Journal of Research and Development

Aruna, J Res Development 2017, 5:1
DOI: 10.4172/2311-3278.1000152

Review Article Open Access

Survey: Classification and Validation of Security Patterns in SDL

Aruna ER*

Department of IT, Vardhaman College of Engineering

Abstract

In software engineering, expert guidance is encapsulated in the form of security design patterns which provide reusable solutions to recurring security specific problems. More security design patterns catalogs are available and the security pattern community has produced significant contributions with these patterns, many of them are applicable to design phase. We believe it is better to explain how to use a proven methodology (Security design patterns) to design security architectures that accurately fit the needs of application design, rather than depends on a range of fixed architectures. We propose secure aware software development life cycle (saSDLC) by using the security patterns as reusable component during SDLC.

Keywords: Secure aware; Software development life cycle; Nonfunctional requirements; Security design patterns

Introduction

The Security is the major concern in all software product implementations in recent years. "Security concerns must be concentrated on every phase of software development life cycle" stated by Devanbu et al. [1]. It is quite difficult to do so because all the software development teams are not security experts. Security patterns contain the expert knowledge about the security concerns which will guide the software engineers in the right direction. Security patterns are proven solutions for the security problems, hence the software engineers achieves good results with nominal efforts. These security patterns are organizing as the patterns catalogs, by the time of use an engineer can select the suitable pattern and make use for the problem.

Design pattern, "describes a problem which occurs again and again in our environment and describes the core of the solution to that problem, in such a way that we can use this solution a million times over, without ever doing it the same way twice "defined by Gang of Four (GoF) authors. As these are object oriented reusable patterns, catalogs are organizing in two ways based on the purpose or scope. With respect to purpose patterns categories are creational, structural, and behavioral type patterns and based on scope object patterns and class patterns [2].

This paper is organized as Back ground of the security design patterns in section II, survey on various classifications of security patterns in section III, Representation and Validation of security patterns in section IV, and conclusion and future work in section V.

History of the Security Patterns

Christopher Alexander was introduced the concept of design pattern in 1997 [3]. Then these are spanned as object oriented reusable design patterns by the GoF [2]. Yoder and Barcalow were published the first security pattern in 1997. As the evolving of more patterns, the patterns are categorized into 13 procedural and 13 structural patterns and 3 mini structural patterns by Darrell Kienzle et al. [4].

Security patterns are treated as available system patterns and protected system patterns by Blakley B et al. [5]. Security patterns (cyber patterns and attack patterns) guide us to move from offhand skill to engineering discipline because they transfer knowledge about proven solutions in an understandable and reusable format to experienced users by Trowbridge et al. [6].

Many security concerns, security taxonomy with related patterns and their characteristics, integrating security into system defined by Markus Schumacher et al. [7]. A pattern language was defined to select all the security patterns easily by Munawar Hafiz et al. [8]. Nine privacy patterns and their relationships, detailed descriptions were given by Hafiz et al. [9].

Fernandez et al. proposed the two new classification schemes first is based on literature survey from 1997 to 2012 and the second is pattern recognition in security aspects in various domains [10]. Subsequently, more and more patterns, pattern catalogs were emerged as per security issues. Dougherty et al. reported secure design patterns are descriptions, describing a general solution to a security problem that can be applied in many different situations. Rather than focus on the implementation of specific security mechanisms, these patterns are used to eliminate the accidental insertion of vulnerabilities into code or to mitigate the consequences of vulnerabilities [11]. Eduardo Fernandez et al. examined the structure and purpose of security patterns [12].

Many security patterns are emerged based on their domain; currently more than 400 security patterns are available. These enormous number of patterns needs to be classified. Many researchers proposed the various classification techniques based on domain specific patterns, application specific patterns, design level patterns, implementation level patterns etc. [13,14]. All of them used the pattern template most similar to design patterns for describing the security patterns.

Security Patterns Classifications

Some of the classifications reported by various practitioners

1. Hafiz et al. analyzed various classification schemes for security patterns and proposed a classification scheme that uses the threat model and application context to partition patterns [15], Hafiz et al. reported classification based on the security objective, but not as domain specific [16].

***Corresponding author:** Aruna ER, Department of IT, Vardhaman College of Engineering, Hyderabad, Tel: 9440015489; E-mail: ayam.ac2003@gmail.com.

Received January 13, 2017; **Accepted** February 25, 2017; **Published** March 05, 2017

Citation: Aruna ER (2017) Survey: Classification and Validation of Security Patterns in SDL. J Res Development 5: 152.

2. Pattern Diagrams are used to classify the patterns by Fernandez et al. [17].

3. Nobukazu et al. are categorized the security patterns are in the form of software life cycle point. Security patterns for requirement phase, design phase and implementation phase. For requirement phase security patterns are Analysis process patterns and model based patterns. The design phase patterns are pattern for specific security concern and domain specific patterns. Implementation phase patterns are secure programming guidelines, attack patterns, secure refactoring [18].

4. The well-known existing classification method of GoF classification is re-used by Konrad et al. for security patterns as creational, structural and behavioral with additional level such as purpose of each pattern [19].

5. Schumacher's security patterns classification is based on Zachman's framework for enterprise architecture [20,21]. They have classified in two dimensions, first dimension is on "what", "how", "where", "who", "when", and "why" interrogatives. The second dimension is different information model views such as business model or technology model and Sahumacher et al. added new column as security view.

6. Steel et al. classify the JEE patterns as n-tire based such as web, business and web service based [22].

7. Security patterns are differentiated as architecture patterns and design patterns categories by Rosado et al. [23].

8. Swiderski et al. focused on well know security acronym such as CIA (Confidentiality, Integrity and Availability) and STRIDE (Spoofing, Tampering, Repudiation, Information disclosure, denial of service, and elevation of privilege) with a total 14 patterns they have classified [24,25].

9. VanHilst et al. classified multi-dimensional matrix which addresses the problem coverage and pattern classification. The list of security concerns represented in each matrix. In this the first dimension is life cycle activities, second dimension is security patterns source [26].

10. Fernandez et al. states the architectural security patterns which are classified into two types. One type of classification is hierarchy of layers and other type is based on the relationships among the patterns. The relations among the patterns are identified using an automatic relationship extraction and analysis. This classification covers very less number of security patterns [27].

11. VanHilst et al. work was enhanced by Washizaki et al. by adding more dimensions for the security pattern classification. They are life cycle stage, Architecture phase, domain concern, pattern types and constraints, relations association, generalization, and aggregation [28].

12. Koen Yskout et al. classified the security patterns based on the annotations; this covers the four dimensions, based on security objectives, applicability, trade-off labels and relationship among patterns [29].

13. Poonam pande et al. classified the security patterns using Hierarchical Cluster Analysis; they performed cluster analysis around 211 patterns. They have identified the identical patterns and removed the redundant patterns using maximum' distance metric [30].

Representation and Validation of Security Patterns during Software Development Life Cycle

Few researchers stated how to validate the security patterns during the design level.

1. Abramove et al. have suggested the stereotype for validating the patterns during design phase [31] they have designed the data base application.

In Figure 1, for protecting the object while accessing by other object they are validation of security concern using the security pattern. This is only with in design level, further it is not confirming how it got evolved in the implementation phase to resolve the vulnerabilities.

2. Model checking methodology is used for verifying the composition of security patterns by Dong et al. [32].

In Figure 2, The Authentication Enforcer pattern encapsulates the authentication logic. Here the two clients are authenticating each other by the Authentication Enforcer class. Using credentials the Authentication Enforcer class authenticates two clients and represents the authenticated user. Here also the security patterns verified at the model level, it does not guarantee that threats and vulnerabilities are resolved in the corresponding implementation code.

3. Hamid et al. proposed a method which is used to validate the application of patterns at the model level by UML They have applied the GoF mediator pattern and UML class diagram to represent air traffic controller application at design level [33] but this is also not applicable to implementation level.

4. Jan Jurjens presented extension UML allows to represent security related information with in the design diagrams.

In Figure 3, the instances which are critical, those are representing in the stereo type, this instances are supposed to be protected [34].

5. Denis Hatebur et al. proposed a pattern system for security requirements engineering, this is based on security problem frames. This system produce complete description of all the sub problems solution and this will solve a complex security problem using g Jackson's problem frames [35].

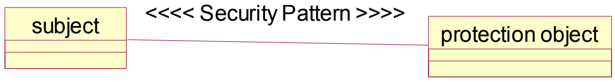

Figure 1: Security pattern.

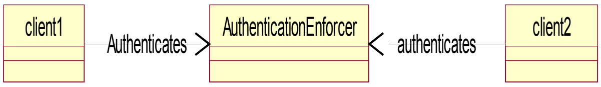

Figure 2: Authentication enforcer pattern.

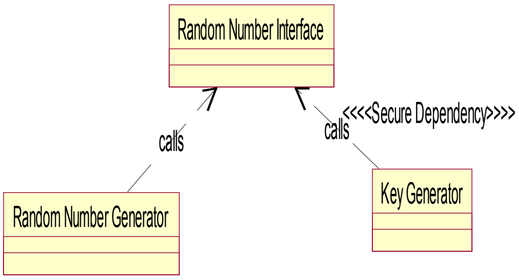

Figure 3: Extension UML.

6. Lodderstedt et al. designing access control policies for model driven software development, this is based on Role Based Access Control (RBAC) and this model ensures the security. This model introduces the authorization constraints for granting privileges to perform operations [36] (Figure 4).

In Figure 4, The RBAC concepts are represented as Meta models. Permission is a relative object between role and model element, here every UML model element is to take the role of protected resource.

The above two models are supporting to security concerns in modeling but they cannot be directly used at implementation level to know the correctness of the security patterns.

Fernandez et al. proposed three patterns corresponding to most common models for security i.e., Authorization, RBAC, and Multilevel Security. These security concerns can be applied to all the levels of SDLC [37].

8. Haralambos Mouratidis et al. proposed a methodology that considers security as an integral part of the whole system development process, explained through health and social care information system as a case study. This process is characterized with five key ideas considering security issues in overall development process, identify security requirements at early stage and continues until implementation stage, secondly, using Tropos hierarchical approach, third, iteration of redefinition of security requirements, fourth considering the organization security policy , fifth functional and nonfunctional requirements are defining together with clear distinction (Figure 5).

In Figure 5 designing of above nonfunctional requirements including the design of the client server application [38]. They have concluded this process yet to be refined with different case studies and extensions has to be made for Formal Tropos specification language.

9. Golnaz Elahi et al. proposed the extension to the i* framework for security trade-off analysis using multi-agent and goal orientation. But these models become more complex and inefficient for goal models scale and also proposed process is knowledge of security mechanisms and corresponding contributions are not existing, their work is concluded with their approach is validated with empirical studies [39].

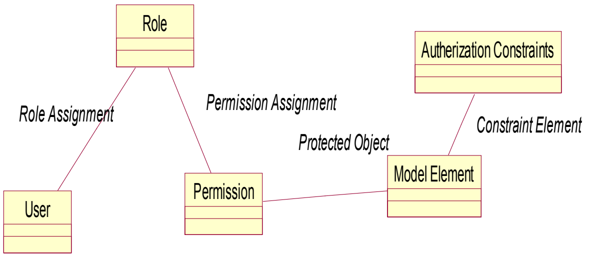

Figure 4: RBAC concepts.

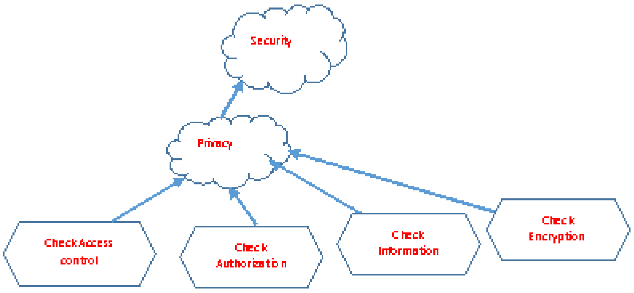

Figure 5: Designing of non-functional requirements.

10. Masatoshi Yoshizawa et al. perform a test template to perform validation of security pattern and supported for pattern implementation. Their approach is tested for finding implementation defects and corrects the defects [39].

Discussion and Conclusion

Most of the security pattern classification done on few patterns; these classifications are not covering all the security issues. There is no universally accepted catalog for all the security objectives. Most of the researches classified the pattern depends on the application specific, domain specific. Some researchers classified few patterns based on security objectives. There is no specific process to select the security pattern from the catalog. Some researchers concentrating on security requirements and patterns right from the inception, but there is no guarantee that identified patterns got implemented at the ending of the SDLC? Most of the researchers proposed approaches to represent the security patterns during design, but they are failed to guarantee to validate the same patterns in the implementation phase. The Security pattern template differs from one to other in certain aspects. Very few researches reported that how to use the combination of the patterns to resolve more problems.

Most of the patterns having different names with same purpose due to abstract description of the patterns. Many of the patterns serve for two or three security concerns. The presentation of security patterns during design phase does not have specific model, many of them proposed stereotypes to represent the security concerns. Very few practitioners design and validate the security patterns from design phase to implementation phase. Few researchers concentrated on the impact of security patterns on the coding and the productivity. Most of the research on security patterns is empirical study.

In future scope, the global security pattern catalog has to be emerged with all the dimensions that must be applicable for all the security concerns in the presence of all other different requirements irrespective of platform, domain or application specific. Security concerns and patterns must be identified during the inception, and the same must be representation in the design phase, must be implemented same in the construction phase and the outcome must be test with same requirement during transition phase. By making, this as possible in future, we can include the most critical nonfunctional requirement in SDLC this will leads to achieve secure aware SDLC.

References

1. Devanbu PT, Stubblebine S (2000) Software engineering for security: A road map. Conference proceeding on the future of software engineering, pp: 227-239.
2. Gamma E, Helm R, Johnson R, Vlissides J (1995) Design patterns –Elements of reusable object-oriented software. Addison-Wesley, US.
3. Alexander C, Ishikawa S, Silverstein M (1977) A pattern language: Towns, buildings, construction. Oxford University Press, New York.
4. Kienzle DM, Elder MC, Tyree D, Edwards- Hewitt J (2001) Security patterns repository version 1.0.
5. Blakley B, Heath C (2004) Security design patterns technical guide - version 1. Open Group.
6. Trowbridge D, Cunningham W, Evans M, Brader L, Slater P (2004) Describing the enterprise architectural space.
7. Schumacher M, Fernandez EB, Hybertson D, Buschmann F (2005) Security

patterns: Integrating security and systems engineering. John Wiley & Sons, US.

8. Munawar H, Paul A, Ralph J (2011) Growing a pattern language (for security).
9. Hafiz M (2006) A collection of privacy design patterns. Proceedings of the 13th Conference on Patterns Language of Programming.
10. Fernandez EB, Yoshioka N, Washizaki H (2007) Using security patterns to build secure systems. Information Processing Society of Japan, pp: 47-48.
11. Dougherty C, Sayre K, Seacord RC, Svoboda D, Togashi K (2009) Secure design patterns.
12. Fernandez-Buglioni E (2013) Security patterns in practice: Designing secure architectures using software patterns. John Wiley & Sons, US.
13. Hafiz M, Adamczyk P, Johnson RE (2007) Towards an organization of security patterns. IEEE Software 24: 52-60.
14. Fernandez EB, Washizaki H, Yoshioka N, Kubo A, Fukazawa Y (2008) Classifying security patterns. Proceedings of the 10th Asia-Pacific Web Conference.
15. Bunke M, Koschke R, Sohr K (2012) Organizing security patterns related to security and pattern recognition requirements. International Journal on Advances in Security 5: 1-2.
16. Hafiz M, Adamczyk P, Johnson RE (2012) Growing a pattern language (for security). ACM international symposium on new ideas, new paradigms, and reflections on programming and software, pp: 139-158.
17. Dangler J (2013) Categorization of security design patterns. Electronic Thesis and Dissertations Paper.
18. Zachman J (1987) A framework for information systems architecture. IBM Systems Journal 26: 3.
19. Nobukazu Y, Washizaki H, Mruyama K (2008)l A survey on security patterns. Natinal Institute of Informatics 5: 35-47.
20. Konrad S, Cheng BH, Campbell LA, Wassermann R (2003) Using security patterns to model and analyze security requirements. International Workshop on Requirements for High Assurance Systems, pp: 13-22.
21. No Authors Listed (2012) The zachmann framework for enterprise architecture.
22. Steel C, Nagappan R, Lai R (2005) Core security patterns: Best practices and strategies for J2EE(TM), web services, and identity management. Prentice Hall International, New Jersey.
23. Rosado DG, Gutierrez C, Fernandez-Medina E, Piattini M (2006) Security patterns related to security requirements. International Workshop on Security in Information Systems, pp: 163-173.
24. No Authors Listed (1991) Commission of European Communities, Information technology security evaluation criteria.
25. Swiderski F, Snyder W (2004) Threat modeling. Microsoft Press.
26. VanHilst M, Fernandez EB, Braz FA (2008) A multidimensional classification for users of security patterns. International Workshop on Security in Information Systems, pp: 89-98.
27. Washizaki H, Fernandez EB, Maruyama K, KuboA, Yoshioka N (2009) Improving the classification of security patterns. Database and Expert Systems Applications, pp: 165-170.
28. Yskout K (2013) Connecting security requirements and software architecture with patterns.
29. Ponde P, Shirwaikar S, Gore S (2016) Hierarchical cluster analysis on security design patterns. ACM, AICTC.
30. Abramov J, Shoval P, Sturm A (2009) Validating and implementing security patterns for database applications. Proceedings of the 3rd International Workshop on Software Patterns and Quality (SPAQu).
31. Dong J, Peng T, Zhao Y (2010) Automated verification of security pattern compositions. J Inf Softw Technol 25: 274-295.
32. Hamid B, Percebois C, Gouteux D (2012) Methodology for integration of patterns with validation purpose. European Conference on Pattern Language of Programs (EuroPLoP), Irsee, Germany.
33. Jürjens J (2005) Secure systems development with UML.
34. Lodderstedt T, Basin DA, Doser J (2002) SecureUML: A UML-based modeling language for model-driven security. Proceedings of the 5th International Conference on the Unified Modeling Language (UML), Dresden, Germany. pp: 426-441.
35. Eduardo BF, Rouyi P (2001) A pattern language for security models.
36. Haralambos M, Paolo G, Gordon M (2003) Integrating security and systems engineering: Towards the modelling of secure information systems.
37. Golnaz E, Eric Y (2007) A goal oriented approach for modeling and analyzing security trade-offs. 26th International Conference on Conceptual Modeling, pp: 5-9.
38. Masatoshi Y, Hironori W, Yoshiaki F, Takao O, Haruhiko K, et al. (2016) Implementation support of security design patterns using test templates. Information 7: 34.
39. Denis H, Maritta H, Holger S (2014) A pattern system for security requirements engineering. EuroPLoP '14 Proceedings of the 19th European Conference on Pattern Languages of Programs, ACM.