International Journal of Security and Its Applications
Vol. 11, No. 10 (2017) pp.45-54
http://dx.doi.org/10.14257/ijsia.2017.11.10.05

Integration of Security Non-Functional Requirements and Architectural Design: A Comparative Analysis

Muhammad Babar[1]*, Shahid Azeem[2] and Fahim Arif[1]

[1]*National University of Sciences and Technology, Islamabad, Pakistan*
[2]*National Database and Registration Authority, Pakistan*
[1]*babar.phd@students.mcs.edu.pk*, [2]*shahid.azeem@nadra.gov.pk*,
[3]*fahim@mcs.edu.pk*

Abstract

For the last few decades, security in software has gained too much attention by the industries. Developing secure software needs to emphasis on the functional and non-functional requirements both. Functional requirements are taken into account during the early stages of development while unfortunately the non-functional requirements are either ignored or less considered which results in the high cost of maintenance after delivery of the software. This article presents a detailed and comprehensive survey with regard to the integration of security non-functional requirements into architectural design. This paper thoroughly analyzes the existing approaches which are dealing the non-functional requirements at architecture level. The architectural design can be integrated with general non-functional requirements, but the scope of this particular article is only the security related non-functional requirements. The approaches which are comprehensively described and analyzed are use case/misuse cases, goal-based analysis, scenario-based, reused-based, pattern-based, and aspect-based. We have evaluated each approach by some parameters which are described based on the existing literature and comparison has been made between the current approaches thorough proper evaluation.

Keywords: *Security Non-Functional Requirements, Architectural Design*

1. Introduction

Threat modeling proved very useful to capture the threats for the business critical system. Threat modeling process consists of the formally characterizing the system, identifying assets, threats and documentation. If the security requirements are faulty then the threat modeling is applied to ensure correctly capturing the security requirements. In the threat-modeling threats can be ranked or prioritized either by damage or by likelihood [23]. Security requirements can be elaborated by building intentional anti-models approach [13]. In this goal based approach security is threatened by the malicious obstacles which are setup by the attackers to breach the security and achieve their ends. There are three steps in building of security counter measures. In the first step vulnerabilities or anti-requirements threat-trees are determined through anti-goal refinement by using the threat trees. In the second step intentional anti-models are used to build the alternative counter measures to handle different vulnerabilities and threats found. In the third step alternative counter measures are found, preference is made to select the best suitable measures based upon the criticality of the goal. The weakness of the intentional anti-models is that it does not describe the cycle of requirements and build catalogues of threat patterns for the relevant best patterns. Another drawback is that we can’t incorporate the trust models and probabilistic frameworks.

Received (February 1, 2017), Review Result (August 29, 2017), Accepted (September 26, 2017)

ISSN: 1738-9976 IJSIA

Mamadou *et al* made a comparison of the three approaches for the security requirements, namely common criteria, misuse cases, and attack trees [4]. The common criteria evaluate the information technology security by examining the target of evolution (TOE) to find the security environment and its objectives. The security requirements are then deduced from these objectives. They mapped the threats with the objectives and security requirements. They found that the attack trees are better in dealing the security requirements by taking five parameters *i.e.*, learnability, usability, solution inclusiveness, clarity of output and analyzability. Moreover they concluded that by combining these three techniques more information can be collected. GolnazElahi, *et al* proposed a methodological framework for security requirements elicitation and analysis while putting focus on vulnerabilities [22]. They analyzed the importance of attacks and vulnerabilities and found the types of attacks. In their work they chose i* requirement model and extended it with analysis of security requirements. Their proposed model consists of identifying requirements, adding and propagating vulnerabilities, adding attacks to relate with vulnerabilities, developing attack profiles; assessing risk using goal model evaluation, adding counter measures and their impacts and finally analyzing goal model evaluation.

Kenneth Edge *et al* performed a case study on an online banking system and suggested the use of attack trees and protection trees [25]. At first they used the attack trees to identify the vulnerabilities and then used the protection trees for the mitigation of those threats. Protection trees behave the same way as of attack trees as it uses the AND/OR decomposition of goals and threats. After applying the protection tree for the mitigating the threats first iteration completes and again the attack tree is implemented to identify the remaining vulnerabilities. A metric is applied to estimate the severity of risks associated but the before the metrics are applied the characteristic of metrics must be known. They mentioned the specific metrics for attack as probability, cost, impact, and risk and assigned numerical values to calculate the impacts. It is very important to use the suitable architecture which provides security as well as cost effectiveness. To this end Takao Okubo *et al* proposed a method called TMP-SA2 (Twin peaks model Application for security Analysis) while refining the security requirement and architectures [24]. They avoided backtrack during the refinement process. The proposed method provides the benefit of prediction of impacts on the NFR related to the security and thus we can minimize the development costs. They highlighted the issues in the previous work as there is gap between requirements and the design; no method exists to choose the architectural alternative. To address these issues they applied the goal oriented requirement analysis (GORA), knowledge based approaches, pattern based approaches to multiple architectures and based on the impact they showed that their proposed method works well.

Due to the time pressure and requirement of reduced time to market Software product line (SPL) has gained more industry attention as the software can be developed by sharing the common features from packages. However, the verification of the NFR related to SPL is not feasible so Fatima Zahra conducted a literature survey and proposed a methodology to verify the software product lines [26].

2. Existing Approaches for Dealing Security Non-functional Requirements at Architectural Design

2.1. Goal Based

Axel van Lamsweerde proposed the idea of building two types of models iteratively and concurrently [13]. First is a model of future system and second is the anti-model which describes how the specifications of model elements are threatened. They suggested a three step process. In the first step goals are decomposed, in the second step goals are refined and in the third step agents are assigned to goals. They also suggested to building the threat trees, intentional anti-models and alternative counter measures. They used intentional anti-models like questionnaire (containing WHO and WHY), regressed anti-goal specifications and applied formal refinement patterns. In the GQM approach the goals are refined into the questions and then metrics are used to get the results in quantitative way [2]. They mapped the SOA key indicators with the related security goals and metrics to elaborate the requirements and their prioritization. The GCT used the Soft goal Interdependency Graph (SIG) to model the non-functional requirements and their tradeoffs. The road management system is applied as a case study [3].

Oladimeji and Sam Supakkul proposed a three step process for the security threats mitigation [16]. In the first step security threats are elicited, in second step threats and risks are analyzed and associated, and in the third step counter measures are applied to achieve the objectives. The catalogs of common threats are made then to be reused at the later stages from the repository. The threats are associated to security goals by using soft goal independency graph. Qingfeng He and Annie I. Anton employed a healthcare system for implementing the proposed framework [12]. They applied security parameters and evaluated with the other existing methods like the KAOS Framework, i* framework, NFR Framework and Analytical Role Modeling Framework. They classified the level of support as Yes, No, and Partial. Their approach serves as a bridge between requirement analysis and design activities.

Benjamin *et al* made a comparison of the security requirement engineering approaches such as common criteria, secure Tropos, SREP, MSRA, UML based method and problem frame based methods [1]. They compared the goal based security requirement methods with the parameters like stack-holders' views, multi-lateral, orientation towards system or machine, Quality assurance and formality. They found by the comparison that KAOS along with anti-models Secure i*, Secure Tropos are useful for the requirements elicitations and dealing the security related issue in software architecture.

2.2. Patterns-Based

Thomas Heyman, *et al* proposed a mechanism, where three constraints of the approach [18]. First is discovering security metrics, second is the selection of discovered metrics and third is the interpretation of the metrics. The researchers here focused on third constraint of selecting and interpretation of the metrics. They suggested that the patterns when assigned the values of metrics can measure the security of software in quantitative way. The other task was to select the appropriate metric which was done by the dependency graph. The main advantage of associating the metrics with patterns is that it can be easily integrated into applications.

M. Weiss wt all, In their work patterns are selected and formalized in terms of goal oriented requirement languages and by using the prolog rules [19]. The effects of combining the security patterns can be visualized by the prolog rules. Eduardo B. Fernandez proposed an approach that is based on security principals and object oriented development techniques[5]. They modeled the security requirements for the voting systems through the use of use cases and relating them with the possible attacks. They used the patterns for the development of secure software as patterns use the knowledge and experience of the professionals. Yijun Yu and YingfeiXiong used the *i goal model to capture the security requirements through the applying security patterns [20]. They presented how ATL model query and transformation framework is used to express and enforce the role based access control security pattern to capture stakeholder's requirements.

Thomas *et al* presented a model called security twin peaks model which integrates the security requirements into architecture [10]. The security twin peaks method addresses the problem peak and solution peak. For the problem peak they used the goal based method and for the solution peak they used the attribute driven approach for designing the architecture. The goals are decomposed into the sub-goals and then the conflicts are managed. They mentioned that the security patterns consist of security requirements, roles and goals and proposed the security patterns should be integrated into the requirements by means of these three components. They used the traceability links for the integration of requirements from time to time and selection of the solutions for the changing requirements.

2.3. Misuse Case Based

Misuse cases are easy to learn, use and identify the solutions but understanding the diagrams is a tedious activity [4]. Raimundas et all supported the use of textual templates for misuse cases and advised to get more details about the threats so that misuse cases can be analyzed in depth [21].J McDermott has used abuse cases to determine that how interaction of the system and actor can be harmful. They applied the abuse cases on the internet based information security lab to determine the negative impacts [17]. It was found that the strength of the abuse case models is that they save a lot of time by dealing the requirements at abstract level. Another advantage is that abuse cases can be used to make tradeoffs at architectural level. However, there are two weaknesses of the abuse cases. Firstly, abuse case models are that they don't get into details of specific user's domains and security specialist need at work to implement. Guttorm *et al* proposed a systematic approach for elicitation of misuse cases and highlighted the strengths and weaknesses of the misuse cases [6].

FabricioBraz *et al* suggested using the security policies for the identified misuse cases in order to stop or mitigate them [11]. They considered three aspects of dealing with threats. First is the use of use case, second is about the set of security attributes and third is about the source of the threats. They performed a case study of banking system to model the security requirements and listed the actions, actors' sources of threats & threat types. They created a relationship between threats, policies and security patterns. They mentioned that misuse cases alone provide no solution of what should be considered and in what context they suggested to apply the security patterns based on the responses of misuse cases.

They introduced three types of relationships *i.e1*) includes and extends relationship 2) maintenance relationship, and 3) threatens & mitigates relationship, for modeling requirements using the use cases and misuse cases. The use cases/ misuse cases which share the same steps are in include relationship whereas extends relationship exists in case of optional behaviors from high level case. Once the requirements are elaborated and relationships are specified the integration by using the mentioned types of relationships is modeled. However, the proposed methodology works only in the presence of automated tools. In addition, misuse case based approaches require a comprehensive list of all identified threats, misuse actors, sequence of actions and attack patterns [4][7][11].

2.4. Aspect Based

In order to enhance the software architecture for supporting aspectual components, they have considered an Aspectual hyper-layer which is affixed on the top of base architecture model. In order to map methods onto the base architectural elements an aspectual component must have some weaving capability. The weaver is usually described in an XML file which is separated from base architecture to make conventional description of architectural elements unchanged. The approach is supported by a case study of Content Management System in which it was analyzed how to transmit the documents on the network. They studied the working of integrity checks for preventive or detective mechanisms. Three security parameters like integrity, auditing and access control were checked. The main benefits of the approach is injecting security into architecture without disturbing the base architecture, higher degree of separation of concerns, using the existing ADLs instead of designing a new one [9].

2.5. Repository Based

Daniel Melladoa *et al* supported the iterative development of the secure components [15]. They used the process description patterns to support the integration of the security requirements into the architecture. The purpose of their research is to identify the systems that could be used by a little modification to reduce the development cost. They used the common criteria (CC) to take the guidelines but the CC method cannot provide the methodological support. They stepped through the requirements, developing the artifacts and identifying the dependencies among those artifacts to improve the repository. Ambrosio *et al* presented that security must be integrated at the requirement engineering level and if incorporated properly, less issues are faced at the later stages [14]. They supported an approach based on the repository that contains all the known threat types.

GuttormSindre, Donald G. Firesmithpresented a re-used based approach to determine security requirements that involves identifying the threats and relating these to the requirements [8]. Later they made a repository of the threats and requirements. The main benefit of the approach is that it maintains a knowledge base for the threats identified and improves the quality of work by the re-use process. Their framework focuses on the architecture level at requirement stage of development and more generic decisions are taken. They mentioned that there are two key-processes in re-use oriented development. First process is related to develop re-usable artifacts and second one is related to develop end user applications. In the proposed approach they addressed the two issues. Firstly which artifacts should be stored in repository and Secondly how to arrange in the repository. They worked on repository of reusable misuse cases to cover security and integration with the architecture for the rapid development. I order to elicit the security requirements they identified the assets, criticality level, threats to each asset, risks to each asset, and security goals for each asset. They proposed bottom up search, top-down search, and threat brainstorming techniques to find the threats from the repository.

3. Evaluation Criteria

The parameters selected for the proposed approach are re-usability, generalization, domain independence, application independence, threat leverage and knowledgebase. The description of theses parameter is given below and it is also depicted by Table 4.1.

3.1. Reusability

It means that the architectural features provided can be used for future. Aspect based approach provides this feature for architectures and achieves a higher degree of encapsulation. Repository based approaches contain the reusability feature.

3.2. Generalization

Generalization means that the architectural features can be used for multiple requirements from multiple domains. A generalized approach can be applied is best suitable for the components based development.

3.3. Domain Independence

An approach with domain independent is not restricted a particular domain to integrate the specific components in the architecture. It gives strength to be used in the multiple domains and makes the requirement capturing processes generalized.

Table 4.1. Possible Values of Approaches

Serial #	Parameters	Description	Possible Values
1	Re-Usability	The features provided by approach are usable for different scenarios.	Yes/No/Partial
2	Generalization	Architectural features can be used generally for multiple threats/misuse cases.	Yes/No/Partial
3	Domain Independence	Dependence of approach to a domain(s).	Yes/No/Partial
4	Threat Leverage	The capability to accommodate more threats and misuse cases.	Yes/No/Partial
5	Knowledgebase	Historical knowledge for architects to integrate the features.	Yes/No/Partial
6	Application Independence	Does the approach provide mitigation specific to application or not.	Yes/No/Partial

3.4. Application Independence

It provides the information that an approach is not restricted to provide certain application based features and can be applied on cross-platforms.

3.5. Threats Leverage

Leverage parameter is the capacity of the approach to provide the mitigation techniques for future threats/misuse cases.

3.6. Knowledgebase

This feature provides historical knowledge to the architects about features like repositories to integrate new mitigation techniques.

4. Analysis and Discussion

Based on the parameters defined (Table 4.1) a comparison has been made for the approaches mentioned in the literature in Table 4.2. It can be noted form the Table 4.2 all the other approaches lack in provisioning of some features. Eduardo B. Fernandez recommended using the patterns in the layers to increase the functionality [5]. Yijun Yu *et al* stated that using patterns early stages leads to early solution by providing the usability feature [20]. Goals based are not providing any of the parameters. Axel van Lamsweerde worked on application layer for the banking system by using the goal-based approach and stated that this approach deals specific to a domain or application [13].

Similarly, Qingfeng Annie *et al* worked on the goal-based approach for the role based access control systems for health care systems but their goal-based framework is also specific to the domain [12]. A goal question metric approach also takes the questions from specific domain to assign the metrics to the questions [2]. Misuse-case based provides generalization because threats identified helps in identification of security requirements that can be used for the multiple domains for wide range of applications [6][21]. The aspect-based provides all the features except the knowledge-based as it works as layers based approach in which an additional layer is affixed on the top of base layer and modification in this additional layer does not affect the base layer [9].

Table 4.2. Comparison of Existing Approaches

Parameters	Approaches				
	Pattern Based	Goal Based	Misuse Case Based	Aspect Based	Repository Based
Re Usability	Yes	No	No	Yes	Yes
Generalization	Yes	No	Yes	Yes	Yes
Domain Independence	Yes	No	No	Partial	Yes
Threat Leverage	No	No	No	Yes	No
Knowledgebase	Yes	No	No	No	Yes
Application Independence	Yes	No	No	Yes	Yes

The repository based features does not provide threat leverage. Daniel Mellado *et al* proposed the security resources repository for the re-use of security requirements for multiple applications and multiple domain. The development of the repository is step-wise procedure and it contains the patterns for the requirements. The newer threats cannot be handled by the repository as it only provides the mitigation techniques to the already stored threat profiles [15]. Ambrosio *et al* suggested the use of the requirements repository based approach for the generalization and re-usability feature. They proposed SIREN (Simple Reuse of Software Requirements) approach for the requirements re-usability [14]. The proposed approach provides all the features. The detailed discussion of the approaches is given in the next section.

5. Conclusion

This article presents a detailed and comprehensive survey with regard to the integration of security non-functional requirements into architectural design. The current approaches for dealing the requirements at architecture level work well for specific situations. The approaches analyzed and compared include Goal based, Patterns based, Use/Misuse case Based, Aspect Based and Repository based. The uses case are suitable for the simple requirements while the misuse case are suitable for handling the more complex scenarios by starting with the negative scenarios. However, the misuse cases cannot identify the misusers and sequence of actions so alone misuse approach is not desirable. This gap of the identification of users/agents and their actions is covered by the goal-based approach. However, the goal based approaches lacking the reusability feature which is provided by the repository based and pattern-based approaches. The goal-based approach is suggested to apply before the pattern-based approach as it has the dependency of goal-based approach. One of the drawbacks of the goal-based approach is that it requires resolving the constraints in advanced but for large and complex systems it is not desirable alone. The pattern based approach contains the knowledgebase and provide the easy understanding of the requirements and provide generalization. Aspect oriented approach does not provide a generic solution as of the pattern based because it covers only certain aspects or features in depth.

References

[1] B. Fabian, M. Heisel, T. Santen and H. Schmidt, “A Comparison Of Security Requirements Engineering Methods Special Issue”, Security Requirements Engineering, vol. 15, Issue 1, **(2010)**, pp. 7-40.

[2] M. Kassou and L. Kjiri, A Goal Question Metric Approach For Evaluating Security In A Service Oriented Architecture Context by IEEE 7th International Conference on Software Security and Reliability-Companion (SERE-C), **(2013)**.

[3] J.C. Huang R. Settimi and O.B. Khadra, “Goal-Centric Traceability For Managing Non-Functional Requirements 27th International Conference on Software Engineering”, ICSE Proceedings, **(2005)**.

[4] M.H. Diallo, J.R. Mariana, S.E. Sim, T.A. Alspaugh and D.J. Richardson, “A Comparative Evaluation of Three Approaches to Specifying Security Requirements”, Published in Proceedings of the Twelfth Working Conference on Requirements Engineering: Foundation for Software Quality, **(2006)**.

[5] F. Khan, S.R. Jan, M. Tahir, S. Khan and F. Ullah, “Survey: Dealing Non-Functional Requirements at Architecture Level”, VFAST Transactions on Software Engineering, ISSN(e): 2309-6519; ISSN(p): 2411-6327, **(2016)**.

[6] G. Sindre and A.L. Opdah, “Eliciting Security Requirements with Misuse Cases”, Requirements Engineering, vol. 10, Issue 1, **(2005)**, pp. 34-44.

[7] J.J. Pauli, “Refining Use/Misuse/Mitigation Use Cases for Security Requirements”, Journal of Software Engineering and Applications, vol. 7, **(2014)**, pp. 626-638.

[8] G. Sindre and D.G. Firesmith, “A Reuse-Based Approach to Determining Security Requirements by”, **(2004)**.

[9] M.G. Jaatun and P.H. Meland, “Security requirements for the Rest of us”, A Survey: Software IEEE, vol. 25, Issue 1, **(2008)**.

[10] T. Heyman and K.Y. Skout, “The Security Twin Peaks”, Proceedings of the Third International Conference on Engineering Secure Software And Systems, **(2011)**.

[11] F. Braz and E.B. Fernández, “Eliciting Security Requirements through Misuse Activities”, 19th International Workshop on Database and Expert Systems Application, DEXA, **(2008)**.

[12] A.I. Anton, “A Framework for Privacy-Enhanced Access Control Analysis in Requirements Engineering”, International Conference on Electronics and Communication Systems (ICECS'15), At Karpagam College of Engineering, Coimbatore - 641 032, vol. 2, **(2014)**.

[13] A.V. Lamsweerde, “Elaborating Security Requirements by Construction of Intentional Anti-Models”, Software Engineering, ICSE. Proceedings. 26th International Conference, **(2004)**.

[14] A. Toval, J. Nicolas and B. Moros, “FernandoGarcia; Requirements Reuse for Improving Information Systems Security”, A Practitioner's Approach, vol. 6, Issue 4, **(2002)**, pp. 205-219.

[15] D. Melladoa, E.F. Medina and M. Piattini, “A Common Criteria Based Security Requirements Engineering Process for the Development of Secure Information Systems by. Science direct, **(2007)**.

[16] E.A. Oladimeji, S. Supakkul and L. Chung; “Security Threat Modeling and Analysis: A Goal Oriented Approach”, Proceedings of 10th International Conference on software engineering and applications, **(2006)**.

[17] J. McDermott, “Using Abuse Case Models for Security Requirements Analysis”, Computer Security Applications Conference. (ACSAC '99), **(1999)**.

[18] T. Heyman, R. Scandariato, C. Hurgens and W. Joosen, “Using Security Patterns To Combine Security Metrics”, Third International Conference on Availability, Reliability and Security by, **(2008)**.

[19] M. Weiss, “Selecting Security Patterns that Fulfill Security Requirements”, International Requirements Engineering, 16th IEEE, **(2008)**.

[20] Y. Yu and Y. FeiXiong, “Enforcing A Security Pattern in Stakeholder Goal Models published in QoP”, Proceedings of the 4th ACM workshop on Quality of protection, **(2008)**, pp. 9-14.

[21] R. Matulevicius, N. Mayer and P. Heymans, “Alignment of Misuse Cases with Security Risk Management”, Third International Conference on Availability, Reliability and Security, **(2008)**.

[22] G. Elahi, E. Yu and N. Zannone, “A Vulnerability-Centric Requirement Engineering Framework: Analyzing Security Attacks”, Countermeasures, and Requirements Based on Vulnerabilities, Requirements Engineering, vol. 15, no. 1, **(2010)**, pp. 41-62.

[23] S. Myagmar, A.J. Lee and W. Yurcik, “Threat Modeling as a Basis for Security Requirements”, Published in: Symposium on Requirements Engineering for Information Security (SREIS), **(2005)**.

[24] T. Okubo, N. Yoshioka and H. Kaiya, “Security Driven Requirements Refinement And Exploration Of Architecture With Multiple NFR Points Of View”, Software Engineering Advances, **(2014)**.

[25] K. Edge, R. Raines, R. Bennington and C. Reuter, “The Use of Attack and Protection Trees to Analyze Security for an Online Banking System”, Published in: Proceedings of the 40th Hawaii International Conference on System Sciences, **(2007)**.

[26] F.Z. Hamzani, “Survey of Non-Functional Requirements Modeling and Verification of Software Product Lines by”, IEEE Eighth International Conference on Research Challenges in Information Science (RCIS), **(2014)**.