Algorithm for detecting illegal links using the association rule for improving the web attack detection accuracy of web application firewall

Nguyen Manh Thang

Abstract – Illegal links appear more often through social networks with "dizzying" speed. When users click on a "malicious" link it can bring them potential danger. One of the most popular social networks is Facebook. It is one of the ways for hackers to share malicious links. For example, there are many advertisements with links, and when the user clicks on these links all the information of the user practically falls into the hands of hackers. Hence, a system administrator needs to check requests before running them on the server to ensure security. One of the most common approaches is the Web Application Firewall (WAF). The article presents an algorithm for detecting illegal links based on tf-idf technology for evaluating the "importance" of keywords, symbols in the links of user requests from the user's browser with machine learning method to improve the accuracy of identifying illegal links.

Keywords– illegal links, classification, signature method, anomaly detection method, machine learning method, support vector machine, tf-idf technology.

I. INTRODUCTION

Illegal or malicious links lead the user to phishing sites. These websites steal confidential user information by using the fake informational form, active content, and insertion of symbols in the links [2, 3]. These attacks bring in malicious codes on the client computer and it controls the computer and spreads malicious code to other machines on the same network [4, 5].

Malicious websites resemble trusted websites such as those of banks, government agencies, and e-commerce sites. Malicious code is automatically downloaded from the server of a hacker without the user's permission to generate an attack [6]. This behavior is an important feature of detecting a web attack.

In recent years, many attack detections approaches have been developed. These include the detection of suspicious websites [7], training users [8], fault detection in white list and blacklist, etc. Most web browsers have built-in detection of illegal links based on white and blacklists. There is no high accuracy approach to testing for specialists to check for suspicious links. Moreover, adversaries can use Cross-site scripting performance vulnerabilities (XSS), SQL injections for hacking servers.

The proposed approach detects illegal links based on the machine learning method (Support vector machine – SVM) with tf-idf technology. Most of the existing approach detects malicious links by using tf-idf technology.

The contribution of the proposed approach is as follows:

- detection of illegal links based on the machine learning method and the assessment of the importance of component links;
- improving the accuracy of the classification of links;
- evaluation of the advantages and disadvantages of this approach.

The article is structured as follows. Paragraph 2 describes the mathematical foundations of SVM and tf-idf technology. The architecture of the proposed system is given in paragraph 3. Paragraph 4 is devoted to the algorithm for detecting illegal links using the association rule. Paragraph 5 describes the experimental assessment of the accuracy of the developed algorithm for detecting attacks on web resources**.** Finally, paragraph 6 is a summary of this article.

II. MATHEMATICAL FOUNDATIONS OF TF-IDF AND SUPPORT VECTOR MACHINE

A. Tf-idf Technology

The tf-idf technology is a statistical measure used to assess the importance of a word in the context of a document that is a part of a document collection or corpus. The weight of a certain word is proportional to the number of words used in the document and inversely proportional to the frequency of words in other documents in the collection.

Let's apply the tf-idf technology in our problem; for each request, the author will find words as part of the request. For each word t in request d, in the aggregate requests D, the formulas are used:

$$tfidf(t,d) = tf(t,d) * idf(t) \quad (1)$$

in this formula, the values of tf, idf are calculated as:

$$tf(t,d) = \frac{count(t,d)}{\sum_{v \in d} count(v,d)} \quad (2)$$

v: the remaining words in the request d, and

$$idf(t) = \log \frac{|D|}{|d \in D : t \in d|} \quad (3)$$

After the tf-idf calculation process, the author will convert the request string data into vectors. This is an important step in the learning and detecting process for the continuation of the algorithm since the support vector machine works with numeric values.

B. Support Vector Machine

The support vector machine refers to linear classification methods. Two sets of points belonging to two different classes are separated by a hyperplane in this space. At the same time, the hyperplane is constructed in such a way that the distances from it to the nearest instances of both classes (support vectors) are maximum, which ensures the greatest accuracy of classification.

The concerned study case with precedents $< X, Y, y^*, X^l >$ is considered where X is the space of objects, Y is a set of answers, $y^* : X \to Y$ is target addiction, the values of which are known only on the objects of the training data.

In our problem: $X = R^n, Y = \{-1, +1\}$ the author will build a linear threshold classifier:

$$a(x) = sign(\sum_{j=1}^{n} \omega_j x^j - \omega_0) \quad (4)$$

Where: $x = (x^1, x^2, ..., x^n)$ – inherent description of the object, vector $\omega \in R^n$ and ω_0 – the scalar threshold is called the algorithm parameters.

Expression $\langle \omega, x \rangle = \omega_0$ – hyperplane. The separating hyperplane is as far as possible from the points of both classes close to it. The author needs to maximize the margin between the classes.

For all $x_i \in X^l$:

$$< \omega, x_i > -\omega_0 = \begin{cases} \leq -1, \ if \ y_i = -1 \\ \geq +1, \ if \ y_i = +1 \end{cases} \quad (5)$$

Condition $-1 \leq < \omega, x > -\omega_0 \leq +1$ sets the strip that separates the classes. None of the points in the training set can lie within this strip. The boundaries of the strip are two parallel hyperplanes with the direction of the vector ω. The points closest to the separating hyperplane lie exactly at the boundaries of the strip.

The construction of an optimal separating hyperplane reduces to the problem of minimizing a quadratic form under l-constraints (5) with (n + 1) variables:

$$\begin{cases} < \omega, \omega > \ \to \ \min \\ y_i(< \omega, \omega > -\omega_0) \geq 1, i = 1, 2, ..., l \end{cases} \quad (6)$$

According to the Kuhn-Tucker theorem, such a problem is equivalent dual problem of finding the point in the Lagrange function:

$$\begin{cases} L(\omega, \omega_0, \lambda) = \frac{1}{2} < \omega, \omega > -\sum_{i=1}^{l} \lambda_i (y_i(< \omega, x_i > -\omega_0) - 1) \to \min_{\omega, \omega_0} \max_{\lambda} \\ \lambda_i \geq 0 \\ \lambda_i = 0, if < \omega, x_i > -\omega_0 = y_i \\ i = 1, 2, ..., l \end{cases} \quad (7)$$

Where: $\lambda = (\lambda_1, \lambda_2, ..., \lambda_l)$ – vector of dual variables. To solve this problem, the author will calculate:

$$\frac{\partial L}{\partial \omega} = \omega - \sum_{i=1}^{l} \lambda_i y_i x_i = 0 \Rightarrow \omega = \sum_{i=1}^{l} \lambda_i y_i x_i \quad (8)$$

$$\frac{\partial L}{\partial \omega_0} = -\sum_{i=1}^{l} \lambda_i y_i = 0 \Rightarrow \sum_{i=1}^{l} \lambda_i y_i = 0 \quad (9)$$

Putting (8) and (9) in the formula of the Lagrange function, one gets:

$$\begin{cases} -L(\lambda) = -\sum_{i=1}^{l} \lambda_i + \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \lambda_i \lambda_i y_i y_j (< x_i, x_j >) \to \min_{\lambda} \\ \sum_{i=1}^{l} \lambda_i y_i \geq 0 \end{cases} \quad (10)$$

Then vector ω is calculated by the formula (8). For determining ω_0 one must take a vector x_i and express ω_0 from equality: $\omega_0 = < \omega, x_i > -y_i$. As a result, the classification algorithm can be written in the form:

$$a(x) = sign(\sum_{i=1}^{l} \lambda_i y_i < x_i, x > -\omega_0) \quad (11)$$

With (11) one can define the class of incoming links.

When checking SVM with some datasets one notes the main advantages of the support vector machine [9, 10]:

- SVMs are considered the fastest method for finding decisive functions;
- the SVM has a single solution;
- the SVM finds a dividing strip of maximum width, which makes it possible to further carry out a more confident classification.

The following disadvantages of SVM are noted:

- requirement of high computing power when data size is increased;
- noise sensitivity and data standardization;
- the lack of a general approach to the automatic selection of the kernel (and the construction of a rectifying subspace as a whole) in the case of linear inseparability of classes.

Having understood the mathematical foundations of tf-idf and SVM in paragraph 2, in paragraph 3, the author will consider the advantages and disadvantages of the existing approaches to detect illegal links and the architecture of our proposed system.

III. THE ARCHITECTURE OF THE PROPOSED SYSTEM

There are two main approaches to the detection of attacks from the network data: detection based on signatures and anomaly detection [11 – 13]. The signature method allows us to find attacks by searching for predefined attack signatures. Since this approach is usually accurate, it is successfully used in the detection of intrusions. However, the disadvantage of this approach is that it cannot detect attacks for which it was not programmed. Therefore, it is prone to ignoring all the new types of attacks if the system is not updated with the latest signatures.

The anomaly detection approach examines the normal behavior of users and detects an attack, observing patterns that deviate from the established norms. Thus, systems using an anomaly detection method can detect new attacks. However, the number of false warnings is likely to increase, because not all anomalies are intrusions.

Each method has both advantages and disadvantages, so our approach uses two methods (Fig. 1). The first, WAF uses a signature method to determine known

illegal payload links. And then, the machine learning method is used in identifying illegal links for all other suspicious links.

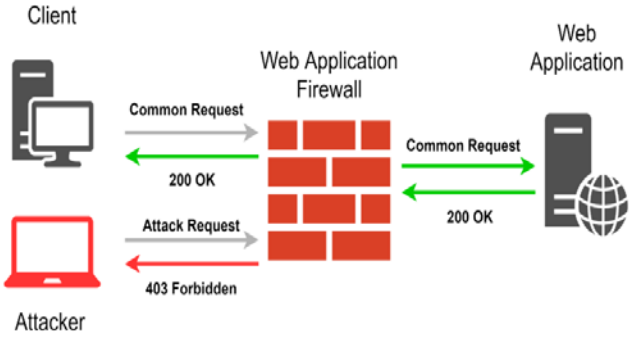

Figure 1. Location of WAF in the system

Our approach is based on detecting illegal links as part of HTTP requests registered by most common web servers (for example, Apache). All requests sent from a web browser to a web server will be checked by the web application firewall. WAF [14] analyzes all requests and decides whether or not to execute requests on the server.

Signature method

According to the requests received from the client, the author will compare payloads, which are stored in the database with payloads as part of the requests sent from the user's browser.

Anomaly Detection Method

The author used the support vector method to determine the threshold for checking a given dataset. After checking with the help of the SVM [15–17], if the link belongs to the class of illegal links, it will be blocked. Otherwise, the link will be executed on the server.

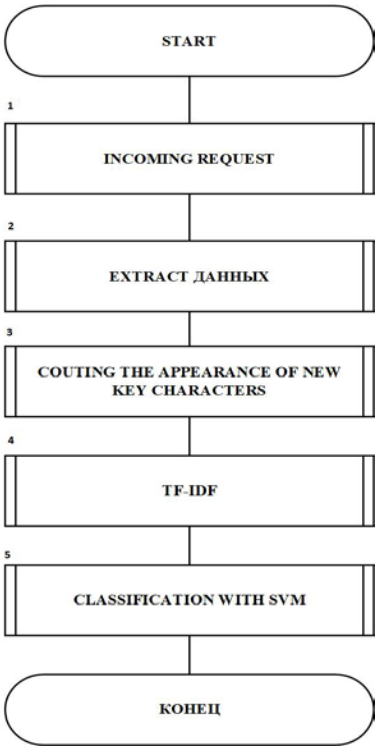

Figure 2. Architectural diagram

IV. ALGORITHM FOR DETECTING ILLEGAL LINKS USING THE ASSOCIATION RULE

In [18], the author used the Support Vector Machine method for checking illegal links, and his approach yields an accuracy result of approximately 98% of positive responses. Therefore, the author proposes an association rule using SVM and tf-idf technology to improve the accuracy of detecting illegal links.

The algorithm consists of 2 phases: the training phase and the detection phase. At this point, the author considers the performance of each phase of the algorithm.

The training phase consists of 3 modules:

- **Extraction module:** upon requests received from the browser of the client, the author will filter the necessary parts for vector space module, including links, payloads, keyword symbols. The author calculated the appearance of new key characters and saved them in the database.
- **Vector space module**: used to transform string data in vector. Using tf-idf technology author can assess the importance of the word and keyword symbols in the string.
- **Data Processing Module:** the author used SVM for classification of links to a given dataset and evaluated the accuracy of illegal links $F1_{score}$.

Table 1. Key symbols used in paragraph 4

Symbols	Value
Ω	Feature class
$F1_{score}$	Accuracy F– measure
Q	List of request strings
yBad	Vector of illegal links
yGood	Vector of legal links
y	Sum of vectors yBad, yGood
X	Links vector after conversion
X_{train}	Vector X for training
y_{train}	Vector y for training
X_{test}	Vector X for testing
y_{test}	Vector y for testing

Before the training phase, in the process of collecting data, the author with specialists defines data classes. All data belongs to two classes either "0" (not attack) or "1" (attack). When analyzing this dataset before running the web application firewall, all payloads of illegal requests will be stored.

Algorithm of training phase

Algorithm 1 Process working of training phase

Input: y, Q

Output: $F1_{score}$

1: Data extraction: links and payloads of requests.

2: Saving payloads and keyword symbols as part of the links.

3: Calculating the appearance of new key characters.

4: Transform data (words and keyword symbols) into vectors for all links using tf-idf technology.

5: Separation of data: data for training 80% (X_{train}, y_{train}) and data for testing 20% (Xtest, y_{test}).

6: Training data using the support vector machine.

7: Calculating the $F1_{score}$ after applying the SVM.

In the training phase of a new data, after the data exaction phase, all payloads of illegal requests will be updated in the database for the detection phase in the comparison module.

The detection phase consists of 4 modules (comparison module and 3 modules as in the training phase). After the extraction module has been run, all request data have been sent to the comparison module. This module will compare the payload of each incoming request with the payload of illegal requests that are stored in the database. If the payload is found, this link will be blocked. Otherwise, all the data of the other links will be sent to the vector space module.

Algorithm of detection phase

Algorithm 2 Process operation in the detecting phase

Input: request x

Output: 1 (illegal) or –1 (legal)

1: Extracting data from request ***x***: links and payload of request ***x***.

2: Comparing payload of request ***x***. If payload is found in the database, the link is blocked. Otherwise, the algorithm proceeds to step 3.

3: Calculating the appearance of new key characters in ***x***.

3: Conversing data (words and keyword symbols) ***x*** into vectors $\vec{x}$ by tf-idf technology.

4: Calculating the result by the SVM method.

V. EXPERIMENTAL ASSESSMENT OF THE ACCURACY OF THE DEVELOPED ALGORITHM FOR DETECTING ATTACKS ON WEB-RESOURCES

To make a comparison, the author evaluates the performance of the algorithm based on the precision, recall, and F–measure. Precision means the percentage of your results that are relevant. On the other hand, recall refers to the percentage of total relevant results correctly classified by your algorithm. Sometimes they are used by themselves, sometimes as a basis for derived metrics, such as the F–measure.

The author denotes some symbols such as TP – true positive; TN – true negative; FP –is false positive; FN – false negative. Then, precision and recall are defined as follows:

$$precision = \frac{TP}{TP + FP} \quad (12)$$

$$recall = \frac{TP}{TP + FN} \quad (13)$$

It is clear that the higher the precision and recall, the better. But in real life, maximum precision and recall are not achievable at the same time and one has to look for some balance. F–measure is the harmonic average between precision and recall. It tends to zero if precision or recall tends to zero.

$$F = 2 \bullet \frac{precision \times recall}{precision + recall} \quad (14)$$

In our work, the author used a dataset that consists of 4,000 illegal links of some types of attacks (XSS, SQL–injection) and 10,000 legal links. The dataset from several data sources of system protection tools such as log files of an intrusion detection and prevention system, HTTP requests (GET, POST method) of a web application firewall of Academy Cryptography techniques Viet Nam and dataset CSIC 2010 will be used. The author has separated our datasets such as 80% of the links for training and 20% of the links for testing. In this paper, the author tested some machine learning methods to compare methods that produce the best result.

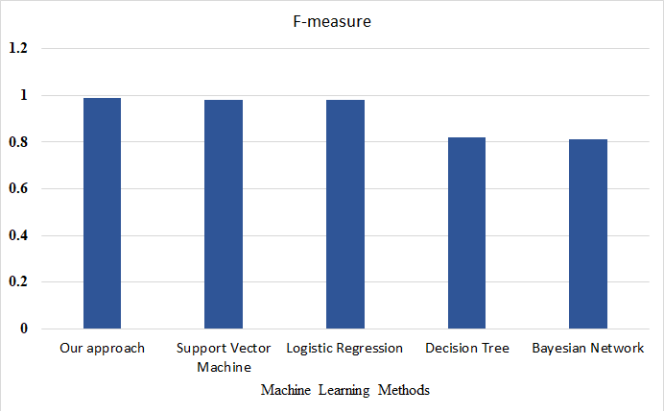

Figure 3. The comparison of accuracy of the F–measure detection of illegal links with some methods of machine learning.

The result obtained is 98.9261%: its better than that shown in the work [18]: it was 98%. Using the tf-idf technology (3–gram and the assessment of the importance of keyword symbols) with the support vectors method gives the best result of some methods considered (since our problem has 2 classes: legal reference and illegal reference). But support vector machine requires high computing power and large dictionaries, so it takes a lot of time to learn when increasing the size of a dataset.

VI. CONCLUSION

Web attacks are a complex issue for web users. Detecting illegal or malicious links is a difficult problem. The proposed classification algorithm for detecting illegal links is based on the application of the machine learning method with tf-idf technology. The illegal link detection algorithm analyzes the links in a sequence of steps to determine if the link is genuine or malicious. Although the proposed algorithm improves the classification accuracy of illegal links, but with an increase in the number of parameters contained in the requests, the classification accuracy will decrease. Therefore, in future work, the author needs to find a combination of anomaly detection methods to improve the classification accuracy not only of new suspicious links but also of the new types of attacks.

REFERENCES

[1] Han Byeong Woo, Yoon Ji Won. Illegal and Harmful Information Detection Technique Using Combination of Search Words // Journal of the Korea Institute of Information Security and Cryptology. – 2016. – Vol. 26, no. 2. – P. 397–404.

[2] Sampat Hemali, Saharkar Manisha, Pandey Ajay, Lopes Hezal. Detection of Phishing Website Using Machine Learning. – 2018.

[3] Gupta Abhishek, Jain Ankit, Yadav Samartha, Taneja Harsh. Literature Survey on Detection of Web Attacks Using Machine Learning. – 2018.

[4] Kim Tae Ghyoon, Choi Young Han, Choi Seok Jin, Lee Cheol Won. System and method for detecting malicious script. – 5 12/2015. – US Patent 9,032,516.

[5] Li Zhi–Yong, Tao Ran, Cai Zhen–He, Zhang Hao. A web page malicious code detect approach based on script execution // Natural Computation, 2009. ICNC'09. Fifth International Conference on. Vol. 6. – IEEE. 2009. – P. 308–312.

[6] Chitra S, Jayanthan K, Preetha S, Shankar RN Uma. Predicate based algorithm for malicious web page detection using genetic fuzzy systems and support vector machine // International Journal of Computer Applications. – 2012. – Vol. 40, no. 10. – P. 13–19.

[7] Shahriar Hossain, Zulkernine Mohammad. Trustworthiness testing of phishing websites: A behavior model–based approach // Future Generation Computer Systems. – 2012. – Vol. 28, no. 8. – P. 1258–1271.

[8] Irani Danesh, Webb Steve, Giffin Jonathon, Pu Calton. Evolutionary study of phishing // ECrime Researchers Summit, 2008. – IEEE. 2008. – P. 1–10.

[9] Lifshits Yuri. Support Vector Method // URL: http://logic.pdmi.ras.ru/~ yura/internet/07ia.pdf. - 2006.

[10] Vorontsov KV. Lectures on the support vector method // Computing Center of the Russian Academy of Sciences, Moscow: URL: http://www.ccas.ru/voron/download/SVM.pdf (accessed: 03.03.12). - 2007.

[11] Stasyuk AI, Korchenko AA. A method for identifying anomalies generated by cyberattacks in computer networks // Zahist shformatsl. - 2012. - T. 4, No. 57. - S. 127–132.

[12] Golovko VA, Bezobrazov SV. Design intelligent anomaly detection systems. - 2011.

[13] Petrenko Sergey Anatolyevich. Methods for detecting intrusions and anomalies in the functioning of cybersystems // Transactions of the Institute for System Analysis of the Russian Academy of Sciences. - 2009. - T. 41. - S. 194–202.

[14] Appelt Dennis, Nguyen Cu D, Panichella Annibale, Briand Lionel C. A machine learning driven evolutionary approach for testing web application firewalls // IEEE Transactions on Reliability. – 2018. – Vol. 67, no. 3 – P. 733–757.

[15] Ma Justin, Saul Lawrence K, Savage Stefan, Voelker Geoffrey M. Beyond blacklists: learning to detect malicious web sites from suspicious URLs // Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. – ACM. 2009. – P. 1245–1254.

[16] Basnet Ram, Mukkamala Srinivas, Sung Andrew H. Detection of phishing attacks: A machine learning approach // Soft Computing Applications in Industry. – Springer, 2008. – P. 373–383.

[17] Sahoo Doyen, Liu Chenghao, Hoi Steven CH. Malicious URL detection using machine learning: A survey // arXiv preprint arXiv:1701.07179. – 2017.

[18] Yadav BV Ram Naresh, Satyanarayana B, Vasumathi D. A Vector Space Model Approach for Web Attack Classification Using Machine Learning Technique // Proceedings of the Second International Conference on Computer and Communication Technologies. – Springer. 2016. – P. 363–373.

Nguyen Manh Thang, Contributor, Academy of Cryptography Techniques,10000, House. 141, Chien Thang, Tan Trieu, Thanh Tri, Ha Noi. Tel.: +79202862939.
E-mail: chieumatxcova@hotmail.com