XSS ATTACK DETECTION

Batch-09:**V.Vyshnavi-393, V. Shanker-394, V. Sai Sreya-395, Kranti Kumar J-396**

Department of AIML, School of Engineering, MALLA REDDY UNIVERSITY, 500100.

Abstract

Cross-site scripting (XSS) is an attack in which an attacker injects malicious executable scripts into the code of a trusted application or website.With the increased use of the internet, web applications and websites are becoming more and more common. cyber-attacks on web applications and websites are also increasing. Of all the different types of cyber-attacks on web applications and websites, XSS (Cross-Site Scripting) attacks are one of the most common forms of attack. XSS attacks are a major problem in web security and ranked as number two web application security risks in the OWASP (Open Web Application Security Project). Traditional methods of defence against XSS attacks include hardware and software- based web application firewalls, most of which are rule and signature-based. Rule-based and signature-based web application firewalls can be bypassed by obfuscating the attack payloads. As such, rule-based and signature based web application firewalls are not effective against detecting XSS attacks for payloads designed to bypass web application firewalls. This project aims to use Deep learning to detect XSS attacks using RFC in detecting XSS attacksin web applications and websites.

Introduction

XSS attack is a subset of injection attack wherein an attacker injects a malicious code (also known as malicious payload) into the contents of a legitimate and trusted websites in order to gain accesscontrol of the viewer’s system. XSS attack can be executed on any vulnerable website be it the one written in HTML code, JavaScript, VBScript, PHP, etc.

Step 1: Attacker finds a vulnerable website which allows the injection of untrusted malicious code into its webpage.

Step 2: Attacker inserts malicious client-side JavaScript/ActiveX/VBScript/HTML code on the web application. This code is either sent to the victim’s web browser or the web server depending upon the type of XSS attack.

Step 3: User click on the malicious link either while visiting the website or accessing service from web server.

Step 4: Attacker has access to private credentials or details of the victim through vulnerable website by bypassing the SOP (Same Origin Policy).

XSS attacks occur when an attacker uses a web application to send malicious code, generally in the form of a browser side script, to a different end user. Flaws that allow these attacks to succeed are quitewidespread and occur anywhere a web application uses input from a user within the output it generates without validating or encoding it. An attacker can use XSS to send a malicious script to an unsuspecting user, the malicious script can access any cookies, session tokens, or other sensitive information retained by the browser and used with that site. These scripts can even rewrite the content of the HTML page.

Mathematical Modelling

The accuracy of an ML model is measuredbased on TP (True Positive), TN (True Negative), FP (False Positive), and FN (False Negative). TP is the number of correctly classified positive samples; TN negative is the number of correctly classified negative samples; FP is the number of samples that are incorrectly classified as positive and FN is the number of samples that are incorrectly classified asnegative. Concerning our problem, TP is the number of samples correctly classified as XSS attacks, and TN is the number of samples correctly classified as benigninputs.

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \qquad (1)$$

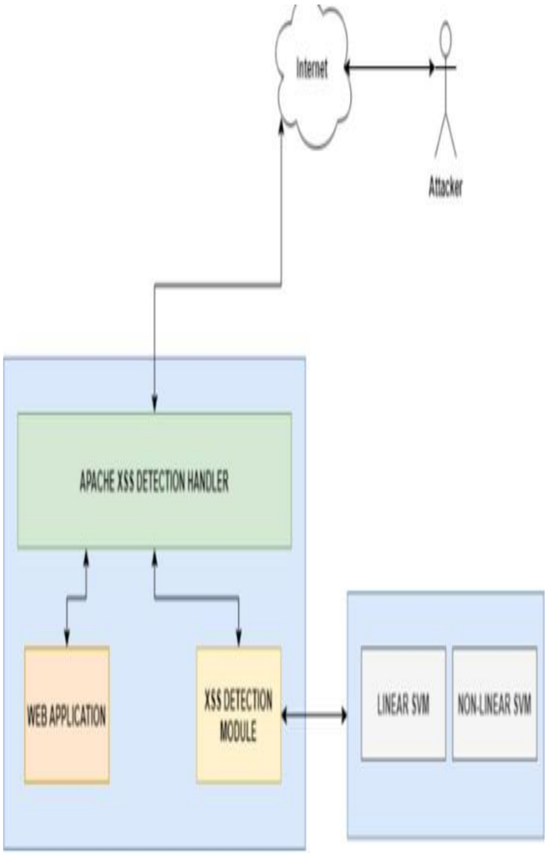
$$Recall = \frac{TP}{(TP + FN)} \qquad (2)$$

$$Precision = \frac{TP}{(TP + FP)} \qquad (3)$$

$$F1 = \frac{(2 \times Precision \times Recall)}{(Precision + Recall)} \qquad (4)$$

Metric	**Value**
Precision	0.99
Recall	0.97
F1 Score	0.98

Architecture



Methods and Algorithms

Random Forest Classifier: The Random forest or Random Decision Forest is a supervised Machine Learning algorithm used for classification, regression, and other tasks using decision trees. The Random forest classifier creates a set of decision trees from a randomly selectedsubset of the training set. It is basically a set of decision tree(DT) from a randomly selected subset of the training set and then it collects the votes from different decision trees to decide the final prediction.

Support Vector Classifier: Support Vector Machines (SVM) is a widely used supervised learning method and it can be used for regression, classification, anomaly detection problems. The SVM based classier is called the SVC (Support

Vector Classifier) and we can use it in classification problems. It uses the C regularization parameter to optimize the margin in hyper-plane and it is also called C-SVC. The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyper-plane. SVM chooses the extreme points/vectors that help in creating the hyper-plane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyper-plane.

Logistic Regression: Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables. Logistic regression predicts the output of a categorical dependent variable.Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.Logistic Regression is much similar to the Linear Regression except that how they are used.

Decision Tree Classifier: In the Decision Tree classifier, first we compute the entropy of our database. It tells us the amount of uncertainty of our database. The smaller the uncertainty value, the better is the classification results. Each feature's information gain is calculated. This then tells us how much uncertainty reduces after spitting the database. Finally, all the information gain is calculated for all features, and now, we split the database which has high information gain. The process repeats until all nodes are cleared.

Results

The results for an XSS web attacks project can include a survey on detection and prevention of cross-site scripting attacks, which is considered one of the top 10 web application vulnerabilities of 2013 by the Open Web Application Security Project:

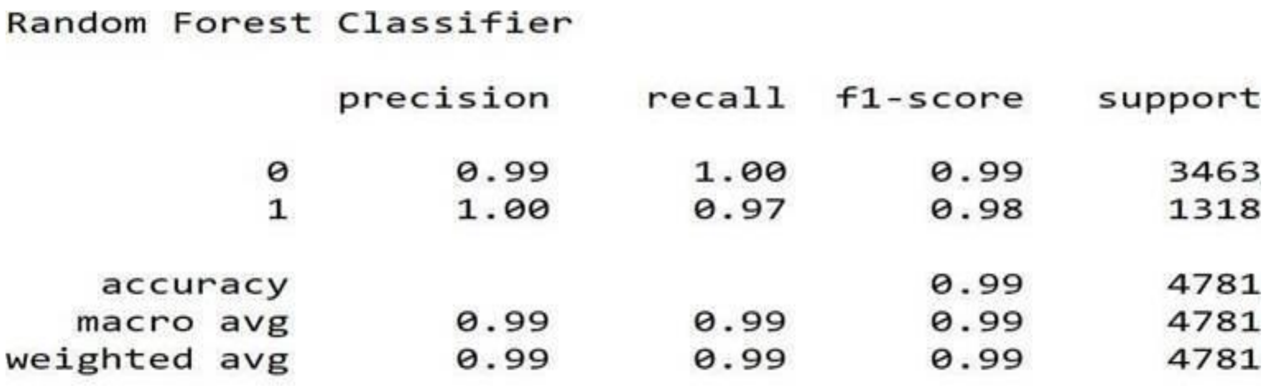
1. A study of existing cross-site scripting detection and techniques can be conducted to understand the ways to detect and prevent XSS attacks.

2. Additionally, a machine learning approach can be used to detect XSS attacks using various machine learning algorithms

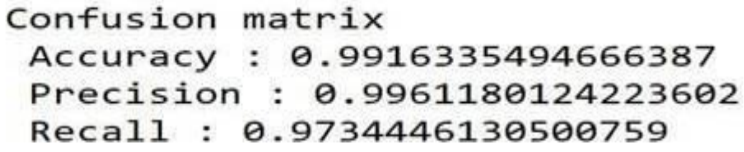
3. A review on detection of cross-site scripting attacks (XSS) in web security can be conducted to understand the techniques used for detecting XSS attacks.

4. The project can also explore what XSS attacks are, examples of popular attacks, and ways to detect and prevent them.

We will be extending our system to detect more attacks and we will also work to improve its accuracy. We will also make this library robust and scalable so that it can be integrated easily within various applications.

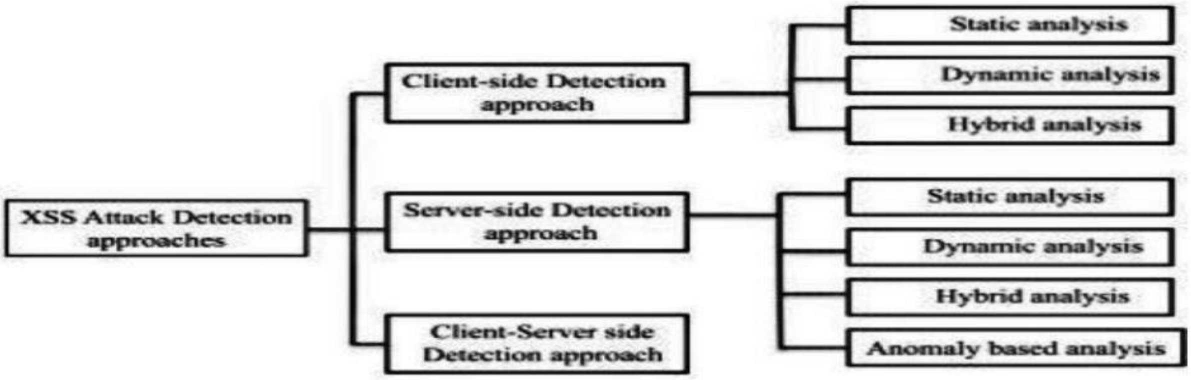


```
Random Forest Classifier

              precision    recall  f1-score   support

           0       0.99      1.00      0.99      3463
           1       1.00      0.97      0.98      1318

    accuracy                           0.99      4781
   macro avg       0.99      0.99      0.99      4781
weighted avg       0.99      0.99      0.99      4781
```



```
Confusion matrix
 Accuracy : 0.9916335494666387
 Precision : 0.9961180124223602
 Recall : 0.9734446130500759
```

Approaches



XSS Attack Detection approaches
Client-side Detection approach
Server-side Detection approach
Client-Server side Detection approach
Static analysis
Dynamic analysis
Hybrid analysis
Static analysis
Dynamic analysis
Hybrid analysis
Anomaly based analysis

Conclusion

From the experimental results, we can conclude that, DL approaches have advantages over traditional approaches of detecting XSS attacks. ML approach combined with traditional approaches can detect XSS attacks with a higher accuracy. This project presents a comparison of a machine learning method RFC, SVM, Decision Tree, and LR in detecting XSS attacks. What distinguishes this research from previous research is that in this we use machine learning and deep learning method specifically on the script feature. This research can improve the ability and effective detection of XSS attacks. From the simulation results, the RFC method that has a 97% recall value, 99% precision, and 99% accuracy.

References

[1] Amodei, Dario, Ananthanarayanan, Sundaram, Anubhai, Rishita, Bai, Jingliang, Battenberg, Eric, Case, Carl, Casper, Jared, Catanzaro, Bryan, Cheng, Qiang, Chen, Guoliang, et al. Deep speech

2: End-to-end speech recognition in english and mandarin. In ICML, 2016.

[2] Amodei, Dario, Olah, Chris, Steinhardt, Jacob, Christiano, Paul, Schulman, John, and Mane,´ Dan. Concrete problems in ai safety. arXiv preprint arXiv:1606.06565, 2016.

[3] Carlini, Nicholas and Wagner, David. Adversarial examples are not easily detected: Bypassing ten detection methods. In ACM workshop on AISec, 2017.

[4] Chrabaszcz, Patryk, Loshchilov, Ilya, and Hutter, Frank. A downsampled variant of imagenet as an alternative to the cifar datasets. arXiv preprint arXiv:1707.08819, 2017.

[5] Deng, Jia, Dong, Wei, Socher, Richard, Li, Li-Jia, Li, Kai, and Fei-Fei, Li. Imagenet: A large-scale hierarchical image database. In CVPR, 2009.

[6] Evtimov, Ivan, Eykholt, Kevin, Fernandes, Earlence, Kohno, Tadayoshi, Li, Bo, Prakash, Atul, Rahmati, Amir, and Song, Dawn. Robust physical-world attacks on machine learning models. In CVPR, 2018.

[7] Feinman, Reuben, Curtin, Ryan R, Shintre, Saurabh, and Gardner, Andrew B. Detecting adversarial samples from artifacts. arXiv preprint arXiv:1703.00410,2017.

[8] Gal, Yarin, Islam, Riashat, and Ghahramani, Zoubin. Deep bayesian active learning with image data. In ICML, 2017.

[9] Girshick, Ross. Fast r-cnn. In ICCV, 2015.

10] Goodfellow, Ian J, Shlens, Jonathon, and Szegedy, Christian. Explaining and harnessing adversarial examples. In ICLR, 2015.

[11] Guo, Chuan, Rana, Mayank, Cisse, Moustapha, and van der Maaten, Laurens. Countering ´ adversarial images using input transformations. arXiv preprint arXiv:1711.00117, 2017.

[12] He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, and Sun, Jian. Deep residual learning for image recognition. In CVPR, 2016.

[13] Hendrycks, Dan and Gimpel, Kevin. A baseline for detecting misclassified and out-ofdistribution examples in neural networks. In ICLR, 2017.

[14] Huang, Gao and Liu, Zhuang. Densely connected convolutional networks. In CVPR, 2017.

[15] Krizhevsky, Alex and Hinton, Geoffrey. Learning multiple layers of features from tiny images. 2009.

[16] Kurakin, Alexey, Goodfellow, Ian, and Bengio, Samy. Adversarial examples in the physical world. arXiv preprint arXiv:1607.02533, 2016.

[17] Lasserre, Julia A, Bishop, Christopher M, and Minka, Thomas P. Principled hybrids of generative and discriminative models. In CVPR, 2006.

[18] Lee, Kibok, Lee, Kimin, Min, Kyle, Zhang, Yuting, Shin, Jinwoo, and Lee, Honglak. Hierarchical novelty detection for visual object recognition. In CVPR, 2018.

[19] Lee, Kimin, Hwang, Changho, Park, KyoungSoo, and Shin, Jinwoo. Confident multiple choice learning. In ICML, 2017.

[20] Lee, Kimin, Lee, Honglak, Lee, Kibok, and Shin, Jinwoo. Training confidence-calibrated classifiers for detecting out-of-distribution samples. In ICLR, 2018.

[21] Liang, Shiyu, Li, Yixuan, and Srikant,
R. Principled detection of out-of-distribution examples in neural networks. In ICLR, 2018.

[22] Bronjon Gogoi, Tasiruddin Ahmed, and Hemanta Kumar Saikia, "Detection of XSS Attacks in Web Applications: A Machine Learning Approach", International Journal of Innovative Researchin Computer Science & Technology (IJIRCST) ,ISSN: 2347-5552, Volume-9, Issue-1, January 2021 https://doi.org/10.21276/ijircst.2021.9.1.1 ,Article ID D10962, Pages 1-10 .

[23] YunZhou and PeichaoWang ,"Anensemble learning approach for XSSattack detection with domain knowledgeand threat intelligence", 25 June 2018,Revised26 October 2018, Accepted31December 2018, Available online 11January 2019, Version of Record 17 January2019, https://doi.org/10.1016/j.cose.2018.12.016

[24] Gulit Habibi and Nico Surantha
,"XSS Attack Detection With Machine Learning and n-Gram Methods", 2020 International Conference on Information Management and Technology(ICIMTech), DOI.

GUIDE : Dr. Siva Subramanyan