Volume 66, Issue 2, 2022
Journal of Scientific Research
of
The Banaras Hindu University

Formalization of Security Requirements-A Case Study on a Web- Based Application

Aditya Dev Mishra [1,2][0000-0002-7060-2600] and K. Mustafa [1][0000-0002-0540-5408]

[1] Jamia Millia Islamia University, New Delhi, India
[2] DIT University, Dehradun, India

adityadev.mishra@dituniversity.edu.in

Abstract. Security of web-based applications is one of the main concerns today. With the expanding cyberspace, utility and threats are increasing simultaneously. Like earlier, even today security is taken as an afterthought. Functional requirements override the security assurance. Therefore, different security issues such as threat, vulnerability, security breaches, etc. may arise after the deployment of web-based applications. The use of formal specifications for the security requirements while developing web-based applications is cost-effective, time-saving, and error-free. Most of the exiting models rarely deal with a formal approach. The objective of this research paper is to provide an idea about the formal specification and formal verification of web-based applications. In this paper, a novel and broad approach are proposed to specify the security requirement. The proposed approach has been verified through a case study of a web-based mobile banking application. This research paper mainly focused on the specification of security properties by considering some security requirements such as authentication, authorization, confidentiality, and integrity. These security properties are formally verified by the existing more formal tools which is an extension of the proposed work.

Keywords: Formal Specification, Formal Verification, Security Requirements, Web-Based Applications, Security Property.

1 Introduction

With increased web-based applications nowadays, security requirements involved in all the phases of the software development life cycle have become critical. Security issues related to web-based applications have drawn attention to security experts. Therefore, there is a pressing need to evolve security requirements from the top to bottom phase during the development of web-based applications. Ambiguity and bugs in security requirements have ever been a prime concern in the software development process. Security requirements are the constraint that is imposed on the system to avoid vulnerability and other security issues. Identification, elicitation, and integration of security requirements at the early phase of software development are important because it reduces the security vulnerability that might be found in the later phase.

The formalization of security requirements involves formal specification and formal verification both. The formal specification is the process to specify security requirements through any formal language such as Z, VDM, and Larch, etc. Formal specification is technology-independent of any formal language. The specified security property has been converted into a model with tool support through model verification. Formal verification verified the specified security properties through formal tools such as JSPIN, NuSMV, theorem provers, etc. Formal methods are well recognized and research challenging field with many mathematical models and tools support. Since formal methods are mainly scientific and higher mathematics approaches, therefore, the implementation of formal specification and formal verification is a research-oriented challenging task. Transformation of natural language security requirements into formal specification is also rigorous research work.

Security requirements of web-based applications are prime concerns in today's world as the use of the internet grows day by day. Mobile banking application is one of the web-based applications provided by the bank to their customer for the financial transaction by a remote access device such as a mobile, tablet, etc. As the uses of these devices increased, their security matter. Therefore, it is a matter of concern for the security experts to scan all the security requirements during development from the requirement phase to the design phase and then testing. Many of the industries do but we cannot ignore the importance of formal specification and formal verification in the development of any web-based applications.

Many researchers proposed different views about formal specification and security requirements. Hussain et al. [16] proposed an idea about the importance of the formal method by most of the existing models are based on informal or semiformal and have flaws. Kazhamiakin et al. [17] proposed a novel framework for formal specification and formal verification of distributed processes in web services. Specification of security property of system by formal method produces an unambiguous, complete, and precise result. Considering the need and significance of the security requirements and formal method, this research work has been undertaken.

The structure of the paper is as follows. Section 2 described the related work in the proposed area by authors and also compares the proposed approach with some existing approaches in terms of four-parameter security requirements, formal model, specification, and verification. Section 3 describes the proposed approach for the formalization of the four security property AACI i.e. authentication, authorization,

DOI: 10.37398/JSR.2022.660214

confidentiality, and integrity. Section 4 of the paper includes a case study of a mobile banking application that shows the way to specify security properties. In section 4, the formal language specifies some security properties authentication, authorization, confidentiality, and integrity. Finally, section 5 includes the summarized proposed work, the lesson learned, and a brief discussion about future work.

2 Related Work

Many researchers proposed a different approach to the specification and verification of security requirements. Agudo, I., & Lopez, J. [1] discuss the importance of the gap between formal analysis and security requirements by introducing the specification of security requirements in the design phase of software development. Biondi, F., & Legay, A. [2] discuss the use of formal methods and tools to improve security at both software implementation and protocol levels. Breaux, T. D. et.al.[3] introduce a formal specification language called Eddy for specification of privacy requirements that enable developers to detect conflict and data flows within policies. Bugliesi et. al. [4] mention in their survey paper the importance of the formal method in web platforms by classifying and review exiting protocols in the area of the formal method for web security. Denisse Muñante et. al. [6] discussed different types of security requirement engineering (SRE) methods (approx. 13) based on model-driven engineering (MDE) approach, risk analysis, and conclude that KAOS and secure i* are the most suitable approach for model-driven because they used standers of development and also validated formally. Hassan El-Hadary and Sherif El-Kassas [7] have proposed a five steps iterates methodology for the elicitation of security requirements based on problem frames. The proposed methodology is compared with the already existing Haley's security requirement methodology. The proposed methodology is used to find threats and eliciting a corresponding security requirement. The authors of this paper also suggested adopting a formal framework into methodology instead of informal language for more preciseness and automation. Busalire Onesmus Emeka and Shaoying Liu [8] in the field of formal methods have presented an idea for identifying security vulnerability from software requirement specification using structured object-oriented formal languages (SOFL). The authors also verified their proposed idea by taking a case study of an online banking system. The proposed method is cost-effective, further be tested for security vulnerability without converting them into executable code. This paper is an extension of the work proposed by Busalire in 2017 [9]. Mariana Gerber et al. [10] proposed a two-dimensional formalization-based approach for determining security requirements. Charles B. Haley et al.[11] have proposed a framework for security requirement elicitation and analysis that satisfies the criteria of definition, assumption, and satisfaction to meet security goals and evaluated by apply security requirements in air traffic control. The proposed framework is an extension of the work published by Charles B. Haley et.al. in 2006 [12]. Riham Hassan et al. [14] proposed a novel approach for design specification from the security requirement by integrating the KAOS method and B-Method. The proposed framework constructs a consistent, complete, and clear security requirement formal model. The main contribution of the paper include derive security requirements in design specifications while preserving security properties. Brahim Hamid and Christian Percebois [15] have proposed a framework for specification and validation of security patterns by using metamodeling technique (semi-formal representation) and theorem proving approach (rigorous formal representation) by choosing an example of secure communication pattern (SCP). Shafiq Hussain et. al. [16] used Z Language to formal specified security properties for syntax and type checking, automatic proofs of the model. Rouland et al. [20] proposed an approach to specify security requirements by using first-order logic, formalized and verified by using the Alloy tool. Seung Ju Jang et. al. [21] take a case study of ACS (Access Control System) and demonstrate how formal specification and verification methodology is used to develop a vulnerability-free secure software system. Subburaj and Joseph E. Urban [22] used Descartes specification language (formal specification language) to specify a variety of application examples such as agent system. Axel van Lamsweerde [23] introduced an approach to modeling, specification, and analysis of application-specific security requirements. The proposed method is based on a goal-oriented framework, for reducing barriers to goal satisfaction. His proposed extended framework explained malicious obstacles set up by attackers to threaten an application`s security goals. Wan, K., Kapoor et.al.[24] proposed a technique for modeling business processes in CSP, translate them into promela languages, and analysis by SPIN tool but leave formal specifications for future work. M.Weiss and H. Mouratidis [25] described a formalization-based novel approach for selecting a security pattern using Goal-Oriented Requirement Language (GORE). This approach describes effectively the contribution of patterns in nonfunctional security requirements. The lesson learned from this paper is formalization automates the pattern selection process. Zhao, Y., and Rozier, K. Y. [26] use formal specification and formal verification techniques for automated air traffic control systems by writing LTL specifications using NuSMV and Cadence SMV for operations and model checking for system verifications.

Based on the survey, it is observed that different authors proposed different techniques for specification and verification of security requirements but some of them used formal specification methods and theorem provers. Specification of security properties is a research-challenging task because it needs a lot of manual work for automation. The formal specification also required the expertise of higher mathematics. Table 1 depicted below shows the comparison of the existing approach with the proposed approach based on some parameters such as security requirements, formal model, specification, and verification.

Table 1. Comparison of the proposed approach with some exiting approach

Parameter / **Ref. Paper**	**Security Requirements**	**Formal Model**	**Specification**	**Verification**
[1]	✓	✓	✓	✓
[3]	✓	x	✓	x

[8]	✓	✓	✓	✓
[10]	✓	✓	✓	x
[11]	✓	✓	✓	✓
[14]	✓	✓	✓	x
[15]	✓	✓	✓	✓
[20]	✓	✓	✓	✓
[21]	✓	✓	✓	✓
[22]	x	x	✓	x
[23]	✓	✓	✓	x
[24]	x	✓	x	x
[25]	✓	✓	✓	x
[26]	✓	✓	✓	✓
Proposed approach	✓	✓	✓	✓

3 Approach

According to the proposed simplistic approach, formalization of security requirements involved in all the SDLC phases from the requirement to the testing phase of web-based application developments is to be done strategically. All the phases are formalized from informal specification to formal specification and then formal verification is to be undertaken, as depicted in Fig 1 as follows. Informal Specification refers to specify the security properties in natural language. Formal Specification calls to specify the security properties in any Formal language, and Formal Verification calls to verify the security properties from formal tools. The proposed approach is depicted as per Fig 1 as follows.

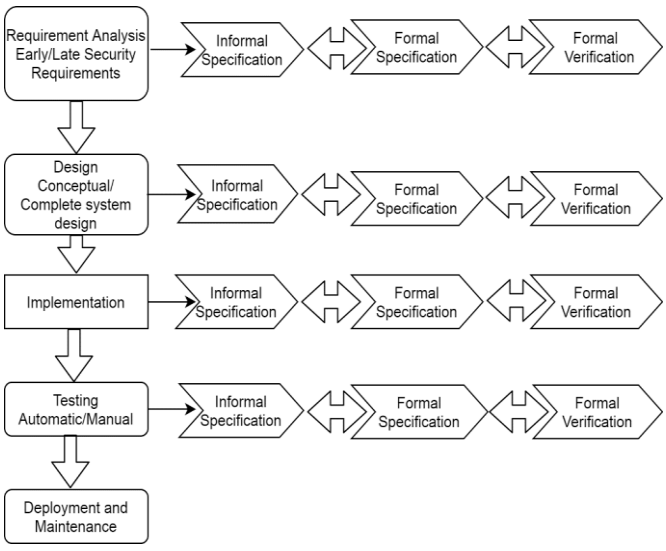

Fig. 1. Formalization of Software Development Life Cycle

Formal specification and Formal verification at each step of software development lead to strong security and produce a better result by the software. Formal methods integrate with the entire software development lifecycle. Early or late security requirements in the Requirement analysis phases are one of the major issues that affect the productivity of the software. These issues will be completely resolved using formal methods. Effective use of Formal Methods in the design phase from conceptual to complete design is also recommended. Implementation of software design by Formal specification is clear, error-free, unambiguous, and timesaving. Manual testing has one of the major issues as all the possible cases are not always considered i.e. completeness property. The formalization of security requirements in the testing phase may solve these issues in terms of reduced ill-definition and automatic testing. Therefore, it is recommended to use formal specification and formal verification at each phase of software development to avoid security vulnerability, effective and efficient outcomes.

3.1 Overview of the Proposed Approach

In this study, the authors proposed an approach for formal specification and formal verification of security properties illustrated below in fig. 2.

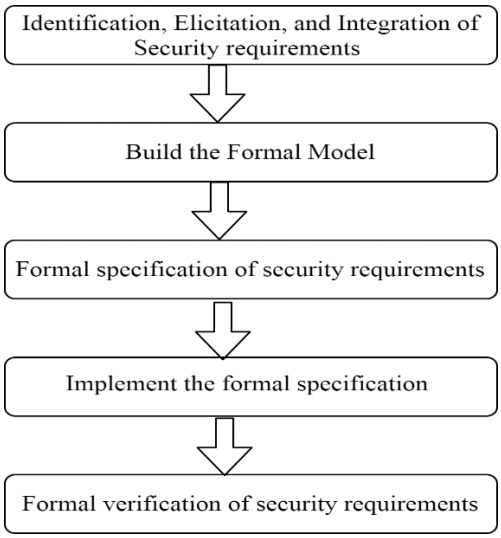

Fig. 2. A process of the proposed approach

3.2 Broad Steps of the methodology

The proposed methodology can be described in discrete steps as follows.

a. Identification, Elicitation, and Integration of Security requirements

Identify the security requirements initially for the system and define the security property that corresponds to each security requirement. The objective is to specify the security property for the system that satisfied the security requirements.

b. Build the Formal Model

In this step, the security requirements model are represented in any formal framework. The formal model described the process description involved in the particular phase of SDLC related to security property. The formal model depicted below in fig. 3 starting with initial states (user) and for every input, every state moves to either 'accept' state or 'reject' state. Various inputs of the formal model are AACI i.e.

authentication, authorization, confidentiality, and integrity.

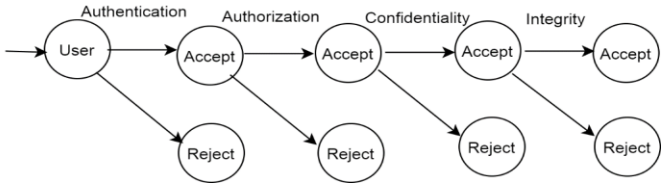

Fig. 3. Formal Model of Security Property

c. Formal specification of security requirements

Formal specification of different model checkers is generally based on the concept of predicate, assertion, and fact. Formal specification of security requirements takes place in two steps as follows.

- Abstraction level- This level describes the security requirements of the web-based application in terms of security properties such as confidentially, integrity, availability, completeness, consistency, authentication, authorizations, etc.
- Process level- This level describes the operation of these security requirements in terms of formal languages such as propositional logic, Z, Tropos, Alloy, Larch, etc. [16,20]

d. Implement the formal specification

In this step, specific security properties are converted into any script language supported by the formal tools. For example, specified security properties are converted into promela used in SPIN Formal tools.

e. Formal verification of security requirements

In this step, verified the security property satisfied security requirements or not. The violence of the security requirements leads the vulnerability and security breaches. Verified security requirements lead better results for the system.

4 Case Study

Let us take a case study of web-based applications such as mobile banking applications (MBA) to evaluate the effectiveness of the proposed approach.

4.1 Identification, Elicitation, and Integration of Security requirements for MBA

In the first step, try to identify the basic security requirements, which are essential during the development of mobile banking applications. We have only considered the four security requirements such as authentication, authorization, confidentiality, and integrity because these security requirements are essential for the development of any mobile banking application. Identification and elicitation of these security requirements must be according to the stakeholder's needs. Integration of these security requirements also fulfills the customer expectations.

4.2 Build the Formal Model for Mobile banking application(MBA)

Formal model for the Mobile banking application depicted below in fig. 4 has been made by considering limited assumptions. This formal model is made by using the concept of deterministic finite automata. In this formal model, there are 9 different states. q_{00} is the initial state and q_{11}, q_{21}, q_{31}, q_{41} are the dead states. q_{10}, q_{20}, q_{30}, q_{40} are the different states starting from q_{00}. The meaning of all the states is described in table 2.

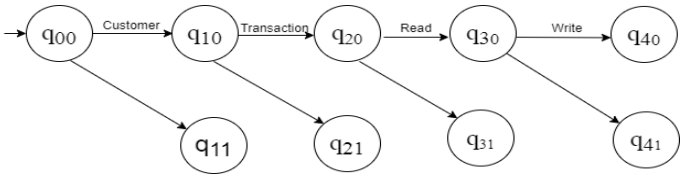

Fig. 4. Formal Model of MBA security properties authentication, authorization, confidentiality, and integrity (AACI)

Table 2. Different states of the model, their meaning and specification

State	Natural Language	Specification
q_{00}	Initial State (all customer)	$(\forall x? : x \in customer)$
q_{10}	Registered customer	$(\forall x? \in Registered\_Customer \wedge id? Registered\_customer\ x?)$
q_{11}	Unregistered customer	$(\forall x? \notin Registered\_customer \wedge id? Registered\_customer\ x?)$
q_{20}	Customer valid transaction	$(\forall tr? \in transaction\_valid \wedge id? registered\_customer\ \ x?)$
q_{21}	Customer not valid transaction	$(\forall tr? \notin transaction\_valid \wedge id? registered\_customer\ \ x?)$
q_{30}	Read operation performed	$(x? \in authenticated\_customer \wedge x? \in authorized\_customer) \rightarrow (tr = READ)$
q_{31}	Read operation not performed	$(x? \notin authenticated\_customer \wedge x? \notin authorized\_customer) \rightarrow (tr \neq READ)$
q_{40}	Write operation performed	$(x? \in authenticated\_customer \wedge x? \in authorized\_customer) \rightarrow (tr = WRITE)$
q_{41}	Write operation not performed	$(x? \notin authenticated\_customer \wedge x? \notin authorized\_customer) \rightarrow (tr \neq WRITE)$

4.3 Formal Specification of Security properties for MBA

The specified security requirements are mapped into the corresponding security property. These security properties are considered as constraints or conditions for the model. Negligence of these security properties indicates that security requirements are not fulfilled for the particular model. Formal specifications of security properties such as Authentication,

Authorization, Confidentiality, Integrity, and Availability are presented in this section as follows [16].

Authentication: This security property described the authenticity of the user credentials in MBA.

$$\Delta\ Mobile\ Banking \text{ Application}$$
$$\forall x? : x \in \text{CUSTOMER}$$
$$\text{id?} : \text{LOGIN DETAILS}$$
$$(\forall x? \in registered_{customer} \wedge id?\, registered_{customer}\ x\,?) \rightarrow authenticated_{\text{customer}'} = authenticated'_{\text{customer}} \cup \{(x? \leftrightarrow id)\}$$
$$els\ authenticated'_{\text{customer}} \neq authenticated_{\text{customer}}$$

Authorization: This security property described only the valid account holder do the transaction in MBA.

$$\Delta\ Mobile\ Banking \text{ Application}$$
$$\forall x? : x \in \text{CUSTOMER}$$
$$\text{id?} : \text{LOGIN DETAILS}$$
$$\text{tr?: TRANSACTION}$$
$$x? : \in authenticated\_customer$$
$$(\forall\ tr? \in transaction\_valid \wedge id?\, registered\_(customer)\ x\,?) \rightarrow authorized\_(\text{customer}^{\wedge\prime}) = authorized\_\text{customer} \cup \{(x? \leftrightarrow tr)\}\ \ els\ authorized_{\text{customer}\prime} \neq authorized_{\text{customer}}$$

Confidentiality: This security property describes the read operation in MBA. Only valid customers check the balance of their account.

$$\Delta\ Mobile\ Banking \text{ Application}$$
$$\forall x? : x \in \text{CUSTOMER}$$
$$\text{tr?: TRANSACTION}$$
$$x? : \in authenticated\_customer$$
$$x? : \in authorized\_customer$$
$$if\ tr = READ$$
$$then\ customer\_transaction' = customer\_transaction \cup \{(x? \leftrightarrow tr)\}$$
$$els\ customer\_transaction' \neq customer\_transaction$$

Integrity: This security property described the write operation in MBA.

$$\Delta\ Mobile\ Banking \text{ Application}$$
$$\forall x? : x \in \text{CUSTOMER}$$
$$\text{tr?:TRANSACTION}$$
$$x? : \in authenticated\_customer$$
$$x? : \in authorized\_customer$$
$$if\ tr = WRITE$$
$$then\ customer\_transaction' = customer\_transaction \cup \{(x? \leftrightarrow tr)\}$$
$$els\ customer\_transaction' \neq customer\_transaction$$

4.4 Implement the Formal specification for MBA

Formal specification of security properties authentication, authorization, confidentiality, and integrity (AACI) is converted into any formal specification language such as Alloy, SPIN, etc. To model the security requirements specified in section 4.3 using Alloy model, we use Predicate, Assertion, and Fact [20]. The security requirements (AACI) are modeled as an assertion authenticationReq, authorizationReq, confidentialityReq, integrityReq as listed below (partially code) in table 3.

Table 3. Listing of security requirements authentication, authorization, confidentiality, integrity by Alloy Model

Authentication Property	Authorization Property
pred *authentication {* *all m: MBA, x:customer, id:logindeatils |* *get_tr{m,x,id} implies sent_to {m,x,id,}* *}* ***assert*** *authenticationReq {* *authentication* *}* ***check*** *authenticationReq* ***for*** *10*	***pred*** *authorization {* *all m: MBA, x:customer, id:logindeatils, tr:transaction |* *get_tr{m,x,id,tr} implies sent_to {m,id,rd}* *}* ***assert*** *authorizationReq {* *authorization* *}* ***check*** *authorizationReq* ***for*** *10*
Confidentiality Property	**Integrity Property**
pred *confidentiality {* *all m: MBA, x:customer, id:logindeatils, tr:transaction, rd:read |* *get_tr{m,x,id,tr} implies sent_to {m,x,id,rd}* *}* *assert confidentialityReq {* *confidentiality* *}* *check confidentialityReq for 10*	***pred*** *integrity {* *all m: MBA, x:customer, id:logindeatils, tr:transaction, wrt:write |* *get_tr{m,x,id,wtr} implies sent_to {m,x,id,wrt}* *}* ***assert*** *integrityReq {* *integrity* *}* ***check*** *integrityReq* ***for*** *10*

4.5 Formal Verification of Security property for MBA

Security properties are verified by different formal verification tools such as SPIN model checker, Alloy model. Considering the case of Alloy Model, these security properties are model as a command such as authenticationReq, authorizationReq, confidentialityReq and integrityReq and verify by the security expert through command check secReq for n. The absence of the counterexample ensures that the security property holds within the model. The check command shows that the model is secured concerning particular security property.

5 Conclusion

This research paper mainly focuses on precise specification and analysis of security requirements for the web-based application (MBA). The novelty of this research included the five steps process for formalization in an effective way. In this paper, we take a case study of a web-based application (MBA) and specified four security properties through formal language by taking a certain case. The proposed approach needs rigorous manual work for the explicit specification, which is probably a difficult research-challenging task. However, a new framework will be developed for checking the syntax, completeness, correctness, and consistency, etc. security properties. Therefore, the objective of the proposed work is to evolve a prescriptive framework that enables security experts to formalized security properties and in turn avoid the inception of security vulnerabilities.

5.1 Future Scope

This research paper includes the specification of the security requirement by formal language. In future work, the proposed broad formalization approach will be extended to specify and

verify more security properties. Formal specification and verification of the security property by formal tools such as SPIN model checker will also consider as future work. The extension of this research work will be a future research-challenging task.

References

[1] Agudo, I., Lopez, J.: Specification and Formal verification of security requirements. In Proceedings of the 5th international conference on Computer systems and technologies, pp. 1-6. ACM (2004).

[2] Biondi, F.,Legay, A.: Security and privacy of protocols and software with formal methods. In International Symposium on Leveraging Applications of Formal Methods, pp. 883-892, Springer, Cham (2016).

[3] Breaux, T. D., Hibshi, H., Rao, A.: Eddy, a formal language for specifying and analyzing data flow specifications for conflicting privacy requirements. Requirements Engineering, 19(3), pp.281-307(2013).

[4] Bugliesi, M., Calzavara, S.,Focardi, R.,: Formal methods for web security. Journal of Logical and Algebraic Methods in Programming, 87, pp.110-126(2017).

[5] Chunlei, W., Minhuan, H., Ronghui, H.: Formally Analyzing Software Vulnerability Based on Model Checking. *International Conference on Networks Security, Wireless Communications and Trusted Computing*, Vol. 1, pp. 615-618. IEEE (2009).

[6] Denisse Muñante, Vanea Chiprianov, Laurent Gallon, Philippe Aniorte.: A Review of Security Requirements Engineering Methods with Respect to Risk Analysis and Model-Driven Engineering. International Cross-Domain Conference and Workshop on Availability, Reliability, and Security (CD-ARES), Fribourg, Switzerland. pp.79-93,10.1007/978-3-319-10975-6_6.hal 01403987(2014).

[7] El-Hadary H, El-Kassas: Capturing security requirements for software systems, Journal of Advanced Research, http://dx.doi.org/10.1016/j.jare.2014.03.001(2014).

[8] Emeka, B. O., and Liu, S.: Assessing and extracting software security vulnerabilities in SOFL formal specifications. In 2018 International Conference on Electronics, Information, and Communication (ICEIC), pp. 1-4. IEEE (2018).

[9] Emeka, B. O., Liu, S.: Security Requirement Engineering Using Structured Object-Oriented Formal Language for M-Banking Applications. In 2017 IEEE International Conference on Software Quality, Reliability and Security (QRS), pp. 176-183. IEEE (2017).

[10] Gerber, M., von Solms, R., Overbeek, P.: Formalizing information security requirements. Information Management & Computer Security, 9(1), pp.32-37(2001).

[11] Haley, C., Laney, R., Moffett, J.,Nuseibeh, B.: Security requirements engineering: A framework for representation and analysis. IEEE Transactions on Software Engineering, 34(1), pp.133-153(2008).

[12] Haley, C. B., Moffett,J. D., Laney, R., Nuseibeh, B.:A framework for security requirements engineering. In Proceedings of the 200 international workshop on Software engineering for secure systems, pp. 35-42. ACM (2006).

[13] Haley, Charles B.; Laney, Robin C., Nuseibeh, Bashar: Deriving security requirements from crosscutting threat descriptions. In Proceedings of the 3rd international conference on aspect-oriented software development, ACM Press, New York, USA, pp. 112–121(2004).

[14] Hassan, R., Bohner, S., and El-Kassas, S.: Formal derivation of security design specifications from security requirements. In Proceedings of the 4th annual workshop on Cyber security and information intelligence research: developing strategies to meet the cyber security and information intelligence challenges ahead, pp. 10. ACM (2008).

[15] Hamid, B., Percebois, C.: A modeling and formal approach for the precise specification of security patterns. In International Symposium on Engineering (2014).

[16] Hussain, S., Dunne, P., Rasool, G.: Formal Specification of Security Properties using Z Notation. Research Journal of Applied Sciences, Engineering and Technology, 5(19), pp.4664-4670(2013).

[17] Kazhamiakin, R., Pistore, M., Roveri, M.: Formal verification of requirements using spin: A case study on web services. In Proceedings of the Second International Conference on Software Engineering and Formal Methods, 2004. SEFM 2004, pp. 406-415. IEEE (2004).

[18] Menzel, M., Thomas, I., Schüler, B., Schnjakin, M., Meinel, C.: Security Requirements Specification in Process-aware Information Systems. In ISSE 2009 Securing Electronic Business Processes, pp. 145-154. Vieweg+ Teubner (2009).

[19] Mishra, A. D., Mustafa, K.: Security Requirements Specification: A Formal Method Perspective. In 2020 7th International Conference on Computing for Sustainable Global Development (INDIACom), pp.113-117.IEEE (2020).

[20] Rouland, Q., Hamid, B., Bodeveix, J. P., Filali, M.:A Formal Methods Approach to Security Requirements Specification and Verification. In 2019 24th International Conference on Engineering of Complex Computer Systems (ICECCS), pp. 236-241. IEEE (2019).

[21] Seung-Ju, J., Jungwoo, R., and Chang, L. Y.: Design of software security verification with formal method tools. International Journal of Computer and Network Security, 6(2006).

[22] Subburaj, V. H., Urban, J. E.: Formal specification language and agent applications. In Intelligent Agents in Data-intensive Computing, pp. 99-122. Springer, Cham (2016).

[23] Van Lamsweerde, A.: Elaborating security requirements by construction of intentional anti-models. In Proceedings of the 26th International Conference on Software Engineering, pp. 148-157. IEEE Computer Society (2004).

[24] Wan, K., Kapoor, H. K., Das, S., Raju, B., Man, K. L., Krilavičius, T.: Modelling and verification of compensating transactions using the Spin tool. In Engineers and computer scientists: IMECS: proceedings of the international Multiconference, pp.14-16, Hong Kong. Vol. 2. Newswood Limited (2012).

[25] Weiss, M., Mouratidis, H.: Selecting security patterns that fulfill security requirements. In 2008 16th IEEE International Requirements Engineering Conference, pp. 169-172. IEEE (2008).

[26] Zhao, Y., Rozier, K. Y.: Formal specification and verification of a coordination protocol for an automated air traffic control system. Science of Computer Programming, 96, pp.337-353(2014).

[27] Akhawe, D., Barth, A., Lam, P. E., Mitchell, J., & Song, D. (2010, July). Towards a formal foundation of web security. In 2010 23rd IEEE Computer Security Foundations Symposium (pp. 290-304). IEEE.

[28] Bugliesi, M., Calzavara, S., & Focardi, R. (2017). Formal methods for web security. Journal of Logical and Algebraic Methods in Programming, 87, 110-126.

[29] Bansal, C., Bhargavan, K., Delignat-Lavaud, A., & Maffeis, S. (2014). Discovering concrete attacks on website authorization by formal analysis 1. Journal of Computer Security, 22(4), 601-657.

[30] Fett, D., Küsters, R., & Schmitz, G. (2014, May). An expressive model for the web infrastructure: Definition and application to the browser id sso system. In 2014 IEEE Symposium on Security and Privacy (pp. 673-688). IEEE.

***