Impact Factor (SJIF): 3.632

International Journal of Advance Research in Engineering, Science & Technology

e-ISSN: 2393-9877, p-ISSN: 2394-2444

(Special Issue for ITECE 2016)

Fault Injectioning: A Review

Kavan Dave[1], Bhavesh Tanawala[2], Hemant Vasava[3]

[1]Computer Engineering, Birla Vishwakarma Mahavidhyalaya
[2]Computer Engineering, Birla Vishwakarma Mahavidhyalaya
[3]Computer Engineering, Birla Vishwakarma Mahavidhyalaya

Abstract —Testing is one of the important phases of SDLC life cycle as it checks for fault tolerance & reliability for software. Websites & Web portals get infected by malicious Adware in real life scenario. So some specific mechanism is required to acknowledge this kind of problem in after deployment stages. One of the mechanisms to get rid of this situation is to use the web service fault injectioning. Fault Injection can be done in websites and resolution is provided by real time web services. Fault injection methods and tools can be used. Fault injection can be done at different layers. Appropriate method/tool and injection layer can be decided.

Keywords-fault injection; web service; software engineering; adware; testing;

I. INTRODUCTION

All Arising of Adware to and forth on the websites has been one of the burning problems faced by the users. These Adware cause many problems like redirecting network users, wastage of bandwidth, secretly installation of software on end users' machine. These Adware modules can add advertising links to browser bookmarks, change default settings, add advertising.

According to the statistical survey about Adware, almost half of Kaspersky security top 20 programs were occupied by Adware program [14]. Motive security labs surveyed that about 16 million mobile devices worldwide have been infected with malware 25percent more compared to 2013[15]. Another malware attack Dangerous.Object.Multi.Generic was the frequently attack on mobile phones in the 3rd quarter of 2015 with around 46.6% of all malware [16]. Almost each and every country users are affected with these malware but about quarter of all Bangladesh's users was affected in 2015 [17]. Trojan SMS named Malware is leading the field in damaging machine devices by 36% [18].
Talking about Android phone users then over 95% of mobile Malware was distributed on the Android platform in 2013 as Android phones are today's most wanted devices [19]. Financial Adware which steals credentials and banking information attacked more than 90 thousand Brazil users [20]. According to Kaspersky lab products survey, almost billion Malware attacked were found Android's open architecture has attracted 98.05% of known Malware [21]. Thus, some strong ideas should be implemented in order to remove Adware.

Fault injection is one the technique a tester due to check the fault tolerance of the software. He injects the faulty code into software. He injects the faulty code into software whether it is as per expectation or not. Fault injection can be done at compile time or runtime. It can be inserted at top, middle, bottom of the source code even it can be added in source, binary or assembly code. Thus, as per our requirement we perform injectioning and do testing

Through fault injectioning concept, we can check for Adware, arising now and then on end users machine considering Adware as fault generated on machine.

The rest of this paper is organized as the follows. Section 2 describes the basic terms that are useful to understand. In Section 3, the evaluation of the topic is described in brief. Section 4 and Section 5 provides a conclusion, and outlines a future work.

II. BACKGROUND THEORY & RELATED WORK

A. WEB SERVICE

The phrase "Web service" narrate a standardized way of integrating web-based applications using XML (Extensible Markup Language), SOAP (Simple Object Access Protocol), WSDL (Web Service Description Language) and UDDI (Universal Description, Discovery and Integration) open standard over standard internet. SOAP is used for exchanging information, WSDL for description and UDDI works as a registry for web services [8].

B. UNIT TESTING

Unit testing is a software development process in which the smallest testable parts of an application, called units, are individually and independently scrutinized for proper operation. Unit testing is often automated but it can also be done manually. [7]

C. FAULT INJECTION

Fault injection is a phrase covering a variety of techniques for inducing faults in systems to measure their response to those faults. Inserting the fault checks the reliability [6].

III. LITERATURE SURVEY

For Khaled Farj, Yuhui Chen and Neil A. Speirs in [1] have stated a fault injection toolkit- NetFIS (Network Fault Injector Service) that tests the performance and fault tolerance of SOAP based composite web services at design time. To implement this, virtualized WAN is achieved through physical LAN. It injects the faults at network and application level. When any request comes from client, communication fault is injected and passed to the Service provider and vice versa. After testing, more background faults are added so that more reliability is achieved. This toolkit has been experimented on client side. Here, no modification is required in the code so independent of hosting environment for portability.

Nik Looker and Jie Xu in [2] check the dependability of SOAP RPC based web service interface method calls and middleware as a whole. Here, fault injectioning is done at application and networking layer i.e before any encryption or signing takes place. It has no concern of whether the application runs on server or clients. Various initial experiments are done on networking layer and test scripts are designed and evaluated.

Mr. Zhang Xu and Dong Yan in [3] show how test cases are designed automatically at runtime. First they are parsed and stored in DOM structure. Here, test case generation involves four steps:- test data generation and test operation generation. Test cases are generated from four data types and operations are generated from dependency-input, output and input/output dependency. In the framework, Test case generator, controller, test agents and evaluator are included. Then these cases are describes into STS (Service Test Specification) file.

Mr. Lin Shan and Zhang Qun in [4] have discussed software implemented injection techniques, fault injectioning in web services and a tool named FIWI is designed to implement SOAP and RESTful services and check the robustness of web services.

K. Umadevi and S. Brintha Rajakumari in [5] stated various methods and tools for fault injection. The discussed methods test the coverage of the system. The fault can be injected during compile time or run-time. Various fault injection methods such as SWIFI, Interface error injection, Reflective programming, code Mutation, Assertion Violation and Perturbation Functions and different tools namely Jaca (tool written in java), DOCTOR (for distributed systems), Safe (for automatically generation of tests and injection in C/C++), Xception (injection in binary mode into C source code), Byteman (byte code injection for Java code) have been discussed in detail. Thus, according to our requirement we can use any of them.

IV. CONCLUSION

After reviewing all the papers and the web content related to FAULT INJECTIONING, we can conclude that the existing methods/ tool have been used at different layers to check the reliability but it still has room for improvement. Test cases can be generated automatically which can reduce the manual work
Still this fetched HTML can be further used for different purpose

V. ACKNOWLEDGEMENTS

It is a matter of great pleasure for me to get this opportunity of expressing my deep and sincere gratitude to those who has always helped me throughout my work. I heartily thank my guides Mr. Bhavesh A. Tanawala & Mr. H.D.Vasava for providing me their valuable guidance and constant support in my work. They guide me with their excellent knowledge in the area of "Software Engineering." I am also thankful to my friend Mr. Dhavalkumar H. Joshi for his constant cooperation, help and guidance throughout this journey. I would also like to thank all my colleagues who have helped me in some way or the other. I would like to thank my parents & family member for their love and support to build my moral during the work.

REFERENCES

[1] Khaled Farj, Yuhui Chen and Neil A. Speirs, "A Fault Injection Method for Testing Dependable Web Service Systems" *IEEE 2012 15th International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing (ISSORDC)-UK.*

[2] Nik Looker and Jie Xu, "Assessing the Dependability of SOAP RPC-Based Web Services by Fault Injection," *IEEE 2004 International Workshop on Object-Oriented Real-Time Dependable Systems (WORDS'03*) – UK.

[3] Xiaoying Bai, Wenli Dong, "WSDL-Based Automatic Test Case Generation for Web Services Testing", *IEEE 2005 International Workshop on Service-Oriented System Engineering (SOSE'05)*- Beijing, China.

[4] Marek Rychl´y, Martin Zouˇzelka, "Fault Injection for Web-Services," *IEEE 2012 IEEE 14th International Conference on Enterprise Information System* - Brno, Czech Republic.

[5] K. Umadevi and S. Brintha Rajakumari , "Educational resources metadata automatically extracted strategy study," *2015 International Journal of Innovative Research in Computer and Communication Engineering* (IJIRCCE) Vol. 3- Wuhan, China.

[6] *Fault injection* [Online]. Available: http://users.ece.cmu.edu/~koopman/des_s99/fault_injection/

[7] *Unit testing* [Online] Available: http://searchsoftwarequality.techtarget.com/definition/unit-testing

[8] *Web-Service*[Online] Available: http://www.webopedia.com/TERM/W/Web_Services.html

[9] *Fault Injection methods and tools:* [Online]
Available: http://www.ijircce.com/upload/2015/march/27_A_Review.pdf

[10] *Fault Injection* [Online]
Available: http://www.cs.umd.edu/~atif/Teaching/Fall2009/Jonathan.pdf

[11] Hsueh, M.-C., Tsai, T. K., & Iyer, R. K. (1997). Fault injection techniques and tools. Computer, 30, 75–82.

[12] Valenti, A. W., Maja, W. Y., Martins, E., Bessayah, F., & Cavalli, A. (2010, July). WSInject: A fault injection tool for web services (Tech. Rep. No. IC-10-22) Campinas: Institute of Computing, University of Campinas.

[13] S. Debnath, P. Mitra, and C. L. Giles, "Automatic extraction of informative blocks from web pages," The 2005 ACM symposium on Applied computing, 2005, pp. 1722-1726, doi: 10.1145/1066677.1067065

[14] Kaspersky security bulletin 2014 overall statistics [Online] Available: https://securelist.com/analysis/kaspersky-security-bulletin/68010/kaspersky-security-bulletin-2014-overall-statistics-for-2014/

[15] 16 Million Mobile Devices Infected with Malware [Online] Available: http://www.esecurityplanet.com/mobile-security/16-million-mobile-devices-infected-with-malware.html

[16] Share of malicious mobile programs of all attacks* on mobile devices in the 3rd quarter of 2015 [Online] Available: http://www.statista.com/statistics/325159/malicious-mobile-programs-2014/

[17] The Statistics Portal [Online] Available: http://www.statista.com/statistics/325201/countries-share-of-malicious-attacks-2014/

[18] The Statistics Portal [Online] Available: http://www.statista.com/statistics/325261/mobile-malware-distribution-by-behavior-type-worldwide-2013/

[19] The Statistics Portal [Online] Available: http://www.statista.com/statistics/325252/mobile-malware-distribution-worldwide-by-platform-2013/

[20] The Statistics Portal [Online] Available: http://www.statista.com/statistics/325226/countries-by-users-attacked-by-financial-malware/

[21] Secure list [Online] Available: https://securelist.com/analysis/kaspersky-security-bulletin/58265/kaspersky-security-bulletin-2013-overall-statistics-for-2013/