IJCTA, 9(36), 2016, pp. 533-541
© International Science Press

Continuous Security Assessment of Applications in Cloud Environment

K. Vijayakumar* and C. Arun**

Abstract : Nowadays cloud computing is gaining popularity due to the services and resources available. Organizations are preferring cloud to reduce costs and attain scalability but the factor of security in the cloud is not taken into much consideration. Vulnerabilities and security issues of the application are checked outside the cloud, but not on the cloud. This paper aims to propose a system where the primary concern is developing security vulnerability environment on the cloud for testing and maintaining applications being migrated to the cloud and already residing on the cloud. The security vulnerability testing environment is developed based on cloud server specifications and the results of the tests are stored for future use. A new application on the verge of being migrated to the cloud can use the testing tool to periodically check for vulnerabilities and security threats and also from the data's stored on the cloud server from previous tests for the prevention and removal of security threats and vulnerabilities. The security environment should be deployed on a cloud platform and should run on a typical cloud server. The vulnerabilities detected are also stored on the cloud and the entire testing procedure is a recursive process. Time and again the environment tests the application for any security or vulnerability thread. The security tool is up and running all the time to detect any structural or data changes on the targeted applications so as to inform the client of any possible vulnerability occurring.

Keywords : Cloud computing, Cloud migration, continuous assessment; security assessment.

1. INTRODUCTION

Cloud computing is the practice of rather than using local server or personal computer, using a network of remote servers hosted on the internet to store, manage and process data. Cloud computing provides on demand resources with minimal management requirement. It relies on sharing of resources to achieve economy of scale [22]. Most organizations shift to cloud to reduce the upfront infrastructure cost. It requires adaption to cloud pricing model. Cloud services in the form of software as a service (SaaS), platform as a service (PaaS), and infrastructure as a service (IaaS) reduce application the maintenance and development costs dramatically. Cloud migration involves moving data or other business elements from local server or personal computer to cloud environments. The process of transitioning all or parts of an organization's data, applications and services from on-site premises behind the firewall to the cloud, where the information can be provided over the Internet on an on-demand basis. However, the migration of applications to the cloud platform raises security issues because applications are no longer within their organization's secure domain. Security threats vary from one application to another and as such, the promise of generalized security measures supplied by cloud service providers (CSPs) might not be sufficient to alleviate security concerns., cloud computing can also enable a company to potentially reduce capital expenditures and operating costs while also benefiting from the dynamic scaling, high availability, multi-tenancy and effective resource allocation advantages cloud-based computing offers.

* Research Scholar, Sathyabama University, Faculty of Computer Science & Engineering, St.Joseph's Institute of Technology, OMR, Chennai, India

** Professor, Dept. of ECE, R.M.K College of Engineering and Technology, Chennai, Tamilnadu, India

2. ISSUES OCCURRING IN THE CLOUD ENVIRONMENT

The migration issues faced by organizations require serious considerations. These issues might cause serious problems later. But the main issue lies after the migration process. The application is made secure on the local server before migration [1]. But there are no dedicated client side security testing tools on cloud which provide the security to the application by testing the application on the cloud for any vulnerability occurring. This requires concentration by developers as most of the applications these days are dynamic in structure and might undergo structural changes while on cloud platform. Dynamic nature of applications may also introduce to a large number of vulnerabilities where the hackers take advantage of the dynamic nature of applications to cause malicious attacks. Also the Cloud Service Providers (CSP) might also make any unauthorized changes to the application which might lead the application to be vulnerable to attacks and many security problems as the organization shift to cloud to be economical but not at the cost of the loss or theft of the data.

3. PROPOSED SYSTEM

Here we propose a system where a security environment is developed on cloud. Here the applications must be integrated with the testing environment so as to keep it under the surveillance of the testing application. The proposed system is developed for PHP applications. An open source code analyzer by the name of RIPS is used. Here we enhance the functionality of RIPS and deployed in cloud. The RIPS analyser scans the source code line by line searching for any vulnerability. It divides the code in tokens and parses into a model. Further it detect sinks which could influenced by the end user during the control flow of the application. This approach would help in risk assessment of the application.

The vulnerabilities are tested based on the operations like Code execution, Command execution, File Disclosure, File Inclusion, File Manipulation, LDAP injection, PHP object injection, Protocol injection, Reflection injection, SQL injection, XPath injection, Cross-site scripting, HTTP Response Splitting and Session Fixation. The RIPS also tests the code in two different approaches they are Bottom-up approach and Top-down approach. Different type of code styles in also checked such as: Ayti, Notepad++, Barf, Term, Espresso,Twilight, Code dark, Phps, Print. In the paper [1] Sanjay Madria et al propose a system where the application is tested based on Microsoft's STRIDE tool which and use CAPEC database with a risk ranking algorithm where the application is made safe before migration *i.e.* offline. Here in the proposed system we take the measures proposed in paper [5] and enrich the process by adding the security vulnerability to the cloud environment for after migration security. The CAPEC database used by Sanjay Madria et al is integrated with CVSs database published National Institute of Standards and technology.

This integration gives us the advantage of examining the application for more vulnerability and compares the testing results with these databases so as to ensure the application is of a standard one. This accomplished using four steps in the proposed system where the entire application is migrated along with the testing application also being migrated. The four modules are

1. Migration of the application to the cloud environment
2. Migration of the vulnerability scanner application into the cloud
3. Completing the scanning of the application in cloud
4. Storage and analysis of scan results for action

The application chosen is of client development. The application taken here for sample is PHPWiki. The PHPWiki is an open source content management system. Here is the application has to be integrated with the testing application. The testing application is migrated first into the cloud. Now the application is also migrated to cloud. The migration process is of standard one followed in [2]. The application is then integrated with the testing application. This is done by placing the application is the same directory of that of the testing tool in the cloud file system. The testing application then tests the application on cloud. If any structural changes identified by the testing application on the client application then the testing application again tests the cloud and also informs the user. The results of all the testing process are stored in a database. The main purpose of storing the results in the database is to display the results to the client in desired format and also to standardize the result storing process.

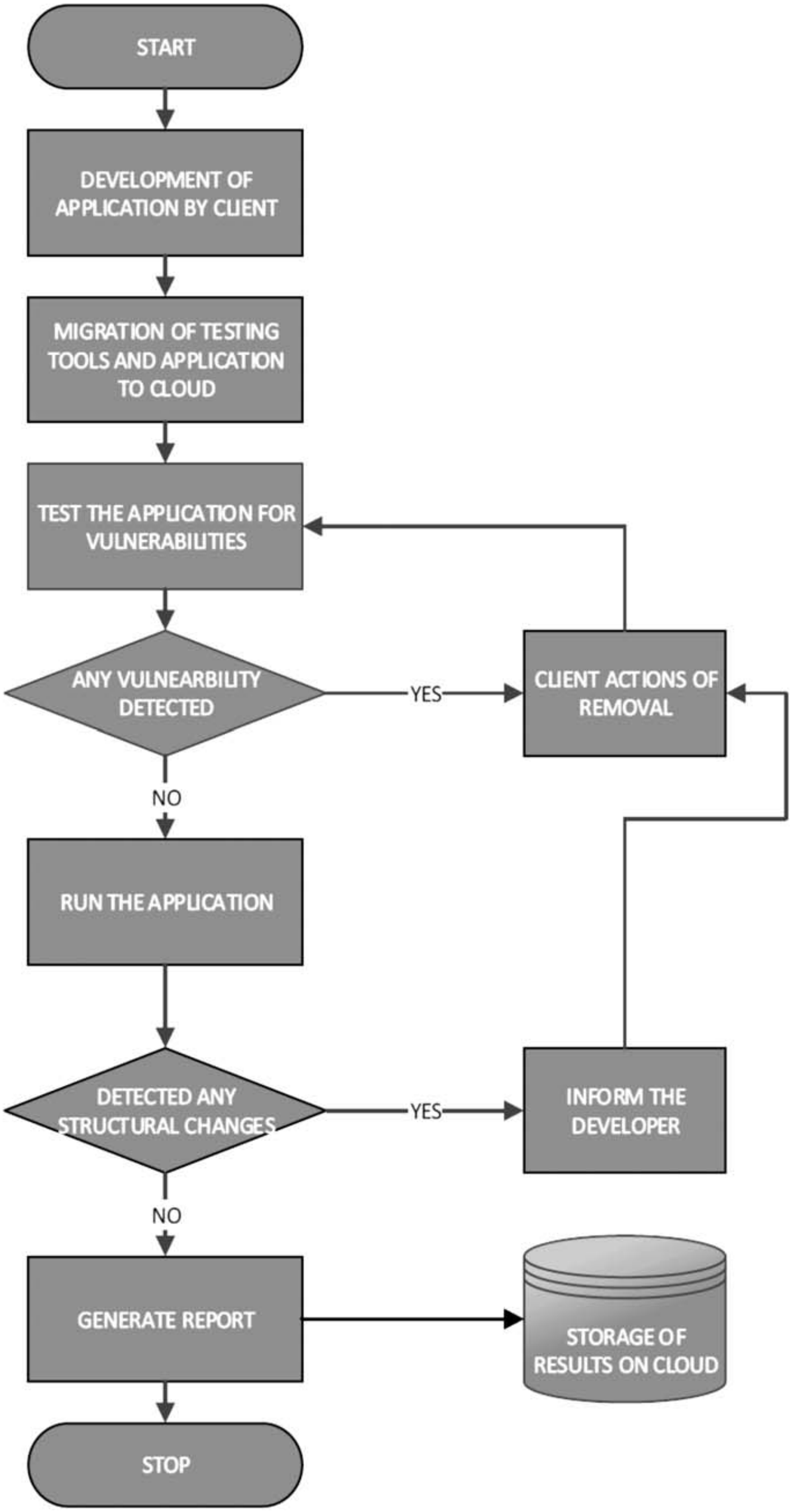

START
DEVELOPMENT OF APPLICATION BY CLIENT
MIGRATION OF TESTING TOOLS AND APPLICATION TO CLOUD
TEST THE APPLICATION FOR VULNERABILITIES
ANY VULNEARBILITY DETECTED
YES
CLIENT ACTIONS OF REMOVAL
NO
RUN THE APPLICATION
DETECTED ANY STRUCTURAL CHANGES
YES
INFORM THE DEVELOPER
NO
GENERATE REPORT
STORAGE OF RESULTS ON CLOUD
STOP

Figure 1: Flow Chart of the Proposed System

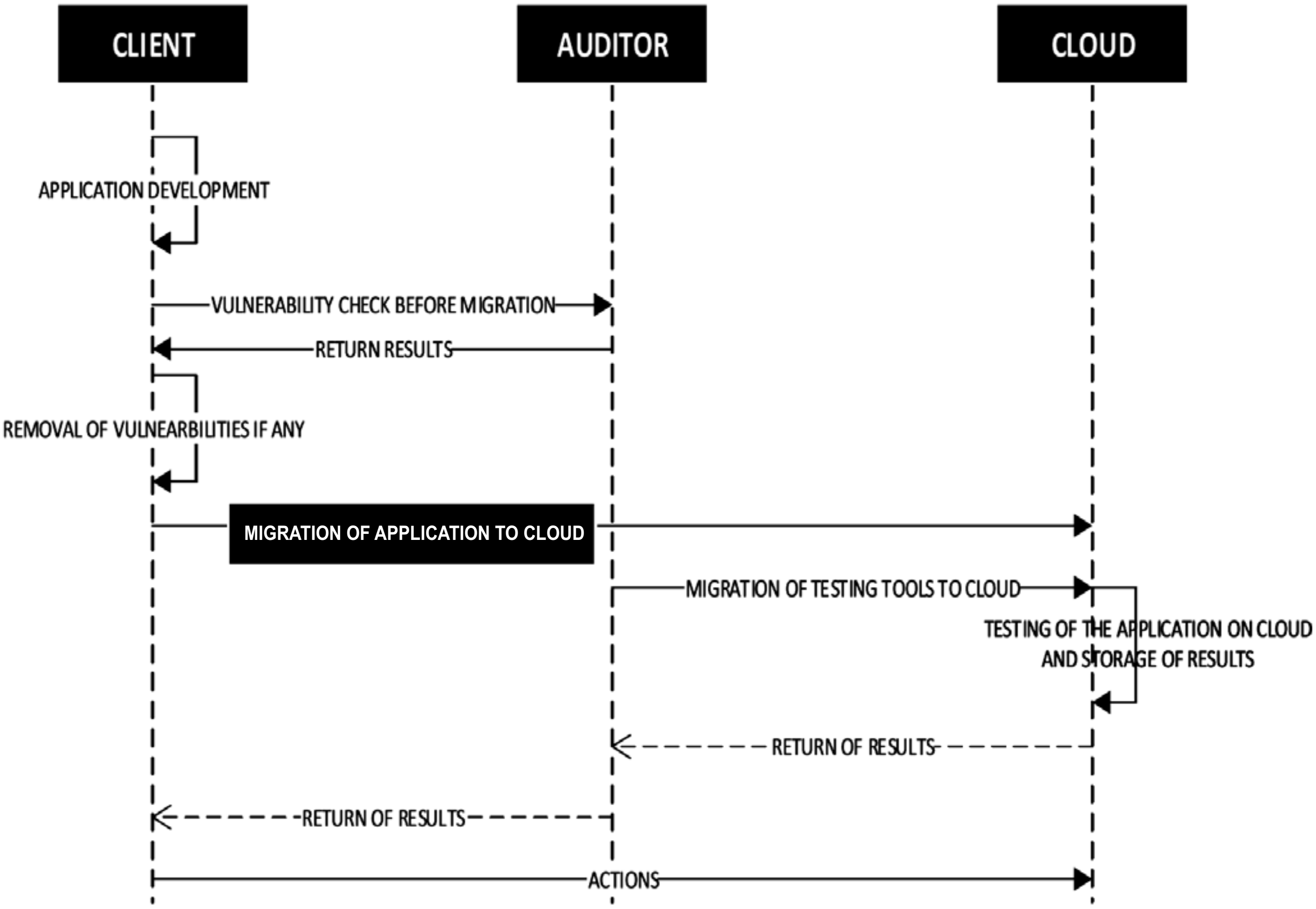

CLIENT
AUDITOR
CLOUD
APPLICATION DEVELOPMENT
VULNERABILITY CHECK BEFORE MIGRATION
RETURN RESULTS
REMOVAL OF VULNEARBILITIES IF ANY
MIGRATION OF APPLICATION TO CLOUD
MIGRATION OF TESTING TOOLS TO CLOUD
TESTING OF THE APPLICATION ON CLOUD AND STORAGE OF RESULTS
RETURN OF RESULTS
RETURN OF RESULTS
ACTIONS

Figure 2: Sequence Diagram of the Proposed System

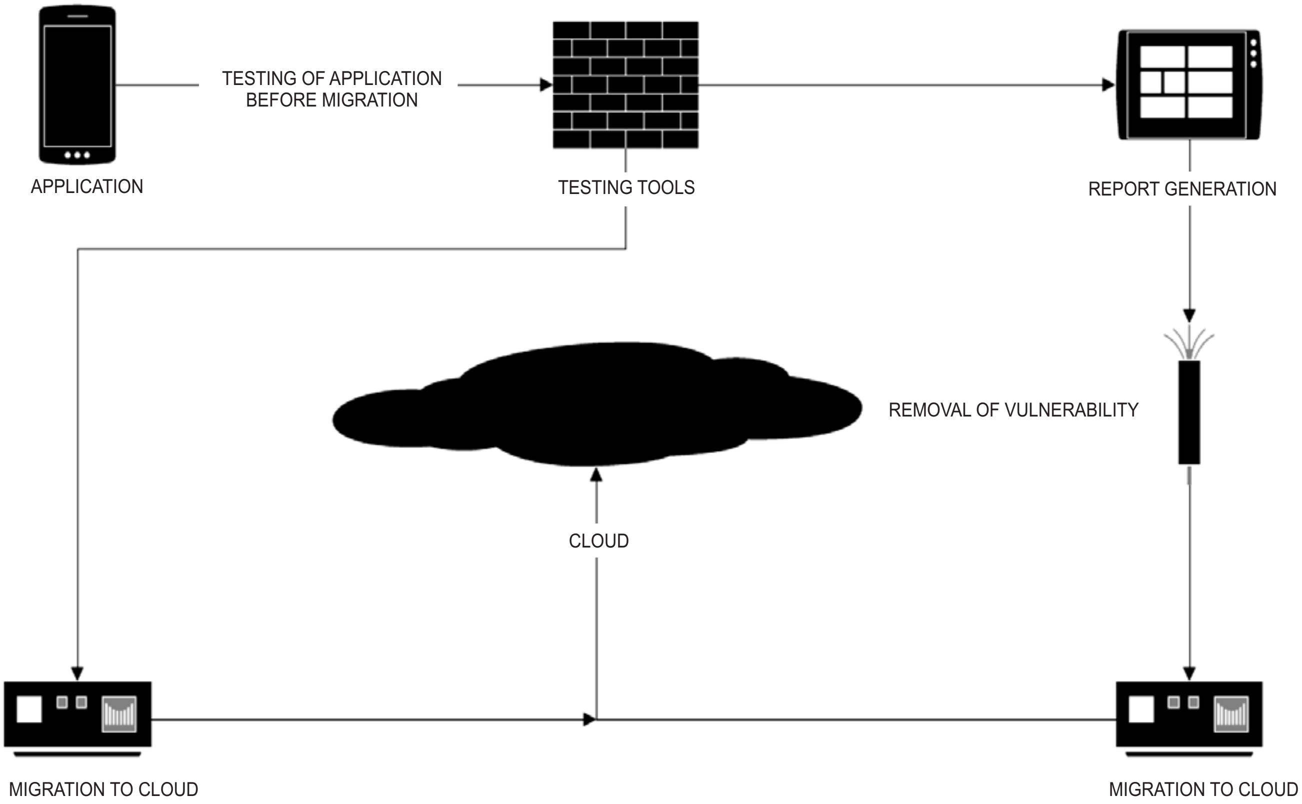


Figure 3: Architecture of the Proposed System

The sequence diagram above in figure 2 show the timeline based activities of the proposed system. The first activity is of the client to develop the application. The client checks the application for vulnerability and security threats outside the cloud. This step makes the application free of any vulnerability outside cloud. Now the application is migrated to the cloud through the auditor, now the auditor also migrated the testing framework to the cloud. While the application is up and running on the cloud it is under the examination process of the testing framework. Any changes detected on the application the testing environment re-check the entire application for vulnerabilities again. Then the results of the checks are then sent to the auditor who records these for future use and send it to the client in form of report so as to determine the actions to take. The below diagram gives an architectural view of the entire proposed system. The system consists of client application which is tested for vulnerabilities before being migrated to cloud. Next after testing a report is generated of the testing process and results. After the report, the vulnerabilities if any detected are removed for safety of the application. Then the application is migrated to cloud along with the testing framework. The insight of the cloud is exhibited in another other diagram figure 4. This diagram illustrated the working architecture of the proposed system inside the cloud environment. The process also includes storage of the results of the testing purpose on the cloud for future use when any other application is migrated it may use these test results as reference and avoid the issues or take preventive measures accordingly.

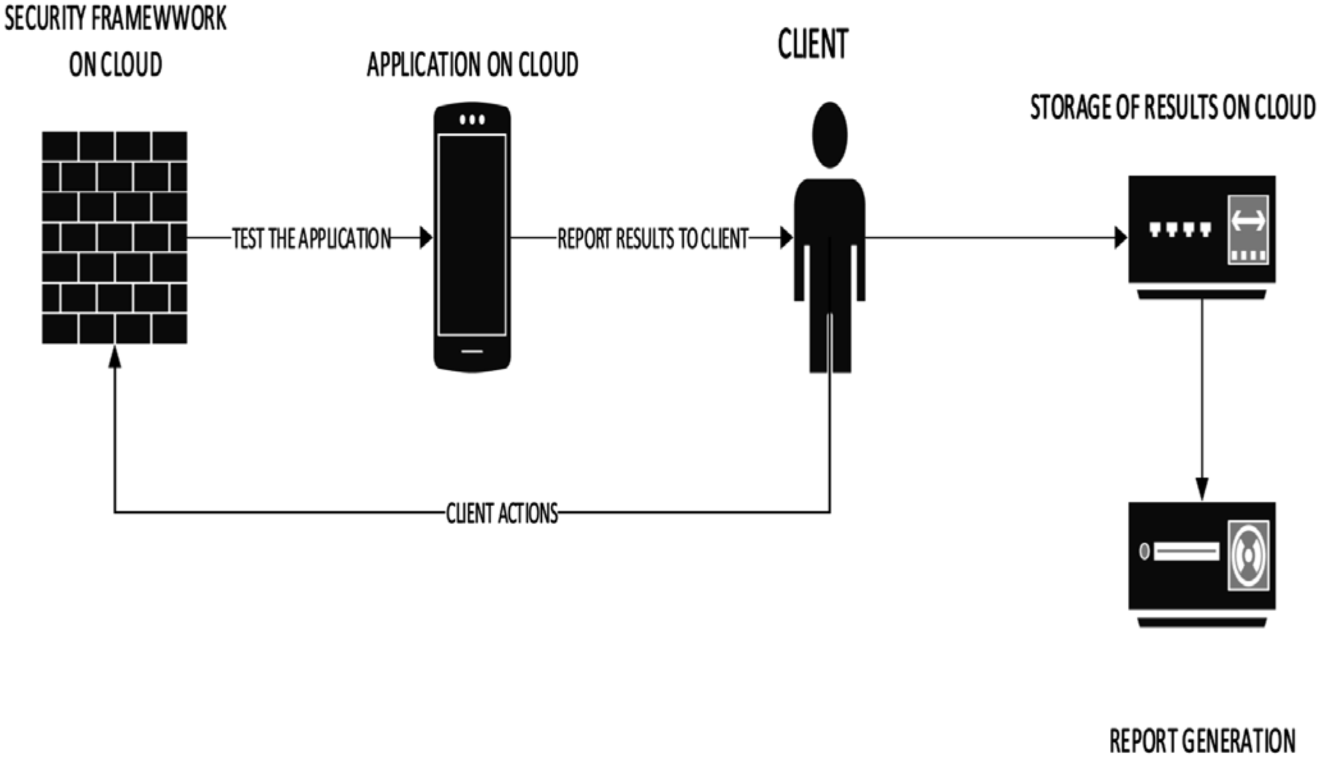
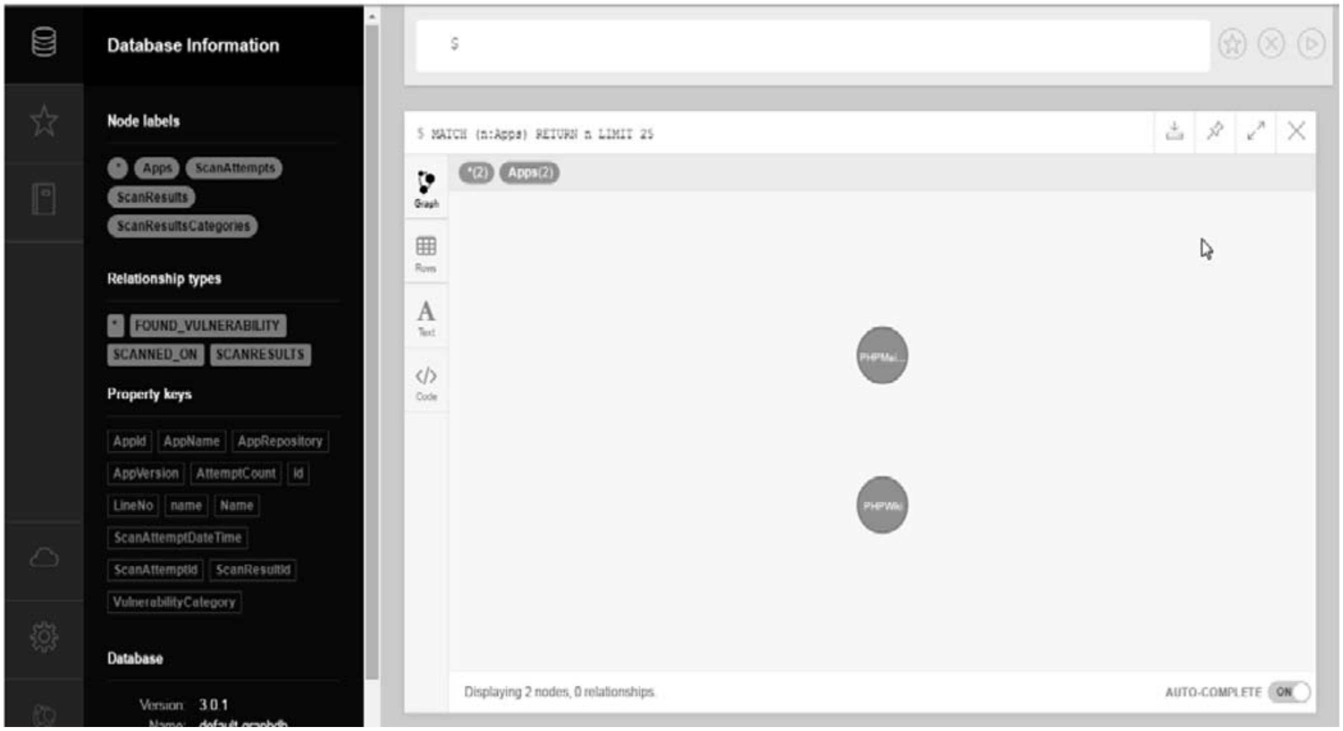


Figure 4: Process of Testing on Cloud

4. RESULTS AND INFERENCE

The entire application scan results are stored the Neo4j Graph database which is essential to analyse the relationship between the code analyses. In the below given figure you can identify that two applications are available.



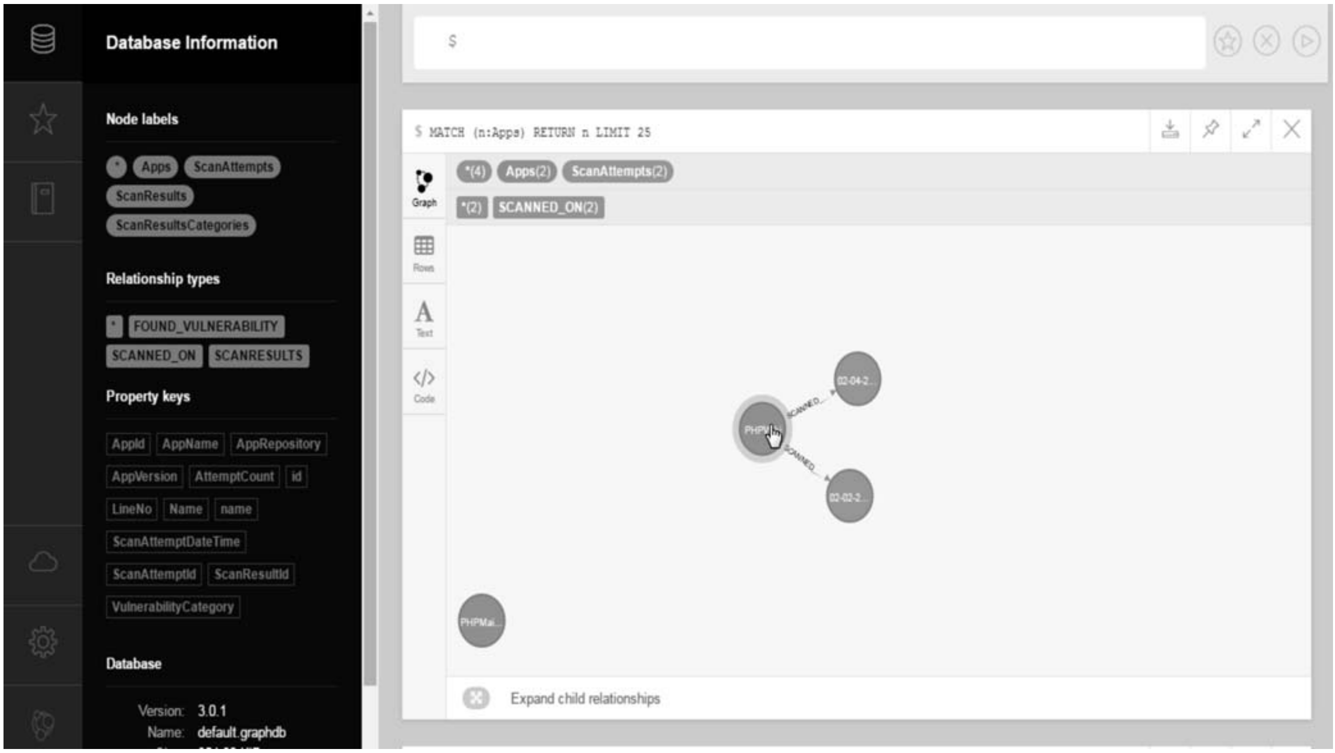
```
$ MATCH (n:Apps) RETURN n LIMIT 25
```

Displaying 2 nodes, 0 relationships.

Database Information
Node labels
* Apps ScanAttempts ScanResults ScanResultsCategories
Relationship types
* FOUND_VULNERABILITY SCANNED_ON SCANRESULTS
Property keys
AppId AppName AppRepository AppVersion AttemptCount id LineNo name Name ScanAttemptDateTime ScanAttemptId ScanResultId VulnerabilityCategory
Database
Version: 3.0.1
Name: default.graphdb
$
*(2) Apps(2)
Graph
Rows
Text
Code
PHPMai...
PHPWiki
AUTO-COMPLETE ON

Figure 5: Graph database for application scan result

In this case on expanding the PHPWiki application we can realize that it has been scanned on two different days as given in the below given figure (2nd Feb 2016 and 2nd April 2016) with the relationship SCANNED_ON.

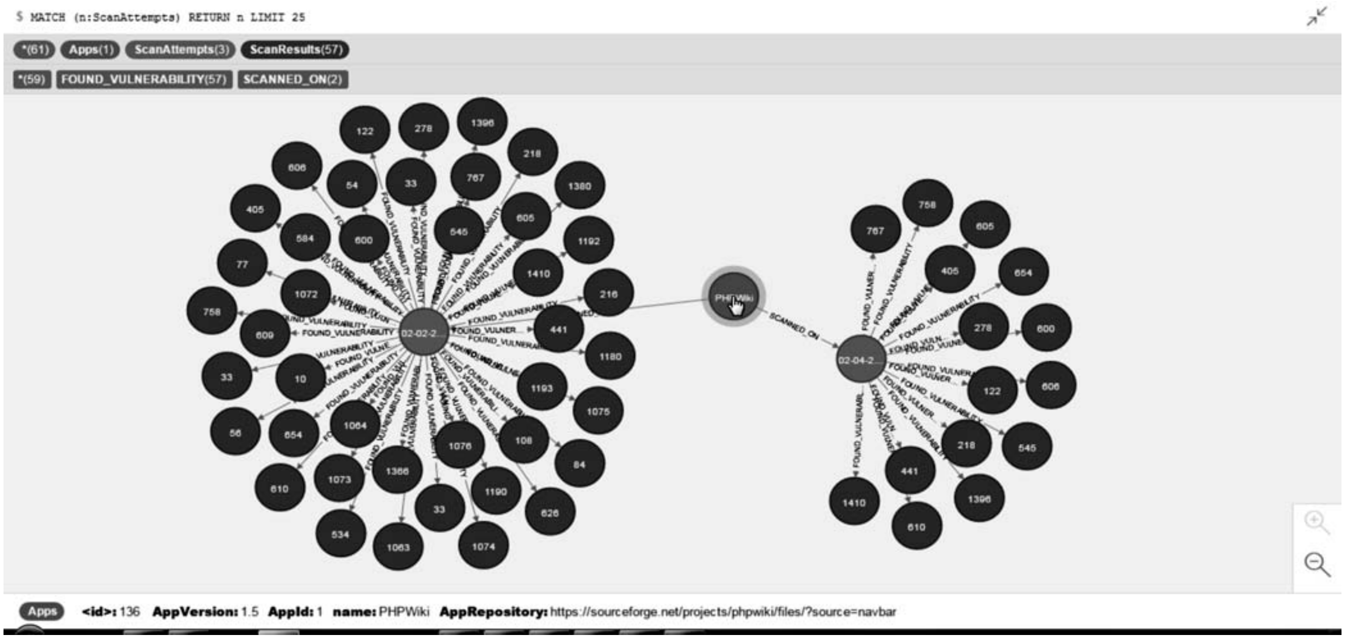


```
$ MATCH (n:Apps) RETURN n LIMIT 25
```

Database Information
Node labels
* Apps ScanAttempts ScanResults ScanResultsCategories
Relationship types
* FOUND_VULNERABILITY SCANNED_ON SCANRESULTS
Property keys
AppId AppName AppRepository AppVersion AttemptCount id LineNo Name name ScanAttemptDateTime ScanAttemptId ScanResultId VulnerabilityCategory
Database
Version: 3.0.1
Name: default.graphdb
$
*(4) Apps(2) ScanAttempts(2)
*(2) SCANNED_ON(2)
Graph
Rows
Text
Code
PHPW...
SCANNED...
SCANNED...
02-04-2...
02-02-2...
PHPMai...
Expand child relationships

Figure 6: Graph to analyse the relationship between the code analyses

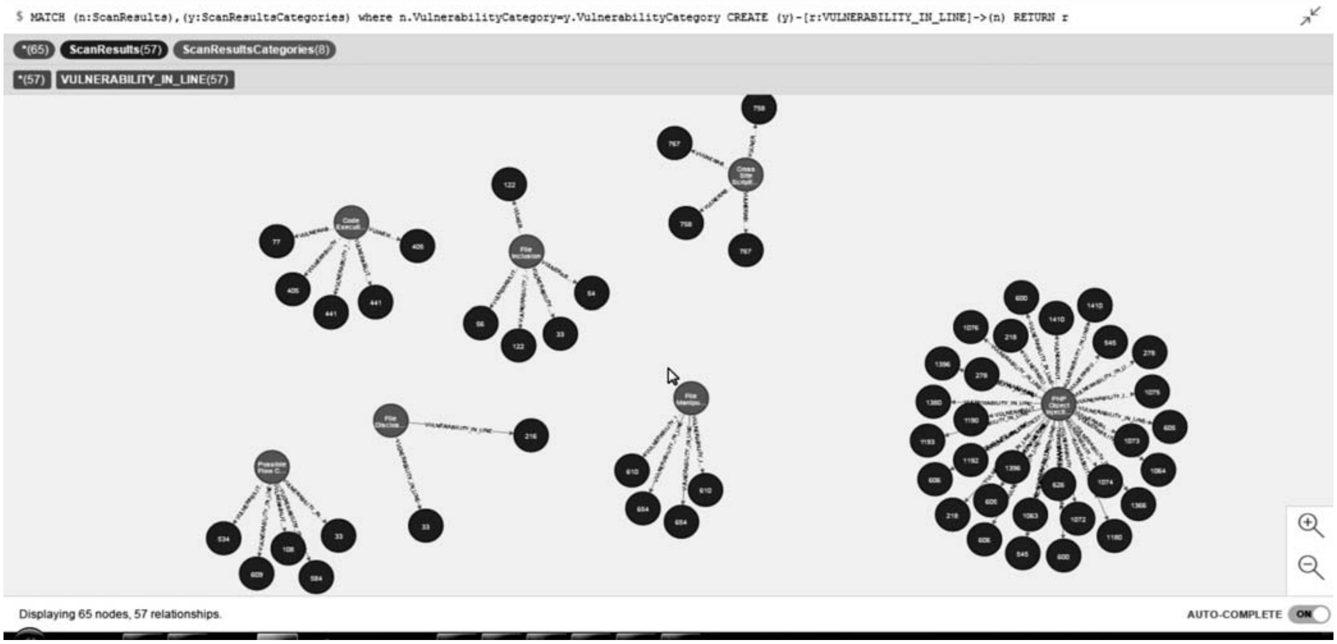
The below given figure gives the complete view of the scan attempt made and the number of vulnerabilities found on that date. You can see that # of vulnerabilities found in 2nd April 2016 is less than the vulnerabilities found 2nd Feb 2016.



```
$ MATCH (n:ScanAttempts) RETURN n LIMIT 25
```

Apps <id>: 136 AppVersion: 1.5 AppId: 1 name: PHPWiki AppRepository: https://sourceforge.net/projects/phpwiki/files/?source=navbar

Figure 7: Finding Vulnerabilities on different dates



```
$ MATCH (n:ScanResults),(y:ScanResultsCategories) where n.VulnerabilityCategory=y.VulnerabilityCategory CREATE (y)-[r:VULNERABILITY_IN_LINE]->(n) RETURN r
```

Displaying 65 nodes, 57 relationships.

Figure 8: Vulnerabilities and its relationship

The following are the list of vulnerabilities found in the application as given below, the all the vulnerabilities have the relationship of an scan attempt on a application in cloud:

Session Fixation
PHP Object Injection
Cross Site Scripting
File Manipulation
Code Execution
File Inclusion
File Disclosure
Possible Flow Control

In this below given figure we can figure out the lines which has the same vulnerability. This would help you identify the problem quickly in visual way and there by triggering quick action on vulnerabilities.

With above given graphical visualization of vulnerabilities using Neo4J would help in determining the vulnerabilities and its relationship. It also provides the opportunity to understand the vulnerabilities not fixed and occurring in repeated attempts as well.

5. CONCLUSION AND FUTURE WORK

The project proposes new ways to scan, capture and visualize the scan results. The key challenge addressed is to move to cloud based on Scan Analyser to automate the scanning of vulnerabilities in the code during the continuous integration process itself. This effort introduces a dedicated cloud based side scanning tool which can scan and store the results in cloud. The tool is deployed on cloud which makes the testing process a continuous one and thus the application is free of vulnerabilities all the time. This reduces overload of clients to check the applications migrated to cloud for vulnerabilities. Each time the client has to come out of cloud to test the application but this problem is solved. Also it allows the application on cloud to be under the testing process all the time. Here the testing process is a recursive process where as in general the testing process is a onetime procedure and happens only at the beginning. Thus the testing process is fool proof and accurate. Since the detection process is automated the client need not detect the vulnerabilities but client needs to specify and carry out the action suitable to remove or reduce the effect of vulnerabilities. The system depicts the areas in the source code where vulnerabilities occur. Thus the system identifies the vulnerabilities and areas off code where these occur and provide a graphical output along with storage of results, all on cloud which is the need of the hour. The proposed system is developed only for testing of application built using PHP scripting language. If the system receives a positive feedback, I can develop the tool for all languages ranging from java to python to .Net so on and so forth. Also the presentation of results can be done as per the client requirements. One more enhancement that can be made after the testing of the application, the application can also undergo more series of scanning by standardized tools so has to compare the results and make the system even more fool proof.

6. REFERENCES

1. Sanjay Madria and Amartya Sen, “Offline Risk Assessment of Cloud Service Providers”, 2015 IEEE Cloud Computing.
2. Eva Kühn, Vesna Šešum-Čavić, Thomas Schmid, “Dynamic Migration of Cloud Services”, 2014 IEEE 3rd Symposium on Network Cloud Computing and Applications.
3. Amir DJENNA, Mohamed BATOUCHE, “Security Problem in Cloud Infrastructure”, Department of Computer Science, University of Constantine2, Constantine, ALGERIA, 2014 IEEE.
4. Jun Wu, Zhengyuan Wang and Sisi Gao, “Assessing the Cloud Migration Readiness”, School of Economics and Management Beijing University of Posts and Telecommunications Beijing, China, 2014 IEEE.
5. Georgiana Mateescu, Marius Vlădescu, Valentin Sgârciu , “Auditing Cloud Computing Migration”, University Polytechnics/Automatic and Computer Science, Bucharest, Romania, 9th IEEE International Symposium on Applied Computational Intelligence and Informatics, 2014, Timişoara, Romania.
6. Omar SEFRAOUI, Mohammed AISSAOUI and Mohsine ELEULDJ, “Cloud Computing Migration and IT resources rationalization”, Ecole Nationale des Sciences Appliqu´ees. Universit´e Mohammed Premier, Oujda, MAROC, 2014 IEEE.
7. Michael Menzel, Rajiv Rajan, Lizhe Wang, Samee U Khan, Jinjun Chen, “CloudGenius: A Hybrid Decision Support Method for Automating the Migration of Web Application Clusters to Public Clouds”, IEEE Transactions on Computers , May 2015.
8. Jijun Zhang, Dejian Sun, Donghang Zhai, “A Research on The Indicator System of Cloud Computing Security Risk Assessment”, Library Air Force Aviation University of China Changchun, China, IEEE 2014.
9. Daniel W.K. TSE, “Challenges on Privacy and Reliability in Cloud Computing Security”, Department of Information Systems City University of Hong Kong, Hong Kong SAR, 2014 IEEE.

10. Talal H. Noor, Quan Z. Sheng, Lina Yao, Schahram Dustdar and Anne H.H. Ngu, “CloudArmor: Supporting Reputation-based Trust Management for Cloud Services”, Member IEEE, 2014 IEEE.

11. Wiem Abderrahim, Zied Choukair, “A Framework Architecture Based Model For Cloud Computing Adaptive Migration”, Higher School of Communications of Tunis Mediatron Lab - Tunis, Tunisia, 2014 IEEE.

12. Ankit Upadhyay, Prashant Lakkadwala, “Secure Live Migration of VM’s in Cloud Computing: A Survey”, Department of Computer Science & Engineering Acropolis Technical CampusIndore, India, 2014 IEEE.

13. Niu Haichun, Liu Yong, “A Mobile Agent-based Task Seamless Migration Model for Mobile Cloud Computing”, Information Engineering College Henan University of Science and Technology, Luoyang, China, 2014 IEEE Workshop on Advanced Research and Technology in Industry Applications (WARTIA).

14. Satish Narayana Srirama*, Vladislav Ivanistsevy*, Pelle Jakovits+, Chris Willmore*, “Direct Migration of Scientific Computing Experiments to the Cloud”, *Institute of Computer Science, University of Tartu, J. Liivi 2, Tartu, Estonia, +Institute of Chemistry, University of Tartu, Ravila Str. 14A, Tartu, Estonia, 2013 IEEE.

15. Noor-ul-hassan Shirazi, Steven Simpson, Angelos K. Marnerides, Michael Watson, Andreas Mauthe and David Hutchison, “Assessing the Impact of Intra-Cloud Live Migration on Anomaly Detection”, 2014 IEEE 3rd International Conference on Cloud Networking (CloudNet).

16. Xiumin Wang, Xiaoming Chen, Chau Yuen, Weiwei Wu, Wei Wang, “To Migrate or to Wait: Delay-Cost Tradeoff for Cloud Data Centers”, Globecom 2014 - Symposium on Selected Areas in Communications: GC14 SAC Cloud Networks.

17. Kan Yang, Xiaohua Jia, “An Efficient and Secure Dynamic Auditing Protocol for Data Storage in Cloud Computing”, IEEE Transactions on Parallel and Distributed Systems.

18. Cluster based Key Management Authentication in Wireless Bio Sensor Network “, ,International Journal of pharma and bio sciences.

19. http://cloudaudit.org/CloudAudit/About.html

20. http://csrc.nist.gov/publications/nistpubs/800-144/SP800-144.pdf

21. http://www.ibm.com

22. http://www.google.com

23. http://www.wikipedia.com