INTERNATIONAL JOURNAL OF SOFTWARE ENGINEERING & COMPUTER SYSTEMS (IJSECS)
ISSN: 2289-8522 e-ISSN: 2180-0650
VOL. 8, ISSUE 1, 45 – 52
DOI: https://doi.org/10.15282/ijsecs.8.1.2022.5.0095

ORIGINAL ARTICLE

SecRS template to aid novice developers in security requirements identification and documentation

N.Q. Tunio[1], and R. Ahmad[1]

[1]Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur, Malaysia

ABSTRACT – The security requirements are one of the non-functional requirements (NFR) which acts as a constraint on the functions of the system to be built. Security requirements are important and may affect the entire quality of the system. Unfortunately, many organizations do not pay much attention to it. The security problems should be focused on the early phases of the development process i.e. in the requirements phase to stop the problems spreading down in the later phases and in turn to avoid the rework. Subsequently, when security requirements are to be focused, proper guidance should be provided which should assist requirements engineers. Many security requirements engineering methods were developed in the past which require different level of expertise such as SQUARE process which requires requirements engineer to have a certain level of security expertise. Moreover, it lacks proper guidance especially for novice developers in applying the existing security requirements engineering (SecRE) methods to identify security requirements. Hence, this study intends to address the gap by developing a guided template to assist novice developers in the security requirements identification and documentation. The main objectives of the research are: 1) to study and investigate the existing security requirements engineering (SecRE) methods. 2) To develop a template to aid novice developers in identifying and documenting security requirements. The developed template is applied to two case studies of software projects to determine its usability and applicability. The results of the case studies evaluation show that both the usability and applicability of the template is good. The template is also evaluated by several experts and software practitioners. The evaluation results show that the SecRS template is found to be satisfying the usability and applicability factors; thereby confirming that the proposed template achieves its desired objective of aiding the novice developers to identify and document security requirements correctly.

ARTICLE HISTORY
Received: 23 Sept 2020
Revised: 7 Oct 2021
Accepted: 28 Oct 2021

KEYWORDS
Security requirements
Security requirements
Engineering methods
SQUARE
CLASP
Usability
Applicability

INTRODUCTION

Requirements Engineering (RE) is a process of determining the requirements of any software product. It is the most important phase of any software development approach which affects product success or failure. Errors are expected to come in this process as the result of mistakes caused by humans such as grammatical errors in carrying one or more requirements, communication issues, etc. Jaffe et.al [2] quote that the errors found in software requirement specification (SRS) have been a major cause of software production failures. Errors introduced during the RE process can cost lesser to correct than errors introduced later in the software development life cycle (SDLC). The Standish Group (1995-1996) surveyed project failure factors which show that RE-related factors are more as compared to the other factors [12]. The lack of security is one of the important factors observed in past project failures [3]. It grounds lots of incidents and attacks targeting the software systems. Industry and academia do not focus on security issues systematically as a topmost priority when performing requirements engineering [16]. Security is always considered in the later stages of the SDLC [25]. It must be focused on the initial phases of the development lifecycle i.e. requirements analysis phase. The early incorporation of security with the initial phase will eliminate all security issues thus the economy of billion dollars can be saved [14].

The motivation of this study is the growing concern over security over the past years due to the many software failures lacking [4,5]. The guidance to the security requirements identification and documentation will aid novice developers to do their job effectively. The lack of security is one of the factors behind the system failure that grounds a lot of incidents and attacks targeting the software systems [3]. The requirements engineers are the source of the errors in the requirements identification process that grounds these failures [6, 18, 19]. Most requirements engineers have good expertise in functional requirements, but only a few have basic security architectural knowledge such as password security, encryption, and decryption [28]. Most of the requirements engineers do not have good knowledge of authentic security requirements enigneering they tend to mix it with the architectural concepts [6, 19, 26]. The identification of security requirements for the applications has been the main research issue for a long time [9, 20].

*CORRESPONDING AUTHOR | N.Q. Tunio | ✉ nuzhat.qadir@siswa.um.edu.my

© The Author(s) 2022. Published by Universiti Malaysia Pahang Al-Sultan Abdullah Publishing. This is an open access article under the CC BY-NC 4.0 International license

Many security requirements engineering methods were developed in the past which require different levels of expertise such as SQUARE, CLASP, etc., require requirements engineer to have a certain level of security expertise. Moreover, they lack proper guidance especially for novice developers in applying the existing security requirements engineering (SecRE) methods to identify security requirements [7, 8]. The empirical evaluation of these methods to determine or confirm their effectiveness in security requirements identification is found to be very few, which could aid novice developers in the selection of the most effective method [21, 22, 23, 24].

The template should be developed to facilitate non-security experts to engineer security requirements [6]. The objectives of the research are:

To study and investigate the existing security requirements engineering (SecRE) methods.

To develop a template to aid novice developers in identifying and documenting security requirements.

To evaluate the proposed template.

The template will be useful for the requirements engineer or novice developer who doesn't have security expertise. It will facilitate the novice developers to identify and document the security requirements correctly and thoroughly.

The remainder of the paper is divided into the following parts: section 2 presents the overview of SRE methods. Section 3 presents the proposed method, and template. Section 4 discusses the proposed template's evaluation and results. Section 5 presents the conclusions, limitations, and future work of the proposed template.

RELATED WORK

The literature is collected from various sources such as articles, books, journal papers, websites, and students' theses. The existing SecRE methods are reviewed to identify their strengths and weakness, usability, and applicability according to the novice developer. The six different SecRE methods studied are SQUARE, MSRA, SREP, UML-based approach, Secure Tropos, and CLASP.

Security Quality Requirements Engineering (SQUARE) is a nine-step process introduced by SEI for gathering and ranking security requirements. It is centered on the communication between the stakeholders assisted via the RE team, which they are regarded as foremost important in the entire process [9].

Multi-lateral Security Requirements Analysis (MSRA) is a seven-step process, which emphasizes applying multilateral security principles during the RE phase [10]. It can be practiced as soon as the system functionalities are recognized. It is inspired by the current concept on the viewpoint of RE and multilateral security [10].

Security Requirements Engineering Process (SREP) is a nine-step process; partly depends on SQUARE. It differs from the SQUARE process as it introduces the concept of reuse and common criteria as part of the process [9]. The Common Criteria (CC) (ISO/ IEC 15408) standard is used in SREP for security requirements management [11].

The misuse-cases is a UML-based approach developed for security requirements identification. It represents the unwanted actions performed to violate the security of the system [10]. The misuse cases are the opposite of the use-case which is used to represent the functionality of the system.

Secure Tropos is the expansion of tropos methodology which was developed for agent-oriented software systems. The new concepts on security modeling are introduced in secure tropos [10]. Concepts such as constraint, dependency, and entity according to the security perspective are concealed in this methodology. It assists the developers to recognize security problems in the SDLC [1].

Secure Software Inc. [13] introduces the Comprehensive Lightweight Application Security Process (CLASP) to be used publicly free of charge. It is the basis of the Open Web Application Security Process (OWASP) project. It is a life-cycle process with a focus to improve security by implementing a different set of activities throughout the development life cycle.

The studied SecRE methods are compared to some factors such as user, scope, concern, consistency, flexibility, applicability, and usability shown in table 1. Table 1, illustrates that some of the methods are applied in the RE phase while some are in the later phases of development. Only SQUARE aims at the SecRE phase [9]. The user for each method is different. Only the CLASP facilitates the novice developer while others require the security or requirements knowledge to practice them in real projects such as SQUARE. Flexibility is considered as the ability of a method to be easily modified or flexible to be used in different situations i.e. systems or domains per the varying user's skills. The concern is the focus of the method.

Table 1 SecRE methods comparison

Methods / Factors	SQUARE	MSRA	SREP	Misuse-cases	Secure Tropos	CLASP
User	Requirements engineers (with security expertise) and stakeholders.	Stakeholder	Stakeholder	Stakeholder/ analyst	Developer	Stake holder's / project participant
Novice developer	No	No	No	No	No	Yes
Scope	SRE phase	RE phase	RE phase	RE / Design phase	All phases	RE / All phases

Concern	Gathering and ranking of security requirements	Multi-lateral security principles usage	Common criteria usage and introduce reuse	To identify possible threats	Security modeling	To improve security via a different set of activities
Security Terms Usage	Threats, Risks	Threats	A threat, Vulnerability, and Risk	Threats, Risks.	Threats, Risks.	Threats
Considers Security Requirements	Quality / NFR	-	Functional requirements	Functional requirements	NFR	NFR
Consistency	Yes	-	Yes	Yes	No	No
Flexibility	Yes	No	Yes	-	Yes	Yes

The methods are further compared on their applicability and usability in table 2. It shows the comparison result inspired by the survey result presented by [5]. Each method has various steps to follow some are easy to understand while some are too lengthy to apply in real projects such as CLASP which is very general introducing a 24-set of activities. Few methods are practiced in the industry according to the gathered information on their industrial evaluations such as SQUARE, and CLASP [9, 27]. SQUARE is widely practiced and is considered as best among the other SecRE methods in various survey results in terms of usability, and applicability [9, 5]. Some methods provide automated support for these methods through tools developed such as the secTro2 tool for secure Tropos method, SREP tool for SREP method, and SQUARE Tool for SQUARE [16].

Table 2 SecRE methods comparison according to usability and applicability

Factors / Methods	Applicability				Usability	
	Large Projects	**Small Projects**	**System Oriented**	**Machine Oriented**	**Easy to learn**	**Easy to Use**
SQUARE	Yes	No	Yes	Yes	Yes	Yes
MSRA			Yes			
SREP	Yes	Yes			No	No
Misuse-cases	Yes	Yes	-	-	No	No
Secure Tropos			Yes	Yes		
CLASP	No	Yes	Yes	Yes	Yes	Yes

Some methods are system-oriented while some are machine-oriented, some provide automated support while some do not, some are easy to use and learn while some are not. Only the CLASP facilitates the novice developer while others require the security or requirements knowledge to practice them in real projects such as SQUARE. Thus, among all the methods reviewed, SQUARE and CLASP are found to be most satisfying and most effective according to the novice developer, hence selected for the template design.

PROPOSED METHOD & TEMPLATE

The Proposed 'SQUARE-Extended'

SQUARE-Extended is a proposed method in which the activities of CLASP for the RE phase are selected and merged with the SQUARE steps based on the goal of the step. SQUARE steps names are a bit modified to demonstrate the combination of CLASP activity introduced in that step. Table 3 shows the steps and activities of the SQUARE-Extended process.

The first three steps are decomposed into a few activities to make the process clear and understandable for the novice user. The activities are decomposed into four sub-sections namely tasks, process, document the activity, and outcome. Some steps of SQUARE are combined into one step i.e. step 5 of SQUARE-Extended to reduce the time required to complete the step; as these SQUARE-steps are performed as one of the tasks in this step. The fifth step of SQUARE-Extended is the most important as a new pattern for requirements elicitation is introduced.

Table 3 SQUARE-Extended process steps and activities

SQUARE-Extended					
Step		**Activity**			
No.	**Title**	**No.**	**Title**	**Method**	**Section**
1.	Security Awareness and Terms Selection	1.1.	Institute Security Awareness Program	CLASP	i. Tasks ii. Process iii. Document the activity iv. Outcome
		1.2.	Select Security Terms Definition	SQUARE	
2.	Security Goals and Operational Environment Identification	2.1.	Identify Security Goals	SQUARE	
		2.2.	Specify Operational Environment	CLASP	
3.	Artifacts Development	3.1.	Design System Architecture	SQUARE	
		3.2.	Design Use-case scenarios		
		3.3.	Design Misuse-case scenarios	SQUARE-CLASP	
		3.4.	Identify Attack Surface		
4.	Perform Risk Assessment			SQUARE	
5.	Requirements Elicitation and Prioritization				
6.	Requirements Inspection				

Proposed 'Security Requirements Specification (SecRS) template'

The template is proposed titled 'Security Requirements Specification (SecRS)' to follow the proposed method 'SQUARE-Extended'. The template is designed in Microsoft Word 2016. The template has various sections designed for novice developers' ease of use. The template starts with the project overview section; where details about the project such as the purpose of the project, and stakeholders of the project are recorded. The second section is the business goals of the project. The third or last section is the SQUARE-Extended section, where all the steps and their subsequent activities are performed sequentially. All the steps and their subsequent activities are decomposed into four subsections: tasks, process, documentation, and outcome. The tasks subsection lists the tasks which the participant going to perform in completing the step or activity. The process subsection details the process followed in the activity. The activity is documented for management purposes in the documentation subsection. The outcome subsection details the outcome of the activity. The outcomes for each step or its subsequent activities are different and are presented as a list in tabular form while some are specific documents such as security terms definition agreement (STDA), system architecture document, etc. The outcomes per step and activities are presented in Table 4.

Table 4 The outcomes of SQUARE-Extended process step and activities

Step		**Activity**		**Outcome**
1.	Security Awareness and Terms Selection	1.1.	Institute Security Awareness Program	Meeting Minutes
		1.2.	Select Security Terms Definition	Security Terms Definition Agreement (STDA)
2.	Security Goals and Operational Environment Identification	2.1.	Identify Security Goals	Security goals list
		2.2.	Specify Operational Environment	Operational environment specification table
3.	Artifacts Development	3.1.	Design System Architecture	System Architecture Document
		3.2.	Design Use-case scenarios	Use-case scenarios document
		3.3.	Design Misuse-case scenarios	Misuse-case scenarios document
		3.4.	Identify Attack Surface	Attack Surface Document
4.	Perform Risk Assessment			Risk Assessment Results
5.	Requirements Elcitation and Prioritization			Secuirty Requirements Documnet
6.	Requirements Inspection			Security Requirements Inspection results

The fifth step is the most important; it aims to elicit security requirements and then categorize and prioritize them. For the security requirements elicitation, the pattern defined by Pohl [15] is selected as presented in Figure 1.

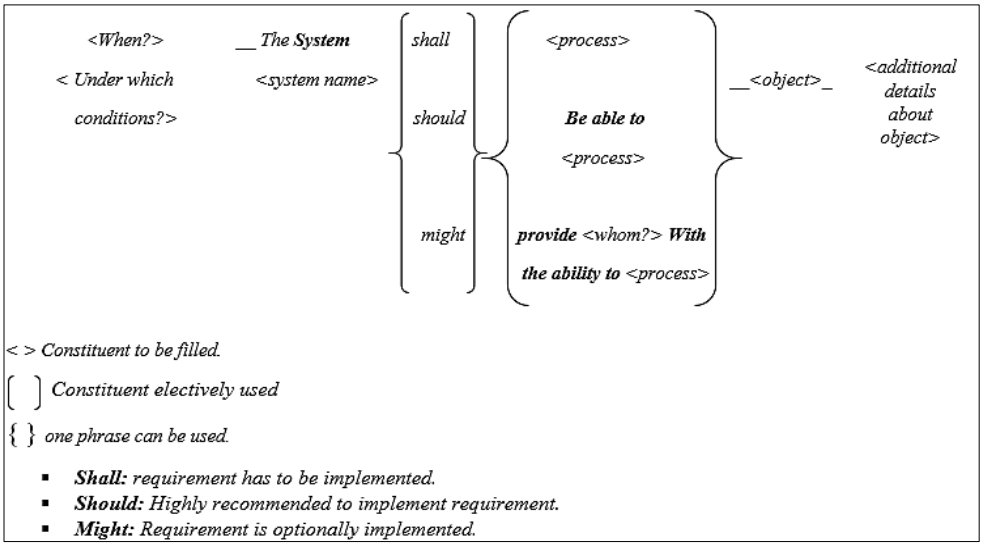

Figure 1 Pohl's Requirements Elicitation Pattern

The pattern is enhanced to be used for security requirements elicitation. The security requirements must be implemented or highly recommended for implementation. The enhanced pattern is presented in Figure 2.

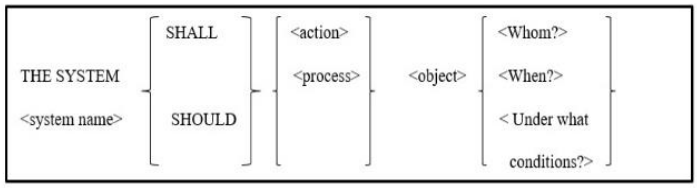

Figure 2 Enhanced Pattern for Security Requirements Elicitation

EXPERIMENTAL RESULTS

The proposed SecRS template is evaluated and validated to demonstrate its usability and applicability according to novice developers. The two evaluation methods are used: case study, and expert's review. The case study is selected in this research; to demonstrate the usability and applicability of the proposed template while applying it to software projects. While review's evaluation is selected in this research; to demonstrate the usability and applicability of the proposed method and template per the user's experience.

The SecRS template is applied on the two software projects: online book store and student registration system; to prove the usefulness and effectiveness of template in identification and documentation of security requirements through the practical evaluation of real-world software projects. The case studies evaluation results are presented in Table 5.

Table 5 Case studies evaluation results

Factors	Case study 1	Case study 2
Usability	Found to be easy to use in following the proposed method through the series of steps and activities.	
Applicability	Found to be applicable in guiding step-by-step for any software product's security requirements identification and documentation process.	

The results of both studies are the same in terms of usability and applicability. The problem encountered in this evaluation is time complexity, as various sub-sections decomposed to many activities which makes it highly time-consuming.

For the review's evaluation, the proposed template is given to the few users which are postgraduate students, and a few experts which are faculty members at FCSIT, Universiti Malaya. The review's evaluation makes use of both qualitative and quantitative data collection techniques. The data is collected through the questionnaire which is provided to the reviewers to record their feedback on the proposed template. The questionnaire consists of three sections; the first section contains the reviewer's personal information; the second section evaluates the template through the set of questions, and the third section contains suggestions for improvement which reviewer can provide. The questions in the second section are rated from 1-5 starting from strongly disagree to strongly agree. The questionnaire assesses the quality factors i.e. usability, and applicability for evaluating the SecRS template; thus, for both the reviews the same questionnaire has been used to get template evaluation results assessing these quality factors from different reviewers. The frequency of the reviewer's answers is illustrated in Figures 3 and 4.

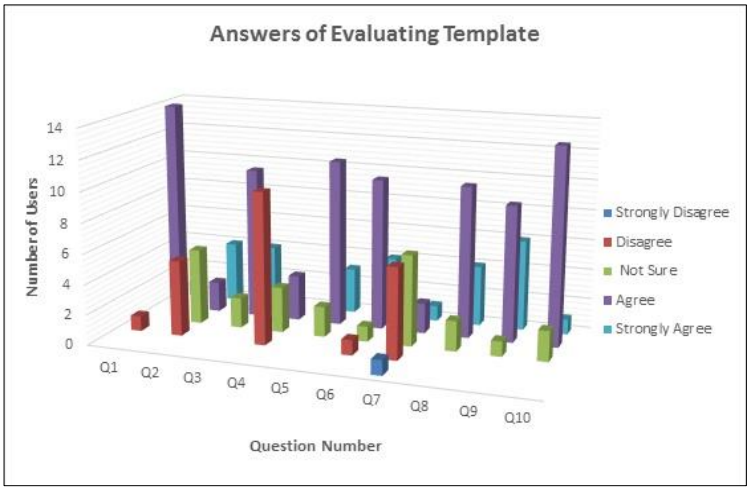

Figure 3 Frequency of Users answers

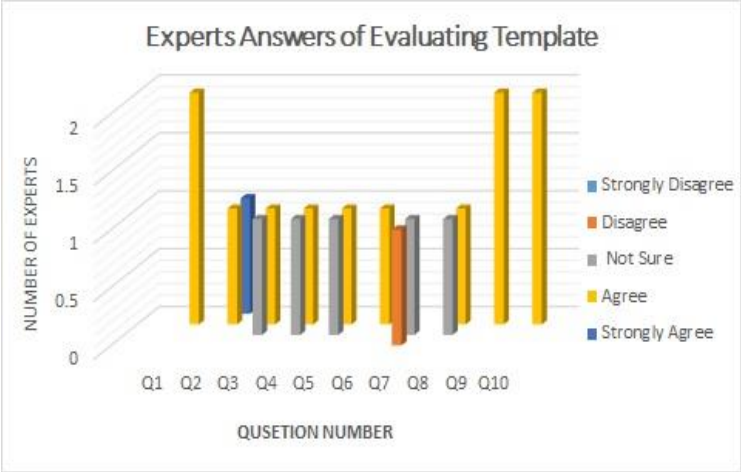

Figure 4 Frequency of Experts answers

The overall evaluation results of the selected methods are analyzed according to two factors: usability, and applicability. The case studies evaluation results presented in Table 5 already show the selected two factors, while for the review evaluation they are derived from the frequency of the reviewer's answers. Table 6 presents the overall evaluation results which show that the proposed template can satisfy the usability and applicability factors, which are important in aiding novice developers.

Table 6 Overall evaluation results

Evaluation method selected			Results	
Method No.	**Method**	**Process**	**Usability**	**Applicability**
1.	Case study	Applied on two software projects	High	High
2.	User's review	Postgraduate students	High	High
	Experts review	Faculty members	High	High

The reviewers highlighted the time complexity in applying the template, as various sub-sections decomposed to many activities which make it highly time-consuming, some tool support is suggested for future work. The SecRS template and previously studied SecRE methods are compared in Table 7, per usability, applicability, time, and guidelines factors. The time complexity is determined to be high as compared to other methods. This problem is planned to be focused on in our future works, where some tool support will be provided to perform some steps/ activities.

Table 7 SecRS template and SecRE methods comparison according to usability and applicability

Factors / Methods	Applicability				Usability		Time	Guidelines
	Large Projects	Small Projects	System Oriented	Machine Oriented	Easy to learn	Easy to Use		
SQUARE	Yes	No	Yes	Yes	Yes	Yes	Less	No
MSRA			Yes				-	No
SREP	Yes	Yes			No	No	Less	No
Misuse-cases	Yes	Yes	-	-	No	No	Less	No
Secure Tropos			Yes	Yes			-	-
CLASP	No	Yes	Yes	Yes	Yes	Yes	High	-
SecRS	Yes	Yes	Yes	Yes	Yes	Yes	High	Yes

CONCLUSION

The SecRS template is proposed to address the research problem. The SQUARE-Extended method is followed in the SecRS template, which combines the CLASP activities of the RE phase with the SQUARE steps. The template is designed to guide novice developers in following the proposed method easily. The template is evaluated to determine its usability and applicability criteria. The evaluation results show that the proposed template can satisfy the usability and applicability factors; thereby confirming that the proposed template achieves its desired objective of aiding the novice developers in identifying and specifying security requirements.

The research contributed to the RE field in the following ways:

- The research reviewed existing literature about SecRE methods to determine their usability and applicability per novice developer; the results will aid in the selection of the most appropriate method for the security requirements identification process.
- The research provides a new method SQUARE-Extended for security requirements identification.
- The research facilitates novice developers in the security requirements identification process through the SecRS template.
- The research helps the students to have a deeper understanding of the security requirements identification and documentation; by presenting the case studies.

Few problems encountered in the research are:

- Limited resources: The resource on available literature is limited.
- Time: the time required in template evaluation to perform the proposed method steps is a lot. The proposed method comprises six steps where the first three steps are decomposed into various activities; each following the four subsections which makes it difficult to get the industry people to apply it.

The proposed SecRS template in the research can be improved in many ways to meet the new emerging trends in research such as:

- Development of an automated tool.
- Provide integration with the agile development method.
- Enhancement of the risk assessment process.
- Incorporation of the change management process for requirements management.

ACKNOWLEDGEMENT

This work is part of the MSE dissertation titled: “Security Requirements Specification (SecRS) template to aid novice developers in identifying and documenting security requirements”, Faculty of Computer Science and Information Technology, University of Malaya.

REFERENCES

[1] Mouratidis H, Giorgini P. Secure tropos: a security-oriented extension of the tropos methodology. International Journal of Software Engineering and Knowledge Engineering 2007.

[2] Jaffe M, Leveson N, Heimdahl M, Melhart B. Software Requirements Analysis for Real-Time Process-Control Systems. IEEE Transactions on Software Engineering 1991, Vol.17, No.3.

[3] Haley C, Laney R, Moffett J, Nuseibeh B. Security Requirements Engineering: A Framework for Representation and Analysis. IEEE Transactions on Software Engineering 2008, Vol. 34, No. 1.

[4] Hadary H, Kassas S. Capturing security requirements for software systems. Journal of Advanced Research 2014, Cairo University.

[5] Ramesh MRR, Reddy DrCS. A survey on security requirements elicitation methods: classification, merits, and demerits. International Journal of Applied Engineering Research 2016. Volume 11, Number 1, pg. 64-70.

[6] Firesmith D. Engineering Security Requirements. Journal of Object Technology 2003, Vol. 2, No.1, pg. 53-68.

[7] Salini P, Kanmani S. A Survey on Security Requirements Engineering. International Journal of Reviews and Computing 2011. Vol.8.
[8] Houmb SH, Islam S, Knauss E, Jurjens J, Schneider K. Eliciting security requirements and tracing them to design integration of Common Criteria, heuristics, and UMLSec. Special Issue – Security Requirements Engineering. Requirements Engineering 2010, pg. 63-93.
[9] Salini P, Kanmani S. Survey, and analysis on Security Requirements Engineering. Computers and Electrical Engineering 2012, Vol. 38, Issue.6, pg. 1785–1797.
[10] Fabian B, Gurses S, Heisel M, Santen T, Schmidt H. A comparison of security requirements engineering methods. Special Issue – Security Requirements Engineering. Requirements Engineering 2010, pg. 7-40.
[11] Mellado D, Blanco C, Sanchez L, Medina E. A systematic review of security requirements engineering. Computer Standards & Interfaces 2010. Vol. 32, Issue.4, pg. 153–165.
[12] Hull E, Jackson K, Dick J. Requirements Engineering. Springer Books, 2011.
[13] The CLASP Application Security Process. Secure Software, Inc. Book in 2005.
[14] Mead N, Hough E, Stehney T. Security Quality Requirements Engineering (SQUARE) Methodology. Software Engineering Institute, Carnegie Mellon University. Technical Report CMU/SEI-2005-TR-009, November 2005.
[15] Pohl K. Requirements Engineering: fundamentals, principles, and techniques. Springer Publishing Company, Incorporated in 2010. Conference proceedings.
[16] Crook R, Ince D, Lin L, Nuseibeh B. Security Requirements Engineering: When Anti-Requirements Hit the Fan. In: IEEE 2002 Joint International Conference on Requirements Engineering (RE'02); 9-13 September 2002; Essen, Germany. IEEE. pg. 203-205.
[17] TKhilji WA. Evaluation Framework for Software Security Requirements Engineering Tools. Master's Thesis (30 ECTS). Faculty of Mathematics and Computer Science, University of Tartu, 2014.
[18] Salini, P., & Kanmani, S. A novel method: Ontology-based security requirements engineering framework. In Emerging Trends in Engineering, Technology, and Science (ICETETS), International Conference on (pp. 1-5). 2016. IEEE.
[19] Yahya, S., Kamalrudin, M., & Sidek, S. A review on tool supports for security requirements engineering. In Open Systems (ICOS), 2013 IEEE Conference on (pp. 190-194). IEEE.
[20] Khan, N. F., & Ikram, N. Security Requirements Engineering: A Systematic Mapping (2010-2015). In Software Security and Assurance (ICSSA), 2016 International Conference on (pp. 31-36). IEEE.
[21] Marquez, G., Silva, P., Noel, R., Matalonga, S., & Astudillo, H. Identifying emerging security concepts using software artifacts through an experimental case. In Chilean Computer Science Society (SCCC), 2015 34th International Conference of the (pp. 1-6). IEEE.
[22] Massacci, F., & Paci, F. How to select a security requirements method? A comparative study with students and practitioners. In Nordic Conference on Secure IT Systems (pp. 89-104), 2012. Springer Berlin Heidelberg.
[23] Schmitt, C., & Liggesmeyer, P. Getting grip on security requirements elicitation by structuring and reusing security requirements sources. 2015, Complex Systems Informatics and Modeling Quarterly, (3), 15-34.
[24] Anwar Mohammad, M.N., Nazir, M. & Mustafa, K. A Systematic Review and Analytical Evaluation of Security Requirements Engineering Approaches. Arab J Sci Eng 44, 8963–8987 (2019).
[25] Khan, R. A., Khan, S. U., Ilyas, M., & Idris, M. Y. The State of the Art on Secure Software Engineering: A Systematic Mapping Study. 2020, Proceedings of the Evaluation and Assessment in Software Engineering, 487-492.
[26] Li, T., & Chen, Z. An ontology-based learning approach for automatically classifying security requirements. 2020, Journal of Systems and Software, 165, 110566.
[27] Mufti, Yusuf, Mahmood Niazi, Mohammad Alshayeb, and Sajjad Mahmood. A readiness model for security requirements engineering. *IEEE Access* 6 (2018): 28611-28631.
[28] Ansari, Md Tarique Jamal, Dhirendra Pandey, and Mamdouh Alenezi. STORE: security threat-oriented requirements engineering methodology. Journal of King Saud University-Computer and Information Sciences, 2018.