International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 4 Issue 11, November-2015

Design of Non-Functional Requirement(Security) using Security Design Patterns in Architecture Phase to Develop Secure SDLC

E. R. Aruna
Department of Information Technology
Vardhaman College of Engineering
Hyderabad, Telanagana, India

A. Rama Mohan Reddy
Professor of CSE
SVUCE, Tirupathi
Andhra Pradesh, India

K. V. N Sunitha
Principal,
BVRIT Hyderabad
India

***Abstract*— Software Development life cycle is the concrete process to develop software product, but it is still to be refined to handle Non-functional requirements at all stages of the life cycle of the product development. To develop 100% successful product, there must be equal importance between functional requirements and Non functional requirements. Some of those NFR's are Performance, Reliability, Security, Operability, Modifiability etc. The design decisions are resolved at early stages if it is possible to encompass all non functional requirements along with the functional requirements in all phases of SDLC. This paper focuses on importance of NFR (security) design in architecture phase of SDLC and proposes a methodology to include security feature using security design patterns. As a case study we have illustrated this methodology to iLocking system Application. We believe "including security design patterns in Architecture is resistant to insecure features". Our approach is most suitable for the security critical applications; such applications can be run on any device irrespective of mobile platform or fixed platform with high security. With this we will achieve secure SDLC at one phase i.e. Architecture phase.**

Keywords: secure SDLC, NFR, security design patterns, iLocking system

I. INTRODUCTION

The focus of my paper is "Reducing reengineering cost by designing the application through ingesting non functional requirements termed as quality attributes within the SDLC right from its beginning". Software Development Team must recognize software security as an intrinsic part of the software development process, not a reconsideration which requires additional cost and time.

Most of the software development methodologies improvement is towards to speed up the product outmoded to market by delivering software Functional Requirements [6]. Many industry people support that the above methodologies are suffering from inability to express Non-functional Requirements (NFRs) explicitly, as primary artifacts in SDLC phases [8]. In traditional process they have been treating the non functional requirements as secondary artifacts [2] [4] [9] [18]. There is a serious limitation in emphasizing the project schedules pertaining to technical and non-functional requirements of the product to meet the product dead line. M. Farid and J. Mitropoulos proposed visualization framework that is used to schedule software non-functional requirements implementations by utilizing agile project management agile techniques [19]. Non-functional requirements must be treated as Primary or Specific requirements [20] in SDLC.

The Security is the major concern in all software product implementations in recent years. In addition to the government agencies and Industry, the individual people also using the software products in public as well as private domains. This enormous usage of the software product systems has resulted software products need to be maintain and manage large amount of critical information. It is significant to safeguard the software products which are developed as per the customer's needs commonly called as functional requirements. It is coequally important to assure that these systems are safeguarded [10].

Most of the software practitioners recognized it is better to design security from scrape. It is prominent to use security models at early stages of SDLC for improving the software product security. Ivan Victor krsul [7] describes best practices and how to apply those practices to produce security artifacts of software product during development. Many researchers agreed to include security features within the software development life cycle from the beginning can improve overall software security.

Many of the researchers uses Misuse case analysis to find the security threat analysis. Misuse case analysis creates interactions among application and attackers. With this interactions the researchers can find the attackers who crosses security boundaries [11][12]. The application must protect itself from security threats during applying misuse cases [15]. Researchers build misuse cases at initial phases of SDLC, meanwhile they should care from misuse cases not to access data, interfaces, and methods of the actual product. Software development team must use security technology to protect and remove the vulnerabilities [17] while using misuse cases to deal security concerns. In our proposed research we are working with the alternative model to design the product by including security features, patterns. In our approach we are

not dealing with the misuse cases but we are appending the security feature as security design patterns in the design phase.

This paper is organized as related work in section 2, significance of considering non functional requirements in SDLC architecture phase in section 3, design of case study using security design patterns in section 4 and finally concluded with future work in section 5.

2. Related work

Many approaches are there to design and develop the product using secure SDLC. Some of them are used security design patterns for analyzing the potential attacks [26]. Some approaches treated Security design patterns as a tool for improving product security in Architecture phase [16]. The insistence of security in software product at the Architecture phase can reduce the high cost and effort in continuation with coding phase [27].

Halkidis, Nikolaos, Alexander Chatzigeorgiou, George Stephanides proposed a method Architecture risk analysis of software systems using security patters using mathematical model based on the fuzzy set theory and fuzzy fault trees by using class diagram of security critical application[29].

Our approach, we are design the security issues whose security is utmost importance in the architecture phase. Here we are representing security features as patterns, exceptions, techniques and process depends on the applications. With this, the application is properly communicated about security concerns among the development team right from the beginning stages of development.

3. Significance of inclusion of Non Functional Requirement in Architecture Phase

Architecture Phase is the most critical of the SDLC four phases [5]. Overall structure of the software is defined in Architecture phase of SDLC. Identification of essential components, whose security is uttermost serious, must be identified. Security architecture and design guidelines are included in this phase. Architects achieve complete system quality with optimum balance of system characteristics by means of active communication during entire development process. Architects as stakeholders involves directly or indirectly for quality balance in construction Phase. Hence Architecture people are responsible for overall system qualities [3].
Architecture is a representative for the system qualities to retain acceptable harmony between the functional and non-functional requirements during product development. Architect people emphasize entire structure of software product; achieve predictable results through diagrams and descriptions for the communication with developers[13].

As an existing problem point of view there is a discrepancy in the definition of non-functional requirements i.e illustration problems raises whether NFR must be treated as a function or quality or a constraint in the SDLC. Once it is accustomed then a representation problem arises at various stages of the life cycle. In the literature there are different views on functional and non-functional requirements, still there is in need to develop consensus model which facilitates how non functional requirements are elicited, represented, integrated and tested in the software development process.

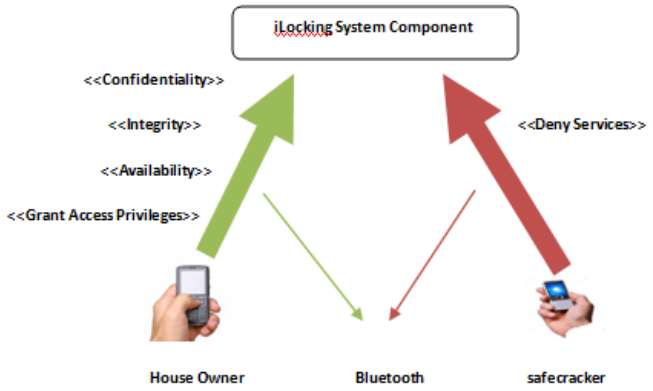
In design phase, security policies can be easily incorporated by means of secure and insecure states. If the system enters any unauthorized state, security system starts authorizing the state and makes never let it enters into unauthorized state [23].

In our approach we are using UML tools and object oriented design patterns together to represent secure software architecture. Unified modeling language is the universal standard language for software product design. UML is the most accepted tool to present proper objects, classes, interfaces, inheritance and establish key relationships among them [1]. Design Patterns can be used to specific problem, they are reusable, provides flexible design, addresses future problems and avoid redesign [21]. Multifold security principles are manifested in security design patterns [24].Security patterns and Analysis patterns can be used to build secure systems and conceptual models [5] [22] [25].

4. iLoacking System design using security patterns

To model Non functional requirement (security) in architecture phase we used the application “Secure door locking system using Bluetooth technology (ilocking System)” [14]. This system and its iLock App are wirelessly linked together using Bluetooth technology. This Application provides secure keyless access depends upon key generation and key matching. In this application, House owner is locking or unlocking the door using iLock App from his mobile. The iLock System device is fixed to the door. From his registered mobile the owner has to send request for pairing with iLock device at the time of locking or unlocking door. The iLock app will generate key, key has to be entered in the iLock device and gets authenticated with matching key, then the door gets locked/unlocked. If safecracker tries this, the iLock component will deny the services and issue alert to owner because the key generation is done not with registered mobile number.

We are illustrating this application using use case diagram, activity diagram, sequence diagram and state chart diagrams. We are modeling security concerns in the architecture with security patterns, security principles (integrity, confidentiality, Non-reputability). We have chosen this application because the security is most critical and is useful for enormous users.



Secure Door Locking System Using Blutooth

The following is the design of iLocking system using security design patterns:

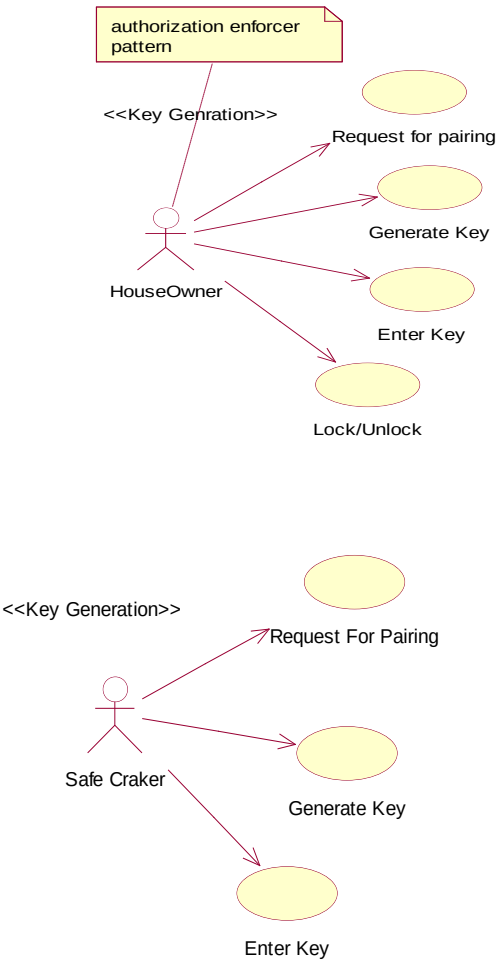


Fig.1Use case Diagrams

In Fig.1 We introduce the concept of NFR by means of security in this use case diagram, representing with security principal confidentiality in terms of key generation. We are also representing the authorization enforcer pattern used to manage and delegate authorization process. This security pattern enforces authorization constraints in every situation of designing. House owner requests for pairing with iLock device by generating a key. The secret key generation process must dependent on security pattern.

In fig.2 here we are achieving Integrity through key mapping process. The key mapping must be based on Authentication enforcer pattern and Intercepting validator pattern. The authentication enforcer pattern illustrates how a client should authenticate with the security application. Intercepting validator pattern validate input from client for security purpose. Based on the key mapping, iLock device gives privileges for locking/unlocking the door or simply deny privileges.

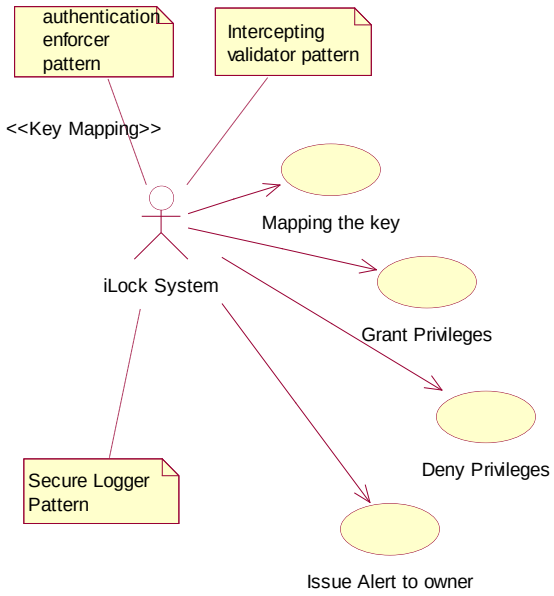


Fig.2 iLock System Use Case Diagram

We use activity diagram to show activities flow one after other. In this design representation we are achieving confidentiality, integrity and Non-repudiation by performing key generation and mapping. In addition to authentication and authorization enforce patterns; we can take other security patterns into consideration for performing security activities in this application.

There is a standard approach of selecting required pattern from the patterns catalog is available developed by Weiss M [28]. Once the Architect people and Developers knows about which type of security feature required like confidentiality, integrity, availability they can enter into the search engine of the repository catalog available, the search engine search and indexes them according the need of NFR.

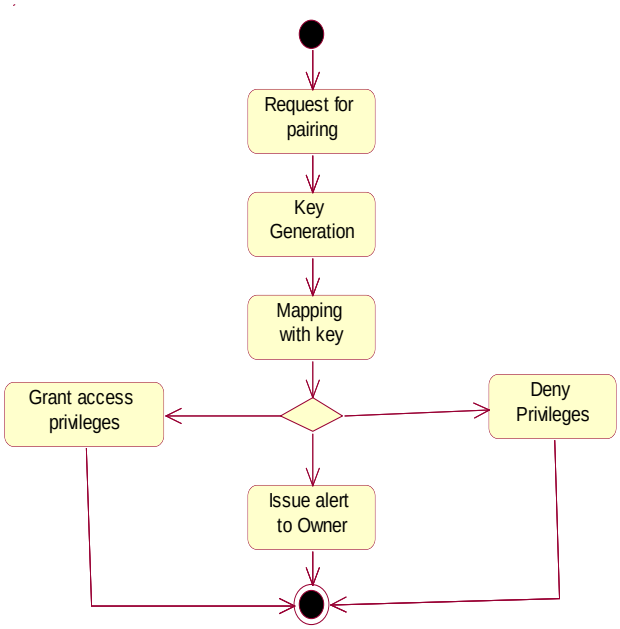


Fig.3 Activity Diagram

We use sequence diagram to show object interactions in timely manner. Here we can precise security protocols in the form of representing with message sequences. Here the objects owner, iLock system and safecracker exchanges cryptographic operations in the form of key generation and mapping. These two use cases must be developed using suitable proven security patterns to achieve confidentiality, integrity which mentioned in figure.1.

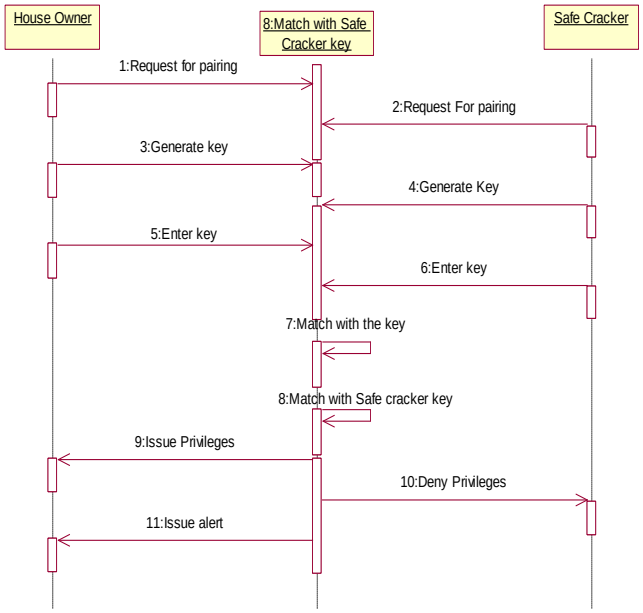


Fig.4 Sequence Diagram

We use statechart diagram to specify various states of a component. The states are specific to either component or system but not both. This diagram shows dynamic behavior and state of an in-dividable object may change with respect to events generated. This kind of representations shows the security context in terms of secure and insecure states. If the system gets into insecure state, we must protect the system by concentrating on lacking security features by selecting appropriate design patterns.

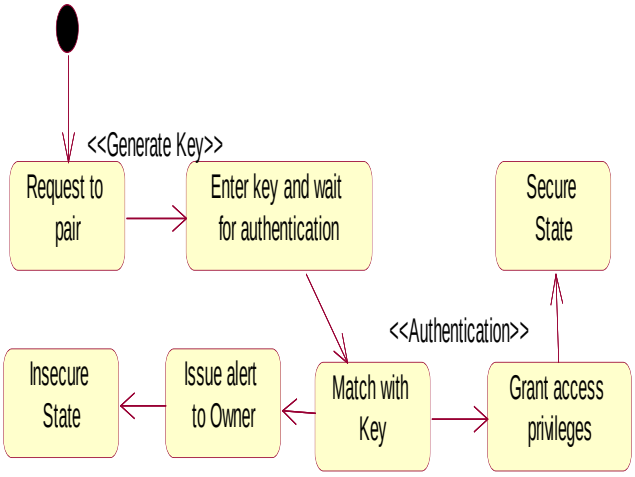


Fig.5 State chart Diagram

5. Conclusion and Future Work

In this paper, we have presented architecture with security concerns using UML, security principles and security patterns. Our design methodology proposed ‘security features must be introduced in all phases of SDLC right from the beginning’.
The combination of security patterns, principles and concrete UML design methodology improves software quality and development productivity.

In the Architecture phase, secured design can be extracted if most appropriate security patterns are selected and applied specific to problem. There is still required to develop universally accepted design practice to include a collection of non functional requirements in SDLC. So that we can reduce over head effort, cost and time in software development process. In our extension work, we are comparing the performance of our method of design with already implemented software products and also we would like calculate the performance of each security pattern and combination of patterns which are used to represent as a security feature in the Design.

REFERENCES

[1] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerland, and M. Stal., Pattern oriented software architecture, Wiley 1996.

[2] N. Mead, V. Viswanathan, and D. Padmanabhan, "Requirements Engineering into the Dynamic Systems Development Method", Proc. of the 2008 32nd Annual IEEE International Computer Software and Applications Conference, *2008*, pp. 949-954.

[3] Bass, L., Clements, P., Kazman, R., 2003, Software Architecture in Practice, Addison-Wesley

[4] J. Araujo and J. Ribeiro, "Towards an Aspect-Oriented Agile Requirements Approach", In proceedings of the Eighth International Workshop on Principles of Software Evolution (IWPSE'05), 2005, pp. 140-143.

[5] "Software Project Management", A Unified Frame Work by walker Royce

[6] M. Qasaimeh, H. Mehrfard, and A. Hamou-Lhadj, "Comparing Agile Software Processes Based on the Software Development Project Requirements", In Proceedings of 2008 International Conferences on Computational Intelligence for Modeling, Control and Automation; Intelligent Agents, Web Technologies and Internet Commerce; and Innovation in Software Engineering, 2008, pp. 49-54.

[7] I. V. Krsul, "Computer vulnerability analysis", PhD thesis, Purdue University, 1998

[8] A. Marcal, F. Furtado Soares, and A. Belchior, "Mapping CMMI Project Management Process Areas to SCRUM Practices", In 31st IEEE Software Engineering Workshop (SEW 2007), 2007.

[9] F. Paetsch, A. Eberlein, and F. Maurer, "Requirements Engineering and Agile Software Development", In IEEE Twelfth International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003, pp. 308.

[10] CERT Coordination centre, (2003), Annual Report,www.cert.org

[11] I. Alexander, "Misuse cases: Use cases with hostile intent", IEEE Software 20 , 2003, pp. 58-66.

[12] McDermott, C. Fox, "Using abuse case models for security requirements analysis", Proc. Annual Computer Security Applications Conference (ACSAC'99), 1999.

[13] Roland Faber "Architects as Service Providers" IEEE Software, Volume:PP , Issue: 99

[14] http://www.locksmithledger.com/product/12101366/alarm-lock-systems-inc-architech-with-bluetooth-le-ilock-app

[15] D. Firesmith, "Security Use Case," Journal of Object Technoly (JOT), Vol. 2, No.3, May/Jun 2003, pp.53-64.

[16] M-A. Laverdi`ere, A. Mourad, A. Hanna, M. Debbabi, "Security Design Patterns: Survey and Evaluation", IEEE CCECE/CCGEI, Ottawa, May 2006

[17] Petersen, G. , Steven, J Building Security in "Defining Misuse within the Development Process" Security & Privacy, IEEE , Volume:4 , Issue: 6

[18] R. Pressman, Software Engineering, A Practitioner"s Approach, New York, NY, McGraw-Hill, 2005.

[19] Weam M. Farid, Frank J. Mitropoulos " Visualization and Scheduling of Non-functional Requirements for Agile Processes" , in Southeastcon, 2013 Proceedings of IEEE ,pp.1-8.

[20] Rubey, R. J., and R. D. Hartwick, Quantitative Measurement of Program Quality, in Proceedings of ACM National Conference, pp. 671–677, 1968.

[21] E. Gamma, R. Helm,R. Johnson, and J. Vlissides, Design patterns – Elements of reusable object-oriented software, Addison-Wesley 1995.

[22] E.B.Fernandez, "Layers and non-functional patterns", Procs of ChiliPLoP, 2003. Phoenix,March 10-15, 2003.http://hillside.net/chiliplop/2003/

[23] Gaurav Raj, Dr. Dheerendra Singh, Dr. Abhay Bansal, "Analysis for Security Implementation in SDLC" 2014 IEEE,2014, pp.221-226.

[24] J.H.Saltzer and M.D.Schroeder, "The protection of information in computer systems", Procs. of the IEEE, Vol. 63, No 9, 1975, 1278-1308.
Awebversionisin:http://web.mit.edu/Saltzer/www/publications/protection/index.html

[25] E.B. Fernandez and X. Yuan, "Semantic analysis patterns", Procs. of 19th Int. Conf. on Conceptual Modeling, ER2000, 183-195. Also available from: http://www.cse.fau.edu/~ed/SAPpaper2.pdf

[26] Eduardo B. Fernandez, Maria M. Larrondo-Petrie, "Designing Secure SCADA Systems Using Security Patterns", Proceedings of the 43rd Hawaii International Conference on System Sciences – 2010.

[27] Spyros T. Halkidis,Nikolaos Tsantalis,Alexander Chatzigeorgiou, George Stephanides, "Architectural risk analysis of software systems based on securitypatterns", ieee transactions on dependable and secure computing, vol.5,no.3,july-september2008.

[28] Weiss. M, Mouratidis, H, "Selecting Security Patterns that Fulfill Security Requirements" International Requirements Engineering, 2008. RE'08.16thIEEE.

[29]. Spyros T. Halkidis, Nikolaos Tsantalis, Alexander Chatzigeorgiou, George Stephanides,"Architectural Risk Analysis of Software SystemsBased on Security Patterns", IEEE transactions on dependable and secure computing volume 5, No.3 July-Sept 2008.