Machine Learning with Logistic Regression for Web Application Firewall

Dr. T. Arumuga Maria Devi,
Associate Professor
Center for Information Technology & Engineering,
Manonmaniam Sundaranar University,
Abishekapatti,Tirunelveli-627012.

B. Akshay Kumar,
M.Sc.Cyber security,
PG Scholar
Center for Information Technology & Engineering,
Manonmaniam Sundaranar University,
Abishekapatti,Tirunelveli-627012.

***Abstract*— As data packets move to and from a website or web application, a web application firewall (WAF) keeps track of, filters, and stops them. Companies commonly use web application firewalls as a security measure to guard web systems from known and unknowable dangers and vulnerabilities as well as zero-day exploits, malwares, impersonation, and other attacks. This paper proposed a machine learning-based approach for a web application firewall. Utilizing various payloads, our suggested model produced classification accuracy of 99% using the method of logistic regression.**

Keywords—Web Application Firewall, machine learning, signature-based WAF , vulnerabilities, Zero-day attacks.

I. INTRODUCTION

When an organisation uses technology for its various types of work (applications, operating systems, databases, network services, etc.), cyber attackers targeting web servers and applications were and are still one of the significant factors that are taken into consideration. These attacks continue to pose a high risk despite the wide variety of countermeasures available. This reduced the effects of these attacks but was powerless to have any real influence.

Dedicated software or products that support these defensive procedures and function in an integrative way with all these defensive procedures are urgently required because attacks are constantly growing despite the defensive measures that web application developers have put in place [1]. It will increase the security level of web applications. To help developers and white hat hackers in enhancing the security, security projects and standards such as OWASP were developed[2]. Web application firewalls deal in interactions with web requests in the application layer[3] as opposed to traditional firewalls, which engage in interactions with packets in the network and transport layers.[4]

We actually think that a computer cannot learn and make decisions like humans, but rather that it has evolved into a competitor to human abilities as artificial intelligence (AI) has become a scientific revolution[5] in recent decades and has obtained unparalleled superiority in mastering the work that humans do. Many human jobs are anticipated to be eliminated by artificial intelligence in the ensuing decades. Researchers and information security experts have explicitly moved to utilise artificial intelligence's ability to detect and combat assaults.

In this paper, one of the classification algorithms in machine learning is Logistic regression which is used to classify the good and bad queries. The queries were extracted and those queries were classified as normal and malicious using Logistic Regression Classifier.

II. TECHNICAL BACKGROUND

Web application vulnerabilities remain the same in essence, but the methods for exploiting them have changed. The following are the most popular web application vulnerabilities:

1. **Injections**: The most well-known injection attack is SQL injection, which enables the attacker to interact with the database by reading, writing, and altering records. It manipulating the input to cause a web application to perform commands in the operating system and queries in databases.[6]
2. **Broken authentication**: gaining access to and controlling accounts by taking advantage of logical and structural flaws in the authentication process.
3. Exposing sensitive data involves tricking a web application into throwing exceptions and revealing sensitive information, including database login passwords.
4. **XML external entity (XXE)**: modifying inputs by using XML parsing tools to execute arbitrary commands.
5. **Broken access control**: Accessing unrestricted resources in a web application without authorization due to weak access control rules, such as accessing the administrator panel when it isn't blocked, is an example of broken access control.
6. **Security misconfigurations**: Using brute force to locate and exploit security issues including unpatched bugs, default setups, unused pages, unsecured files and directories, and superfluous services are examples of security misconfigurations.
7. **Cross-site scripting**: Cross-site scripting (XSS) is the act of inserting JavaScript code into a web application to change how it appears and force the user to run it in their browser.

There are various varieties of XSS, including DOM XSS and Reflected XSS.

8. **Insecure deserialization**: Deserializing inputs from a web application, altering them, and then serialising them once more is known as insecure deserialization. This compromises the web application.
9. Using components in web applications that have known vulnerabilities: stop updating the utilised component allows attackers to take use of its known weaknesses; this kind of vulnerability is common, especially in CMS web applications.
10. Inadequate logging and monitoring refers to a lack of processes and procedures for logging and monitoring that enable attackers to locate and exploit without being noticed.

Machine Learning:A mathematical representation of the output of the training process is known as a machine learning model. The study of various algorithms that may develop a model automatically through practise and historical data is known as machine learning. A machine learning model is comparable to software created for computers that can identify patterns or behaviours based on past experience or data. A machine learning (ML) model that captures the patterns found in the training data is produced by the learning algorithm after it analyses the training data for patterns.

There are three learning models for algorithms that are based on various business objectives and data sources. Each algorithm for machine learning settles into one of three models:

1. Supervised Learning
2. Unsupervised Learning
3. Reinforcement Learning

Supervised Learning: The easiest machine learning model to understand is supervised learning, in which input data is referred to as training data and has a known label or outcome as an output. As a result, it operates on the idea of input-output pairs. In order to conduct prediction, it is necessary to develop a function that can be learned using a training set of data before being applied to unknowable data. Task-based supervised learning is evaluated using labelled data sets.

On straightforward real-world issues, we can put a supervised learning model into practise. For example, We could create a supervised learning model to predict a person's height based on their age, if we had a dataset that included both their age and height.

Models for supervised learning are further divided into two groups:

1. Classification
2. Regression

Regression: A continuous variable is the output in regression problems. Few examples of frequently used regression models are Linear Regression, Logistic Regression, Decision Tree, Random Forest, etc.,

Logistic regression: One of the most often used Machine Learning algorithms, within the category of Supervised Learning, is logistic regression. Using a predetermined set of individual variables, it is used to predict the categorical dependent variable.

Classifiers: Classifiers are supervised machine learning algorithms that are skilled at labelling data, or categorizing input data into several groups.

When supplied to the classifier, the input data from the data sample is typically not in its original format. Instead, a vector is created from the data sample by extracting several significant features. The word "vector," which is frequently used in machine learning language, is equivalent to the vector of extracted characteristics.

An example classifier classifies the different shapes using the color feature of different shapes in Figure 1.

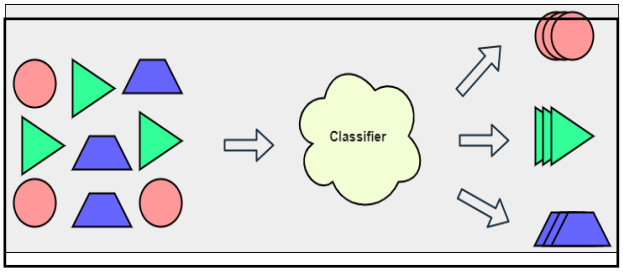

Figure 1: Classify the different shapes using classifier

Finding a useful set of characteristics to extract from the data samples that may help distinguish between the various classes is one of the main challenges when developing a classifier.

Logistic regression classifier: This classifier employs regression to fit class boundaries, as the name implies. The sigmoid function uses the regression line as input, which can be any arbitrary function of any order. From the regression line, the sigmoid function draws a line dividing two classes. Typically, the point where the sigmoid function equals 0.5 is used to define the border between classes zero and one. Equation displays the Equation 1 used to establish a boundary at 0.5 and Figure 2 and 3 displays an illustration of utilizing this equation to separate a one-dimensional data collection.

$$y(x,f) = \begin{cases} 1 \ if \ \frac{1}{1+e^{-f(x)}} > 0.5 \\ else \ 0 \end{cases}$$

$$where \ f = \ regression \ function$$

Equation 1

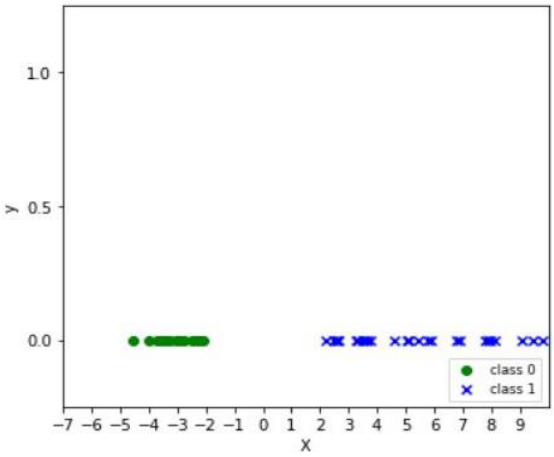

Figure 2

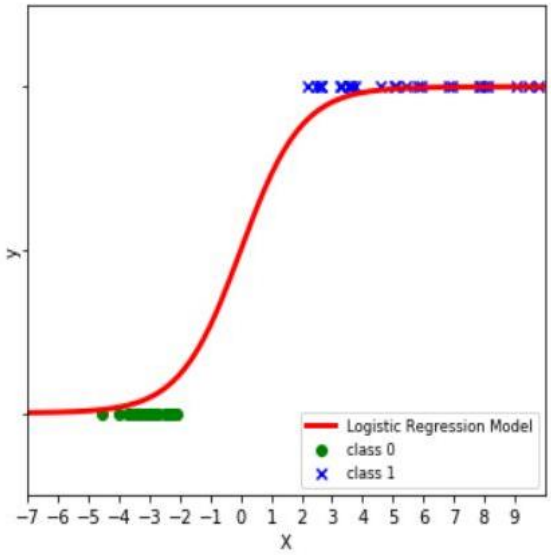

Figure 3

Fig.2 & 3: A one-dimensional data collection using a logistic classifier as an example

Bag-of-words: Raw text data is not appropriate for putting directly into a model since machine learning classifiers usually require numbers as input. Bag-of-words is a common method for handling text data entry problems that turns the text string into a vector of word counts. As a result of the bag-of-words transformation, each distinct word in the full data set is represented as a separate feature. A data point's feature vector is simply 0 for all features other than those that represent the words in the string. The values of those features are proportional to how many times each word appears in the string.

Principle component analysis: PCA, often known as principle component analysis, is a type of unsupervised learning technique. It is a method for reducing the dimensionality of a data set by projecting it from the original feature space into a reduced feature space. The original features are linearly combined to create the new features.

The transformation process aims to keep as much diversity as possible. When a three-dimensional hand is projected onto a two-dimensional plane, as in Figure 4, the variation is greatest when the palm's surface is facing the plane. Compressing feature space, feature selection, and visualization are all possible with PCA. PCA is used in this report to visualize highly dimensional data.

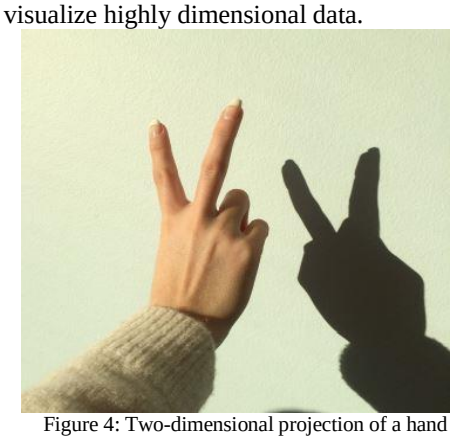

Figure 4: Two-dimensional projection of a hand

Training Data, Validation and Test Data: The data set is typically divided into three parts: a training set, a validation set, and a test set for developing and testing a machine learning model. To reduce the chance of curve-fitting, the data set was divided. This phenomenon happens when the model is tested using the same data it was trained on. As a result, the model will perform well at predicting previously known data points, but it will typically perform poorly when predicting previously unknown data points.

Cross-validation is a prominent method frequently employed while optimizing a model. By adjusting the validation data piece to an interval inside the previous training data and then setting the validation data back to training data, the technique's basic idea is to reuse the training and validation data. For improved accuracy of the validation performance, repeat this process. In other words, it eliminates any potential bias the validation data may have had by creating a new validation data for numerous iterations.

Performance Measures − Bias and Variance: The effectiveness of a program's learning process can be determined using a variety of indicators. Numerous performance metrics for supervised learning issues count the number of predictions made incorrectly.

A model's prediction error can be attributed mostly to bias and variation. Assume you have a large number of training sets that are each distinct but equally representative of the population. Regardless of the training set used, a model with a strong bias would yield similar errors for given input because it favors its own assumptions about the genuine relationship over the relationship shown by the training data. On the other hand, a model with high variance will provide various mistakes for an input depending on the training dataset that it was trained with. High variance models have the potential to be so flexible that they can model the noise in the training set, in contrast to high bias models, which are rigid. In other words, a high variance model overfits the training data, and a high bias model underfits the training data.

Performance metrics that reflect the costs of making errors in the real world should be used to assess machine

learning systems. Although it seems insignificant, the example that follows shows how to utilize a performance measure that is appropriate for the activity in general but unsuitable for its particular application.

Performance metrics: A performance metric must be compared when evaluating models based on test data or outcomes of validation data. The most straightforward is to order the findings according to accuracy, such as the percentage of correctly classified data. However, this statistic might be misleading if the distribution of classes among the data points is highly skewed. The confusion matrix is used to derive additional standard performance indicators.

In addition to accuracy, the performance metrics considered in this study include : f1-score and AUC

When the distribution of labels within the data set is skewed, the F1-score from Equation 2 is a reasonable all-around indicator to employ instead of accuracy.

$$F1\text{-}score = \frac{2 * precision * recall}{precision + recall}$$

$$where\ recall = sensitivity$$

$$precision = \frac{TP}{(TP + FP)}$$

Equation 2

An indicator derived from the Receiver operating characteristic is called "Area under the Curve," or AUC (ROC). The true positive rate (sensitivity) and the false positive rate (1 specificity) are plotted on the y-axis and x-axis, respectively, of a model's ROC curve. The percent of the graph that is under the ROC curve is all that the AUC metric measures. (0 < x-axis < 1 and 0 < y-axis < 1)

III. METHOD

A. *Preparatory work*

Studies and research into safeguarding web applications from malicious requests used one of two methodologies to identify an attack: classify requests as anomalous or typical in general, regardless of the type of attack, or identify and detect a specific attack (such as detection of the SQL injection attack only or cross-site scripting attack detection).

In order to translate the experience to computers, it also used two methods: signature-based detection, which makes use of databases that contain patterns of attacks, and behavioural-based detection, which is designed and implemented using artificial intelligence techniques like classification algorithms or a custom algorithm.

The majority of studies used outdated datasets like ECML-PKDD 2007 and CSIC 2010. Modern datasets were not used to evaluate the proposed models, and some researchers' datasets are not readily accessible online.

Then used few scripts to label the dataset which containing lot of http logs. After the data was properly prepared, I wanted to gather several more malicious queries. As a result, I continued my search for payloads and discovered some well-known GitHub repositories that had XSS, SQL, and other attack payloads, which I then utilised in my dataset of malicious queries.

We now had two files, one with good web requests and the other with malicious web queries. The only data we require to train our classifier is that.

B. *Programming language*

Python was supposed to be the in-demand programming language. The language has become the most widely used programming language for machine learning because of the libraries that are described below and is particularly suitable for data analysis.

- **Scikit-learn**: Scikit-learn Python's version of the Swiss Army Knife of machine learning. All of the machine learning algorithms, validation tools, and model selection tools we require for the project are all included in this library.
  Website: https://scikit-learn.org/stable/index.html
- **Matplotlib**: Matplotlib is a large library of plots that makes it simple to visualise any type of data analysis.
  Website: https://matplotlib.org/

Since this is a security project, our focus will be on applying machine learning methods rather than actually implementing them. This was possible because of scikit-learn.

C. *Documentation*

All experiments were programmed and evaluated in Jupyter Notebooks, an interactive Python environment for data science (website: http://jupyter.org). Using its integrated support for Matplotlib, markup language, plots, and tables, the code flow may be presented in a way that is much more visually appealing and easy to understand. To follow our development process, the reader can also step-by-step run the code components.

The notebooks will contain each part's results, code documentation, and a description of the implementation's model be developed.

IV. IMPLEMENTATION

The following section will describe how the theory and methods from the technical background were used, from locating and extracting high-quality data to categorising it with Python.

A. *Data Collection*

After extensively searching the internet for good data, collected legal input, XSS-attacks, SQL and command injections. The majority of the data collected from various github repositories. The data acquired, however, wasn't always in the format required for this project, i.e. just the input string. For instance, some of the data were contained in HTTP,GET and POST queries. The data was converted into a format appropriate for this purpose using Python scripts.

B. *Feature engineering*

Bag-of-words technique is used for converting the input data to numeric features.

Bag of words feature space: To convert our payloads into better feature spaces, we tested a number of well-

known bag-of-words techniques, including a count vectorizer and a TF-IDF vectorizer. The first employs the methodology described in Technical Background , whereas the later makes use of a weighted variant. Less common words are automatically given more weight when using the weights.

We had to perform additional pre-processing before transforming the payloads using the bag-of-words vectorizers because our payload data samples aren't organised like typical text documents. Our payload data samples were converted into "words" of size N using the N-grams method. We specifically employed 1-gram (unigram), 2-gram, and 3-gram; a sample of each application is presented in Table 1.

N-gram	Payload input	Payload Output
1-gram	"<script>"	['<','s','c','r','i','p','t','>']
2-gram	"<script>"	['<s','sc','cr','ri','ip','pt','t>']
3-gram	"<script>"	['<sc','scr','cri','rip','ipt','pt>']

Table 1: Example of '<script>' string transformed in 1-gram, 2-gram and 3-gram

For each vectorizer, these N-grams were merged, resulting in six distinct feature spaces:

- 1-gram count vectorizer (175 features)
- 2-gram count vectorizer (4357 features)
- 3-gram count vectorizer (55424 features)
- 1-gram TF-IDF vectorizer (175 features)
- 2-gram TF-IDF vectorizer (4357 features)
- 3-gram TF-IDF vectorizer (55424 features)

We were unable to go higher than 3-grams because of our limited computational capacity, as shown by the 3-gram vectorizers' output feature spaces of 55424 features.

The bag-of-words feature spaces were defined, and then these spaces were projected into two dimensions using PCA. In order to determine whether the data behaved randomly or if there was structure between the non-malicious and malicious payloads in the feature spaces, this was simply done for visualisation purposes. We can observe that the bad and good points are definitely emerging from various points of view in Figure 5. Every feature space moved on to the next stage, training and model selection, since it appeared promising when visualised using PCA.

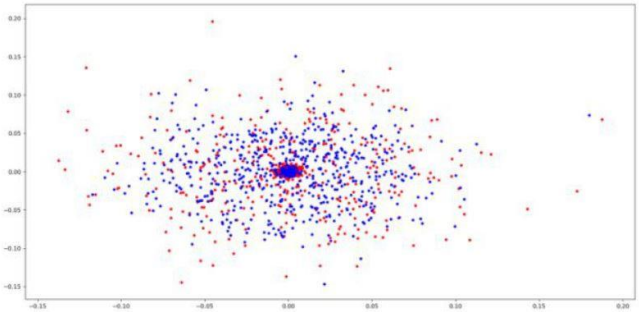

Figure 5: Projection of the dataset onto two dimensions to visualise the three-grams TF-IDF feature space with PCA

Let's utilise our classifier after using Tfidvectorizer to turn the data into tfidf values. We are using tfidf values because we want to give our ngrams weights. For example, the ngram '<img' should have a high weight because a query that contains it is most likely harmful. Then apply logistic regression.

V. RESULTS AND DISCUSSION

The project's objective was to assess the application-level potential of machine learning for firewalls. A set of classifiers that can recognise dangerous payloads will be offered as the answer reflecting this question.

As input to the classifiers, The feature extraction of distinct characteristics and the bag-of-words technique were integrated into a unique feature space.

The results were acquired with the accuracy of 99.9%, Precision of 98.2% and the F1-score of 99% as shown in Figure 6

```
Bad samples: 44533
Good samples: 1265974
Baseline Constant negative: 0.966018
------------
Accuracy: 0.999325
Precision: 0.982081
Recall: 0.998065
F1-Score: 0.990008
AUC: 0.999972
```

Figure 6: Results

We looked into the possibility of supervised learning in this research to identify dangers like SQL injections and cross-site scripting. We may train the model on a larger dataset containing all varieties of harmful queries to increase the range of malicious queries that it can identify.

VI. CONCLUSION

The primary goal of the study was to examine the feasibility of employing machine learning in conjunction with a web application firewall to detect harmful attacks. Given the short timeline of our experiment and the small amount of internet data sets available, we think machine learning and even cyber security in general has enormous potential in this field. The logistic Regression classifier (where a 3- grams vectorizer) had 99.93% accuracy and a precision of 98.83%.

REFERENCES

[1] M. Chora´s and R. Kozik, "Machine learning techniques applied to detect cyber attacks on web applications," Logic Journal of IGPL, vol. 23, no. 1, pp. 45–56, 2015.

[2] D. Wichers and J. Williams, "Owasp Top Ten," The open web application security project, vol. 3, 2017.

[3] A. H. Yaacob, M. Nazrul, N. Ahmad, and M. Roslee, "Moving towards positive security model for web application firewall," International Journal of Computer and Information Engineering, vol. 6, no. 12, pp. 1763–1768, 2012.

[4] Z. J. Huang, O. Ai, and X. U. Hong-xian, "Network Security and Firewall Technology," Journal of Naval University of Engineering, vol. 1, 2002.Z. J. Huang, O. Ai, and X. U. Hong-xian, "Network Security

[5] W. Wang and K. Siau, "Artificial intelligence, machine learning, automation, robotics, future of work and future of humanity," Journal of Database Management, vol. 30, pp. 61–79, 2019.

[6] D. Mitropoulos, V. Karakoidas, P. Louridas, and D. Spinellis, "Countering Code Injection Attacks: A Unified Approach," Information Management & Computer Security, vol. 19, no. 3, 2011.

AUTHOR'S PROFILE

Dr. T. Arumuga Maria Devi, Assosiate Professor Received B.E. degree in Electronics & Communication Engineering from Manonmaniam Sundaranar University, Tirunelveli, Tamil Nadu, India, in 2003, M.Tech degree in Computer & Information Technology from Manonmaniam Sundaranar University, Tirunelveli, Tamil Nadu, India, in 2005, also received Ph.D degree in Information Technology—Computer Science and Engineering, from Manonmaniam Sundaranar University, Tirunelveli, Tamil Nadu, India, in 2012 and also the Associate Professor of Center for Information Technology and Engineering of Manonmaniam Sundaranar University since November 2005 onwards. Her research includes Signal Processing, Remote Communication, Multime- dia and Mobile Computing.

Akshay Kumar.B, Msc. Cyber Security, Centre for Information Technology & Engineering, Manonmaniam Sundaranar University, Abishekapatti, Tirunelveli - 627012, Tamilnadu, India. He received Bachelor of Physics in Sri Sankara Arts and Science College,Kanchipuram. His research includes Machine learning, Blockkchain and Ethical Hacking .