ISSN: 2583-9888
DOI: doi.org/10.51219/JAIMLD/sandeep-phaniredy/500

URF PUBLISHERS
connect with research world

Journal of Artificial Intelligence, Machine Learning and Data Science
https://urfpublishers.com/journal/artificial-intelligence

Vol: 1 & Iss: 1

Research Article

Adaptive AI Web Applications Firewalls to Analyze Web Traffic in Real-Time to Flag Malicious Payloads or Unusual Access Attempts

Sandeep Phaniredy*

Citation: Phaniredy S. Adaptive AI Web Applications Firewalls to Analyze Web Traffic in Real-Time to Flag Malicious Payloads or Unusual Access Attempts. *J Artif Intell Mach Learn & Data Sci 2022*, 2(1), 2297-2301. DOI: doi.org/10.51219/JAIMLD/sandeep-phaniredy/500

Received: 03 September, 2022; **Accepted:** 28 September, 2022; **Published:** 30 September, 2022

***Corresponding author:** Sandeep Phaniredy, USA, E-mail: phanireddysandeep@gmail.com

ABSTRACT

As malicious actors devise new and sophisticated techniques and tactics to gain unauthorized access to systems and networks, artificial intelligence (AI) is undoubtedly a powerful technology that cybersecurity teams can leverage to automate repetitive tasks in security. This technology helps organizations enhance their threat detection capabilities, response and action accuracy when dealing with security issues, such as SQL Injection, DDoS and cross-site scripting designed to sabotage, steal or destroy information.

This paper looks at the architecture of an adaptive AI web application firewall and how it analyzes web traffic in real-time to flag malicious payloads and unusual access attempts.

Keywords: Adaptive AI, Web application firewalls, Real-time traffic analysis, AI-driven web security

1. Introduction

Conventional web application firewalls (WAF) enhance security as an extra defense layer on web applications that provides application-level filtering. However, these security tools cannot swiftly detect new and complex attacks, allowing attackers to steal information between initial detection and mitigation.

AI-based web application firewalls provide context-aware risk-based adaptation, optimization and learning that enable the configuration to be changed in response to different scenarios. The adaptive AI-based WAF adjusts the security mitigation approach to the specific IT environment and state, allowing real-time risk prediction and quantification. This capability involves monitoring the security tool, the protected web application and the operating context to allow autonomous behavior change to quantify and keep risk at a desired level.

2. Background of Adaptive AI Web Application Firewalls

2.1. Static WAF detection falls short in dynamic threat intelligence

Web application firewalls operating on the application layer of the Open Systems Interconnections (OSI) model analyze the HTTP traffic to detect and block attacks. As a commonly used WAF, signature or rule-based firewalls rely on rules written in regular expressions that detect different web application attacks[1]. These security tools are transparent and deterministic, easy to deploy, effective in compliance, resource-efficient and have a low rate of false positives. Apart from signature-based firewalls, anomaly-based WAFs rely on prior knowledge about legitimate or standard traffic flowing in or out of the protected web server, applying protection rules when detecting an anomaly[2,5].

However, traditional cloud-based WAFs are not efficient in detecting complex attacks quickly. These security tools rely heavily on signature-based detection and static firewall ruleset mechanisms. These tools are only effective in detecting and mitigating known threats.

These legacy techniques expose systems and data due to detection gaps when malicious actors change their payloads slightly. The methods rely on hardcoded pattern recognition, which requires manual updates of attack signatures or legitimate patterns to accommodate new threat detection. This approach is inefficient at an age when user behavior evolves rapidly.

Zhang et al.[3] demonstrate this by designing a grammar for SQLi payloads and using ARTSQLi to accelerate testing. Their approach includes first decomposing each payload into tokens characterized as a feature vector. The researchers randomly generate a size-fixed candidate set from the payloads to identify a promising payload. This adaptive selection and execution of promising payloads based on the defined metrics increased the success of attack trials.

Demetrio et al.[4] leverage several mutation actions, like whitespace substitution, comment injection and case swapping, to develop new payloads. Their study first tests a malicious payload that a standard WAF quickly detects. However, the applied mutation operators preserve the original payload's semantic integrity, so it remains functionally the same. After the modifications, the WAF's classification algorithm fails to detect the payloads as malicious, which allows successful SQLi attacks. These studies form the foundations of the evolving cyber-attacks to evade rule-based WAF detection.

The example below shows a simple SQL injection pattern:

“UNION SELECT + user_pass+FROM+wp_users+WHERE+ID=

Then, a genuine regular request:

Select from the menu options where available.

A WAF relying on the regex rule looks for select.*from* statement, the security tool might match the above two statements, resulting in a false positive. Unfortunately, developing clear WAF rules and not being susceptible to this conflict is complicated.

2.2. Enhancing pattern detection through AI

In response to the shortcomings of static techniques, modern web applications integrate AI to automate and speed up threat detection and mitigation. AI-based WAFs feature machine learning (ML) algorithms that effectively detect anomalies in requests and data that conventional firewalls miss. In particular, these tools replace or augment regex-based mechanisms by detecting complex patterns often missed by static firewall rulesets and signature-based techniques. Also, AI-based tools can accurately detect patterns in massive datasets.

AI-based WAFs can also adapt, optimize and learn in real-time as attacks emerge, ensuring efficient detection and response to malicious patterns. The technology knows to make protection recommendations by applying automatic threat detection updates to the WAF. This capability allows AI-based WAFs to identify zero-days in ways that traditional static firewalls cannot. Notably, zero-days are infamously hard to detect and mitigate since they have limited indicators of compromise or known signatures. For example, the technology detects sudden changes in web traffic patterns as potential attacks even when the activity does not match any existing signature.

Additionally, AI-based WAF analyzes and processes massive datasets faster than conventional security tools. The firewalls leverage in-context learning and large-scale neural networks to synthesize new outputs based on user requests, making them adaptable in security web applications. In other words, the security tools do not require signatures or legitimate traffic updates, but their capabilities effectively decrease false positive and false negative detections.

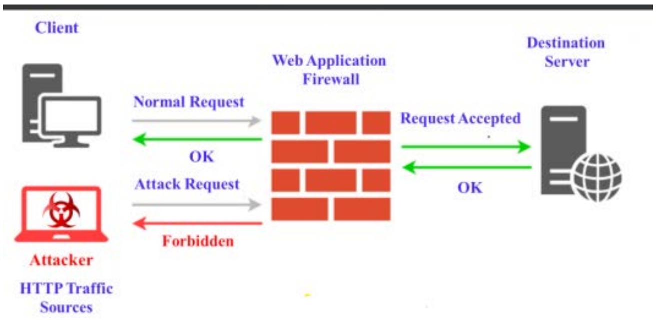

Figure 1: How a WAF works[8].

3. Adaptive AI Web Applications Firewalls

3.1. Architecture of adaptive AI WAFs

Adaptive AI WAFs feature a traffic inspection engine with policies containing rules for inspecting traffic before allowing or blocking it based on the outcomes. The engine comprises access rules that contain criteria and actions that define how the inspection engine treats network traffic from different sources. Other rules include inspection elements that determine how the firewall looks for traffic patterns that pass past access rules.

Example of traffic inspection rule:

Allow traffic from the internal network IP 10.0.0.X. to the public server 196.23.6.Y on port 80.

Inspect inbound traffic using the inspection engine.

Deny all traffic with an invalid source address for incoming traffic or an invalid external address for outbound traffic.

Another component of the AI WAF architecture is the feature extraction engine. This paper demonstrates how the component works by deploying K- means and DBSCAN clustering algorithms to classify normal and anomalous traffic. K- means is a partitioning clustering technique that separates data points into clusters; each data point belongs to the closest mean cluster. The K-means example below shows two steps: the expectation stage assigns data to the nearest mean cluster, while the maximization step computes the mean of the cluster points to set a new mean cluster or centroid.

```
Specify the number k of clusters to assign.
Randomly initialize k steroids.
Repeat
    Expectation: Assign each point to its closest centroid.
Maximization: Compute the new mean centroid for each cluster
Until the centroid position does not change.
```

All data points in a cluster should be similar to each other and if a cluster is different, then that might be considered anomaly detection. Increasing data points within a cluster improves the engine's performance. The data points in regular traffic will be different from the ones in a cluster considered to be a threat.

Density-based Spatial Clustering of Applications with Noise (DBSCAN) is another widely used algorithm for outlier detection. In DBSCAN, cluster performance requires minimum points or minpts and epsilon (eps). Minpts are the minimum number of neighbors, while eps is the radius that forms a cluster. DBSCAN algorithm starts with a random point and expands it to create a cluster until eps and minpts criteria are met. This process is repeated until all points are processed and new clusters are created. Data points that are missing in the created clusters are treated as noise or anomalies. The algorithm can accurately determine different cluster sizes, noise and arbitrary patterns.

A sample DBSCAN algorithm follows these steps: identify core points by counting the number of points within a dataset's eps neighborhood and if a count exceeds the minpts, it is recognized as a core point. Next, the algorithm forms clusters for each core point not assigned to existing clusters. In the density connectivity step, two points, a and b, are density connected if there is a chain of points where the points are within the eps radius, with one point in that chain being a core point. Finally, all points outside the clusters are labeled as outliers or noise.

```
DBSCAN (dataset, eps,MinPts)
{
# cluster index
C = 1
For each new point a in a data set
{
        Mark a as visited
        # find neighbors
        Neighbors N = find neighboring points of a
        If |N| >= MinPts:
        N =N U N
        If a' is not a member of any cluster:
                Add a to cluster C
```

An anomaly detection system is another component of the AI WAF architecture that uses algorithms like Support Vector Machine (SVM) to detect outliers in a dataset containing data points belonging to one class[6]. SVM is effective for high-dimensional datasets with diverse features, such as those found in anomaly detection activities. Miller et al.[7] note that SVMs leverage a predictive algorithm to learn multivariate patterns that optimally discriminate between clusters.

Threat intelligence integration is another essential component of adaptive AI WAFs. The security tools can integrate with up-to-date threat intelligence feeds that provide information on emerging threats to enable proactive and automatic updates on the model. Part of this integration includes aggregating data gathered from multiple sources to enhance the firewall's overall detection capabilities.

3.2. Real-time traffic analysis methods in adaptive AI WAFs

Supervised machine learning is a popular technique that adaptive AI-based WAFs leverage to enhance their capabilities. In this method, the detection algorithm is trained on labeled datasets that feature known threats to allow the system to classify similar future traffic as attacks. Labeled data contains features or X variables and the target or the Y variable.

Under unsupervised anomaly detection, algorithms examine unlabeled traffic or datasets to identify anomalies and report threats without prior knowledge found in attack signatures. Instead, learning happens continuously and the WAF extracts real-time data from live traffic. This autonomous capability allows the technology to identify innate data structures that have not been labeled, such as supervised learning. The data in unsupervised learning only features the input variables X but no corresponding output variable.

Additionally, adaptive AI WAFs feature natural language processing (NLP) capabilities that enable security systems to understand, analyze and interpret human language, which is necessary for processing communication logs and text data. In NLP, adaptive AI firewalls determine the sentiment behind a text to identify potentially malicious requests. The tools categorize text data into predefined categories to streamline detection, analysis and response.

Adaptive AI-based WAFs feature machine-learning algorithms for monitoring user behavior patterns to detect and flag anomalies that could indicate compromised user accounts or insider threats. Besides user and entity behavior analysis, the technology establishes a baseline of regular network activity. Any deviations are detected and treated as malicious activities. The foundation of user behavior analysis requires collecting data such as login details (location, times and login methods); file access (the file accessed, when and by whom), application use patterns and network behavior. For instance, if a user typically logs in from China between 9 am and 9:30 am, any login attempt from the US at 2 am is flagged as suspicious. Adaptive AI WAFs use the collected data to establish baselines, such as developing a typical behavior profile for each user. For instance, the security systems analyze keystroke dynamics like typing patterns and mouse movements to differentiate legitimate users from potential cybercriminals. The tool can set thresholds, such as limiting what is considered normal behavior and creating alerts from significant deviations while users interact with protected applications.

3.3. AI-based WAFs response approach

Adaptive AI WAFs respond appropriately based on detected anomalies to mitigate cybersecurity attacks. For instance, the security tools can immediately block access or require additional authentication. Also, the WAF can send a notification alert to security teams to investigate further on the potential threats.

Adaptive AI WAFs can respond to suspicious user requests with a challenge response that requires multi-factor authentication or CAPTCHA. A reCAPTCHA rule protects web application log-in from abuse and spam. The firewall evaluates conditions for incoming traffic to allow, block or redirect requests based on the specified action. The regex below shows a reCAPTCHA firewall rule applied to the authentication function to block access if a score is less than 0.5.

```
Policy {
    Path: login.php
    Condition: recaptcha score <0.5
    Action: block
    }
```

4. LLM Models for Adaptive AI WAFs

4.1. Finetuning LLaMA 2 and mistral 7B for on-prem firewall

Finetuning LlaMA and Mistal 7B can be adapted while setting an on-premises firewall. As mentioned, a good model requires a high quality and quantity of training data. To keep the code simple, this experiment uses QLoRA and other tools, such as basci PyTorch and Hugging Face packages.

The code below is used to get the specific versions and latest tools from Hugging Face.

pip install -U accelerate bitsandbytes datasets peft transformers tokenizers

After installing the tools, the next step involves loading or crafting a dataset and formatting it based on the ChatML structure. Datasets can be sources from different sources, such as the Open Herms 2 finetune of Mistral, which contains approximately 900,000 samples from various datasets. demonstrates how you can create datasets based on podcasts or books into training sets after transforming them into a uniform format, such as ChatML structure, that works with the training tools.

The next step involves preparing the model and tokenizer to ensure they process the ChatML tags correctly. The tool torch and tokenizer can be used for this flow.

Accurate batching and tokenization ensure proper data processing. The code below specified outliers that need either modification or are blocked.

4.2. Other LLM models for AI WAFs

Rahali and Akhloufi[9] propose malware detection using Bidirectional Encorder Representations from Transformers (BERT), which performs static analysis on Android application source codes to detect malicious software. In the same way, SecBERT and MalBERT can be finetuned to develop threat intelligence tools. Once the data is prepared in the proper format, BERT is leveraged to train and test to predict threats. The transformer implementation allows the creation of unique IDs and tokens that will enable the classification.

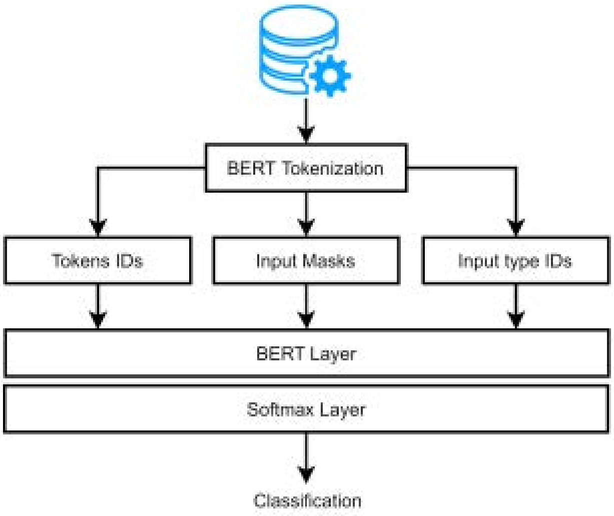

Figure 5: Finetuning process in BERT model used for threat intelligence[9].

Sanh et al.[10] introduce DistilBERT, a smaller and faster version of BERT, which is suitable for deployment in edge devices that require low-latency intrusion detection. The researchers assess the language understanding and generalization capabilities of DistilBERT on the General Language Understanding Evaluation (GLUE), which includes a collection of 9 datasets for analyzing natural language understanding systems. Their experiment shows that a general-purpose language model can be trained and analyzed, which is a compelling option for applications running on edge devices.

5. Conclusion

The shift towards adaptive AI web application firewalls from anomaly- and rule-based security systems marks a significant step that enables proactive, efficient and dynamic defense capabilities for web applications. The technology offers advanced and robust adaptive capability to security by analyzing vast traffic to identify and respond to potential threats, including zero days. Most importantly, adaptive AI WAFs feature automatic protection updates that detect new threats without overburdening security personnel.

Since adaptive AI-based WAF technology is still evolving, more research and improvement is needed to overcome drawbacks, such as recording a high false positive rate in weakly trained models. Notably, the effectiveness of AI WAFs depends on the quantity and quality of the training data, particularly for supervised learning. With an accurate and comprehensive data set, the model can improve its learning to minimize prediction errors. Additionally, there needs to be solutions to overcome the intensive resource demands and make the technology less complex and transparent in its decision-making process. While finetuning LlaMA and Mistral 7B for firewall logs, it is possible to encounter out of memory (OOM) error, which might require reduction of batch size, compromising the quality of training dataset.

Integrating adaptive AI WAFs into existing cybersecurity environments can be complicated and requires advanced specialization. AI technology also requires substantial computational resources, such as RAM, which might impact the overall performance of organizational networks.

6. References

1. Applebaum S, Gaber T and Ahmed A. “Signature-based and machine-learning-based web application firewalls: A short survey,” Procedia Computer Science, 2021;189: 359-67.
2. Torrano-Gimenez C, Perez-Villegas A and Alvarez G. “A self-learning anomaly-based web application firewall,” Computational Intelligence in Security for Information Systems, 2009: 85-92.
3. Zhang L, Zhang D, Wang C, Zhao J and Zhang Z. “ART4SQLi: The ART of SQL injection vulnerability discovery” IEEE, 2019;68: 1470-1489.
4. Luca D andrea V, Gabriele C and Giovanni L. “WAF-A-MoLE: Evading web application firewalls through adversarial machine learning,” In Proceedings of the 35th Annual ACM Symposium on Applied Computing, Brno, Czech Republic, 2020: 1745-1752.
5. Tran NT, Nguyen VH, Nguyen-Le T and Nguyen-An K. “Improving ModSecurity WAF with machine learning methods,” In: Communications in Computer and Information Science, Springer, Singapore, 2020: 93-107.
6. Hu X. “Support Vector Machine and Its Application to Regression and Classification,” 2017.
7. Miller C, Sacchet MD and Gotlib I. “Support Vector Machines and Affective Science,” Emotion Review, 2020;12.

8. George AS and George ASH. “A Brief Study on The Evolution of Next Generation Firewall and Web Application Firewall,” IJARCCE, 10: 31-37.

9. Rahali A and Akhloufi MA. “MalBERT: Malware Detection using Bidirectional Encoder Representations from Transformers,” 2021 IEEE International Conference on Systems, Man and Cybernetics (SMC), Melbourne, Australia, 2021: 3226-3231.

10. Sanh V, Debut L, Chaumond J and Wolf T. “DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter,” 2020.