DESIGN OF PATIENT MONITORING SYSTEM (PMS) APPLICATION USING SECURITY DESIGN PATTERNS IN ARCHITECTURE PHASE OF SECURE SDLC

Ms. E.R. Aruna¹, Dr. A. Rama Mohan Reddy², Dr. K.V. N. Sunitha³
¹Associate Professor, Department of IT, Vardhaman College of Engineering
²Professor of CSE, SVU College of Engineering
³Principal and Professor of CSE, BVRT for Women

Abstract— A successful Software product is achieved through fulfillment of all the functional Requirements and nonfunctional Requirement (NFR)s. In SDLC the design decisions must be resolved by encompassing the functional requirement along with the nonfunctional requirements at early stages to avoid the problems in further evolving stages. This paper focuses on representation of NFR (security) in architecture phase of Software Development Life Cycle (SDLC) and proposes a methodology to include security feature using security design patterns. We believe “Security design patterns representation in Architecture phase as a resistant tool to insecure features”. This enables the software development team understanding to reconcile the software’s requirements and the architecture in the context of security. Including of security features as security patterns in Architecture phase makes the product more secure in further phases, by making this we can redefine the SDLC as secure SDLC.

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

I. INTRODUCTION

In this paper we are focusing on designing of NFR (Security) as a part of SDLC design tool, not as reconsideration which reduces the reengineering cost. Here we are using the security patterns as the security design tool in the Architecture phase. Patterns are the recurring solutions for a Software engineering specific problems. Patterns are reusable which describe, in a generic for a specific problem and a concern solution [1]. To construct a secure software sufficient, more security patterns are available, but structuring and organization is required for more usability of available patterns. We are illustrating the use of such patterns in the software architect viewpoint.

The software architecture plays vital role to evaluate the product with cognizant design decisions about the structure and relations among the modules of the software. The aim of these decisions is, it ensures all the critical concerns are taken into account right from the beginning of the development product. The major role of the software architect is to develop an architecture that fulfills all the requirements and ready to accommodate changes in future. Architecture always find the best possible solutions with largest number of possible constraints (including time, budget etc.) for every problem, and made the final solution [5].

Security patterns are promising well-known techniques for security objectives for secure architectural design. The pattern description is about intent, applicability, problem, solution, structure, consequences, and related patterns [1]. With this description architect need not be a sound knowledge in security concerns, as the security concern arises, the architect can pick the suitable security pattern from the catalog and represent it in the design, and then the selected pattern become a time proven solution for that security concern. In the past ten years, around 220 security patterns documented [2], as it shows there is no shortage in security design patterns. Multifold security principles are manifested in security design patterns [15]. Security patterns and Analysis patterns can be used to build secure systems and conceptual models [16] [17].
In spite of the advantages of security patterns, there are several significant issues yet to be resolved. These large number of security patterns, some of them functioning for the same purpose with different names. Some security patterns are overlapping with other [3]. Some of the security patterns are over and under specification [4]. Most of the research on the security patterns are empirical study not experimental research. Design patterns are successful solutions in software engineering domain, but the same is not in all security patterns. In our next research focusing on organizing the catalog based on security objectives, security taxonomy we are analyzing security patterns using the experimental study.

This paper is organizing as related work in section II, importance of nonfunctional requirements elicitation and design at early stages of the SDLC in section III, design of case study using security design patterns in section IV and Conclusion with future work in section V.

II. RELATED WORK

Many approaches are existing, for the design of security awareness software development life cycle (saSDLC). Security (NFR) must be treated as invitro perimeter to develop saSDLC[5]. Some of them are used security design patterns for analyzing and handling the potential attacks [6]. Some approaches demonstrated Security design patterns as a tool for product security improvement in Architecture phase [7]. The insistence of security in software product at the Architecture phase can reduce the overhead cost in development phase [8].

Halkidis, proposed a method for Architecture risk analysis of software applications using security design patterns with mathematical model (fuzzy set theory and fuzzy fault trees) by using UML class diagrams of security critical application [9]. In the security development community a well-known proven methods must be used than adhoc solutions from scratch [10].

Wide survey was done on security patterns and published [25]. Detailed J2EE Core security patterns are available [26]. Many empirical studies are existing [27]. In the past ten years, different proposals are available for classifying and organizing the catalogs.

III. IMPORTANCE OF NONFUNCTIONAL REQUIREMENTS ELICITATION AND DESIGN IN EARLY STAGES OF SDLC

The most critical phase in the SDLC is the Architecture Phase [11]. The complete structure and the relationships, identification of modules of the software is defined in Architecture phase of SDLC. With balanced architecture Architects achieve complete system quality with optimum system characteristics by means of communication during entire coding phase. Architects involves directly or indirectly for quality balance in coding module. Hence Architects are responsible for complete system qualities [12].

To attain the harmony between functional and nonfunctional requirements, Architecture is the best module to represent. The complete software product is emphasized, predicted in the Architecture phase diagrams and descriptions for the communication with developers [13].

In the public as well as private domains individual or government agencies are using enormous software products, hence the security is most critical concern in the product implementation. Most of the software researchers recognized and accepted that it is better to use the design of security from the scrape. It is advised to inculcate the security feature at early stages of SDLC for secure product.

Most of the software development methodologies are more focusing on how to speed up the development of the product with functional requirements but the above methodologies have inability to elicit nonfunctional requirements as the primary artifacts.[17][18]. In the traditional approach they are treating the NFRs as secondary artifacts [19] [20] [21] [22]. Non-functional requirements must be treated as Primary or Specific requirements in SDLC [23].

IV. PATIENT MONITORING SYSTEM DESIGN USING SECURITY PATTERNS
In our approach we are using the universal standard language UML tools and security design patterns together to develop architecture to represent secure software architecture. Security Design Patterns are reusable to specific problems, provides feasible design, addresses all the security problems and avoid redesign [14].

To develop an Architecture with security concerns, we have chosen an electronic health domain application named as Patient Monitoring System (PMS) [27]. In this application, a doctor is monitoring the patient remotely using the sensors. A wearable unit of the patient is continuously measuring the patient condition such as a pulse rate, blood pressure. These readings are collected and stored in the PMS device (smart phone). The stored PMS database, analyzes and alerts the doctor about the patient health condition if it comes to abnormal condition. When it analyzes and indicates as an emergency, it notifies the emergency services. The same system is accessible by the doctor, patient care taker as well as the patient relatives. They can review the patient’s status. The same system also connected to the hospital information system (HIS).

In this system, the following are the tasks that are to be designed to support the security requirement in the architecture phase to achieve a security objective.

1) An attacker tries to bring down the sensor reading database, this leads to non-availability of updated data to doctor about patient condition (Security Objective: Availability).
2) Any unknown user may tamper the database, so that wrong data about patient may receive by the doctor (Security Objective: Accountability).
3) Sensitive medical details, Sensor Data may be accessed by the unauthorized persons (Security Objective: Confidentiality).
4) The web interface of the PMS is targeted by a malicious intruder, by giving malicious inputs in order to cause PMS malfunctioning or unauthorized access to patient information(Security Objective: Data Integrity).

For the above security critical tasks the suitable patterns must be selected by the Architect.

We are illustrating the above security critical application using the Use case diagrams. We are developing the design with security concern tasks using the security patterns.

To model the above tasks with security concerns, the following are the security objective wise security patterns are identified for the PMS application. The following is the table given the above values.

<table>
<thead>
<tr>
<th>Task No</th>
<th>Security Objective</th>
<th>Security Design Patterns</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task 1</td>
<td>Availability</td>
<td>Firewall</td>
</tr>
<tr>
<td>Task 2</td>
<td>Accountability</td>
<td>Authentication Enforcer</td>
</tr>
<tr>
<td>Task 3</td>
<td>Confidentiality</td>
<td>Secure Pipe</td>
</tr>
<tr>
<td>Task 4</td>
<td>Integrity</td>
<td>Input Guard</td>
</tr>
</tbody>
</table>

The Firewall security pattern prevents the attackers from probe to internal network.
The Authentication Enforcer, and Audit Interceptor; Secure logger patterns performs how an application client must authenticate with application.
The Secure Pipe security pattern secures the connection between the systems.
The Input Guard Security pattern stop the propagation of the error from outside to inside of the system.

The following is the design of Patient Monitoring system using security design patterns:
In Fig. 1, we presented NFR (Security) by means of security pattern Authorization Enforcer to achieve the security objective: Accountability. We represent the authorization enforcer pattern to delegate the authorization process between the Doctor and the PMS device database. This security pattern ensures authorization in all constraints of communication between the PMS Device and the doctor.

Once the doctor can authenticate himself from the remote login, he can access the patient database, analyze the data, and intimate if any emergency can intimate to the patient care taker.

In Fig. 2, we presented NFR (Security) by means of security pattern Secure Logger to achieve the security objective: Accountability. Patient care taker authenticate first, then read the patient database and continuously monitoring the patient. Secure logger pattern provides security auditing.

In Fig. 3, here we are presenting the availability security objective in terms of Firewall security pattern while the malicious attacker tries to bring down the PMS database.

In Fig. 4, we are presenting the confidentiality security objective in terms of secure pipe security pattern when the unauthorized entities tries to access patient information in which they are not relevant to any means.
Data in between the patient’s gateway & PMS system is encrypted, (Security Objective: Confidentiality)

Unathorized User

Tries to access the patients sensitive Information

<<Secure Pipe security pattern>>

Fig.:4 Use case Diagram for Unauthorized user

Data Integrity is achieved through the Input Guard security pattern which is targeted by a malicious entity providing the malicious inputs to the system to make it malfunctioning. The presentation of security concern data integrity is represented in the form of security pattern i.e. input guard is shown in Fig. 5.

Malicious Entity

Tries to provide malicious inputs for system malfunctioning

<<Input Guard security pattern>>

The external input is validated before it is used (Security Objective: Data)

Fig.5: Use case Diagram for Malicious Entity

V. CONCLUSION AND FUTURE WORK

In this paper, we have design the PMS Application using the UML with security concerns, security objectives and corresponding security patterns. We believe the proposed illustrations as the security features introduced in Architecture phases of SDLC which is useful for the next evolving stages. How to use the combination of patterns in the design, the concrete UML tool to present this security concerns in the Design phase instead of stereo types and labels yet to be redefined.

REFERENCES

“Software Project Management”, A Unified Frame Work by walker Royce


I. V. Krav, "Computer vulnerability analysis", PhD thesis, Purdue University, 1998


Chris steel, ramesh nagappan...ray lai, “ core security patterns “.

Koen Yskout, Riccardo Scandariato, Wouter Joosen, “Does Organizing Security Patterns Focus Architectural Choices?”.