Machine Learning Based Web Application Firewall

Batuhan IŞIKER
Computer Engineering Department
Gebze Technical University
Kocaeli, Turkey
bisiker@gtu.edu.tr
ORCID: 0000-0001-7737-2214

İbrahim SOĞUKPINAR
Computer Engineering Department
Gebze Technical University
Kocaeli, Turkey
ispinar@gtu.edu.tr
ORCID: 0000-0002-0408-0277

2021 2nd International Informatics and Software Engineering Conference (IISEC) | 978-1-6654-0759-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/IISEC54230.2021.9672335

***Abstract*—Internet and computer systems are an indispensable part of daily life. The number of web applications have increased with the development of technology and digital transformation. Web applications have high risk for security since the applications is not developed securely, contains vulnerabilities and easily accessible by hackers. Web application firewall is used to protect web applications from attacks. Signature-based and anomaly-based methods are used in web application firewalls. In this research, anomaly-based web application firewall model was developed using natural language processing techniques and linear support vector machine learning algorithm. Word n-gram and character n-gram natural language processing techniques were compared by performing separate models. Proposed model achieve higher detection performance with using the character n-gram compared to other studies. According to the results of the experiment proposed model is capable of detection web attacks effectively with the overall detection accuracy rate of 99.53%.**

Keywords—web application firewall, machine learning, natural language processing, n-gram, tf-idf

I. Introduction

The intensive use of the internet caused to an increase in the number of web applications. Web technologies are widely used in many security-critical systems such as banking, finance, education, health, electronic government applications. Web applications runs on servers as platform independent. Web applications contain security vulnerabilities because of they are easily accessible via browsers, application developers do not have sufficient security knowledge and secure software development methods are not applied. Information systems and web applications are targeted by hackers to obtain critical information or damage it. Web applications have to be secure to ensure the confidentiality, integrity, and accessibility of information. The security of web applications can be ensured by using secure software development methods and web application firewall. The web application firewall is used as a defensive solution to prevent attacks.

The Open Web Application Security Project (OWASP) is a community that provides information and projects for the security of web applications. [1] Web application vulnerabilities are reported by OWASP based on a certain point system and popularity.

Intrusion detection and prevention systems ensure an important role against attacks by protecting critical information of the system. Since intrusion detection and prevention systems are generally active on the network layer, they are incapable of analyzing application protocols. Web applications use HTTP/HTTPS protocols as application layer. Basically, the difference between web application firewall and intrusion detection system is that web application firewall performs web intrusion detection and prevention by checking OSI application layer protocols. The web application firewall inspects the outgoing and incoming http traffic to the application. To ensure the security of web applications, studies were carried out related to the web application firewall. Web application firewall uses either anomaly-based or signature-based methods. Signature-based method, the examined http traffic is compared with known attacks. Only known attacks can be detected. In the anomaly-based approach, deviations from the normal http traffic are considered abnormal and abnormal http traffic can be detected as an attack. Unknown attacks called zero-days can be detected with this approach. In this research, we study on web application firewall with supervised machine learning method. Our aim was to produce a model with high detection accuracy. We used natural language processing techniques for feature extraction. We experimentally applied our machine learning model on an http dataset. The use of character n-gram and 1 to 6 n-gram range distinguishes from other studies.

In short, the contributions are the following. We propose machine learning based web attack detection method for web application firewall. The main idea is to use the character n gram and tf-idf model to obtain feature representations of HTTP requests. We experimented our proposed feature representation approach with popular machine learning algorithms to prove detect effectively. The proposed model is trained using the inexpensive model to obtain high web attack detection performance. Our model is capable of detection common web attacks.

The remainder of the paper includes web vulnerabilities and web application firewall detection methods in section II, relevant studies on web application firewall in section III, machine learning model and experiment results in section IV, experimental result and evaluation in section V and conclusion in section VI.

II. Web Vulnerabilities and Web Application Firewall

A. Web Vulnerabilities

Web applications are generally divided into three different layers. These are the presentation, application, and data layers. The presentation layer contains the user interface, content is presented to the end user. The application layer contains functional business logic that drives the core capabilities of an application. Java, Python, PHP, .NET etc. programming languages are used as backend technologies. The data layer consists of the database and the data access layer. Security risks arise because of not performing security checks for each different layer.

Web applications are vulnerable to attacks because of easily accessible. Web application vulnerabilities are usually caused by insufficient security checks of inputs from the end user.

978-1-6654-0759-5/21/$31.00 ©2021 IEEE

Different studies have been carried out to reveal the security of web applications. A report was shared by Positive Technologies that includes daily attack amounts for different industry sectors. [12] Figure 1. shows the report.

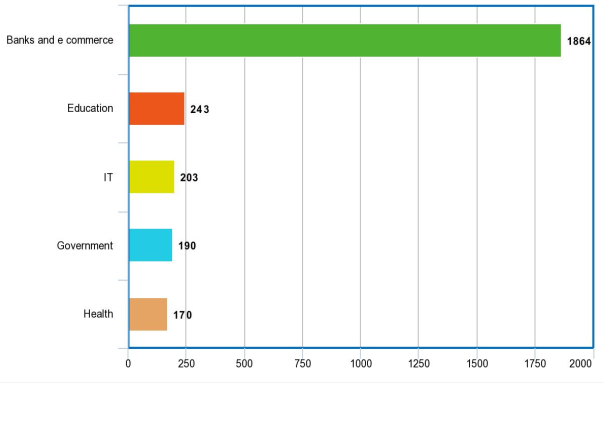

Fig. 1. Number of daily attacks by industry

The Open Web Application Security Project (OWASP) is an open community dedicated to enabling organizations to develop, purchase, and maintain applications and APIs that can be trusted. OWASP Top 10 is a standard awareness document for developers and web application security. OWASP Top 10 2017 application security risks are given as item below.

1) Injection

Injection attacks such as SQL, NoSQL, OS and LDAP occur when untrusted data is sent to the interpreter in part of the command or query.

2) Broken Authentication

Authentication and session management related application functions are often the result of improper implementation. With this vulnerability, passwords, keys, session tokens or credentials of other users can be compromised temporarily or permanently. [5]

3) Sensitive Data Exposure

Web applications do not properly protect sensitive data. Attackers try to get hold of such weakly protected sensitive data to commit credit card fraud, identity theft and other crimes. [5]

4) XML Externak Entities (XXE)

It is a type of attack against parsing XML inputs. External assets can be used to expose internal files using the file URI handler, internal file shares, internal port scanning, remote code execution, and denial of service attacks. [5]

5) Broken Access Control

Authorization checking of authenticated users is often not properly performed. Attackers can exploit this vulnerability to access data or unauthorized functions such as accessing other users' accounts, viewing sensitive files, modifying other users' data, changing access rights.

6) Security Misconfiguration

It is one of the most common vulnerabilities. This is often the result of unsafe default configurations, missing or temporary configurations, misconfigured HTTP headers, and detailed error messages containing sensitive information. [5] All operating systems, applications, and libraries need to be securely configured and updated rather than left with the default configuration.

7) Cross Site Scripting (XSS)

This vulnerability occurs when applications do not validate the data when sending or receiving untrusted data. This allows attackers to execute scripts on the victim's browser that can hijack user sessions, tamper with websites, or redirect the user to malicious sites.

8) Insecure Deserialization

As a result of the lack of access control or other protection measures of the references of internal application objects used in the application, attackers can change their references by accessing data without authorization. This vulnerability could lead to remote code execution.

9) Using Components with Known Vulnerabilities

Components such as libraries, frameworks, and other software modules run with the same privileges as the application. If a vulnerable component is exploited, such an attack could facilitate serious data loss or server takeover. Applications and APIs using components with known vulnerabilities can weaken application defenses and enable various attacks. [5]

10) Insufficent Logging and Monitoring

Combined with incomplete or ineffective integration with incident response, it allows attackers to attack systems further, maintain persistence, break into more systems and access, extract or destroy data. Most attack studies show that the time to detect an attack is more than 200 days, and it is usually detected by external parties rather than internal processes or monitoring.

B. Web Application Firewall

A web application firewall is hardware or software that protects from attacks that exploit vulnerabilities in web applications. [2] It is not a solution that fixes the vulnerability in the web application, it is a security component that can reduce the impact of the attack. There are commercial web application firewalls and open-source web application firewall software. ModSecurity is an open-source web application firewall project. [3] The web application firewall can be deploy on the network or on the web server. [3] The web application firewall deploy on the network can be located at the point between the user and the web server. Network based web application firewall does not depend on the environment of the web server, the number of web servers, it needs to be reconfigured when deploying an existing web application firewall. [4] A web server based web application firewall is provided as software and runs as part of the web server. Its operation depends on the working environment of the web server, it needs to be deployed to all servers that make up the website, it can reduce the performance of the web server when deploying to an existing network, the network does not need to be reconfigured. The reverse proxy method is one of the most common deployment options among web application firewalls. All traffic goes through web application firewall. It forwards the incoming request to the backend web server.

1) Web Application Firewall Methods

There are two different methods in the web application firewall: signature and anomaly-based methods. The signature-based approach looks for signatures of known attacks, known as the negative approach. Traffic outside of attacks that are considered malicious is allowed to pass.

Known attacks can be detected with this method, but unknown zero-day attacks or variants of known attacks cannot be detected. The anomaly-based approach detects behavior that deviates from what is considered normal. Only traffic that is considered normal is allowed to pass, but other traffic blocks. Anomaly method detects unknown zero-day attacks. However, in this approach, the false positive rate is higher than the signature-based method. Resource usage is lower in the anomaly-based approach. The traffic that is considered normal in large and complex web applications is difficult to define. The system working with the anomaly method is scalable, the signature-based method system is not.

Machine learning and statistics-based approaches are generally used in the anomaly method. Http traffic is examined and the map of the application is drawn. Anomalies related to the http protocol and anomalies in the http message are detected. The web application firewall supports two different operating modes. These are detection and prevention modes. In detection mode, attacks are detected and logged, but traffic is allowed to pass. In the prevention mode, the traffic detected as an attack is blocked and the attack is eliminated.

III. Related Works

Researchers studied web application security anomaly detection and machine learning approaches. Pham et al. [6] worked on a web-based intrusion detection system with machine learning. They used different machine learning algorithms such as random forest, logistic regression, decision tree, AdaBoost and SGD. Researchers compared machine learning algorithms using the CSIC 2010 HTTP dataset which includes traffic generated from an e-commerce web application. They used some feature extraction and feature selection techniques to extract more information from text features and improve web intrusion detection performance. Logistic regression performed best results for recall and precision.

Nguyen et al. [7] studied on feature selection related to web application attack detection. For HTTP traffic inspection 30 attributes were determined by taking expert opinions. They propose a framework for using genetic feature selection (GeFS) measurement for web intrusion detection. For intrusion detection they applied the GeFS method together with two measures combined the correlation feature selection (CFS) measure and the minimum redundancy-maximum relevance (mRMR) measure. They applied the GeFS method together with two measures combined the correlation feature selection (CFS) measure and the minimum redundancy-maximum relevance (mRMR) measure for intrusion detection. Genetic feature selection is often used to select feature from high-dimensional datasets such as network traffic or web logs. Correlation feature selection defines the relevance of the features and their relationship in terms of linear correlation, and the minimum redundancy-maximum relevance selects features from datasets with many non-linearly related features. Their results conclude that the correlation feature selection performs well on the CSIC 2010 dataset, while minimal redundancy-maximum relevance performs well on the ECML/PKDD 2007 dataset, which is a collection of real-world web traffic. Torrano et al. [8] have developed a web application firewall that can detect unknown web-based attacks with an anomaly-based detection approach. Their model is detects whether an HTTP request is an attack through the XML file. Figure 2. shows the architectural execution of the study. Attack and normal request database data are created and an xml file that determines the normal behavior of the system is produced from these data. If there is a deviation from the rules set in the XML file, the system will detect it as an attack. The system works in detection and prevention modes. The test was performed with a thousand normal requests and a thousand attacks. When the results are evaluated, the false alarm rate is close to zero and the detection rate is close to 1. As a result, the detection rate is excellent and the false positive rate is high. Consequently, sufficient requests to reveal the characteristic behavior of the web application enable it to reach the successful inspection rate and reduce the false positive rate.

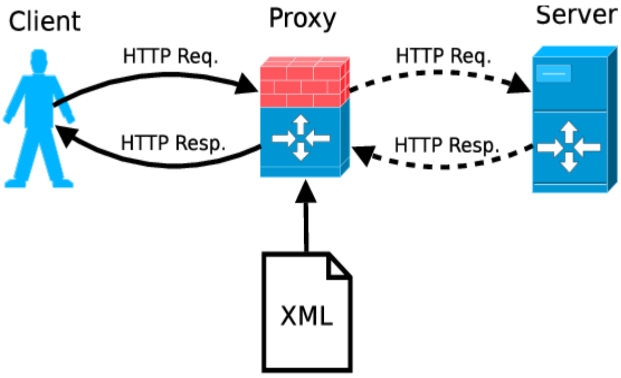

Fig. 2. Web application firewall architecture

Adem et al. [9] proposed a hybrid-based approach for web application firewall that includes signature-based and anomaly-based model. Alphanumeric Character, Letter Frequency and Request Length attributes were determined as attributes and anomaly-based control was performed with Bayesian classification algorithm, one of the data mining methods. Since signature-based inspection is faster than abnormal-based inspection, signature-based inspection database is updated with abnormal HTTP requests detected because of abnormal-based inspection. They tested the model with different data sets and as a result successful anomaly detection rate of 95%.

Xuan [10] proposed the tf-idf (Term Frequency - Inverse Text Frequency) method along with feature extraction, a decision tree-based machine-learning model. Proposed model tested on http parse dataset. As a result, the proposed model was able to detect attack types such as sql injection, cross-site request forgery, command injection, directory roaming with accuracy rate of 98.5%.

Duy et al. [11] compared some feature extraction approaches for the CSIC 2010 dataset. They have created a set of common features that include different parameters such as the length of the http query, the number of parameters, the number of numbers, letters, and special characters of the web page link. As another method, feature extraction operations were applied in the data set with the tf-idf representation method. The performance accuracy values were measured by applying the common feature method and tf-idf methods separately with the random forest machine learning algorithm. TF-IDF (Term Frequency - Inverse Text Frequency) representation method, 4% higher accuracy value was obtained compared to the common feature method. Li et al [14] proposed an anomaly detection over http traffic with weighted word2vec paragraph vectors. They worked with LightGBM and CatBoost algorithms using Word2Vec, gloVe, FastText models on CSIC 2010 dataset. They achieved

99.43% accuracy with the Word2Vec model and the CatBoost algorithm.

IV. Methodology

In this study, machine learning model with different natural language processing techniques were proposed for web application firewall. The proposed model detection accuracy has been increased by using characters n-gram instead of words n-gram. We use CSIC 2010 dataset which is used extensively other studies.

A. Dataset

CSIC 2010 was created by the Information Security Institute of the Spanish National Council for Research. This data set was created by creating automatic traffic on an e-commerce application. The reason why the dataset was created is that the darpa dataset is not up-to-date and does not contain web attack traffics for web application attack detection. The dataset contains 36,000 normal requests and 25,000 abnormal requests. There are static, dynamic and unintentional illegal anomaly requests in the data set. Static attacks try to demand hidden or non-existent resources. Requests are made to default files, such as configuration files. In dynamic attacks, valid request arguments are changed. Attacks such as SQL injection, cross-site request forgery, buffer overflows are in this category. Unintentional illegal requests are requests that are not malicious but do not involve the normal behavior of web applications.

B. Proposed Model

Figure 3 shows the architecture of the model.

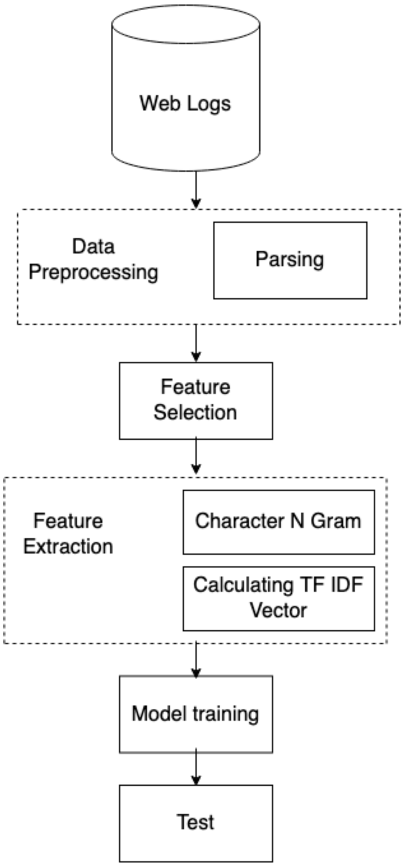

Fig. 3. Proposed machine learning model architecture

Feature selection and feature extraction were applied respectively. Then, the training of the model was carried out with machine learning classification algorithms. The trained model is classified as attack or normal on the test data. In the results section, performance values such as accuracy, F1 measure and complexity matrices of the models used are extracted. The machine learning model with the highest performance value was used in the architecture.

Firstly, http traffics were separated according to http headers in the data preprocessing step on the CSIC 2010 dataset. In the data preprocessing step on the dataset, http traffics were separated according to http headers. In feature selection method name and uri are used as features from these parsed data. Attribute values are grouped using the character n-gram method. N-gram method is preferred in uri feature extraction process because it is simple and fast. Character n-gram technique is used instead of word n-gram. To extract the features, n-grams were selected as 1 to 6 gram and grouped. These extracted features data were also vectorized by calculating the tf-idf value. Data were vectorized using the tf-idf (Term Frequency - Inverse Text Frequency) method. tf-idf is a natural language processing method. It is a numerical statistic that aims to reflect how important a word is to a document in a collection. The more frequently a term is used in the document, the closer to zero the tf-idf value will be.

$$tf(t, d) = \frac{f(t,d)}{\max\{f(w,d): w \in d\}} \tag{1}$$

$$idf(t, D) = log \frac{N}{|\{d \in D : t \in d\}|} \tag{2}$$

$$tf\text{-}idf(t, d, D) = \mathrm{tf}(t, d) \times \mathrm{idf}(t, D) \tag{3}$$

where tf(t, d) is the value where the word occurs in the document is calculated with the formula; idf(t, D) is the idf value is calculated by dividing the total number of documents in the formula by the document in which the word occurs, and taking its logarithm; tf-idf(t, d, D) is the tf-idf value is obtained by multiplying the values obtained in the previous formulas.

Machine learning models were training after data preprocessing and feature extraction. Our machine learning model, different classification algorithms were used and compared. In supervised learning, the outputs of the examples are known. Training data consists of inputs and outputs, a new data is received and the output is estimated. Supervised learning is used in classification and regression problems. In this study, popular classification algorithms were used, the performance values were compared and the algorithm with the highest performance was applied in the model. Machine learning training and testing applications were implemented with the python scikit-learn library.

The dataset is divided into 20% test and 80% training data. Support vector machines, decision trees, random forest, k nearest neighbor, logistic regression algorithms are used. The k nearest neighbor algorithm classifies the data to be classified according to the proximity relationship. By determining a certain k value, the distance of the new data to the data equal to the determined k value is measured and classified according to the relevant group. In the decision trees algorithm, a tree structure is created. Each node represents an attribute, and each branch represents a classification tag. The class label is set after all attributes have been represented. The random forest algorithm is an algorithm that classifies by generating more than one decision tree. It is used in regression and classification problems.

The logistic regression algorithm is extensively used in linear classification problems. Logistic regression is a statistical method. In statistics, the logistic model result can be used for problems that can be binary categorical. Support vector machines are also supervised learning algorithms used for classification and regression problems. Finds a decision boundary between the two classes that are furthest from any point in the training data. After the model is trained with machine learning algorithms, estimation is performed on the unlabeled test data.

V. Experimental Results and Evaluation

A. Experimental Results

In this study, a server with Intel(R) Xeon(R) CPU @ 2.30GHz, 12 GB RAM was used as a processor on Google Colab [13] for the experiment.

Accuracy and F1 measurement metrics were chosen to evaluate and compare the effectiveness of the models. Equation (4) is calculated with the related formula over the error matrix. The precision criterion is the criterion that gives how many actually classified correctly among all samples classified as positive. It is calculated as in equation (5). The recall is the ratio of all positive results were correctly predicted. High sensitivity classifiers are expected to have a small number of incorrectly classified negative samples. It is calculated as in equation (6). F1 measure is a harmonic mean of precision and recall and calculated as in equation (7). The model performance results are as in Table I.

$$Accuracy = \frac{\text{TP} + \text{TN}}{\text{TP+FP+TN+FN}} \quad (4)$$

$$Precision = \frac{\text{TP}}{\text{TP+FP}} \quad (5)$$

$$Recall = \frac{\text{TP}}{\text{TP+FN}} \quad (6)$$

$$F1 = \frac{2*(\text{Precision}*\text{Recall})}{(\text{Precision}+\text{ Recall})} \quad (7)$$

As a result, the highest performance values were gotten with linear support vector machine. The training of the model was carried out with linear support vector machines in the forming model, and then the estimation process was performed on the test data.

TABLE I. Experiment results

Algorithm	*Accuracy (%)*	*F1 (%)*
Linear Support Vector Machine	99.53	99.48
Logistic Regression	99.0	98.84
Random Forest	95.7	95.21
K Nearest Neighbor	99.52	99.48
Decision Tree	96.88	96.64

In addition, experimentally linear support vector machine algorithm and n-gram metrics were chosen by different n-gram intervals as in Table II, and the word and character were tested separately as n-grams and compare results. The character n-gram model achieved better performance results than the word n-gram model. We get the best performance results with the character n-gram and 1 to 6 gram range.

TABLE II. Experiment results of n-gram models

N-gram range	*Accuracy (%)*	
	Character N-gram	*Word N-gram*
2	97.87	95.56
3	98.81	94.46
1 to 3	98.42	95.46
1 to 4	98.42	95.17
1 to 5	99.47	95.61
1 to 6	99.53	95.08

B. Evaluation

The comparison of the results obtained with the existing studies is given in Table III. We get a higher accuracy compared to other studies with our proposed character n-gram model. We observed that successful attack detection can be achieved with the proposed web application firewall model.

TABLE III. Comparison of the studies

Study	*Method*	*Dataset*	*Accuracy Rate (%)*
Adem et al. [9]	Signature based and anomaly based with bayes (hybrid)	CSIC 2010, ECML-PKDD 2007, WUGD 2015	95.0
Xuan [10]	TF-IDF and decision tree	HTTP Param	98.5
Duy et al.[11]	3 word n-gram and random forest	CSIC 2010	99.50
Li et al. [14]	Word2Vec – Cat Boost	CSIC 2010	99.36
Li et al. [14]	GloVe - CatBoost	CSIC 2010	99.19
Proposed model	Character n-gram and linear support vector machines	CSIC 2010	99.53

The limitation of the proposed study is since we use supervised learning method, the model needs labeled data for training and can only detect known attack types in the training data. Another limitation is the character gram model has a longer training time compared to the word gram model.

VI. Conclusion and Future Works

In this study, a machine learning-based model was developed by researching on web application firewall. Two different NLP techniques word gram and character n-gram were compared in web application firewall for web attack detection. According to our experiments, the character n-gram model achieved a higher detection rate than the word n-gram model. We used popular machine learning algorithms for classification in experiments. In our proposed model, we applied feature extraction by grouping with characters n-grams and converting them to vectors with tf-idf. Use of character n-gram instead of word n-gram and the use of n-grams range by selecting 1 to 6 gram with linear support vector machine algorithm are techniques used differently from other studies. According to experiments on the labelled dataset, our model achieves an overall high detection rate of 99.53% and F1 measure is 99.48%. It is capable detect common web attack effectively. Our proposed model gets higher performance results when the compared to the word embedding models such Word2Vec, fastText, gloVe and other studies.

In future studies, we can test the model different data sets, so that different types of attacks can be detected. The more detectable attack types, the more successful the model will emerge. Performance evaluations can also be made using semi-supervised machine learning algorithms. Different natural language processing methods can be used for feature extraction.

References

[1] OWASP, https://owasp.org/

[2] AbuHmed, Tamer, Abedelaziz Mohaisen, and DaeHun Nyang. "A survey on deep packet inspection for intrusion detection systems." arXiv preprint arXiv:0803.0037 (2008).

[3] ModSecurity, https://www.modsecurity.org/

[4] K S. Dharmapurikar, P. Krishnamurthy, T. S. Sproull, and J. W. Lockwood. Deep packet inspection using parallel bloom filters. IEEE Micro, 24(1):52-61, 2004.

[5] OWASP Top 10 Application Security Risks, https://owasp.org/www-project-top-ten/2017/Top_10.html

[6] Pham, Truong Son, Tuan Hao Hoang, and Vu Van Canh. "Machine learning techniques for web intrusion detection—A comparison." 2016 Eighth International Conference on Knowledge and Systems Engineering (KSE). IEEE, 2016.

[7] Nguyen, Hai Thanh, et al. "Application of the generic feature selection measure in detection of web attacks." Computational Intelligence in Security for Information Systems. Springer, Berlin, Heidelberg, 2011. 25-32.

[8] Torrano-Giménez, Camen, Alejandro Perez-Villegas, and Gonzalo Alvarez Maranón. "An anomaly-based approach for intrusion detection in web traffic." (2010).

[9] Tekerek, Adem, Cemal Gemci, and Ömer Faruk Bay. "Web tabanlı saldırı önleme sistemi tasarımı ve gerçekleştirilmesi: yeni bir hibrit model." Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi 31.3 (2016).

[10] Hoang, Xuan Dau. "Detecting Common Web Attacks Based on Machine Learning Using Web Log." *International Conference on Engineering Research and Applications*. Springer, Cham, 2020.

[11] Duy, Pham Hoang, Nguyen Thi Thanh Thuy, and Nguyen Ngoc Diep. "Anomaly detection system of web access using user behavior features." Southeast Asian Journal of Sciences 7.2 (2019): 115-132.

[12] Positive Technologies, Web application attack statistics: Q4 2017,https://www.ptsecurity.com/ww-en/analytics/webapp-vulnerabilities-2017-q4/

[13] Google Colab, https://colab.research.google.com/

[14] Li, Jieling, Hao Zhang, and Zhiqiang Wei. "The weighted word2vec paragraph vectors for anomaly detection over HTTP traffic." IEEE Access 8 (2020): 141787-141798.