International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-9 Issue-3, February 2020

Development of IoT Health Monitor System using Security Patterns

E. R. Aruna, A. Rama Mohan Reddy, K. V. N. Sunitha

Abstract— Context: The most important non-functional requirement of the software application is the security. Developing Secure Software is a challenging Process. Software vulnerabilities and defects may disclose by developers, users, hackers due to Software-intensive systems get connected more and more in every day's lives. A better way to develop secure software is, enhance security processes in all the phases in SDLC. To enhance security in SDLC process required lots of mechanisms and systematic measures to assess the security during the development process. Objective: In this paper, we propose a method "Security aware-Software Development Life Cycle (Sa-SDLC) using Security Patterns". We also measure our security efforts in SDLC. This method fills the insecurity gaps from root level to top level in Granular style approach. Our method is suggestible for security critical applications such as Medical, Finance, Legacy and Communication (Messaging like email) Systems. Results: we successfully implemented our approach on remote health monitor since IoT devices are convenient in everyday life, these devices are using in home, environment, healthcare due to its feasible networking, storage and process features etc. In IoT health care applications, security of the sensitive data is paramount since humans are part of the IoT platform. IoTs heterogeneous network connectivity and expected growth, opens many new threats and attacks which impacts on life of a patient. Conclusion: Hence, our proposed methodology is implemented on Security Essential IoT based health care application and measures shows our method is improved software security.

Keywords: Sa-SDLC Methodology, Secure Patterns, IoT Application

I. INTRODUCTION

Internet becomes an integral part of everyone's life. Due to this, viruses, malicious actors, hackers, attackers increasing day by day, this badly affects vital information of the users. On the other hand Software Vulnerabilities are enabled due to implementation flaws or human weakness such as careless configuration, poorly chosen passwords. Hence Security must be injected into the application in all the software development phases such as requirements gathering and analysis, design, implementation, testing and deployment to achieve more secure software. Security must be an inbuilt part of iterative development process [1].

Revised Manuscript Received on February 05, 2020.

* Correspondence Author

E. R. Aruna*, Department of IT, Vardhaman College of Engineering, Hyderabad, India

A. Rama Mohan Reddy, Department of CSE, SVU College of Engineering, S.V University, India

K. V. N. Sunitha, Department of CSE, BVRIT College of Engineering, Hyderabad, India

Due to the lack of security measurement to quantify the performance, most of the software products are unsuccessful to produce protected environment for the end-users, still those products undergone through several formal methods and techniques for security. Security Process must be followed in consistent manner as per the procedures and rules throughout the SDLC. On the other hand, security measures such as metrics, frameworks, procedures, reviews etc. must use to quantity the efforts and progress [2].We can minimize the Software vulnerabilities by auditing and controlling every step in the software development Process [3].The Sa-SDLC method proposed several security mechanisms and measures to evaluate level of security in each process within the iteration and also after each development module.

Security Patterns mechanisms are used by information security engineers. There are vast number of patterns in the literature, to choose the appropriate pattern (Requirement phase) from the repository, model the pattern context (design phase) using security Modeling tools like secureUML, UMLsec etc. and implementation and validation of the pattern (development phase) and measure the efforts for security before and after applying these patterns are key functionalities in our work.

Motivation towards Sa-SDLC using SPs methodology:

Internet of Things (IoT) provides connectivity between the devices via wireless protocols, due to its heterogeneous potential the things get connected home, hospital, environment etc. The data through these devices must stored, managed and disposed whenever required. It is imperative to secure sensitive data since in remote health care applications humans are part of IoT Infrastructure. IoT devices low computation devices where as security algorithms are power hungry and computationally expensive. Hence we felt our Sa-SDLC methodology using security patterns is suitable for Security critical applications like IoT, cloud etc. The above problems are evaluated and measured using our proposed methodology.

II. RELATED WORK

Karen Goertzel et al discussed about existing security techniques, secure software in SDLC, secure SDLC principles, practices, security oriented requirements, Modeling for security capturing, Secure design principles and practices, secure design modes, Secure coding practices, Secure coding guidelines, security challenges, evaluating security[3].

Adel Mohammad et al evaluated three security top approaches McGraw's Touchpoints, Comprehensive Lightweight Application Security Process (CLASP) from Open Web Application Security Project (OWASP) organization, and Microsoft Security Development Lifecycle (SDL) with their process commonality, strengths, limitations and recognized cost, time and Lack of security knowledge are the reasons for security is not considered in software development. They suggested Adaptive Risk Framework and Automated Tool to enhance security activities within all SDLC phases and to consider security from the beginning of the project to the end [4].Anuradha Sharma et al compared existing 8 security techniques in the view of advantages and disadvantages for secure SDLC. All the techniques emphasized in design phase [5].

Measurement is a fundamental tool in Professional security management. Alfonso Bilbao et al proposed several parameters such as effectiveness, efficiency, performance, evolution of the threats, maturity to analyze security resources. They conclude security measure is the essential in risk management which focused on continuous improvement, they applied measures an example on "clothing store chain" [6]. Francis et al proposed attack graph analysis technique quantifies and compare the baseline state and attack surface after implementation of security framework. This technique also create quantitative measure allows to compare the efficiency of security framework across different applications [7].

Effective methods and tools must be used to evaluate software security and also Security performance must be measured in each process development iteration [8]. Minela Grabovica et al, analyzed about communication technologies such as RFID, Bluetooth, Wireless network and ZigBee and available measures for them In IoT. Finally discussed about the advantages of those technology measures [9]. Rizwana A.R. Shaikh et al proposed data security measurement tool is proposed to measure the data security and privacy of the cloud service provider. These security measurement enabled trust value parameter acts as checklist to be verified before selecting the cloud provider services [10].

Internet of Things (IoT) technological innovations impacts on economy in one hand, but the cyber risk of IoT impacts on the economy damage in other hand [11].

Stefano Tedeschi et al introduced an design approach for end point security in IoT connectivity to upgrade the legacy production machinery. This approach raised the future research includes mitigation mechanisms, systematic test generation and validation of solutions, automated risk assessment, impact of endpoint vulnerabilities, virtual isolation, systematic assessment and management of IoT security[12].

The major strengths of IoT: IoT is emerging in recent years with the capabilities of wireless protocols such as RFID, Wi-Fi, LTE etc. IoT promoting its potential with the performance of the protocols in terms of interoperability, scalability, reliability, quality of services, real-time, cost-effectiveness etc. In the other side, security challenges such as privacy and confidentiality w.r.to these protocols need to be handled to get the full essence of the IoT [13]. Rob van Kranenburg et al stated the major weakness of Health care IoT(HIOT) is authorization, authentication, asset access control, real-time monitoring policies, access privileges and technical challenges [14]. Some of the open challenges like standardization, quality of health services, real-time monitoring, data storage space in implementing HIoT are acknowledged by S. M. Riazul Islam[15].

Security Pattern (SP) encapsulates reusable solutions for specific security problem. Hironori Washizaki et al [16] proposed taxonomy for security patterns, this taxonomy is used to select and use of the security patterns effectively. They prepared this SP taxonomy from research work on 200 papers about security patterns, attack patterns and misuse patterns. Continuous remodeling of systems for new deeds increases the complexity and more vulnerability, to avoid this problem, reliable and validated solutions are needed. Roberto Ortiz et al [17] suggested security patterns for this task, since these patterns provide documentation and solutions for recurrent security problems. Eduardo B. Fernandez et al[18] focused on core areas like cloud and IoT security. They have emphasized on secure architectures using the security patterns. They proposed future research scope on testing and empirically evaluating the security patterns to prove these pattern solutions are superior to other approaches.

Rest of the paper is organized as follows, section 3 presents the proposed Sa-SDLC methodology, section 4 implementation, section 5 presents results and discussion, and section 6 concludes the paper.

III. THE PROPOSED METHODOLOGY

We adapted Text Categorization Process [19], Restricted Misuse Case Modeling [20], validating security patterns using Test Template [21] approaches to extend our Methodology.

The proposed Sa-SDLC methodology, integrate security patterns mechanisms in each phase of SDLC to reduce impact of vulnerabilities. The block diagram Sa-SDLC is shown figure 1.

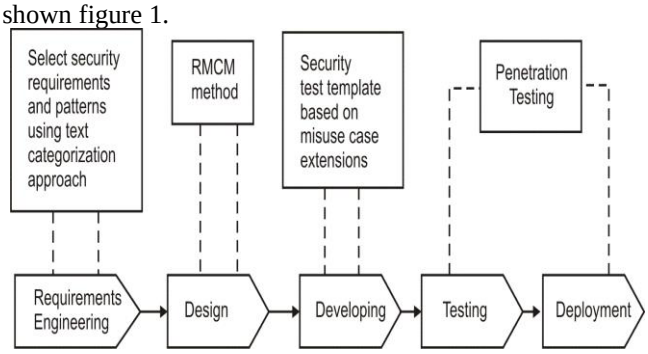

Figure 1: Block Diagram (Sa-SDLC using Security Patterns)

The Steps in our Methodology:

1. Select Functional and Security requirements of an application.

1.1. Identify Security Patterns for each Security Requirement from the pattern repository(To select the pattern,

we have used Text Categorization Process with Security Pattern Repository and Software Requirement Specification (SRS)).

2. Model the functional use cases along with security requirements and Pattern solutions using a Restricted Misuse Case Modeling (RMCM).
2.1. Elicit use cases with misuse case extensions diagram.
2.2. Derive the specifications for Using RMCM-V tool, check the consistency of diagram and specifications.
2.3. Elicit mitigation schemes (for specified threats in Misuse cases).
3. Develop the functional code and derive Test templates to support for mitigation of Misuse case.
3.1. Derive aspect test template based on Misuse case Extension
3.2. Use Test Driven Development (TDD) approach to create reusable Test case based on aspect test template.
4. Perform penetration Testing.
5. Deploy the Application with secure configuration.

Goals of Sa-SDLC:

- Projects Requirements stability
- Security ready product
- Reducing huge cost of post implementation for remediation.

As a justification, we applied "**Sa-SDLC using Security Patterns**" approach on Remote Health Monitor using IoT.

IoT Health Monitoring System: In figure 2, Health Monitor System Design, Patient bio-information like temperature, pulse rate, heart beat is captured through sensors. Sensor information is received by arduino. Through smart gateways patient data is transferred to cloud database from arduino. Remote care taker reviews the patient data through web application access.

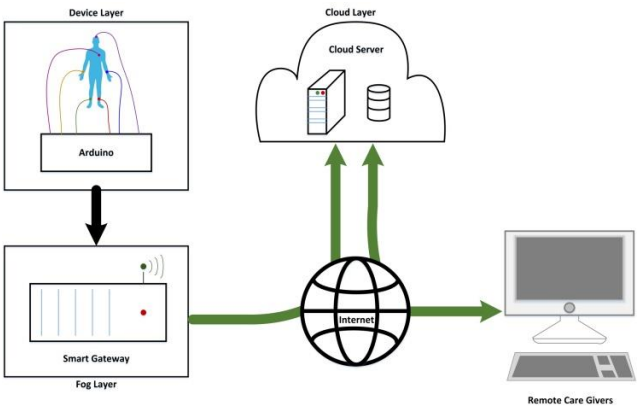

Figure 2: Iot Health Monitor System Design

IV. IMPLEMENTATION

Step 1: Functional and Security Requirements of Remote Health Monitor System is

Functional Requirements:

1. This application enables, monitoring and tracking of Temperature, Heart rate of patient who is in remote location.
2. Sensors read Patient data, through arduino this data has to be transferred to Database using Wi-Fi.
3. The remote care takers (Doctor) reviews the patient data from time to time by accessing web application.

Security Requirements of Remote Health Monitor:

1. Arduino has to be authorized before storing data to the database.
2. Input data from trusted boundaries must be validated before process and grants access privileges.
3. Database has to be authenticated before store the data to the database.
4. Ensure the sensitive patient records remains confidential in transit, using encrypt forms. The sensitive data stored in database must be encrypted and also web application logon credentials needs to be encrypted and stored.
5. Access rights to be authorized before granting privileges to access the patient sensitive data by the remote care takers and all roles need to be defined and access rights for the records to be encapsulated.
6. Establish secure session, when user logged in and logged out. Web application Need to be invalidated the session when user is in inactive for certain period. Session Id must be unpredictable.
7. Provide maximum use of functionality to the users without presence of errors. But any security exceptions, while handling them not to disclose technical or sensitive details in simple error messages.
8. Need to be capture all logs and events, these audit records must be protected from being tampered.
9. Ensure the patient records do not alter/manipulate during usage of records by the malicious user.

Step 1.1 Identification of Security Patterns for Security Requirements: To identify the security patterns, Security Requirement Specification (SRS) and Security Pattern Repository (SPR) has to be prepared [19][22][23] as show in Table 1 and 2 respectively.

Part of structure of SRS *Query:*

Table 1: SRS Query

SRS No	1
SRS Name	Remote Health Monitor
Domain	IoT
Purpose/Scope	Purpose: Design a Remote health Monitor system. Scope: Capturing patient data like heart beat using sensors, Transit data to cloud database storage and provide data to remote viewing.
Functional Requirements	This application enables, monitoring and tracking of Temperature, Heart rate of patient who is in remote location.
	Arduino and Sensors read Patient data, this data has to be transferred to Database using Wi-Fi.
	Data collecting from Arduino, storing in the database and presenting to remote care takers done by web application
Non-Functional Requirements	Arduino has to be authorized before storing data to the database and Database has to be authenticated before process and store the data.
	Access rights to be checked before granting privileges to access the patient sensitive data by the remote care takers.
	Establish secure session, when care taker logged in and logged out. Web application need to be invalidated the session when the care taker is inactive for certain period. Session Id and other session related details must be unpredictable.
	Provide maximum use of functionality to the users without presence of errors. But any error issues, error messages do not disclose technical or sensitive details.

	Need to be capture all logs and events, these records must be protected from being tampered.
	All roles need to be defined and access rights for the records to be encapsulated.
	Input data from trusted boundaries must be validated before process and store.
	Ensure the sensitive patient records remains confidential in transit, using encrypt forms.
	The sensitive data stored in database must be encrypted and also web application logon credentials needs to be encrypted and stored.

Part of Security Patterns Repository:

Table 2: Security Pattern Repository

Pattern Id	#1	
Pattern Name	Secure Directory	
Abstract	Ensure attacker cannot alter the files during execution of application	
Problem	Attacker may alter/delete a file during the file data relies on "unmodified", it causes "race condition".	
Solution	Canonical Path referred by Directory (contains files) is secure (valid path), if and only if allow the file to read/write.	
Tradeoffs	Accountability	Ensures the security for the directory and files
	Confidentiality	Does not allow malicious user to access file for modification
	Availability	Files available only to the users request through valid paths
	Integrity	Reliability of files in the directory

Process of selecting the pattern from repository is shown in Pattern retrieval model in figure 3.

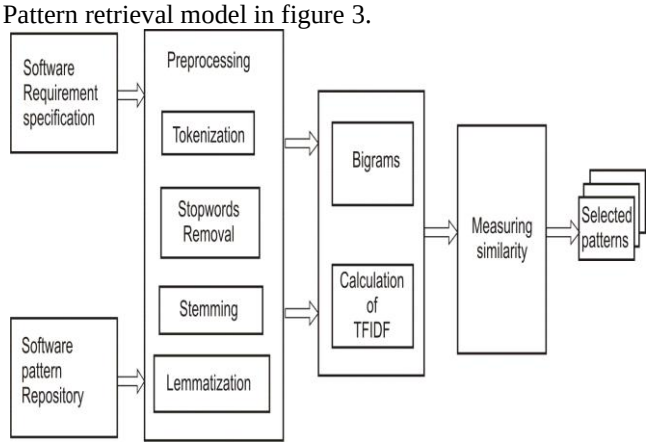

Figure 3: Pattern Selection Model

For SRS and SPR, preprocessing is done by performing Tokenization, stopwords removal, stemming, Lemmatization. and Bigrams are used to differentiate the security patterns. Singular Value decomposition (SVD) may apply for noise reduction and dimension reduction. Text categorization is done using Vector space model (VSM) Term-document category out of 3 VSM categories (other two are word-context and pair pattern document) for indexing. For SRS, indexing attributes are SRS No, SRS Name, purpose and Scope, Functional requirements, Non-functional requirements.

For SPR, the indexing attributes are Pattern Id, Pattern Name, Tradeoffs and single featured vector together Abstract, Problem and Solution. To know the weight of term, we used the filter based approach know as Term Frequency Inverse Document Frequency (TFIDF) to remove the features.

The security design repositories have selected from the resources [24][25][26][27]. Based on the word similarity, the patterns resulted from Text categorization Approach shown in Table 3.

Table 3: Identified Patterns using Text categorization

Security Requirement No in section IV.	Security Pattern
1 & 3	Secure Adapter Pattern
2	Input Validation Pattern and Password Design and Use Pattern
4	Secure Channels Pattern
5	Authorization Pattern , Role-based Access Control Pattern
6	Security Session Pattern
7	Exception Manager Pattern
8	Secure logger Pattern
9	Secure Director Pattern

The prezaise context of pattern is analyzed with security requirements, this context provides a better communication between the requirements engineering to design. Throughout the next stages security requirements must be evaluated based on solution provided by security patterns.

Step 2. We model security requirements using the Restricted Misuse Case Modeling approach.

2.1 The part of usecase diagram with Misuse case extensions of IoT Health Monitor is shown in figure 4.

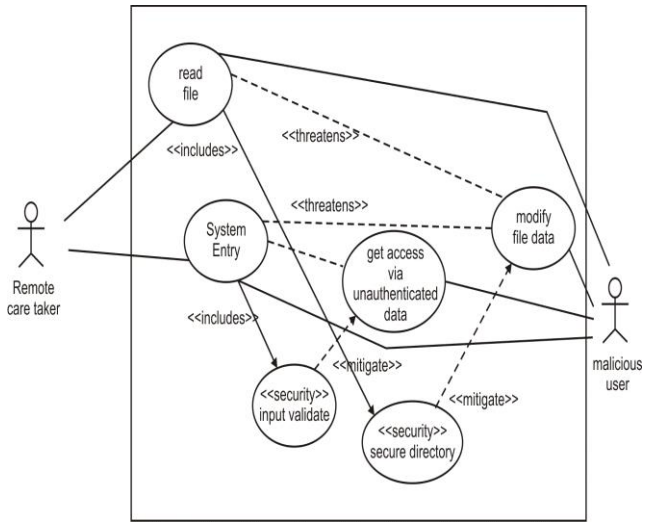

Figure 4: Misuse Case extension

2.2 The specifications for misuse case and security use case are shown in table 4 and table 5 as per the RMCM template.

Table 4: Part of Misuse Case Specification

1	**MISUSE CASE** Get System entry via unauthenticated
2	data
3	**Precondition** At least one web application user must
4	registered
5	**Primary Actor** ATTACKER
6	**Threats** log in
7	**Assets** Patient sensitive DATA
8	**Basic Threat Flow**
9	1. The ATTACKER get system entry via
10	unauthenticated(SQLi) or spoofing registered
11	username and password
12	2. The username and password query evaluated by
13	the system from Database
14	3. The system VALIDATES THAT, if query is
15	successful, system sends welcome message to
16	ATTACKER
17	**Postcondition** The attacker accessed Patient sensitive
18	DATA
19	**Specific Alternative Threat flow**
20	RFS 3
21	1. DO
22	2. SENDS the error message to ATTACKER
23	3. The ATTACKER EXPLOITS error message
24	4. The ATTACKER PROVIDES SQLI VALUES
25	IN username and password UNTIL the query is
26	successful
27	5. RESUME STEP 4.
28	**Postcondition** The ATTACKER access the Patient
29	DATA
30	Bounded Alternative Flow
31	RFS SATF1 1-4
32	1. IF the login attempts is reached to maximum
33	THEN
34	2. Invalid login message send by the System TO the
35	ATTACKER
36	3. ABORT
37	4. ENDIF.
38	**Postcondition** The ATTACKER cannot access Patient
39	Sensitive DATA
40	Mitigation Scheme Secure Mechanism using Password
41	Design and Use Pattern
42	**MISUSE CASE** request for FILE to alter/delete
43	**Precondition** FILE exists the database
44	**Primary Actor** ATTACKER
45	**Threats** alter/delete the FILE
46	**Assets** FILE contains sensitive DATA
47	**Basic Threat Flow**
48	1. The ATTACKER requests the FILE, FROM the
49	database.
50	2. The system grants access to FILE to
	ATTACKER.
	Postcondition The ATTACKER obtained the FILE can
	ALTER/DELETE.
	Specific Alternative Flow
	RFS 2
	1. IF the FILE path is INVALID
	2. ABORT.
	3. ENDIF.
	Postcondition The ATTACKER did not obtain the FILE
	DATA.
	Mitigation Scheme Secure Mechanism using Secure
	Directory Pattern

Part of Security Use case Specification shown in Table 5.

Table 5: Security Use case Specification

1	**SECURITY USE CASE** FILE path Validation
2	**Precondition** at least one File existing in the system
3	**Compliance** System and application access control(ISO/IEC
4	27001:2013,clause A.9.4)
5	Mitigate access through unauthenticated DATA, Delete or
6	Modify File DATA
7	**Basic Flow**
8	1. The system receive the FILE request
9	2. The system VALIDATES THAT the FILE
10	Path
11	**Postcondition** successfully validated Path using
12	canonicalization.
13	**Specific Alternative Flow**
14	RFS 2
15	The system displays an invalid FILE path.
16	ABORT.
17	**Postcondition** Error message is displayed.

Step 2.3: The mitigation schemes provided by Security Pattern solutions shown in table 6 as per the RMCM template

Table 6: Part of Mitigation Scheme from Pattern Solution

Scheme Name Secure Implementation of Remote Health monitor based on IoT
Brief Description This mitigate scheme, mitigates the vulnerabilities for IoT based remote health monitor application
Actors Software Developer (Need not be a security specialist)
Mitigated Misuse Cases Expose sensitive data and access, alter/modify the FILE DATA due to lack of authentication and authorization
Compliance ISO/IEC 27001:2013 clauses A.6.1.5, A.9.4, :A.10.1 Information security, access control and cryptographic controls respectively.
Mitigation Tasks
1. VALIDATE input data from trusted boundaries with registered data
2. ENCRYPT patient sensitive DATA and logon credentials of input data.
3. VALIDATE roles before grant privileges for read/write permissions.
4. Check path canonicalization before FILE granted
5. Apply selected security patterns solutions for the identified specific problems

In the extended level, the consistency of the misuse case extension and its specifications may be verified by Restricted Misuse case Modeling –Verifier (RMCM-V) tool.

Step 3: Develop the functional code and derive reusable Test template from mitigation scheme to support remediation of Misuse cases (vulnerability).

3.1 Create decision Tables and Pointcut Advice Model (Using Aspect Oriented Programming).

As per the Misuse case extension, prepared the decision table and pointcut advice pair to understand the internal processing of secure directory pattern is

The decision tables and Pointcut advice is shown in table 7 and figures 5, 6.

Table 7: Decision Table of Security Directory Pattern

		1	2	3	4
Conditions	Directory path for requested File is valid	**Yes**	**Yes**	**NO**	**NO**
	User permission to access the File is true	**Yes**	**No**	**Yes**	**No**
Actions	File can Access from the Directory	✓	X	X	X
	Deny to access the File from the Directory	X	✓	✓	✓

```
pointcut validateSecureDirectory():
call(* *.Secure_Directory.checkSecureDirectory(..);
call (* *.Secure_Directory.validatePath(..) ;
```

Figure5: Pointcut for validating directory

```
after () returning(Boolean right):
Permit_accessfile() { setTemporary ("validateSecureDirectory",right);}
```

Figure 6: advice to provide file access from the directory

3.2 Create reusable Test case based on Test Driven Development (TDD) approach using 3.1.

Part of the test case template is shown in figure 7 derived from Security Directory Pattern. This test case is reusable. This will support the pattern validation during development phase. The developers may not have security knowledge can use this Template.

Step 4: The application, Penetration Test has done and verified the functional and security requirements. The security penetration is conducted dynamically. One Test case results is shown in figure 8.

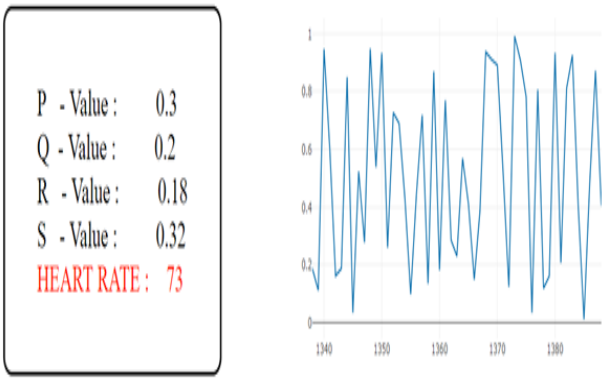

Figure 8: Heart Rate of Patient

```
package secureDirectoryPattern;
import org.junit.Before;
import org.junit.Test;
import secureDirectoryPattern.SecureDirectorRepository.Permission;
import static org.junit.Assert.assertEquals;
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
public class SecureDirectoryCheck {
private static SecureDirectorRepository repository = null;
private static String userId, userRequestfilePath;
private static Permission requestAccess;
@Before
public void setUp() throws Exception {
repository = new SecureDirectorRepository();
setUserData("uttej","D:\\data\\ARUNA\\log.txt",
Permission.READ);}
@Test
public void validateUser() {
assertEquals(repository.usersMap.containsKey(userId), true);
assertEquals(repository.permissionsMap.containsKey(repository.user
sMap.get(userId)), true);
String user = repository.usersMap.get(userId);
if (repository.permissionsMap.get(user) == requestAccess) {
try {
String data;
StringBuilder dataBuilder = new StringBuilder();
@SuppressWarnings("resource")
BufferedReader br = new BufferedReader(new
FileReader(userRequestfilePath));
while ((data = br.readLine()) != null) {
dataBuilder.append(data);}
System.out.println(dataBuilder.toString());}
catch (IOException e) {
e.printStackTrace();
} finally {}}
else {throw new RuntimeException("Unauthorized!!!");}}
public void setUserData(String id, String filePath, Permission
permitAccess) {
userId = id;userRequestfilePath = filePath;requestAccess =
permitAccess;
}}
```

Figure 7: Part of Reusable Test Case Template

The test case using Security Directory Pattern solution is shown in figure 9.

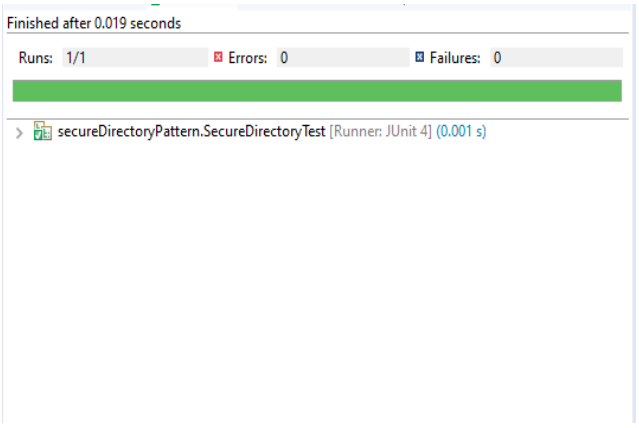

Figure 9: Secure Directory Pattern Test Case Result

Step 5: The application is deployed with secure configuration, such as use of legacy equipment, standard devices and time to time OS updation.

At this phase the risk is low profile, simple remedies like disable auto-run features without authorization, remove unnecessary accounts, authenticate is mandatory before access data, software uninstall if they are unnecessary etc.

V. RESULTS AND DISCUSSION

Based on the methodology, the developed code exhibits the following results shown in table 8.

Table 8: PQRS complex values and Heart Beat of 5 Patients

Patients	P	Q	R	S	Heart beat rate
Patient 1	0.02	0.009	0.6	0.06	71
Patient 2	0.14	0.12	0.65	0.05	70
Patient 3	0.002	0.001	0.82	0.123	109
Patient 4	0.2	0.31	0.92	0.287	80
Patient 5	0.4	0.091	0.88	0.282	89

Whenever sensors are attached to the patient body, automatically secure code runs to capture the data and transfer to database with secure communication. Doctor can see and review the patient condition from anywhere after authenticated through web application.

VI. CONCLUSION

In literature, selection and application of security patterns for security requirements of IoT does not have empirical methodology. The limitation in [21] may overcome i.e Object Constraint Language (OCL) may not necessary to prepare Decision Table and pointcut advice Model (Aspect Programming) to understand the pattern internal processing. Instead of OCL, we can use Misuse case extension to prepare decision table and Pointcut Advice pair model.
Using Sa-SDLC, we successfully integrated security in each phase using security patterns. This Sa-SDLC is most suggestible for security critical Applications. We believe, security patterns are the powerful tool for Secure SDLC.

REFERENCES

1. Pieter Frijns,Robert Bierwolf,Tom Zijderhand,” Reframing Security in Contemporary Software Development Life Cycle”, IEEE International Conference on Technology Management, Operations and Decisions (ICTMOD), 2018.
2. Nunes FJB, Belchior AD, Albuquerque AB. “Security engineering approach to support software security. In SERVICES. 2010”, July; 48–55.
3. Karen Mercedes Goertzel, Booz Allen Hamilton,” Enhancing the development life cycle to produce secure software”, Software Assurance Forum, 2008.
4. Adel Mohammad, Ja'far Alqatawna, Mohammad Abushariah,” Secure Software Engineering: Evaluation of Emerging Trends”, 8th International Conference on Information Technology (ICIT),IEEE,2017.
5. Anuradha Sharma, Praveen Kumar Misra,” Aspects of Enhancing Security in Software Development Life Cycle”,Advances in Computational Sciences and Technology, 2010.
6. Alfonso Bilbao, Enrique Bilbao,” Measuring Security”, 47th International Carnahan Conference on Security Technology (ICCST), IEEE, 2013.
7. Francis J. Manning, Frank J. Mitropoulos, Utilizing Attack Graphs to Measure the Efficacy of Security Frameworks Across Multiple Applications”, 47th Hawaii International Conference on System Science,IEEE,2014.
8. Shehab A. R. Farhan and Mostafa G. M. Mostafa,” A Methodology for Enhancing Software Security During Development Processes”,21st Saudi Computer Society National Computer Conference (NCC), 2018.
9. Minela Grabovica, Dražen Pezer,” Provided security measures of enabling technologies in Internet of Things (IoT): A survey”, Zooming Innovation in Consumer Electronics International Conference (ZINC), IEEE,2016.
10. Rizwana A.R. Shaikh, Masooda M. Modak, “Measuring Data Security for a Cloud Computing Service”, International Conference on Computing, Communication, Control and Automation (ICCUBEA).2017.
11. Petar Radanliev, Dave De Roure , Stacy Cannady, Rafael Mantilla Montalvo , Razvan Nicolescu, Michael Huth,” Economic Impact of IoT Cyber Risk - Analysing past and present to predict the future developments in IoT risk analysis and IoT cyber insurance”, Living in the Internet of Things: Cybersecurity of the IoT, IEEE,2018.
12. Stefano Tedeschi, Christos Emmanouilidis, Jörn Mehnen ,Rajkumar Roy,” A Design Approach to IoT Endpoint Security for Production Machinery Monitoring”,Sensors,MDPI,2019.
13. Jiangchuan Liu,Feng Wang,2Xiaoqiang Ma, and Zhe Yang, "Recent Advances in Wireless Communication Protocols for Internet of Things" ,Hindawi Wireless Communications and Mobile Computing, WILEY,2017.
14. Rob van Kranenburg, Alex Bassi, “IoT Challenges”, "IoT Challenges," Communications in Mobile Computing”, A springer open,2012.
15. S. M. Riazul Islam, Daehan Kwak, MD. Humaun Kabir, Mahmud Hossain ; Kyung-Sup Kwak, "The internet of things for health care: a comprehensive survey," vol. 3, IEEE Access, 2015.
16. Hironori Washizaki, Tian Xia, Hideyuki Kanuka, Dan Yamaoto, Takao Okubo,” Taxonomy and Literature Survey of Security Pattern Research”, IEEE Conference on Applications, Information and Network Security (AINS),2018.
17. Roberto Ortiz,Javier Garzás,Eduardo Fernández-Medina,” Analysis of Application of Security Patterns to Build Secure Systems”, International Conference on Advanced Information Systems Engineering, Springer,2011 and Part of the Lecture Notes in Business Information Processing book series (LNBIP, volume 83).
18. Eduardo B. Fernandez, Hironori Washizaki, Nobukazu Yoshioka,” Using Security Patterns to Develop Secure Systems—Ten Years Later”, International Journal of Systems and Software Security and Protection Volume 9, Issue 4, October-December, IGI Global,2018.
19. Ishfaq Ali, Muhammad Asif,Muhammad Shabaz,Adnan Khalid, “Text Categorization Approach for Secure Design Pattern Selection Using Software Requirement Specification”, Volume 6, IEEE Access, 2018.
20. Phu X. Maia, Arda Goknil, Lwin Khin Shar, Fabrizio Pastore, Lionel C. Briand,” Modeling Security and Privacy Requirements: a Use Case-Driven Approach”, Information and Software Technology, Elsevier, 2018.
21. Masatoshi Yoshizawa, Hironori Washizaki, Yoshiaki Fukazawa, Takao Okubo, Haruhiko Kaiya, Nobukazu Yoshioka,” Implementation Support of Security Design Patterns Using Test Templates”, https://www.mdpi.com/journal/information,2016.
22. M.Riaz,J.King,J.Slankas,andL.Williams,‘‘Hidden in plain sight:Automatically identifying security requirements from natural language artifacts,’’ 22nd International Requirements Engineering(RE) conference, IEEE,2014.
23. Alessio Ferrari, Giorgio Spagnolo, and Stefenia Gnesi, ‘‘Towards a dataset for natural language requirements processing,’’ in Proceedings of CEUR Workshop, 2017 and also available ISTI Open portal.
24. Markus Schumacher, Eduardo Fernandez-Buglioni, Duane Hybertson, Frank Buschmann , Peter Sommerlad ,”Security Patterns: Integrating Security and Systems Engineering”, John Wiley & Sons Ltd, Chichester, UK, 2006.
25. Eduardo Fernandez-Buglioni, “Security Patterns in Practice: Designing Secure Architectures Using Software Patterns”, Wiley publishing,ACM, 2013.
26. Chad Dougherty, Kirk Sayre, Robert C. Seacord, David Svoboda, Kazuya Togashi,” Secure Design Patterns”, CERT Program, http://www.cert.org/ , software engineering Institute,2009.
27. Exception Manager Pattern, https://www.blackhat.com/presentations/bh-federal-03/bh-fed-03-peterson-up.pdf.

AUTHORS PROFILE

Mrs. E..R.Aruna, is working as Associate Professor in Department of Information Technology, Vardhaman College of Engineering. She is pursuing her PhD in Computer Science and Engineering in JNTUH, Hyderabad, India. She received Master if Information Technology in 2008, Sathyabhama University, Chennai, India and received Bachelor of Computer Science and Information Technology in 2004, SITAM, Chittoor, AP, India.

Dr. A. Rama Mohan Reddy is a Professor in the Computer Science and Engineering division at Sri Venkateswara University College of Engineering.
His research interests include Software Architecture, Software Engineering, Data Mining and optimising compilers. He received his B.Tech. from JNT University, Hyderabad in 1986, M. Tech degree in Computer Science from National Institute of Technology in 2000 Warangal and Ph. D in Computer Science and Engineering in 2007 from Sri Venkateswara University, Tirupathi, Andhra Pradesh, India.

Dr. K. V. N. Sunitha, is working as Principal and Professor of CSE in BVRIT for Women, Hyderabad, India. She has done her B. Tech (ECE) from Nagarjuna University, M. Tech Computer Science from REC Warangal. She completed her Ph.D from JNTU, Hyderabad in 2006.