DESIGN AND IMPLEMENTATION OF AN ARTIFICIAL INTELLIGENCE-BASED WEB APPLICATION FIREWALL MODEL

Article *in* Neural Network World · January 2019

DOI: 10.14311/NNW.2019.29.013

2 authors:

Adem Tekerek
Gazi University
39 PUBLICATIONS **401** CITATIONS
SEE PROFILE

Omer Faruk Bay
Gazi University
73 PUBLICATIONS **603** CITATIONS
SEE PROFILE

DESIGN AND IMPLEMENTATION OF AN ARTIFICIAL INTELLIGENCE-BASED WEB APPLICATION FIREWALL MODEL

A. Tekerek[*], *O.F. Bay*[†]

Abstract: Attacks on web applications and web-based services were conducted using Hyper-Text Transfer Protocol (HTTP), which is also used as the communication protocol of web-based applications. Due to the dynamic structure of web applications and the fact that they have many variables, detection and prevention of web-based attacks are made more difficult. In this study, a hybrid learning-based web application firewall (WAF) model is proposed to prevent web-based attacks, by using signature-based detection (SBD) and anomaly-based detection (ABD). Detection of known web-based attacks is done by using SBD, while detection of anomaly HTTP requests is done by using ABD. Learning-based ABD is implemented by using Artificial Neural Networks (ANN). Thus, an adaptation of the model against zero-day attacks is ensured by learning-based ABD by using ANN. The proposed model is tested by using WAF 2015, CSIC 2010 and ECML-PKDD datasets which are open source datasets. According to the test results, a high mean achievement percentage (96.59 %) was obtained. Detection results are also compared to previous studies. After comparison, the proposed model promises higher performance than what the existing studies until now have to offer.

Key words: *anomaly-based detection, artificial neural networks, hybrid web application firewall, web-based attacks, web application security*

Received: July 13, 2018 **DOI:** 10.14311/NNW.2019.29.013
Revised and accepted: August 19, 2019

1. Introduction

Services provided on the Internet cover a large area in all sectors. As the complexity, size and number of the services provided by the Internet increase, there will be a gradual increase of attacks through web-based applications. Thus, the security of web applications should be ensured to prevent data losses and vulnerability of the Internet, which is at present extremely insecure [1]. In order to ensure the security of web applications, the attacks to be carried out on the application can be eliminated using secure coding methods during the design pattern of the web application

[*]Adem Tekerek – Corresponding author; Information Technology Department, Gazi University, Ankara, Turkey, E-mail: atekerek@gazi.edu.tr

[†]Omer Faruk Bay; Electric Electronic Engineering Department, Technology Faculty, Gazi University, Ankara, Turkey, E-mail: omerbay@gazi.edu.tr

software. Some application developers have inadequate knowledge concerning secure coding standards and they give more priority to application functionality than to security requirements [1]. Furthermore, they often do not have sufficient knowledge with regard to security requirements and web applications, which carry high risks of security weaknesses [2]. While traditional firewalls successfully block packets coming from the network layer, they may not be effective enough for web-based attacks coming from HTTP to the web applications [3]. Web application security requirements can be satisfied at the network layer by means of traditional firewalls and intrusion detection systems. The attacks targeting the web applications do not only use the network layer but also HTTP. Yet the tools used to ensure security in the network layer may not be effective enough at HTTP level. So, web-based attacks can be blocked by detection of HTTP requests. New web-based attack techniques threatening web applications are being developed day after day. Web Application Firewall (WAF) development studies are currently conducted to ensure the security of web applications and prevent web-based attacks targeting web applications in a growing number and variety. Both academia and the information technology sector have been studying web application security to produce solutions to prevent attacks. However, in order to develop a reliable method, further studies are needed. WAF capabilities should be improved to decrease web application vulnerabilities. The purpose of this study was development of learning-based a hybrid WAF model to prevent web-based attacks. This model was developed using Signature-Based Detection (SBD), Anomaly-Based Detection (ABD) and Artificial Neural Networks (ANN) as one of the artificial intelligence techniques. By using SBD, the detections were made against known web-based attack types such as SQL (Structured Query Language) Injection, Cross-Site Scripting (XSS), Command Injection, and Directory Traversal Attacks. HTTP requests that do not conform to the structure of the web application architecture were detected using ABD. HTTP requests are digitized using three features: Alphanumerical Character Analysis, Letter Frequency Analysis, and Request Length Analysis. Using the results obtained from these features, the learning-based ABD was implemented by using ANN. These features were used as ANN input data.

The rest of this paper is organized as follows: The second section explores some study findings regarding the prevention of web-based attacks under two subheadings, the vulnerability of web applications and the Web Application Firewall (WAF). The third section presents the proposed WAF model by means of a detailed explaination of feature selection and datasets, signature based detection, anomaly based detection and Artifical Neural Networks (ANN). The fourth section reports the results of the present study. The fifth section is the conclusion.

2. Literature review

In this section, in order to support the purpose of this study and to present why our proposed model is superior to existing models, some findings of previous studies discussed.

Stephan et al. (2015) [4] developed a prototypic WAF to detect zero-day web-based attacks which did not require signature updates. They used a neural network back-propagation approach for identifying attacks that are not detected at the stage of signature analysis. The proposed study was tested on a set of parameters and some additional information about user behaviours on web application. The system was founded to obtain a good performance in comparing and matching the test patterns with the existing patterns therefore the test data has been correctly identified by a 95 % detection rate success. In another study, proposed by Nguyen et al. (2013) [5] a web attack detection model was proposed using Generic Feature Selection (GeFS). For GeFS, CSIC 2010 and ECVL/PKDD 2007, datasets were used, and the detection was accomplished by generating 30 different features that were determined by expert opinions in datasets with 4 different sorters consisting of C4.5, CART, RandomTree, and RandomForest. Sorting was made in this study for Cross-Site Scripting (XSS), SQL injection, LDAP Injection and XPATH Injection attack types. Pałka and Zachara (2011) [6] discussed a selection of issues related to the implementation and deployment of the WAF protecting the target application by verifying the incoming requests and their parameters by way of matching them to recorded usage patterns. These patterns, in turn, were learned from the traffic generated by users of the application. A learning-based WAF offered a flexible, application-tailored yet easy to deploy solution. There were certain concerns, however, regarding the classification of the data that was used for the learning process which could, in certain cases, impair the firewall ability to classify traffic correctly. Basile and Lioy (2015) [7] have developed a model for the analysis of packet filters to the policy anomaly analysis in application firewalls. Both rule-pair and multiple anomalies were detected, in order to see reducing the likelihood of conflicting and suboptimal configurations. The expressiveness of this model was successfully tested against the features of Squid, a popular Web caching proxy offering various access control capabilities. The tool implementing this model was tested on the basis of various scenarios, and it exhibited good performance. In another study, Torrano-Gimenez et al. (2009) [8] developed a system that followed the anomaly approach. It could detect both known and anomaly web-based attacks. The system decides whether the incoming requests are attacks or not by an XML signature list. Any request which deviates from the normal behavior is considered an anomaly. The system has been applied to protect a real web application. An increasing number of training requests have been used to train the system. Experiments showed that when the XML file has enough data to closely characterize the normal behavior of the target web application, a very high detection rate is reached while the false alarm rate remains very low. Kruegel and Vigna (2003) [9] proposed a character distribution method for web attack detection. They explained that normal web access data would possess a normal character distribution, and that anomaly data, on the contrary, would have a different character distribution. In a study conducted by Singh et al. (2011) [10], an iSVM-Improved Support Vector Machine algorithm developed for sorting cyber-attack data sets was proposed. Results exhibited a 100 % detection and false alarm accuracy against iSVM, Normal and Denial of Service (DoS) types as well as comparability of training and test durations. Traditional SVM performance was enhanced to enlarge spatial resolution around a conformal mapping margin with Gauss Kernel. Thus, the divisibleness between the attack

types will increase. This procedure is based upon the Riemann geometrical structure stimulated by the kernel function. In our previous study (Tekerek et. al., 2016) [11] a hybrid WAF model was proposed which used the SBD and ABD together to prevent web-based attacks. Different from that study, we preferred to use ANN for ABD instead of Bayesian classification. The ANN-based proposed model presented better results than the Bayesian classification model.

All these studies have been conducted in order to prevent web-based attacks. Unlike previous findings, in our study, a hybrid learning-based WAF model which uses SBD and ABD methods together in order to prevent web-based attacks is proposed. While the SBD method is effective against known web-based attacks and runs faster than ABD, the ABD method is effective against zero-day attacks and is able to self renew itself for new types of attack patterns. The disadvantage of ABD is that it is slow in running to identify attack patterns. Thus, methods like ABD are not preferred at real-time intrusion detection systems. In this study, the proposed algoritm uses the SBD method which is effective against known web-based attack and runs faster as well together with the ABD method, renewing itself for new types of attack patterns, for its effectiveness against zero-day attacks.With the SBD, detections have been made against known web-based attack types such as SQL Injection, Cross-Site Script (XSS) coding, Command Injection, and Directory Traversal Attacks. Through the ABD, anomaly HTTP requests have been detected with three selected features which are using ANN on a learning basis. While SBD method works quite fast, it does not work efficiently against zero-day attacks. The ABD method, however, is effective against zero-day attacks. Therefore, disadvantages of SBD and ABD methods have been eliminated by using both methods simultaneously. The Signature-Based Detection of Anomaly Requests (SBDAR) checklist has been updated with HTTP requests that were previously identified as an anomaly. When HTTP requests that were previously identified as an anomaly reach to a web application a second time, it can be blocked by SBDAR without using ABD. The other distinctive characteristic of the proposed WAF model is its ability to determine the normal requests that reach to the web applications. We also updated Signature-Based Detection of Normal Requests (SBDNR) with HTTP requests that were previously identified as normal HTTP request. With this new proposed model, clients who requested normal HTTP requests will easily access a web application without the interference of ABD. This feature will increase the speed performance of the proposed model.

2.1 Web applications and vulnerabilities

Due to the fact that web applications are easily accessible and their vulnerability level are extremely high, data security vulnerability occurs within web applications more commonly than in other internet-based applications. Web applications development and working principles have different properties, and they have static or dynamic structure containing Hypertext Mark-up Language (HTML). Static web applications contain HTML scripts only, allowing related web pages to be displayed upon requests from the client. On the other hand, dynamic web applications are running in line with user requests which are developed by using server-side web programming languages. Dynamic web applications work in a 3-layer structure, as shown in Fig. 1 [12].

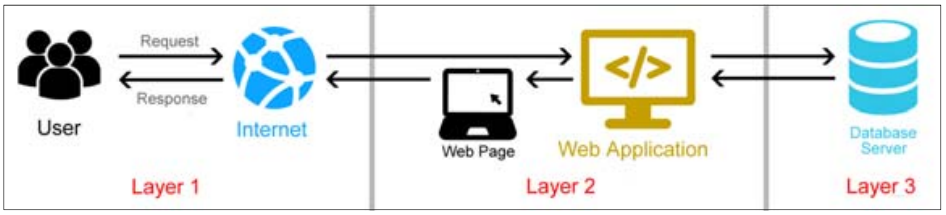

Fig. 1 *Schematic display of a dynamic web applications working structure*

As presented in Fig. 1, Layer 1 represents the web browser where requests for a web application begins. Users send their requests to the web servers using web browsers. Layer 2 is where dynamic pages are generated. Layer 3 represents the database where data used by the web applications are stored. 70 % of the attacks carried out on web applications originate from attacks in the application layer and 75 % of web applications with commercial content is unprotected [13].

2.2 Web Application Firewall (WAF)

WAF is a firewall type that filters, monitors, and blocks HTTP traffic on the application layer of a web application server. It is used as a software or hardware with the ability to listen to traffic between web server and the web browser. It detects web-based attacks according to the defined policies. At the same time, it is a kind of intrusion detection system but it does not operate on the network level of a web application; instead, it is exclusively applied on the application layer [14]. WAF can work with any web traffic filtering mechanism (positive, negative, or session). It works either as an embedded firewall with the web server or as a separated layer of security in a reverse proxy form [15]. WAF is used to detect HTTP requests. There are two kinds of detection methods; blacklist and whitelist. Blacklist is a list of known attack types. When any HTTP request listed in the blacklist is detected, the blacklist blocks the request. Most of the WAF tools have blacklisting features. Conversely, the whitelist is a method used to determine unknown requests [16]. Blacklist model grants permission for anything except for specifically blocked processes. Blacklist security model detects known anomaly requests, generic behaviors, and generic web application attacks. It grades each and every anomaly HTTP request and keeps IP addresses, application cookies and user accounts [17]. Whitelist model blocks access to anything except for specifically permitted processes. It stipulates the determination of permitted requests only and blocks all others. Generally, learning-based algorithms are used in the whitelist model. HTTP requests are examined and a map of web applications is generated. Based on the principle of determining the behavior of the incoming request, similar new attacks that may be carried out on a web application can be deactivated after this process. All types of web-based attacks, to be detected should be defined on WAF with no learning capability [17].

3. Proposed Web Application Firewall model

Proposed WAF model in this study is a hybrid system which is developed by using SBD and ABD methodologies simultaneously. The proposed model flow diagram is presented in Fig. 2. The detection process consists of SBD and ABD methods. The SBD method consists of three parts: Signature-Based Detection of Normal Requests (SBDNR), Signature-Based Detection of Known Intrusion Types (SBDKIT), and Signature-Based Detection of Anomaly Requests (SBDAR). For ABD, Alphanumeric Character, Letter Frequency and Request Length features are used as ANN input features. By the results obtained from the detections, anomaly HTTP requests and normal HTTP requests are determined through ANN. As a result of these detections, while SBDAR database is updated with anomaly HTTP requests, the SBDNR database is updated with normal HTTP requests. This section consists of three parts. In the first part, we explained why we chose the three features for ABD. In the second part, we will give detailed information about SBD. In the final part, the ABD which also includes ANN is explained.

3.1 Feature selection and datasets

To evaluate intrusion prevention and detection systems, several datasets containing data believed to be normal when compared to those containing attacks are used. However, there are not enough datasets, which are commonly usable and contain HTTP traffic. Furthermore, most of the existing datasets do not contain HTTP request and response data required [18]. In this study, CSIC 2010, ECML-PKDD 2007 and WAF 2015 datasets, which were previously generated by the authors of this paper [11], are used. ECML-PKDD 2007 dataset contains HTTP requests, attacks and normal requests including 5,000 samples, of which 20 % are attack requests. CSIC 2010 is generated for the detection of web-based attacks which consist of 36,000 normal and 25,000 anomaly HTTP requests. In addition, the WAF 2015 dataset, where approximately 88 % of the requests are anomaly containing SQL Injection, Memory Overflow and XSS Injection types have been used. With the proposed study, the HTTP request detection is ensured. The structure of HTTP requests is based on a string. Thus, we need to select features which fit to the string-based detection. As a result, we selected 4 features from other existing studies. The number of features that are used to implement ABD directly affects the speed of the ABD. Thus, if ABD with lesser features is used, a higher speed for the detection can be implemented. Through the ABD, we have not detected any known web based-based intrusion type, but we implemented ABD for HTTP requests that do not conform to the request structure of the web application. In order to carry out a detection on HTTP requests that do not conform to the normal HTTP requests, four features have been determined as Request Letter Frequency, Sentence (Request) Length, Alphanumerical Character and Number of Requests. The selection of features has been determined as per detection performance values of each feature individually. Anomaly detection values of features defined in Tab. I are compared between WAF 2015, CSIC 2010 and ECML-PKDD 2007. The datasets are given in percentages (%). As shown in Tab. I, the average value of the Number of Requests feature's detection performance is too low. Thus, the Number of Requests feature is not selected for the anomaly detection.

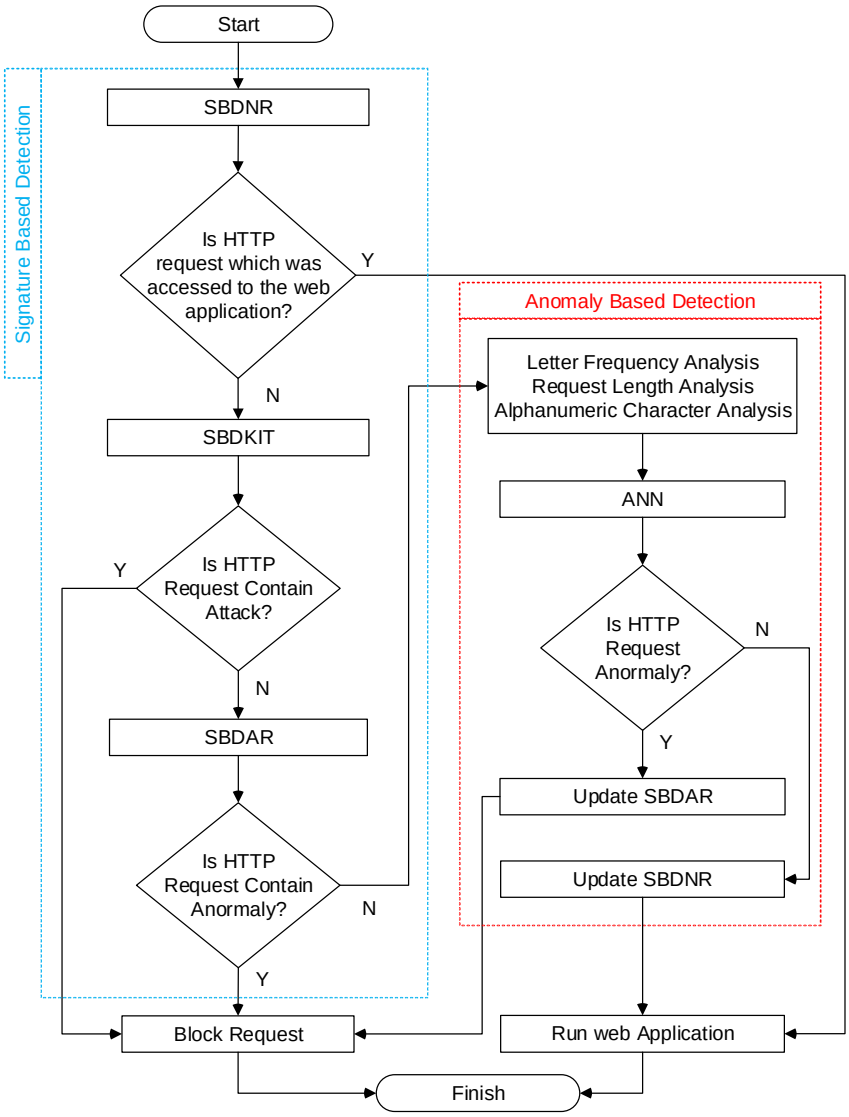

Fig. 2 *Proposed model flow diagram.*

3.2 Signature-Based Detection (SBD)

SBD is a rule-based model, defined as misuse. It is revealed by examining the characteristics, behaviors, and content of the attack [19]. Signature-based methods generally work fast and are effective against attack types listed in the signature database [20]. In general, intrusion detection systems and anti-virus softwares work on the basis of a signature database [5]. When a new attack technique is developed, the signature database needs to be updated in order to set the system to be efficient.

	Request Length [%]	Request Letter Frequency [%]	Alphanumerical Character [%]	Number of Requests [%]
WAF 2015	23	33	39	6
CSIC 2010	49	21	20	11
ECML-PKDD 2007	43	10	45	3
Average	38	21	34	6

Tab. I *Feature selection values for different datasets.*

Otherwise, it would be ineffective against any new type of attack. Through the our model, signature-based detection is accomplished on SQL Injection, Cross-Site Script (XSS) coding, Directory Traversal Attacks, and Command Injection.

3.2.1 Signature-Based Detection of Normal Requests (SBDNR)

SBDNR is a signature-based detection stage where HTTP requests directed at web applications are normally detected through ABD. Normally, the detected HTTP requests are added to the normal signature lists so when they appear in the web application again, they are directed to the web application without running the anomaly detection process. As such, an increased speed of its performance can be ensured.

3.2.2 Signature-Based Detection of Known Intrusion Types (SBDKIT)

Known attacks are the web-based attack types that target web applications revealing their weaknesses by using security vulnerabilities of web applications and containing intrusion qualities. In this study, the most common web-based attack types are used as an example. These are: SQL Infection, Cross-Site Scripting (XXS), Directory Traversal Attack, Command Infection and Signature-Based Detection of Anomaly Requests (SBDAR).

SQL Injection: SQL is a structural language specialized in querying within database management systems. Relational database applications in various sizes are reached through SQL queries. Many database management systems that support SQL apply special add-ons in the standard language [21]. Web applications may be used to generate different SQL sentences for user originated inputs and requests. The SQL injection method is a penetration test. It allows for misuse of SQL sentences using vulnerabilities that arise from user inputs not being verified or being verified inadequately [22]. If the web application does not efficiently detect user requests, the structure of the SQL sentences inside the web application could be changed; a database management system penetrated and privileges of the web application's administrator could be captured through the SQL Injection method. When performing a SQL injection, characters containing special meanings unique to SQL need to be used.

Cross-Site Scripting (XSS): It is generally defined as running desired client-originated scripts in users browser by means of adding client-originated scripts inside HTML codes. XSS are generally written in HTML/JavaScript languages;

however, scripting can also be done in VBScript, ActiveX, Java, Flash or other languages that are supported by web applications [23]. When the malicious script is running on the web browser of the user through the XSS method, the malicious script would run on the security configuration of the website as defined for the browser. If no restrictions have been applied on the web browser, all sensitive information reached by the malicious script through the browser can be read, modified or sent somewhere else via email.

Directory Traversal Attack: Directory traverse occurs when file names and file paths are misused that contain inadequate security verification provided by the user. The purpose of this attack is to authorize an application to have access to files that are meant to be not accessed. This intrusion type misuses the security weakness rather than misusing any mistake in software scripts, even though the software runs as requested. Directory traversal is also known as directory climbing or backtracking. It may supply sequences of "../" to break out of the server's directory root [24].

Command Injection: A command injection attack is an intrusion type aimed at running scripts on the operating system containing the application through an unprotected application. These attacks are only possible when an application runs on system shell user-originated data that are unsafe (forms, cookies, HTTP Headers etc). Command injection attacks are generally possible due to inadequate access verification. Since the command injection method lets attackers add their own script that will be run by the application afterwards, this intrusion method differs significantly from a command injection. In script injection, the attacker expands the default functions of the application without having to run system commands. Command injection attacks are more destructive when scripts are created, successfully parsed [25].

Signature-Based Detection of Anomaly Requests (SBDAR): SBDAR is a signature-based detection process accomplished to block HTTP requests detected as an anomaly through the ABD process of HTTP requests. When HTTP requests are detected as an anomaly and then re-enter the web application, they will be blocked via SBDAR without ABD processes.

3.3 Anomaly-Based Detection (ABD)

Anomaly HTTP requests behave differently than normal HTTP data. In order to carry out detection of HTTP requests which do not conform to the normal HTTP request structure, three features have been selected. According to the results of the experiment conducted with WAF 2015, CSIC 2010 and ECML-PKDD 2007 data sets, Letter Frequency Analysis, Request Length Analysis and Alphanumerical Character Analysis features are selected.

Alphanumerical Character Analysis: The term alphanumeric is used to define a character sequence of letters and numerals (A-Z, a-z, 0-9) in the Latin alphabet. Similarly, each member of this sequence is defined as Alphanumeric. It is a cluster of definitions generated to allow for optimum memory use during data storage of computers. Generally, in one byte, the ASCII correspondence of the alphanumerical value is kept. HTTP Requests coming to the web application have a specific character sequence. This specific character sequence can be analyzed by using the

Eq. 1. According to the HTTP request structure, we can compute the specific character sequence which is found in defined global cluster (e) and generated by using equation (1). The equation (1) which is used for alphanumeric character analysis is presented. In equation (1), "|" means conditional probability, which suggests that the total alphanumerical character value is determined, based on the specific character sequence which is also a condition in the global cluster (e).

$$a = \sum_{i}^{n} r_i \in e \,|\, (a+1), \tag{1}$$

where a = Total of Alphanumerical Character Values, r = HTTP Request, n = Total Number of Characters Forming Request, e = Global Cluster.

Request Length Analysis: Requests coming to web applications have a specific request structure depending on the developing structure of the web application. One of the request structure feature is the request length. Request length values of memory overflow and cross-site scripting attacks are different to normal requests. Following [9], average length and variance values of the requests are used. For Request Length Analysis, Eq. 2 is used. The Eq. 2 is given for computing the probability of an HTTP request whether it is an anomaly or not. However, we could not decide whether a HTTP request is an anomaly or not by only using the Eq. 2. Eq. 2 produces one of the parameters that is used to digitize a HTTP request for ANN input data.

$$\mathrm{P} = \frac{\sigma^2}{(l-\mu)^2}, \tag{2}$$

where P = Probability of Length, μ = Average (Average Value of Requests), σ^2 = Variance (Average Variance Value of Requests), l = Length (Length Value of Request Checked).

According to Eq. 2, the request length value of zero (0) determines the anomaly limit value. If the value of each request is calculated, as per the probability equation of requests for being an anomaly is smaller than the anomaly value of the request of which the length value is zero (0), the request in question is supposed to be an anomaly. Therefore, as the length of the HTTP request increases, the probability of the request being anomaly increases.

Request Frequency Analysis: Character (letters) frequency values of requests are calculated through a character distribution model. Letter frequency values of the characters forming the normal HTTP requests coming to web applications is higher compared to the letter frequency value of anomaly requests. ASCII characters are used in character distribution. ASCII characters are printable letters and numeric characters having an 8-bit value between 0 and 255. For request frequency analysis, Eq. 3 is used. The Eq. 3 which is used for request frequency analysis is presented below. In Eq. 3, "|" means conditionally implying that the total frequency value is determined, based on the specific character sequence which is also a condition in ASCII.

$$t = \sum_{i}^{n} r_i \in \mathrm{ASCII} \,|\, t + f(r_i), \tag{3}$$

where t = Total Frequency Value, r = HTTP Request, n = Total Number of ASCII Characters Forming HTTP Request, f = Total of Request Letter Frequency.

Letter frequency analysis is a technique mainly used in crypto-analysis methods. In this study, letter frequency analysis is performed to determine the total number and average value of each character in the incoming HTTP requests. For instance, when a request such as index.php?id=4&kid=12 is taken into account, frequency and the average value of the letters forming this sentence are obtained. While frequency values give the number of each character in all incoming requests, the average value is calculated by dividing the total value of each character by the number of incoming calls.

3.3.1 Artificial Neural Networks (ANN)

To implement learning-based ABD, the best ANN learning model is produced by using HTTP datasets. Firstly, ANN input and output data were normalized by using min-max method linearly between 0-1. The purpose of ANN training is to reach minimum error rate, by using the lowest number of hidden layers and neurons. In the training of ANN weights, a feedforward backpropagation algorithm is selected. The most important process at this stage is the process of determining the number of hidden layers, the number of neurons in the hidden layers, and the activation function. Hyperbolic tangent is used as the activation function. The hidden layer and neuron numbers are determined experimentally to obtain the best ANN result. The number of layers can be increased according to the difficulty of the problem, but if the number of layers is much more than required, it causes an increase of process time and memorization of the network. To generate the best ANN model, Mean Squared Error (MSE) value is selected as the lowest possible (close to 0) and the value of regression is selected as the highest (close to 1). In order to train the network, 80 % of all datasets were determined as training data, and 20 % of the dataset as test data. The network has one hidden layer and ten neurons in that hidden layer.

The structure of the neural network model is shown in Fig. 3. The ANN model consists of three inputs, one hidden layer with ten neurons and one output layer. The "W" given in Fig. 3 represents the weight values multiplied by the input values, and "B" represents the bias value used to improve the network training rate. The representation of the developed ANN model on the basis of nodes is given in Fig. 3. The most important stage of ANN training is the determination of the weights between the input layer, the hidden layer, and the output layer. In the proposed ANN model, there are three inputs, one output, one hidden layer with ten hidden layer neurons. Input feature layers request letter frequency, length, and alphanumeric character analysis. Fig. 3 is the topological structure of the network. In addition, a bias value that contributes to each hidden layer neuron and output layer neuron is added.

Neural network weight vectors are initially generated randomly at infinite values between -10 and $+10$. The total number of weights according to the parts of the neural network are given in the Eq. 4. Training Regression and Test Regression performance values in 3-10-1 hidden layer numbers have come up more successfully when compared to other layer numbers. Thus, when the ANN model is being developed, a 3-10-1 network structure has been selected. The network designed

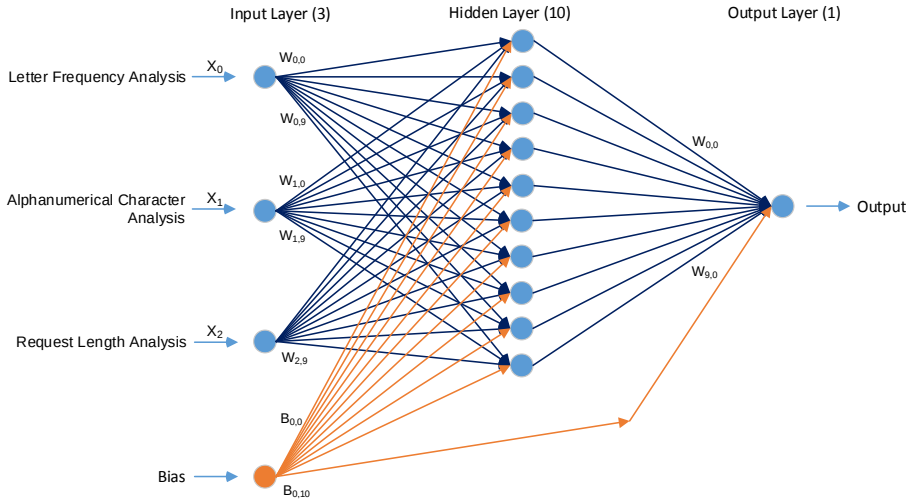

Fig. 3 *Proposed artificial neural network model (3-10-1).*

in the 3-10-1 architecture has a total of 51 weight values. The weight number is calculated as follows:

$$\begin{aligned} & \text{Total Number of Weights} = \\ = \ & (\text{Number of Input} \times \text{Number of Hidden Layer Neurons}) + \\ + \ & (\text{Number of Hidden Layer Neurons} \times \text{Number of Outputs}) + \\ + \ & (\text{Number of Bias} \times \text{Number of Hidden Layer Neurons}) + \\ + \ & (\text{Number of Outputs}), \end{aligned} \tag{4}$$

$$\text{Total Weight} = (3{\times}10) + (10 \times 1) + (1 \times 10) + (1 \times 1) = 30 + 10 + 10 + 1 = 51.$$

The ANN model is developed to determine the learning method for the purpose of determining anomaly or normality of the HTTP requests within the web application. The WAF software is developed on the .NET Framework platform using Microsoft Visual Studio 2015 toolkit, with the use of the C# ASP.NET programming language. The feedforward backpropagation algorithm is used in the training of the specified weights. The recursive algorithm is the most appropriate shape insertion technique designed to minimize a target function. The most commonly used target function is the MSE.

The feedforward backpropagation algorithm may encounter error values that can cause the local minimum problem at a distance from the randomly selected initial value to the global minimum. Fig. 4 shows local minimums. According to the selected performance values for network training, the global minimum value is determined at 0.1002138106.

The MSE graph of the proposed ANN model is presented in Fig. 5. The MSE is calculated every 100 iterations, and is repeated for each training. It is calculated every 100 iterations to increase the speed of training.

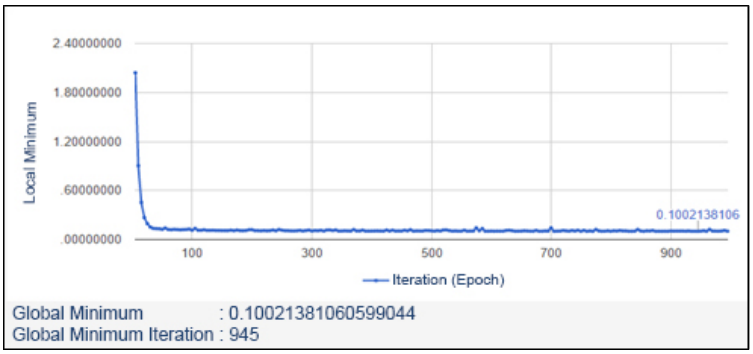

Fig. 4 *Local minimum values.*

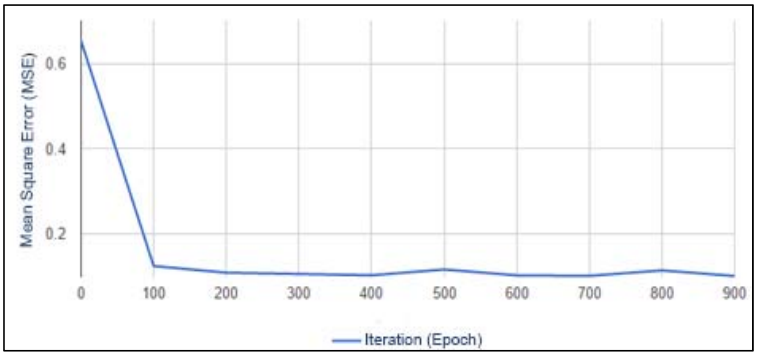

Fig. 5 *ANN, MSE Graph.*

4. Results and discussion

In this section, evaluation of the proposed WAF model is presented to determine detection results of the proposed model to select the most appropriate configuration features. CSIC 2010, ECML-PKDD 2007 and WAF 2015 datasets are used to evaluate the proposed model. In order to detect the anomaly and normal HTTP requests, they are detected by normalizing between zero (0) and one (1), using ANN results. Some of the estimation results that real output values have obtained are shown in Fig. 6. For the ANN inputs, request length, letter frequency, and alphanumeric character features are used. The output value of the ANN model according to the input values is shown in Fig. 6 according to the developed WAF model, while results that are close to one (1) refer to anomaly HTTP requests, results that are close to zero (0) refer to normal HTTP requests, as a result of ABD. While HTTP request results between 0 and 0.4 are generally accepted as normal requests, results between 0.6 and 1 are accepted as an anomaly. The results, that range between 0.4 and 0.6, were fed back to the users to verify for classification as either an anomaly or a normal HTTP request. The limits were determined on the basis of the results after testing and validation of ANN.

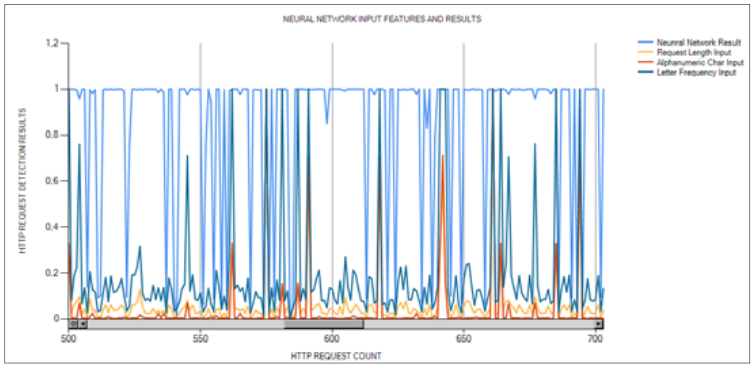

Fig. 6 *Input features and ANN results.*

The detection duration of the proposed model's detection phases is shown in Fig. 7 differences are demonstrated between three-stage SBD time and ABD time as performed by the ANN-based learning method. Detection times of stages in the SBD are identified as 2.7 ms, 2.4 ms and 3.2 ms for SBDNR, SBDAR and SBDKIT, respectively. Detection time of the ABD is identified as 12.3 ms. Because the ABD detection time is higher than the SBD detection time, considering the system's speed performance, SBD is implemented before ABD. The web application is executed directly by performing SBDNR, when the normal HTTP requests that are detected by the ABD results approach a web application for a second time. Anomaly HTTP requests that are detected by the ABD are then blocked by performing SBDAR before reaching the ABD stage when they come to the web application again. Thus, by proposing a hybrid detection model, the speed performance of the model is increased by eliminating the slow working feature of anomaly-based detection.

Fig. 7 *Comparison time performance of detection stages.*

The proposed model is an approach for a WAF algorithm combined with ANN. It is intended for the prevention of web-based attacks. The model test results were shown in Fig. 8 with different datasets. The test results as shown in Fig. 8 were obtained from WAF 2015, CSIC 2010 and ECML-PKDD 2007 datasets by using the WAF software. According to the test results, 98.52 % of the HTTP requests that contain attacks were detected in the WAF 2015 dataset. 96.74 % of the HTTP requests that contain attacks were detected in the CSIC 2010 dataset. 94.53 % of the HTTP requests that contain attacks were detected in the ECML-PKDD 2007 dataset.

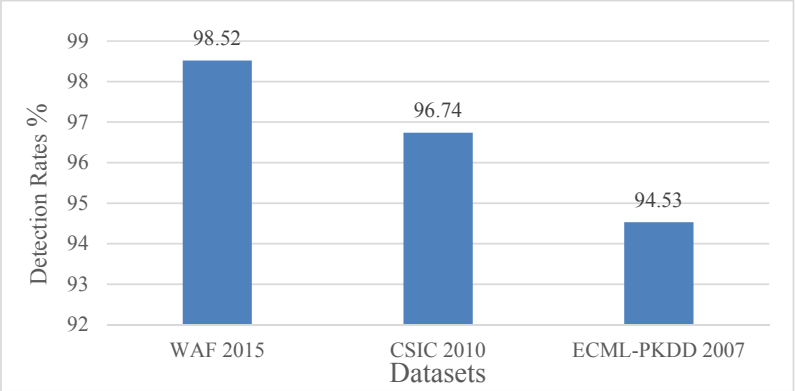

Fig. 8 *Success rates of test results for datasets.*

Similar studies were proposed to prevent web-based attacks. WAF 2015, CSIC 2010 and ECML-PKDD 2007 datasets detection results are presented in Tab. II by using the proposed algorithm. The main value for using these studies from a comparative point of view is the use of similar datasets in the detection process. The most important distinctive property of the suggested model is that it is a hybrid model. The main reason for proposing a hybrid model is to decrease the detection duration and to eliminate disadvantages of the previously used methods, which are SBD and ABD. According to the detection results, 98.52 % of the HTTP requests in the WAF 2015 dataset, 96.74 % of the HTTP requests in the CSIC 2010 dataset, and 94.53 % of the HTTP requests in the ECML-PKDD 2007 dataset are obtained in our proposed model.

According to the results of other, comparable studies, it is observed that the hybrid method proposed in this study gives better results. The only use of rule-based methods are not enough to detect attacks on web applications. Anomaly detection is performed to prevent zero-day attacks using artificial intelligence methods for web-based attack detection. Both the rule-based and the anomaly-based methods are performed in the proposed hybrid model to detect malicious HTTP requests. The other motivation for better detection result is the selection of features for ABD. Features were used to digitize the HTTP requests. Digitized features were then used as input values of ANN. Input values of ANN effect the training success of ANN. As such, more successful results are obtained. The other advantage of

Paper	SBD	ABD	Detection Method	Dataset	[%]
(Stephan et al. 2015)[4]	×	√	YSA	HTTP Dataset	95.00
(Nguyen et al. 2013)[5]	×	√	C4.5	CSIC 2010	94.49
				ECML-PKDD 2007	96.30
			CART	CSIC 2010	94.12
				ECML-PKDD 2007	96.11
			Random Tree	CSIC 2010	92.30
				ECML-PKDD 2007	96.89
			Random Forest	CSIC 2010	93.71
				ECML-PKDD 2007	98.80
(Kirchner 2010) [26]	×	√	K-Means	HTTP Dataset	90.90
(Zolotukhin et al. 2012) [27]	×	√	N Gram	HTTP Dataset	97.70
(Tekerek et al. 2016)[11]				WAF 2015	97.60
	√	√	Bayes	CSIC 2010	94.50
				ECML-PKDD 2007	93.30
				WAF 2015	98.52
Proposed Model	√	√	ANN	CSIC 2010	96.74
				ECML-PKDD 2007	94.53

Tab. II *Comparison for datasets, feature, and detection method.*

this approach is the signature generation property. As a result of ABD, detected normal HTTP requests are updated to the SBDNR and anomaly HTTP requests are updated to SBDAR signature database. In this way, as is shown in Fig. 7, the proposed model decreases the detection duration.

5. Conclusion

In this study, a learning-based hybrid WAF model is proposed and implemented to prevent web-based attacks. Detection of web-based attacks to the web applications is performed by using SBD and ABD methods in tandem. In this way, respective disadvantages of SBD and ABD methods could be eliminated. Through the SBD, three detection stages are designed. The first stage is the SBDKIT which ensures detection of some known web-based attack types such as SQL Injection, Cross-Site Script (XSS) coding, Command Injection and Directory Traversal Attacks. The second is the SBDAR checklist, which is updated with HTTP requests that were previously identified as an anomaly. When such HTTP requests reach a web application a second time, they can be blocked by SBDAR without using ABD. The last stage of SBD is to update SBDNR with HTTP requests that were previously identified as normal HTTP requests by using ABD. HTTP requests which are not able to be detected by using the SBD, are directed to the ABD. For ABD, three features are selected to digitize the HTTP requests. These features are Alphanumerical Character Analysis, Letter Frequency Analysis and Sentence (Request) Length Analysis. ABD was implemented by using ANN. The latter is

one of the artificial intelligence methods according to the determined three features which are used as input data values. The distinctive characteristics of the proposed WAF model are its ability to determine the normal and anomaly HTTP requests that reach the web applications. With this new proposed model, clients who requested normal HTTP requests will easily access the web application without using ABD. This feature will increase the detection speed performance of the proposed model. Therefore, considering the web application HTTP request structure, with the update of SBDNR and SBDAR checklists, the disadvantage of SBD that is not efficient in zero-day attacks are possibly eliminated. So, in this way, the proposed model ensures adaptation against new attack types. As shown in Tab. II, a high mean achievement percentage (96.59 %) with three datasets (WAF 2015, CSIC 2010, ECML-PKDD 2007) was obtained when compared to other study results that suggest better results.

Acknowledgement

This study is supported by the project numbered 0235.STZ.201-1 within the SANTEZ projects of the Republic of Turkey, Ministry of Science, Industry, and Technology.

References

[1] BHOR R.V., KHANUJA K.H. Analysis of Web Application Security Mechanism and Attack Detection Using Vulnerability Injection Technique, *2016 International Conference on Computing Communication Control and automation (ICCUBEA)*, Pune, 2016, pp. 1–6, doi: 10.4236/jcc.2015.39004.

[2] VALEUR F., MUTZ D., VIGNA G. A learning-based approach to the detection of SQL attacks. In: *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, pp. 123–140, 205, Springer, Berlin, Heidelberg, doi: 10.1007/11506881_8.

[3] SANGHI D. Web Application Firewall, Kanpur, 2007.

[4] STEPHAN J.J., MOHAMMED S.D., ABBAS M.K. Neural network approach to web application protection. *International Journal of Information and Education Technology*, 2015, 5(2), p. 150.

[5] NGUYEN H.T., TORRANO-GIMENEZ C., ALVAREZ G., FRANKE K., PETROVİĆ S. Enhancing the effectiveness of web application firewalls by generic feature selection, *Logic Journal of IGPL*, 2012, 21(4), pp. 560–570, doi: 10.1093/jigpal/jzs033.

[6] PAŁKA D., ZACHARA M. Learning web application firewall-benefits and caveats. In: *International Conference on Availability, Reliability, and Security*, 2011, pp. 295–308, Springer, Berlin, Heidelberg, doi: 10.1007/978-3-642-23300-5_23.

[7] BASILE C., LIOY A. Analysis of application-layer filtering policies with application to HTTP. *IEEE/ACM Transactions on Networking (TON)*, 2015, 23(1), pp. 28–41, doi: 10.1109/TNET.2013.2293625.

[8] TORRANO-GIMENEZ C., PEREZ-VILLEGAS A., ALVAREZ G. A self-learning anomaly-based web application firewall. In: *Computational Intelligence in Security for Information Systems,* 2009, pp. 85–92, Springer, Berlin, Heidelberg, doi: 10.1007/978-3-642-04091-7_11.

[9] KRUEGEL C., VIGNA G. Anomaly detection of web-based attacks. In: *Proceedings of the 10th ACM conference on Computer and communications security*, 2003, pp. 251–26, doi: 10.1145/948109.948144.

[10] SINGH S., AGRAWAL S., RIZVI M.A., THAKUR R.S. Improved Support Vector Machine for Cyber Attack Detection. In: *Proceedings of the World Congress on Engineering and Computer Science WCECS*. San Francisco, USA, Vol. 1, 2011.

[11] TEKEREK A., GEMCİ C., BAY O.F. Design and Implementation of a Web-Based Intrusion Prevention system: A New Hybrid Model, *Journal of the Faculty of Engineering and Architecture of Gazi University*, 2016, 31(3), pp. 645–653, doi: 10.17341/gummfd.63355.

[12] JIA X. Design, Implementation and Evaluation of an Automated Testing Tool for Cross-Site Scripting Vulnerabilities. *Darmstadt University of Technology (TUD),* 2006.

[13] VURAL Y. Enterprıse Informatıon Securıty and Penetratıon Testing. Gazi University, 2007.

[14] KAPODISTRIA H., MITROPOULOS S., DOULIGERIS C. An Advanced Web Attack Detection and Prevention Tool. *Information Management & Computer Security*, 2011, 19(5), pp. 280–299, doi: 10.1108/09685221111188584.

[15] DESMET L., PIESSENS F., JOOSEN W., VERBAETEN P. Bridging the gap between web application firewalls and web applications. In Proceedings of the fourth ACM workshop on Formal methods in security, 2006, pp. 67–77, doi: 10.1145/1180337.1180344.

[16] TAKAHASHI H., AHMAD H.F., MORI K. Application for autonomous Decentralized multi Layer Cache System to Web application Firewall. In 2011 Tenth International Symposium on Autonomous Decentralized Systems, 2011, pp. 113–120, doi: 10.1109/ISADS.2011.20.

[17] AUXILIA M., TAMILSELVAN D. Anomaly Detection Using Negative Security Model in Web Application, Computer Information Systems and Industrial Management Applications (CISIM), 2010 International Conference, 2010, pp. 481–486, doi: 10.1109/CISIM.2010.5643461.

[18] INGHAM K.L. Anomaly Detection for HTTP Intrusion Detection: Algorithm Comparisons and the Effect of Generalization on Accuracy. University of New Mexico, 2007.

[19] ROESCH M. Snort: Lightweight intrusion detection for networks, In LISA '99: 13th Systems Administration Conference. Washington, DC, USA, 1999, 99(1), pp. 229–238.

[20] RAUT A.S., SINGH K.R. Anomaly based intrusion detection-a review, *International Journal on Network Security*, 5(3), 7, 2014.

[21] GROFF J.R., WEINBERG P.N. SQL: The Complete Reference, Berkeley, California 94710, 1999.

[22] RAHMAN C.M., FARID D.M., HARBI N., BAHRI E., RAHMAN M.Z. Attacks classification in adaptive intrusion detection using decision tree, *World Academy of Science, Engineering and Technology*, 2010, 63(3), pp. 86–90.

[23] COOK S. A Web Developer's Guide to Cross-Site Scripting, SANS Institute 2003, 2003, pp. 1–11.

[24] VIJAYAKUMAR H., JAEGER T. The Right Files at the Right Time. In G. X. Ehab Al-Shaer, Xinming Ou, ed. Automated Security Management. The Pennsylvania State University, University Park, PA, USA: Springer International Publishing, 2013, pp. 119–133, doi: 10.1007/978-3-319-01433-3_7.

[25] THIELECKE H. Command Injection Attacks, Continuations, and the Lambek Calculus. In: *Electronic Proceedings in Theoretical Computer Science*. 2016, pp. 81–96.

[26] KIRCHNER M. A framework for detecting anomalies in http traffic using instance-based learning and k-nearest neighbor classification. In: *2010 2nd International Workshop on Security and Communication Networks (IWSCN),* 2010, pp. 1–8, doi: 10.1109/IWSCN.2010.5497997.

[27] ZOLOTUKHIN M., HAMALAINEN T., JUVONEN A. Online Anomaly Detection by Using N-Gram Model and Growing Hierarchical Self-Organizing Maps. *IWCMC 2012 – 8th International Wireless Communications and Mobile Computing Conference*, 2012, pp. 47–52, doi: 10.1109/IWCMC.2012.6314176.