


RESEARCH ARTICLE | MAY 06 2024

Using artificial intelligence techniques in web application firewall (WAF) FREE


Dilfuza Makhmudova 




AIP Conf. Proc. 3147, 040021 (2024)

<https://doi.org/10.1063/5.0210409>



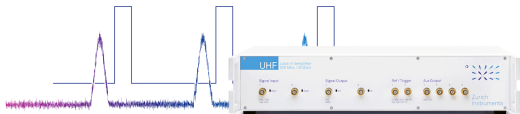


Lock-in Amplifier



Zurich Instruments

Find out more



Boxcar Averager

Boost Your Optics and Photonics Measurements

Using Artificial Intelligence Techniques in Web Application Firewall (WAF)

Dilfuza Makhmudova^{1, a)}

¹Chirchik State Pedagogical University, Tashkent, Uzbekistan

^{a)}Corresponding author: baxtiyorturdibekov04@gmail.com

Abstract. Attacks on web applications and web-based services were carried out using the Hyper-Text Transfer Protocol (HTTP). Web application security has ended up a key necessity for any commerce, despite web application defense measures and the constant development of software and servers, web attacks are becoming more common. Our proposed model analyzes request to the web server, analyzes these require to extract four features that fully describe HTTP request sections and classes if the request is normal or abnormal. The property values for anomaly queries are large query length, small percentage of allowed characters, large proportion of uncommon characters, and very large numerical attack weight. A number of studies highlight the benefits of using Machine Learning to generate new rules for detecting malware and malicious web requests. This work aims to demonstrate a distributed WAF architecture using ML classifiers as one of its components. This architecture has a machine learning classifier to analyze and detect the full HTTP protocol for violations. The first part of this work examines the viability of using classifiers based on metrics such as precision and recall. We analyze two datasets and compare their usage. The fourth part of this article compares the prediction processing time of ML models and the processing time of the rule-based engine.

Keywords. Web application, artificial intelligence, dataset, firewall, HTTP, injection, machine learning, SVM, SQL XSS, web server, WAF, web attack.

INTRODUCTION

Cyber-attacks focusing on web applications and servers have become among the most significant aspects to consider when a company incorporates technology into all of its operations. and despite the diversity of methods to combat them, these attacks remain a high risk. Although web application developers have put protective mechanisms in place, threats are continually changing, necessitating the need of specialized software to assist these security procedures [1]. Security initiatives and guidelines for programmers and ethical hackers OWASP publications aim to enhance security [2]. Web application firewalls engage with web requests at the application level while traditional firewalls interact with packets at the network and transport layers [3]. These firewalls are signed [4] because they identify attacks based on their unique traces, which necessitates the usage of massive databases and the storage of attack traces after each assault is carried out. It is challenging to harness expert knowledge by transferring it to computers due to reliance on databases and hard-coded logic and rules [5].

Technologies such as digital control systems, IoT, cloud computing, and artificial intelligence have opened up new opportunities for humanity, but also created new challenges. With the rapid development of technology, sophisticated attack methods have also been specially developed. Thus, in modern information security systems, most organizations use an intrusion detection system, an intrusion prevention system, and a network firewall to monitor the system as well as detect network-level attacks.

In recent decades, artificial intelligence has become a scientific revolution and has gained an unprecedented advantage in mastering the work that humans do, and we believe that a computer cannot learn and make decisions like humans, but it has instead become a human rival. Researchers and information security professionals are attempting to leverage artificial intelligence capabilities to detect and combat assaults [6-7].

The work done on anomaly detection using machine learning makes many contributions with different approaches. For example: Autoencoder, One-Class SVM, One-class SVM, Elliptic Envelope and Random Forest, CNN, RNN. On

the other hand, there is no equivalent number of performance analyzes of the proposed solutions. In the industry, web application firewalls (WAF) are used to protect web applications from vulnerabilities and other security assaults [8].

The aspect we evaluate in this work is related to the architecture used to implement WAFs. We can predict that implementing a WAF will require increasing the computing power available to applications, as tools will analyze each request before it is routed to the target web application.

WAF AND RELATED WORKS

The concept of web application vulnerabilities does not change; what varies is how they are used. The following are the most common web application vulnerabilities:

- Injections: manipulation of access by forcing a web application to execute commands and queries in databases that exist in the operating system, SQL injection is the most popular injection attack, and to this attacker read, write and allows you to interact with the database by changing.
- Compromised authentication: exploiting logical and weak points in the authentication mechanism to take over and control accounts.
- Exposure of sensitive data: Exclusion of a web application and manipulation to reveal sensitive data such as database credentials.
- XML External Object (XXE): manipulation of inputs using functions that parse XML to execute arbitrary commands.
- Broken access control: access to unauthorized resources in the web application due to weak access control rules, for example, access to the admin panel or restricting access to it.
- Security Misconfigurations: Using brute force to discover and exploit security flaws such as unpatched vulnerabilities, default configurations, useless pages, unsecured files and directories, and superfluous services.
- Cross-site scripting (XSS): inserting JavaScript code into a web application to alter its appearance and force the victim to run it in their browser. There are numerous types, including mirrored XSS and DOM XSS.
- Untrusted Deserialization: Manipulation of web application data by deserialization, modification and reserialization to compromise the web application.
- Use of known vulnerabilities: preventing the updating of a component used in a web application allows attackers to exploit its known flaws; this type of vulnerability is fairly widespread, particularly in CMS online applications.
- Inadequate logging and monitoring refers to a lack of logging and monitoring systems and approaches that allow attackers to identify and exploit them undetected [9].

Two methodologies have been followed for research on web application protection against malicious requests and attack detection: Detection of a specific attack (for example, only detection of SQL injection attack or detection of cross-site scripting attack) or regardless of attack type regardless of whether it is an anomaly or completely normal, classifying requests. It also used two approaches to carry out this experiment on computers: designing and implementing behavior-based detection using artificial intelligence techniques such as classification algorithms or the use of a special algorithm, and signature-based detection using databases containing patterns of attacks. The majority of CSIC 2010 [10] investigations relied on older datasets such as ECML-PKDD 2007. The proposed models have not been tested using recent datasets, and some researchers' datasets are not publicly available [9].

Web application firewall (WAF) [11] is a firewall used for specific web applications. It is placed in front of web applications and analyzes two-way web traffic (HTTP) - detects and blocks all malicious content.

WAF controls package content at the application level [12]. They offer greater security than packet filters, but at the tradeoff of reduced transparency for managed services. WAFs work as both a server and a client for the real server, handling requests from the real server rather than the users they protect. WAF works in one of two modes: passive or active. Active WAFs inspect all incoming requests, identifying vulnerabilities that allow SQL injection, cross-site scripting, and spoofing parameters or cookies.

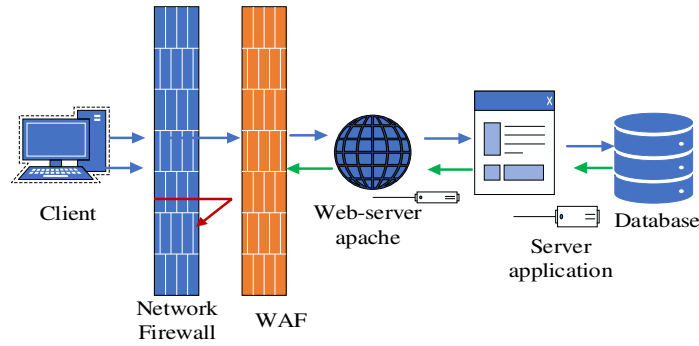


FIGURE 1. The position of web application firewall (WAF) in security

Passive WAFs work on the principle of an intrusion detection system: they also scan all incoming requests, but do not block them when a potential cyberattack is detected. When using a WAF, all connections go through it. The connection begins on the client system and enters the firewall interface, as shown in Fig.1. The firewall receives the connection, examines the packet content and protocol, and assesses whether the traffic complies with the security policy requirements. As a result, if the firewall establishes a new link between its external interface and the server system. Incoming connections are handled by access modules in WAFs. Before transferring traffic to the receiver, the ingress module in the firewall accepts the incoming connection and processes the commands. As a result, the firewall defends systems from application-based threats [13].

Access modules for the most regularly used protocols, such as HTTP, SMTP, FTP, and telnet, are included in WAFs. A certain protocol cannot be utilized to connect over the firewall if there is no access module. WAFs can perform additional message inspection that a simple packet filter does not.

Disadvantages of WAF are low performance, but higher cost than packet filters; Inability to use RPC and UDP protocols.

GENERAL ARCHITECTURE

The suggested WAF paradigm functions as an operating system service that acts as a go-between for the web server and the clients. This service accepts a request, analyzes it, extracts features, classifies it, and takes actions based on the classification outcome.

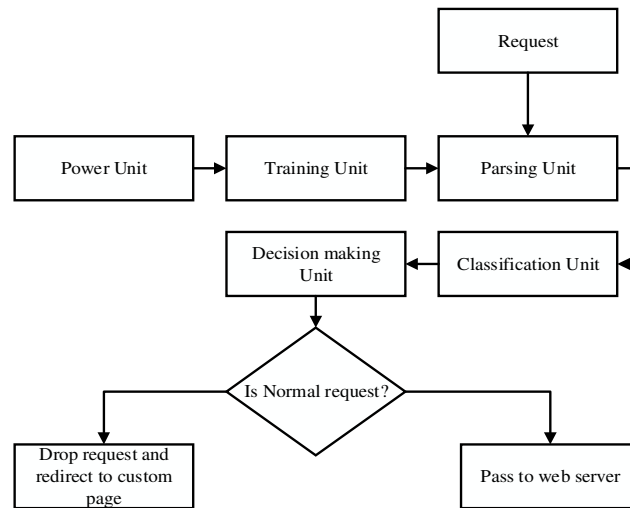


FIGURE 2. Proposed model

WAF can be set up using a specific web application. Fig.2 depicts the proposed WAF, which is made up of the five primary units listed below:

- Turn on/off the device.
- Educational department.
- Unit of analysis.
- Classification.
- Decision department.

When the WAF is started, the OS service communicates with the database to obtain the configurations for running the WAF, opens a listener, and waits for incoming requests for the WAF, which mediators between the client and the web server.

After starting the WAF, the training process begins with the selected data collection and classification method.

After the first and second units have been completed, the WAF will be ready to handle requests after the training procedure is done.

When a request arrives, the WAF parses it as a vector into a unit, and the first unit that handles it decomposes the request, extracts the features, and classifies it based on the classification method selected by the administrator in the training unit.

A classification algorithm is built with the description of the mathematical basis of the above algorithm. The scheme of the developed algorithm is shown in Fig.3.

The educational stage will consist of 4 modules:

- Extraction module: according to the requests received from the client, the author filters the parts necessary to process the requests, including paths, payload, key characters.
- Comparison regular expression module: regular expressions are patterns used to search for sets of characters combined into character strings. Regular expressions are not limited to a particular language, so programmers and experts can apply them to any programming language. In the training phase: by learning risky queries, the author created regular expressions. And then, in the detection phase: after reading the queries from the target dataset, the author runs the regular expression validation module.
- Data transformation module: used to convert string data into vector. Using tf-idf technology evaluates the importance of a word in a query string.
- Data Classification Module: A random forest method to classify a given data set.

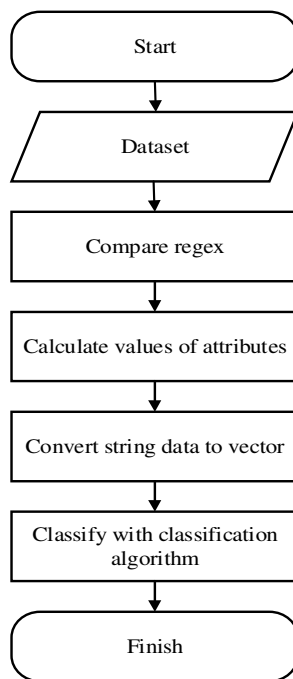


FIGURE 3. Flowchart of algorithm

TABLE 1. Attack weight feature subfeatures

Subfeature	Description
URL weight	Weighted sum of discovered URL manipulations
Number of attack words found in inputs	Weighted sum of found attack words in inputs
Manipulate payload weight	Sum of weights of discovered manipulation in payloads
The proportion of alphanumeric characters to special characters	The amount of alphanumeric characters divided by the number of nonalphanumeric characters.
Files weight	Sum of weights of the malicious files

One of the most popular machine learning methods, which is a cross-validation method to assess the accuracy of dangerous query detection, uses the F1 score.

The algorithm is described using some of the symbols presented in Table 1. To detect common assaults on online services, a set of generalizable attributes collected from HTTP requests is employed. CSIC 2010, HTTPParams 2015, a hybrid dataset (CSIC+HTTPParams), and bespoke web server logs (cracked genuine server) were used. We employed four primary elements retrieved from HTTP requests to determine the final features: HTTP protocol (HTTP method), absolute URL (URL), payload, headers, and files. The extracted properties are query duration, character percentage allowed, special character percentage, and attack severity. We employed numerous classification methods that function better in binary classification issues, such as Linear Regression, Decision Tree, Random Forest, and Naive Bayes.

EXPERIMENTS AND ANALYSIS OF RESULTS

WAF experiments were performed on a Linux Ubuntu web server. This server includes an Apache service (a web service), a web control panel (a Django-Python web application), and a WAF service.

In this investigation, four datasets were used: CSIC 2010, HTTPParams, a hybrid dataset (CSIC 2010 and HTTPParams), and a special dataset of hijacked web server logs. These datasets were prepared for use with machine learning techniques in Python by being exported as CSV.

For classification, we employed four algorithms: Naive Bayes, Logistic Regression, Decision Tree, and SVM. The classifier was fed four datasets using two methods: split train test and cross-validation, and the results were quite close.

Attack detection accuracy. In the field of machine learning, the main task is to divide a set of observations into distinct groups, called classes, based on the analysis of their formal description. In the classification, each observation unit belongs to a certain group or nominal category according to a certain qualitative characteristic [10].

The problem of classification is solved for the use of training with the teacher, because the classes are defined in advance, for example, the learning set, given class marks. Analytical models that solve the problem of classification are called classifiers.

To evaluate the performance of the algorithm, the following concepts are used: true positive response (TP); true negative response (TN); false positive response (FP); false-negative response (FN). In the simplest case, the accuracy value is a numerical estimate of the quality of the classification algorithm and is determined by formula (1):

$$A = \frac{TP+TN}{(TP+TN+FP+FN)} \quad (1)$$

In addition to using precision, many studies use the F1-measure value [14]. Since the F1-measure is the harmonic mean between precision (P) and recall (R) in formula (2). If precision or recall tends to zero, it tends to zero. To relate precision to recall, the F1 measure is introduced as the harmonic mean of precision and recall [13]:

$$F1 = 2 * \frac{P * R}{(P+R)} \quad (2)$$

$$P = \frac{TP}{(TP+FP)} \quad (3)$$

$$R = \frac{TP}{(TP+FN)} \quad (4)$$

In this work, datasets like those in Table 2 were initially used, and Table 3 shows the accuracy indicators for the studied algorithms.

TABLE 2. Sample of dataset

Payload length	Alph	Non alph	Attack feature	Label
42	95	4	200	1
242	100	0	0	0
8	100	0	0	0
25	94	5	2700	1
53	7	22	95000	1
76	100	0	0	0
104	87	12	6000	1
91	84	15	90000	1

In Table 3, we can compare each indicator. We can see that using symbols in vectorization gives better results than words. Comparing the vectorization performance symbol by symbol, we can see that each algorithm has similar results. Table readings in this work were obtained by averaging 10 times cross-validation.

Most of the related works used the CSIC2010 dataset with or without custom datasets, and we used it in the proposed model to be able to compare the proposed model with previous models.

Results compared to related jobs. Our proposed model achieved a high accuracy of 98.6% compared to related works. The following Table 4 shows the results of the CSIC 2010, HTTP parameters and custom datasets created by the researchers.

Researchers have presented many models to detect web attacks, and despite their different characteristics [15], there are some common weaknesses among these studies, which can be summarized as follows:

- Extracted features cannot be generic, and most of these features are only compatible with web applications that extract from it.
- Using an old data set like CSIC and evaluating the model depends on its training results. In addition, not all modern data sets used are available on the Internet.
- There are some errors and incorrect information in some documents of related works, for example, the study of Sharma S., Zavorsky P. and Butakov S. (2020) [16].
- They used features that could not be extracted from CSIC 2010 (eg the _cookie_len feature).
- Most related jobs only process the payload without considering the headers and files.
- Hybrid models are very rare (only the paper of Tekerek A. and O.F.Bay(2019) is a hybrid model in related works) [17].
- While most of the related works detect common web attacks such as XSS and SQL injection, no proposed model can detect attacks that use simple queries such as DOS attacks.

The proposed model implementation includes the functionality to export WAF records as a new data set with the ability to correct records. Administrators can train the proposed model using this exported data set to strengthen WAF in protecting web applications.

TABLE 3. Results by cluster

	Precision	Recall	F1	Accuracy
LogReg+char	0.9825	0.9974	0.993	0.99930
LogReg+word	0.9083	0.9700	0.990	0.9956
LSVM+char	0.9985	0.9971	0.988	0.99985
LSVM+word	0.9906	0.9495	0.981	0.9979
Perteptron+char	0.9888	0.9980	0.990	0.9995
Perceptron+word	0.7345	0.9716	0.991	0.9857

TABLE 4. Comparison of our proposed model and relevant works

Dataset	Our model	Tekerek and Bay	Sharma	Ghafarin
CSIC 2010	99.59	96.74	94.7	88.32
ECML-PKDD 207	98.7	94.53	96.4	89.4
HTTP Params 2015	97.61	96.4	94.5	95.3
Custom	98.8	98.62	97.8	97.6

CONCLUSION

Thus, we conclude that among all existing web application firewalls, a database system with artificial intelligence has yet to be implemented. An updated security system needs to be armed with artificial intelligence to recognize attack patterns and behaviors by creating its own database and mitigation system to eliminate false positives individually and treat each packet independently. We have developed a neural network-based AI engine for web application firewalls that can mitigate all loopholes using artificial intelligence. This work shows that machine learning can lead to practical results that go beyond improving accuracy and performance in detecting exploits. The proposed architecture is one way to improve resource utilization efficiency without neglecting the guarantees and controls offered by a full analyzer. To increase the level of security, we propose to train our proposed model on web server records of web applications protected by WAF. Custom. We utilized techniques that perform well for binary classification issues, such as Logistic Regression, Decision Tree, and Naive Bayes. With a typical dataset (CSIC 2010) utilized in research in this sector, our suggested model obtained a high classification accuracy of 99.59%, and 98.8% with a genuine hacked web server dataset.

REFERENCES

1. M. Chora's and R. Kozik, "Machine learning techniques applied to detect cyber attacks on web applications," *Logic Journal of IGPL*, vol. 23, no. 1, pp. 45–56, 2015.
2. D. Wichers and J. Williams, "Owasp Top Ten," 9e open web application security project, vol. 3, 2017
3. A.H. Yaacob, M. Nazrul, N. Ahmad, and M. Roslee, "Moving towards positive security model for web application firewall," *International Journal of Computer and Information Engineering*, vol. 6, no. 12, pp. 1763–1768, 2012.
4. P.P.Mukkamala and S. Rajendran, "A survey on the different firewall technologies," *International Journal of Engineering Applied Sciences and Technology*, vol. 5, no. 1, pp. 363–365, 2020
5. D. M. Makhmudova, "Using information technology tools in mathematics lessons for teaching future teachers," *International Journal of Scientific and Technology Research*, 9(3), 4168–4171, 2020.
6. W. Wang and K. Siau, "Artificial intelligence, machine learning, automation, robotics, future of work and future of humanity," *Journal of Database Management*, vol. 30, pp. 61–79, 2019.
7. J.H. Lee, "Cybersecurity meets artificial intelligence: a survey," *Frontiers of Information Technology & Electronic Engineering*, vol. 19, no. 12, pp. 1462–1474, 2018.
8. M. Domingues Junior, and N.F.F. Ebecken, "A new WAF architecture with machine learning for Resource-efficient use," *Computers & Security*, 106, p. 102290. 2021. doi:10.1016/j.cose.2021.102290.
9. A. Shaheed, and M.H. Kurdy, 'Web application firewall using Machine Learning and features engineering', *Security and Communication Networks*, 2022, pp. 1–14. doi:10.1155/2022/5280158.
10. C. T. Gimenez, A. P. Villegas, and G.A. Marañón, "Http dataset CSIC 2010," 2010, <https://www.isi.csic.es/dataset/>.
11. H. Yuan, et al., Research and implementation of WEB application firewall based on feature matching, *Proc.Int. Conf. on Application of Intelligent Systems in Multi-modal Information Analytics*, Springer, 2019, pp.
12. Akbar Memen, Ridha Muhammad Arif Fadhly, et al., SQL injection and cross site scripting prevention using OWASP ModSecurity WebApplication firewall, *Int. J. Inf. Visualization*, 2018, vol. 2, no. 4. pp. 286–292.
13. N.M. Thang, "Improving efficiency of web application firewall to detect code injection attacks with random forest method and analysis attributes HTTP request," *Programming and Computer Software*, 46(5), pp. 351–361. 2020. doi:10.1134/s0361768820050072.
14. Accuracy, Precision, Recall or F1? | by Koo Ping Shung | Towards Data Science. <https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37e5cb9>. Accessed 5 May 2023.

15. A.A. Abdushukurov, D. M. Makhmudova, "Semiparametric Estimation of Distribution Function in the Informative Model of Competing Risks," *Journal of Mathematical Sciences (United States)* [this link is disabled](#), 227(2), 117-123, 2017.
16. S. Sharma, P. Zavorsky, and S. Butakov, "Machine learning based intrusion detection system for web-based attacks," in *Proceedings of the 2020 IEEE 6th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS)*, pp. 227–230, IEEE, Baltimore, MD, USA, May 2020
17. A. Tekerek and O. F. Bay, "Design and implementation of an artificial intelligence-based web application firewall model," *Neural Network World*, vol. 29, no. 4, pp. 189–206, 2019.