Review Article
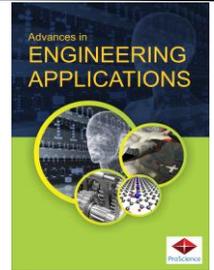
Advances in Engineering Applications

Journal Home Page: www.proscience-journals.com

ProScience

Survey on Software Security Testing Techniques in Cloud Computing

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

[1] SWE, SRM University, India
[2] Department of computer science Engineering, SRM University, India
[3] Department of Information Technology, SRM University, India

Abstract

Cloud Security Testing is becoming a Popular 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.

1.0 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 use 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).

*Corresponding author: krishnaveni.s@ktr.srmuniv.ac.in

31

Krishnaveni et al. Adv.in Eng.Appl.Vol.1 (2018)31-39

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% and 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.

2.0 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. However, they provide no information about the methodology to follow in order to complete a cloud testing. 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 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 security 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.

3.0 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.

4. 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. Software testing comprises of validating Software 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 behaviour 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.

Table: 3.1 Test environment in the Cloud

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

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

33

Krishnaveni et al. Adv.in Eng.Appl.Vol.1 (2018)31-39

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.

4.4. Classification of Security Testing

Security testing can be classified as security Testing techniques and Security Testing methods. There are many security testing techniques which exist and can be used to assess the security level of systems and software, are grouped into the following three categories:

1.Review Techniques. These e techniques are used to evaluate systems and software in order to discover vulnerabilities. These techniques are generally performed manually.

2.Target Identification and Analysis Techniques. These testing techniques are used to identify services, and risky vulnerabilities.
3.Target Vulnerability Validation Techniques. These testing techniques identifies the presence of vulnerabilities depending on the specific technique being used.

5. 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; Vulnerability scanning; Fuzz testing; Risk analysis; Source code fault injection; 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.

Table 4.2: Depicts its classification of Security Testing Techniques and Security Testing Methods

35

Krishnaveni et al. Adv.in Eng.Appl.Vol.1 (2018)31-39

	Formal security Testing	Model-based Security Testing	Fault Injection based Testing	Fuzzy Testing	Vulnerability Scanning Testing	Property-based Testing	White box-based Security Testing	Risk-based Security Testing
M E T H O D S	Formal security testing methods are basically theorem proving and model checking methods. **Limitations**: Need Quality Expert practical implementation states are required Less Efficient	MBST is a relatively new field and especially dedicated to the systematic and efficient specification and documentation of security test objectives, security test cases and test suites, automated or semi-automated generation.	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.	Mechanism of injecting random data in to application to determine whether it can run normally under the jumbled input	Vulnerability scanning is bit similar to the concept of a port scanning to some extent. A vulnerability scanner as similar to port scanner identifies hosts and open ports	Property based testing validates that the software's implemented functionality satisfies its specifications. it does this by examining security relevant properties revealed by the source code	To evaluate information directly from the source code without executing it. It can find security bug, such as buffer overflow. design issue causing the vulnerability can be identified.	Risk-based security testing and the risk analysis, security testing with software development life cycle is combined, as early as possible to risky security vulnerabilities

	Review Techniques		Target Identification and Analysis Techniques			Target Vulnerability Validation Techniques		
T E C H N I Q U E S	**Log Review:** Log review is used to detect suspicious activities as soon as possible in order to minimize the impact of incidence. Log review and analysis can also provide a dynamic picture of present system activities which can be compared with security policy. **Documentation Review:** Document review is a way of collecting and analyzing data by reviewing existing documents. **File Integrity Checking:** Identify certain forms of unwanted files, Verifying File Integrity against corruption using Checksum. **Rule set Review:** helps in revealing holes in rule set-based security control, establishes a database of file checksums , It identifies unauthorized changes to important files and certain forms of unwanted files. **System Configuration Review:** System configuration review is the process that evaluates the strength of system configuration. **Network Sniffing:** Network sniffing refers to a utility that examine data packets on a network to gather data from a packet or to identify the content of packet, Capture and Analyze all of the Network traffic.		**Network Discovery**: It used to discover the active devices, process of identifying edges and non-edges of unknown devices on a network. **Vulnerability Scanning:** Identify the known vulnerabilities, It has high false positive rates. **Wireless Scanning :**Identifies unauthorized Wireless devices with in the ranges of scanners **Bluetooth Scanning:** It is a method to test the security of devices in a limited range. It can provide ample information about discoverable devices. The security mode in which Bluetooth device is working can be used to evaluate their security. **WAR Dialing:** Using a Modem to dial a list of Telephone Numbers, Searching for Computers, Bulletin Board System and Fax Machines.			**Password Cracking:** Identifies weak password and password policies, Collecting Passwords from the Stored or Transmitted Data .Using Brute Force and Dictionary Attacks. Quickly identify weak password Provides clear demonstration of password strength or weakness, Easily implemented, Low cost. **Penetration Testing**: Simulating Attack from a Malicious Source, Includes Network Scanning and Vulnerability Scanning, Simulates Attack from someone Unfamiliar with the System, Simulates Attack by having access to Source Code, Network, Passwords. **Social Engineering:** Describe the acquisition of sensitive information or inappropriate access privileges by an outsider, it allows testing of both procedures and human element. **Remote Access Testing:** Identifies vulnerabilities on remote access.		

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 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.

6. Review of Current Software Security Testing Scenario

Software security testing is an important method of ensuring software security. Therefore, security testing research has become very important and practical, which is still in its infancy. There are works cited in literatures on software security testing. Researchers and practitioners are advocating for traditional software security testing which plays more attention to security functions based on software requirement. Unfortunately, traditional security testing cannot ensure software security effectively. Most of the testers are unfamiliar with application security testing, and generally executing penetration test at the end of the development life cycle. Vulnerabilities found latter requires a lot of rework for mitigation. Every approach has its own advantage.

But unfortunately, the contributions made in the direction lacks with the realistic problems and are not very efficiently implemented in real world. Some of the pertinent issues involved with most of the approaches are as follows:

- They do not help to test for the issues that commonly appear in software and are not easily fixed by using a more programmer friendly platform;
- Most of the techniques fail to continue under the adverse conditions often encountered in security testing;
- Generally, the approaches available are not very robust and do not work well with incomplete information about the target of testing.
- Most of the technique do not provide guidelines to their users throughout the testing task;
- The result obtained on implementing approaches are not very useful under real world conditions;
- Most of the approach do not provide useful abstractions;
- The current state-of-art fails to address the issue raised above and solutions to all the above research challenges. There is an urgent need to narrow the gap and create common ground for the integration of security issues into the development process.

37

Krishnaveni et al. Adv.in Eng.Appl.Vol.1 (2018)31-39

Table 6.4: Literature study of Current Software Security Testing

YEAR	AUTHOR	DESCRIPTION	TECHNIQUE ADOPTED	REFERENCE
2004	G. McGraw	The complete background details of software security came into view, explained that in software engineering evolving best practices security of software should be taken into consideration.	Network port scanning technique in order to check for behaviour of complex portion of the software system	R[5] The IEEE computer society IEEE security & privacy
2005	G. McGraw Mouratidis .,Giorgini , Mansoni G	They work on analyzing the responses and try to confirm the existence of particular vulnerability. Black box testing tools work similar to pen-test tools.	Black box testing tools were introduced	R[1] The IEEE computer society,Elsevier Science ltd, Oxford,UK, 2005.
2006	C. Braz et al	They provided their idea on enhancing the efficiency of security systems. As Knowledge based authentication work out to be efficient in minimizing the effect of unauthorized access.	The usability and user acceptance of security systems had been become a major issue in research on the efficiency	R[2] IHM Montréal, Centre-ville Montreal, Canada.
2007	A C. Dias Neto et al	A systematic review was performed on model based security testing approaches. includes representation models, support tools, test coverage criteria, the level of automation, intermediate models, and the complexity	Model based security testing (MBST)	R[3] ", Federal University of Rio de janerio-COPPE, 2 Siemens Corporate Research – SCR
2008	R J Barnett et al. Sven Turpe	A review of security testing literature is performed ,It discusses about security, vulnerability and used better tools and techniques	Fuzz testing is Risk based security testing	R[4], Security and network research group (SNRG) Department of computer science, IEEE Computer Society
2009	Hao Chen and Jean-Pierre Corriveau	The aim of study is suggesting to testers how to design security testing and identify potential vulnerabilities in current online banking system	Risk-Based Security Testing	R[7] Software Assurance Pocket Guide Series: Development, Volume III, Version 0.5,
2010	E.T. Baadshaug et al.,	An overview of some approaches to graphical security modeling has been given and presented	Graphical security modeling tools	R[12] International Conference on availability, reliability and security.
2010	G Tian yang	The definition in addition to classification of software security testing was discussed and methods and tools of software security testing have been investigated.	Fault Injection based Testing, Risk and Fuzz testing	R[8] World Academy of science, engineering and Technology.
2011	J. Bayuk et al	Security metrics have evolved side by side with the advent of security tools and techniques.	Security testing Metrics , risk analysis for risk based testing	R[7]Measuring system security, "Wiley Online Library, 2011
2012	Aaron Mar back et el	Proposed a threat model based security testing approach that automatically generates security test sequences from threat trees.	MBST includes e.g. security functional testing, model-based fuzzing, risk- and threat-oriented testing, and the usage of security test patterns, A project "European ITEA2-project DIAMONDS" is helpful to develop tools and methods based on MBST	R[10] , IEEE International Conference on software Testing Verification and Validation Workshop,

		technique.		
2013	D.Gupta et al.	Discussed about the how the data can be secured form unintended loss or from malicious hackers .It presents key concepts on security, also providing the basis for understanding existing challenges on developing and deploying secure software systems.	Vulnerability scanning testing and Fault Injection based Testing	R[6]A Framework for Security Testing," Springer
2014	Radhika Batra et al	This paper gives a brief review of cloud based testing techniques. It includes what is cloud computing ,all its major techniques ,being some major challenges in the cloud testing	Penetration testing, verification and validation	R[26] International Journal of Computer Science and Mobile Computing, Vol.3 Issue.6, June-2014, pg. 314-319©2014

A recent survey has revealed that the least secure software generally carries six times higher business risk than the most secure software product. The survey highlighted the fact that security quality of software product can vary drastically depending on its design and implementation. Since decades, vulnerable software is invaded and modified making it cause damage to the healthy software. These damages are replicated across networks to cause dangers to the entire society. The main motive behind survey is that the techniques and methods that are discussed in past reflect their suitability for the present need and are considered to be useful to fulfill the needs of the future perceptivity. We have gone through the research from 2004-2014.

Conclusion and future work

Cloud software security testing is becoming a popular research fields in the future. Now a day's 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] Mcgraw G., "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 MOstashari 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.

39

Krishnaveni et al. Adv.in Eng.Appl.Vol.1 (2018)31-39

[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.

[21] Huang Song,Wang Liang, Zheng Changyou & Hong,A Software Security Testing Method Based on Typical Defects, International Conference on Computer Application and System Modeling (ICCASM 2010), IEEE Computer Society, 2010, pp.V5-150-153.

[22] SOA Test Methodology, Torry Harris Business Sollution, www.thbs.com/soa.

[23] Aaron Marback, Hyunsook Do, Ke He, Samuel Kondamarri and Dianxiang Xu, A Threat Model based Approach to Security Testing, Software Pract. Exper. (2012), Published Online in Wiley Online Library, DOI: 10.1002/spe.2111.

[24] Gu Tian-yang, Shi Yin-sheng & Fang You-uan, „Research on Software Security Testing, World Academy of Science, Engineering and Technology 69 2010, pp. 647-651.

[25] Dr.SacharPaithis, The Future of Software Security, www.testingexperience.com, P: 667-672

[26 Radhika Batraet al,International Journal of Computer Science and Mobile Computing, Vol.3 Issue.6, June-2014, pg. 314-319©2014

[27] Neha Mehrotra (2011), "Cloud Testing Vs Testing a Cloud", *Infosys Viewpoint*.

[28] Vinaya Kumar Mylavarapu (2011), "Taking Testing to the Cloud", *Cognizant Reports*.

.