Security Requirements Classification into Functional and Non-functional

Baraa KHAMAYSEH
College of Graduate Studies
Palestine Polytechnic University, Palestine
206034@ppu.edu.ps

Nabil ARMAN
Department of Computer Science and IT
Palestine Polytechnic University, Palestine
narman@ppu.edu

Faisal KHAMAYSEH
Department of Computer Science and IT
Palestine Polytechnic University, Palestine
faisal@ppu.edu

Abstract

Throughout the requirement engineering, the classification of user requirements has been a major concern of software engineering researchers. In this paper, we first highlighted the contributions of recent authors and researchers and their discussions regarding different methods of classification techniques. We mainly presented recent research endeavors that investigate the classification techniques regarding security requirements. We hence devoted a special concentration on security requirements definition as functional and non-functional as the first step in security requirements classification. Finally and by incorporating the different features of requirements, we determined the research gap towards future efforts to use some specified classification factors towards classifying security requirements into functional and non-functional requirements having in account the system and language features.

Keywords*:* Functional requirements, non-functional requirements, classification techniques, security requirements, security metrics.

Introduction

Unfortunately, some software engineers hardly make a distinction between functional requirements (FR) and non-functional requirements (NFR) and forget that some of them can be considered as both FR and NFR. One of these requirements is security. It is known that NFRs describe important constraints upon the development and behavior of a software system, while FR describes what the system should do or deliver; as discussed by Farkhani and Razzazi (2006); Wiegers and Beatty (2013); Missaoui et al (2019); Alsaleh and Haron (2016); Chung et al (2000); Glinz (2007); Dabbagh et al (2015); Cleland-Huang et al (2007) and Cleland-Huang et al (2006). There are numerous techniques and tools available to enable the analysis, creation, and verification of functional requirements right from the start of a project. The classification activity of security needs into functional and non-functional groups is crucial for the engineering and for the development of secure systems. Farkhani and Razzazi (2006) mentioned that this is important because security requirements are essential to a system's operations and inevitably raise the cost when delayed.

The objective of this research paper is to classify security requirements into FR and NFR. In this study, we present the classification techniques used in classifying general requirements into FR and NFR as well as our proposed classification technique used in identifying and classifying the security requirements into FRs and NFRs.

In fact, a substantial and considerable requirement is security in which most software systems require to be applied. This type of requirement is usually viewed as NFR to mean that the system should be secure enough against external attacks or unauthorized actors. Security, on the other hand, also defines a functional action that the system should perform. We addressed this issue in detail in this research paper.

Security Requirements

Security requirement is a specific issue that should exist in most systems. Security as a constraint is a user and system requirement that fulfill the security objectives. Security is one group of other non-functional requirements. There exists different sets of non-functional requirements. Examples of non-functional requirements include: performance, scalability, portability, usability, compatibility, localization, reliability, security, maintainability, capacity, availability, recoverability, serviceability, regulatory, manageability, environmental, data integrity and interoperability. Security is now considered as one of the primary quality features of the system. Wheeler (2015) stated that when security is to be viewed as a function, it means that the system should perform security actions and hence functional requirements can be tested.

When security is weakly applied, the system is then prone to misuse via security flaws that are discovered by an attack while the system is in use. According to Dawson et al (2010), there is a considerable chance in this situation for attackers to misuse system assets from within and threats outside and hence gain access to sensitive data before a leak is detected and discovered. According to Hofbauer et al (2019), when this situation occurs, the cost will be extremely unexpected. Therefore, when talking about system security or information security, security is more of an art than a science, and mastering protective information necessitates cross-disciplinary knowledge of a vast amount of knowledge, along with experience and abilities. This also means that the accuracy of the user account information must be guaranteed by the program as stated by Whitman and Mattord (2022) and Nieles et al (2017).

Schneider et al (2012) indicated that security requirements may be implicit, hidden, and spread out over different documents throughout different locations. For example, one requirement may call for easy web access; a different statement may require online shopping features. According to the study by Irshad et al (2018), regarding software applications, the software requirements contain the information on the needs of a user or the details of a contract that is formally imposed.

User Requirements Classification Techniques

Software requirement classifications affect the other activity of software development (SD). Abad et al (2017) assumed that organizing the requirements whether they are user requirements or other in different categories help further activities and may ease the next classification steps. For example, prioritization which deals with ordering the requirements is facilitated by such classification.

Most requirements are classified using two main categories: Functional Requirement (FR) and Non-Functional Requirement (NFR). Authors usually concentrate on non-functional requirements classifications. Two main approaches were discussed in Cleland-Huang et al (2007), mainly elicitation and detection techniques. Authors highlighted the keyword classification method where they discussed security and performance where they classified security into keyword terms including Confidentiality, integrity, completeness, accuracy, perturbation, virus, access, authorization, rule, validation, audit, biometrics, card, key, password, alarm, encryption, noise as discussed in the study of Cleland-Huang et al (2007) and Cleland-Huang et al (2006).

Types of Classification Techniques

There are three main techniques used for classifying requirements:

1) Automated classification: Using machine learning techniques, classifiers have grown significantly in importance outside of the realm of software engineering. Abad et al (2017) stated that ML classifiers have produced the greatest results across a variety of disciplines when compared to other classification methods. According to the study by Pranckevicius and Varcinkevicius (2017), the well-known classification Algorithms include Logistic Regression, Naive Bayes, K-Nearest Neighbors, Decision Tree and Support Vector Machines.

2) Manual classification: As manual classification consumes time and requires effort from both analysts and experts, many research papers such as Elazhary (2011) have tried to successfully transfer the task of classification to automation either via traditional software solutions or using ML models.

Some methods may provide a separate security engineering process, however this increases administrative burden and reduces analysis integrity. Regarding FR and NFR, researchers use specific set of classification techniques:

a. *NFR classification techniques*: Researchers classified NFR into different sets where they agree on the classification in general regardless of some sub-classifications or details. Khurshid et al (2022) classified NFR considering machine learning algorithms into Logistic Regression (LR), Support Vector Machine (SVM), Multinomial Naive Bayes (MNB), K-Nearest Neighbors (KNN), ensemble, Random Forest (RF), and hybrid KNN rule-based machine learning (ML) algorithms.
   - *Naive Bayes:* Naive Bayes is one method for categorizing the non-functional needs from the requirement document. According to studies by Khurshid et al (2022) and Aruna (2020), authors argued that the likelihood is estimated in this method to categorize non-functional requirements. The PROMISE dataset is used and focuses on nine non-functional requirements. With 15 iterations, leave one out, cross-validation is used for evaluation. Precision for the classifier is 12.4%, which is a very bad result. The above classifier needs to be improved, according to its classification error.
   - *Decision Trees***:** Decision trees are also used to classify the non-functional requirements (Hussain et al, 2008). PROMISE dataset is used, and 10-fold cross-validation is performed for evaluation. The accuracy of classification is 98.56%. The study above has not mentioned the types of non-functional requirements.

b. *FR classification techniques*: Researchers used many similar techniques of NFR in classifying FR with some different constraints to discriminate between functional and non-functional, such as Naïve Bayes, Bayes net, decision tree and KNN as mentioned by Sharma (2017).

3) Semi-automated Classification: Singh et al (2016) analyzed the non-functional requirements based on specified rules. Recently, some researchers including Shehadeh et al (2021); Nassar and Khamayseh (2015); Alami et al (2017) and Alami et al (2020) classified NFR using NLP tools. They classified user requirements written in Arabic language into FR and NFR using a set of heuristics based on basic constructs of Arabic sentences in order to extract required information. Many NLP tools may help in tokenizing and tagging sentence tokens such as MADA-TOKAN, MADAMIRA, Stanford tagger, etc.

User Requirements Metrics

Recent research attempts such as Alami et al (2017) and Alami et al (2020) presented a structured methodology for measuring the quality of requirements whether requirements are functional or non-functional. In this research, authors characterized requirements by ten quality factors, each with an associated metric. The related quality factors include: correctness, Completeness, Connectivity, Consistency, Clarity, Non-ambiguity, Singularity, Testability, Modifiability and Feasibility. The metric is only of value when applied to assist in decision making. For example, the metric for a single requirement is a number between 0 and 1 where 1 represents a "perfect" requirement and 0 represents a totally defective one. Sommerville (2011) classified metrics based on non-functional features including speed, size, ease of use, reliability, robustness and portability. Speed is measured mainly as transaction per second, user per event-response-time and screen refresh time. The Size is measured as Mbytes and the number of ROM chips. Ease of use is measured via training time and available help. Reliability can be measured mainly by assessing availability, failure mean time and unavailability probability. Robustness can be measured via restarting time and corruption data when failure occurs. The portability can be measured via assessing the target systems and the target dependent statements.

Security requirement metrics are related to a quantifiable measurement that understands and evaluates the security level in the system and determines whether security protection mechanisms exist and fulfill defined security requirements, and hence helps organizations to appraise the trust and confidence.

Classification of Security into FR and NFR

Most systems, especially business, have security requirements. Security is mostly known as a non-functional requirement assuring all data inside the system be protected against malware attacks or unauthorized access. On the other hand, security non-functional requirements can be translated into concrete functional counterparts or itself be a functional requirement. In this case the functional security requirement specifies a security action that the software must be able to deliver. Therefore, the functional security requirements are a subset of the overall required functional requirements.

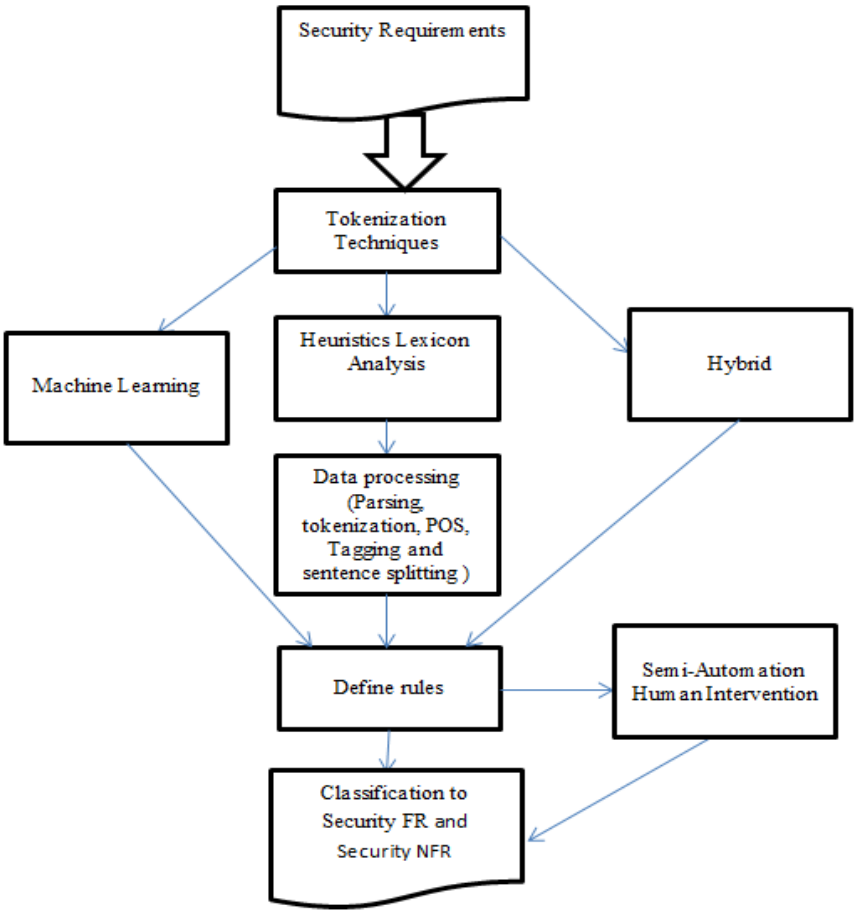

Fig 1: Security Requirement Classification Techniques

Firesmith (2003) presented the various types of security requirements with brief discussion. These types of security needs include the following:

1. Identification Requirements
2. Authentication Requirements
3. Authorization Requirements
4. Immunity Requirements
5. Integrity Requirements
6. Intrusion Detection Requirements
7. Non Repudiation Requirements
8. Privacy Requirements
9. Security Auditing Requirements
10. Survivability Requirements
11. Physical Protection Requirements
12. System Maintenance Security Requirements

In Fig 1, we depict a proposed approach as our main finding in our study. These steps classify the security requirements into corresponding FR and NFR. In this approach we assume the security requirements were provided as a list of straight forward statements. So, rhetoric of language phrases is not assumed since we do not analyze this structure mode. In our procedure, the given security statements should be tokenized using one of suitable NLP tools. Each security requirement is then classified using the set of defined heuristics. The security requirements can be classified into FR, NFR or other where ambiguity may occur.

Conclusion

The presence of security flaws means that the system is misuse-prone via the security flaws that are exposed to an attack while the system is in use. So, the classification of requirements and determining the security ones is extremely required. Various classification methods have been identified and stated based on several previous efforts. Different metrics regarding measuring the security requirements have been mentioned. In this study, we highlighted the previous contributions regarding security classification methods. Finally, we characterized the next future research concerning classification techniques of security requirements into functional and non-functional.

References

Farkhani T. R. and Razzazi M. (2006), 'Examination and Classification of Security Requirements of Software Systems,' 2nd International conference on Information and Communication Technologies, 2778 - 2783. 10.1109/ICTTA.2006.1684851.

Dawson M., Burrell D., Rahim E. and Brewster S. (2010), 'Integrating Software Assurance into the Software Development Life Cycle (SDLC),' Journal of Information Systems Technology and Planning, 3(6), 49-53.

Hofbauer D., Ivkic I., Maksuti S., Aldrian A. and Tauber M. (2019), 'On the Cost of Security Compliance in Information Systems.' 10th International Multi-Conference on Complexity, Informatics and Cybernetic, (IMCIC).

Whitman M. E. and Mattord H. J. (2022), 'Principles of Information Security,' Seventh Edition. Cengage Learning Inc.

Schneider K., Knauss E., Houmb, S. et al. (2012), 'Enhancing security requirements engineering by organizational learning,' Requirements Eng. 17, 35–56. https://doi.org/10.1007/s00766-011-0141.

Irshad M., Petersen K. and Poulding S. (2018).'A systematic literature review of software requirements reuse approaches,' Information and Software Technology, 93, 223-245,ISSN 0950-5849.

Firesmith D. (2003). 'Engineering Security Requirements,' Journal of Object Technology, 2(1), 53-68. http://www.jot.fm/issues/issue_2003_01/column6 Engineering.

Wiegers K. and Beatty, J. (2013). 'Software Requirements,' Third Edition. Microsoft Press. ISBN 978-0-7356-7966-5.

Missaoui E., Mazigh B., Hilaire V. and Bhiri S. (2019). 'Non-Functional Norms Specification and Verification Approach for Normative Multi-agents Systems,' Procedia Computer Science, (159), 717-726.

Wheeler D. A. (2015), 'Secure Programming HOWTO,' v3.71. https://dwheeler.com/secure-programs.

Nieles M., Dempsey K. and Pillitteri V. Y. (2017).' An Introduction to Information Security , ' National Institute of Standards and Technology, Special Publication 800-12, Revision 1.

Alsaleh S. and Haron H. (2016), 'The Most Important Functional and Non-Functional Requirements of Knowledge Sharing System at Public Academic Institutions: A Case Study, ' LNSE, Vol .4(2). http://www.lnse.org/show-45-293-1.html.

Abad Z. S. H., Karras O., Ghazi P., Glinz M., Ruhe G. and Schneider K. (2017). 'What Works Better? A Study of Classifying Requirements, ' Proceedings of the 2017 IEEE 25th International Requirements Engineering.

Elazhary H.(2011). Translation of Software Requirements. Int. J. Sci. Eng. Res, 2, 1–7.

Khurshid I., Imtiaz S., Boulila W., Khan Z., Abbasi A., Javed A and Jalil Z. (2022). 'Classification of Non-Functional Requirements From IoT Oriented Healthcare Requirement Document,' Front Public Health. 10:860536. doi: 10.3389/fpubh.2022.860536. PMID: 35372217; PMCID: PMC8974737.

Hussain I., Kosseim L. and Ormandjieva O. (2008). 'Using linguistic knowledge to classify non-functional requirements in SRS documents,' International Conference on Application of Natural Language to Information Systems. London: Springer, 287–98.

Chung L, Nixon B, Yu E, Mylopoulos J. (2000). 'Non-functional requirements in software engineering,' ISBN: 978-1-4613-7403-9, Kluwer Academic Publishers, Boston.

Dabbagh M., Lee S. and Parizi R. (2015), 'Functional and non-functional requirements prioritization: empirical evaluation of IPA, AHP-based, and HAM-based approaches,' Soft Computing. 20. 10.1007/s00500-015-1760-z.

Cleland-Huang J., Settimi R., Zou X. and Solc P. (2007). 'Automated classification of non-functional requirements,' Requirements Engineering, 12(2), 103–120. doi:10.1007/s00766-007-0045-1. https://www.researchgate.net/publication/221222666_The_Detection_and_Classification_of_Non-Functional_Requirements_with_Application_to_Early_Aspects.

Cleland-Huang J., Settimi, R., Zou X. and Solc P. (2006). 'The Detection and Classification of Non-Functional Requirements with Application to Early Aspects, ' 14th IEEE International Requirements Engineering Conference (RE'06), 36-45. 10.1109/RE.2006.65.

Glinz M. (2007). 'On Non-Functional Requirements,' Proceedings of the 15th IEEE, International Requirements Engineering Conference,2007, Delhi, India.

Pranckevicius T. and Varcinkevicius V. (2017). 'Comparison of Naïve Bayes, Random Forest, Decision Tree, Support Vector Machines, and Logistic Regression Classifiers for Text Reviews Classification'' Baltic J. Modern Computing, (5), No. 2, 221-232 http://dx.doi.org/10.22364/bjmc.2017.5.2

Aruna E., Reddy A., Sunitha K. (2020). 'Elicitation and analysis of security requirements and patterns for IoT based health monitor,' Advances in Cybernetics, Cognition, and Machine Learning for Communication Technologies. Singapore: Springer, 49–56.

Sharma R. (2017). 'Grounding Functional Requirements Classification in Organizational Semiotics, ' In (Ed.), Interdisciplinary Approaches to Semiotics. IntechOpen. https://doi.org/10.5772/67608.

Shehadeh K., Arman N. and Khamayseh F. (2021). 'Semi-Automated Classification of Arabic User Requirements into Functional and Non-Functional Requirements using NLP Tools,' 2021 International Conference on Information Technology (ICIT), 2021, 527-532.

Singh P., Singh D. and Sharma A. (2016). 'Rule-based system for automated classification of non-functional requirements from requirement specifications,' International Conference on Advances in Computing Communications and Informatics (ICACCI), 620-626, doi: 10.1109/ICACCI.2016.7732115.

Nassar I. and Khamayseh F. (2015). 'Constructing Activity Diagrams from Arabic User Requirements using Natural Language Processing Tool,' The 6th International Conference on Information and Communication Systems. ICICS 2015, Amman, Jordan.

Alami N., Arman N. and Khamayseh F. (2017). 'A semi-automated approach for generating sequence diagrams from Arabic user requirements using a natural language processing tool,' 8th International Conference on Information Technology (ICIT), 309-314, IEEE.

Alami N., Arman N. and Khamayseh F. (2020). 'Generating sequence diagrams from Arabic user requirements using Mada+Tokan tool,' Int. Arab J. Inf. Technol., 17(1), 65-72.

Sommerville I. (2011). Software Engineering, 9th ed, Pearson Education, Inc., publishing as Addison-Wesley.