International Journal of Advanced Research in Engineering and Technology (IJARET)
Volume 11, Issue 11, November 2020, pp. 2170-2177, Article ID: IJARET_11_11_215
Available online at http://iaeme.com/Home/issue/IJARET?Volume=11&Issue=11
ISSN Print: 0976-6480 and ISSN Online: 0976-6499
DOI: 10.34218/IJARET.11.11.2020.215

STUDY OF SOFTWARE SECURITY MEASURES ADAPTION IN SME'S IN BENGALURU

S G Gollagi
Research Scholar, East West Institute of Technology, Bengaluru and Faculty at Hirasugar Institute of Technology, Nidasoshi,
Affiliated to Visvesvaraya Technological University Belagavi, India

Dr. Narasimha Murthy M S
East West Institute of Technology, Bengaluru,
Affiliated to Visvesvaraya Technological University Belagavi, India

Aditya Pai H
K.S.Institute of Technology, Bengaluru,
Affiliated to Visvesvaraya Technological University Belagavi, India

Dr. Swathi K
K.S. Institute of Technology, Bengaluru,
Affiliated to Visvesvaraya Technological University Belagavi, India

Dr. Piyush Kumar Pareek
East West Institute of Technology, Bengaluru,
Affiliated to Visvesvaraya Technological University Belagavi, India

ABSTRACT

In recent years the number of cyber attacks has greatly increased, as has their complexity and effect. Therefore, new evolving software development models are needed, helping to build safe software by default. To achieve this, examining and comparing safe software development models in detail is particularly important. This paper provides a study of the most common secure software models and introduces a new secure software methodology tailored to all current environments. In this Research article experimental analysis is carried out based on the data collected from SME's in Bengaluru with help of qualtrics, It has been investigated the relationships between various variables considered for the study related to security aspects adopted in SME's . Sample size was 253 for the study.

Keywords: Secure, environment, SME's, relationships etc.

Cite this Article: S G Gollagi, Narasimha Murthy M S, Aditya Pai H, Swathi K, Piyush Kumar Pareek, S G Gollagi, Dr. Narasimha Murthy M S, Aditya Pai H, Dr. Swathi K, Dr. Piyush Kumar Pareek, *International Journal of Advanced Research in Engineering and Technology,* 11(11), 2020, pp. 2170-2177.
http://iaeme.com/Home/issue/IJARET?Volume=11&Issue=11

1. INTRODUCTION

A review of current systems, process models and standards defines four SDLC focus areas for safe software creation.

1. Protection engineering operations.

Security engineering tasks include activities to build a safe solution. Examples include elicitation and specification of security requirements; stable architecture based on safety standards, use of static analysis software, secure reviews and inspections, and secure testing. Other parts of the Create Security Website described engineering activities.

2. Activities in defense insurance.

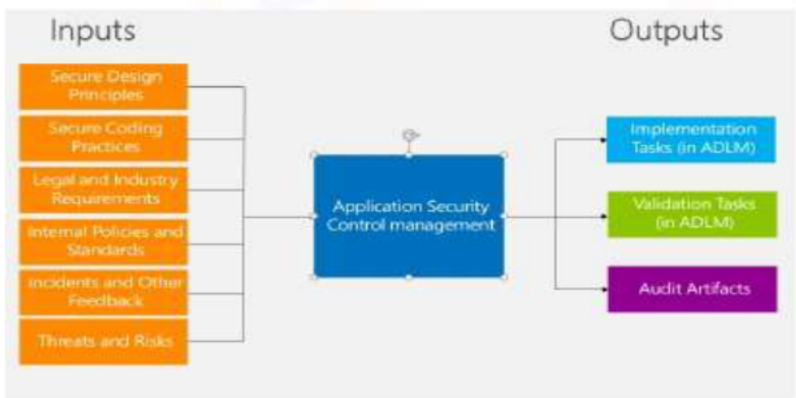
Assurance tasks include inspection, confirmation, expert analysis, artifact review, and assessment.

3. Organizational security and project management operations.

Organizational activities include organizational policies, sponsorship and supervision of senior management, organizational functions, and other organizational activities that promote security. Project management tasks include project preparation and resource distribution and use monitoring to ensure security engineering, security assurance, and risk detection activities are organized, handled, and monitored.

4. Identifying and managing security threats.

There is strong community agreement that detecting and managing security threats is one of the most critical tasks in a stable SDLC and is the catalyst for subsequent activities. In exchange, security threats guide other security engineering operations, project management and security assurance activities. Danger is also covered in other Create Security sites [3]



Application Security Control Management

Each security requirement identified should be tracked through implementation and verification. A best practice is to manage the controls as structured data in an Application Development Lifecycle Management (ADLM) system rather than in an unstructured document.

Source: Basic Practices for Secure Software Creation, Secure Development Lifecycle Program, Third Edition, March 2018, 2018 SAFECode

2. REVIEW OF LITERATURE

Mobile app security is a big problem facing mobile internet today. Traditional detection and security approaches cannot shield users from the introduction of a large number of malware, nor can they prevent hackers from decompiling and repackaging normal applications. This paper proposes a mobile device protection model that involves two key phases. First, static and dynamic analysis technologies are merged, and a dual-staining algorithm is proposed to capture application behaviors to enhance analysis accuracy. A strengthened mobile device platform is proposed based on encryption and custom loader. The proposed model can not only detect whether applications are malicious, but also prevent apps from being exploited by malicious developers, supporting mobile internet protection system development [1].

Tele health services provide remote care for elderly and physically inferior patients as well as remote surgery, treatment and diagnosis. To ensure the functionality of Tele- health systems, many structural properties (such as security) must be fulfilled. Although current studies address various security episodes involving Tele-health systems, it is difficult to provide a clear view on which security problems are most documented and which solutions have been proposed. Moreover because Tele-health systems consist of many software systems, it is unclear the critical areas of software engineering are important to the creation of safe Tele-health systems. This article reports a systematic mapping analysis (SMS) to identify, coordinate and classify security problems in Tele-health systems. Based on the SMS findings, we investigate how Software Engineering can help build safe Tele-health systems. From over 1,000 studies, we distinguished and listed 41 primary studies. Results show that I four security classifications (attacks, vulnerabilities, flaws, and threats) focus the most documented security issues; (ii) three security techniques (detecting attacks, preventing or minimizing attacks, and reacting to attacks) define security issues; and (iii) the most important research topics are vulnerable data transmission and privacy. Results of the SMS indicate that software architecture, specifications and models are key areas for developing stable Tele-health systems [2].

Software Defined Networking (SDN) expands current network capabilities by providing different features, such as customizable network controls. However, there are several security threat vectors in SDN, including current and emerging vectors resulting from new features that which impede SDN use. To tackle this issue, several countermeasures have been developed to mitigate various SDN threats. However, their efficacy must be evaluated and compared to fully understand how SDN's security posture shifts when the countermeasure is taken. It also becomes difficult to optimize SDN security without a systematic approach to assess SDN's security posture. In this paper we propose a new method to systematically model and evaluate SDN's security posture. We build a novel graphic security model formalism called the Threat Vector Hierarchical Attack Representation Model (TV-HARM), which offers a systematic approach to evaluating SDN risks, attacks and countermeasures. The TV-HARM captures numerous threats and combinations, allowing SDN's security risk assessment. We also identify three new security metrics for SDN security. Our experimental results showed that the proposed framework for security assessment can catch and analyze various security threats to SDN, demonstrating the applicability and feasibility of the proposed framework [3].

Smart contract protection is an evolving research field that addresses security concerns resulting from smart contract execution in a block chain system. Generally, a smart contract is a piece of executable code that runs automatically on the block chain to implement a pre-set agreement between the transaction parties. Smart contracts were implemented as revolutionary technology in different market areas such as digital asset exchange, supply chains, crowd funding, and intellectual property. Unfortunately, media reported many security problems in smart contracts, frequently leading to significant financial losses. These security issues pose new challenges to security study, as smart contract execution environment is focused on block chain computing and its decentralized nature of execution. To date, several partial solutions have been proposed to fix specific aspects of these security problems, and the trend is to develop new methods and tools to identify common vulnerabilities automatically. However, smart contract protection is systemic engineering to be explored from a global viewpoint, and a thorough analysis of smart contract security issues is urgently needed. We perform a literature review of smart contract protection from a software lifecycle perspective. We first examine key block chain features that can trigger security issues in smart contracts, then summaries common security vulnerabilities in smart contracts. To fix these vulnerabilities, we review recent developments in smart contract security, covering four development phases: 1) security design; 2) security implementation; 3) pre-deployment testing; and 4) monitoring and analysis. Finally, we summaries emerging problems and opportunities in block chain engineers and researchers' smart contract security [4].

3. RESEARCH METHODOLOGY

Table 1

Population Size	>1 Lakh
Sample Size	>200
Validity Test	Cronbach alpha Test
Hypothesis Testing	T test
Respondents	Software Engineers
Company Type	SME's
Geographical Location	Bengaluru
Missing Values	7.1%
Period of Data Collection	6 Months

3.1. Reliability Test

Table 2 Case Processing Summary for Reliability Test

Case Processing Summary			
		N	%
Cases	Valid	235	92.9
	Excluded[a]	18	7.1
	Total	253	100.0
a. List-wise deletion based on all variables in the procedure.			

Reliability Statistics	
Cronbach's Alpha	N of Items
.864	52

From Table 2, it can be concluded that the questionnaire data is reliable and valid and internal consistency of questionnaire is acceptable, coefficient of alpha being 0.864 for 52 numbers of items. Out of 253 respondents, it was found that 235 were valid and had no missing data, where as there were 18 cases which were excluded from the case processing summary.

Table 3 One – Sample T-Test statistics summary

One-Sample Statistics				
	N	Mean	Std. Deviation	Std. Error Mean
Depth of inheritance(DIT)	253	2.69	1.151	.072
Code complexity(Complexity)	253	2.90	1.017	.064
Weighted methods per class (WMC)	253	2.71	1.273	.080
Coupling between objects (CBO)	253	2.03	1.119	.070
File or Class size (LoC)	253	2.44	.832	.052
Lack of Cohesion of Methods (LCOM)	253	2.72	.920	.058
Number of previous Bugs	253	2.81	1.120	.070
Less number of planned test cases	253	3.26	.580	.036
Change in code	253	2.94	1.210	.076
Number of modified lines	253	3.62	.975	.061
Determining ownership (which is often unclear)	253	2.74	1.388	.087
More number of revisions(releases)	253	2.98	.857	.054
Uncover problems	253	3.51	1.006	.063
Uncovered Problem	253	2.42	1.083	.068
Less number of planned milestones	253	2.08	.825	.052
Less potential risk	235	1.79	.839	.055
Response from Messages (RFC)	253	2.98	.857	.054
Work flow	253	2.48	.824	.052
Unmovable development deadlines	253	3.51	.933	.059

One-Sample Test						
	Test Value = 0					
	t	df	Sig. (2-tailed)	Mean Difference	95% Confidence Interval of the Difference	
					Lower	Upper
Depth of inheritance(DIT)	37.192	252	.000	2.692	2.55	2.83
Code complexity(Complexity)	45.384	252	.000	2.901	2.78	3.03
Weighted methods per class (WMC)	33.831	252	.000	2.708	2.55	2.87
Coupling between objects (CBO)	28.869	252	.000	2.032	1.89	2.17
File or Class size (LoC)	46.642	252	.000	2.439	2.34	2.54
Lack of Cohesion of Methods (LCOM)	47.039	252	.000	2.719	2.61	2.83
Number of previous Bugs	39.957	252	.000	2.814	2.68	2.95
Less number of planned test cases	89.421	252	.000	3.261	3.19	3.33
Change in code	38.603	252	.000	2.937	2.79	3.09
Number of modified lines	59.158	252	.000	3.625	3.50	3.75
Determining ownership (which is often unclear)	31.352	252	.000	2.735	2.56	2.91
More number of revisions(releases)	55.340	252	.000	2.980	2.87	3.09
Uncover problems	55.410	252	.000	3.506	3.38	3.63
Uncovered Problem	35.575	252	.000	2.423	2.29	2.56
Less number of planned milestones	40.001	252	.000	2.075	1.97	2.18
Less potential risk	32.727	234	.000	1.791	1.68	1.90
Response from Messages (RFC)	55.340	252	.000	2.980	2.87	3.09
Work flow	47.906	252	.000	2.482	2.38	2.58
Unmovable development deadlines	59.786	252	.000	3.506	3.39	3.62

Hypothesis Testing

Statement 1: To understand the level of security measures followed in SME's by employees.

Null Hypothesis H0: There exist less procedures being followed in regard to security measures in SME's by employees.

Alternate Hypothesis H1: There exist procedures being followed in regard to security measures in SME's by employees.

With reference to Table 3, Consideration of the values of significance of coefficient of T-Test, it is found that there exists a stronger relationship between variables at 95% level of confidence and 5% standard error rate, It can be concluded that, there exist procedures being followed in regard to security measures in SME's by employees. Hence we accept the alternate hypothesis H1 based on the results obtained after T-Test.

4. CONCLUSION

A realistic experiment is tested in a software development company, considering data from actual software ventures. The findings are presented and compared in two development scenarios: a classic with a reactive protection approach, and another emerging and preventive one that applies security by default in all software life cycle phases. The total amount of vulnerabilities is decreased by 68.42 percent in the case study, reducing their criticality and temporal effect of their resolutions. Application protection and efficiency are enhanced methodologically with the proposed model, showing that the latest evolving solution offers more stable software

REFERENCES

[1] Chun Shan, Boyang Chen, Changzhen Hu, Jingfeng Xue and Ning Li, "Software defect prediction model based on LLE and SVM," *2014 Communications Security Conference (CSC 2014)*, Beijing, 2014, pp. 1-5, doi: 10.1049/cp.2014.0749.

[2] S. Moyo and E. Mnkandla, "A Novel Lightweight Solo Software Development Methodology With Optimum Security Practices," in *IEEE Access*, vol. 8, pp. 33735-33747, 2020, doi: 10.1109/ACCESS.2020.2971000.

[3] S. Siboni *et al.*, "Security Testbed for Internet-of-Things Devices," in *IEEE Transactions on Reliability*, vol. 68, no. 1, pp. 23-44, March 2019, doi: 10.1109/TR.2018.2864536.

[4] Xiaolin Zhao, Jingfeng Xue, Changzhen Hu, Rui Ma and Shanshan Zhang, "Research on software behavior modeling based on extended finite state automata," *2014 Communications Security Conference (CSC 2014)*, Beijing, 2014, pp. 1-5, doi: 10.1049/cp.2014.0744.

[5] Hang Dong, Chengze Li, Ting Li, Yuejin Du and Guoai Xu, "Research on the security model of mobile application," *2014 Communications Security Conference (CSC 2014)*, Beijing, 2014, pp. 1-5, doi: 10.1049/cp.2014.0751.

[6] G. Márquez, H. Astudillo and C. Taramasco, "Security in Telehealth Systems From a Software Engineering Viewpoint: A Systematic Mapping Study," in *IEEE Access*, vol. 8, pp. 10933-10950, 2020, doi: 10.1109/ACCESS.2020.2964988.

[7] T. Eom, J. B. Hong, S. An, J. S. Park and D. S. Kim, "A Systematic Approach to Threat Modeling and Security Analysis for Software Defined Networking," in *IEEE Access*, vol. 7, pp. 137432-137445, 2019, doi: 10.1109/ACCESS.2019.2940039.

[8] Y. Huang, Y. Bian, R. Li, J. L. Zhao and P. Shi, "Smart Contract Security: A Software Lifecycle Perspective," in *IEEE Access*, vol. 7, pp. 150184-150202, 2019, doi: 10.1109/ACCESS.2019.2946988.

[9] Cornish PL, Knowles SR, Marchesano R, Tam V, Shadowitz S, Juurlink DN, Etchells EE. Unintended medication discrepancies at the time of hospital admission. Arch Intern Med. 2005;165:424–9. [PubMed] [Google Scholar]

[10] Tam VC, Knowles SR, Cornish PL, Fine N, Marchesano R, Etchells EE. Frequency, type and clinical importance of medication history errors at admission to hospital: a systematic review. CMAJ. 2005;173:510–5. [PMC free article] [PubMed] [Google Scholar]

[11] Knudsen P, Herborg H, Mortensen AR, Knudsen M, Hellebek A. Preventing medication errors in community pharmacy: root-cause analysis of transcription errors. Qual Saf Health Care. 2007;16:285–90. [PMC free article] [PubMed] [Google Scholar]

[12] Reason JT, Carthey J, de Leval MR. Diagnosing 'vulnerable system syndrome': an essential prerequisite to effective risk management. Qual Health Care. 2001;10(Suppl. II):ii21–5. [PMC free article] [PubMed] [Google Scholar] Marz, N.; Warren, J. Big Data: Principles and Best Practices of Scalable Realtime Data Systems, 1st ed.; Manning Publications Co.: Greenwich, CT, USA, 2015. [Google Scholar].

[13] NIST Big Data Public Working Group. NIST Big Data Interoperability Framework: Volume 6, Reference Architecture. Available online: https://bigdatawg.nist.gov/_uploadfiles/NIST.SP.1500-6r1.pdf (accessed on 19 January 2019).

[14] Apache Software Foundation. Apache Mahout: Scalable Machine Learning and Data Mining. Available online: https://mahout.apache.org (accessed on 19 January 2019).

[15] Apache Software Foundation. Apache Spark. Available online: https://spark.apache.org (accessed on 19 January 2019).

[16] Pareek, P. K., Nandikolmath, T. V., & Gowda, P. (2012). FMEA Implementation in a Foundry in Bangalore to Improve Quality and Reliability. International Journal of Mechanical Engineering and Robotics Research, 1(2), 81-87.

[17] Nandikolmath, T., Pareek, P. K., & SA, V. K. (2012). "Implementation of a Lean model for carrying out value stream mapping in manufacturing industry", International Journal of Mechanical Engineering and Robotics Research, 1, no. 2, 88-95.

[18] Nandikolmath, T., Pareek, P. K., & SA, V. K. (2012). "Implementation of a Lean model for carrying out value stream mapping in manufacturing industry", International Journal of Mechanical Engineering and Robotics Research, 1, no. 2, 88-95.

[19] Dr. Piyush Kumar Pareek, ROC Structure Analysis of Lean Software Development in SME's Using Mathematical CHAID Model (May 17, 2019)

[20] S Vikas, GV Attimarad, P Pareek , Implementation of a Low-Cost and Non-invasive System for the Measurement and Detection of Faulty Streetlights , Journal of Advancement in Electronics Design 1 (1), 19-30

[21] Piyush Kumar Pareek , Questionnaire Survey using CHI-Square Test in Six Sigma in SME's in Bengaluru" , Advancement in Image Processing and Pattern Recognition 1 (1), 20-24

[22] Piyush Kumar Pareek , Analysing Tools in Six Sigma in SME's in Bengaluru , Journal of Advancement in Software Engineering and Testing 1 (1), 19-29

[23] Piyush Kumar Pareek, Six Sigma Approaches Used In Implementing In Supply Chain Management: A Review, Journal of Advancement in Software Engineering and Testing 1 (1), 14-18

[24] A Pai, VS Veesam, Piyush Kumar Pareek, BS Babu , Challenges in SME's ANOVA ANALYSIS PART-2 in Bengaluru , Research and Reviews: Advancement in Robotics 1 (1), 9-15

[25] A Pai, VS Veesam, BS Babu, Piyush Kumar Pareek, ANOVA Analysis Part One of Challenges in SME'S in Bengaluru , Research and Reviews: Advancement in Robotics 1 (1), 1-8

[26] A Pai, VS Veesam, BS Babu, Piyush Kumar Pareek, ANOVA Analysis Part One of Challenges in SME'S in Bengaluru , Research and Reviews:, Understanding the Adaptability of SME'S in Bengaluru , Research and Reviews: Advancement in Robotics 1 (1), 16-22

[27] D Chandramma Piyush Kumar Pareek, ANOVA Analysis Part One of Challenges in SME'S in Bengaluru , Research and Reviews:, Fast and Accurate Parts of Speech Tagging for Kannada-Telugu Pair , International Journal of Applied Engineering Research 13 (10), 7857-7867

[28] PK Pareek , An adoptive Model for lean software development in small and medium level firms in Bengaluru,2016

[29] KV Rao, R Balakrishna, HA Pai, Piyush Kumar Pareek , Data Mining for Healthy Tomorrow with the Implementation of Software Project Management Technique , Artificial Intelligence and Evolutionary Computations in Engineering Systems

[30] Piyush Kumar Pareek Dr. AN Nandakumar,'Lean software development Survey on Benefits and challenges in Agile and Lean usage in small and medium level firms in Bangalore' ,International Journal of Advanced Research in Computer Science and Software

[31] Piyush Kumar Pareek, TV Nandikolmath, P Gowda , FMEA implementation in a foundry in bangalore to improve quality and reliability , ,International Journal of Mechanical Engineering and Robotics Research

[32] Piyush Kumar Pareek ,Survey on Challenges in Devops , International Journal of Innovative Research in Computer Science

[33] A Pai, BN Ramesh, PK Pareek, P Prasoon , Failure Mode Effective Analysis for Software Processes

[34] Piyush Kumar Pareek , Identifying Wastes in software , , International Journal of Engineering Studies and Technical Approach

[35] PK Pareek, DAN Nandakumar Failure Mode Effective Analysis of Requirements Phase in small software Firms', Paper ID: ICSTM