International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-9 Issue-2, December 2019

Identification and Remediation of Vulnerabilities in IoT based Health Monitor

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

Abstract— Internet of Things (IoT) changes the trend "Connect the unconnected" likes human or environmental and technical interactions, interactions among the machines through Radio Networks, Sensor Networks and many more simple devices like fitbits, thermostats etc. Besides this, they are highly prone to vulnerable due to its open and heterogeneous nature. To secure IoT Applications, we proposed a methodology called "Process to enhance software security" (PESS) method. The main Moto of PESS is to design Security aware Software Development Life Cycle (Sa-SDLC). In PESS methodology we are applying and assessing the secure assurance activities and security patterns. In our work, we have applied secure assurance activities and secure adapter pattern for securing user information of IoT based Health Monitor Application. Our PESS methodology accomplished the security implementation faults identification and remediation at early phases of IoT application development life cycle.

Keywords: PESS Methodology, Secure Adapter Pattern, IoT Application

I. INTRODUCTION

Conscious devices automation, smart home, remote health care is the key strength due to its ability to integrate virtual space and real world on single platform successfully [1]. Internet of Things adds huge value to Remote Health care, Home or Environmental automation, Industry. Security is the paramount in IoT based health care applications since disclosure or loss of patient sensitive data leads to loss of patient life [2]. Some of the most recent survey for proposed solutions for security in IoT applications are blockchain and Software Defined Networking(SDN), Multi-agent approach to threat detection with machine learning(Vulnerabiities-Patterns) for H-IoT , Two Firewalls with vendor diversity, security integration in SDLC, Authorization of Users using Security Patterns in IoT, security design patterns for vulnerabilities in IoT systems, Remote security management server for safety and security for IoT environment, Mutual Authentication Mechanism using Multi-key [3,4,5,6,7,8,9,10] respectively.

Motivation towards PESS methodology: The Security aware Software Development Life Cycle (Sa-SDLC) is security must be considered as functional parameter, these parameters must be identified, designed and implemented throughout SDLC. Software intensive systems connected and pervaded our everyday lives, this opens up cyber attacks, most of these attacks can only be resolved by secure software_development life cycle [11]. Improve the information security using the SecDLC principals, the main goal of SecDLC is to preserve, monitor and maintain security practices, policies, and standards [12]. For business-critical application, improved security is an easy sell because of the impact of a failed business application. Microsoft adopted Security development life cycle (SDL) for creating more secure software, to develop SDL, Microsoft put more effort in terms of time, cost because they felt "the upfront benefits far outweigh the cost of revisions" [13]. Security Patterns are reusable proven solutions to achieve specific security goals in software development. Hironori Washizaki reported together with researchers and practitioners "the support of security patterns for secure software development and introduced the achievements of projects with security patterns" [14]. Major findings from [15], Research filed on Security Patterns is quite active since many years, because security pattern guaranteeing security by demonstrating its specific context in the application. Still Security Pattern (SP)s has enough ambiences for further research aligned with types and formalization. This shows the effectiveness of SPs and further SP research.Takao Okubo et al[16] analyzed identification of threats using security requirement patterns and countermeasures for threats using Security design patterns, this security pattern integration analysis prioritize possible countermeasures and choose the appropriate countermeasure for identified threat. Annanda Rath et al[17] used security patterns as best practices and knowledge for cloud SaaS, these security aspects protects data security and privacy.

II. RELATED WORK

Secure Design and Deployment is essential in future implementations of interconnected things like Internet of Things (IoT). Ishfaq et al[7], described eight security patterns(Choose the Right Stuff, Security Goal Pattern ,Reference Monitor pattern, Third Party Enrollment Pattern ,Access matrix authorization rules, Remote Authenticator/Authorizer and Role Based Access Control) documentation for authorizing event driven IoT devices and authenticating the users and also ensure devices and users privileges and actions in the System.

Wen-Tin Lee and Po-Jen Law [8] applied five security design patterns(secure logger, security directory, Exception Manager , Secure Adapter and Input validation Pattern), for security issues in IoT Applications.

Revised Manuscript Received on December 30, 2019.
***** Correspondence Author
E.R. Aruna*, Department of IT**,** Vardhaman College of Engineering, Hyderabad, India.
A. Rama Mohana 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

The authors design the IoT system with class and sequence diagrams for these five patterns.

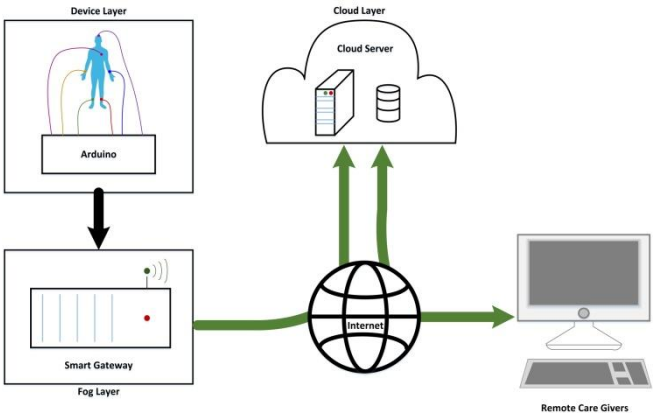
Masatoshi Yoshizawa et al[18], proposed the validation method for security design patterns by creating the aspect test template and Test case template to support the developer during the implementation. Arnon Sturm et al[19] proposed method for implementing security patterns database applications which preserves the access control. This method provides implementation guidelines how to model the application, validating the correct usage of the security patterns, and also secure database schemata automatic generation.

Takanori Kobashi etal[20] proposed a method for validating the security pattern in the applications. This provided extended security patterns at requirement level and design level, and also testing process. Developers identify vulnerabilities and threats in the software to be developed during an early phase of development life cycle. This proposed method validates whether the patterns applied properly and also review all the threats or vulnerabilities resolved or not.

Anika Behrens [21] presented a formal definition which is more suitable for automated processes of code generation or verification for security patterns. His work demonstrate and show the formal security patterns of Chomsky type 3 can be reduced to particular expressions based on some assumptions. His works formalize the security patterns abstractions and produce a mathematical model which is basis for a data structure. These data structures process the security patterns with software tool.

Rest of this paper organized as IoT Health Monitor System Design in section III, Proposed PESS Methodology implementation in section IV, Measurement of efforts of our Methodology in section V and conclusion and future work in section VI.

III. IOT HEALTH MONITOR SYSTEM DESIGN



. Figure 1: Design overview of IoT Health Monitor Application

Figure 1 shows the design of IoT Health Monitor application with three layers enables tracking of remote patient data. This three layers such as Device, Fog and Cloud layers. Bio signals are captured through sensors from Patient. The captured bio information is transferred from sensors to the fog layer through Internet/Wi-Fi. The fog layer contains smart gateways for efficient mobility and extends its mobility to network edge. Fog layer Smart gateways behave like secure communication protocols between device layer and cloud layer. The cloud layer contains data server and maintains the patient's sensitive data.

Avelet Maria Fernandes[23] created threat model for the IoT Health Monitor , using this threat model risk is calculated using STRIDE and DREAD methods. The identified security requirements to be critical at each of the following interactions between IoT Microcontroller(Arduino) and Fog Gateway, Fog Layer and Web Application Server, Web Application server and Database and Web Application server and Client Browser.

In IoT, the interaction between Arduino to cloud server database through Fog layer ,the threats and vulnerabilities are identified and remediated using secure assurance activities such as security requirements, model security, threat analysis, code review, penetration testing, secure deployment configuration and secure patterns. Using this PESS method we authorize the Arduino and authenticate sensor data to store into cloud server database. Using PESS methodology, we model security adapter pattern using mitigation use case diagram and providing implementation support for this pattern during construction.

IV. PESS METHODOLOGY

We are using security patterns and security assurance activities like Threat analysis, Code review, Penetration Testing and security configurations at each phase to achieve security aware Software Development life cycle (Sa-SDLC) in our PESS methodology.

Process to enhance software security (PESS) method has the following steps:

1. Identify all the functional requirements and the security requirements of the application using Misuse case Techniques and analyze the security objective for each security requirement.
2. Identify security patterns for each security objective based on the security properties, application domain and Constraints and relate the precise context of security pattern to objective.
3. Design the application using UML standards, Model the security patterns using UMLsec modeling or any modeling technique to address the security concerns and perform threat analysis.
4. Derive the aspect test template to observe the internal processing using aspect oriented programming, point cut and advice pair, decision table. Test case template can be designed from pattern behavior and decision table using the six steps process in [24].
5. Develop the functional code and execute the derived test case based on Test Driven Development (TDD) and perform the code review. The concrete test aspects are generated from aspect test templates to validate the applied security design pattern in the implementation phase.
6. Test the developed product.
7. Deploy the product with security configurations.

In our work, we have identified and remediated the risk critical at interaction between the Aurdion to cloud server through smart gate way. The rest of the risks critical at other interactions are remediated in our further research.

V.PESS METHODOLOGY IMPLEMENTATION FORIOT HEALTH MONITOR

Step 1: The security requirement is "Patinet sensitive data is received by Arduino through sensors, then the Arduino sends patient sensitive data to temporary buffer. Before storing this data to cloud server database, needs to authorize the arduino using role class, then buffer will connect to the database. Then the database must authenticated by database class.

Step 2: Using STRIDE and DREAD we performed threat analysis. The risk is critical confirmed between arduino and Database. Security requirement for step 1, security adapter pattern is identified, this pattern is applied between Arduino and cloud server to preserve the security of adapted entities like Arduino and server database.

Step 3: To model the Security Adapter pattern, we have used the Mitigation usecase diagram (Integration of Usecase, Misusecase with Pattern Solution). Figure 2 shows the Mitigation use case to authorize Arduino using role class and authenticate the database using DB class before storing the patient data in the cloud server.

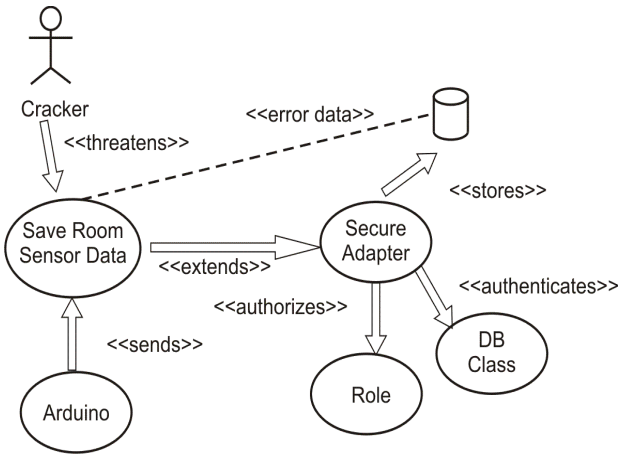


Figure 2: Security Adapter Pattern (Mitigation Use case Diagram)

Step 4: The security requirement in step 1, we derived the aspect test template and test case template with security adapter pattern solution is shown in the following sections.

The Aspect Test Template:

The aspect test case is derived from OCL description of Secure Adapter Pattern i.e Decision Table and pointcut-advice model we are creating to understand the internal processing of the pattern.

Decision Table for Arduino authentication:

		1	2	3	4
Conditions	Arduino Id agrees with Registered ID	**Yes**	**Yes**	**Yes**	**Yes**
	Arduino Data agrees with Secret Keys	**Yes**	**No**	**Yes**	**No**
Actions	Consider as Authorized Arduino	✓			
	Can receive data from Arduino	✓			
	Consider as Non-Authorized Arduino		X	X	X
	Cannot receive data from Arduino		X	X	X

Decision Table for Database authentication:

The database can be authenticated with multiple parameters, but in our work we have taken two parameters Database Id and unique name.

		1	2	3	4
Conditions	Database Id agrees with registered ID	**Yes**	**Yes**	**Yes**	**Yes**
	Database name agrees with registered database unique name	**Yes**	**No**	**Yes**	**No**
Actions	Send and store the data in database	✓			
	Do not send the data to database		X	X	X

Aspect oriented Pointcut-Advice Model:

The figures 3.(a) and 3.(b) shows the pointcut considering to authorize the arduino and advice to send data from arduino to buffer respectively.

```
pointcut Arduino_check():
call(* *.Secure_adpter.authorize(..);
```

Figure 3.(a). : Pointcut to authorize Ardunio

```
after () returning(Boolean right):
Data_send() { setTemporary (" Arduino_check",right);}
```

Figure 3.(b): Advice to send data to temporary buffer

The figures 4.(a) and 4.(b) shows the pointcut considering authenticating the database and advice to send data from buffer to database respectively.

```
Pointcut DB_check();
Call(* *.DB_authenicate(..);
```

Figure 4a: pointcut considering authenticating the database

```
after() returning(Boolean right):
Data_request_store(){ setTemporary("DB_check",right);}
```

Figure 4.(b).: advice to send data from buffer to database

The structure and behavior of the pattern is shown in figure 5a and 5.(b).

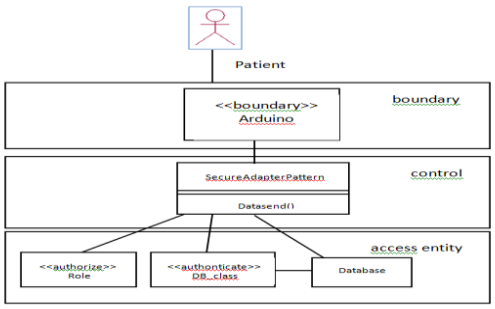
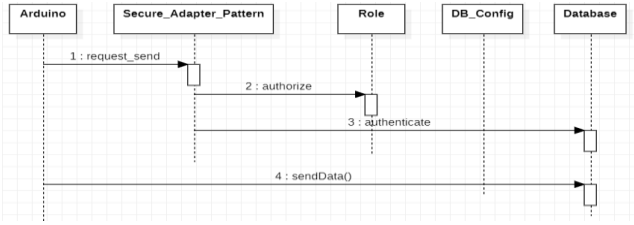


Figure 5.(a). Structure of the Interaction between Arduino to Database under control of secure adapter pattern.



5. (b). Behavior of the Secure Adapter Pattern

The secure adapter pattern preserves the security between the Arduino and Database.

Figure 6 shows the part of the derived test case, this authorize the arduino based on its id, role and data.

```
import org.junit.Test;

import static org.junit.Assert.assertEquals;

public class SecureAdapterCheck implements
RoleValidation

{

        ArudinoRepository repo;

        Arudino arudino = new Arudino();

        @Before

        public void setUp() throws Exception {

                repo = new ArudinoRepository();

                arudino.data = "adfadf adfadf adfasdfere";

                arudino.id = "1";

                arudino.role = Role.ROLE_ADMIN;

        }

@Override

        @Test

        public void authorize() {

        assertEquals(repo.arudinoRepo.contains(arudino),
true);

                saveDetails(arudino);}

                private void saveDetails(Arudino arudino) {

        }

}
```

Figure 6. Test case Part

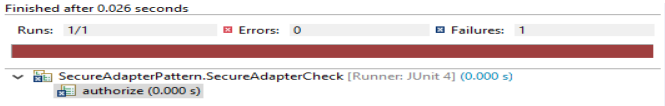
Figure 6. Test case is reusable; this will support the developers during implementation.

We have used the JUnit to test case of pattern. The derived test case is executed to validate the pattern before and after fix the vulnerability. Figure 7.(a),(b)shows the test results.

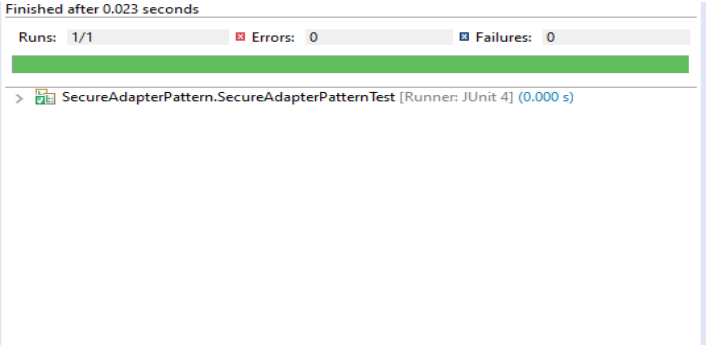
Figure 6: Part of the Test Case

In 7.(a) and 7.(b) shows the status of proper application of secure adapter pattern.

7.(a). Test Results (Secure Adapter Pattern)



7. (b). Re-Test Results (Secure Adapter Pattern)



Step 5: Developed and executed the IoT health Care application with the support of Secure Adapter Pattern. It is observed the interaction between the Arduino and Databse is secure and security is preserved between these components based on the test results.

The test results are shown in figure 8.

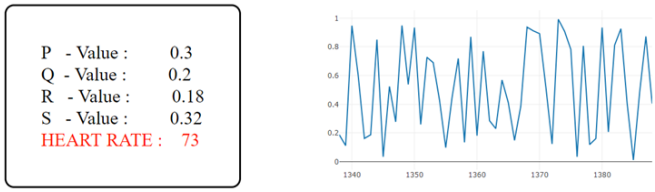


Figure 8: Data received from sensor and shown in Remote

Step 6: Testing of IoT health monitor based on 'test-a-user' approach than the application requirements. We performed the IoT tests such as Pilot test, IoT security, Database connectivity, Performance.

Step 7: In this phase, there may be low risk threats, simple remedies like "enable auto complete, cookie without secure flag, Httponly flag set with secure configuration can be applied.

VI. CONCLUSION AND FUTURE WORK

In our PESS methodology, we identified the security requirement; security pattern for the threat in IoT health application.

Using our method, we remediated the vulnerability with the support of secure pattern and security assurance activities. The vulnerabilities which is critical at the interaction point i.e between arduino and database is protected in the early phases.

We want to protect the remaining interaction points in IoT health application using security patterns. In our further work, we want to extend our work to measure the security for our methodology efforts.

REFERENCES

1. Gerd Kortuem, Fahim Kawsar, Vasughi Sundramoorthy, Daniel Fitton," Smart objects as building blocks for the Internet of things", Internet computing, volume: 14 , Issue: 1 , Jan.-Feb, IEEE, 2010.
2. Amir Djenna, Diamel Eddine Saïdouni, "Cyber Attacks Classification in IoT-Based-Healthcare Infrastructure" 2nd Cyber Security in Networking Conference (CSNet), IEEE, 2018.
3. Djamel EddineKouicem, Pr. Abdelmadjid Bouabdallah, Dr. Hicham Lakhlef ," Internet of things security: A top-down survey", Elsevier, 2018.
4. Aine MacDermott, Phillip Kendrick, Ibrahim Idowu, Mal Ashall, Qi Shi, "Securing Things in the Healthcare Internet of Things", IEEE,2019.

5. Jean Pierre Nzabahimana," Analysis of Security and Privacy Challenges in Internet of Things", The 9th IEEE International Conference on Dependable Systems, Services and Technologies, DESSERT'2018,24-27 May, IEEE, 2018, Kyiv, Ukraine.
6. Avelet Maria Fernandes, Anusha Pai" Secure SDLC for IoT Based Health Monitor" , Proceedings of the 2nd International conference on Electronics, Communication and Aerospace Technology ,ICECA, IEEE,2018.
7. Ishfaq Ali, Muhammad Asif," Applying Security Patterns for authorization of users in IoT Based Applications", International Conference on Engineering and Emerging Technologies, ICEET,IEEE,2018.
8. Wen-Tin Lee and Po-Jen Law, "A Case Study in Applying Security Design Patterns for IoT Software System", International Conference on Applied System Innovation (ICASI), IEEE,2017.
9. Seungyong Yoon, Jeongnyeo Kim," Remote security management server for IoT devices", International Conference on Information and Communication Technology Convergence, ICTC, IEEE, 2017.
10. Trusit Shah, S Venkatesan," Authentication of IoT Device and IoT Server Using Secure Vaults 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), 2018.
11. Eric Bodden," State of the Systems Security", 40th International Conference on Software Engineering: Companion Proceedings,ACM/IEEE, 2018.
12. Ezhil Kalaimannan, Jatinder N.D. Gupta," Th e Security Development Lifecycle in the Context of Accreditation Policies and Standards", IEEE Security and Privacy(Archive), volume 15 Issue 1, Jan,2017, Pages 52-57.
13. Michael Howard and Steve Lipner," The Security Development Lifecycle, SDL: A Process for Developing Demonstrably More Secure Software"2006.
14. Hironori Washizaki, "Security Patterns: Research Direction, Metamodel, Application and Verification", International Workshop on Big Data and Information Security (IWBIS),IEEE, 2017.
15. Y. Ito, H. Washizaki, M.Yoshizawa, Y. Fukazawa, T. Okubo, H. Kaiya, A. Hazeyama, N. Yoshioka, and E.B. Fernandez, "Systematic Mapping of Security Patterns Research," 22nd Conference on Pattern Languages of Programs (PLoP), 2015.
16. Takao Okubo, ,Haruhiko Kaiya, Nobukazu Yoshioka, "Effective Security Impact Analysis with Patterns for Software Enhancement", Sixth International Conference on Availability, Reliability and Security, IEEE,2011.
17. Annanda Rath, Bojan Spasic, Nick Boucart and Philippe Thiran," Security Pattern for Cloud SaaS: From System and Data Security to Privacy Case Study in AWS and Azure",MDPI,2019.
18. Masatoshi Yoshizawa , Takanori Kobashi , Hironori Washizaki , Yoshiaki Fukazawa ,Takao Okubo, Haruhiko Kaiya ,Nobukazu Yoshioka," Verifying Implementation of Security Design Patterns Using a Test Template", 9th International Conference on Availability, Reliability and Security,IEEE,2014.
19. Arnon Sturm, Jenny Abramov, Peretz Shoval," Validating and Implementing Security Patterns for Database Applications",Proceedings of the Third International Workshop on Software Patterns and Quality,GRACE-TR 2009.
20. Takanori Kobashi, Nobukazu Yoshioka, Takao Okubo," Validating Security Design Pattern Applications Using Model Testing", International Conference on Availability, Reliability and Security,IEEE, 2013.
21. Anika Behrens," What are Security Patterns? A Formal Model for Security and Design of Software", Proceedings of the 13th International Conference on Availability, Reliability and Security,ARES-2018, ACM ISBN 978-1-4503-6448-5/18/08, 2018, Hamburg, Germany.
22. Moosavi,Gia,Nigussie,Rahmani,Virtanen,Tenhunen, Isoaho" End-to-End security scheme for Mobility Enabled HealthCare Internet of Things, Journal of Future Generation Computer Systems, 2016.
23. Avelet Maria Fernandes, Anusha Pai," Secure SDLC for IoT Based Health Monitor", Proceedings of the 2nd International conference on Electronics, Communication and Aerospace Technology ICECA, IEEE,2018.
24. Masatoshi Yoshizawa, Hironori Washizaki, Yoshiaki Fukazawa, Takao Okubo, Haruhiko Kaiya, Nobukazu Yoshioka," Implementation Support of Security Design Patterns Using Test Templates", www.mdpi.com/journal/information,2016.

AUTHORS FROFILE

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.