IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 9, Issue 9, September 2020

DOI 10.17148/IJARCCE.2020.9901

# Prove the Importance of MOSRE Framework to Elicit Security Requirements at Early Stages of SDLC to Minimize Vulnerabilities at Later Phases in Developing Web Applications

**Dr. Osama Ahmad Salim Safarini**[1]

Computer Science and Engineering Researcher, Tulkarem, Palestine[1]

**Abstract:** To develop security critical web applications, specifying security requirements is important, since 75% to 80% of all attacks happen at the web application layer. I adopted security requirements engineering methods to identify security requirements at the early stages of software development life cycle to minimize vulnerabilities at the later phases. In this paper, I present the evaluation of Model Oriented Security Requirements Engineering (MOSRE) framework and Security Requirements Engineering Framework (SREF) by implementing the identified security requirements of a web application through each framework while developing respective web application. I also developed a web application without using any of the security requirements engineering method in order to prove the importance of security requirements engineering phase in software development life cycle. This study led the requirements engineers to use MOSRE framework to elicit security requirements efficiently and trace security requirements from requirements engineering phase to later phases of software development life cycle for developing secure web applications.

**Keywords:** Requirements engineering, security mechanism, security requirements, security requirements engineering, web applications and vulnerabilities.

## I. INTRODUCTION

MY hypothesis that "If security requirements for software systems are considered as functional requirements and elicited in the early phase of Software Development Life Cycle (SDLC) then vulnerabilities can be minimized than the software system developed without Security Requirements Engineering (SRE) phase". Early awareness improves the integration of security during requirements development and thus produces a more secure system versus having to retrofit/fix the system.

Requirements engineering is the first phase of the (SDLC). In this phase, the customer and developer come to an agreement about the software to be developed. This is a critical part of development and good requirements engineering is therefore essential for successful software system development. Security requirements have received far less attention than general requirements [1]. I argue that security requirements should receive similar attention as business requirements. For instance, the security picture is complicated in web applications since they are often written in high-pressure environments on tight schedules by developers who have little or no security knowledge. Once development is complete, the applications are put through quality assurance testing that focuses on performance and functionality, rather than security. It is no surprise, which Gartner reported in [2] that 75% of hacks happen at web sites target the application level than network, database and web server layers. Nowadays, the organizations, employees and customers prefer to do business online, and expect to be able to access a variety of information and transactions through web sites and services. As a result, web applications hold the treasure of data behind their front ends: like credit card numbers, health care records, confidential financial results, the list goes on. The attackers are well aware of the valuable information accessible through web applications, and their attempts to get at it, who have figured out thousands of ways to penetrate web applications. Attackers exploit vulnerabilities to compromise the system, which is the weakness of web application or its environment in conjunction with an internal or external threat that lead to a security failure. This is due to that, the vulnerabilities are used rarely to elicit security requirements of web applications.

In recent years, software and government sectors are aware of security risks that vulnerabilities impose on the web applications and have started analysing and reporting detected vulnerabilities in web applications. For instance, the Context Information Security, London, in a statistics [3], "Web Application Vulnerability Statistics 2010-2011" shows the average number of vulnerabilities identified within a web application affects various ranges of business sectors in 2010 and 2011. To solve these issues, I propose to adopt Security Requirements Engineering (SRE) framework in the early phases of SDLC and to elicit Security requirements for web applications. Though a variety of SRE approaches have

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 9, Issue 9, September 2020

DOI 10.17148/IJARCCE.2020.9901

been proposed by the researchers, they lack attention and support from the requirements engineers to elicit and specify security requirements. This is due to the lack of exposure and skills on SRE and thus security requirements are identified in the later phases of SDLC. In this paper, I evaluate the performance of existing SRE frameworks such as Model Oriented Security Requirements Engineering (MOSRE) and Security Requirements Engineering Framework (SREF), and downside of not using SRE method, to recommend the requirements engineers for a better approach to elicit and specify security requirements [4]. The remaining part of the paper is organized as follows: section 2 gives the related works to SRE methods. Section 3 presents the overview how MOSRE framework is applied to identify security requirements for a web application. Section four, discusses about the implementation and evaluation of identified security requirements. The experimental setup is given in section 5. The analysis of experimental results in sections 6 and 7 presents with discussions. At last, section 8 concludes with future enhancement.

## II. RELATED WORKS

The research community has looked into security issues of web applications with utmost care and brings method to incorporate security at various levels. The most accepted approach is to incorporate security in the software development cycle. Secure modelling and model driven development approach are becoming popular. The protection could be done at the network level, operating system level, database level or application level. As the web applications with vulnerabilities have been exploited by hackers, application scanning is the significant method to assess the security issues. In this section, I discuss about various approaches and the significant work done for web application security. Jűrjens [5] defines model security concepts and focus on the importance of incorporating security concerns during the SDLC. Jűrjens [5] also explores the needs of developing secure-critical systems as there are many security weaknesses exploited. He proposes a systematic methodology to aid developing security- critical systems based on the Unified Modeling Language (UML). The extension of UML, UML sec [6] Allows expressing security-relevant information within the diagrams in a system specification. This UMLsec approach can be used in design phase to model security requirements with low level of abstraction.
Security analysis framework by Fu et al. [7] proposes a static approach to Structured Query Language (SQL) injection identification for the testing phase. The work proposes a framework that uses Compile time vulnerability detection. Ismail et al. [8] discuss on the Cross-Site Scripting (XSS) vulnerability. The client side XSS are detected and solved in testing phase. Scott and Sharp [9] suggest some ways to protect web applications. They illustrated the difficulties in adding security to web applications and this approach is applied on coding level.
Many scanning tools are available to assess the vulnerabilities for web application in the testing phase. Kals and Kirda [10] discusses that many web application security vulnerabilities result from generic input validation problems. They presented SecuBat, a generic and modular web vulnerability scanner that analyses web sites for exploitable SQL and XSS vulnerabilities. The SecuBat also has a crawling component to determine the doors of attack and attack plugins are used to detect them.
The Acunetix web vulnerability scanner is a commercial tool available to assess the vulnerability of applications and find how far a web application is vulnerable. In aspect-oriented approach, security models can be designed separately and then weaved to web applications. Fuentes and Sanchez [11] introduces an approach that can be used to weave multiple aspects into the executable UML model. This work throws light on the design phase.
Model driven web application development approach enables secure web application design and code generation. Koch and Kraus in [12] propose a methodology for web application development. It uses UML as the base modeling language and defines stereotypes for modeling the domain specific aspects. They have given a very general approach to web application development and security issues. Model driven security for process-oriented systems' by Lodderstedt et al. [13] has shown how model driven paradigm can be adapted to introduce security. Secure UML [13] is used to specify the access control policies. There are many requirements engineering methods [14] For web applications, but they are applicable for whole SDLC and consider security as one of the non- functional requirements. They also consider security analysis as the part of the design and implementation phase, which results in system failures.
From the survey, I understand that very few works have been done on SRE for web applications. Since web applications are more prone to vulnerabilities and failures, good requirements analysis and specification should be developed to solve the security issues. This led me a motivation and frame our hypothesis that "security requirements for web application should be elicited and analysed in the early phase of SDLC and to be considered as functional requirements".
Extensive work has been carried out on security requirements during the last few years, and there are several works that deal with security requirements [15] in the early stages of the development life cycle. Security Quality Requirements Engineering Methodology (SQUARE) [16] is a model made up of nine steps in which it provides a means of eliciting, categorizing and prioritizing security requirements for information technology systems and applications. The evaluation of SQUARE was conducted in [17] on the advanced metering infrastructure of the smart grid as a case study. The effectiveness of SQUARE with respect to its ability to elicit a set of artifacts, threats, and vulnerabilities; to perform likelihood, impact analysis, and risk level determination; and to elicit, categorize, and prioritize the security requirements

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 9, Issue 9, September 2020
DOI 10.17148/IJARCCE.2020.9901

are evaluated [17]. SQUARE methodology is better useful to assess the quality and document the elicited security requirements rather to elicit security requirements.

### i.Overview of SREF and MOSRE

In SREF [18], security requirements were defined as constraints over functional requirements by Haley et al [18]. SREF consists of four steps, which are executed, in iterations. They consider context as an important factor having a deep effect on security requirements. The framework is one of the recent SRE methods, which have four activities performed in iteration. They are:

- Stage 1: Identify Functional Requirements.
- Stage 2: Identify Security needs.
- Stage 3: Identify Security Requirements.
- Stage 4: Verification of the System.

These steps are discussed in detail in [18], which has been implemented to elicit security requirements of a web application in [19]. The results were taken as a reference to perform our evaluation of SRE frameworks in this paper. I found many limitations within this framework they lack in risk analysis, categorizing and prioritizing threats and vulnerabilities, the artifacts are very complex to the developers and not suitable to elicit security requirements of a web application.

MOSRE [20] activities are partially based on steps of SQUARE. MOSRE is framework with steps to identify assets, threats and risks for the establishment of security requirements in the development of secure web applications and whose focus seeks to build security concepts in the early phases of the development life cycle. This framework describes to express security requirements as use cases, and threats expressed as misuse cases. MOSRE framework has 16 steps to elicit security requirements, they are:

- Step 1. Identify the Objective of the Software Systems.
- Step 2. Identify the Stakeholders.
- Step 3. Identify the Assets.
- Step 4. Select an Elicitation Technique.
- Step 5. High level of Architecture Diagram.
- Step 6. Elicit Non-Security needs and Requirements.
- Step 7. Generate Use Cases Diagram.
- Step 8. Identify the Security needs / Objectives.
- Step 9. Identify Threats and Vulnerabilities.
- Step 10. Risk Assessment.
- Step 11. Categorize and Prioritize the Threats and Vulnerabilities for mitigation.
- Step 12. Generate Misuse Cases Diagram.
- Step 13. Identify Security Requirements.
- Step 14. Generate Use Cases Diagram considering Security Requirements.
- Step 15. Generate Structural Analysis models.
- Step 16. Develop UML diagrams.

In order to show the benefit of identifying security requirements, three web applications were developed, first web application by adopting MOSRE, second with SREF and third without using any SRE method. The developed web applications were scanned for number of vulnerabilities as an attempt to assess the performance of SRE frameworks and prove the importance of SRE phase.

## III. APPLICATION OF THE MOSRE FRAMEWORK

In this Section, a practical example of how MOSRE framework can be used in the elicitation and analysis phases for identifying security requirements for Electronic-voting (E-voting) system. The E-voting systems are highly sensitive in nature, and they are a prime target for biasing the results of an election. An attacker on an E-voting system might be tempted to exploit any software vulnerability in these systems to break the integrity and secrecy of ballots. Some of the current E- voting systems suffer from exploitation of vulnerabilities [21].

These voting systems are highly dependent on the security of the software and therefore they are vulnerable to any flaw in the security requirements analysis and design. Web-based voting are providing organizations with more flexibility to conduct their internal elections, which can be extended to political elections. For deploying E-voting systems, web technologies can be employed to protect systems on the server side. However, such technologies are limited on the client side, which may make systems vulnerable to different attacks. The impact of both server and client security on the confidentiality, integrity and availability of the E-voting systems must be carefully considered. Conducting elections for public over the Internet raises grave security risks.

The E-voting system needs to maintain both integrity of the election result and secrecy of the voters' choices; it must remain available on the network, and serve voters connecting from untrusted clients. Many security researchers have

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 9, Issue 9, September 2020

DOI 10.17148/IJARCCE.2020.9901

found different threats to E-voting [22, 23], others have proposed systems and protocols that may be solutions someday [24, 25]. Analysing the security requirements in E- voting is an essential task to guarantee adequate security levels, as threats and vulnerabilities may not only derive from pitfalls in the electronic systems, but also from the web applications. These reasons motivated us to take an E-voting system as the case study to apply MOSRE framework and the overview of the application [26] to E-voting is presented in this section.

The objective of the E-voting system is a system in which the election data is recorded, stored and processed primarily as digital information. E-voting system development should be based on the multilateral view of the stakeholders, so I should include people from the voters' community, the candidate, security experts, election officers, government representatives, developers and requirements engineering team. The business assets are voters and candidates' details, votes, voter's credentials, voters secret, the number of votes casted for each candidate and the system assets are application software, database, network, server, web voters systems.
The brainstorming technique can be used for elicitation of requirements for the E-voting system. With the objective, I can identify the number of tiers in the applications. A rough architecture diagram can be drawn with high level of abstraction. Network or hierarchical style of architecture can be chosen based on the application domain. The high level of architecture diagram [27] shown in the Figure1 for E- voting system is obtained to analyse the data flow and the entry points to the system.

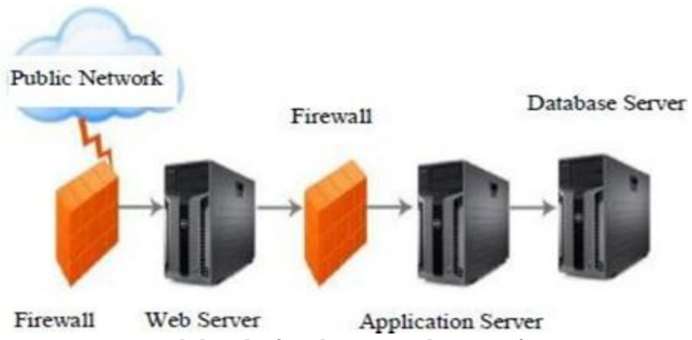

Figure 1. High level of architecture diagram for E-voting.

The next step is to elicit non-security needs and requirements. The following are some of the requirements for the E-voting system collected from the stakeholders.
1. Vote Casting:
- The voters must identify themselves in order to vote.
- All possible choices must be displayed for the voters.
- Record the selection of individual vote choices for each contest. Indicate that a selection has been made or cancelled.
- Notify the voter when the selection is completed.
- Before the ballot is cast, the voter is allowed to review his choices and, if he desires, to delete or change his choices before the ballot is cast.
- Prevent the voter from changing his casted vote.
2. Ballots Tally:
- Vote tally will occur once the polls have officially closed.
- Check for total votes for each candidate.

The requirements (i.e., business requirements) are categorized as essential and nonessential requirements and prioritized according to the stakeholder's preference. After the business requirements are gathered, for better understanding the use case modelling of the applications should be developed. The security needs / security objectives can be identified with respect to assets, business goals and organizational principles of the organization. The authentication, confidentiality, integrity, availability, accuracy, anonymity democracy, and auditability are some of the security needs identified with the help of the stakeholders to elicit the security requirements for the system.

Table I. List of threats to E-voting system.

| |
|---|
| Authentication Token from the smart card can be forged. |
| A malware accesses to the selected voting options at the Voter PC |
| Voter Impersonation and Vote Casting |
| Man-in-the-Middle Voter Contest Modification between the electoral roll service and the authentication service |
| A malware modifies the client application at the Voter PC |
| Authentication Server redirection to a fake Electoral Roll |
| Man-in-the-middle- Ballot template modification between voter and voting servers |

| |
|---|
| A malware modifies the voting options at the Voter PC |
| Change of Vote while storing |
| Denial of Service attack over the Voting Platform |
| Decrypted storage of votes modification at Counting |
| Controlled environment Authentication Token reply attack |

The threats and vulnerabilities to the applications can be identified with the identified assets, business goals and security needs. The stakeholders and the high-level architectural diagram identify the threats and vulnerabilities for the E-voting system at network, application and database levels. The list of threats and vulnerabilities can also be gathered and identified for E-voting system from the standards like Online Web Application Security Projects (OWASP) [28], National Vulnerability Database (NVD) [29 , 30] and Web Application Security Consortium (WASC) [31]. Table 1 shows some of the identified threats from [32] for the E-voting system
The next step is to assess and determine the risk when the threats and vulnerabilities occur. The impact of threats and vulnerabilities are analysed and risk determination process is carried out. Risk assessment can be performed using various techniques such as attack trees [33] and anti-models. In [34] Microsoft method of risk analysis for the E-voting system was performed by the authors. The threats and vulnerabilities can be categorized with respect to the security needs and security policies of the organization. They can be prioritized based on the level of security and value of the assets.
The detailed set of misuse case diagram [35] of the web applications should be developed that encompass the most significant threats to the system e.g., tamper misuse case, unauthorized users misuse case. The security requirements for E-voting system is identified based on the business and system assets, which are the countermeasures implemented with the applications. This process is repeated for a certain number of iterations based on the level of security to be achieved. The security requirements are gathered; for better understanding, the use case diagram of the applications that encompasses the security requirements of the system is generated. The security requirements can be categorized with the security needs. Some of the identified Security Requirements (SR) is shown in the Table 2 for the E-voting system.

Table II. Security requirements for E-voting system.

| |
|---|
| SR 1 Authorization shall be required to insert, delete, or modify any votes in the E-voting system. |
| SR 2 It shall be ensured that the E-voting system presents an authentic Ballot to the voter. |
| SR 3 The solution for voting in an uncontrolled environment shall issue A message to inform the voter whether the vote has been successfully cast. |
| SR 4 The E-voting system shall provide the e-voter with 'end-to-end' Proof that the casted vote is received and recorded. |
| SR 5 The E-voting system shall ensure that the voter’s choice is accurately represented in the vote and that the sealed vote is successfully Stored. |
| SR 6 To allow for a delay in messages when passing over the election channel, the acceptance of electronic votes into the E-voting system. Shall remain open for a configurable period after the end of the polling phase. |
| SR 7 The voter can vote at any time up to the point of vote casting, abort His polling process without losing his right to vote due to timeout or errors during communication. |
| SR 8 A voter shall only be able to vote in contests that he/she is entitled to vote in. |
| SR 9 The E-voting components of the E-voting system shall be Configurable to authenticate for contest, vote and session. |
| SR 10 The voter authentication shall expire after an idle period. The Length of the idle time-out period shall be configurable. |
| SR 11 The E-voter’s decision or the display of the e-voter’s choice shall Be destroyed after the vote has been cast. |
| SR 12 It shall not be possible during transfer in the network, or between System modules, to alter, delete or add vote records undetected. |

The functional requirements considering security requirements, the use case diagrams considering security requirements and the UML diagrams for the E-voting system is generated with high level of abstraction. The steps of MOSRE can be repeated for iterations in the elaboration phase of security requirements engineering and low level of abstraction models can be obtained.

## IV. OUTCOMES OF UNDERTAKING SRE

The methodology I used for evaluating MOSRE Framework is by implementing the web application with the identified security requirements and implementing the web application using SREF and without using any of the security

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 9, Issue 9, September 2020

DOI 10.17148/IJARCCE.2020.9901

requirements engineering methods. I scanned three systems for vulnerabilities using the web vulnerability-scanning tool and compared the results. This method of evaluation will help me to prove that security requirements should be given equal importance like business requirements and security requirements must be considered as functional requirements. It also proves that they should be analysed in the early phases of the software development life cycle and not at the time of design or coding. In this Section, I discuss about a part of the security requirements implemented for an E-voting system. First, I discuss on the authentication and access control security mechanisms for the web application. Next, I give a view on other security mechanisms adopted for E-voting system environment

**i.Implementation of Identified Security Requirements**

I implemented the security requirements identified by MOSRE framework for E-voting system into a software web application system. For implementation, I used the Java/ Java 2 Platform Enterprise Edition (J2EE) technology in windows platform. The server used was Apache Tomcat Server and oracle for database. The Acunetix web vulnerability scanner was used to scan the vulnerability.

The security requirements with the business requirements were implemented using encryption standards: Advanced Encryption Standard (AES) for managing the data in and out of the database and the data are in encrypted form. The encryption and decryption Java class are isolated from the web application.

**ii.Authentication and Access Control**

In an E-voting system authentication and access, control is the most important security objectives since it affects the democracy of the country. In order to satisfy the security requirements like “secure votes” and to “avoid duplicate votes”, the authentication should be provided based on the roles of the user. For example, if the actor of the system is a voter then he is provided with a single time authentication password to cast vote. The J2EE Servlet filter is one of the web modules in J2EE technology. It can intercept the request and responses on the website. It checks data transmitted between voter client and the server. The voter authentication can be realized after the user inputs his username and one time password.

I used J2EE Servlet filter to filter the user request and only the legitimate voter can login and access the required web pages to cast vote. The password Cracking is not possible because the login page is not available for others those who are on the network. Only the election officer will be provided rights for accessing the register page, and only a particular system will be able to access this page, by this I mean that the authority for accessing the page is only the person on the server machine.

The username and password for the election officer is stored in encrypted form and he is forced to change the password periodically. The username and password given by the election officer are taken into the Servlet, they are encrypted, and then it has to be checked in the database by using prepared statement class in Java for avoiding SQL injection. If any of the unauthorized users access the server system, they will be displayed with the error message and if the session is expired or caught, the login page automatically redirects to the index page. The parameters from the login page will be processed by the login Servlet and based on the access rights, i.e., the access control and the client is returned with corresponding pages.

**iii.Other Security Mechanisms for E-voting System**

The malicious code injected will not be capable of executing, since our web pages are running in the JVM (Java Virtual Machine). To avoid cross site scripting all the input boxes are having maxlength attribute according to the category. The number of characters to be placed in the input box is already defined and Servlets do input validation. Firewalls can be the effective method to protect the system from network threats at the same time it provides access to the networks and the internet. The ability of the single web server for the E-voting system is limited since massive users access the application at the same time and to avoid server breakdown I have to provide with multi-servers.

I use oracle as back-end for security, as it integrates security mechanisms into its database management system. Since I have sensitive data and oracle controls to prevent the unauthorized access. It has high quality backup and recovery management. It also guarantees Atomicity, Consistency, Isolation, Durability (ACID) properties of data.

The traceability matrix is used to trace the identified security requirements from the requirements engineering phase to the implementation phase, i.e. to trace security requirements to the security mechanism. Table 3 gives the traceability matrix for the security mechanism implemented for E-voting system for some of the important security requirements given in Table2. The security requirements are categorized under standard security needs namely Confidentiality (C), Integrity (I) and Availability (A) and tabulated in Table 3. These are the some of the security mechanisms I used to achieve the security Requirements given in the Table 2.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 9, Issue 9, September 2020

DOI 10.17148/IJARCCE.2020.9901

TABLE III. TRACEABILITY MATRIX-SECURITY REQUIREMENTS TO SECURITY MECHANISM.

| **Security Requirements with Security Mechanisms** | **Security Mechanism** | | | | |
|---|---|---|---|---|---|
| | **Authentication** | **Authorization** | **Access Control** | **Cryptography** | **Technology** |
| **SR 1- C, I** | □ | □ | □ | □ | |
| **SR 2- A** | □ | - | □ | - | |
| **SR 3- C** | □ | - | □ | □ | |
| **SR 4- I** | □ | □ | □ | □ | **Java /J2EE Technology** Sup ports for the implementation of all the Security Requirements |
| **SR 5- C** | - | □ | □ | □ | |
| **SR 6- A** | □ | □ | □ | - | |
| **SR 7- A, I** | □ | - | □ | □ | |
| **SR 8- I** | □ | □ | □ | □ | |
| **SR 9- A, C** | □ | □ | □ | □ | |
| **SR 10- I, A** | □ | □ | □ | □ | |
| **SR 11- C** | □ | □ | □ | - | |
| **SR 12- C, I** | □ | □ | □ | □ | |
| | | | | | |

In the next section, the experimental setup is discussed to evaluate how far MOSRE framework is feasible and effective to adopt in the security requirements engineering phase for the development of secure web application.

## V. EXPERIMENTAL SETUP

The evaluation was conducted with 30 participants, academic users such as 6 professors, 6 research scholars, 10 postgraduate students, and 8 industry professionals. They acted as one of the stakeholders such as voter, candidate, election officer, requirements analyst, developer, security expert, tester, designer, etc. to participate, provide their views and idea, and to identify security requirements. They were divided into three groups and each group developed E-voting software system and performed security requirements analysis for three different E-voting software systems using SREF, MOSRE framework and without using any SRE methods respectively in a period of 6 months. These 30 participants are considered valid participants because they have developed requirements specifications in the past in both academic and industry settings, but none has worked with security in the context to requirements analysis and specifications. In order to observe the feasibility of MOSRE framework for eliciting security requirements of web applications and to adopt in the SRE phase, the evaluation of MOSRE and SREF frameworks was conducted in two phases namely, requirements engineering and testing phases of SDLC.

### i.Comparative analysis of effectiveness of MOSRE in identifying the vulnerabilities in requirements engineering phase.

The evaluation method used was to find the number of vulnerabilities identified in each category of vulnerability given in the Table 4 for each E-voting system by following the SRE methods such as SREF and MOSRE, and without using any SRE methods respectively. Table 4 lists the important category of vulnerability such as XSS, authentication, authorization, Cross-Site Request Forgery (CSRF), etc., of web application.Each vulnerability category has a number of sub- vulnerabilities, for example, authentication has sub- vulnerabilities such as no password change after first login, and password reset mechanism weakness, no logout Functionality, mixing personalization with authentication, storing clear text credentials in configuration files, etc.

Table IV. List of vulnerability categories.

| **Vulnerability Categories** |
|---|
| XSS |
| Buffer Overflow |
| Path |
| Authentication |
| Authorization |
| SQL Injection |
| Information Leakage |

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 9, Issue 9, September 2020

DOI 10.17148/IJARCCE.2020.9901

| |
|---|
| CSRF |
| Session Management |
| Denial of Service |
| Other |

These vulnerability categories can be used to identify vulnerability in the web application being developed, in order to solve and countermeasure it by identifying security requirements.

**ii.Comparative analysis of performance of MOSRE by the vulnerabilities found in testing phase.**
The methodology used was to calculate the percentage of vulnerabilities detected in each vulnerability category in the applications implemented by the participants in different groups using a web vulnerability-scanning tool. If the percentage of vulnerabilities detected is less, then the performance is high. This method of evaluation will help to prove that security requirements should be given equal importance similar to business requirements and must be considered as functional requirements. It will also prove that security requirements should be analyzed in the early phases of the SDLC and not at the time of design or coding.

## VI. RESULT ANALYSIS

The result analysis was performed in two phases: first in the requirements engineering phase and second in testing phase of SDLC.

**i.Effectiveness comparison of MOSRE in requirements engineering phase.**
The primary aim is to evaluate and analyze the deliverable of requirements engineering phase i.e. the Security Requirements Specification (SRS) after the application of different SRE methods to the E-voting system. From the SRS, the vulnerabilities identified in web application are classified under vulnerability categories, which are based on the standards given in NVD by National Institute of Standards and Technology (NIST), OWASP, and WASC. Figure 2 shows the effectiveness in identifying vulnerabilities during requirements engineering phase by applying existing and proposed SRE methods to E- voting system respectively. It is observed that more Number of vulnerability is identified by adopting MOSRE framework than any other SRE methods, which can be solved by identifying the security requirements with respect to business requirements.

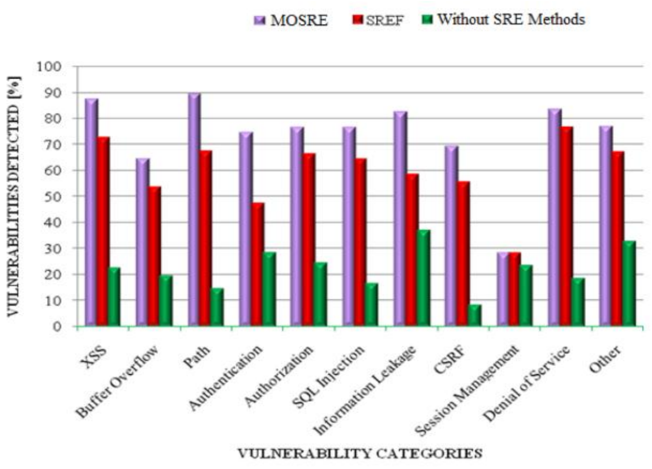

Figure 2. Effectiveness of MOSRE in requirements engineering phase.

MOSRE adopted group-conducted iterations to find more number of vulnerabilities in the web application. The SREF group, even though they proceeded with iterations, they failed to identify more vulnerability, since they lack in vulnerability identification and risk analysis, which are the important activities of SRE. The group without using SRE methods did not identify the threats and vulnerabilities in the requirements engineering phase since they consider them only at the time of design and coding. Therefore, the SRS developed without using SRE methods have identified less number of vulnerabilities.

**ii.Performance comparison of MOSRE in testing phase.**
In the second phase of the evaluation, the percentage of vulnerabilities present in each web application was calculated using a web application-scanning tool. Figure 3 shows the chart for the percentage of vulnerabilities detected during

testing. It is observed that for the MOSRE-based E-voting system, the percentage of vulnerabilities detected was less than other applications developed with other SRE methods. There by the performance of MOSRE is high and chart proves the hypothesis that "If security requirements for software systems are considered as functional requirements and elicited in the early phase of SDLC then vulnerabilities can be minimized than the software system developed without SRE phase". The level of security can be increased by debugging the errors, the vulnerabilities identified during testing phase, and E-voting system can be developed with higher likelihood of security.

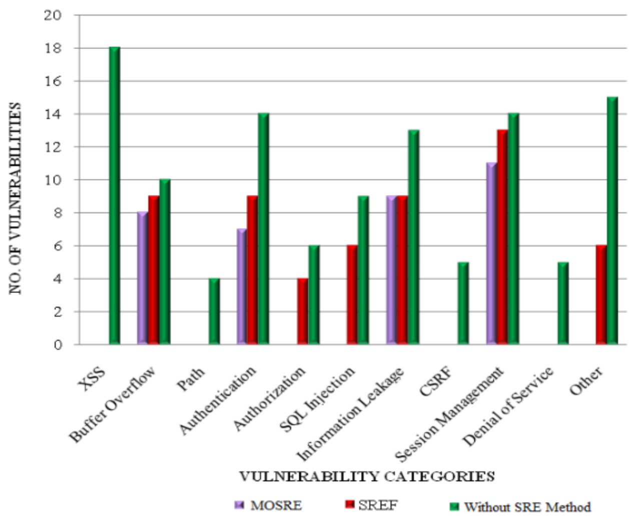

Figure 3. Performance of MOSRE in testing phase.

The security levels can be improved based on business criticality, and as given by NIST.
The assurance level standards for security are:

a. Very high for mission critical business/safety of life.
b. High for exploitation causes serious brand damage and financial loss with long-term business impact.
c. Medium for applications connected to the internet that process financial or private customer information.
d. Low for typical internal applications with non- critical business impact.
e. Very low for applications with no material business impact.

The relative cost to fix vulnerabilities and threats at requirements engineering is minimal because the good software requirements specification is the base for developing error free application systems. If vulnerabilities and threats are identified earlier, they can be mitigated by security requirements and traced to later phases of SDLC.
The overhead of the developers is reduced, since security requirements are specified in the requirements engineering phase. They also have less rework after testing or deployment phase, since the system developed will be less prone to vulnerabilities. Moreover, the cost of fixing the vulnerabilities and threats is 100 times higher after deployment of the web application [36, 37]. This includes only development charges of the web application and not the business and customer's loss due to poor security requirements analysis.
As given by Mouratidis and Jűrjens [38] mistakes, i.e. not analyzing security requirements in early software process can have far-reaching consequences in subsequent stages that are difficult and costly to remedy. Therefore, it is best to analyze and specify security requirements in the requirements engineering phase by giving high priorities to security requirements.

From the experimental results given in Figures 2 and 3, it can be inferred that:

1. MOSRE has improved the identification of assets, threats and vulnerabilities in the requirements engineering phase;
2. MOSRE has improved in vulnerabilities identification, by an average of 50% compared to SREF at requirements engineering phase.
3. MOSRE show less percentage of vulnerabilities detected in the testing phase, thereby the performance of MOSRE increased by 47% compared to SREF.

### iii.Performance Comparison of MOSRE with other SRE Methods

The results for each SRE method are shown in the Table 5 The number of assets, threats, and vulnerabilities matched with the considered assets, threats, and vulnerabilities using SREF, SQUARE, MSREF, and MOSRE, and without using any SRE methods for E-voting system.

Vol. 9, Issue 9, September 2020

DOI 10.17148/IJARCCE.2020.9901

Table V. Performance Comparison of Assets, Threatsand Vulnerabilities identified for E-voting system

| SRE Methods | No. of matched Assets (out of 25) | No. of matched Threats (out of 34) | No. of matched Vulnerabilities (out of 15) |
|---|---|---|---|
| Without SRE | 9 | 3 | 4 |
| SREF | 17 | 16 | 7 |
| SQUARE | 10 | 23 | 10 |
| MSREF | 18 | 21 | 8 |
| MOSRE | 20 | 25 | 12 |

From the Table 5 the performance can be analyzed, which is calculated with the number of identified and matched assets, threats, and vulnerabilities in SRS to the total number of considered assets, threats, and vulnerabilities.

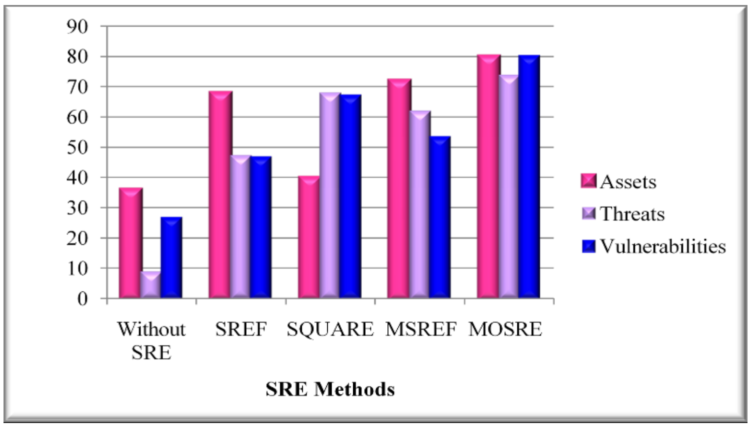

Figure 4. Performance of MOSRE in identifying Assets, Threats and Vulnerabilities

From the Figure 4, it reports that MOSRE framework shows high performance in identifying assets, threats and vulnerabilities compared to other SRE methods.

## VII. DISCUSSION

Our experiences in using MOSRE framework outlined in this paper showed benefit over SREF and significant benefit over not using SRE methods at all. Since security, experts involve in security requirements engineering phase to identify the security requirements, which will reduce the overhead of the developers to have security knowledge. It integrates the security requirements specification techniques such as UML, security use cases and misuse cases. MOSRE is easy to use and the framework involves stakeholders to identify security requirements. The framework analyses and prioritizes the assets, threats and vulnerabilities related to business and system, but these steps are lacking with SREF and are very complex to adopt to elicit security requirements. In MOSRE, the business requirements are considered for eliciting security requirements, because it helps to find the conflicts and resolve them between business and security requirements. Moreover, MOSRE framework is a model based approach and can be used to develop secure web application and it is based on the concept of iterative software construction of the requirements engineering process.

## VIII. CONCLUSION

The security requirements play an important role in developing secure web applications but recent research is on security mechanisms [39] rather than security requirements. The security requirements have to be given equal importance as business requirements.
I need a method to elicit and analyze security requirements for secure web application development. At the same time, I need to have a complete, clear security requirements specification that can be used by the developers without the help of security experts. As it is given in [40], good requirements specification document should include both functional and non- functional requirements. I suggest a better method for the requirements engineers to adopt for eliciting specifying security requirements, the SRE methods such as MOSRE and SREF were evaluated. The specifications were compared with respect to number of vulnerabilities identified. Since, the vulnerabilities help to identify the security requirements efficiently. To study the performance of MOSRE, the identified security requirements has been implemented with the web application and scanned for vulnerabilities.

To justify the importance of SRE phase in SDLC, a web application was also developed without adopting any SRE methods. The results were promising that, little vulnerability detected in MOSRE adopted web application. Finally, in this study, the security requirements were traced to the security mechanism to confirm that security requirements are carried from requirements engineering phase to the implementation phase. From the evaluation, it is found that, MOSRE helps the requirements engineers to engrave specification for security requirements effectively than SREF and SRE and should be included in the early phases of SDLC. MOSRE can be extended for other aspects of security such as trust and privacy. It can also be extended to support risk analysis and management, since it is another broader area of research. It would be beneficial to a project, if test cases could be automatically generated from specified security requirements.

Thus, the resultant security requirements obtained with MOSRE framework were very promising to attain threat and vulnerability- free software systems. However, it is not able to reach 100% results. It is because, the assessments are carried out at very early phase (RE phase) of SDLC. It is also due to the variations realized in design and implementation; it may be possible to cover all threats and vulnerabilities by testing phase. The evaluation of MOSRE on web application was carried out by a group of participants to elicit and specify security requirements. The SRS gathered from the participants were examined for the lists of assets, threats and vulnerabilities and compared with the SRS of existing SRE methods. On investigating the SRS of MOSRE it is found that, the effectiveness and performance is comparatively better than the existing methods.

It is also inferred from the evaluation that, MOSRE is simple and understandable for the requirements engineers to elicit security requirements, which guarantee the desired level of protection to the software systems. The identified security requirements can be reused with the activities of any SRE methods to elicit effective security requirements. I intend to extend further my work for security requirements reusability, in order to reduce security knowledge and dependency on security experts. It will help to save time/cost and make better choices in applying security requirements, since the framework allows requirements engineers to exploit the accumulated knowledge. Thus, very high level of security can be achieved for the software systems.

## REFERENCES

[1] Mead N., "Security Requirements Engineering," Carnegie Mellon University, https://BuildSecurityin.us-cert.gov/Daisy/Bsi/Articles/Best-Practices/Requirements/243.html, Last Visited, 2010.
[2] Gartner Research, http://www.gartner.com/ technology/research.jsp, Last Visited, 2012.
[3] Hopkins A., "Web Application Vulnerability Statistics 2010-2011," White Paper, Context Information Security, 2012.
[4] Haley C., Laney R., Moffett J., and Nuseibeh B., "Security Requirements Engineering: A Framework for Representation and Analysis,"
[5] Jűrjens J., "UMLsec: Extending UML for Secure Systems Development," in Proceedings of fifth International Conference on the Unified Modeling Language, Dresden, pp. 412-425, 2002.
[6] Fuentes L. and Sanchez P., "Designing and Weaving Aspect-Oriented Executable UML Models," Journal of Object Technology, vol. 6, no. 7, pp.109-136, 2007.
[7] Fu X., Lu X., Peltsverger B., Chen S., Qian K., and Tao L., "A Static Analysis Framework for Detecting SQL Injection Vulnerabilities," in Proceedings 31st Annual International Computer Software and Applications Conference, Beijing, pp. 87-96, 2007.
[8] Ismail O., Kadobayashi Y., Yamaguchi S., and Etoh M., "A Proposal and Implementation of Automatic Detection/ Collection System for Cross Site Scripting Vulnerability," in Proceedings 18th International Conference on Advanced Information Networking and Applications, Fukuoka, pp.145-151, 2004.
[9] Scott D. and Sharp R., "Abstracting Application- Level Web Security," in Proceedings of 11th International Conference on World Wide Web, Honolulu, pp. 396-407, 2002.
[10] Kals S., Kirda E., Kruegel C., and Jovanovic N., "SecuBat-A Web Vulnerability Scanner," in Proceedings of 15th International Conference World Wide Web, Edinburgh, pp. 247-256, 2006.
[11] Fuentes L. and Sanchez P., "Designing and Weaving Aspect-Oriented Executable UML Models," Journal of Object Technology, vol. 6, no. 7, pp.109-136, 2007.
[12] Koch N. and Kraus A., "The Expressive Power of UML-Based Web Engineering," in Proceedings of 2nd International Workshop on Web-Oriented Software Technology, Malaga, pp. 105-119, 2002.
[13] Lodderstedt T., Basin D., and Doser J., "SecureUML: A UML-Based Modeling Language for Model-Driven Security," in Proceedings of fifth International Conference on the Unified Modeling Language, Dresden, pp. 426-441, 2002.
[14] José Escalona M. and Koch N., "Requirements Engineering for Web Applications-A Comparative Study," Journal of Web Engineering, vol. 2, no. 3, pp. 193-212, 2004.
[15] Mellado D., Medina E., and Piattini M., "A Common Criteria Based Security Requirements Engineering Process for the Development of Secure Information Systems," Computer Standards and Interfaces, vol. 29, no. 2, pp. 244- 253, 2007.
[16] Mead N., Houg E., and Stehney T., "Security Quality Requirements Engineering (SQUARE) Methodology," Technical Report, 2005.
[17] Suleiman H. and Svetinovic D., "Evaluating the Effectiveness of the Security Guality Requirements Engineering (SQUARE) Method: a Case Study Using Smart Grid Advanced Metering Infrastructure," Requirements Engineering, vol. 18, no. 3, pp. 251-279, 2013.
[18] Haley C., Laney R., Moffett J., and Nuseibeh B., "Security Requirements Engineering: A Framework for Representation and Analysis,"
[19] Salini P. and Kanmani S., "Evaluating Security Requirements Engineering Framework for Web Applications," CiiT International Journal of Software Engineering and Technology, vol. One, no. 3, pp. 106-112, 2009.
[20] Salini P. and Kanmani S., "Model Oriented Security Requirements Engineering (MOSRE) Framework for Web Applications," in Proceedings of Second International Conference on Advances in Computing and Information Technology, Chennai, pp. 341-353, 2012.
[21] Jefferson D., Rubin A., Simons B., and Wagner D., "A Security Analysis of the Secure Electronic Registration and Voting Experiment (SERVE)," http://servesecurityreport.org/paper .pdf, Last Visited, 2004.

[22] Kiayias A., Korman M., and Walluck D., "An Internet Voting System Supporting User Privacy," in Proceedings 22nd Annual Computer Security Applications Conference, Miami Beach, pp.165-174, 2006.
[23] Kohno T., Stubblefield A., Rubin A., and Wallach S., "Analysis of an Electronic Voting System," in Proceedings of IEEE Symposium on Security and Privacy, Berkeley, pp. 27-40, 2004.
[24] Liu L., Yu E., and Mylopoulos J., "Security and Privacy Requirements Analysis within a Social Setting," in Proceedings 11th IEEE International Conference on Requirements Engineering, Monterey Bay, pp. 151-161, 2003.
[25] Rubin A. "Security Considerations for Remote Electronic Voting over the Internet," http://avirubin.com/e-voting.security.html. Last Visited, 20014.
[26] Salini P. and Kanmani S., "Model Oriented Security Requirements Engineering (MOSRE) Framework for Web Applications," in Proceedings of Second International Conference on Advances in Computing and Information Technology, Chennai, pp. 341-353, 2012.
[27] Salini P. and Kanmani S., "Security Based Requirements Engineering for E-Voting System," in Proceedings of Third International Conference on Recent Trends in Information, Telecommunication and Computing, Springer New York, pp. 451-455, 2013.
[28] Online Web Application Security Projects, https://www.owasp.org, Last Visited, 2014.
[29] List of Vulnerabilities, http://nvd.nist.gov/ full_listing.cfm, Last Visited, 2014.
[30] Mell P., "The National Vulnerability Database," National Institute of Standards and Technology, http://csrc.nist.gov/groups/SMA/ispab/documents/minutes/2005-12/P_Mell-Dec2005-ISPAB.pdf 1, Last Visited, 2005.
[31] Web Security Threat Classification. WASC, https://files.pbworks.com/download/Ps7eci3iTm/ webappsec/ 13247059/WASC-TC-v2_0. Pdf. Last Visited, 2014.
[32] Salini P. and Kanmani S., "Security Based Requirements Engineering for E-Voting System," in Proceedings of Third International Conference on Recent Trends in Information, Telecommunication and Computing, Springer New York, pp. 451-455, 2013.
[33] Karabey B. and Baykal N., "Attack Tree Based Information Security Risk Assessment Method Integrating Enterprise Objectives with Vulnerabilities," The International Arab Journal of Information Technology, vol. 10, no. 3, pp. 297-304, 2013.
[34] Salini P. and Kanmani S., "Security Based Requirements Engineering for E-Voting System," in Proceedings of Third International Conference on Recent Trends in Information, Telecommunication and Computing, Springer New York, pp. 451-455, 2013.
[35] Jacobson I., "Modeling with Use Cases: Formalizing Use Case Modelling," Journal of Object-Oriented Programming, vol. 8, no. 3, pp.139-149, 1995.
[36] https://www.celerity.com/the-true-cost-of-a-software-bug
[37] https://azevedorafaela.com/2018/04/27/what-is-the-cost-of-a-bug/
[38] Mouratidis H. and Jűrjens J., "From Goal-Driven Security Requirements Engineering to Secure Design," International Journal of Intelligent Systems, vol. 25, no. 8, pp. 813-840, 2010.
[39] Subramaniam U. and Subbaraya K., "A Biometric Based Secure Session Key Agreement Using Modified Elliptic Curve Cryptography," The International Arab Journal of Information Technology, vol. 12, no. 2, pp. 155-162, 2015.
[40] LamSweerde A., "Elaborating Security Requirements by Construction of Intentional Anti-Models," in Proceedings of 26thInternational Conference on Software Engineering, Edinburgh, pp. 148-157, 2004.

## BIOGRAPHY

**Dr. Osama Ahmad Salim Safarini** had finished his PhD. from The Russian State University of Oil and Gas Named after J. M. Gubkin, Moscow, 2000, at a Computer-Aided Information Control system Department. He obtained his BSC and MSC in Engineering and Computing Science from Odessa Polytechnic National State University in Ukraine 1995, 1996 respectively. He worked in different universities and countries.