# Design and Analysis Model of Testing as a Service for Cloud Computing

<sup>1</sup>K. Venkat Raman, <sup>2</sup>Dr G. Ranjith, <sup>3</sup>Dr.A.Subash

<sup>1</sup>Research Scholar, <sup>2</sup>Associate Professor, <sup>3</sup>Assistant Professor
 <sup>1,2,3</sup> Department of Computer Science and Engineering,
 <sup>1</sup>Allahabad Central University, Allahabad, Uttar Pradesh, India
 <sup>2</sup>Warangal Institute of Technology and Science, Warangal, Telangana, India.
 <sup>1</sup>VNR Vignana Jyothi Institute of Engineering and Technology, Hyderabad, Telangana, India.

*Abstract:*Cloud Computing is popular for its characteristics and Essentials, majorly it is used for storage, where the personal storage as services such as drop box and one drive attracting towards the cloud. The designing and analyzing of TasS, that is testing as a service is an automated service, where the research is to be carried out in the domain of testing especially when run in cloud is discussed along with the practical implementation. We majorly focused on different types of testing technologies such as testing major enterprise resource planning and monitoring and testing cloud based applications, As the scope of the application testing in all areas is focused, as it is to be noted that all applications cannot be tested in cloud and all testing techniques are not be appropriate, to be tested for applications running in cloud. The case study is done on different testing methods and methodologies such as security testing, performance testing and software testing. We even focused on the most common security characteristics such as confidentiality, Data Integrity, Availability, Non-Reputation and Authentication. The Genetic algorithmic approach is given for the wide scope and effective testing in cloud. We even concentrated on different testing domains in cloud, carried out acceptable to achieve effectiveness of testing in cloud.

*IndexTerms* - Testing as a Service, Storage as a Service, Performance, Measurement, authentication.

#### I. INTRODUCTION

Software Testing systems become more and more multifaceted and complicated due to multitasking applications in different domains which made more complex. Traditional and manual techniques of software testing come across with some complex situations, such as high-test cost, long test time and difficult test implements with limited resources. In cloud computing, user's data is stored in one or more remote server(s) and at different data centres all over the world, a user will not have any idea were his confidential data is stored, which poses more security challenges for the system. One of the most important concerns is to protect user's sensitive information from other users and hackers that may cause data leakage in cloud storage, for data protection in cloud computing penetration testing plays a major role in identifying vulnerabilities in cloud and to protect cloud from attacks and intruders. Cloud Testing includes scalability, performance and security, different testing tools are embedded with the cloud, so picking the right testing tools at each stage plays a great role in locating and recognizing exploitable defects. cloud testing methods includes remote access testing, recovery testing and Integration Testing for improvement of performance and reliability as many organizations are moving to Cloud. Software testing is an evaluation process or series of process, to show that a testing program performs its intended functions correctly, it is designed to make sure that what the computer code does what it was designed to do, the main objective of Testing is the process of executing a series of programs with the intent of finding errors. Testing as a service is a model that works as multiple user interface available around the world, around the clock with test infrastructure having web as an interface based on cloud computing. The goal of cloud based application is the process to establish confidence that a program does what it is supposed to do. The testing methodologies and mechanisms such as performance testing, load testing, stress testing, security testing, web testing, application testing, penetration testing which are series of programs, all these programs under one umbrella are known as Cloud Testing as a Service [1].

#### **II. RELATED WORK**

Testing tools are encompassed of many mechanisms, these can range from clients and servers to more complex distributed schemes such as cloud data centres. The cloud based Testing tools are as dictated by previous requirements based on the user scenarios, will distribute its components which are designed based on test cases and previous experiences and user requirement over the cloud deployment. Given the potentially large number of Virtual Machines in a deployment to place multi-tasking and testing components based upon resource usage patterns, locality and availability. Cloud Testing services allow cloud users, auditors, developers and website managers to automate and speed up the testing. By using the cloud based Testing state that the tool has collected, it should determine the most advantageous placement scheme which ensures the availability, monitors the goals, generates the goal and Delivers the result [2].

443

Testing as a Service is a technology of Cloud computing rapidly growing due to its characteristics as pay as you use, Central to its success is elasticity; the means to rapidly change in scale and composition. Testing strategy allows the detection of problems, optimisation, cost forecasting, intrusion detection, auditing and other use cases. The lack of an appropriate testing strategy risks downtime, data loss, unpredicted costs and other unwanted outcomes. Testing as a Service is an important phase of the Cloud based Software testing that suffers from an abundance of unsolved problems. Testing has common research topics that focus on challenges, this research intends to solve to identify and quantify hard problems in software testing as defined by testing professionals in-order to provide a better research direction, to resolve the most frequently occurring and problematic challenges that the industry faces today. Testing as a Service is a cloud based web service, promoting testing software available to the cloud user at any time and at any place, that theoretically resolve some of the major testing issues plaguing the industry. The Testing as a Service is a potential candidate for solving the major problems identified by the industry by analysing the gap between the existing tools and the needs of the industry [3].

## **III. BLOCK DIAGRAM**

Web Interface is the cloud based Testing quality can be achieved by simply monitoring and identifying software metrics such as data flow metrics, control flow metrics and size metrics, where the metrics are the measurement of quality applied to all phases of the software life cycle. These are majorly used to analyse the complexity of the cloud based software testing. The Test Infrastructure is a service process which includes planning, discovery, vulnerability or attack and finally reporting the result [4].



# IV. GENETIC ALGORITHMIC APPROACH FOR THE CLOUD TESTING

To reduce the cost of testing and reconstructing the software, the evaluation should be conducted at early stages, any deficiency should be identified at the early stages, the testing process should be done at every stage, for it cloud based testing will be available to find and rectify the problems in the software. Achieving the reliability in software testing is by ensuring complete test process coverage, the basic concept is to test case generation approach to generate tests using test data method to generate data test [5].

The Genetic Algorithm has been rapidly developed and widely utilized. It can be applied to solve the complicated issues in science and engineering. Genetic Algorithm brings a remarkable impact to many growing fields, such as cloud computing, Testing Tools, artificial intelligence, knowledge discovery, pattern recognition, image processing, decision analysis, product process design, resource scheduling, and stock market analysis [6].

- 1. [Begin] Create random population of n chromosomes
- 2. [Evaluate] Assess the fitness f(c) of each chromosome c in the current population

3. [New population] Produce new population (of chromosomes) by repeating following steps until the generation of new population is complete

a. [Selection] Choose two parent chromosomes from the population in accordance to their fitness value (the better the fitness value, the higher chances of selection of chromosomes)

b. [Crossover] Allow crossover of selected parents to produce new offsprings, with a predefined crossover probability. Offsprings are the exact copy of the parents, in case of no crossover.

c. [Mutation] Allow recently generated offsprings to mutate at each locus (position in chromosomes), with a predefined mutation probability.

- d. [Accept] Add new offsprings to the new population
- 4. [Reinsertion] Create next population from the current and newly generated population for a further run of the algorithm
- [Test Condition] If the test condition is/are satisfied, stop, and return the best or optimal solution in the current population
   [Loop] Go to step 2

444

Automation of test case and test data generation in software testing can be done using Genetic Algorithm. It's impossible to cover all paths of the complex test program. A subset of the paths, named prime paths, based on all statement coverage criteria could be identified for testing. The searching of prime paths could be done using Genetic Algorithm, as it is capable of exploiting information available for initial search space. To guide the search, the genetic algorithm uses fitness function. A good fitness function helps focus attention, by providing useful information early in the search process. A fitness function with zero fitness does not provide any information for directing the search process. Therefore, it is desirable to find suitable fitness functions with respect to the nature of the problem. In test data generation, the Genetic Algorithm must search the adequate and accurate input values from the input domain, for exercising the prime paths. Although Genetic Algorithm does not require any domain knowledge, but there are conceivable outcomes and chances to assimulate domain knowledge [7].

## V. EXPERIEMENTAL EVALUATION OF CLOUD COMPUTING

An Experimental Approach is given to study and apply on the cloud frame work for web based Security Testing as a Service system to implement the prototype system, called Penetration Testing. The Open stack is associated with different projects or services among them basic services are the identity service(keystone), which consist of many other services, such as Compute service (Nova), the Network service (Neutron) and the Image service (Glance). Open stack storage provides object storage which is a distributed data storage system on different virtual machines [8].

Design and implementation of cloud test is done by using open stack cloud server which was built and configured prior to the start of the testing effort. Open stack cloud management software Essex was installed on a virtual box having an operating system of ubuntu, with a machine of 15 processors. An open-source Linux based fuzzing tool that is comparatively easy to use, we used BED that is Brute Force Exploit Detector for penetration testing in cloud, to check the exploits and vulnerabilities in cloud computing. It is mainly used to check the buffer overflows and handle string bugs [9].



Screen shot 1: Identification of the Internet Protocol Address in Ubuntu on Virtual Machine.

The forked version of Brute Force Exploit Detector is the Doona, which is generally uses the kali linux operating system instead of ubuntu. Now Install the Doona by using the following unix based command in the terminal. The Brute Force Exploit Detector is the BED is installed on operating system ubuntu. To Install the BED by using the following Unix based command in the terminal.

\$ cd /data/src/ \$ wget https://github.com/wireghoul/bed/zipball/master \$ mkdir -p /pentest/fuzzers/ \$ mv doom / /pentest/fuzzers/doom/

Once the Brute Force Exploit Detector is installed and internet protocol address is identified, then the next step is to install unzip package so that the installed packages can be unzipped using the following command

\$ sudo apt-get install zip unzip



Screen shot 2: Installation of unzip package on Ubuntu on Virtual Machine to unzip the software.

 $sudo unzip c:\vc\os\doona.zip$ 

\$apt-get update Apt-get install doona

Then you need an target internet protocol displayed on screen is "192.168.xx.xx"

i ping target.com see in below screen shot



Then you need to enter whole command for attack

Command:-doona -m http -t <target ip> -p <target port> -M 10 (max not after exit)

```
This is my command :- doona -m http -t 216.xx.xx.xx -M 10
```

/WHOIS	= PINGER/FIP/HTTP/ING/EP/PSC/POP/PROXT/RIGP/SHIP/SOCKS4/SOCKS5/TETP
-t <target></target>	= Host to check (default: localhost)
-p <port></port>	= Port to connect to (default: modil Cale of the Condard port)
-o <timeout></timeout>	= seconds to wait after each test (demanter 2 seconds)
-r <index></index>	= Resumes fuzzing at test case inde
-d	= Dump test case to stdout (use in combination with -r)
-M <num></num>	= Exit after executing onums number of fuzz cases
uen " /doona	- Help (this text)
use 17 uuonia	the an Employed an include a sective option.
Only -m is a	mandatory switch
rootgkali:-#	doona -m http -t
and an and a second	
Doona 8 7 by	wiregnout readlistand berhacker.com) based on BED by nim and snakebyte
+ Buffer over	rilowitestingin you been at this more you are this to hair.
1/3/ 1/3/	XAX ]
and the second se	
1	

Use the HTTP plugin (-m HTTP) to fuzz the target (-t 192.168.1.15), stopping after 5 cases (-M 5):

root@ubuntu:~# doona -m HTTP -t 192.xx.xx.xx -M 5

## VI. CONCLUSION

Penetration tests or security test for the cloud security is done by using the doona tool on the open stack cloud. The open stack is being used by many large companies for their private, as well as public clouds. Improving the overall security posture of open-stack through security based applications and data security on the cloud, it also provides the vulnerability verification and protection from a session hijacking attack is done [10].

# VII. REFERENCES

[1].J. Chen, C. Wang, F. Liu and Y. Wang, "Research and Implementation of a Software Online Testing Platform Model Based on Cloud Computing," 2017 Fifth International Conference on Advanced Cloud and Big Data (CBD), Shanghai, 2017, pp. 87-93.
[2]. M. S. N. Murthy and V. Suma, "Software testing and its scope in CLOUD: A detailed survey," 2017 International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), Bangalore, 2017, pp. 269-273.

[3]. L. Xiaobing, C. Wei, Z. Feng, X. Bin and S. Zhiqiang, "Design of a security smart meter software testing cloud service system," 2016 IEEE Information Technology, Networking, Electronic and Automation Control Conference, Chongqing, 2016, pp. 874-878.

[4]. H. Li, L. Wang, X. Wen, Z. Lu and L. Ma, "Constructing Service Function Chain Test Database: An Optimal Modelling Approach for Coordinated Resource Allocation," in *IEEE Access*, vol. PP, no. 99, pp. 1-1, 2017.

[5].J. Vijaya Chandra, Dr. NarasimhamChalla, Dr. Sai Kiran Pasupuleti, Dr. V. Krishna Reddy, Dr. K. Thirupathi Rao, "Numerical Formulation and Simulation of Social Networks Using Graph Theory on Social Cloud Platform", Global Journal of Pure and Applied Mathematics, Volume 11, Issue 3, Pages 1253-1264, 2015.

[6]. K. Zhu, H. Song, L. Liu, J. Gao and G. Cheng, "Hybrid Genetic Algorithm for Cloud Computing Applications," *IEEE Asia-Pacific Services Computing Conference*, Jeju Island, 2011, pp. 182-187.

[7]. M. R. Reddy and P. Yalla, "Mathematical analysis of Penetration Testing and vulnerability countermeasures," 2016 IEEE International Conference on Engineering and Technology (ICETECH), Coimbatore, 2016, pp. 26-30.

[8].J. Vijaya Chandra, NarasimhamChalla, Mohammed Ali Hussain, "Data and Information Storage Security from Advanced Persistent Attack in Cloud Computing", International Journal of Applied Engineering Research, vol. 9 no. 20 pp. 7755-7768, 2014.

[9]. Marri Rami Reddy, P.Yalla, J. Vijaya Chandra, "Design and implementation of integrated testing tool based on metrics and quality assurance", International Journal of Applied Engineering Research, Volume 9, Issue 21, Pages 10463-10472, 2014.

[10]. K. P. Haubris and J. J. Pauli, "Improving the Efficiency and Effectiveness of Penetration Test Automation," 2013 10th International Conference on Information Technology: New Generations, Las Vegas, NV, 2013, pp. 387-391.



447