Detecting Web-based Attacks Using Machine Learning Techniques

Contributors: Chenfeng Nie, Shuaichen Wu, Xiaoyi Rao
Students: Johns Hopkins Information Security Institute

1. Abstract:

With the rise of the Internet, web application have taken a massive leap forward. At the same time, web-based attacks such as SQL injection, Cross-site Scripting Attack, Remote Control Execution and Path Traversal Attack pose a serious security threat to web applications. Adversaries can launch web application attacks to steal sensitive data, deface database management system and manipulate entire information system from their targets. Although there are a couple of successful web-based attack detection models such as signature-based IDS in the past, these models turn out become more difficult to adapt modern web app attacks because attacks become more complicated and sneaky. To address this problem, the goal of our project is to understand the advanced concepts about types of web-based attack, applying various machine learning techniques to detect web-based attacks and seeking an approach to build modern machine learning driven web application firewall.

Keywords: **Web application firewall, Detection, Web application, Machine learning, SQL injection, XSS attack, Path traversal intrusion, Evaluation**

2. Introduction

This paper begins with an overview of various types of web-based attacks, then narrows our focus to provide a front-to-end procedure to prove whether there is an approach to implement machine learning techniques to detect web-based attacks, aka web application firewall(WAF) in order to improve detection efficiency and accuracy. Later on, we are going to present the evaluation regarding our approach and address the limitation of our work. Then we are also going to note down an ideal model which could be used as a reference for future expectations.

As we increasingly replying on internet and web applications such as Facebook, Amazon and Google, attacks upon web application have continued to evolve and increase steadily overtime. As SANS institute pointed out[1], around 70% of all attacks occur at the application layer, in other words, large portion of attacks is web-based since web-based attacks mainly focus on an application itself and critical functions on layer 7 (a.k.a the application layer) [2]. For instance, SQL Injection and Cross-site Attack(a.k.a XSS attack) as two main type web based attacks of The Ten Most Critical Web Application Security Risks according to OWASP[3], have been used frequently to achieve their conspiracies.

On the other hand, the rapid development of internet had a huge impact on traditional mechanism such as traditional signature-based intrusion detection system to detect sophisticated, sneaky and large-volume attacks in the modern era. Therefore, it is urgent to find a feasible approach to face and solve these challenges. Machine learning techniques, widely used in researchers, have been gradually proven to be able to assist with this process by the use of algorithms to detect patterns and the creation of

analytical models. In our case, we attempted to implement and evaluate our own web application firewall by various machine learning techniques.

3. Types of Web-based Attacks

Refer to Justin Crist's definition of Web-based attacks [1], web-based attacks can negatively affect the confidentiality, integrity, and availability (C-I-A) of data, specifically, the purpose of a web based attacks usually target to commercial network or host for stealing, falsifying and even manipulating crucial business data and information systems. First we need to know how web application attack works. By definition [1], all web application attacks attempt to comprise at least one normal request or a modified request aimed at taking advantage of poor parameter checking or instruction spoofing. Like Figure 1 shown below, adversary could tamper the HTTP request or query(a.k.a the URL) and attach payload scripts to invade victim's system and database.

Figure 1. Legit HTTP Request and Malicious HTTP Request[7]

In our project, we collected four main types of web application attacks based on top 10 web attacks of OWASP, namely SQL injection(a.k.a SQLi), XSS attack, Remote Control Execution and Path Traversal Intrusion. We are going to describe each of them briefly below.

I. SQL Injection

SQL injection is a kind of attack where the adversary send malicious Structured Query Language code to web application owner's server through address bar, user input bar, web form element and etc to read sensitive data from database, modify database data and even execute administration operations on the database **[9]**. In our project, we mainly focus on the attack via address bar. In fact, there are various types of SQL injection, however, we treated them as one type. Malicious SQL injection scripts are collected as malicious dataset for researching purpose [8][13].

II. Cross-site Scripting (a.k.a XSS attack)

Cross-site scripting attack is another type of injection, it exploits malicious scripts, for example, Javascript code to redirect users to malicious website, compromise credentials and even terminate websites. Different from SQL injection, Cross-site scripting attack focus on stealing data from the web application's front end rather back end. Malicious XSS scripts are collected as malicious dataset for researching purpose [8][13].

III. Remote Control Execution

RCE is a kind of arbitrary code execution via network. Adversary could inject its

malicious program or script as a payload into victim's web server in order to grant root privilege and execute malicious script in administration terminal such as CMD and root shell remotely. Malicious RCE scripts are collected as malicious dataset for researching purpose[8][13].

IV. Path Traversal Intrusion

Path traversal attack is also called directory traversal attacks[**12**], it is an HTTP attack targeted file system in victim's web server which allows adversary to access restricted and classified directories and execute commands outside of web application server's root directory. Malicious Path Traversal scripts are collected as malicious dataset for researching purpose[8][13].

4. Build the Wall via Machine Learning Techniques

1. Prepare Dataset

As mentioned above, we collected four main types of web-based-attack scripts as text format and wrote a script to concat all of them into one large malicious dataset(6961 malicious queries in total) [8]. Regarding the legit dataset in normal usage (contrast to malicious dataset), we downloaded a dataset included raw HTTP log file in normal usage from a third-party website called SecRepo[13]. Since the dataset only provided raw HTTP header information, we wrote a script to extract and prune all raw data to same format as malicious dataset. For example, after extracted and pruned, raw data: 68.180.228.229 - - [01/Jan/2017:02:17:59 -0800] "GET /robots.txt HTTP/1.1" 200 233 "-" "Mozilla/5.0 (compatible; Yahoo! Slurp; http://help.yahoo.com/help/us/ysearch/slurp)" became "/help/us/ysearch/slurp" as a sample data row in our legit dataset(24781 legit queries in total).

2. Preprocessing Dataset

In this project, we built three detection models, respectively KNN, SVM and Logistic Regression. To fit dataset into three models, preprocessing of dataset is needed.

There are three steps, normalization, transformation and features extraction applied in preprocessing dataset. In normalization step, all queries in dataset should be transferred to unicode characters to keep every character share the same encoding method. In transformation step, function in pandas library is applied to deal with our dataset. In feature selection step, firstly add the target label, "1" for malicious query, "0" for normal query. Second, add features for each data. "length" is the length of each query. "entropy " describes the purity of each query. To ensure two features are suitable to be predictors, historical diagrams are plot.

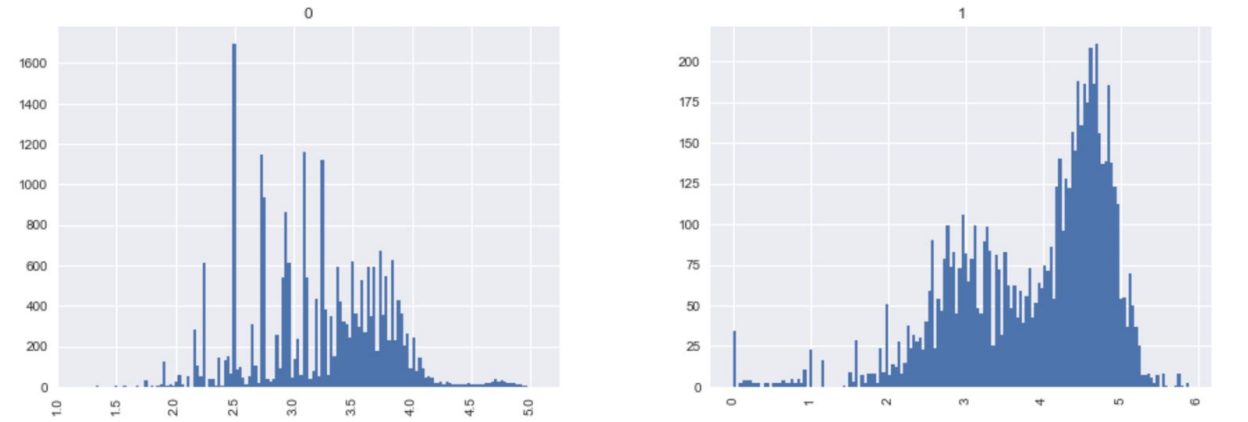

Figure 2: Historical diagram by entropy

The diagram above shows malicious queries have higher entropy than normal queries. And most of malicious' entropy value is between 4 to 5.

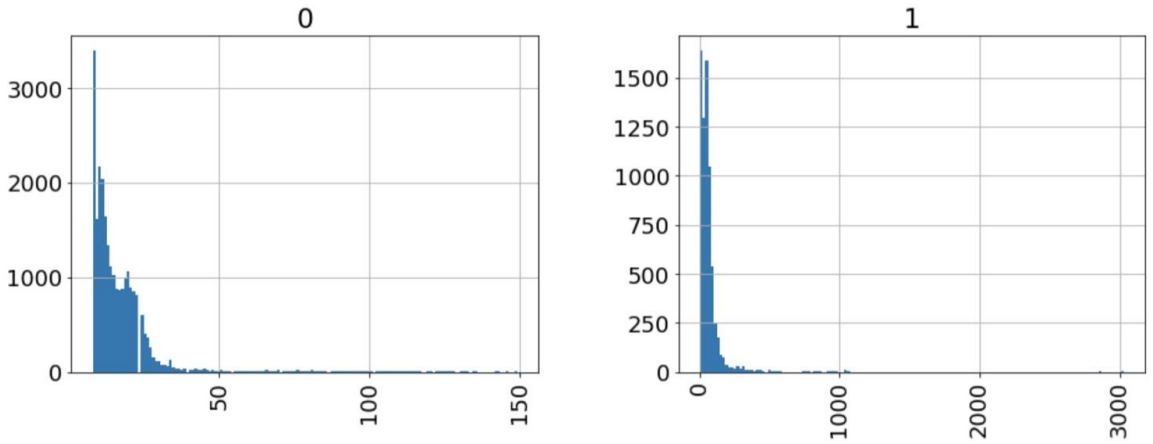

Figure 3: Historical diagram by length

Diagram above can clearly show that most malicious query is much longer than normal queries. Almost all normal queries' length is less than 20, malicious query could even has 500 characters. Length and entropy are suitable to be predicate features for malicious and legit queries.

After three steps, the preprocessed dataset is shown below.

	label	queries	unicode	length	entropy
6956	1	<XML SRC="http://ha.ckers.org/xsstest.xml" ID=...	b'<XML SRC="http://ha.ckers.org/xsstest.xml" I...	106	4.955310
6957	1	'';!--"<XSS>=&{()}	b'\'\';!--"<XSS>=&{()}'	18	3.836592
6958	1	Execute(MsgBox(chr(88)&chr(83)&chr(83)))<	b'Execute(MsgBox(chr(88)&chr(83)&chr(83)))<'	41	3.973916
6959	1	document.__parent__._=alert	b'document.__parent__._=alert'	27	3.676391
6960	1	top.__proto__._= alert	b'top.__proto__._= alert'	22	3.226731

Figure 4: Preprocessed dataset using pandas

Different from other two models, in Logistic regression model,text mining is used to extract features. Tf-idf using Ngram adds different tokens for each query. n-gram is n successive words extracted from a text corpus. In our project, N is set in range from 1 to 5. For example, query "<script src="http:>" could be split into { "script" : 1, "script src" : 2, "script src=" :3, "script src="" :4, script src="http : 5 }. Tf-idf describes how important a token is to every query in whole collection and could add weighting factor to each token.

The preprocessed dataset is shown below:

	\t	\t6	\t6&	\t6.	\ta	\tas	\tp	\tp&	\tp:	\n ...	s s	\
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.025888 ...	0.0	
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.016378 ...	0.0	
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.021441 ...	0.0	
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.010629 ...	0.0	
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.022389 ...	0.0	
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.014977 ...	0.0	
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.009504 ...	0.0	
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.018135 ...	0.0	
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.012731 ...	0.0	
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.014840 ...	0.0	
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.025598 ...	0.0	
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.020387 ...	0.0	
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.007956 ...	0.0	
13	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.010923 ...	0.0	
14	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.018779 ...	0.0	
15	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.015956 ...	0.0	
16	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.008064 ...	0.0	
17	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.070483 ...	0.0	

Figure 5: preprocessed dataset using TF-IDF matrix

3. Apply Machine Learning Algorithms

I. KNN

The simplest and most elementary classifier is to record all the corresponding classes of training data. When the attributes of a test object match perfectly with the attributes of a training object, they can be classified as one category. But it rarely happens. And the other situation is that when a test object matches several training objects, this object will be classified to several categories. According to these situations, K-nearest neighbor (KNN) can solve them perfectly.

KNN is to classify data by measuring the distance between different eigenvalues. The idea is very simple: if a sample is most similar to the k most similar samples in the feature space that belongs to one category, the sample will belong to this category.[10] K is usually an integer that is not more than 20. In this algorithm, the selected neighbors are all objects that have been correctly classified. In the decision making, the classification of the samples to be divided is determined only by the category of the nearest one or several samples. Therefore, the result of KNN algorithm is largely decided by the K number. And the following is the description of KNN algorithm:

a. Calculate the distance between the points of existing data sets and the new point;
b. Order the distance sorted increasingly;
c. Select the k points with the minimum distance from the new point;
d. Determine the occurrence frequency of the category that the previous k points belong to;
e. Return the category that has the highest frequency of the previous k points and this is the prediction classification of the new point.

In this project, we use KNN package integrated in *sklearn* library. The data we used here only has two attributes, “length” and “entropy”. According to the two attributes, we can divide the data into two parts, including training data and testing data. First, we need to use different K numbers here. Then fit the model with the training data and calculate the accuracy by comparing the predicting labels related to the

test data with the original label in the testing data. According to the different accuracies, the k corresponding to the best accuracy is nearly the best in the model.

II. SVM

Support Vector Machine (SVM) is a two-category classification model. The idea of classification is to give a sample set containing positive and negative examples. The purpose of SVM is to search for a hyperplane to segment samples based on positive and negative examples.

SVM has two kinds of classifiers, linear classifier and nonlinear. If a sample set can be totally divided into two parts by a linear function, the sample is linear separable. Oppositely, it is non-linear separable. As for linear classifier, it is not enough to just divide them. The kernel idea of SVM is to try best to make the separate two categories have the maximum interval, which can make the separation more trustworthy and have a good classification for unknown new samples. The kernel function is to map sample features into high-dimensional space, which can make the related features separate.

To construct a SVM with good performance, the selection of the kernel function is the key and the most important step. In SVM algorithm, there are several main kernel functions here in SVM library, such as Linear Kernel, Radial Basis Function (RBF) Kernel. When selecting kernel functions to solve practical problems, there are two usual methods: one is to use the expert knowledge of pre selected kernel function; the other is to use Cross-Validation method. The method is to try different kernel functions respectively to fit the model in the selection and the kernel function with minimum induction error is the best kernel function. Generally, Linear Kernel and RBF Kernel are two of the most widely used methods.

In this project, we used three kernel functions to analyze the dataset, including Linear Kernel, RBF Kernel, and Sigmoid Kernel. Fit the model with the training data and predict the testing data to calculate its accuracy. Then compare them to evaluate which kernel is the most suitable one.

III. Logistic regression model based on Tfidf

TF-IDF, short for Term Frequency-Inverse Document Frequency, is text mining technique and a numerical statistics. Term frequency is how often a word appears in a text or a document. Document frequency describes the times word appears in collection or corpus the document belongs to.

TF(t) = (Number of times term t appears in a document) / (Total number of terms in the document)

IDF(t) = log_e(Total number of documents / Number of documents with term t in it).

TF-IDF Value = TF * IDF

A word could be general to all document when its DF is really high. TF-IDF could be used to calculate how important a word to a document in a corpus. TF-IDF value is in direct proportional to TF(Term Frequency) and in inverse proportional to DF(Document Frequency). Which means, the importance of a word increase proportionally to the number of times the word appears in the document and decrease proportionally to the number of times the word appears in whole collection or corpus. TF-IDF is often used as a weighting factor for text mining or term searching. For example, websites could add tags for blogs based on TF-IDF.

N-gram add tokens for every query in our project. TF-IDF value is calculated for every token of every query in dataset.

Logistic regression model is statistical model that could find the relationship between target label and features of instances in a corpus or collection. Target label or dependent variable should only have two outcomes or values, which means, the corpus is categorical. Logistic regression model could be used to predict target value of new instance by calculating the possibility through regression formula. In our project, to detect if a query input is web-based attack, Logistic regression model is appropriate. A query is an instance, a query is extracted into n-grams(n is from 1 to 5) that are tokens or features belonging to the query. Token value is TF-IDF value.

5. Evaluation

a. KNN Algorithm

KNN algorithm is one of the simplest classification models. The limitation for KNN is that the large amount of the test sample will result in the large calculation and the large using memory while KNN is easy to understand and implement.

KNN algorithm is completely determined by the neighbor number K used in predicting test data, thus here we use six different K numbers, [1,5,15,50,80,100], to compare the accuracies.

```
Accuracy with k = 1 is 0.899
Accuracy with k = 5 is 0.924
Accuracy with k = 15 is 0.921
Accuracy with k = 50 is 0.910
Accuracy with k = 80 is 0.906
Accuracy with k = 100 is 0.906
```

Figure 6. KNN Accuracy

As the figure shown above, the accuracy is better than others when k =5 and when k is larger after 5, the accuracy goes down. We can conclude that the accuracy may be highest, which is 0.924, when k is around 5. If k is too small, the classification result is easy to be affected by noise points while if k is too large, the neighbor points may contain too many other categories of points.

b. SVM Algorithm

SVM is based on the statistical theory. The comparison between SVM and the BP network simulation results shows that SVM has strong approximation ability and generalization ability. However, the space consumption of SVM is mainly to store the training sample and the kernel matrix so the training time would be longer when the data amount is large.

SVM has two important parameters, “C” and “gamma”. C is the penalty coefficient, that is the tolerance of the error. The higher C, the more intolerance of error and easy fitting. Whether C is too large or too small, it will make the generalization ability worse. Gamma implicitly determines the distribution of data after mapping to a new feature space, thus the value of gamma should not be too small or too large.

In this project, the used three kernels need to have the same values of “C” and “gamma”, which is set “1” and “ 20”. At this condition, we trained the data separately in the model and use the model to predict the testing data.

```
Linear accuracy with test dataset is 0.879
RBF accuracy with test dataset is 0.923
Sigmoid accuracy with test dataset is 0.780
```

Figure 7. SVM Accuracy

The following figure shows three pictures of the testing results of the three kernels using matplotlib. In the first picture, there is a clear line to separate the background. The left of the line represents the predicted “0” (normal) label district while the right represents the predicted “1” (malicious) district. The white points means the “0” label data before testing and the red points are the “1” label data. We can see that only a few data is false predicted. In the second picture, we can see a blue line around the white points. This line, which is the separate line of the predicting districts, is not so clear but most of white points are surrounded by the blue line. At least, we can see that most of predicted malicious data is out of the blue line. Besides, the third picture is not accurate as the previous two pictures clearly in picture.

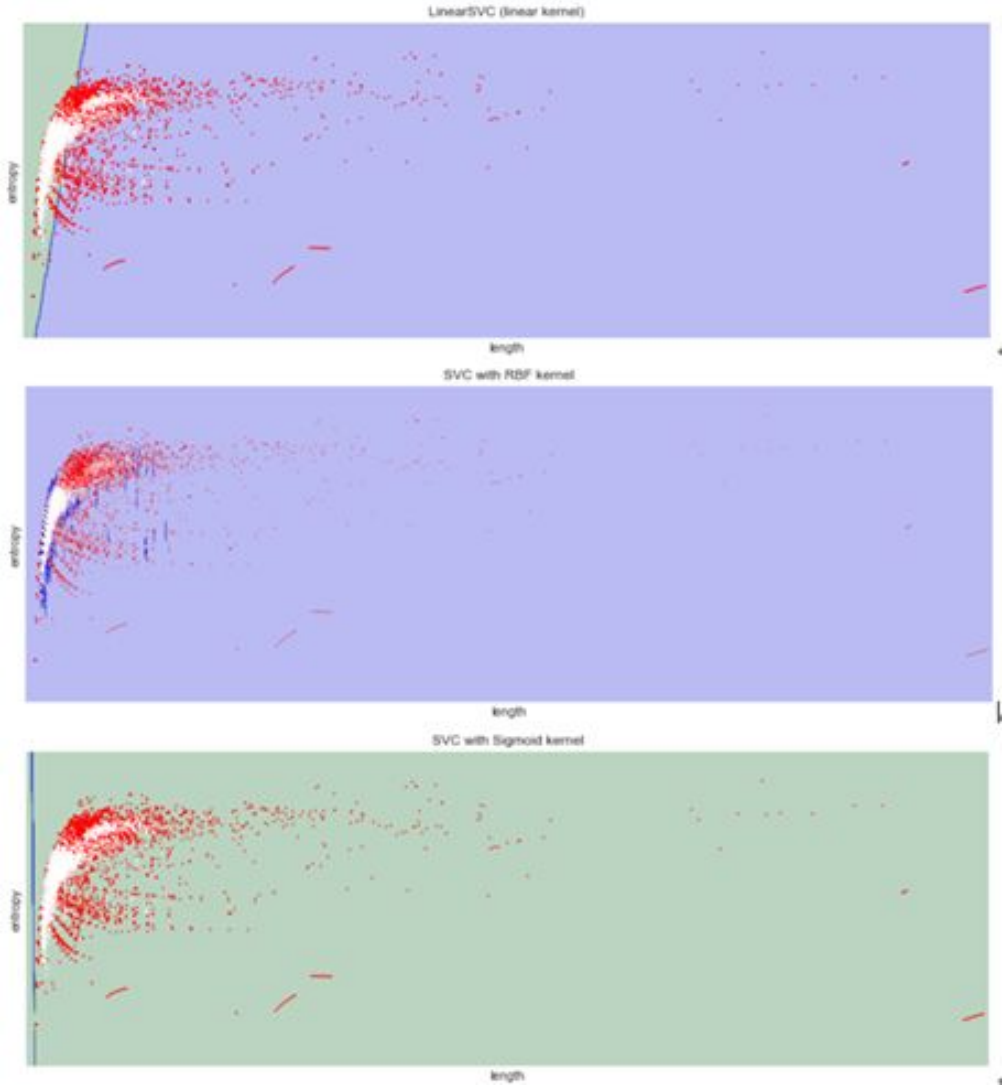

Figure 8. SVM Results in Three Different Kernels

As the figure shown above, the accuracy of RBF Kernel is the best among the three kernels and Sigmoid Kernel is the worst. Because RBF largely applies better on non-linear dataset while linear dataset only applies on the linear separate dataset. But the linear kernel runs faster than RBF. Our large dataset can help RBF get better classification result. But RBF spends more time than linear and sigmoid.

c. TF-IDF

There are some limitation for TF-IDF even it is still the most widely used algorithm in practice. TF-IDF is based on bag of word ignoring of semantic and position in text. TF-IDF over n-gram could fix the limitation in some extent. TF-IDF assumes the count of different words or word groups provide independent evidence of similarity.
Logistic regression model is to discover the relationship between target label and all variables or features. Logistic regression model offer coefficient to all features and use the sum of product of coefficients and corresponding features value to calculate the possibility of target value. Logistic regression model could make up the limitation of TF-IDF by weighting features.
In our project, score of logistic regression model is following:

```
Bad samples: 4860
Good samples: 20828
Baseline Constant negative: 0.810807
------------
Accuracy: 0.990852
Precision: 0.973931
Recall: 0.976987
F1-Score: 0.975457
AUC: 0.998887
```

Figure 8. TF-IDF Results

Accuracy indicates the ratio correctly predicted instance to the total instances. The logistic regression model got accuracy 0.993084 which is really high. Precision indicates the ratio of correctly predicted positive instance to the total predicted positive instances. Recall indicates the ratio of correctly predicted positive instance to all real positive instances.
F1 score indicates weighted average of precision and recall. The AUC value is equivalent to the probability that a randomly chosen positive example is ranked higher than a randomly chosen negative example[7] All values suggests how a model do the classification job, the model is better, when value is higher. The logistic regression model got high score which means it is a great model for web-based attack detection. The score of model is too high also suggests malicious queries are too obvious compared to normal queries.

6. Conclusion

To evaluate three models, firstly choose models with best performance over respectively three algorithm. KNN model has the best performance when k=[5,15], SVM model has the best performance when RBF kernel function is applied. To compare three models entirely, we calculate accuracy, precision,

recall and F1 values, for accuracy describes true positive and true negative, precision relates to false positive, recall relates to false negative, F1 score take both false negative and false positive into account.

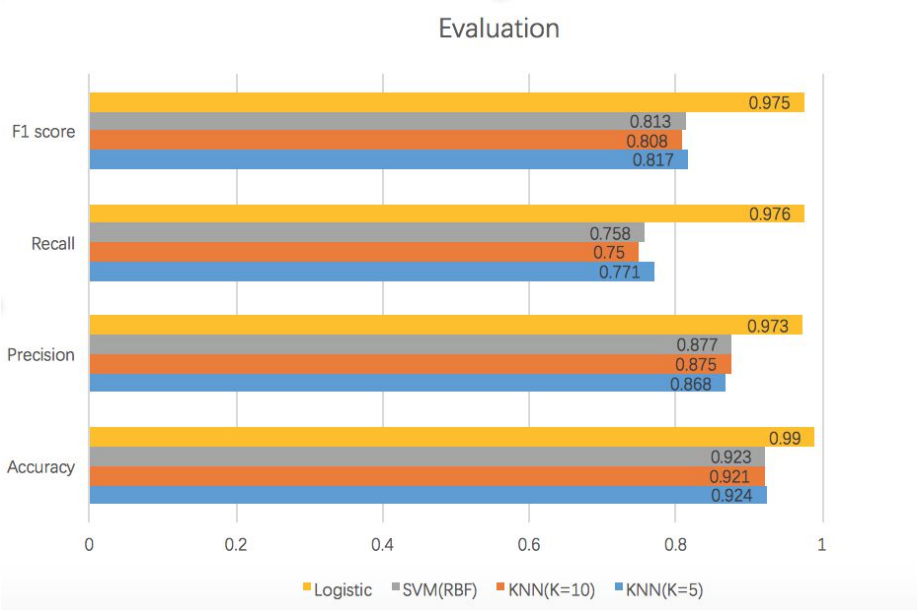

Figure 10. Overall Performance

The above diagram shows logistic regression model is best for the model has highest accuracy, precision, recall and F1 score. KNN model and SVM model almost share same performance. Even with high accuracy, the recall value is still small for both KNN and SVM model. Recall value describes the ratio correctly predicted malicious queries to actual malicious queries, which means KNN and SVM model are not sensitive to malicious queries.

The reason why logistic regression model has best performance is that TF-IDF algorithm using n-gram, kind of text mining, deals with data more in detail and offer more tokens. Although KNN and SVM has almost the same accuracy results, SVM runs slower and needs more memory than KNN. SVM is more suitable to the small dataset. To conclude, logistic regression using n grams is the best model for our dataset.

7. Future Work

By now, we have proven the machine-learning-driven web based attack detection approaches worked well based on our evaluation. For future expectations, we could build a commercial integral real-time WAF(web application firewall) architecture or system which inspired by Rathore and Paul[16]. (proposed real-time WAF architecture shown as Figure 2)

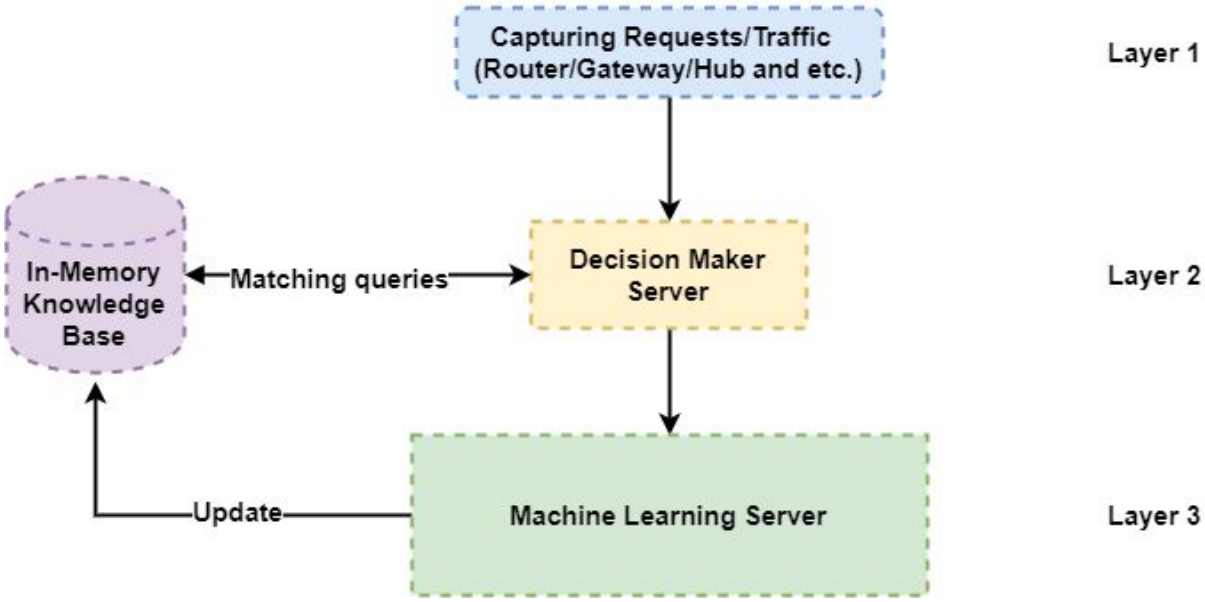

Figure 11. Proposed Architecture for Real-time Machine Learning Driven Web Application Firewall

To explain our model, in the layer 1, there are multiple high-speed network traffic capturing devices deployed for capturing and collecting huge amount of raw data transmitted through network terminals such as routers, gateways, switches, hubs. After that there is data extractor which used to preprocess raw data included extracting and formatting raw data into HTTP request like our training dataset. Later on, all extracted data will be delivered to layer 2.

In layer 2, decision maker server will analyze and evaluate whether one HTTP request/query is legitimate by matching in-memory knowledge base. If the certain HTTP request could not be identified by decision maker server, the HTTP request/query will be transferred into layer 3 for further machine learning approach. If the certain HTTP request could be identified as malicious or legit, it will also be transferred into layer 3 for training and testing purpose.

In layer 3, the machine learning server will use various machine learning techniques to identify each unidentified HTTP request/query, and then once it is identified, it will be sent to in-memory knowledge base for updating purpose. Thus, if the decision maker server analyzed the same attack, it were able to identify this certain attack by matching updated in-memory knowledge base.

All in all, more and more researchers and companies interested in machine learning mechanism to build and optimize their web application firewall, for example, the Zenedge Cybersecurity Suite[15]. And the real-time machine learning driven WAF would become more and more popular and reliable in modern internet era.

8. Reference

[1]. Web Based Attacks
https://www.sans.org/reading-room/whitepapers/application/web-based-attacks-2053

[2]. Vangie,. (n.d.). The 7 Layers of the OSI Model. Retrieved December 18, 2017, from https://www.webopedia.com/quick_ref/OSI_Layers.asp

[3]. OWASP Top 10 - 2017 The Ten Most Critical Web Application Security Risks
https://www.owasp.org/images/7/72/OWASP_Top_10-2017_%28en%29.pdf.pdf

[4] Lee, Stolfo, Chan, Eskin, Fan, Mille, Hershkop, and Zhang, "Real Time Data Mining-based Intrusion Detection," IEEE, 2001.

[5] A multi-model approach to the detection of web-based attacks
https://pdfs.semanticscholar.org/d42c/31d59759168036c449fbeb251e313b21f87e.pdf

[6] Web threat detection via web server log analysis
http://web.stanford.edu/class/cs259d/lectures/Session8.pdf

[7] Data Science usage in Web Application Attacks, http://www.jianshu.com/p/942d1beb7fdd

[8] Git All the Payloads! A collection of web attack payloads.
https://github.com/foospidy/payloads

[9] SQL Injection, https://www.owasp.org/index.php/SQL_Injection

[10] Zhang M L, Zhou Z H. ML-KNN: A lazy learning approach to multi-label learning[J]. Pattern Recognition, 2007, 40(7):2038-2048.

[11] Cross-site Scripting (XSS)
https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)

[12]Directory Traversal Attacks
https://www.acunetix.com/websitesecurity/directory-traversal/

[13] SecRepo, http://www.secrepo.com/

[14] Baranwal, A. K. (n.d.). Approaches to detect SQL injection and XSS in web applications. Retrieved December 18, 2017, from http://www.bing.com/cr?IG=06F9B0141E8F43C6A6C3BBF19155379E&CID=05CDE15038BB65782061EA0D391464C5&rd=1&h=IrV1rmPK0i14ptjasQId_0gO0WLKggN1Db8-tZE7Jsc&v=1&r=http%3a%2f%2fblogs.ubc.ca%2fcomputersecurity%2ffiles%2f2012%2f04%2fABaranwal_ApproachesToDetectSQLinjection_XSSinWebApplication.pdf&p=DevEx,5035.1

[15] Zenedge, https://www.zenedge.com/

[16] Ahmad, A., Paul, A., & Rathore, M.M. (2015). “Real time intrusion detection system for ultra-high-speed big data environments.” The Journal of Supercomputing, 72, 3489-3510.

Chenfeng Nie, 10
Shuaichen Wu, 10
Xiaoyi Rao, 10

The workload is pretty balanced and every member in group has positive attitude and make effort for one same target. Great Job!