. Nuseibeh, "Security Requirements engineering: A Framework for Representation and Analysis," IEEE Transaction on Software Eng. Vol 34, no. 1, pp. 133-152, Jan/Feb 2008.
- [17] Eric Dubois , Haralambos Mouratidis, "Guest editorial: security requirements engineering: past, present and future", Requirements Eng (2010) 15:1-5, Published online: 1 January 2010, Springer-Verlag London Limited 2009.
- [18] Benjamin Fabian , SedaGurses , Maritta Heisel,Thomas Santen • Holger Schmidt," A comparison of security requirements engineering methods", Requirements Eng (2010) special issue security requirements engineering ,15:7-40, Published online: 26 Nov 2009, Springer-Verlag London Limited 2009.
- [19] Siv Hilde Houmb , Shareeful Islam ,Eric Knauss • Jan Jurjens • Kurt Schneider," Eliciting security requirements and tracing them to design: An integration of Common Criteria, heuristics, and UMLsec Requirements Eng (2010) special issue security requirements engineering ,15:63-93, Published online: 28 Nov 2009, Springer-Verlag London Limited 2009.
- [20] Dhirendra Pandey, Ugrasen Suman ,A. K. Ramani,"Security Requirement Engineering Issues in Risk Management ", International Journal of Computer Applications (0975 – 8887)Volume 17– No.5, March 2011,pg:12-14.
- [21] Donald Firesmith: "Engineering Security Requirements", in Journal of Object Technology, vol. 2, no. 1, January-February 2003, pages 53-68. http://www.jot.fm/issues/issue_2003_01/column6
- [22] J.Jurjens.Umlsec:Extending uml for secure systems development. In ProcofUML'02, pages 412-425.Springer,2002.
- [23] T.Lodderstedt,D, A.Basin,and J.Doser, "Secureuml: A uml-based modeling language for model-driven security. In UML'02:Proceedings of the 5th International Conference on The Unified Modeling Language,pages 426-441,London,UK,2002.Springer-Verlag.
- [24] P. Salini and S. Kanmani. "Model Oriented Security Requirements Engineering (MOSRE) Framework for Web Applications". In Proceedings of the Second International Conference on Advances in Computing and Information Technology (ACITY 2012), India, July 13 - 15, 2012, Vol.2 and in Advances in Intelligent and Soft Computing book Series, Vol.177, pp.341-353.