e-ISSN: 2349-9745
p-ISSN: 2393-8161

Scientific Journal Impact Factor (SJIF): 1.711

International Journal of Modern Trends in Engineering and Research

www.ijmter.com

ANALYSIS OF SOFTWARE SECURITY TESTING TECHNIQUES IN CLOUD COMPUTING

Krishnaveni S [1], Dr.Prabakaran[2], Sivamohan S[3]

[1]*Software Engineering,SRM University*

[2]*Computer Science and Engineering, SRM University*

[3]*Information Technology, SRM University*

Abstract-Cloud Security Testing is becoming a Popular field of Research Topic in Cloud Computing and Software Engineering. As the advance of cloud technology and services, more research work must be done to address the open issues and challenges in cloud security testing and More innovative testing techniques and solutions, Although there are many published papers discussing cloud Security testing, there is a lack of research papers addressing new issues, challenges, and needs in Software Security Testing. however, there is no clear methodology to follow in order to complete a cloud security testing. Since there is an increasing demand in Software usage there is more in for Software Security Testing. This paper presents an overview of Cloud Computing, Cloud security testing and comprehensive survey of security Testing Techniques and methods. from this we have identified problems in the current security testing techniques. This work has to presents a roadmap for new testers on the cloud with the necessary information to start their test.

Keywords-Cloud Computing ; Software Testing; Security Testing ; Vulnerability; Software Testing Techniques.

I. INTRODUCTION

Now a days, we have seen that the popularity of cloud computing is fastest growing trend in the field of Information Technology. As the advance of cloud technology and offers services, more companies take the decision to migrate their data to the cloud and uses the computing services. Cloud Computing resources such as computing power, storage, network and software are abstracted and provided as services on the Internet in a remotely accessible fashion. Categories of Cloud computing are Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). Also cloud can be deployed as Public, Private, Hybrid or Community. In cloud SaaS model, software applications or products have been marketed as 'on demand' business model. Many surveys have been conducted for independent users on SaaS adoption where the major concern for reluctance is the challenge of security. According to the IDC Cloud Services survey published the following IT cloud computing concerns: Security 87.5%, Availability 83.3%,Performance 82.9%, High Cost 81%,Vendor Lock In 80.2%.software security problems are becoming even more ruthless and excruciating. Many critical software applications and services need integrated security measures against malicious attacks. The purpose of security testing of these systems include identifying and removing software flaws that may potentially guide to security violations, and validating the effectiveness of security measures. Cloud security Testing is the solution to all these problems. Security testing is a great resource for identifying and rectifying vulnerabilities or flaws in applications so that they are less susceptible to compromise in the event of cyber attacks. Now a day's online transaction are rapidly increasing, so security testing on web application is one of the most important thing to be carried out while testing web applications.

II. RELATED WORK

Software Testing in cloud computing is very recent in the history of information technology ,Software security testing this paradigm is even more recent. Some papers addressed testing the cloud in variant ways. Gao, Bai, and Tsai give an importance to the needs of cloud testing. However, they provide no information about the methodology to follow in order to complete a cloud testing. On the other hand, Chan et al. focus on modeling a cloud application. They provide only one testing technique on the cloud that is based on cloud graph and its nodes. This technique is advanced and complex; the testers should be already familiar with the cloud testing paradigm in order to conduct this technique. In this paper, we discussed different techniques and tools used in cloud computing and presented a methodology in terms of a roadmap that helps the tester perform his tasks in the cloud in a simpler, more logical and more efficient way. Also, we made the comparison between software testing techniques and methods and also identified problems in the literature and investigated the typical scope for security testing techniques assessments with different deployment models of cloud computing. this comprehensive survey will help the Researchers in this field can benefit from the results in selecting their research direction and identifying new research opportunities for future work. we could conclude that testing over the cloud is more beneficial as it offers some useful features such as availability, visibility, automation and infrastructure independency at a reduced cost.

III. TESTING IN CLOUD – MOTIVATION

Testing in cloud will allow the users to mitigate the risks and errors when applications are deployed to the cloud. Besides, the use of cloud computing for testing means less costs and fewer expenditure. Now that testing the offerings of the cloud is compulsory, specific techniques, methods, and tools will need to be applied to this new type of testing. The traditional testing tools were not designed to test this complex and dynamic computing environment. An adaption of old techniques and tools needs to be performed in order to make these methods fit this different type of computing environment. At some point, new tools and methods should be introduced to test some specific offering of the cloud. In this paper, the focus is on security testing methodologies for software as a service (SaaS); testing cloud application on-demand.

Testing requires the existence of a test environment .establishing and managing a proper test environment is critical to the efficiency and effectiveness of testing.

Table:3.1 Test Environment in the Cloud

Attributes of Cloud solutions	Characteristics	Benefits
Advanced virtualization	Test resources (infra, tools and people) are pooled and virtualized	Providing efficient implementation of independent infrastructure
Rapid Provisioning	Test resources are provisioned on demand	Reducing test setup and execution time and eliminating errors
Service Catalog ordering	Test environment are readily available	Enabling visibility, control and automation
Elastic scaling	Test environment can be scaled up or down by large factor as the need emerges	Optimizes, infra and software license usage
Flexible pricing	Test resources are priced on supported topology and project phases	Offering pricing options tailored to user resource need
Metering and billing	Test resources used in reserved are charged back to LOBs	Prioritizing innovative projects
Maintenance of multiple test beds based for multiple release testing	Release based configurations (for testing) can be created and managed	Complete assurance on maintenance of product /service
Service virtualization	In a multi component architecture, availability of a dependant component(s) managed for testing	Effective completion of component level testing, despite the dependencies on Critical components
ALM (Application Lifecycle Management) TLM(Test Lifecycle Management) support	In a multi component architecture, availability of a dependant component(s) managed	Effective completion of component level testing, despite the dependencies on Critical components.
Capability	**Past**	**Present and future**
Server / Storage utilization	10-20%	70-90%
Cost Reduction	Nil	20-30%
Self Service	None	Unlimited
Test Provisioning	Weeks	Minutes
Change Management	Months	Days / Hours
Release Management	Weeks	Minutes
Metering / Billing	Fixed cost model	Granular
Re-platform ability (Compatibility Testing)	Prohibitively expensive	Engineering possibility with affordable cost
Maintenance of multiple test best beds for multiple release testing	Prohibitively expensive	Engineering possibility with affordable cost
Multiple Tools testing (Application Security for false positives – false negatives triangulations)	Prohibitively expensive	Affordable because of utility pricing and improved coverage
Test factories and TCoE Setup for clients	8-12 months	3-6 months

IV. OVERVIEW OF SOFTWARE SECURITY TESTING

Security testing is defined as the process of testing specialized towards security, where testing is the process of exercising the system to verify that it satisfies specified requirements and to detect errors. SaaS testing comprises of validating SaaS applications with respect to business workflows, multi-tenancy, integrity, reliability, ease of deployment, scalability, availability, accuracy, deployability, ease of use, testability, portability live updating. All these applications are tested with cloud based resources and among the testing criteria mentioned above the focus will be on three key components they are performance, compatibility and security. Security testing is a great resource for identifying and rectifying vulnerabilities or flaws in applications so that they are less susceptible to compromise in the event of cyber attacks. .

4.1. Objective of Software Security Testing

The objectives of Software Security Testing are threefold:

- To verify that the software's dependable operation continues even under hostile conditions, such as receipt of attack-patterned input, and intentional (attack-induced) failures in environment components;
- To verify the software's trustworthiness, in terms of its consistently safe behavior and state changes, and its lack of exploitable flaws and weaknesses; and
- To verify the software's survivability, by verifying that its anomaly, error, and exception handling can recognize and safely handle all anticipated security-relevant exceptions and failures, errors, and anomalies; this means minimizing the extent and damage impact that may result from intentional (attack-induced) failures in the software itself, and preventing the emergence of new vulnerabilities, unsafe state changes etc.

4.2. Six Attributes Of Security Testing

Authentication: is a security measure designed to establish the validity of a transmission, message, or originator, or a means of verifying an individual's authorization to receive specific categories of information, Authorization: provides access privileges granted to a user, program, or process, Confidentiality is the assurance that information is not disclosed to unauthorized individuals, processes, or devices, Availability guarantees timely, reliable access to data and information services for authorized users, Integrity: is provided when data is unchanged from its source and has not been accidentally or maliciously modified, altered, or destroyed , Non-repudiation: is the assurance that none of the partners taking part in a transaction can later deny of having participated.

4.3.Key Terms Used in Security Testing

Denial of Service (DoS) attacks on SaaS: In a multitenant environment, the DoS testing have become even more critical as a DoS attack on one tenant (which may utilize 100% CPU or other resources) may cause other tenants to cease working, SQL Injection: This is code injection technique through the web application. It Causing undesired SQL queries to be run on your database, Packet Sniffing: is a Listening to traffic sent on a network. Many internet protocols (http, aim, email) are unsecure, Password Cracking: In security testing of a web application Password cracking programs can be used to identify weak passwords. Hidden Form Parameters: None of the SaaS application developers should include hidden form fields. Cookie Values: Security Testing should ensure that data in the cookies is encrypted with strong encryption algorithm and limited sensitive IAM information is being sent out as cookies. Vulnerability scanning is the best technique to perform this testing. XSS (Cross Site Scripting): It is a type of injection which is typically found in web applications. SaaS are susceptible because they share application access and data among various tenants. Vulnerability scanning and risk based testing can be used to verify whether SaaS offering is susceptible to XSS. Vulnerability: The Vulnerability is a weakness in a system under test which may cause the malicious

attaches by unauthorized users. URL Manipulation: Also known as URL rewriting, is the process of modifying parameters. Websites communicate with servers for sharing information to client (browser).

V. TESTING TECHNIQUES

The cloud testing methodology is the set of techniques, tools and processes to be followed while undergoing tests for cloud services . Some of these methods and techniques will be an adaptation of conventional techniques, and others were specially developed to fit the testing needs of cloud services. While dealing with cloud computing application testing, it is necessary to take into consideration the background .We have reviewed many articles on security testing techniques and brief here. Basically in software engineering the:

- Code reviews
- Fuzz testing
- Source code fault injection
- Risk analysis
- Vulnerability scanning
- Penetration testing

5.1. Code Review

Source code review also known as static analysis is the process of manually checking source code for security weaknesses. Many serious security vulnerabilities cannot be detected with any other form of analysis or testing. Most security experts agree that there is no substitute for actually looking at code for detecting subtle vulnerabilities. With the source code, a tester can accurately determine what is happening (or is supposed to be happening) and remove the guess work of black box testing. Source code analysis can also be extremely efficient to find implementation issues such as sections of the code where input validation was not performed or where fail open control procedures may be present. Operational procedures need to be reviewed as well, since the source code being deployed might not be the same as the one being analyzed.

5.2. Source Code Fault Injection

Source code fault injection is a testing technique originated by the software safety community. It is used to induce stress in the software, create interoperability problems among components, simulate faults in the execution environment, and thereby reveal safety-threatening faults that are not made apparent by traditional testing techniques. Security fault injection extends standard fault injection by adding error injection, thus enabling testers to analyze the security of the behaviours and state changes that result in the software when it is exposed to various perturbations of its environment data. These data perturbations are intended to simulate the types of faults that would result during unintentional user errors as well as intentional attacks on the software *via* its environment, as well as attacks on the environment itself.

5.3. Fuzz Testing

Fuzz testing inputs random invalid data (usually produced by modifying valid input) to the software under test *via* its environment or *via* another software component. The term *fuzzing* is derived from the fuzz utility which is a random character generator for testing applications by injecting random data at their interfaces. In this narrow sense, fuzzing means injecting noise at program interfaces. Fuzz testing is implemented by a program or script that submits a combination of inputs to the software to reveal how that software responds. The idea is to look for interesting program behavior

that results from noise injection and may indicate the presence of a vulnerability or other software fault.

5.4.Vulnerability Scanning

Automated vulnerability scanning is supported for application level software, as well as for Web servers, database management systems, and some operating systems. Application vulnerability scanners can be useful for software security testing. These tools scan the executing application software for input and output of known patterns that are associated with known vulnerabilities. These vulnerability patterns, or "signatures", are comparable to the signatures searched for by virus scanners, or the "dangerous coding constructs" searched for by automated source code scanner, making the vulnerability scanner, in essence, an automated pattern-matching tool.

5.5. Risk Analysis

To review security requirements and to identify security risks, risk analysis is carried out during the design phase of development. Threat modeling is a methodical process that is used to identify threats and vulnerabilities in software. It helps system designers to analyze and think about the security threats that their system might face. Therefore, threat modeling is carried out as risk assessment for software development. In fact, it enables the designer to develop mitigation strategies for potential vulnerabilities and helps them focus their limited resources and attention on the parts of the system most at risk.

5.6. Penetration Testing

Penetration testing, also known as ethical hacking, is a common technique for testing network security. While penetration testing has proven to be effective in network security, the technique does not naturally translate to applications. Penetration testing is, for the purposes of this guide, the "art" of testing a running application in its "live" execution environment to find security vulnerabilities. Penetration testing observes whether the system resists attacks successfully, and how it behaves when it cannot resist an attack.

Table:5.2 Software Security Testing Techniques

SECURITY TESTING TECHNIQUES	DESCRIPTION	APPLICATION
Fault Injection based Testing	Fault injection focuses on interface or boundaries on the environment in which the application is running, the testing includes verification for all input fields, networks interface,file system and environment variables etc.	It determines hoe the application handles different types of protocol packets. in order to identify security vulnerabilities,fault data should be injected.SDLC Stages: Build,unit testing and integration testing.
Fuzzy Testing	Mechanism of injecting random data in to application to determine whether it can run normally under the jumbled input	Helps in discovering security vulnerabilities.it is very effective mode of testing and may even find flaws of tested software,which are difficult for the other logical testing method.SDLC stages: Unit Testing,integration testing and system testing.
Vulnerability scanning Testing	It includes:testing space scanning,running the application to determine leakage that the application might have created,for eg. network port scanning may identify open ports set of vulnerabilities .	SDLC stages: unit testing,integration testing and system testing
White box Testing	Static code review and walkthrough are common tools used here	Good at finding security bug such as buffer overflow SDLC stages:Build - coding,code review stage.
Risk based Testing	Some research has been done for combining risk analysis and security testing with software development life cycle on early adoption in recommended to find high risk security vulnerabilities	SDLC stages: this approach emphasized SDL(security development life cycle)for eg. creating security abuse use cases(requirement), risk analysis(pre design and design stage),static analysis tools(code) and penetration testing

VI. CONCLUSION AND FUTURE WORK

Cloud software security testing is becoming a popular research fields in the future. Now a days software testing techniques are being adapted for the cloud computing. As the advance of cloud technology and testing as services, more research work must be done to address the open issues and challenges in cloud security testing . Although there are many published papers discussing cloud Security testing, there is a lack of research papers addressing new issues, challenges, and needs in Software Security Testing. however, there is no clear methodology to follow in order to complete a

cloud security testing . We have made a comprehensive survey of security Testing Techniques and methods. from this we have identified problems in the current security testing techniques. Researchers in this field can benefit from the results in selecting their research direction and identifying new research opportunities for future work.

REFERENCES

[1]McGraw G.,"Software Penetration Testing",IEEE computer society,2005.
[2] Braz C., and Robert J., "Security and Usability: The Case of the user Authentication Methods ," IHM Montréal, Centre-ville Montreal, Canada, 2006. P: 220-238
[3]Arilo C. Dias Neto Rajesh Subramanyan2, Marlon Viera2 GuilhemeH. Travassos, "A survey on Model-based testing Approaches: A systematic Review", Federal University of Rio de janerio-COPPE, 2 Siemens Corporate Research – SCR,2007. P: 387–394.
[4]Barnett J R., and Irwin B., "Toward a Taxonomy of Network scanning Techniques," Security and network research group (SNRG)Department of computer science,2008.P: 135-142
[5]McgrawG.,"Software Security", The IEEE computer society IEEE security & privacy,2004.
[6]Gupta D., Chatterjee K., and Jaiswal S.,"A Framework for Security Testing," Springer,2013. P: 112-120
[7]Bayuk J.and Ostashari A,Measuring system security,Wiley Online Library,2011.p.110-128
[8]Tian-yang G., Yin-sheng S., and You-yuan F., "Research on software security testing", World Academy of science, engineering and Technology, 2010. p.120-128
[9]Antunes N., and Vieira M., "Security Testing in SOAs: Techniques and Tools," Innovative Technologies for Dependable OTS-Based Critical Systems,2013. pp.75-81
[10] Tondel Inger A, Jaatun Gilje M, and Jensen J,"learning from software security testing" IEEE International Conference on software Testing Verification and Validation Workshop, 2012.
[11] A.Oladimeji, E. Supakkul S., and Chung L., "Security thread modelling and analysis: A Goal oriented approach", 2006.
[12]Baadshaug E.T., Erdogan G., and Meland P.H.," Security Modelling and tool support advantages ", Baadshaug E.T., Erdogan G., and Meland P.H., International Conference on availability, reliability and security, 2010 P : 86-91
[13]Tondel, I.A., Jaatun, M.G, Meland, P.H., "Security Requirements for the rest of us: A survey",Software IEEE,2008.
[14] Mouratidis H.,Giorgini P., Mansoni G, "When security meets software engineering : a case of modelling secure information system" , Elsevei Science ltd, Oxford,UK, 2005.
[15] Kitchenham.B., "Procedures for performing systematic reviews " Keele university technical report and NICTA technical report, 2004.
[16]Jiong Y., "Survey of Model –Based Software Testing", Computer science, 2004.
[17] G Kavitha Jayaraman, Incorporating Security in Software Testing Life Cycle, Cognizant Technology Solutions, 2009.
[18] Risk-Based Software Security Testing, Software Assurance Pocket Guide Series: Development, Volume III, Version 0.5, September 1, 2009. P.220-235
[19] Bruce Potter & Gary McGraw,Software Security Testing, IEEE Security & Privacy, 2004, pp. 32-36.
[20] Sven Turpe, „Security Testing: Turning Practice into Theory, IEEE International Conference on Software Testing, Verification and Validation Workshop (ICSTW08), IEEE Computer Society, 2008.

