Automated Vulnerability Detection and Prediction by Security Testing for Cloud SAAS

S. Krishnaveni1*, S. Prabakaran2 and S. Sivamohan3

1Department of Software Engineering, SRM University, Chennai – 603203, Tamil Nadu, India; vanimithila@gmail.com
2Department of Computer Science and Engineering, SRM University, Chennai – 603203, Tamil Nadu, India; prabakaran.man@gmail.com
3Department of Information Technology, SRM University, Chennai – 603203, Tamil Nadu, India; sivamohan7@gmail.com

Abstract

Background: Cloud SaaS becomes a susceptible target because it shares the application access and data among various tenants. So the careful security testing is necessary to avoid the security problems. As per the recent Survey done by OWASP reveals that SQL injection and Cross Site Scripting (XSS) are two of the most serious vulnerabilities in cloud based applications today, because of most dangerous attacks get exploited and steal the user’s credentials such as cookie, credit card number etc. In order to mitigate these security vulnerabilities cloud service providers adopt a number of vulnerability prediction and detection approaches based on static and dynamic analysis techniques. These existing security mechanisms cannot alleviate attacks targeting cloud based applications, which focused on theoretical as well as practical solutions for certain security problems. The developers are necessary to ensure the delivery of safe applications and need to identify potential security issues within the applications before the stored in cloud environment. Method: Security testing is an effective mechanism to find out how vulnerable a Cloud based SaaS may be and to determine whether SaaS offering is Susceptible to XSS and resources are protected from intruders. The proposed work provides the automated Security testing by providing vulnerability detection and prediction model. The proposed Hybrid attributes are static and dynamic, the use of dynamic attributes to complement static attributes in the prediction of vulnerabilities. Findings: This paper propose a prediction models by collecting the static and dynamic attributes that are based on classification in order to predict SQLI and XSS vulnerabilities. Improvement: In order to improve the current work we proposed a hybrid attributes such as dynamic attributes to complement static attributes in the prediction of vulnerabilities in cloud based applications. By efficiently using the developed prototype tool for data collection and its used to evaluate our models on six open source cloud based applications. Based on the experiments average results states that the best predictor is MLP which achieved a pd=72%, pf=12% for predicting SQL Injection and pd=82%, pf=19% for predicting XSS Cross Site Scripting. The results of our experiments states that our prediction model is an ease and effective technique to predict and detect the SQLI and XSS vulnerabilities in Cloud applications.

Keywords: Cloud Computing, Data Mining, SaaS Applications, Security Testing, Static and Dynamic Analysis, Vulnerability Detection, Vulnerability Prediction

1. Introduction

As of late distributed computing have seen an eminent development and now it turns out to be most innovation in IT industry. The benefits of utilizing distributed computing are 1. It decreases noteworthy cost in light of the fact that without putting resources into foundation the capacity of figuring can be expanded 2. Its availability, adaptability and versatility. Distributed computing is characterized as “a style of figuring where hugely adaptable IT-empowered capacities are conveyed as a service to outside clients utilizing Internet technologies”. The

*Author for correspondence
different administrations offered by distributed computing engineering are layered as the Infrastructure as a Service (IaaS) giving framework bolster administrations. Platform as a Service layer gives stage arranged administrations, for example, nature for facilitating the client applications. software as a Service (SaaS) layer which have a trademark that it offers a whole application as administration in view of the request of the customer. In SaaS sending model since the information is saved outside the limit of endeavors, so the administration supplier itself give and receive security checks to guaranteeing information security either from assault of programmers and vindictive clients to access unapproved information to them. In the most predominant vulnerabilities in this layer are information infusion vulnerabilities, for example, SQLI and XSS.

In, Hackers are becomes very smart nowadays, they use new tricky techniques to attack the cloud based web applications to steal the credentials without the user’s knowledge. since cloud computing is open to everyone its avails the resources to many users, so it gets tempting the target to hackers. cloud service providers are giving their software binaries to users. even users may be a hackers, they can easily get in to the cloud and modify the users information. As per the recent Survey done by OWASP reveals that SQL injection and Cross Site Scripting (XSS) are two of the most serious vulnerabilities in cloud based applications today, because of most dangerous attacks gets exploited and steal the user's credentials such as cookie, credit card number etc. In order to mitigate these security vulnerabilities cloud service providers adopt a number of vulnerability prediction and detection approaches based on static and dynamic analysis techniques. Security testing is an effective mechanism to find out how vulnerable a Cloud based SaaS may be and to determine whether SaaS offering is Susceptible to XSS and resources are protected from intruders. The proposed work provides the automated Security testing by providing vulnerability detection and prediction model. The proposed Hybrid attributes are static and dynamic, the use of dynamic attributes to complement static attributes in the prediction of vulnerabilities in cloud based applications.

The paper is organized as follows. Section 2 discusses our motivation. Section 3 presents the Proposed Vulnerability Prediction model in this paper. Section 4 presents result and analysis to assess their accuracy based on the proposed hybrid attributes. Section 5 discusses related work. Section 6 presents the summary and concludes our work.

2. Motivation

2.1 Cloud Security Attacks

Cloud computing is still evolving and in continual development, today many software organizations and educational institutions are moving to cloud because they gets the benefits of cloud resources and utilizing distributed computing. Even though peoples enjoy with cloud computing benefits; the security issues in cloud is a key challenges. nowadays hackers are continues to find the tricky techniques to exploits and steal the user's credentials without the user's knowledge, Malicious users can inject a malicious code to users websites and gains the unauthorized access to users confidential information stored in cloud. A number of vulnerabilities are remains in cloud computing environment. As per the recent Survey done by OWASP reveals that SQL injection and Cross Site Scripting (XSS) are two of the most serious vulnerabilities in cloud based applications today. these two SQLI and XSS vulnerabilities are occurs due to the improper handling of user's inputs in cloud based web application programs.SQL injections access the cloud database servers through SQL language, malicious users inject a malicious code in order gain unauthorized access to cloud server databases. In the Figure 1 depicts the SQL injection scenario in cloud environment. Cross-Site Scripting is to defraud the sensitive credentials such as transaction passwords, credit card numbers, etc. of the online user. As shown in Figure 2 represents the Cross-Site Scripting (XSS) scenario in Cloud environment.
2.2 Vulnerability Trends in Cloud Computing

In order to mitigate the security attacks like SQLI and XSS cloud service providers adopt with various methods and tools, Some of them are keeping away from powerfully produced SQL explanations in code that approve and clean client entered values. As per Symantec MessageLabs Web Security. In Cloud engineering the security of cloud framework lies in two parts a) Multi Layer Security b) URL sifting. Multi Layer Security expects to square malwares and URL sifting lessens means to withdraw from destructive site pages and to ensure data transmission. The study reveals that SQLI is most found shortcoming in SaaS and recommends some bland answers for stay away from SQL Infusion like utilize parameterized questions, approve client contributions by utilizing inbuilt capacities and client characterized work and by utilizing put away methodology.

Helplessness Forecast: Powerlessness expectation in view of examining the code or by gathering properties portrays the weakness and building forecast models goes under this zone. In this space specialists relate programming qualities, for example, static code properties with imperfections. The generally utilized static code characteristics are Halstead unpredictability measures and cyclomatic, many-sided quality properties, Lines of Code and so forth. In Figure 3 shows that the number of XSS and SQL injection by years from 2012 to 2016, As predicted, the generalize of XSS and SQL injection vulnerabilities is also noticeable in the Figure 4, As shown in Figure 4 it depicts evolution of SaaS vulnerabilities over the years by types from 2012 to 2017.

3. Proposed Vulnerability Prediction Model

The Proposed Vulnerability Prediction model provide the hybrid attributes such as static and dynamic approaches that based on classifications. according to the research gap identified in the existing approaches are very challenging task in order to amalgamate with cloud based saas applications. Moreover, we ascertained in existing approaches for prediction of XSS and SQLI Vulnerabilities originated a High rate of false positive and false negatives. In order to improve our Proposed Vulnerability Prediction model we proposed a hybrid attributes such as dynamic attributes to complement static attributes in the prediction of vulnerabilities in cloud based applications. By efficiently using the developed prototype tool for data collection and its used to evaluate our models on six open source cloud based applications. this proposed prediction model has categorized in to two aspects. one is for static and another one is dynamic approaches for collecting the attributes in the web application code. In the static code attributes can be collected by using the program analyzer open source tool called pixy, The dynamic code attributes can be collected by using pattern mining approach, these two
Automated Vulnerability Detection and Prediction by Security Testing for Cloud SAAS

3.1 Hybrid Attributes

The proposed Hybrid approaches based on the static and dynamic analyzers the attribute vector of each sinks in cloud based web application files. A sink denote the node in a control flow graph that may create SQLI or XSS attacks, its consist of two sink SQL sink for database and HTML sink for web client which represents a program statement of a target cloud based web programs. As shown in Table 1 represents the proposed hybrid attributes such as static and dynamic attributes.

3.2 Data Pre-Processing

The Data pre-processing to assure that our proposed prediction model is invariable among datasets with different dispensations. Our proposed predictor must knob the different dispensations to hypothesize the results. We intended to used 22 hybrid attributes boycott the target attribute. In this 22 attributes 20 attributes take on numeric values and two attributes are binary. We take the advantage of min–max data normalization techniques to systematize our data. All the training and testing numeric attributes to be normalized. Our prediction model used for zero to one range for prediction.

3.3 Data Reduction

To improve the predictive act of our proposed model we used attribute ranking technique to analyze that attributes that are utmost significant and prove the excepting the attributes are less significant. This data reduction process is called attribute ranking or feature selection techniques. In As per the survey states that there are many techniques have been used by the researcher for attribute ranking or feature selection process. These techniques are such as chi-square, pick up proportion, data pick up, gain ratio etc, we used gain ratio discover the best quality attributes to improve the classifier’s performance.

3.4 Classifiers

The proposed vulnerability prediction model builds on machine learning based classifiers, which classifiers learnt from such data be accurate at vulnerability predictions. These classifiers designate a sensitive sink to a target code that is either vulnerable or non-vulnerable. The data mining tool WEKA is used to implement the classifiers. There are many number of classification algorithms available. In this experiment we used C 4.5 and Multi-Layer Perceptron (MLP) classification algorithms.

3.5 Model Training and Testing

In this Proposed work, the dataset is divided into 10 fold cross validation is known as standard sampling method its used for training and testing the two classifiers. The classifier is trained on the 9 folds and then tested on the resting fold, this process is repeated 10 times, to evaluate the classifier effectiveness to predict the vulnerabilities.

3.6 Performance Measures

In order to figure out the best vulnerability predictors, we computed recall or probability of detection (pd), probability of false alarm (pf), and precision (pr). Three indicators that are used to determine the performance based on the prediction influence.

- **Accuracy** is measure the sum of True Positives (TP) and True Negatives (TN) over the total number of classifications.
- **Precision** is measure the ratio PR= TP/(TP + FP), where FP is the number of false positive it measures the real vulnerable classes that are accurately predicted in terms of a percentage of total number of sinks predicted as vulnerable.
- **Recall** It is measure the ratio PD=TP/(TP + FN), where FN is the number of false negatives i.e., measures how good our prediction model is in finding actual vulnerable sink.

4. Result and Analysis

In order to figure out our proposed approach used to establish the investigation with open source cloud based applications with XSS and SQLI vulnerabilities. There are two vulnerability prediction models for predicting SQLI vulnerabilities and XSS vulnerabilities using the collected data. We estimate the two different types of classifiers on the data sets with Vulnerable or Not-Vulnerable. For our experiments the dataset are used from six cloud based applications, they are tadalist, dphoto, flickr, yapig, me, Schoolmate, In Table 2 represents the Datasets collected from cloud based applications, and Table 3 depicts the
accomplishment of SQLI vulnerability predictors and accomplishment of XSS predictors, the overall results were averaged. The data mining tool WEKA is used to implement the classifiers. There are many number of classification algorithms available. In this experiment we used C 4.5 and Multi-Layer Perceptron (MLP) classification algorithms. Based on the experiments average results states that the best predictor is MLP which achieved a pd=72%, pf=12% for predicting SQL Injection and pd=82%, pf=19% for predicting XSS Cross Site Scripting. The results of our experiments states that our prediction model is an ease and effective technique to predict and detect the SQLI and XSS vulnerabilities in Cloud applications.

5. Related Work

This section prospects our work is related to vulnerability prediction, and vulnerability detection approaches. The vulnerability prediction and detection approaches use the machine learning techniques to finding SQLI and XSS vulnerabilities in the target web applications. In
Shar et al. predicted vulnerabilities in open source PHP web applications. They proposed static code attributes approaches to predict the SQLI and XSS vulnerabilities in PHP based web applications, they constructs the prediction model based on machine learning algorithms such as C 4.5, NB and MLP ,they achieved Recall 86% and pf 3%. In Walden et al. used PHP tokens and software metrics (i.e. LOC, cyclomatic complexity) to predict vulnerability in PHP-MyAdmin, Moodle and Drupal CMS applications. also they distinguished the software metrics and text mining features and ascertained that text-mining features provide considerably excelling accomplishment in prediction of XSS vulnerabilities. they constructs the prediction model based on Random Forest machine learning algorithm, they attained Recall 80.5% and Accuracy: 75.4%, In Shin et al. used software metrics such as code complexity, code churn, and developer activity metrics to predict vulnerability in Mozilla Firefox Web Browser, Red Hat Enterprise Linux kernel constructs the prediction model based on machine learning algorithms such as Logistic regression, J48, Random forest, NB, Bayesian network and they achieved Recall: 80 %. Likewise in Chowdhury et al. used complexity, cohesion and coupling metrics to predict vulnerability-prone files in Mozilla Firefox. they assemble the prediction model based on machine learning algorithms such as Logistic regression, C4.5, Random forest, NB they accomplished

### Table 2. Cloud apps- data sets

| Test DataSet | #HTMLSink | %Vul XSS | #DBSink | %Vul SQLI |
|--------------|-----------|----------|---------|-----------|
| tadalist     | 120       | 112      | 145     | 132       |
| dphoto       | 64        | 34       | 72      | 56        |
| flickr       | 133       | 28       | 127     | 67        |
| yapig        | 20        | 40       | 32      | 14        |
| me           | 73        | 22       | 63      | 35        |
| Schoolmate   | 172       | 138      | 167     | 152       |

### Table 3. Prediction accomplishment of XSS and SQLI

| DataSet      | Classifier | pd | pf | pr | Acc | pd | pf | pr | Acc |
|--------------|------------|----|----|----|-----|----|----|----|-----|
| XSS Prediction Result |            |    |    |    |     |    |    |    |     |
| tadalist     | C4.5       | 48 | 9  | 34 | 75  | 70 | 20 | 75 | 78  |
|              | MLP        | 82 | 85 | 76 | 90  | 72 | 12 | 72 | 92  |
| dphoto       | C4.5       | 94 | 5  | 84 | 95  | 57 | 8  | 52 | 88  |
|              | MLP        | 85 | 5  | 84 | 93  | 50 | 12 | 75 | 85  |
| flickr       | C4.5       | 60 | 1  | 60 | 60  | 75 | 20 | 75 | 78  |
|              | MLP        | 70 | 20 | 58 | 60  | 100| 20 | 80 | 89  |
| yapig        | C4.5       | 97 | 1  | 97 | 99  | 50 | 12 | 75 | 85  |
|              | MLP        | 98 | 3  | 95 | 98  | 60 | 8  | 50 | 97  |
| me           | C4.5       | 78 | 32 | 80 | 94  | 50 | 12 | 75 | 85  |
|              | MLP        | 93 | 20 | 50 | 82  | 57 | 8  | 50 | 88  |
| Schoolmate   | C4.5       | 76 | 5  | 99 | 79  | 58 | 15 | 75 | 83  |
|              | MLP        | 99 | 2  | 100| 99  | 57 | 8  | 50 | 88  |
| Average      | C4.5       | 78 | 23 | 79 | 84  | 58 | 15 | 75 | 83  |
|              | MLP        | 82 | 19 | 72 | 79  | 72 | 72 | 12 | 91  |
| SQLI Prediction Result |            |    |    |    |     |    |    |    |     |
| tadalist     | C4.5       | 48 | 9  | 34 | 75  | 70 | 20 | 75 | 78  |
|              | MLP        | 82 | 85 | 76 | 90  | 72 | 12 | 72 | 92  |
| dphoto       | C4.5       | 94 | 5  | 84 | 95  | 57 | 8  | 52 | 88  |
|              | MLP        | 85 | 5  | 84 | 93  | 50 | 12 | 75 | 85  |
| flickr       | C4.5       | 60 | 1  | 60 | 60  | 75 | 20 | 75 | 78  |
|              | MLP        | 70 | 20 | 58 | 60  | 100| 20 | 80 | 89  |
| yapig        | C4.5       | 97 | 1  | 97 | 99  | 50 | 12 | 75 | 85  |
|              | MLP        | 98 | 3  | 95 | 98  | 60 | 8  | 50 | 97  |
| me           | C4.5       | 78 | 32 | 80 | 94  | 50 | 12 | 75 | 85  |
|              | MLP        | 93 | 20 | 50 | 82  | 57 | 8  | 50 | 88  |
| Schoolmate   | C4.5       | 76 | 5  | 99 | 79  | 58 | 15 | 75 | 83  |
|              | MLP        | 99 | 2  | 100| 99  | 57 | 8  | 50 | 88  |
| Average      | C4.5       | 78 | 23 | 79 | 84  | 58 | 15 | 75 | 83  |
|              | MLP        | 82 | 19 | 72 | 79  | 72 | 72 | 12 | 91  |
6. Summary and Conclusion

This paper Proposed an automated vulnerability detection and prediction by security testing for cloud saas web applications. This Vulnerability Prediction model provide the hybrid attributes such as static and dynamic approaches that based on classifications. according to the research gap identified in the existing approaches are very challenging task in order to amalgamate with cloud based saas applications, Moreover, we ascertained in existing approaches for prediction of XSS and SQLI Vulnerabilities originated a High rate of false positive and false negatives. In order to improve our Proposed Vulnerability Prediction model we proposed a hybrid attributes such as dynamic attributes to complement static attributes in the prediction of vulnerabilities in cloud based applications. Based on the experiments average results states that the best predictor is MLP which achieved a pd = 72%, pf = 12% for predicting SQL Injection and pd = 82%, pf = 19% for predicting XSS Cross Site Scripting. The results of our experiments states that our prediction model is an ease and effective technique to predict and detect the SQLI and XSS vulnerabilities in Cloud applications.

7. References

1. OWASP. The open web application security project. Available from: Crossref. Date Accessed: 2012 Jan.
2. Jovanovic N, Kruegel C, Kirda E. Pixy: A static analysis tool for detecting web application vulnerabilities. IEEE Symposium on Security and Privacy; 2006. p. 258–63. Crossref
3. Xie Y, Aiken A. Static detection of security vulnerabilities in scripting languages. USENIX Security Symposium; 2006. p. 179–92.
4. Arisholm E, Briand LC, Johannessen EB. A systematic and comprehensive investigation of methods to build and evaluate fault prediction models. Journal of Systems and Software. 2010; 83(1):2–17. Crossref
5. Demsar J. Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research. 2006; 7:1–30.
6. Fagerland MW, Sandvik L. Performance of five two-sample location tests for skewed distributions with unequal variances. Contemporary Clinical Trials. 2009; 30(5):490–6. Crossref PMid:19577012
7. Ferrante J, Ottenstein KJ, Warren JD. The program dependence graph and its use in optimization. ACM Transactions on Programming Languages and Systems; 1987. p. 319–49. Crossref
8. Fisher D, Xu L, Zard N. Ordering effects in clustering. Proceedings of the 9th International Workshop on Machine Learning, Aberdeen, Scotland; 1992. p.163–8. Crossref
9. Fogie S, Grossman J, Hansen R, Rager A. XSS exploits: cross site scripting attacks and defense. Syngress; 2007. p. 395–406.
10. Gao K, Khoshgoftaar TM, Wang H, Seliya N. Choosing software metrics for defect prediction: an investigation on feature selection techniques. Software Practice and Experience. 2011; 41(5):579–606. Crossref
11. Gegick M, Williams L, Osborne J, Vouk M. Prioritizing software security fortification through code-level metrics. Proceedings of the 4th ACM Workshop on Quality of Protection, Alexandria, Virginia; 2008. p. 31–8. Crossref
12. Jovanovic N, Kruegel C, KirdaE. Pixy: A static analysis tool for detecting web application vulnerabilities. Proceedings of the IEEE Symposium on Security and Privacy, Berkeley/ Oakland, CA; 2006. p. 258–63. Crossref
13. Scandariato R, Walden J, Hovsepyan A, Joosen W. Predicting vulnerable software components via text mining. IEEE Transactions on Software Engineering. 2014 Oct; 40(10):993–1006. Crossref
14. Vijayalakshmi K, Zohra KSF, Leema AA. SC-TPDP protocol to secure multi-cloud storage from XSS attacks. Indian Journal of Science and Technology. 2016 May; 9(19):1–7. Crossref
15. Srinivasan S, Raja K. Preventing cloud attacks using biometric authentication in cloud computing. Indian Journal of Science and Technology. 2016 Jun; 9(23):1–9.

16. Shar LK, Tan HBK. Mining input sanitization patterns for predicting SQL injection and cross site scripting vulnerabilities. International Conference on Software Engineering; 2012. p. 1293–6. Crossref

17. Walden J, Stuckman J, Scandariato R. Predicting vulnerable components: Software metrics vs text mining. IEEE 25th International Symposium on Software Reliability Engineering (ISSRE). 2014 Nov; p. 23–334.

18. Shin, Meneely A, Williams L, Osborne JA. Evaluating complexity, code churn, and developer activity metrics as indicators of software vulnerabilities. IEEE Transactions on Software Engineering. 2011 Nov; 37(6):772–87. Crossref

19. Chowdhury I, Zulkernine M. Using complexity, coupling, and cohesion metrics as early indicators of vulnerabilities. Journal of Systems Architecture. 2011; 57(3):294–313. Crossref

20. Scandariato R, Walden J, Hovsepyan A, Joosen W. Predicting vulnerable software components via text mining. IEEE Transactions on Software Engineering. 2014 Oct; 40(10):993–1006. Crossref