Algorithm for detecting attacks on Web applications based on machine learning methods and attributes queries

Nguyen Manh Thang, Tran Thi Luong

***Abstract*—Almost developed applications tend to become as accessible as possible to the user on the Internet. Different applications often store their data in cyberspace for more effective work and entertainment, such as Google Docs, emails, cloud storage, maps, weather, news,... Attacks on Web resources most often occur at the application level, in the form of HTTP/HTTPS-requests to the site, where traditional firewalls have limited capabilities for analysis and detection attacks. To protect Web resources from attacks at the application level, there are special tools - Web Application Firewall (WAF). This article presents an anomaly detection algorithm, and how it works in the open-source web application firewall ModSecurity, which uses machine learning methods with 8 suggested features to detect attacks on web applications.**

***Tóm tắt*—Hầu hết các ứng dụng được phát triển có xu hướng trở nên dễ tiếp cận nhất có thể đối với người dùng qua Internet. Các ứng dụng khác nhau thường lưu trữ dữ liệu trên không gian mạng để làm việc và giải trí hiệu quả hơn, chẳng hạn như Google Docs, email, lưu trữ đám mây, bản đồ, thời tiết, tin tức,... Các cuộc tấn công vào tài nguyên Web thường xảy ra nhất ở tầng ứng dụng, dưới dạng các yêu cầu HTTP/HTTPS đến trang web, nơi tường lửa truyền thống có khả năng hạn chế trong việc phân tích và phát hiện các cuộc tấn công. Để bảo vệ tài nguyên Web khỏi các cuộc tấn công ở tầng ứng dụng, xuất hiện các công cụ đặc biệt - Tường lửa Ứng dụng Web (WAF). Bài viết này trình bày thuật toán phát hiện bất thường và cách thức hoạt động của tường lửa ứng dụng web mã nguồn mở ModSecurity khi sử dụng phương pháp học máy với 8 đặc trưng được đề xuất để phát hiện các cuộc tấn công vào các ứng dụng web.**

***Keywords*—web attack, network security, signature method, anomaly detection method, machine learning method, Web application firewall, ModSecurity.**

***Từ khóa*—Tấn công web, an toàn mạng, phương pháp chữ ký, phương pháp phát hiện bất thường, phương pháp học máy, tường lửa ứng dụng Web, ModSecurity.**

I.INTRODUCTION

The development of information technologies in recent decades has led to the rapid growth of information and telecommunication systems that allow the transmission, processing, and storage of heterogeneous information. At the same time, with the increasing of such systems, the number of information security incidents has also increased. Indeed, according to the InfoWatch analytical center, 2,263 events of confidential information leaks were registered in 2018, which is 6% more than in 2017 [1]. The main area of an information security breach that resulted in confidential data leakage is the actions of external violators (more than 52% of cases). Due to the fault of external violators, 59.9% of the data from the total volume of records were compromised. In the distribution of leaks of information security violations, the main direction of attacks on Web resources (Web applications), which account for a third of the total number of attacks [2]. At the same time, this indicator in 2018 increased by 2% compared to the same period in 2017.

Currently, to protect the area of a computer network (including Web applications) from external attacks, Web Application Firewall (WAF) tools are widely used. Most of the existing WAFs are based on approaches used in intrusion detection systems. These systems can resist attacks on network components that carry out network interaction at various levels (Fig. 1).

This manuscript is received on November 23, 2020. It is commented on December 09, 2020 and accepted on December 22, 2020 by the first reviewer. It is commented on December 09, 2020 and accepted on December 23, 2020 by the second reviewer.

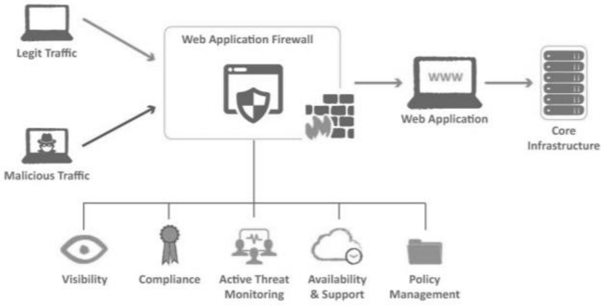


Fig. 1. Position WAF in Web application security system.

Thus, according to the Statista information center, in 2018 the average number of blocked attacks on computer networks increased 2.8 times compared to 2015 and amounted to 953.8 thousand attacks per day [3]. At the same time, the number of attacks successfully implemented through Web applications continues to grow. This feature is due to the presence of shortcomings caused by errors in the design, development, and creation of Web applications. In view of this, the task of increasing the security of Web applications from external attacks is an urgent line of research. To solve this problem, it is necessary to develop new methods of protection. One of these methods can be an algorithm for detecting computer attacks on Web resources, based on the application of machine learning methods using a new feature space. To develop this algorithm, an analysis of current research in the field of detecting attacks on Web applications was carried out.

II. Overview of existing research on detecting attacks on Web applications

Existing intrusion detection systems can be divided into the following groups of approaches: based on the use of signature methods [4-6], anomaly detection methods [7-11].

Signature attack detection methods search for signatures in the analyzed traffic or event logs. A signature is a pattern that matches an attack scenario or an existing vulnerability. Signature detection methods are highly accurate in detecting known attacks but are ineffective against new and unknown types of threats. For example, the methods of this group are not able to detect attacks in which the methods of obfuscation and data hiding are used. In addition, an up-to-date threat signature database is required to detect attacks.

Unlike signature detection methods, anomaly detection methods can detect new and previously unknown types of attacks. In the process of detecting attacks, it searches for deviations in the analyzed traffic or analyzed events. At the same time, models of permitted behavior of subjects (behavior profiles) are determined in the detection system. Profiles are developed by monitoring the characteristics of the typical activity over a period of time. In anomaly detection methods, testing and training datasets are used to simulate traffic that is considered legitimate in a network environment. For the functioning of the attack detection system based on the detection of anomalies, it is necessary to develop a criterion for distinguishing the normal behaviors of subjects from anomalous ones. If the behavior deviates from normal by more than a certain threshold value, the system notifies this deviation. Training datasets are also used to simulate malicious traffic so that the system can recognize patterns of known threats and attacks.

An important feature of the tasks of detecting atypical system behavior and detecting anomalies is the absence of a formal definition of an anomaly. Often it is formalized in the course of the research, depending on the chosen method and feature space. For complex systems, machine learning and other data mining methods are also used to solve the problem of detecting anomalies. Machine learning [12], as a branch of artificial intelligence, is used both in the detection of anomalies and in the detection of abuse. This is explained by the fact that these approaches often use patterns of both normal and abnormal behavior of subjects as initial data for training (Bayesian networks [13-16], Neural networks [17,18], K-nearest neighbors [19,20], Decision trees [22,24,25], Support vector machine [22,24]...).

Each method has its advantages and disadvantages, so it is necessary to combine methods to limit the "minus" of each method, thereby increasing the accuracy of the attack detection algorithm.

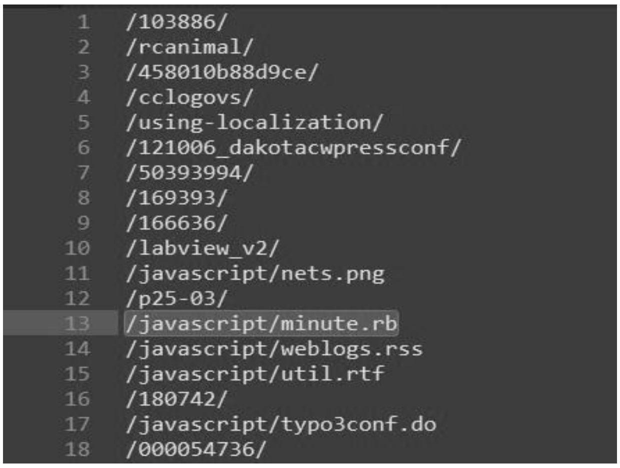
III. Algorithm for detecting attacks on web applications

The algorithm consists of two phases: a training phase and an attack detection phase. The learning phase consists of four modules. All steps are presented as follows:

- Extraction module: upon requests received from the client, authors will filter the parts needed to process requests, including links, session IDs;
- Attribute analysis module: this module is used to calculate the required values for the attributes used in the attack detection phase;
- Vector space module: used to convert string data to vector. Using tf-idf technology, authors can evaluate the importance of a word in a string;
- Data processing module: the authors use machine learning methods to determine the threshold of each method when validating a given dataset. After that, all thresholds will be saved in the database.

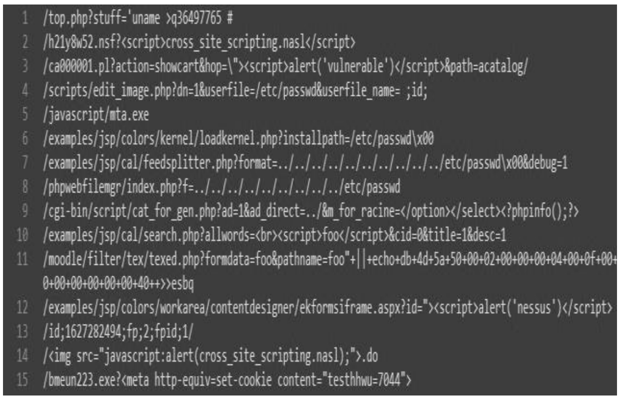
During training, the authors used 8 attributes: the length of the field sent from the browser (A1); the distribution of characters in the request (A2); token finder (A3); field absence (A4); the order of fields sent from the browser (A5); the length of the request sent from the browser (A6); the appearance of new special symbols (A7); the emergence of new keywords (A8). Five attributes [A1-A5], which are presented in [22].

When studying data sets used to learn and detect network attacks, the authors found that between abnormal URLs and normal URLs, there is a difference in URL length or anomalous signs (from special keys, special character sets). These authors consider these anomalies as a sign to detect a cyber-attack. The authors suggest 3 additional attributes [A6-A8]. After the work of the attribute analysis module, all thresholds will be saved in the database.



```
/103886/
/rcanimal/
/458010b88d9ce/
/cclogovs/
/using-localization/
/121006_dakotacwpressconf/
/50393994/
/169393/
/166636/
/labview_v2/
/javascript/nets.png
/p25-03/
/javascript/minute.rb
/javascript/weblogs.rss
/javascript/util.rtf
/180742/
/javascript/typo3conf.do
/000054736/
```

Fig. 2. Normal HTTP requests.



```
/top.php?stuff='uname >q36497765 #
/h21y8w52.nsf?<script>cross_site_scripting.nasl</script>
/ca000001.pl?action=showcart&hop=\"><script>alert('vulnerable')</script>&path=acatalog/
/scripts/edit_image.php?dn=1&userfile=/etc/passwd&userfile_name= ;id;
/javascript/mta.exe
/examples/jsp/colors/kernel/loadkernel.php?installpath=/etc/passwd\x00
/examples/jsp/cal/feedsplitter.php?format=../../../../../../../../../../etc/passwd\x00&debug=1
/phpwebfilemgr/index.php?f=../../../../../../../../../etc/passwd
/cgi-bin/script/cat_for_gen.php?ad=1&ad_direct=../&m_for_racine=</option></select><?phpinfo();?>
/examples/jsp/cal/search.php?allwords=<br><script>foo</script>&cid=0&title=1&desc=1
/moodle/filter/tex/texed.php?formdata=foo&pathname=foo"+||+echo+db+4d+5a+50+00+02+00+00+00+04+00+0f+00+
0+00+00+00+00+00+40++>>esbq
/examples/jsp/colors/workarea/contentdesigner/ekformsiframe.aspx?id="><script>alert('nessus')</script>
/id;1627282494;fp;2;fpid;1/
/<img src="javascript:alert(cross_site_scripting.nasl);">.do
/bmeun223.exe?<meta http-equiv=set-cookie content="testhhwu=7044">
```

Fig. 3. Abnormal HTTP requests.

The algorithm for calculating the attributes A6 to A8 is shown in Fig. 4.

First, the authors create three empty arrays to store the values of the Ksymbols, Kwords, and Lens query lengths (step 1). After creating the required arrays, we calculated the values A6 to A8 for all received requests (step 2).

Secondly, in order to find the appearance of keywords, we need to select the received queries by words and compare these words with the existing keywords in the Kwords array. If the word does not exist in this array, then we need to store it in Kwords (step 3 - step 7). The authors follow the same procedure for the process of updating the Ksymbols array (step 8 through step 11).

Using these procedures, the authors will get three arrays of Ksymbols, Kwords, and Lens query lengths. After defining the three arrays, the authors will calculate the values from A6 to A8. These values are the initial data for the features used in the attack detection phase.

In the algorithm, the authors use tf-idf technology to evaluate the importance of a word in each request. The tf-idf technology algorithm for query words is shown in Fig. 5. The algorithm consists of the following steps:

- Step 1–2: Create empty Kwords and Count arrays to store the words of each query and the number of times each word appears in the query. Vectors of words from received queries are generated.
- Step 3-6: Calculate the number of times each word appears in the query. All received values will be stored in the Count array.
- Step 7-13: for each word kws_j in the set of words of the $ReqW_i$ query we need to perform the procedure for calculating the vector tf-idf Req by the formula:

$$tf(kws_j) = \frac{count_j}{len(ReqW_i)} \quad (2)$$

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

where D is the corpus of requests.

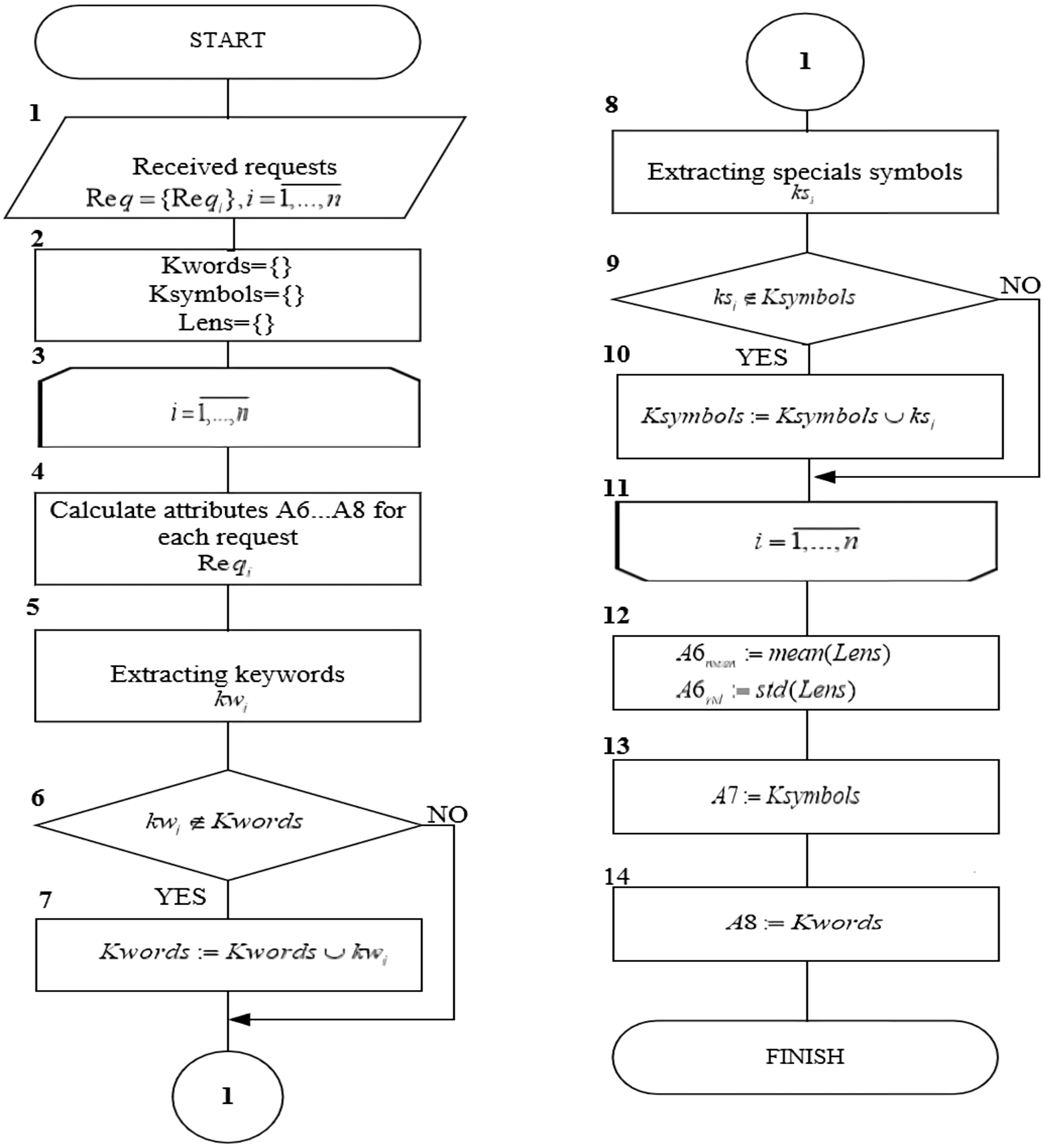


Fig. 4. The process of calculating A6-A8.

Then the tf-idf value for the received requests is represented as:

$$tfidf\,(Req_i) = tf\,(Req_i).idf\,(Req_i) \qquad (4)$$

After applying the tf-idf technology for all queries, the authors receive the set of vectors $\overrightarrow{Req}$ and $\overrightarrow{Req}$ is the input data of the detection phase, which is presented below.

The regsi query attack detection algorithm consists of several stages.

First, for each request, the WAF extracts the kwi keywords and the ksi. Once extracted, these words and symbols will be compared against the existing keywords and symbols in the Ksymbols and Kwords array. If they do not exist in these arrays, then we need to save them for the next step.

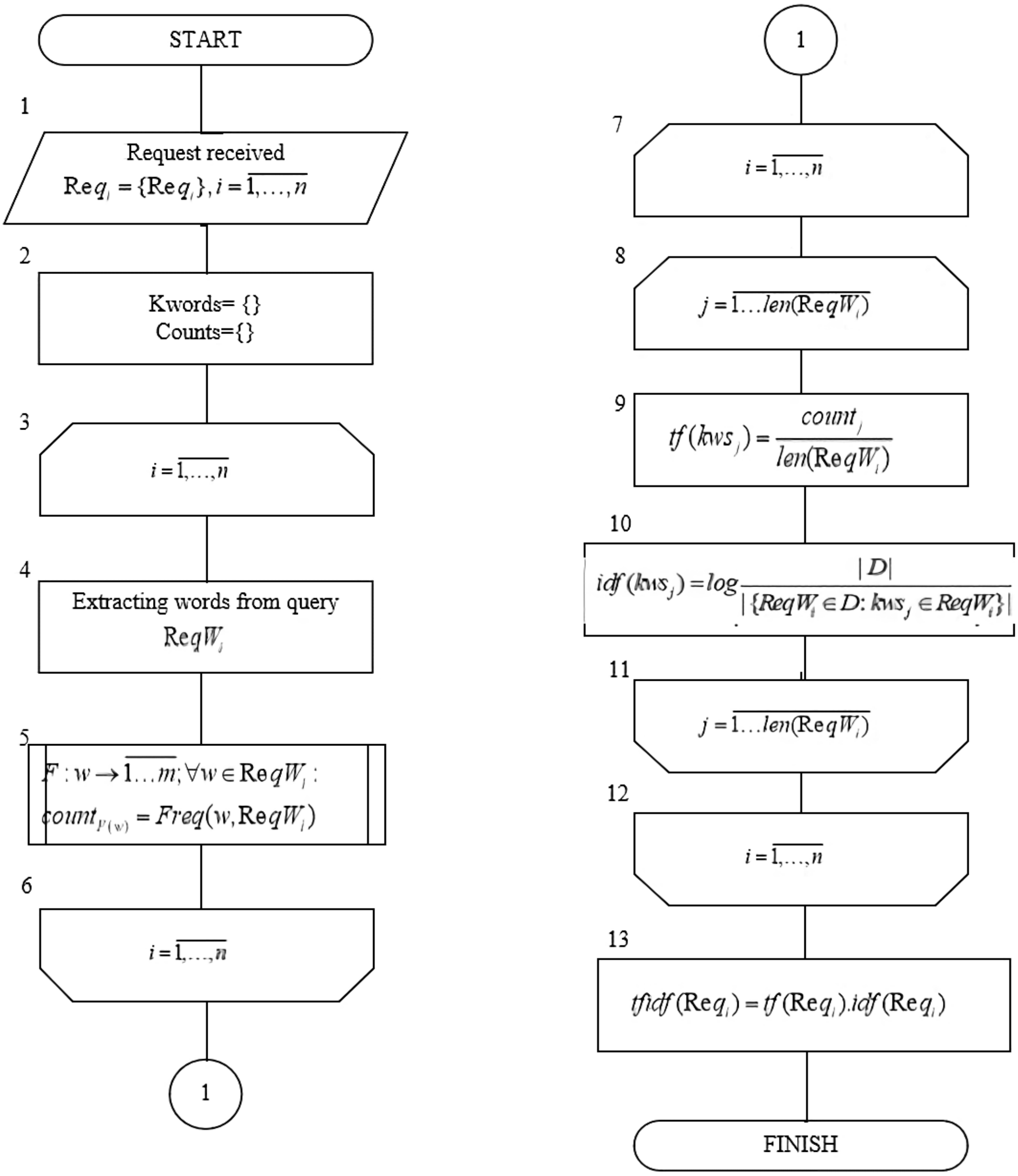

START
1
Request received
$\mathrm{Re}q_i = \{\mathrm{Re}q_i\}, i = \overline{1,\ldots,n}$
2
Kwords= {}
Counts={}
3
$i = \overline{1,\ldots,n}$
4
Extracting words from query
$\mathrm{Re}qW_i$
5
$F : w \to \overline{1\ldots m}; \forall w \in \mathrm{Re}qW_i :$
$count_{F(w)} = Freq(w, \mathrm{Re}qW_i)$
6
$i = \overline{1,\ldots,n}$
1
1
7
$i = \overline{1,\ldots,n}$
8
$j = \overline{1\ldots len(\mathrm{Re}qW_i)}$
9
$tf(kws_j) = \frac{count_j}{len(\mathrm{Re}qW_i)}$
10
$idf(kws_j) = log\frac{|D|}{|\{ReqW_i \in D: kws_j \in ReqW_i\}|}$
11
$j = \overline{1\ldots len(\mathrm{Re}qW_i)}$
12
$i = \overline{1,\ldots,n}$
13
$tfidf(\mathrm{Re}q_i) = tf(\mathrm{Re}q_i).idf(\mathrm{Re}q_i)$
FINISH

Fig. 5. Workflow tf-idf technology for query sets.

Secondly, WAF calculates A1 to A8 values and converts the data into vectors using tf-idf technology (Fig. 5).

Third, the WAF detects machine learning attacks. The values after detection by machine learning methods and analysis of request attributes are given to us only 1 or 0. If the result of the decision module returns to the value 1, then the algorithm will end and the request will be blocked. Otherwise, the result will return to 0 and the request will be executed on the server.

IV. EXPERIMENTAL EVALUATION OF THE ALGORITHM

In this paper, the authors used the CSIC 2010 dataset, which is specialized in testing web application firewall solutions, proposed by the Institute for Information Security of the Spanish National Research Council.

CSIC 2010 includes 36,000 secure queries and 25,000 attacks of many types: SQL injection, OS command execution,... The authors highlighted a dataset such as 80% of queries for training and 20% of queries for testing.

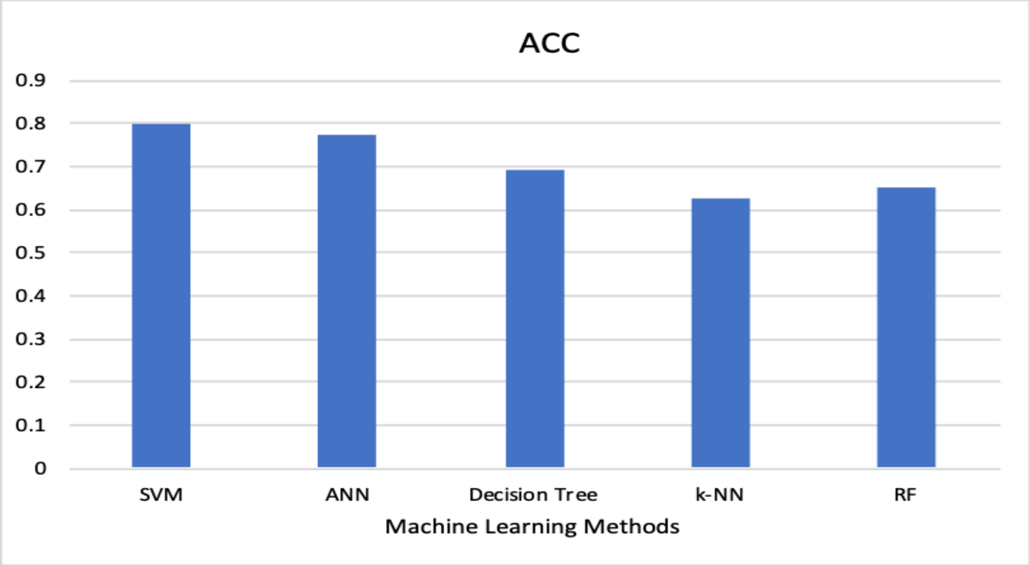


Fig. 6. The result of detection when using the tf-idf technology without considering the attributes.

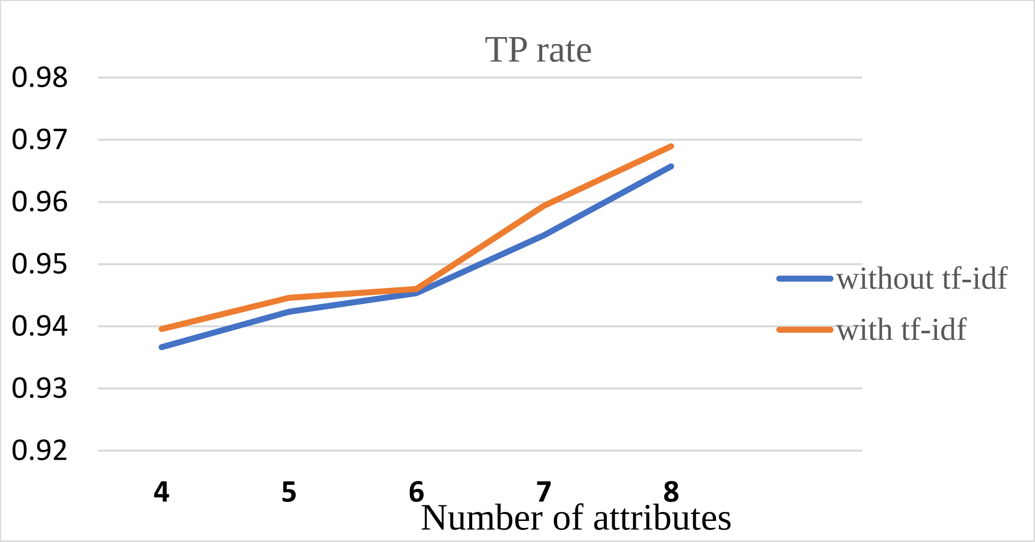


Fig. 7. Result of detection by number of attributes.

When applying the tf-idf technology without considering the attributes, the authors will receive the result shown in Fig. 6. (ACC: Accuracy, SVM: support vector machine, ANN: neural network, RF: random forest, k-NN: k-nearest neighbors).

With an increase in the number of attributes, the authors will get the result shown in Fig. 7.

V. Discussion

The research works to detect attacks on the application layer specifically with code injection attacks studied by many scientists around the world. Especially when using machine learning algorithms in the task of attack detection we realize that each algorithm has its advantages (good results) over several specific datasets. There is no one best algorithm for all datasets used for training and detecting. In this case, the author has also compared the results with the best results from other papers with CSIC 2010 dataset (see Table 1).

The author has chosen metrics TPR and FPR for comparison.

TABLE 1. Comparison of our results with other researches on CSIC 2010

Detection method	TPR	FPR
Nguyen [26]-C4.5	94.49%	5.9%
Kozik-ELM[a] [27]	94.98%	0.79%
Kozik-REPTree[b] [28]	98%	1.5%
Loffler [29]-RF[c]+SVM	96.27%	14.38%
Soltes [30]-DDCA[d]	87.88%	12.71%
Our results	98.76%	3.2%

[a] Extreme Learning Machine Forest

[b] Reduced Error Pruning Tree

[c] Random Forest

[d] Deterministic Dendritic cell Algorithm

Conclusion

Web attacks must be handled with techniques that account for signature-based detection accuracy with the flexibility of an anomaly-based intrusion detection system.

This article introduces a new approach to anomaly detection by using HTTP requests containing parameters as input.

First, the training set of HTTP requests is analyzed, which does not contain any attacks. After all the necessary information has been extracted from the logs, several anomaly detection schemes are applied to describe the behavior of the average user. This model is then used to detect network attacks as abnormalities. The authors use a combination of machine learning anomaly detection techniques and add three signs to improve the accuracy of the network attack detection model.

Future work will focus on further reducing the number of false positives by refining the developed algorithms and exploring additional possibilities. The ultimate goal is to be able to detect anomalies in real-time for websites that handle millions of requests per day with virtually no false positives.

Acknowledgement

We would like to express my sincere thanks to the Doctor of science Kozachok Alexander Vasilyevich - Military Academy of the Federal Guard Service Russian Federation for helping me to complete this research paper.

References

[1] Аналитический центр InfoWatch. Глобальное исследование утечек конфиденциальной информации в 2018 году. – 2019. URL: https://www.infowatch.ru/resources/report2018 (дата обращение 03.07.2018).

[2] Ростелеком Solar. Solar JSOC Security Report 2018 и тренды 2019. 2019. URL: https://rt-solar.ru/analytics/reports/ (дата обращение 03.07.2018).

[3] Statista. Global number of web attacks blocked per day from 2015 to 2018 (in 1,000s). 2019. URL: https://www.statista.com/statistics/494961/web-attacks-blocked-per-day-worldwide/ (дата обращение 03.07.2018)

[4] Modi, C. A survey of intrusion detection techniques in cloud / C. Modi, D. Patel, B. Borisaniya [et al.] // Journal of Network and Computer Applications. – 2013. – Vol. 36, no. 1. – P. 42–57.

[5] Khamphakdee, N. Improving intrusion detection system based on snort rules for network probe attack detection / N. Khamphakdee, N. Benjamas, S. Saiyod // Information and Communication Technology (ICoICT), 2014 2nd International Conference On. – IEEE. 2014. – P. 69–74.

[6] Scarfone, K. A., Mell, M. Guide to Intrusion Detection and Prevention Systems (IDPS)| NIST. –2007. URL: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-94.pdf (дата обращение 03.07.2018).

[7] Vigna, G. A stateful intrusion detection system for world-wide web servers / G. Vigna, W. Robertson, V. Kher, [и др.] // Computer Security Applications Conference, 2003. Proceedings. 19th Annual. – IEEE. 2003. – P. 34–43.

[8] Sekar, R. An Efficient Black-box Technique for DefeatingWeb Application Attacks. NDSS. – 2009.

[9] Mutz, D. An experience developing an IDS stimulator for the blackbox testing of network intrusion detection systems / D. Mutz, G. Vigna, R. Kemmerer // Computer Security Applications Conference, 2003. Proceedings. 19th Annual. – IEEE. 2003. – P. 374–383.

[10] Li, X. BLOCK: a black-box approach for detection of state violation attacks towards web applications / X. Li, Y. Xue // Proceedings of the 27th Annual Computer Security Applications Conference. – ACM. 2011. – P. 247–256.

[11] Saxena, P. Efficient fine-grained binary instrumentationwith applications to taint-tracking / P. Saxena, R. Sekar, V. Puranik // Proceedings of the 6th annual IEEE/ACM international symposium on Code generation and optimization. – ACM. 2008. – P. 74–83.

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

[13] Heckerman, D. A tutorial on learning with Bayesian networks. D. Heckerman. Innovations in Bayesian networks. – Springer, 2008. – P. 33–82.

[14] Friedman, N. Bayesian network classifiers / N. Friedman, D. Geiger, M. Goldszmidt // Machine learning. – 1997. – Vol. 29, no. 2–3. – P. 131–163.

[15] Goldszmidt, M. Bayesian network classifiers // Wiley Encyclopedia of Operations Research and Management Science. – 2010.

[16] Barbara, D. Detecting novel network intrusions using Bayes estimators / D. Barbara, N. Wu, S. Jajodia // Proceedings of the 2001 SIAM International Conference on Data Mining. – SIAM. 2001. – P. 1–17.

[17] Емельянова, Ю. Г. Нейросетевая технология обнаружения сетевых атак на информационные ресурсы / Ю. Г. Емельянова, А. А. Талалаев, И. П. Тищенко [и др.] // Программные системы: теория и приложения. – 2011. – Т. 2, № 3. – С. 3–15.

[18] Tavallaee, M. A Detailed Analysis of the KDD CUP 99 Data Set / M. Tavallaee, E. Bagheri, W. Lu [и др.] // Proceedings of the Second IEEE International Conference on Computational Intelligence for Security and Defense Applications. – Ottawa, Ontario, Canada: IEEE Press, 2009. – Pp. 53—58. – (CISDA’09). – URL: http://dl.acm.org/citation.cfm?id= 1736481.1736489.

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

[20] Su, M.-Y. Real-time anomaly detection systems for Denial-of-Service attacks by weighted knearest neighbor classifiers / M.-Y. Su // Expert Systems with Applications. – 2011. – Vol. 38, no. 4. – P. 3492–3498.

[21] Lee, C. H. Network intrusion detection through genetic feature selection / C. H. Lee, J. W. Chung, S. W. Shin // Soft-ware Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2006. SNPD 2006. Seventh ACIS International Conference on. – IEEE. 2006. – P. 109–114.

[22] Ireland, E. Intrusion detection with genetic algorithms and fuzzy logic / E. Ireland // UMM CSci senior seminar conference. – 2013. – P. 1–6.

[23] *Kruegel Christopher, Vigna Giovanni.* Anomaly detection of web-based attacks, Proceedings of the 10th ACM conference on Computer and communications security. ACM. 2003. – P. 251–261.

[24] Kruegel, C. Using decision trees to improve signature-based intrusion detection / C. Kruegel, T. Toth // Recent Advances in Intrusion Detection. – Springer. 2003. – P. 173–191.

[25] Bouzida, Y. Neural networks vs. decision trees for intrusion detection. Y. Bouzida, F. Cuppens. IEEE/IST Workshop on Monitoring, Attack Detection and Mitigation (MonAM). Vol. 28. – 2006.

[26] Nguyen, H.T., Torrano-Gimenez, C., Alvarez, G., Petrovic, S., and Franke, K., Application of the generic feature selection measure in detection of web attacks, in Computational Intelligence in Security for Information Systems, Herrero, Á. and Corchado, E., Eds., Berlin, Heidelberg: Springer, 2011.

[27] Kozik, R., Choraś, M., Holubowicz, W., and Renk, R., Extreme learning machines for web layer anomaly detection, in Image Processing and Communications Challenges 8, Choraś, R.S., Ed., Cham: Springer Int. Publ., 2017, pp. 226–233.

[28] Kozik, R. and Choras, M., Adapting an ensemble of one-class classifiers for a web-layer anomaly detection system, Proc. 10th Int. Conf. on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), Krakow, 2015, pp. 724–729.

[29] Loffler, M., Improvement of intrusion detection using multiple classifier model, Diploma Thesis, FIIT STU, 2017.

[30] Šoltes, F., Improving security of a web system using biology inspired methods, Diploma Thesis, FIIT STU, 2016.

[31] Manh Thang Nguyen, Alexander Kozachok. Representation Model of Requests to Web Resources, Based on a Vector Space Model and Attributes of Requests for HTTP Protocol. *Journal of Science and Technology on Information Security*. Vol. 10, No. 2, 2019.

ABOUT THE AUTHORS

Nguyen Manh Thang

Workplace: Information Technology Faculty – Academy of cryptography techniques.

Email: chieumatxcova@gmail.com

Education: Student at the Military Technical Academy in 2005-2007, Student at the Applied Mathematics and Informatics Faculty - Lipetsk State Pedagogical University – Russia Federation in 2007-2013, Post-graduate student at the Military Academy of the Federal Guard Service Russian Federation and got Ph.D. degree in 2020.

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

Tran Thi Luong

Workplace: Academy of cryptography techniques.

Email: luongtranhong@gmail.com

Education: She received Bachelor degree in Mathematics and Informatics from The Hanoi University of Science in 2006, Master and Doctor degree in Cryptographic Techniques from Academy of Cryptographic Techniques in 2012, 2019 respectively.

Research today: Cryptography and Database Security.