International Journal of Electrical and Computer Engineering (IJECE)
Vol. 10, No. 5, October 2020, pp. 5335~5346
ISSN: 2088-8708, DOI: 10.11591/ijece.v10i5.pp5335-5346

An adaptive anomaly request detection framework based on dynamic web application profiles

Cho Do Xuan[1], Nam Nguyen[2], Hoa Nguyen Dinh[3]
[1,3]Information Security Department Posts and Telecommunications Institute of Technology, Vietnam
[1,2]Information Assurance Department FPT University, Vietnam

Article Info

Article history:

Received Nov 15, 2019
Revised Mar 23, 2020
Accepted Apr 13, 2020

Keywords:

Anomaly request
Data clustering
Dynamic profiling
Web application firewall

ABSTRACT

Web application firewall is a highly effective application in protecting the application layer and database layer of websites from attack access. This paper proposes a new web application firewall deploying method based on dynamic web application profiling (DWAP) analysis technique. This is a method to deploy a firewall based on analyzing website access data. DWAP is improved to integrate deeply into the structure of the website to increase the compatibility of the anomaly detection system into each website, thereby improving the ability to detect abnormal requests. To improve the compatibility of the web application firewall with protected objects, the proposed system consists of two parts with the main tasks are: i) Detect abnormal access in web application (WA) access; ii) Semi-automatic update the attack data to the abnormal access detection system during WA access. This new method is applicable in real-time detection systems where updating of new attack data is essential since web attacks are increasingly complex and sophisticated.

Corresponding Author:

Cho Do Xuan,
Department of Information Security,
Posts and Telecommunications Institute of Technology,
122 Hoang Quoc Viet, Cau Giay District, Hanoi, Vietnam.
Email: chodx@ptit.edu.vn

1. INTRODUCTION

Currently, web application security has been a hot topic for many researchers and internet service providers. According to the Symantec report [1], web-based attacks accounted for 10 percent of total malicious requests detected in 2018, and website security is a necessary issue at present. In [2], Mookhey presented the characteristics, compositions and operation principles of WA. Besides, some other works in [2-4] have shown several vulnerabilities and threats that attackers could exploit to attack the web applications. According to the surveys in [3] and [4], the vulnerabilities of the protocol hyper text transfer protocol (HTTP) or hypertext transfer protocol secure (HTTPS) are often preferable to be exploited by attackers. HTTP and HTTPS are the two most popular protocols to communicate for end users. Before returning the contents to display on the web browser, web applications process the content of the user requests. According to the standard described in [5], the structural components of an HTTP or HTTPS packet include request line, status line, header fields, message body, and some other components. Here, in order to attack web applications, attackers will try to change the content of these components, thereby creating a vulnerability in the process of processing request. As a result, the web applications return the outputs as attacker desired.

So far, there are two main methods of detecting web application attacks: signature-based methods based on a set of predefined rules, and anomaly-based methods that rely on data analyzing and statistics to find abnormal characteristics in the requests. Both signature-based and anomaly-based methods have certain

advantages and disadvantages. In general, solutions applying signature-based techniques [2, 3] are not able to detect unusual requests since these methods are mainly based on fixed ruling systems. Anomaly-based methods are capable of detecting abnormal requests because they utilize techniques to analyze and evaluate the behaviors of the requests. There have been many studies using anomaly-based analysis techniques to detect unusual requests. Habeeba et al. [6] listed a number of issues in detecting abnormalities in cyberspace in general and in web access in particular. Especially, they made a survey on methods and techniques commonly applied in big data analysis, and pointed out many benefits to anomaly-based detection approaches. However, anormaly-based methods also have some problems presented as follows:

a. All anormaly requests' characteristics could not be found in training datasets

Accordingly, datasets for unusual access detection used in a number of previous studies were collected through the test results of available security tools, firewalls, etc. As a result, those datasets contain almost all attack requests [7]. More over, in reality, there are also many accesses, which do not contain attack contents, having the same characteristics and structure as abnormal requests. According to this study, normal requests are ones that perform legal operations and contains information that follows the prescribed standards. In contrast, if requests are different from the specified criteria, they will be considered as abnormal requests. Generally, abnormal requests can be expressed in two cases:

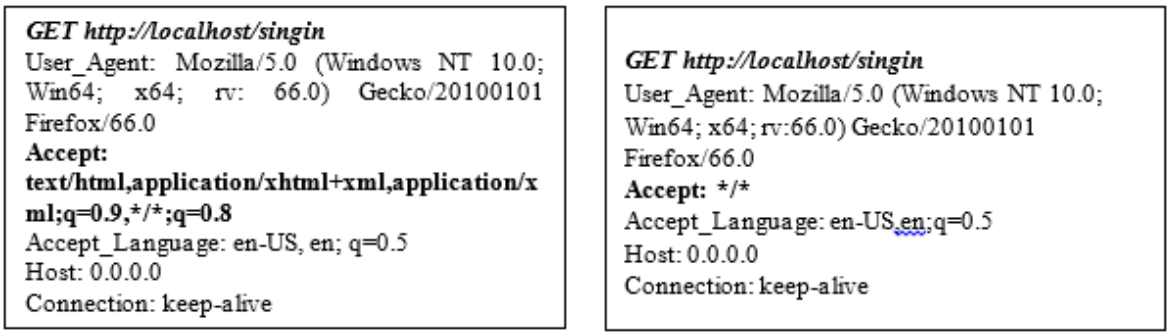
- Website attack requests,
- The requests identify and exploit website vulnerabilities.

It can be said that attacking requests are just one part of abnormal requests. From the incomplete definition of abnormal accesses, it may lead to the lack of objectivity when building properties for the model for attack request detection from previous studies. Since anomalous attributes are built mainly on the attacking request data, those attributes only focus on representing the characteristics of the attack requests.

b. Feature extraction methods could not present the characteristics of web applications

Previous studies on abnormal detection systems mainly used website data, but did not characterize each Uniform Resource Identifier (URI). The URI is the path that identifies a website's resources. The resources in each URI can be information in form of HTML, or it can be login tasks, registration, information search. If the URI characteristics are not carefully exploited for attack request detection, some issues may emerge as follows:

Model may not dig into the insightful characteristics of each URI since the extracted features are made to present the characteristics of all URIs. For example, the feature presenting the URI information length used in previous studies is usually short for URIs having information in HTML form, while it is usually long for URIs in a request form following GET method. This leads to the fact that when initialize this variable with the entire URI, its value becomes spare with a large variance, and may not present the abnormality when used in the model. Consequently, the model will cause confusion in detecting abnormal requests at different URIs. Figure 1 below shows an example of the contents of a normal access 1(a) and a abnormal access 1(b).



(a) (b)

Figure 1. Sample contents of normal request (a) and an abnormal request (b)

In Figure 1, it can be seen that in normal requests, all accepted values are "text/html,application/ xhtml+xml,application/xml;q=0.9,*/*;q=0.8," while in operations that are not like normal users, may be due to the scanner tools or intentional change of attacker, the accepted value is set to "*/*" (accepted all). This is an important characteristic to determine the abnormal requests in Figure 1(b). There are two types of missed classification: False Negative (FN) and False Positive (FP). FN is when an abnormal request is missed classified as normal. This usually happens when the number of request samples having value Accept being

"*/*" in the whole website is considerable greater than other types of requests. In this situation, the abnormal characteristic of the abnormal request will be ignored and that request is decided as normal. On the contrary, FP presents normal request missed classified as abnormal ones. For example, in case the website has a form filling function and when the content of the form contains the keyword found to be abnormal based on rule-based systems like Mod-security, it will be blocked. However, when that content is acceptable for the website, the administrator needs to edit the filter to add an exception to this case and needs to enrich the rule list. This process sometimes is complicated and can make web application protection less effective.

The issues discussed in a) and b) are our practical experience when implementing different unusual access detection models. The effectiveness in practice of previous studies in detecting unusual requests is not high. In fact, when those approaches are applied on abnormal data using ModSecurity tool [8], the highest recall score is only about 30%. In order to overcome the weaknesses that have been pointed out, in this paper, a new method to build a web firewall based on Dynamic Web Application Profiling (DWAP) analysis is proposed. DWAP is a method to summarize the characteristics of a specific website's URIs. These properties include methods (GET, POST), headers and parameters of the URIs. Based on the DWAP application analysis, the following contributions are presented in this paper:

- Applying DWAP to abnormal detection systems. The problems discussed above can be solved if the detection model is trained using each URI since the variable values generated from each URI are no longer spare and anomalies are easily recognized if the abnormal feature described in the example in b) is detected. Moreover, by developing a separate model for each URI, it is possible to extract new features presenting the characteristics of the method, header and parameters of each URI. This could not be done in previous approaches. These features fulfill the ultimate purpose of DWAP application that is to optimize the abnormal request detection on each URI.
- Apart from applying DWAP to detect abnormal requests, a real-time optimization for model update method is also presented. This issue plays a very important role in the web attack detection model as well as anomalous access detection based on anomaly-based algorithms. However, previous works did not pay much attention to this problem. All current security applications need to be constantly updated to accommodate new attacks. That is the main reason why Mod Security is still a popular security detection tool today because its rule system is kept up to date and maintained by community contributions. It can be seen that anomaly-based models need to be trained based on the data from the specific concerned website. In fact, the number of unusual requests is much smaller than normal requests, which generates a burden job for administrators in composing training data. In order to tackle this problem, a request grouping method is proposed to support the data classification process. This method can help reduce administrator's data composing time by 50-70%, thus making our proposed anomaly-based detection model easy to deploy in practice.

Experimental results on the same data set show that there is a significant improvement in detection performance of our method. The recall index of the new approach can reach 90%. The following content of the paper is organized as follows. Section 2 presents all related works on abnormal request detection techniques. The newly proposed method is presented in section 3. Section 4 introduces some main applications of the new framework. Experimental results and all discussions are included in section 5. Section 6 concludes what have been done and discusses some suggestions for the future works.

2. RELATED WORK

2.1. Web attack detection research

There are two main types of web attack detection systems. The classification is mainly based on the detection mechanism of the methods.

- Signature-based methods [3, 4]: this is a well-known approach and has been investigated by many researchers. So far, the research community of web attack detection has built up a complete Core Rule Set [9] to support network users. Currently, the Core Rule Set is used in most of the web firewalls [3].
- Anomaly-based methods: there have been many different anomaly based approaches on network security. One of those approaches is based on the manual feature extraction techniques. Shi and et al. [10] present a list of features for queries that include URI's properties, such as length, quantity, type and dangerous levels of each feature. After that, they applied Naïve Bayes, Decision tree and SVM algorithm on those features to detect abnormal requests. Aother approach is based on the natural language processing. Zhang M and et al. [11] introduced a method that uses CNN to classify the attacks. Word2vec model is used to transform the raw request into a matrix, and then a CNN is adopted to extract request's features. The research [12] introduces another approach using Gated recurrent unit (GRU) to analyze the contents of the requests. Every character in the request is converted into a one-hot vector with 129 dimensions, and every cell in GRU is used to analyze this request's content. Yang [13] also attempts a similar method that

uses GRU to classify requests. In this research, he uses an encoding method which reconstructs a character into a 2-dimensional matrix. The authors of the research [14] use N-gram and Generic Feature Selection algorithms to extract features from DARPA and ECML/PKDD2007 datasets [4]. In order to detect abnormal requests, they applied some clustering algorithms like C4.5, CART, Random forest or random tree. Aside from applying anomaly-based to detect abnormal requests in general, there are also some other researches focusing on detecting some common attacks on web application [15, 16]. In particular, Nagarjun and Ahamad [17] presented an attack detection method based on image processing technique to detect special characters that represent XSS attacks. Yong Yang [18] introduced an approach to detect anomalies by analyzing the sequence of web access behaviors. In addition, Jagdish et al. [19] designed an anomaly detection system in E-commerec systems based on features showing business characteristics such as price, goods, etc. Thes features are also adopted in this paper, but at a more general level and the extraction process of these features is implemented automatically.

2.2. Data updating and monitoring research

To overcome data imbalance problem in the training process as well as in the abnormal request detection process, there have been some researches and proposals. Hu Y [20] proposes a human-machine system to improve detection models. On this system, the role of the expert is to re-classify the data after running the unlabeled classification. The author uses K-mean to classify the dataset into two groups and selectes a certain percentage from those 2 groups to reclassify. In the research [21], Dong and et al. presente a solution to reclassify requests which are not in the boundary trained by the SVM algorithm.

3. FEATURE EXTRACTION USING DWAP

3.1. Dynamic web application profiling

DWAP represents the URI that maps a web application into a tree. DWAP contains information of all URIs in a web page, including static URI and dynamic URI. Static URI is a file path of static files such as media file [*.jpg, *.png, *.gif] or files being used to display web page [*.css, *.js, *.html]. Dynamic URI transfers parameters to the web application that it processes. Figure 2 illustrates an example of a DWAP of a web application. There are 2 examples of dynamic URI, which are “https://web.com/login.jsp” showing the purpose of login, and “https://web.com/product/view.jsp” showing the purpose of viewing products. These URIs have different queries. For example, the login.jsp will need the value of user’s id and user’s password, while the view.jsp will need information about the displayed product ids. The property of this information is different in terms of value, data type, etc. It is necessary to build individual models for each URI to be able to recognize the smallest abnormal changes to avoid confusion when analyzing multiple URI’s information at the same time. In this research, we are focusing on analyzing and evaluating the behavior of the dynamic URI.

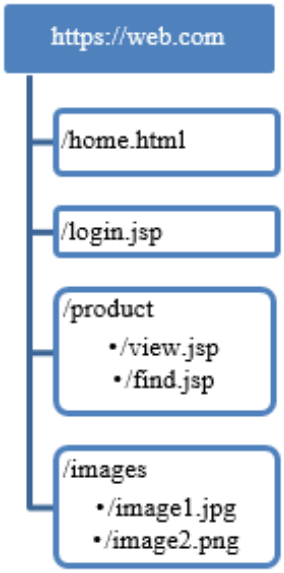


Figure 2. Dynamic web application profiling sample

3.2. Feature extraction in DWAP analysis

The request's features in DWAP analysis method are built to detect the abnormal requests at the component level. By analyzing on each URI, every request's component like header or set of parameters is deeply analyzed. In order to do that, the feature set is divided into 2 groups. The first group is used to look for abnormal characteristics appeared in the attacks. The second group is exploited to analyze abnormal content in each component of the request.

3.2.1. Malicious keywords feature

a. Attack keywords

Keyword is the main identification characteristic of some types of attacks. For example, in SQL Injection attack, the attackers try to find a way to insert their SQL queries into the data sent to server. The appearance of those keywords in the request is a sign to determine whether a request is an attack or not. The keywords listed in Table 1 are summed up from OSWAP's document about web attacks [9]. In those attacks, the keyword is the most important component to insert illegal queries. These keywords are put into OSWAP's rules to detect the attacks. However, the evaluation based only on the appearance of the keyword may lead to the incorrect warning because some websites allow the existence of those keywords. Therefore, when building the URI's features, the appearance of these keywords as a feature should be combined with other factors in order to successfully conclude whether the request is an attack or not. Table 1 summarizes some Malicious keywords.

Table 1. Malicious keywords

Group	List of keyword
SQL injection	Create, drop, alter, truncate, select, insert, update, delete, or, and, union, join, where, if, convert, cast, hextoraw
OS Command Injection	Cat, ls, mv, cp, rm, perl, python, ruby, lua
XML Injection	lt, gt, node, xml, encoding, cdata, doctype, element
Cross-site Scripting	Onload, onmousehover, onerror, script, iframe, var, alert

b. Anomaly non-letter

Non-letter features are one part of the attack signs when they accompany with the keywords shown in Table 1. In order to insert malicious keywords, attackers must find the way to pass the application's compiler. For example, they may insert comments or characters to deceive the input structure. In this paper, the frequency of those characters in the queries is used to verify the input information. These character groups are listed in Table 2.

Table 2. Non letter keyword

Group	List of non-letter characters
The characters used to start a comment or to mark a block comment	-- [start comment] /* [start comment] */ [end comment] * [Asterisk] # [hash] || [double vertical bar] && [double Ampersand] ; [Semicolon] ` [Grave Accent]
Characters used in commands	& [Ampersand] && [double Ampersand] | [vertical bar] || [double vertical bar] ` [Grave Accent] ; [Semicolon] $ [Dollar Sign]
Characters to insert statements	> [Greater-Than Sign] < [Greater-Than Sign] >> [double Greater-Than Sign] # [hash] ! [Exclamation Mark] - [Hyphen-Minus]? [Hyphen-Minus] = [Equals Sign] [[Left Square Bracket]] [Right Square Bracket] ~ [Tilde] . [dot] , [Comma] % [Percent Sign] : [Colon] + [Plus sign] ([Left Bracket]) [Rigth Bracket] { [Left Curly Bracket] } [Right Curly Bracket] /> [backslash greater] ../ [dot dot slash] ..\ [dot dot backslash]
Characters cannot be displayed	Characters with Unicode code smaller than 32

3.2.2. Anomaly request content feature

In previous researches on abnormal access detection, the content values of each request component are not thoroughly analyzed. The main reason is about the difference between the functions of different URIs. Each URI uses different configuration for its header as well as its query values. This causes difficulty in proposing an effective method based on analyzing multiple URIs to detect anomalies such as exploiting logic errors or detecting requests from scanning tools. In this paper, a set of new features extracted through the content analyzing process for each component of a request is utilized for anomaly detection process. Using this feature extraction helps review and evaluate the values of the request components in details and clearly. For instance, in real scenarios, abnormal requests can be identified if there is a strange URI (such as webshell), or a strange field in the header (such as X-Forwarded-For used to bypass firewall), or a new

parameter that seldom or never appears in requests by normal users. The proposed features can be used to build the profile tree. Further, this tree is used to detect abnormal requests by comparing their contents with the data in the tree. If there is any mismatching between the requests and the profile tree, those requests may be considered as abnormal. This is the improvement of the DWAP analysis based method compared with previous methods.

a. Anomaly header value

Headers are important and are frequently changed targets. In this paper, four main header fields are investigated, which are Content-Type, Accept, Accept-Charset, Accept-Encoding. The values of those headers are extracted and normalized to form a vector. In order to facilitate the feature classification process, the features are divided into groups as below:

– Group 1: including the values of Content-Type [5] and Accept [5]. Its structure contains type and sub-type. The value of this type and subtype are compared with that of type and subtype list in normal requests. Methods and procedures of this investigation are described in Algorithm 1:

```
Algorithm 1. Check value group 1
Input: Content-type, Accept
Output: Vector represent existing value of header
1: function: CHECK_VALUE_GROUP_1 (header):
2:  types/subtypes <- header
3:  normal_types <- list type of header in normal requests
4:  normal_subtypes <- list subtype of header in normal requests
5:  type_feature <- a size of normal_types-array of 0
6:  subtype_feature <- a size of normal_subtypes-array of 0
7:  for i <- 0 to size of types:
8:    if types_i exist in normal_types
9:      type_feature [position of types_i in normal_types] = 1
10: for i <- 0 to size of subtypes:
11:   if subtypes_i exist in normal_subtypes
12:     subtype_feature [position of subtypes_i in normal_subtypes]=1
13: return type_feature+subtype_feature
```

– Group 2: including the values of Accept-Charset [5] and Accept-Encoding [5]. The features of Group 2 are extracted by comparing these values with those in the normal requests. Methods and procedures to extract feature values are described in the Algorithm 2:

```
Algorithm 2. Check value group 2
Input: Accept-Encoding, Accept-Charset,
Output: Vector represent existing value of header
1: function: CHECK_VALUE_GROUP_2 (header):
2:  values <- header
3:  normal_values <- list normal value of header in normal requests
4:  values_feature <- a size of normal_values-array of 0
5:  for i <- 0 to size of normal_values:
6:    if value_i exist in normal_values:
7:      values_feature [position of value_i in normal_values] = 1
8: return values_feature
```

b. Anomaly parameter

```
Algorithm 3. Get Structured data feature
Input: request
Output: key-value feature
1: function GET_ STRUCTED_DATA_NAME(structed_data)
2:  normal_key <- list normal key in normal requests
3:  request_name <- list name in request
4:  num_abnormal_key <- 0
5:  name_value_feature <- []
6:  for name in request_key
7:    if key not in normal_key
8:      num_abnormal_key +=1
9:    else
10:       key_value_feature <- key_value_feature + [raito of alphabet character in value,
raito of digit in value, raito of other character in value]
11:   key_value_feature <- key_value_feature + [num_abnormal_key]
12: return num_abnormal_key
```

User requests usually contain important information for web-server to process. The content of the request may be presented in a form of structured data such as the query in GET method or the payload in POST/PUT method, or unstructured data like documents, files, etc. For structured data, the values of length, ratio of letters and numbers in each input string are extracted. Additionally, the existence of abnormal parameters is also checked. Methods and procedures to to examine anomaly parameters are described in the Algorithm 3 above.

4. APPLICATION OF DWAP ANALYSIS ON WEB APPLICATION SECURITY

4.1. DWAP analysis for anomaly request detection

Based on the features obtained from the DWAP analysis technique applied on the request component presented in Section 3.2, further processing steps are needed to discriminate normal accesses from abnormal ones. In this paper, Random Forest classifiers [22] are adopted to distinguish between abnormal and normal requests. Random Forest is an ensemble classification method [23]. This algorithm is based on an ensemble of classifiers, which normally are Decision Trees to make the final prediction. The theoretical foundation of this algorithm is based on Jensen's inequality [24]. According to Jensen's inequality applied to the classification problems, it is shown that the combination of many models may produce less error rate than that of each individual model.

4.2. DWAP analysis for constructing training datasets

The main characteristic of the abnormal request detection method using DWAP analysis method is that it does not use existing datasets for training data, but it utilizes the data of the deployed website. In fact, the number of anomaly requests is much smaller than that of normal requests in the whole dataset. As a result, it is necessary to have suitable sampling methods and techniques to create a good training dataset that helps theabnormal request detection process become more effective. From this point, a new sampling method based-on DWAP analysis technique and unsupervised learning algorithm is proposed. This method firstly divides the data into different clusters. Then, it selects requests from the newly divided data clusters. The combination of the DWAP analysis technique and the unsupervised learning algorithms not only ensures the randomness of sampling, but also increases the rate of abnormal requests that appear in the sample data. Consequently, this helps generate a more balanced training dataset, and reduce time and effort to search for abnormal requests. The proposed method can be summarized as follows:

Step 1: Data clustering: this step aims at aggregating requests that have similar characteristics. Data Clustering is known as a method to gather correlated observations in to separate groups. This method has been adopted by Riyaz [25] for deployment on large databases and has shown that practical applications of these clustering algorithms are promissing. Since the features are extracted such that they can distinguish between normal and abnormal requests, the clustering process of these features not only separates normal and abnormal requests, but also classifies the attack requests in different forms. The remaining issue is to find the optimum number of clusters for the data. In this paper, K-mean algorithm is adopted for clustering task. This clustering method is based on the minimization of the distances from all data points within each cluster to the cluster centroid [26]. In order to find the number of the clusters for the K-mean algorithm, the Elbow method is used. This method is based on the graph presenting the correlation between the total distances from all data points in each cluster to their cluster centroid and the number of clusters. The Elbow criterion is met when the number of clusters N is chosen such that the ratio between the total distance with *N* groups and that of *N+1* groups is smallest. The Elbow method is summarized as follows:

- Let ΔSSE_i is the total sum of squared error distances of *i* clusters
- Let r_i is the ratio between ΔSSE_i and ΔSSE_{i+1}

$$r_i = \frac{\Delta SSE_{i+1}}{\Delta SSE_i} \quad (1)$$

- The optimal number of cluster *N* corresponds to the smallest r_i:

$$N = argmin_i\{r\} \quad (2)$$

Step 2: Sampling data from clusters: the process to take *M* samples from *N* groups**:**

- If the number of samples in one particular cluster is smaller than $\frac{M}{N}$, then all samples in that cluster are selected. The reason for this is that the number of abnormal requests is very small compared to normal ones, and due to anomaly characteristics, they are usually not in the same category as normal requests.

As a result, abnormal requests tend to be separated in small clusters. So after this process, the remaining number of samples need to be taken is M_1 and the remaining number of clusters is N_1.

– Repeat (i) with the number of samples need to be taken as M_1 and cluster number is N_1. The sampling process will end after i iterations when the numbers of samples in all remaining clusters are greater than the ratio $\frac{M_i}{N_i}$.

– From each of remaining clusters, $\frac{M_i}{N_i}$ samples are randomly selected.

The whole process is presented in Algorithm 4:

```
Algorithm 4. Sampling data
Input: clustered_data
Output: sampled data
1: function DATA_SAMPLING (clustered_data, M)
2: sampled_data [] <- empty list
3:  clusters [] <- list of cluster in clustered_data
4:  number_of_data_in_cluster [] <- number of data in each cluster
5:  N <- number of cluster
6:  while existing clusters[i] which has number of data is smaller than M/N:
7:    sampled_data = sampled_data + data in clusters[i]
8:    M = M – number_of_data_in_cluster[i]
9:    N = N -1
10:   clusters pop i
11: for cluster in clusers:
12:   sampled_data <- sampled_data + random choice M/N data in cluster
13: return sampled_data
```

Discussion: If the rate of the abnormal request, K_1, in one dataset is very small, i.e. $K_1 << 1$, then among M selected samples, the rate of anomaly request is still K_1. Moreover, anomaly requests are usually separated from normal requests after the clustering method. Although there is no guarantee that all data samples in each cluster have the same label but if $K_1 << 1$, there is a great chance that the almost all number of anomaly requests are selected from small clusters. As a result, the clustering method combined with sampling algorithm proposed in this paper can efficiently filter out almost all abnormal requests, which can help reduce the building time of the dataset for the DWAP analysis. This sampling process has more advantages than random sampling approach.

5. EXPERIMENT

5.1. Dataset

In order to evaluate the efficiency of the proposed algorithms, two 2 datasets are used.

- Dataset 1: The first dataset is CSIC 2010 [7], which is developed by Carmen. The dataset includes about 36000 normal requests and 25000 abnormal requests. Since most of the samples in the CSIC 2010 dataset are attacking requests, it may not be suitable for evalulating the detection of the abnormal requests. CSIC 2010 dataset is filtered and divided into 8 main URI groups as presented in Table 3.

Table 3. Statistics of the number of normal requests and abnormal requests in the dataset 1

Order	URI	Normal	Abnormal
1	/tienda1/publico/anadir.jsp	4000	2821
2	/tienda1/publico/autenticar.jsp	4000	2783
3	/tienda1/publico/caracteristicas.jsp	4000	1957
4	/tienda1/publico/entrar.jsp	4000	1835
5	/tienda1/publico/pagar.jsp	4000	2722
6	/tienda1/publico/registro.jsp	4000	2781
7	/tienda1/publico/vaciar.jsp	4000	1884
8	/tienda1/miembros/editar.jsp	4000	2774

– Dataset 2: The second dataset is made by using some security tools like Acunetix, Burp Suite, SQLMap to scan the vulnerabilities from our prototype websites. Those scanning tools search and exploit vulnerabilities in both the query and the request's headers. The collected data is classified following the defined criteria in the previous section. Besides the abnormal requests collected by scanning tools, we made some normal requests by normally operate on the same URIs. Each URI contains 5000 normal requests and 5000 abnormal requests.

5.2. Classification measures

5.2.1. Evaluation criteria to detect abnormal request

In this research, three evaluation metrics are used as follows:

- Precision is defined as the ratio between the number of true positive alarms (*TP*) and all the samples classified as positive (*TP* + *FP*). The higher the precision score, the more number of positive alarms are correct.

$$precision = \frac{TP}{TP + FP} \quad (3)$$

- Recall is defined as the ratio of true positive alarms among all samples that are actually positive.

$$Recall = \frac{TP}{TP + FN} \quad (4)$$

- F1-score is the harmonic mean of precision and recall.

$$F1 = \frac{2 * precision * Recall}{precision + Recall} \quad (5)$$

where, *TP* - True positive: is the number of records that are correctly labeled as "abnormal requests"; *FN* - False negative: is the number of records that are actually "abnormal requests" but are classified as "normal requests"; *TN* - True negative: is the number of records that are correctly labeled as "normal requests"; *FP* - False positive: is the number of records that are actually "normal requests" but are misclassified to "abnormal requests".

5.2.2. Criteria for evaluating the effectiveness of applying DWAP analysis for constructing training dataset

In order to evaluate the efficiency of the sampling method in the construction of the training data, the imbalance in sampling process between the random sampling method and the newly proposed sampling method. This value is expressed by parameters K_1 and K_2 as follows:

- The parameter K_1 represents the proportion of abnormal requests in the data recognized by random sampling methods.
- K_2 is the ratio of abnormal requests in this proposed sampling method.

In this paper, these two values are compared with respect to different K_1 values and the number of selected number of samples *M*.

5.3. Experimental results and comments

5.3.1. Criteria experimental scenarios and experimental results for detecting abnormal request

a. Experimental scenarios

The efficiency of DWAP analysis techniques in detecting abnormal requests using Random Forest clustering algorithm is evaluated based on both datasets described in section 5.1. All three performance metrics are recored. Each dataset is divided into two subsets: the training data containing 80% of the dataset is used for training the classification model; remaining 20% of the data is used for testing. The number of trees for Random Forest algorithm is set at 300.

b. Experimental results and comments

Experimental results of using DWAP analysis technique to detect abnormal requests on datasets 1 and 2 are shown in Tables 4 and 5. The results in Table 4 show that using DWAP analysis techniques can accurately and efficiently detect abnormal requests. In particular, Precision scores across all data range from 99.46% to 100%. This result shows that the positive alarm of this method is very reliable.

Table 5 shows that even when the dataset contains a higher percentage of normal request as illustrated in dataset 2 the new DWAP analysis technique is still highly effective, while traditional toolset using ModSecurity rules are not efficient. Specifically, recall value of the toolset is just 30% while that score of the proposed method is more than 90%. Besides, the new method can obtain perfect precision score on all URI sets. The F1 score of the proposed method is also much higher, which is over 95%, compared to the toolset. The results shown in tables 4 and 5 demonstrate that DWAP analysis techniques are not only able to efficiently detect requests attacks, but they are also capable of correctly detecting abnormal requests.

Table 4. Experimental results of abnormal requests detection using DWAP analysis method on dataset 1

Order	URI	Precision	Recall	F1 Score
1	/tienda1/publico/anadir.jsp	100%	100%	100%
2	/tienda1/publico/autenticar.jsp	99.46%	99.28%	99.37%
3	/tienda1/publico/caracteristicas.jsp	100%	100%	100%
4	/tienda1/publico/entrar.jsp	100%	100%	100%
5	/tienda1/publico/pagar.jsp	100%	100%	100%
6	/tienda1/publico/registro.jsp	99.63%	95.33%	97.94%
7	/tienda1/publico/vaciar.jsp	100%	100%	100%
8	/tienda1/miembros/editar.jsp	99.63%	96.04%	98.23%

Table 5. Anomaly detection performance comparison between DWAP analysis methods and ModSecurity tool applied on dataset 2

Order	URI	DWAP analysis + Random Forest			ModSecurity [8]		
		Precision	Recall	F1 Score	Precision	Recall	F1 Score
1	/signin	100%	96.77%	98.36%	98%	30%	46%
2	/signup	100%	91.41%	95.51%	99%	40%	57%

5.3.2. Experimental setup to evaluate the construction of training data using DWAP analysis method

a. Experimental setup

The effectiveness of the data construction process is evaluated by using the changing hyper parameters as follows:

– Distribution rate of anomaly requests in data: different values of K_1 ranging from 1% to 30% on the URIs of the CSIC 2010 dataset is used, based on which the evaluation of K_2 can be obtained through the proposed sampling process. The features extracted in section 3.2 are used to represent all data points.
– Number of samples M: Since the value of K_2 depends mainly on the number of M selected samples, the influence and dependence of K_2 when the value of M changes can be used to show the efficiency of the sampling method. This factor plays an important role in sampling process such that the abnormal request distribution in the dataset is optimized.

Figure 3 presents the clustering results based on the value of the SSE of the clusters. From Figure 3 it can be seen that the SSE value varies a lot when $N = 2$ and $N = 3$, resulting to the ratios r_2 and r_3 are almost equal to 1. When $N = 4$, the variation of SSE decreases significantly, so is the value of r_4. SSE returns to a little variation when $N > 4$ resulting to r_5 and r_6 are almost equal to 1. Therefore, $N = 4$ is chosen as the cluster number of the data.

Figure 4 illustrates the distribution of normal requests and abnormal requests in each cluster after the K-mean algorithm. The data is classified into 4 clusters and the distribution of all labels are shown on the graph. The ratio of abnormal requests in this scenario is $K_1 = 31\%$. When applying the proposed sampling method with $M = 1000$, the percentage of abnormal requests in the sampled data reaches $K_2 = 71\%$.

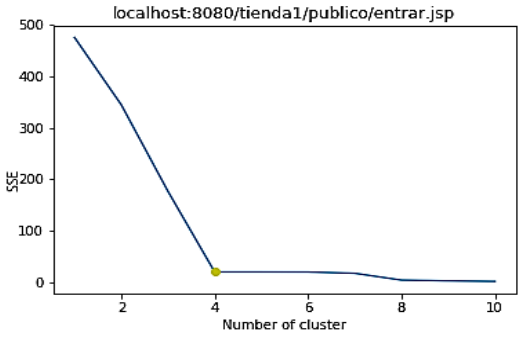


Figure 3. Example of cluster number selection

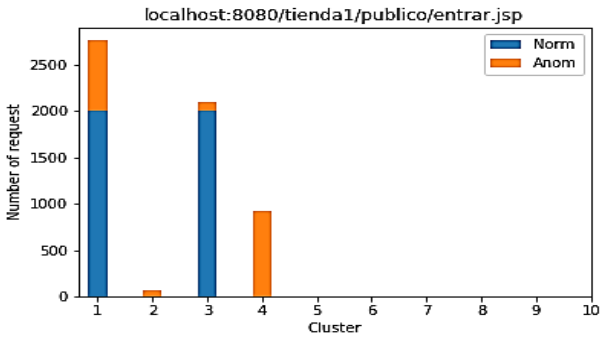


Figure 4. The request distribution of a URI in clusters

b. Experimental results and comments

The results show the effectiveness of the proposed sampling method compared to random sampling method. Additionally, the comparison between the distribution of anomaly requests in the data of both methods are also recorded. Table 6 shows the change in K_2 value when the value of K_1 varies from 1 to 30%. The result of Table 6 shows that K_2 is greater than K_1 in different K_1 distributions.

Table 6. The ratio of K_2 corresponds to the different cases of K_1 of URIs with M = 1000

URI	K_1	K_2	K_1	K_2	K_1	K_2	K_1	K_2	K_1	K_2	K_1	K_2	K_1	K_2	K_1	K_2
1	1	1.31	2	3.27	4	6.83	7	10.57	10	19.14	15	23.81	20	29.19	30	35.95
2	1	1.01	2	2.52	4	5.99	7	10.27	10	18.72	15	15.74	20	30.83	30	56.64
3	1	1.80	2	4.37	4	6.29	7	14.08	10	18.19	15	27.42	20	35.91	30	45.70
4	1	2.48	2	6.00	4	12.65	7	7.19	10	31.99	15	42.14	20	53.12	30	52.26
5	1	2.29	2	3.98	4	10.16	7	7.79	10	22.75	15	15.95	20	34.40	30	62.01
6	1	1.0	2	2.05	4	4.99	7	7.06	10	10.97	15	15.06	20	21.11	30	35.99
7	1	2.11	2	3.66	4	8.26	7	28.86	10	10.62	15	53.79	20	46.14	30	50.22
8	1	1.0	2	2.93	4	4.52	7	7.85	10	10.41	15	17.78	20	20.80	30	32.12

From the results in Table 6, it can be sees that the distribution of anomaly request in the proposed data is higher than that of anomaly request obtained by the random selection method. On average, the ratio of K_2 is 1.5 to 2 times higher than the ratio K_1. It also shows that when the DWAP analysis method is combined with the proposed sampling method, the ratio of detected abnormal requests in clustered data is much higher than using the random sampling method. Table 7 shows some changed results of K_2 when the number of sample M changes.

Table 7. Some examples of K_2 results when changing the number of samples

M	K_1	K_2	K_1	K_2	K_1	K_2	K_1	K_2	K_1	K_2	K_1	K_2	K_1	K_2	K_1	K_2
250	1	4.00	2	8.62	4	19.16	7	26.02	10	33.89	15	36.97	20	43.87	30	49.55
500	1	2.50	2	5.38	4	11.68	7	16.46	10	30.94	15	33.90	20	40.22	30	48.42
1000	1	1.61	2	3.47	4	7.34	7	11.33	10	17.47	15	26.21	20	33.81	30	46.36

Based on the results of Table 7 it can be seen that when reducing M, K_2 tends to increase because the number of items taken in each group will be smaller. As a result, the number of requests retrieved in each group will decrease. This leads to the possibility that there are some small groups taken entirely when they have less than $\frac{M}{N}$ elements. At this time, although the number of selected anomaly requests is reduced, the number of selected normal requests is much more reduced. Consequently, the proportion of the anomaly requests will increase. It means that these groups are treated fairly with normal request groups. When the number of clusters increases, the number of anomaly request groups also increase. This results in the ratio of selected anomaly requests from the dataset approaches the proportional of the number of anomaly clusters over the total number of clusters. The best value M is chosen as:

$$M = Number\ of\ cluster * MIN(number\ requests\ of\ each\ cluster) \quad (6)$$

From formula (6), can see that the DWAP analysis technique can help extract important features for clustering model to separate anomaly clusters. This is because when reducing M, the ratio of K_2 increases proportionally to the ratio of anomaly clusters.

6. CONCLUSION

Anomaly access to web application detection is a challenging problem to ensure information security. Web application attack techniques always seek to transform and hide themselves to bypass the monitoring of web firewalls. This paper has proposed a method to extract request features using DWAP analysis techniques. Experimental results show that the proposed method has many advantages over traditional approaches. The proposed method works well when it does not try to find anomaly information by comparing the requests among all datasets, but it puts this comparison into local data. Specifically, the comparisons between the values of the requests are made within the same URI. Experimental results not only show the significant improvements in the detection accuracy of the new model, but also show that the DWAP analysis method can be applied in many different areas with its ability to extract correct request's characteristics.

REFERENCES

[1] Symantec, "2019 Internet Security Threat Report," vol. 24, Feb. 2019, [Online], Available: https://www.symantec.com/content/dam/symantec/docs /reports/istr-24-2019-en.pdf.

[2] K. K. Mookhey, "Evasion and Detection of Web Application Attacks," *Blackhat*, 2004, [Online], Available: https://www.blackhat.com/presentations/bh-usa-04/bh-us-04-mookhey/bh-us-04-mookhey-up.ppt.

[3] Nancy Agarwal, Syed Zeeshan Hussain, "A Closer Look at Intrusion Detection System for Web Applications," *Security and Communication Networks*, vol. 2018, no. 2, pp. 1- 27, 2018.

[4] M. H. Kamarudin, C. Maple, T. Watson, and N. S. Safa, "A New Unified Intrusion Anomaly Detection in Identifying Unseen Web Attacks," *Security and Communication Networks*, vol. 2017, no. 1, pp. 1-18, 2017.

[5] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, and T. Berners-Lee, "Hypertext Transfer Protocol-HTTP/1.1," The Internet Society, 1999, [Online], Available: https://tools.ietf.org/html/rfc2616#section-5.

[6] R. A. A. Habeeba, F. Nasaruddina, A. Ganib, I. A. Targio Hashemb, E. Ahmedc, and M. Imrand, "Real-time big data processing for anomaly detection: A Survey," *International Journal of Information Management*, vol. 45, pp. 289-307, 2019.

[7] CSIC-dataset, "HTTP DATASET CSIC 2010," 2010, [Online], Available: http://www.isi.csic.es/dataset.

[8] Trustware SpiderLabs, "ModSecurity: Open Source Web Application Firewall," *ModSecurity*, [Online], Available: https://www.modsecurity.org/.

[9] ModSecurity Core Rule Set Project, "OWASP ModSecurity Core Rule Set," 2016, [Online], Available:https://coreruleset.org/.

[10] B. Cui, S. He, X. Yao, and P. Shi, "Malicious URL detection with feature extraction based on machine learning," *International Journal of High Performance Computing and Networking*, vol. 12, no. 2, pp. 166 - 178, 2018.

[11] M. Zhang, B. Xu, S. Bai, S. Lu, and Z. Lin, "A Deep Learning Method to Detect Web Attacks Using a Specially Designed CNN," *Proceedings of the International Conference on Neural Information Processing*, pp. 828-836, 2017.

[12] J. Zhao, N. Wang, Q. Ma, and Z. Cheng, "Classifying Malicious URLs Using Gated Recurrent Neural Networks," *Proceedings of the International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, pp. 385-394, 2019.

[13] W. C. Yang, W. Zuo, and B. Cui, "Detecting Malicious URLs via a Keyword-based Convolutional Gated-recurrent-unit Neural Network," *IEEE Access*, vol. 7, no. 2019, pp. 29891-29900, 2019.

[14] C. Torrano Gimenez, H. T. Nguyen, G. Alvarez, and K. Franke, "Combining expert knowledge with automatic feature extraction for reliable web attack detection," *Security and Communication Networks*, vol. 8, no. 16, pp. 2750-2767, 2015.

[15] N. v. Ginkel, W. De Groef, F. Massacci, and F. Piessens, "A Server-Side JavaScript Security Architecture for Secure Integration of Third-Party Libraries," *Security and Communication Networks*, vol. 2019, no. 6, pp. 1-21, 2019.

[16] I. Ro, J. S. Han, and E. G. Im, "Detection Method for Distributed Web-Crawlers: A Long-Tail Threshold Model," *Security and Communication Networks*, vol. 2018, no. 4. pp 1-7, 2018.

[17] P. M. D. Nagarjun, and S. S. Ahamad, "ImageSubXSS: an image substitute techniqueto prevent Cross-SiteScripting attacks," *International Journal of Electrical and Computer Engineering*, vol. 9, no. 2, pp. 1393-1398, 2019.

[18] Y. Yang, L. Wang, J. Gu, and Y. Li, "Transparently Capturing Request Execution Path for Anomaly Detection," IEEE Transactions on Parallel and Distributed Systems, pp. 1-13, 2020. [Online]. Available: *arXiv.org, arXiv:2001.07276v1*

[19] J. Ramakrishnan, E. Shaabani, C. Li, and M. A. Sustik, "Anomaly Detection for an E-commerce Pricing System," 2019. [Online]. Available: *arXiv, arXiv:1902.09566v5*

[20] Hu Y., Li B., Ye W., and Yuan G., "A Human-Machine Collaborative Detection Model for Identifying Web Attacks," *Proceedings of the International Conference on Collaborative Computing: Networking, Applications and Worksharing*, pp. 109-119, 2017.

[21] Y. Dong, Y. Zhang, and Hua Ma, et al., "An adaptive system for detecting malicious queries in web attacks," *Science China Information Sciences*, vol. 61, no. 3, pp. 032114:1–032114:16, 2018.

[22] Leo Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5- 32, 2001.

[23] T. G. Dietterich, "Ensemble Methods in Machine Learning," *Proceedings of the International Workshop on Multiple Classifier Systems*, pp 1-15, 2000.

[24] D. Chandler, "Introduction to Modern Statistical Mechanics" *Oxford University Press; 1 edition*, pp. 243, 1987.

[25] R. A. A. Habeeb, F. H. Nasaruddin, A. Gani, and M. A. Amanullah, "Clustering-based real-time anomaly detection-A breakthrough in big data technologies," *Transactions on Emerging Telecommunications Technologies*, vol. 30, no. 6, pp. 36-47, 2019.

[26] A. Smola and S.V.N. Vishwanathan, "Introduction to Machine Learning," *Cambridge University Press*, pp. 234, 2008.