Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 189 (2021) 359–367

Procedia Computer Science

www.elsevier.com/locate/procedia

5th International Conference on AI in Computational Linguistics

Signature-based and Machine-Learning-based Web Application Firewalls: A Short Survey

Simon Applebaum[a,*], Tarek Gaber[b], Ali Ahmed[c]

[a]University of Liverpool, Liverpool, United Kingdom
[b]Faculty of Computers & Informatics, Suez Canal University, Ismailia, Egypt.
[c]School of Engineering and Computer Science, Victoria University of Wellington, Wellington, New Zealand

Abstract

Web Application Firewalls (WAF) have evolved to protect web applications from attack. A signature-based WAF responds to threats through the implementation of application-specific rules which block malicious traffic. However, these rules must be continually adapted to address evolving threats. The resultant rules can become complex and difficult to maintain, requiring that the administrator possesses a high-level of skills and detailed knowledge of the application. Not to mention the challenges of zero-day attacks! A WAF can deliver high rates of false positives and false negatives that can adversely impact the performance and can provide poor protection against zero-day attacks. This paper aims to provide a short review showing the development of WAFs based on machine-learning-based methods. It discusses their merits and limitations as well as identifying open issues. It assesses which of them can provide countermeasures to zero-day attacks and are easy to configure and maintain to keep them up to date. It was found that machine-learning-based methods have advantages over signature/rule-based methods as the former can address the vulnerability to zero-day attacks and can be easier to configure and keep up to date. The survey also determined that the effectiveness of machine-learning-based WAFs in protecting current attack patterns targeting web application frameworks is still an open area for further investigation.

Keywords: Web Application Firewalls, WAF, Machine learning, signature-based WAF, Zero-day attacks, ModSecurity, OWASP, XSS, SRF;

1877-0509 © 2021 The Authors. Published by Elsevier B.V.

10.1016/j.procs.2021.05.105

1. Introduction

Early web applications typically relied on the simple common gateway interface, CGI, protocol which spawned C or Perl programs. Attacks were concentrated at underlying web server technology and libraries and could be categorised as buffer overflow, insecure sample code, input validation weaknesses, canonicalization attacks, encoding attacks, form tampering and privilege escalation. Improvements in underlying web server technology and greater awareness of secure development practices resulted in the stabilisation of core web server infrastructure. However, the explosion in popularity of blogging, forums and online commerce increased dramatically the use of database-driven applications. Entirely new classes of attacks evolved around these architectures including Cross-site Scripting (XSS), SQL injection, Cross-site Request Forgery (SRF), access control weaknesses and authentication and session management failures. These attacks were exacerbated by the focus at the time of developers on delivering functionality rather than hardening the security of the applications [1]. In other words, security was not a core function of the system [2]. In recognition of this situation, OWASP was founded in 2001 as an open-source community focused on the security of web applications [3]. The objective of OWASP is to improve software security through awareness of secure development practices and by the provision of associated tools. Supporting these objectives, the OWASP Top Ten[*] identifies the most critical security risks affecting web applications so that appropriate development processes may be used to reduce these risks.

At the time of writing, the OWASP Top Ten identifies the following web application security risks:

- Injection – where untrusted content can be injected into an interpreter as a command or query and the interpreter executes the commands.
- Broken Authentication – where attackers can exploit flaws in session management to assume another user's identity.
- Sensitive Data Exposure – where sensitive data is exposed or stolen by the attacker exposing financial, healthcare or personal identifying information (PII).
- XML External Entities (XXE) – where XML processors incorrectly expose private internal entities exposing them to scanning or denial of service attacks.
- Broken Access Control – where errors in authentication or access controls allow attacks to access unauthorised functionality or data.
- Security Misconfiguration – where insecure default settings or misconfiguration of system settings allow incorrect access to unauthorised functionality or data.
- Cross-Site Scripting (XSS) – where an attacker can inject data into a web page allowing the attacker to execute arbitrary malicious scripts.
- Insecure Deserialisation – where an attacker can cause an application to deserialise arbitrary malicious content leading to remote code execution.
- Using Components with Known Vulnerabilities – where an attacker exploits an application that uses libraries or components containing vulnerabilities.
- Insufficient Logging & Monitoring – where attackers can hide their presence from administrators or monitoring systems.

Around the time of the formation of OWASP, WAFs began to emerge as an approach to securing web applications. The first version of the open-source ModSecurity[†] WAF was launched in November 2002. A WAF provides real-time access to HTTP traffic allowing the firewall to inspect incoming and outgoing traffic. Anomalous and malicious traffic may be detected and blocked, HTTP traffic may be logged for forensic analysis, and HTTP features may be restricted to reduce a web application's attack surface. Virtual patching allows an application vulnerability to be resolved without recourse to software changes and instead the issue may be addressed through the implementation of specific rules [4].

[*] https://owasp.org/www-project-top-ten/

[†] https://github.com/SpiderLabs/ModSecurity

A WAF typically makes use of a security model based on how its policies are implemented. Two main security models are used: the negative security model and the positive security model. A WAF adopting a negative security model allows all traffic to pass unless it matches defined rules, in which case the traffic is blocked. If traffic does not match the rules, the negative security model allows the traffic to pass. The negative security model is commonly implemented through the use of a signature-based approach. The signature-based approach aims to identify malicious incoming data using pattern matching techniques, so is able to block previously identified patterns and variants [5]. Conversely, a WAF adopting a positive security model allows traffic to pass if the traffic conforms to a policy. If traffic does not conform to the policy, it is blocked. Negative and positive security models have advantages and disadvantages. For example, the negative security model may be insufficient since hackers may evolve attacks to work around existing security policies. Conversely, the positive security model may require extensive planning and implementation based on a thorough knowledge of the application and may be inflexible where the functionality or content presented by the application is evolving. A combination of positive and negative security models may provide the best results [6].

An example of a security policy is provided by the OWASP ModSecurity Core Rule Set (CRS)[‡] . The CRS provides detection rules for use with ModSecurity and aims to protect the web application from common attacks such as those defined in the OWASP Top Ten. In addition to the generic detection rules, the CRS also provides web application framework-specific rules covering popular software such as WordPress and Drupal. However, these additional rules are mainly oriented towards disabling generic rules which would otherwise result in false positives when used with WordPress and Drupal.

The evolution of AI techniques has resulted in attention to the application of machine learning to anomaly detection by WAFs. Techniques explored include Artificial Neural Networks (ANN) based on character and keyword-based features [7]; parameter length and character distribution, parameter class (i.e. numerical, URL, email etc.) and enumerated type [8]; Single Class SVM based on length, digit count, a mix of alphanumeric characters and entropy [9]; Single Class SVM based on n-gram [10]; Decision Tree based on n-gram [11]; Artificial Neural Network [12].

This paper aims to provide a short review showing the development of WAFs based on machine-learning-based methods. It discusses their merits and limitations as well as identifying open issues. It assesses which of them can provide countermeasures to zero-day attacks and are easy to configure and maintain to keep them up to date.

The rest of the paper is structured as follows. Section 3 presents a summary and analysis of published work relating to machine-learning-based WAF methods. Discussion of the findings and some open issues are given in Section 4 while the conclusion is given in Section 5.

2. Literature review

This literature review focusses on the development of research relating to anomaly based WAFs featuring machine learning algorithms, which are described in section 3.1. Hybrid WAFs combining signature and machine learning based methods offer advantages of both approaches and a number of approaches are reviewed in section 3.2. Relevant to much of the work are the datasets used for training and testing of machine learning based algorithms. A commonly used dataset is CSIC 2010 which is discussed in section 3.3. Finally, section 3.4 demonstrates the potential of machine learning based approaches to the generation of test cases and illustrating the threat to both signature and machine learning based WAFs.

2.1. Anomaly Based WAFs

Torrano-Giménez et al [13] present an anomaly-based WAF. The firewall is configured using an XML file describing in detail the web application's expected normal behaviour. The XML file contains rules defining normal requests including HTTP verbs, headers and directories corresponding to the web application's folder structure, including input arguments and legal values, defined by statistical properties. Torrano-Giménez et al [13] explore options for obtaining normal traffic, which is ultimately produced by using artificial traffic generators and dictionaries to control variable parameters. The effectiveness of the resultant WAF is tested using normal and anomalous traffic

[‡] https://coreruleset.org/

and calculated using Receiver Operating Characteristic (ROC) curves. The firewall achieves extremely high accuracy and low levels of false positives due to the XML file very closely matching the web application's normal behaviour. A limitation is a difficulty in creating the XML file because of the challenge in obtaining large volumes of non-malicious traffic. Challenges are also experienced with websites that dynamically add resources or content.

Moosa [7] reflects on the proliferation of web applications, threats and attack signatures and how this reduces the effectiveness of signature-based WAFs such as ModSecurity. Moosa proposes high-level requirements that an effective firewall should deliver, including the ability to handle increased number and complexity of rules, it should be user friendly and easy to update, and it should not impact the performance of the webserver when filtering requests. Moosa proposes an artificial neural network (ANN) based approach and focuses on SQL injection. At the time of writing SQL injection ranked the top vulnerability by OWASP and it still occupies that ranking today. Moosa explores SQL injection techniques, categorising them as tautologies, union queries, piggyback queries, malformed queries, inference and alternate coding. Moosa explores the pros and cons of positive and negative security models, for example, the difficulty in adapting to a complex and evolving application and the complexity and difficulty in updating rules in response to evolving attack patterns. Moosa implements an ANN-based firewall using character-based features, then keyword-based features. The firewall is eventually integrated into a Perl-based web server which delivered acceptable accuracy and performance. Moosa suggested that future work would aim to extend the firewall to provide protection against XSS and other attacks and to replace a Matlab based training engine with a graphical user interface to the WAF.

Pałka and Zachara [8] explore the use of a machine-learning WAF for protecting web applications. In particular, they consider the use of the referer request header. This contains the referring page URL which may be used to identify forceful browsing exploit as a result of unusual, referred browsing history. They also explore issues resulting from WAF data storage and how this could represent a vulnerability in its own right if sensitive data is stored or processed within the WAF algorithms. Pałka and Zachara explore the pros and cons of triggered learning versus continuous learning schemes. Triggered learning is resistant to attacks that target the learning phase and has lower ongoing data retention needs. However, triggered learning is less adaptable to changing usage patterns and evolution of the application, which would require re-training, so is more suitable for stable applications that do not change. Continuous learning is less precise but requires a less initial setup. However, continuous learning may trigger false positives with changing usage patterns and could be vulnerable to attacks that target the learning process itself. Pałka and Zachara implemented a machine-learning-based WAF in Apache based on parameter length and character distribution, parameter class (i.e., numerical, URL, email etc.) and enumerated type.

Nguyen et al. [14] recognise the emergence of machine-learning-based WAFs and highlight the importance of feature selection to their effectiveness. Nguyen et al. evaluate the use of generic feature selection (GeFS) [15] using a preselected 30 features considered relevant for web attack based on their expert knowledge. Nguyen et al. [14] analysed the features to identify linear correlation or non-linear relations, then using correlation feature selection (CFS) or minimal-redundancy-maximal-relevant (mRMR) based on the outcome. This provided a large reduction in features due to irrelevance or redundancy with a minor impact on the detection accuracy. The CFS method resulted in 11 features that were tested in conjunction with the CSIC 2010 dataset [16] providing good detection rates and false-positive rates. Nguyen et al. list the full feature set and the subset of features selected by CFS and mRMR.

Ahmad et al. [17] review how common signature-based WAFs are managed by administrators resulting in complex layered configurations which are difficult to manage and have suboptimal performance. Signature-based WAFs require continual adaptation to address evolving threats. In response to this, administrators may create more specialised application-specific rules. Administrators can tend to create new rules in response to threats rather than adjusting existing rules. The result is that the configuration becomes highly complex with layer upon layer of complex rules, increasing the rate of false positives and false negatives, and resulting in low performance [18]. This is exacerbated by the level of expertise required, the detail and complexity of the rules and the absence of user interfaces or other tools to simplify the management task. Ahmad et al. [17] proposed a conceptual framework for structuring WAF management. The proposed framework allowed redundancies and relationships to be identified and for solutions to be more easily evaluated in the context of existing rules.

Torrano-Giménez et al. [11] propose an anomaly-based WAF. The firewall constructs the set of features based on expert knowledge and n-grams on the basis that this combination would provide better effectiveness than the approaches applied separately. Torrano-Giménez et al. [11] explore three approaches to combining expert knowledge and n-gram features: combine-select (combine expert knowledge and n-gram features, then feature selection to reduce dimensionality), select-combine (feature selection to reduce dimensionality then combine selected expert knowledge and n-gram features) and select-n-gram-combine (select n-gram features only to reduce dimensionality then combine selected n-gram features with expert knowledge features). Feature selection is done using the GeFS method [15]. Torrano-Giménez then measures the effectiveness of the firewall using the ECML/PKDD 2007 dataset using the three combination approaches, feature selections and decision tree algorithms.

Epp et al. [9] construct a machine-learning-based WAF using One-Class SVM. They describe their system architecture which comprises four steps: routing, data pre-processing, classification and response. The system is implemented using Python using the scikit-learn machine-learning library, and their source code and training data sets are published on github (https://github.com/nico-ralf-ii-fpuna/paper). Routing groups incoming requests by URL and HTTP method, allowing more focused and accurate models to be created. Data pre-processing uses feature extraction based on characteristics of the messages, including length, digit count, a mix of alphanumeric characters and entropy. Classification trains a One-Class SVM classifier for each URL/HTTP method group. Linear, polynomial and Radial Basis Function (RBF) kernels were tested with RBF giving the best classification results. The response step forwards requests and logs or blocks anomalous requests. Effectiveness was calculated using true positive rate, false-positive rate and a derivative of true positive, false positive and false negative and gave comparable results versus other machine-learning-based WAFs, and significantly better results compared with ModSecurity with default rules. Response time was increased by 2-4mS per request and therefore considered acceptable for production use.

Liang et al. [19] proposed a new approach for anomaly detection in web applications using Recurrent Neural Networks (RNNs). Multilayer Perceptron (MLP) is adopted in this work. It is worth mentioning that MLP is a feedforward artificial neural network. The performance investigation is focused on measuring Accuracy, Sensitivity, and Specificity of the model. The investigation reveals, as claimed by the authors, that the model performs well compared against some of the models in the literature which has not been introduced. In other words, the work does not introduce any benchmark to compare their results against although claiming their performance is comparable to the state-of-the-art! One more concern with this work is it did not consider KDD CUP 99 and its parent DARPA98 datasets which are regarded by Divekar et al. [20] as "some of the most widely used datasets in the annals of Anomaly-based Network Intrusion Detection" and been used by Samrin and Vasumathi [21] in their interesting work that reviewed anomaly-based network intrusion detection systems.

Appelt et al. [22] build on their previous work (Appelt et al.[23]) by enhancing their earlier ML-Driven algorithm. They compare the effectiveness of the enhanced algorithm against previous algorithms; against other prominent WAF testing tools (sqlmap and WAF Testing Framework); and the performance of the algorithm against ModSecurity and a proprietary WAF. The enhanced algorithm is found to have higher performance than previous algorithms. Sqlmap and WAF Testing Framework are found to be orders of magnitude less effective than the machine-learning-based algorithm.

Betarte et al. [10] review the performance of ModSecurity using the out-of-the-box OWASP Core Rule Set. They identify disadvantages of this approach including high rates of false positives, poor recognition of zero-day attacks, and lack of flexibility in protecting dynamic content managed websites. Betarte et al. explore supplementing ModSecurity/CRS with a one-class classification suitable for use in the absence of specific training data, and an n-gram anomaly detection-based approach for when application-specific training data is available. Both approaches were used to supplement ModSecurity/CRS, with one-class classification improving false positive performance, and n-gram anomaly detection providing better resilience against zero-day attacks.

Vartouni et al. [24] measure the effectiveness of a machine-learning-based WAF using a range of algorithms for feature construction, extraction and classification. Feature construction is measured with 1-gram and 2-gram methods. Feature extraction is implemented using a combination of stacked auto-encoder and deep belief network and tested against the CSIC 2010 and ECML/PKDD 2007 datasets. Anomaly detection is implemented using one-class SVM, isolation forest and elliptic envelope. Vartouni et al. compare the various methods against each other considering algorithmic effectiveness, training time and run time performance, concluding that deep models have better performance than traditional models.

Moradi et al. [25] investigate another method of anomaly detection. The work proposes mechanisms based on deep-neural-network and parallel-feature-fusion. The proposal "use stacked autoencoder and deep belief network as feature learning methods, in which only normal data is used in the classification of the training phase, then, one-class SVM, isolation forest, and elliptic envelope are applied as classifiers". The work did not consider data streams though and the resultant WAF is not deployed as a cloud service which limits its adoption.

2.2. Hybrid WAFs

Tekerek and Bay [12] investigate the use of a hybrid WAF comprising signature-based detection (SBD) and anomaly-based detection (ABD) using an artificial neural network. The neural net-based component makes use of three input features: alphanumerical character analysis, letter frequency analysis and request length analysis. The resultant WAF was trained and tested using CSIC 2010, ECML-PKDD 2007 and WAF 2015 datasets. Tekerek and Bay measure the performance and effectiveness of the three-stage-signature-based detection and anomaly-based detection. The proposed WAF achieves good levels of performance through its hybrid architecture. The faster signature-based phases are performed before the slower anomaly-based phase. Previous normal and anomalous requests, identified by the anomaly-based detection phase, are fed back to the signature-based phases to increase detection performance.

Prabhudesai et al. [26] describe a hybrid WAF comprising three phases. The first phase is signature-based, using OWASP based rules, and used to filter requests which match well-known attack patterns. If the request clears this first phase, it is passed to the second phase which is implemented using an AI engine. If malicious traffic is detected in the second phase, the request is added to a knowledgebase, otherwise, the request is passed to the third phase which is IP based. The third, IP-based, phase checks the IP address using the Virustotal API IP-address (https://developers.virustotal.com/) report to determine if the IP address has been reported as malicious. On clearing all three phases, the request is passed to the application. The paper describes a commercial system and gives little detail into the AI engine implementation.

2.3. Datasets

Giménez et al. [16] provide an HTTP dataset intended to be used for the development and testing of Intrusion Detection Systems and WAFs. The dataset generates normal and anomalous HTTP requests and includes common web attacks including SQL injection, XSS, and buffer overflow. The dataset has been used extensively by other researchers including [24] and [28]. There is still a need for a development of a new dataset as previous datasets had become outdated and did not target real systems. The dataset in [16] potentially has similar issues, in that it is itself now over 10 years old and targets a bespoke e-commerce system.

2.4. Test Frameworks

Appelt et al. [23] recognise the difficulty in responding to evolving threats to web-based applications. Rule-based WAFs typically make use of complex pattern-based matching algorithms and these require expertise to create and may be prone to weakness due to rule implementation bugs and configuration errors. Appelt et al. [23] describe an automated test framework that may be used to test the firewall and find weaknesses, focussing on SQL injection. A randomised test case generator identifies initial bypassing test cases, after which a machine-learning-based algorithm mutates bypassing test cases to generate and prioritise new test cases, which are tested and fed back into the test production model. The machine-learning-based test case generator was demonstrated to have higher performance versus the randomised test case generator and was able to detect a high number of bypassing tests against ModSecurity configured with the OWASP Core Rule Set.

Demetrio et al [30] presented a tool that attempts to model an adversary presence in the system. The tool is called WAF-A-MoLE. The tool provides ways to alter the syntax of a payload through the proposal of a set of mutation operators. It is worth noting that this is done entirely without affecting the original semantics. The evaluation of the tool shows it outperforms many existing WAFs that are based on machine learning.

3. Discussion and open issues

The literature review has explored the emergence of WAFs used to protect web applications. In particular, the literature review has identified issues with signature-based WAFs including their vulnerability to zero-day attacks and the difficulty in configuring rules and keeping them up to date. Given the literature reviewed in the previous section, one can conclude the following lessons:

1. Signature-based WAFs can become complex with time, requiring high levels of expertise to create and adjust as new attacks are identified and evolve. Their reliance on human intervention and complexity can result in weaknesses due to rule implementation bugs and configuration errors. Without careful and expert attention, they can be associated with high rates of false positives and false negatives.
2. The paper has explored and demonstrated the effectiveness of machine-learning-based WAFs which may be used to supplement or used instead of signature-based algorithms. In some proposals, the evaluated system achieved 98.8% accuracy [28].
3. While the literature has evaluated various algorithms using test sets on demo applications, it does not provide detailed implementation details of the algorithms. This makes the exact reproduction of the test results difficult. The previous work has focused on detection SQL injection with less attention placed on other classes of vulnerability. Updating standard public datasets could allow researchers to investigate machine and deep learning techniques to suggest a reliable model for ML-based WAFs which could overcome the limitations of the signature-based ones. It is worth mentioning that several workpieces proposed a comprehensive deep learning architecture such as the work of [30], [31], and [32].
4. The discussion of the various workpieces reveals areas for improvement as follows:
    a. The evaluation of open-source machine-learning-based WAFs and algorithms;
    b. The evaluation of performance on the current web hosting hardware;
    c. Application to other classes of attacks beyond injection;
    d. The evaluation of the effectiveness of machine-learning algorithms in protecting against current attack patterns targeting web application frameworks.

It is worth noting that there is no consensus on a framework or an approach to test WAFs although workpieces such as that of [22] exist. The work of [22] is interesting as it used machine-learning to test WAFs by generating SQL injection attacks, bypassing WAFs and identifying attack patterns.

4. Conclusion

Web Application Firewalls (WAF) have evolved to protect web applications from attack. Using signature-based approaches can be restrictive as it requires continuous adaptation of rules to match the evolving threats. The evolution of new attack patterns makes signature-based approaches vulnerable to zero-day attacks where the existing signature-based rules are not able to identify the new attack pattern. The complexity of rule specification increases the difficult of adapting signature-based systems in response to evolving threats. Machine-learning could play a significant role to solve this problem. This paper reviewed the development of machine-learning-based WAF methods to investigate their merits and limitations as well as identifying open issues. The final lesson learnt given the literature survey is, machine-learning-based methods have advantages over signature/signature-based ones as the former can address the vulnerability to zero-day attacks and can be easier to configure and keep up to date. The survey also identified that the effectiveness of machine-learning algorithms in protecting current attack patterns targeting web application frameworks is still an open area for further investigation. It is worth noting that there is no consensus on a framework or an approach to test WAFs. However, it was shown that there a potential of machine learning based approaches to be used for the generation of test cases of WAF systems.

Acknowledgement

This work is part of the MSc dissertation of the 1st author (Simon Applebaum) which has been submitted to the Laureate Higher Education Group (University of Liverpool).

Glossary of terms

ABD – Anomaly Based Detection
ANN – Artificial Neural Network
CGI – Common Gateway Interface

CFS – Correlation Feature Selection
CRS – Core Rule Set
GeFS – Generic Feature Selection
HTTP – Hypertext Transfer Protocol
ML – Machine Learning
MLP - Multilayer Perceptron
OWASP - Open Web Application Security Project
PII – Personally Identifiable Information
RBF - Radial Basis Function
RNN - Recurrent Neural Networks
ROC - Receiver Operating Characteristic
SBD – Signature Based Detection
SQL – Structured Query Language
SVM – Support Vector Machine
URL – Uniform Resource Locator
WAF – Web Application Firewall
XML – Extensible Markup Language
XSS – Cross Site Scripting

References

[1] Watson, D., (2007) The evolution of web application attacks. Network Security, 2007(10), pp.10-14.

[2] Ragab, N., Ahmed, A. and AlHashmi, S., (2015) Software Engineering for Security as a Non-functional Requirement. In Intelligent Data Analysis and Applications (pp. 347-357). Springer, Cham.

[3] Curphey, M., (2001) webappsecsecurityfocus.com" - New list Charter and New Name!. [online] Available at: https://web.archive.org/web/20160316215842/https://archives.neohapsis.com/archives/sf/www-mobile/2001-q3/0112.html [Accessed 2 Nov. 2020].

[4] Folini, C. and Ristić, I., (2018) ModSecurity Handbook. The Complete Guide to the Popular Open-source Web Application Firewall. Second edition. Feisty Duck. London.

[5] Thang, N.M., 2020. 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.

[6] Clincy, V. and Shahriar, H., (2018) Web Application Firewall: Network security models and configuration. In 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC) (Vol. 1, pp. 835-836). IEEE.

[7] Moosa, A., (2010) Artificial neural network-based Web Application Firewall for SQL injection. International Journal of Computer and Information Engineering, 4(4), pp.610-619.

[8] Pałka, D. and Zachara, M., (2011) Learning Web Application Firewall-benefits and caveats. In International Conference on Availability, Reliability, and Security (pp. 295-308). Springer, Berlin, Heidelberg.

[9] Epp, N., Funk, R., Cappo, C. and Lorenzo–Paraguay, S., (2017) Anomaly-based Web Application Firewall using HTTP-specific features and one-class SVM. In Workshop Regional de Segurança da Informação e de Sistemas Computacionais.

[10] Betarte, G., Giménez, E., Martinez, R. and Pardo, A., (2018) Improving Web Application Firewalls through anomaly detection. In 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA) (pp. 779-784). IEEE.

[11] Torrano-Giménez, C., Nguyen, H.T., Alvarez, G. and Franke, K., (2015) Combining expert knowledge with automatic feature extraction for reliable web attack detection. Security and Communication Networks, 8(16), pp.2750-2767.

[12] Tekerek, A. and Bay, O.F., (2019) Design and implementation of an artificial intelligence-based Web Application Firewall model. Neural Network World, 29(4), pp.189-206.

[13] Torrano-Giménez, C., Perez-Villegas, A. and Alvarez, G., (2009) A self-learning anomaly-based web application firewall. In Computational Intelligence in Security for Information Systems (pp. 85-92). Springer, Berlin, Heidelberg.

[14] Nguyen, H.T., Torrano-Giménez, C., Alvarez, G., Petrović, S. and Franke, K., (2011) Application of the generic feature selection measure in detection of web attacks. In Computational Intelligence in Security for Information Systems (pp. 25-32). Springer, Berlin, Heidelberg.

[15] Nguyen, H.T., Franke, K. and Petrovic, S., (2010) Towards a generic feature-selection measure for intrusion detection. In 2010 20th International Conference on Pattern Recognition (pp. 1529-1532). IEEE.

[16] Giménez, C.T., Villegas, A.P. and Marañón, G.Á., (2010) HTTP data set CSIC 2010. Information Security Institute of CSIC (Spanish Research National Council).

[17] Ahmad, A., Anwar, Z., Hur, A. and Ahmad, H.F., (2012) Formal reasoning of Web Application Firewall rules through ontological modelling. In 2012 15th International Multitopic Conference (INMIC) (pp. 230-237). IEEE.

[18] Thomas-Reynolds, Dainya and Butakov, Sergey, (2020) Factors Affecting the Performance of Web Application Firewall. WISP 2020 Proceedings. 8.

[19] Liang, J., Zhao, W., & Ye, W. (2017, December). Anomaly-based web attack detection: a deep learning approach. In Proceedings of the 2017 VI International Conference on Network, Communication and Computing (pp. 80-85).

[20] Divekar, A., Parekh, M., Savla, V., Mishra, R., & Shirole, M. (2018, October). Benchmarking datasets for anomaly-based network intrusion detection: KDD CUP 99 alternatives. In 2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS) (pp. 1-8). IEEE.

[21] Samrin, R., & Vasumathi, D. (2017, December). Review on anomaly based network intrusion detection system. In 2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT) (pp. 141-147). IEEE.

[22] Appelt, D., Nguyen, C.D., Panichella, A. and Briand, L.C., (2018) A machine-learning-driven evolutionary approach for testing Web Application Firewalls. IEEE Transactions on Reliability, 67(3), pp.733-757.

[23] Appelt, D., Nguyen, C.D. and Briand, L., (2015) Behind an application firewall, are we safe from SQL injection attacks? In 2015 IEEE 8th international conference on software testing, verification and validation (ICST) (pp. 1-10). IEEE.

[24] Vartouni, A.M., Teshnehlab, M. and Kashi, S.S., (2019) Leveraging deep neural networks for anomaly-based Web Application Firewall. IET Information Security, 13(4), pp.352-361.

[25] Moradi Vartouni, A., Mehralian, S., Teshnehlab, M. and Sedighian Kashi, S., 2019. Auto-Encoder LSTM Methods for Anomaly-Based Web Application Firewall. International Journal of Information and Communication Technology Research, 11(3), pp.49-56.

[26] Prabhudesai, P., Bhalerao, A.A. and Prabhudesai, R., (2019) Web Application Firewall: Artificial Intelligence Arc. International Research Journal of Engineering and Technology (IRJET).

[28] Ito, M. and Iyatomi, H., (2018) Web application firewall using character-level convolutional neural network. In 2018 IEEE 14th International Colloquium on Signal Processing & Its Applications (CSPA) (pp. 103-106). IEEE.

[29] Demetrio, L., Valenza, A., Costa, G. and Lagorio, G., (2020) WAF-A-MoLE: evading web application firewalls through adversarial machine learning. In Proceedings of the 35th Annual ACM Symposium on Applied Computing (pp. 1745-1752).

[30] Xin, Y., Kong, L., Liu, Z., Chen, Y., Li, Y., Zhu, H., ... & Wang, C. (2018). Machine learning and deep learning methods for cybersecurity. Ieee access, 6, 35365-35381.

[31] Al-Garadi, M. A., Mohamed, A., Al-Ali, A. K., Du, X., Ali, I., & Guizani, M. (2020). A survey of machine and deep learning methods for internet of things (IoT) security. IEEE Communications Surveys & Tutorials, 22(3), 1646-1685.

[32] Aldweesh, A., Derhab, A., & Emam, A. Z. (2020). Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues. Knowledge-Based Systems, 189, 105124.