A new approach to improving web application firewall performance based on support vector machine method with analysis of Http request

Nguyen Manh Thang, Truong Phi Ho, Hoang Thanh Nam

Abstract - **Amount of attacks on information system is rapidly increasing not only in numbers but also in quality. Each attack violates properties of confidentiality, integrity, and accessibility of information, most attacks pursue financial gain, especially web attacks because almost companies use web applications for their businesses. The issue of protecting personal data from these attacks has become critical for all organizations and companies. Thus, the need to use an intrusion detection system and an intrusion prevention system to protect these data is relevant. Traditional means of protecting access to the corporate network (firewalls) are not able to protect against most threats directed at Web resources. The reason is that attacks on such resources most often occur at the application level, in the form of HTTP / HTTPS-requests to the site, where traditional firewalls have extremely limited opportunities for analysis and detection attacks. For protecting web resources from attacks at the application level we have special tools - web application firewall (WAF). The task of the tool is detecting and blocking attacks on Web resources at the application level. However, the analysis of incidents of information security shows that even with a class of means of detecting attacks on Web resources, their effectiveness does not provide a 100% detection level. With an aim of applying machine learning methods to improve WAF performance. The author discusses as popular types of attacks on Web applications and the survey of machine learning methods in the attack detection task to build an algorithm for automatic detection attacks based on the support vector machine and analysis of HTTP request.**

Tóm tắt - **Số lượng các cuộc tấn công vào hệ thống thông tin đang gia tăng nhanh chóng không chỉ về số lượng mà còn về mức độ nguy hại. Mỗi cuộc tấn công đều hướng đến việc ảnh hưởng đến tính bảo mật, tính toàn vẹn và tính sẵn sàng của thông tin, hầu hết các cuộc tấn công nhằm thu lợi về tài chính, đặc biệt là các cuộc tấn công web vì hầu hết các công ty sử dụng các ứng dụng web cho doanh nghiệp của họ. Vấn đề bảo vệ dữ liệu cá nhân khỏi các cuộc tấn công này đã trở nên quan trọng đối với tất cả các tổ chức và công ty. Do đó, nhu cầu sử dụng một hệ thống phát hiện xâm nhập và một hệ thống ngăn chặn xâm nhập để bảo vệ những dữ liệu này là có liên quan. Các phương tiện truyền thống để bảo vệ quyền truy cập vào mạng công ty (tường lửa) không thể bảo vệ khỏi hầu hết các mối đe dọa nhắm vào tài nguyên web. Nguyên nhân là do các cuộc tấn công vào các tài nguyên như vậy thường xảy ra nhất ở tầng ứng dụng, dưới dạng HTTP / HTTPS-request tới trang web, nơi tường lửa truyền thống có rất ít cơ hội để phân tích và phát hiện các cuộc tấn công. Để bảo vệ tài nguyên web khỏi các cuộc tấn công ở cấp ứng dụng, chúng ta có các công cụ đặc biệt - tường lửa ứng dụng web (WAF). Nhiệm vụ của công cụ này là phát hiện và ngăn chặn các cuộc tấn công vào tài nguyên Web ở cấp độ ứng dụng. Tuy nhiên, phân tích các sự cố về an toàn thông tin cho thấy rằng ngay cả với một loại phương tiện phát hiện các cuộc tấn công vào tài nguyên web cũng không thể phát hiện được 100% các nguy cơ. Với mục đích áp dụng các phương pháp học máy để cải thiện hiệu suất WAF. Tác giả thảo luận về các dạng tấn công phổ biến trên ứng dụng web và khảo sát các phương pháp học máy trong nhiệm vụ phát hiện tấn công để xây dựng thuật toán cho các cuộc tấn công phát hiện tự động dựa trên vector hỗ trợ máy và phân tích yêu cầu HTTP.**

Keywords: SQL injection, XSS, path Traversal, DDOS, CSRF, signature method, anomaly detection method, machine learning method, HTTP request.

Từ khóa: SQL injection, XSS, path Traversal, DDOS, CSRF, phương pháp dựa trên mẫu dấu hiệu, phương pháp phát hiện bất thường, phương pháp học máy, truy vấn HTTP

I. INTRODUCTION

Attacks on web applications open up ample opportunities for cybercriminals: access to internal company resources and personal information of users. The attack disrupts the operation of the application or bypasses the business logic of the system. Virtually any attack can bring financial benefit to the attackers and losses, both financial and reputational, - for the Web application owner.

In addition, users of web applications are at risk, as successful attacks can steal credentials, perform actions on websites on behalf of users, and infect workstations with malware.

Figure 1 provides information about the attacks on web applications in 2018 and 2019 respectively. PTAF (Positive Technologies Application Firewall), as well as the results of the PTAF for the protection of web applications by Positive Technologies [1].

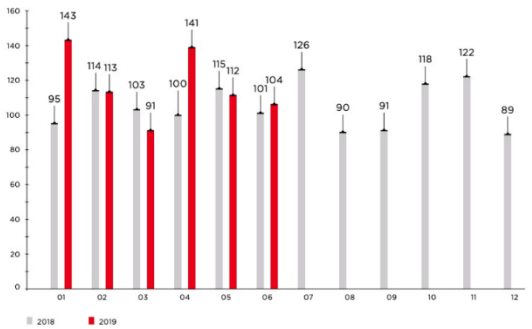


Figure 1. Number of incidents in 2018 and 2019 (by months)

According to Positive Technologies' Web Application Firewall Survey, the largest average number of attacks per day - approximately 3,500 attacks - was recorded during pilot projects in government agencies. Online stores rank second in this ranking: about 2,200 attacks were registered per day, and almost all of them were carried out without the use of automated scanning tools. Therefore, the task of protecting the information technology system of organizations becomes urgent.

II. COMMON ATTACKS ON WEB APPLICATIONS

In the age of information technology, every web application has vulnerabilities. Knowing what types of vulnerabilities are most dangerous and how to mitigate the risks will give the system administrator an edge in protecting the web applications of companies and organizations.

A. CODE INJECTION.

Code injection occurs when an attacker sends invalid and untrusted data to an application as part of a command or request. An attacker has the malicious intent to force an application to perform unintended behavior in order to collect data or create damage.

Some of the most common types of injections are:

- Injection or SQL injection [2 - 4];
- OS command;
- LDAP injection [5];
- OGNL injection.

The main reason for such injections is the lack of validation and cleaning of the data used by the application. Injection prevention guidelines depend on the technology programmers are using. In general, the systems specialist must ensure that his team of employees adheres to the security requirements when using commands in the system.

B. INCORRECT AUTHENTICATION AND SESSION MANAGEMENT

Invalid authentication and session control vulnerabilities allow attackers to use manual or automatic methods to gain control over any account on the system.

Web applications are one of the most vulnerable and common targets. Attackers have access to information on hundreds of millions of username and password combinations for standard accounts. This access allows attackers to easily perform dictionary attacks, to automate brute force attacks, and other GPU hacking tools to gain access to the system.

There are many risks that can arise during authentication:

- The URL can contain the session ID and leak it in another user's Referrer header;
- The password is not encrypted and can be easily decoded during storage;
- Session Fixation vulnerability;
- Session hijacking attacks can occur when the session timeout is not properly implemented or when HTTP is used without SSL.

C. LEAK OF CONFIDENTIAL DATA

Leakage of confidential data is one of the most popular vulnerabilities for users of online resources. An attacker has access to personal data of users without their permission.

Sensitive data such as passwords, credit card numbers, credentials, social security numbers, medical records require additional protection. Therefore, it is important for any organizations to understand the necessity of protecting user data.

According to OWASP, one of the most common and serious situations is when a website does not use TLS for all pages or supports weak encryption. Confidential data must be encrypted at all times, including when sending and storing - no exceptions are allowed. Credit card information and user passwords are never sent or stored unencrypted. Obviously, encryption and hashing algorithms are not a weak security method. In addition, web security standards recommend using AES (256 bits or more) and RSA (2048 bits or more).

D. CROSS-SITE SCRIPTING.

Cross-site scripting (XSS) [6, 7] is a user data validation error that allows attackers to transmit dangerous code on the server from the user's browser.

The complexity of this attack lies in the fact that the algorithm for filtering incoming data should not create unreasonable restrictions for legal users, but at the same time, it should make an XSS attack by an attacker impossible. OWASP XSS Prevention provides details on the required data escaping techniques.

To protect a web application from this attack, the sysadmin needs to apply context sensitive encoding when modifying the browser document on the client side to act against DOM XSS and navigate away from the untrusted context-based HTTP request data in HTML output (body, attribute, JavaScript, CSS or URL) to eliminate stored XSS vulnerabilities.

E. DENIAL OF SERVICE

Denial of service [8-10] is one of the most popular types of attacks on web applications.

DDoS attacks are getting stronger, more sophisticated, and more difficult to prevent. The types of DDoS attacks vary, but they all affect the performance of an organization's website. Therefore, organizations and online stores need to understand and anticipate possible risks.

F. CROSS-SITE REQUEST FORGERY

CSRF (Cross-Site Request Forgery) allows an attacker to perform actions on the server on behalf of the victim. The browser has been misled by some third party that is abusing its powers.

CSRF very rarely appears among CVEs (common vulnerabilities and threats) - less than 0.1% in 2008, but in reality, it is a "sleeping giant". CSRF is becoming an important security issue.

While there are a huge number of attacks on web applications, there are also many known ways to detect them.

Let us now consider the main methods of detection, as well as the advantages and disadvantages of their application in the problem of classifying attacks, which is solved to detect them.

III. METHODS FOR DETECTING ATTACKS ON WEB APPLICATIONS

Many intrusion detection systems (IDS) use 3 main methods: the signature method [11, 12], the anomaly detection method [13-15] and

the most popular method is the method using machine learning.

A. SIGNATURE METHODS

Signature analysis is based on the assumption that the attack scenario is known and an attempt to implement, can be detected in event logs or by analyzing network traffic. Ideally, the system administrator should fix all known vulnerabilities.

For example, the free product Snort [16, 17] is a typical intrusion detection system because, in terms of capabilities, Snort is primarily concerned with detecting attacks with a large signature database, and its prevention functionality is limited compared to other attack prevention systems.

The intrusion detection system works with a signature, tracking network packets and comparing them with a signature database, attributes of known attacks, similar to how anti-virus software works. The main problem with this system is that it may not detect a new attack if the signature for its identification was not updated.

B. ANOMALY DETECTION METHODS

Anomaly-based detection is a way to detect unusual traffic behavior within a network. The IDS will, based on the anomalies, identify network traffic by monitoring methods and compare it with the established baseline.

The attacker, trying to attack the system, the frequent use of applications and different testing methods. Intrusion operations are often different from the usual activities of users working on the system. Any penetration testing software can recognize suspicious activity beyond a certain threshold.

The main problem with this system is the inability to build an accurate model for all attacks, although it can detect a new attack.

Since the above two methods have advantages and disadvantages, the system administrator needs to combine both methods and apply machine learning to improve the efficiency of the system.

C. MACHINE LEARNING METHODS FOR DATA CLASSIFICATION PROBLEM

Machine learning methods [18], as well as methods of computational intelligence, are used both in the detection of anomalies and in the detection of abuse of the granted rights and powers. Here's a quick overview of the main machine learning methods.

1. Bayesian network

Bayesian network [19, 20] is a graph probabilistic model, which is a set of variables and their probabilistic dependences according to Bayes. Many works use a naive Bayesian algorithm.

A naive Bayesian classifier [21, 22] combines a model with a decision rule.

Based on the experience of working with a naive Bayesian classifier on a large dataset, the following conclusions can be drawn:

- The condition of independence of variables in the model is satisfied (depending on the nature of the data), it is believed that the naive Bayesian classifier gives better results than logistic regression when there is less data for training.
- Although the training and classification times are shorter than most machine learning methods when working with a large dataset, the accuracy of this method is low.

2. K-nearest neighbors

The k-nearest neighbor (k-NN) method [23, 24] is a classification method, the basic principle of which is to assign an object of the class that is the most common among the neighbors of a given object. Neighbors are formed from a set of objects, the classes of which are already known, based on a given value $k (k \geq 1)$. It is determined which of the classes is the most numerous among them. If $k = 1$, then the object simply belongs to the class of the single nearest neighbor.

In [25], a mixed approach was used - combining the genetic algorithm [26] and the k-nearest neighbors' classifier to detect denial of service attacks. The purpose of the genetic algorithm - to find the optimal weight vector, which is presented as a "weight" sign i, $1 \le i \le n$. The vector W will influence the computation of the distance and promote the classification of the k-nearest neighbors. For any two weight vectors of features $X = (x_1, x_2, \ldots, x_n)$ and $Y = (y_1, y_2, \ldots, y_n)$ the distance between them is calculated as follows:

$$(X,Y) = \sqrt{\omega_1 (x_1 - y_1)^2 + \omega_2 (x_2 - y_2)^2 + \ldots + \omega_n (x_n - y_n)^2}. \quad (1)$$

Each value belongs to the segment [0,1]. After the evolution of the genetic algorithm at the training stage, the optimal weight vector can be obtained, which leads to the best result of the KNN classification. The detection accuracy of this approach is approximately 96.75%.

3. Decision tree

Decision tree [26, 27] is a decision support tool used in statistics and data analysis for predictive models. A decision tree is a tree-like structure of "leaves" and "branches".

There are two types of decision tree relatives:

- Regression trees estimate value functions with real numbers rather than be used for classification problems;
- The classification tree used in the classification task.

Compared to other data analysis, the use of decision trees has the following advantages:

- The study of characteristics from input and output data is the result of the shape of a decision tree. This means that it is easy to see the characteristics of the input data;
- Other types of machine learning methods require a lot of preprocessing, and the decision tree requires almost no preprocessing;
- For types of machine learning methods such as neural networks that are considered black box models, the decision tree is similar to the white box model;
- Support for assessing the accuracy of the created models.

4. Neural network

An artificial neural network (ANN) [28] is a mathematical model, as well as its software or hardware implementation, built on the principle of the organization and functioning of biological neural networks - networks of nerve cells of a living organism.

Since a conventional intrusion detection system is not always able to identify every attack, it is necessary to have a system that can regularly update the signs of new attacks on the system.

The purpose of using neural networks is to detect attacks in the system, such as: the use of malicious code (viruses, Trojans ...), packet congestion, network scanning, denial of service (DoS) attacks, privilege escalation, and so on.

Solving classification problems used a neural network has a number of disadvantages:

- Most network design methods are heuristic;
- Difficulties in finding a sufficient number of training examples;
- The requirement to perform complex tuning of internal neurons and connections between them;
- The difficulty of understanding the operation of the network, since it works according to the " black box " model.

5. *Support vector machine*

Support vector machine (SVM) [29] was originally invented by V. N. Vapnik, and the standard SVM algorithm was proposed by V. N. Vapnik and Corinna Cortes in 1995.

SVM has been successfully used to solve real-world problems such as text recognition, image recognition, handwriting recognition, electronic

spam classification, network intrusion detection, and so on.

The essence of the method is to transform the original data space into a new final space, in which a simpler classification is possible. Any point in the dataset will be anchored to a specific coordinate.

SVM has some advantages:

- Obtaining a classification function with a minimum upper estimate of the expected risk (level of classification error);
- Using a linear classifier to work with nonlinearly separated data, combining simplicity with efficiency.

The disadvantages of the methods are as follows:

- In the case where the number of attributes of the dataset is much larger than the number of data, the algorithm gives rather poor results;
- Classification, according to this method, is an attempt to differentiate objects into two layers, separated by hyperplanes, and does not explain the likelihood of the appearance of points in the separating set.

Let us consider the essence of the proposed new approach in the problem of classification of queries based on the support vector machine and some additional attributes of queries.

IV. A NEW APPROACH TO IMPROVING WAF PERFORMANCE IN THE CLASSIFICATION PROBLEM

Since there are many algorithms for classifying queries in modern WAF, modern WAF contains several built-in modules: regular expression module, behavior module, tokenization module, artificial intelligence module, and so on. The new approach is to use a machine learning method (support vector machine) with some query attributes.

The scheme of the classification process consists of 7 main stages: data collection to create a query base (A1); preliminary data processing (A2); payload comparison (A3); checking regular expressions (A4); calculation of request attributes (A5); converting text data into vectors (A6); classification of queries based on the support vector machine (A7).

Each stage of the work will be described in detail below.

A. COLLECTING DATA TO CREATE A DATABASE OF QUERIES (A1)

Information from several sources can be used to create a query database:

- The CSIC 2010 HTTP dataset contains thousands of automatically generated web requests. It was developed at the "Information Security Institute" CSIC (Spanish National Research Council http://www.isi.csic.es/dataset/).
- XSSED dataset (http://www.xssed.com/);
- Examples of SQL injection attacks (https://www.acunetix.com/website security / sql-injection /).

All data is written in a file with the .csv extension. This file is the output of Algorithm A1.

B. DATA PREPROCESSING (A2)

Data preprocessing is a very important step in solving any machine learning problem. Most datasets used in machine learning tasks need to be processed, cleaned, and transformed before a machine learning algorithm can be trained on them.

The datasets corresponding to the problems are actually different. In this task, the data is sorted in tables by fields. Before sorting the data, we deleted all words that have no meaning (stop words or stop words). These words are defined by the development of an embedded software function (Anacoda with Python 3.0) and the stop word library by the author of this article.

C. PAYLOAD COMPARISON (A3)

Payload is an important term used in many fields of science, including information technology.

- In the learning phase: with the exploration of dangerous queries, the payload is retrieved and stored in the database.
- In the discovery phase: after processing the data, the payload comparison module is launched. The payload is retrieved from the input request and then compared to the list of saved payloads. If the payload is found, then this request will be blocked. Otherwise, all data remaining requests will be sent to the module "check regular expressions".

D. CHECKING REGULAR EXPRESSIONS (A4)

Regular expressions are patterns used to find character sets that are concatenated into character strings. In many programming languages like JavaScript, C # and so on, regular expressions are also objects.

- In the learning phase: with the investigation of dangerous queries, regular expressions were created.
- In the discovery phase: after the payload comparison module, the regular expression validation module is launched. Input queries are checked against the list of stored regular expressions. If the request template does not match at least one saved template, then this request will be blocked. Otherwise, all data from other requests will be sent to the module "calculating attributes".

In Table 1 presents some regular expressions to detect the type of code injection attacks.

TABLE 1. EXAMPLES OF REGULAR EXPRESSIONS FOR DETECTING SQL INJECTION AND XSS

Attacks	Expression
SQL attack	/ (\% 27) | (\ ') | (\ - \ -) | (\% 23) | (#) / ix
SQL attack	/ ((\% 3D) | (=)) [^ \ n] * ((\% 27) | (\ ') | (\ - \ -) | (\% 3B) | (;)) / i
SQL attack	/ \ w * ((\% 27) | (\ ')) ((\% 6F) | o | (\% 4F)) ((\% 72) | r | (\% 52)) / ix
SQL attack	/ ((\% 27) | (\ ')) union / ix
SQL attack	/ exec (\ s | \ +) + (s | x) p \ w + / ix
XSS attack	/ ((\% 3C) | <) ((\% 2F) | \ /) * [a-z0-9 \%] + ((\% 3E) |>) / ix
XSS attack	/ ((\% 3C) | <) [^ \ n] + ((\% 3E) |>) / I

E. CALCULATION OF REQUEST ATTRIBUTES (A5)

After examining the structure of Http requests from datasets, we added three new attributes in the process of classifying requests: changing the length of requests, changing the length of the request arguments, and the frequency of occurrence of key characters. During the training phase, we saved all the necessary values, such as query lengths, query argument lengths and key symbols. These values are stored in the database (our task uses the MySQL database management system).

1. The length of the request sent from the browser

We assume that the length of the request sent from the user's browser varies slightly within a certain range. However, in the event of a hack, the length of the data field may change, therefore, the length of the request increases. For example, in the case of SQL injection, cross-site scripting. Therefore, it is proposed to use the change in the request length to detect attacks from users.

- In the training phase: suppose that the length of the input data is equal $l_{u_1}, l_{u_2}, \ldots, l_{u_n}$, the mathematical expectation μ_u and variance α_u^2 of the data set.
- In the discovery phase: the mathematical expectation μ_u and variance α_u^2 of the data set are specified. Application of Chebyshev's inequality gives an estimate of the probability that a random variable will take a value far from its mean.

$$P(|x - \mu_u| > t_u) < \frac{\alpha_u^2}{t_u^2}, \qquad (2)$$

where x is a random variable; t_u - any value.

Accordingly, for any probability distribution with mean μ_u and variance α_u^2, a given value is obtained x, then the deviation x from the mean μ_u exceeds any blocked threshold $\frac{\alpha_u^2}{t_u^2}$.

In this case $t_u = \left| l_{a_{ij}} - \mu_u \right|$ (l_a: input request length) is selected. The higher the value, the closer the value l is to the average value, otherwise it is a sign of an attack.

$$P(|x-\mu_u| > |l-\mu_u|) < P(l) = \frac{\alpha_u^2}{|l-\mu_u|^2}. \quad (3)$$

2. The length of the arguments of the request sent from the browser

- In phase of study: we assume that the length of the argument is the input data $l_{a_1}, l_{a_2}, \ldots, l_{a_n}$, the expectation μ_a and variance α_a^2 of the data set.
- In the discovery phase: the mathematical expectation μ_a and variance α_a^2 of the data set are specified. Application of Chebyshev's inequality gives an estimate of the probability that a random variable will take a value far from its mean.

$$P(|x-\mu_a| > t_a) < \frac{\alpha_a^2}{t_a^2}, \quad (4)$$

where x is a random variable, t_a is any value.

Accordingly, for any probability distribution with mean μ_a and variance α_a^2, a given value is obtained x, then the deviation x from the mean μ_a exceeds any blocked threshold $\frac{\alpha_a^2}{t_a^2}$.

In this case, for each request argument, the author chooses $t_{a_{ijk}} = \left| l_{a_{ij}} - \mu_{a_{ij}} \right|$ ($l_{a_{ijk}}$ is the length of the k-th argument of the input request; $i, k = 1,2,\ldots,n$ and $j = 1,2,\ldots,m$; where n is the number of requests; m is the number of arguments). The higher the value, the closer the value $l_{a_{ijk}}$ is to the average value, otherwise it is a sign of an attack.

$$P(\left| x-\mu_{a_{ij}} \right| > \left| l_{a_{ijk}} - \mu_{a_{ij}} \right|) < P(l_{a_{ijk}}) = \frac{\alpha_{a_{ij}}^2}{\left| l_{a_{ijk}} - \mu_{a_{ij}} \right|^2}. \quad (5)$$

3. The frequency of occurrence of key symbols

From the training sample of legitimate requests, separate non-repeating characters (including those taking into account different encodings) must be selected in order to compose a set of alphabet characters S. Thus, when a symbol appears in a request $b \notin S$, the counter value p_b for this attribute is increased by one. The value of the attribute itself is calculated as the ratio of the counter value to the cardinality of the alphabet set:

$$P = \frac{p_b}{|S|}. \quad (6)$$

F. CONVERT TEXT DATA TO VECTORS (A6)

The module for converting string data into vectors is implemented using the *tf-idf* technology, you can evaluate the importance of a word in a query string.

Let's apply the *tf-idf* technology in this task, for each request we will find the words in the request. For each word *t* in query *d* in the set of queries D, the following formulas are used:

$$tfidf(t,d) = tf(t,d) \cdot idf(t), \quad (7)$$

where *tf, idf* values are calculated as:

$$tf(t,d) = \frac{count(t,d)}{\sum_{v \in d} count(v,d)}, \quad (8)$$

where v are the rest of the words in the query d

$$idf(t) = \log \frac{|D|}{|d \in D : t \in d|}. \quad (9)$$

After the *tf-idf* computation process, the query string data is converted to vectors. The 3 values in the above paragraph are added to the query vectors.

G. QUERY CLASSIFICATION BASED ON SUPPORT VECTOR MACHINE (A7)

The problem of teaching by precedents is considered $< X, Y, y^*, X^l >$ where X - space of objects (space of requests), Y - set of answers (set of classes); $y^* : X \to Y$ - the target

dependence, the values of which are known only on the objects of the training set.

In our problem: $X = R^n, Y = \{0,1\}$; 0 is not an attack, but 1 is an attack. We will build a linear threshold classifier:

$$a(x) = sign(\sum_{j=1}^{n} \omega_j x^j - \omega_0), \quad (10)$$

where $x = (x^1, x^2, \ldots, x^n)$ is an attribute description of the object x; vector $\omega \in R^n$ and ω_0 - scalar threshold, are called parameters of the algorithm.

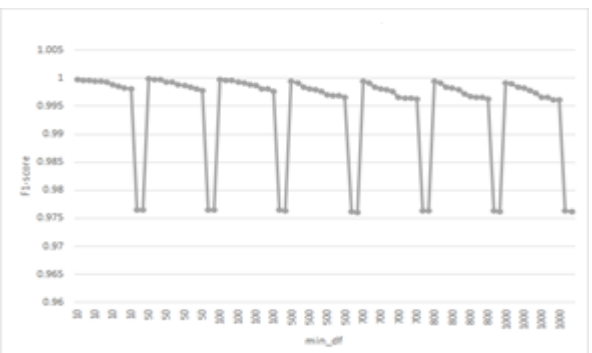
The equation $\langle \omega, x \rangle = \omega_0$ is a hyperplane. The dividing hyperplane is as far as possible from the points of both classes close to it. It is necessary to maximize the margin between the classes.

For everyone $x_i \in X^l$:

$$\langle \omega, x_i \rangle - \omega_0 = \begin{cases} \le 0, \ if \quad y_i = 0 \\ \ge 1, \ if \quad y_i = 1 \end{cases}. \quad (11)$$

The condition $0 \le < \omega, x > -\omega_0 \le 1$ sets the band separating the classes. None of the points of the training sample can lie within this strip. The boundaries of the strip are two parallel hyperplanes with the direction of the vector ω. The points nearest to the dividing hyperplane lie exactly on the boundaries of the strip.

The construction of an optimal separating hyperplane is reduced to the problem of minimizing a quadratic form under l-constraints (11) with $(n+1)$ variables ω and ω_0:



$$\begin{cases} < \omega, \omega > \to \min \\ y_i(< \omega, \omega > -\omega_0) \ge 1, i = 1, \ldots, l \end{cases}. \quad (12)$$

By the Kuhn - Tucker theorem, such a problem is equivalent to the dual problem of finding a point of the Lagrage function:

$$\begin{cases} L(\omega, \omega_0, \lambda) = \frac{1}{2} < \omega, \omega > - \\ \sum_{i=1}^{l} \lambda_i (y_i (< \omega, x_i > -\omega_0) - 1 \\ \to \min_{\omega, \omega_0} \max_{\lambda} \\ \lambda_i \ge 0 \\ \lambda_i = 0, if \ < \omega, x_i > -\omega_0 = y_i \\ i = 1, \ldots, l \end{cases}, \quad (13)$$

where: $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_l)$ is the vector of dual variables.

To solve this problem, we calculate:

$$\frac{\partial L}{\partial \omega} = \omega - \sum_{i=1}^{l} \lambda_i y_i x_i = 0 \to \omega = \sum_{i=1}^{l} \lambda_i y_i x_i, \quad (14)$$

$$\frac{\partial L}{\partial \omega_0} = -\sum_{i=1}^{l} \lambda_i y_i = 0 \to \sum_{i=1}^{l} \lambda_i y_i = 0. \quad (15)$$

Putting (14) and (15) the Lagrange functions, we get:

$$\begin{cases} -L(\lambda) = -\sum_{i=1}^{l} \lambda_i + \\ \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \lambda_i \lambda_l y_i y_j (< x_i, x_j >) \to \min_{\lambda}. \\ \sum_{i=1}^{l} \lambda_i y_i = 0 \end{cases} \quad (16)$$

Then the vector is calculated by formula (14). To determine ω_0 we need to take the vector x_i and expressed ω_0 from the equation: $\omega_0 = < \omega, x_i > -y_i$. As a result, the classification algorithm can be written as:

$$a(x) = sign(\sum_{i=1}^{l} \lambda_i y_i < x_i, x > -\omega_0). \quad (17)$$

Using (17), the class of incoming requests is determined.

After applying the support vector machine to classify queries, two sets of queries will be received with tags "0" or "1". All requests with marks "0" will be executed on the server, and the rest of requests with marks "1" will be blocked.

Figure 1. Support vector machine accuracy versus min-df values

V. EXPERIMENTAL EVALUATION OF THE APPROACH

The experiment used data from popular sources (the source address is specified in paragraph 4.1) and 3-grams with cross-validation to verify the results of the approach. The result of checking the dataset (20,000 dangerous queries and 100,000 normal queries) using tf-idf technology (80% of the data for training and 20% of the data for testing) is presented in Table 2.

TABLE 2. ACCURACY OF SOME MACHINE LEARNING METHODS WITH A GIVEN DATASET

Method	Accuracy	F1-score
New approach	0.999963	0.999879
Support vector machine	0.997563	0.996861
Neural network	0.99701	0.996779
Logical regression	0.996313	0.979886
Random forest	0.966545	0.795221
K-nearest neighbors	0.954301	0.67221

Since the accuracy of the support vector machine operation depends on the choice of its parameters, an experimental evaluation of this method was carried out with a change in some parameters. The results of this assessment are shown in Figure 1. Note that the new three-attribute support vector machine approach gives better results than the known ones. Accuracy values approach 1 (80% training data and 20% testing data). The new approach is only effective for variable query length attacks (code injection, cross-site scripting) and requires high computational power as the dataset grows in size.

VI. CONCLUSION

The paper provides a brief overview of popular attacks on web applications and methods for detecting them, also a comparative analysis of these methods. Each method has its own advantages and disadvantages, hence the study used not only the signature method, but also machine learning methods to improve the performance of the WAF. It is noted that the machine learning method is widespread and is used in many intrusion detection systems, including intrusion detection systems.

Combining signature-based methods with machine learning methods makes intrusion detection systems more intelligent and autonomous when new attacks are detected, since static methods can be bypassed by attackers.

To improve the accuracy of the proposed approach, it is proposed:

- Using a combination of machine learning methods;
- Increasing the number of quality attributes of queries;
- Adding regular expressions for specific attacks;
- Updating signature databases, since WAF works not only with signatures, but also with anomaly detection (including machine learning methods).

Further research will be focused on cloud intrusion detection systems and firewall service for cloud web application, as cloud computing is a major paradigm shift for computer networks.

REFERENCES

[1] Авезова Яна. Веб-приложения: тестируем на защищенность // Positive Research 2019. — 2019. — С. 144—148.

[2] Ross Kevin. SQL Injection Detection Using Machine Learning Techniques and Multiple Data Sources. — 2018.

[3] Uwagbole Solomon Ogbomon, Buchanan William J, Fan Lu. Applied machine learning predictive analytics to SQL injection attack detection and prevention // 2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM). — IEEE. 2017. — P. 1087–1090.

[4] Mishra Sonali. SQL Injection Detection Using Machine Learning. —2019.

[5] Бодров В.А., Белоусова Е.С. Анализ и методы защиты веб-приложений от атак типа LDAP-инъекция. — 2019.

[6] Lakhapati Shweta A, Shirbhate PV, Jagtap Shivani, Shrirang Ashwini. Cross site scripting attack // International Journal of Electronics, Communication and Soft Computing Science & Engineering (IJECSCSE). — 2018. — P. 131–135.

[7] Mereani Fawaz A, Howe Jacob M. Detecting Cross-Site Scripting Attacks Using Machine Learning // International Conference on Advanced Machine Learning Technologies and Applications. — Springer. 2018. — P. 200–210.

[8] Akamai. Q3 2017 State of the Internet / Security Report: DDoS Attack Update Q3 2017 vs. Q2 2017. — 2017. — URL: https://www.akamai. com/us/en/about/our-thinking/state-of-the-internet-report/global-stateof-the-internet-security-ddos-attack-reports.jsp.

[9] Doshi Rohan, Apthorpe Noah, Feamster Nick. Machine learning ddos detection for consumer internet of things devices // 2018 IEEE Security and Privacy Workshops (SPW). — IEEE. 2018. — P. 29–35.

[10] Idhammad Mohamed, Afdel Karim, Belouch Mustapha. Semi-supervised machine learning approach for DDoS detection // Applied Intelligence. — 2018. — Vol. 48, no. 10. — P. 3193–3208.

[11] Fleming Theodor, Wilander Hjalmar. Network intrusion and detection: An evaluation of snort. 2018.

[12] Shah Syed Ali Raza, Issac Biju. Performance comparison of intrusion detection systems and application of machine learning to Snort system // Future Generation Computer Systems. — 2018. — Vol. 80. — P. 157– 170.

[13] Duessel Patrick, Gehl Christian, Flegel Ulrich, Dietrich Sven, Meier Michael. Detecting zero-day attacks using context-aware anomaly detection at the application-layer // International Journal of Information Security. — 2017. — Vol. 16, no. 5. — P. 475–490.

[14] Zhang Ming, Lu Shuaibing, Xu Boyi. An anomaly detection method based on multi-models to detect web attacks // 2017 10th International Symposium on Computational Intelligence and Design (ISCID). Vol. 2. — IEEE. 2017. — P. 404–409.

[15] Ciocarlie Gabriela F, Stavrou Angelos, Stolfo Salvatore J, Keromytis Angelos D. Systems, methods, and media for generating sanitized data, sanitizing anomaly detection models, and/or generating sanitized anomaly detection models. — 1 8/2019. — US Patent App. 10/178,113.

[16] Caesarano Arif Roid, Riadi Imam. Network Forensics for Detecting SQL Injection Attacks Using NIST Method. — 2018.

[17] Olanrewaju Rashidah Funke, Khan Burhan Ul Islam, Najeeb Athaur Rahman, Zahir KN, Hussain S. Snort-based smart and swift intrusion detection system // Indian Journal of Science and Technology. — 2018. — Vol. 8, no. 1. — P. 1–9.

[18] Браницкий А.А., Котенко И.В. Анализ и классификация методов обнаружения сетевых атак // Труды СПИИРАН. — 2016. — Т. 2, № 45. — С. 207—244.

[19] Rangaraju Naveen Kumar, Sriramoju Shoban Babu, Sarma SSVN. A study on machine learning techniques towards the detection of distributed denial of service attacks // International Journal of Pure and Applied Mathematics. — 2018. — Vol. 120, no. 6. — P. 7407–7423.

[20] Shukla Satya Narayan, Sahu Anit Kumar, Willmott Devin, Kolter J Zico. Black-box Adversarial Attacks with Bayesian Optimization // arXiv preprint arXiv:1909.13857. — 2019.

[21] Swarnkar Mayank, Hubballi Neminath. OCPAD: One class Naive Bayes classifier for payload-based anomaly detection // Expert Systems with Applications. — 2016. — Vol. 64. — P. 330–339.

[22] Zhang Bing, Liu Zhiyang, Jia Yanguo, Ren Jiadong, Zhao Xiaolin. Network Intrusion Detection Method Based on PCA and Bayes Algorithm // Security and Communication Networks. — 2018. — Vol. 2018.

[23] Васильев В.И., Шарабанов И.В. Интеллектуальная система обнаружения атак в локальных беспроводных сетях // Вестник Уфимского государственного авиационного технического университета. 2015. Т. 19, 4 (70).

[24] Gupta Jyotika, Chaturvedi Krishna Nand, Gupta Jyotika, Chaturvedi Krishna Nand. Improved Algorithm for Network Intrusion Detection System based on K-Nearest Neighbor: Survey // International Journal. 2016. Vol. 3. P. 81–84.

[25] Su Ming-Yang. Real-time anomaly detection systems for Denial-ofService attacks by weighted k-nearest-neighbor classifiers // Expert Systems with Applications. — 2011. — Vol. 38, no. 4. —

P. 3492–3498.

[26] Lee Chi Hoon, Chung Jin Wook, Shin Sung Woo. Network intrusion detection through genetic feature selection // Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2006. SNPD 2006. Seventh ACIS International Conference on. — IEEE. 2006. — P. 109–114.

[27] Ahmim Ahmed, Maglaras Leandros, Ferrag Mohamed Amine, Derdour Makhlouf, Janicke Helge. A novel hierarchical intrusion detection system based on decision tree and rules-based models // 2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS). — IEEE. 2019. — P. 228–233.

[28] Zhang Ming, Xu Boyi, Bai Shuai, Lu Shuaibing, Lin Zhechao. A deep learning method to detect web attacks using a specially designed CNN // International Conference on Neural Information Processing. Springer. 2017. — P. 828–836.

[29] Gupta Abhishek, Jain Ankit, Yadav Samartha, Taneja Harsh. Literature survey on detection of web attacks using machine learning // International Journal of Scientific Research Engineering & Information Technology. 2018. Vol. 3. P. 1845–1853.

ABOUT THE AUTHOR

Nguyen Manh Thang

Information Technology Faculty – Academy of cryptography techniques.

Email: chieumatxcova@gmail.com

Training process: 2005-2007: Student at the Military Technical Academy.

2007-2013: Student at the Applied Mathematics and Informatics Faculty - Lipetsk State Pedagogical University – Russia Federation.

2017-2020: Post-graduate student at the Military Academy of the Federal Guard Service Russian Federation.

Research direction: Computer network, network security, machine learning and data mining.

Truong Phi Ho

Workplace: Telecommunications University

Email: phihosqtt@gmail.com

Education: Master of Sciences

Recent research direction: cyptography, privacy – preserving datamining and machine learning.

Hoang Thanh Nam

Workplace: Academy of Cryptography of Techniques

Email: hoangthanhnam@actvn.edu.vn

Education: received a bachelor's degree in Information Security in 2010 from the Academy of Cryptography Techniques, received a master's degree in Computer Engineering in 2016 from the Hanoi University of Science and Technology,

Recent research direction: Computer Security, Network Security, Malware Analysis.