IDENTIFYING THE CROSS SITE SCRIPTING (XSS) ATTACK USING XSSER TOOL AND DETECTION USING SUPERVISED LEARNING ALGORITHM

Ms.G.Nivetha, Assistant Professor, Department of Computer Science, Rathinam College of Arts and Science
Dr.C.Meena, Professor, Department of Computer Science, Avinashilingam Institute for Home science and Higher Education for Women Coimbatore.

ABSTRACT

One of the most common assaults detected in the web application is cross-site scripting (XSS). With XSS, hackers can quickly insert harmful code into the web server or web page. They can steal cookies, session tokens, and other sensitive data from the victim visiting the online application by using malicious code. They can also change the website's content. The objective of the project is to detect the Cross Site Scripting (XSS) Attack. Making a sample website, testing it with the XSSer tool, and classifying the results as a dataset are the steps taken to detect an XSS attack. The feature selection is carried out using correlation filter-based features, and the detection is done using machine learning algorithms like XGBoost, Decision Tree, K-Nearest Neighbor(KNN), Naive Bayes, and AdaBoost Classifier. Python-Jupyter Notebook and the XSSer tool were used to create this project. Comparing the Adaboost Classifier to the other four algorithms, it provides the highest level of detection accuracy.

Keywords: *Cross- site scripting, XGBoost, KNN, Naïve Baves and AdaBoost.*

1.1 DEFINITION OF THE PROBLEM

Programming code was used to create every component of the system. The internet is no exception to this. The web browser downloads the developer's code each time a user accesses a specific website, interprets it, and then displays the results. Together, they enable the entire web experience. JavaScript code can do anything with a web page, including changing the content, tracking what users do, changing the location where passwords are delivered, and much more. This is not a problem because the only person who can visit the website is the owner who published the code. For instance, only Google's code is capable of altering google.com. Up until cross-site scripting (XSS) occurs, everything is fine. The procedure for a hacker introducing code is shown in Figure 1.

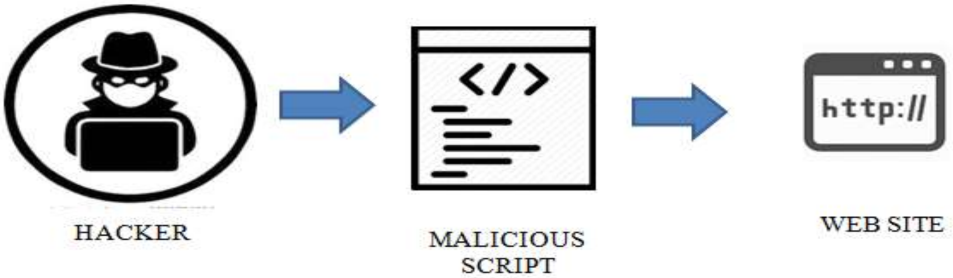


Figure 1: Process of hacker injecting the code.

A hacker's own Java script can be executed thanks to the XSS vulnerability. When a web website outputs user input without first sanitising it, an XSS vulnerability happens. The search form would be one of many instances. JavaScript code applied to user input causes it to be run on the page. All the attacker needs to do is send the victim the specially created URL. The hacker has access to the victim's messages sent to friends, the password used to log in the following time, and much more. The web as we know it today requires JAVASCRIPT, but when a web page permits JAVASCRIPT by anyone, it rapidly turns dangerous. In simple terms, “XSS is the ability to execute your own JAVASCRIPT code under someone else’s domain”.

1.2 OVERVIEW OF THE PROJECT

The project's goal is to find Cross Site Scripting (XSS) Attacks. Create a test website, test it with the XSSer tool, and classify the results as a dataset. Then, choose features based on correlation filters, and use machine learning algorithms like XGBoost, Decision Tree, K-Nearest Neighbor (KNN), Naive Bayes, and AdaBoost Algorithms to detect XSS attacks. When compared to the other four algorithms, the Adaboost method provides the highest level of accuracy when detecting.

2.1 EXISTING SYSTEM

The objective is to improve the performance, especially recall rate, of the machine learning model for the detection of vulnerable code block samples, that is, to find as many vulnerabilities in source code samples as possible, based on the practical significance of the source code vulnerability audit and the thorough consideration of the current mainstream research results. The three possible outcomes of the filter-sink source code sequence are as follows:

(1) User input is output without any filtering.

(2) User input is filtered to some level, but the filtering is insufficient or the filtering function is used incorrectly. This is the most likely situation to cause the XSS vulnerability. This strategy can occasionally stop some XSS attacks, but if the attacker employs a sophisticated bypass technique, it can get over the filter function's limitations and exploit the XSS vulnerability.

(3) The code establishes overly severe filtering functions and fully takes into account all of the relevant output's filtering requirements. An attacker cannot use the XSS vulnerability in this situation. But in actual web applications, this kind of circumstance is really uncommon.

2.2 PROPOSED METHOD

Machine learning algorithms and classifiers have been compared in previous studies. To determine the optimal technique for classifying, detecting, and preventing the XSS attack, accuracy is also analysed. The suggested approach for the project is to build a custom test website, test it using the XSSer tool (one of the Kali Linux penetration testing tools for Cross-Site Scripting (XSS) Attack), and then classify the report as a dataset. The dataset created has 9 instances and 35 features; features are selected using a correlation filter, and after removing duplicates, we are left with 9 instances and 31 features. The XSS dataset is processed without duplicate features thanks to feature selection and detection. Machine learning algorithms including XGBoost, Decision Tree, K-Nearest Neighbor (KNN), Naive Bayes, and AdaBoost Classifier are used to detect XSS attacks. The algorithms were compared to get the maximum level of accuracy.

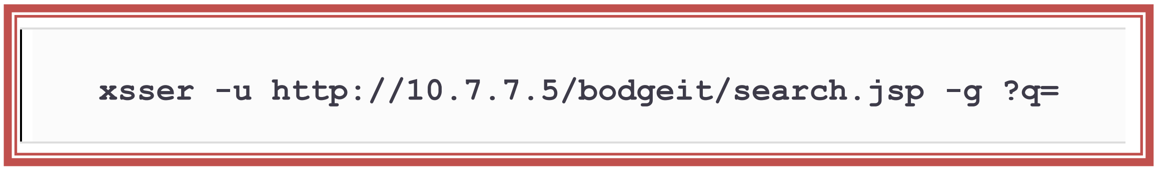
3. **SYSTEM DESIGN AND METHODOLOGY**

3.1 CREATING WEBSITE

Sri Kalpatharu & Sons Culturals Private Limited Company built a website to detect XSS attacks. They also offered permission to test the website and to develop a website using their product for real-time website testing. Additionally, some sample websites for testing were made using Wix.com.

3.2 WEB PENETRATION TESTING WITH KALI LINUX

In this project, the constructed website is tested using the Kali Linux XSSer web penetration testing tool. An automatic framework called Cross Site "Scripter" (XSSer) is created to find, use, and report XSS vulnerabilities in web-based applications. It comes with Kali Linux. Give the following command in Kali Linux's Terminal to accomplish that:



```
xsser -u http://10.7.7.5/bodgeit/search.jsp -g ?q=
```

3.3 DATASET DESCRIPTION

After evaluating the websites with the XSSer tool, a dataset of the results was created. It's known as an XSS dataset. It has 35 features and 9 instances.

3.4 FEATURE SELECTION

Real-world issues in data mining and machine learning frequently entail numerous attributes. By choosing only a limited subset of pertinent features from the initial, substantial amount of features, feature selection seeks to address this issue. Feature selection can decrease the dimensionality of the data, speed up learning, decompose the learned model, and/or improve performance by deleting unnecessary and duplicate features.

Feature selection's key advantage is that it lessens overfitting. By eliminating unnecessary information, it enables the model to concentrate only on the crucial aspects of the data and avoid getting bogged down by unimportant details. The accuracy of the model's predictions is increased as a result of the removal of irrelevant data. Additionally, it cuts down on the amount of time needed to compute the model. Last but not least, having fewer features makes your model easier to grasp and comprehend. Overall, the ability to forecast values with any degree of accuracy depends on feature selection. Three categories of feature selection strategies are shown. The following are available: Wrapper method, Embedded method and Filter method.

3.5.1 WRAPPER METHOD

Users attempt to employ a subset of features and train a model utilising them in wrapper approaches. Essentially, the issue is simplified to a search issue. These techniques are typically highly expensive to compute. Forward, Backward, Stepwise, and Recursive Feature Elimination Method are a few examples of wrapper methods.

3.5.2 EMBEDDED METHOD

Filter and wrapper techniques' advantages are combined in embedded methods. It is carried out by algorithms with built-in feature selection techniques. LASSO for Variable Selection and Random Forest for Variable Selection are two examples of the embedded technique.

3.5.3 FILTER METHOD

In most cases, filter methods are utilised as a pre-processing step. Any machine learning methods have no influence on the feature selection process. Instead, features are picked based on how well they perform in various statistical tests to determine how closely they correlate with the result variable. Correlation, hypothesis testing, and information gain for variable selection are a few examples of filter approaches.

This project makes use of filter-based feature selection. Features are removed from the data before learning since these approaches only take into account the traits of these variables. These techniques are typically the first step in any feature selection pipeline since they are effective, straightforward, and speed up the removal of features.

3.6 CORRELATION FILTER METHODS

The linear link between two quantitative variables, such as height and weight, is measured by correlation. Correlation is a metric that expresses how much one variable depends on another.When two variables are highly correlated, we may predict one from the other, which is a beneficial quality. Therefore, for all characteristics, especially for linear machine learning models, look for features that are substantially linked with the aim.However, if two variables have a high degree of correlation, they give the target redundant information. Essentially, choose just one redundant variable and use it to produce an accurate prediction about the objective.The removal of the second variable in these situations can help to decrease the dimensionality as well as the additional noise since it doesn't provide any new information. The correlation between variables can be determined using a variety of techniques.

3.7 MACHINE LEARNING ALGORITHM

Five detecting methods are employed in this study. These algorithms are XGBoost, Decision Tree, K-Nearest Neighbor (KNN), Naive Bayes, and AdaBoost.

A group of techniques known as machine learning are used to automatically build models out of data. The algorithms that transform a data set into a model are known as machine learning algorithms, and they are the heart of machine learning. According to the method of issue resolution, machine learning algorithms are typically divided into supervised (classification, regression), and unsupervised (clustering).

3.8. XGBOOST ALGORITHM

Extreme Gradient Boosting is the abbreviation for XGBoost. XGBoost is a distributed gradient boosting library that has been developed to be very effective, adaptable, and portable. It uses the Gradient Boosting framework to implement machine learning algorithms. It offers a parallel tree boosting to quickly and accurately address a variety of data science challenges. Due to its scalability, it has recently gained popularity and is now winning Kaggle competitions for structured data and applied machine learning. XGBoost was created specifically to increase performance and speed.

3.9. DECISION TREE ALGORITHM

A supervised learning method called a decision tree can be used to solve classification and regression problems, but it is typically favoured for doing so. It is a tree-structured classifier, where internal nodes stand in for a dataset's features, branches for the decision-making process, and each leaf node for the classification result.

3.10. **K-NEAREST NEIGHBOR (KNN) ALGORITHM**

The K-NN algorithm saves all the information that is accessible and categorises new data points based on similarity. This means that utilising the K-NN method, fresh data can be quickly and accurately sorted into a suitable category.

Although the K-NN approach can be used for both classification and regression problems, classification challenges are where it is most frequently applied.

3.11. NAIVE BAYES ALGORITHM

The Naive Bayes algorithm is a supervised learning method for classification issues that is based on the Bayes theorem. It is mostly employed in text categorization with a large training set. One of the most straightforward and efficient classification algorithms is the Naive Bayes Classifier, which aids in creating quick machine learning models that can generate accurate predictions. Being a probabilistic classifier, it makes predictions based on the likelihood that an object will occur. Spam filtration, Sentimental analysis, and article classification are a few examples of Naive Bayes algorithms that are frequently used.

3.12. ADABOOST ALGORITHM

Yoav Freund and Robert Schapire proposed the ensemble boosting classifier known as Ada-boost or Adaptive Boosting in 1996. To improve classifier accuracy, it combines several classifiers. An iterative ensemble algorithm is AdaBoost. The AdaBoost classifier combines several weak classifiers to create a strong classifier that has the highest accuracy possible. The fundamental idea underlying Adaboost is to train the data sample and set the classifier weights in each iteration so that they can accurately predict uncommon observations.

4. RESULT AND DISCUSSION

By using the XSSer tool to test multiple websites, the database was built. Wix.com, an open-source and simple to use website builder, was used to help generate some of the sample websites. The company was given permission by Sri Kalpatharu & Sons Culturals Private Limited Company to design a real-time website for the business and to test the created website alongside their original website. After developing the website, utilising one of the Kali Linux penetrations testing tool known as XSSer is used to test the website and result is taken as the dataset, fully 9 instances and 35 features are there. On the website, data preparation is carried out. The correlation algorithm is used for feature selection. Correlation is a filter-based technique that aids in getting rid of extraneous columns. In the dataset 3 column feature were duplicated columns after eliminating them 31 significant features and 9 instances are present in the dataset. Algorithms for machine learning were utilised for detection. Five different algorithms have been implemented in this project. These algorithms include XGBoost, Decision tree, KNN, Naive Bayes, and AdaBoost. Adaboost's algorithm provided the best level of accuracy (91%).

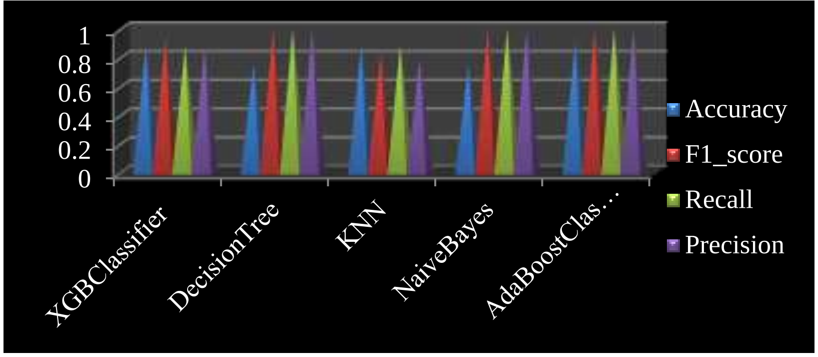


Figure 8: Comparison of all algorithms.

5. CONCLUSION

One of the most common assaults detected in the web application is cross-site scripting (XSS). With XSS, hackers can quickly insert harmful code into the web server or web page. They can steal cookies, session tokens, and other sensitive data from the victim visiting the online application by using malicious code. They can also change the website's content. The main goal of this project is to identify the XSS attack. A prototype website was made, tested using the XSSer tool, and the report was categorised as a dataset. The features were selected using a correlation filter, which produced three duplicate columns.31 features are found to be significant when duplicate columns in the dataset are removed. Machine learning algorithms including XGBoost, Decision Tree, K-Nearest Neighbor (KNN), Naive Bayes, and AdaBoost Classifier are used to implement detection. Comparing the Adaboost Classifier to the other four algorithms, it provides the highest level of detection accuracy. Adaboost's accuracy rating of 91% was the highest. The future scope of the project is to test the dataset with various algorithms. Apply various feature selection techniques and detecting algorithms to the dataset in the future. Should also test more websites with the XSSer tool and create a robust, balanced dataset.

REFERENCE

1) Chenghao Li 1,†, Yiding Wang 1,†, Changwei Miao 1,† and Cheng Huang "**Cross-Site Scripting Guardian: A Static XSS Detector Based on Data Stream Input-Output Association Mining**" ,College of Cybersecurity, Sichuan University, Chengdu 610064, China; Guangxi Key Laboratory of Cryptography and Information Security, Guilin 541004, China Correspondence: codesec@scu.edu.cn.
2) Chaya P ,Anjali N Menon , Madhurya Aithal A ,Pratheeksha J , Varsha S , "**Detection of Cross Site Scripting Attack using MONOSEK**", Department of Information Science and Engineering GSSS Institute of Engineering and Technology for Women Karnataka-India.
3) Guowei Dong1, YanZhang2,Xin Wang1,Peng Wang2, Liangkun Liu2 1,"**Detecting Cross Site Scripting Vulnerabilities Introduced by HTML5**", China Information Technology Security Evaluation Center, China 2 School of Information, Renmin University of China, China dgw2008@163.com,annazhang@ruc.edu.cn.
4) Chih-Hung Wang , Yi-Shauin Zhou , "**A New Cross-site Scripting Detection Mechanism Integrated with HTML5 and CORS Properties by Using Browser Extensions**", Dept. Computer Science and Information Engineering National Chiayi University Chiayi, Taiwan wangch@mail.ncyu.edu.tw , lzyworkuse@gmail.com