

Web Application Firewall for Detecting and Mitigation of Based DDoS Attacks Using Machine Learning and Blockchain

Elva Leka^{1,2}, Luis Lamani¹, Admirim Aliti³, Enkeleda Hoxha²

¹*Polytechnic University of Tirana, Tiranë, Albania*

²*Albanian University, Department of Engineering, Faculty of Applied Sciences and Economics, Tiranë, Albania*

³*Mother Teresa University, Skopje, North Macedonia*

Abstract – Applications have rapidly transformed the data access, business operations, and communication methods. This sudden shift has resulted in significant security challenges such as Distributed Denial of Service (DDoS) attacks, which intensify Internet security issues. This paper introduces a novel approach to enhancing the Web Application Firewall (WAF) for detecting and mitigating botnet-based DDoS attacks through the use of Machine Learning (ML) and blockchain technologies. Legacy security systems often struggle to adapt to evolving digital threats, particularly with the rise of complex botnet designs. The integration of ML and blockchain within the WAF ecosystem represents a substantial advancement in cyber defense mechanisms. Insights are provided into the development of advanced ML algorithms for precise anomaly detection and the formulation of efficient blockchain protocols for streamlined threat intelligence sharing.

The proposed approach addresses current challenges associated with botnet-driven DDoS attacks establishes a foundation for adaptive, future-proof cybersecurity strategies.

Keywords – Traffic analysis, DDoS attacks, cyber threats, web application security, real-time detection.

1. Introduction

The increasing use of web applications and online services has dramatically impacted the way we do business, interact, and gather information. In addition to other common and devastating problems, such as DDoS attacks, this digital revolution has exacerbated cyber risks. By overloading web services with traffic, DDoS attacks can lead to service disruptions [1], [2], [3], significant financial losses and reputational damage for businesses and organisations. Internet of Things (IoT) devices and other hardware are infiltrated into remote botnets by fraudsters known as “botmasters”. Due to their decentralised nature, botnets are difficult to identify and remove using conventional security approaches [4].

The article describes a robust Web Application Firewall (WAF) system developed to defend against DDoS attacks by botnets using state-of-the-art technologies such as machine learning (ML) and Blockchain. The suggested WAF ensures defense against these DDoS attacks through sophisticated traffic analysis methods and exchanging crucial threat information. This article's primary contribution is the creation of a trustworthy and scalable defense against DDoS attacks based on Botnets using the integration of Blockchain and ML technology.

The proposed planning, implementation, and evaluation of the Web Application (WAF) focus on its effectiveness in detecting and mitigating botnet-driven DDoS attacks.

DOI: 10.18421/TEM134-17

<https://doi.org/10.18421/TEM134-17>

Corresponding author: Elva Leka,

¹*Polytechnic University of Tirana, Tiranë, Albania;*

²*Albanian University, Department of Engineering, Faculty of Applied Sciences and Economics, Tiranë, Albania*

Email: elva.leka@fgjm.edu.al


e.leka@albanianuniversity.edu.al

Received: 27 May 2024.

Revised: 16 September 2024.

Accepted: 21 October 2024.

Published: 27 November 2024.

 © 2024 Elva Leka, Luis Lamani, Admirim Aliti & Enkeleda Hoxha; published by UIKTEN. This work is licensed under the Creative Commons Attribution-NonCommercial-NoDerivs 4.0 License.

The article is published with Open Access at <https://www.temjournal.com/>

The primary components of this approach include:

- *Machine Learning-Based Botnet Detection:* The WAF integrates a dataset ML model (NSL-KDD) [5] trained on historical network traffic data to identify patterns associated to botnet-based DDoS attacks. Using the power of ML algorithms, the WAF can distinguish between regular user traffic and malicious botnet traffic, allowing for real-time detection and effective mitigation.
- *Integration of the Ethereum Blockchain:* The WAF effortlessly integrates the Ethereum blockchain, a decentralized and immutable ledger, to enhance communication and information sharing [6], [7]. Participating WAF instances can securely report and share detected botnet threats by utilizing a smart contract. This decentralized approach to threat intelligence allows for improved responsiveness to emerging botnet attacks.
- *User-friendly web application interface:* The proposed WAF provides a user-friendly interface that enables administrators to monitor attack status, observe attack trends, and access attack statistics. Through the Ethereum-based MetaMask login, users can securely interact with the system and gain authentication access to the WAF's functionalities.

The paper is structured as follows: Section 2 summarizes relevant research on DDoS attack detection, machine learning, and blockchain-based security. Section 3 discusses the projected WAF's architectural layout and its components. Section 4 details the implementation specifics and the integration of the Ethereum blockchain with the machine learning model. Section 5 highlights the key findings and the potential of the proposed Web Application Firewall. Finally, the conclusion summarizes the overall contribution of the paper.

2. Background

The DDoS attacks remain significant to the Internet community. This section provides an in-depth discussion of these attacks, examining their current patterns and the complex challenges they present for businesses and individuals alike.

2.1. DDoS Attacks

Web applications and online services are potentially at risk from DDoS assaults. By flooding the targeted servers and networks with overwhelming traffic, these assaults overload them, making the services inaccessible to genuine users [8]. DDoS attacks are often planned by malicious actors that use botnets and coordinated attacks of networks of hacked devices to carry out their operations [9].

Financial losses, damage to credibility, and decreased user trust are some of the adverse outcomes of DDoS assaults. DDoS attacks come in a variety of forms, such as [10], [11], [12], [13]: (a) User Datagram Protocol (UDP) floods are caused by attackers' packets flooding their targets. If attackers bombard their target with excessive Internet Control Message Protocol (ICMP) packets, the network will become less responsive; (b) Flood SYN: If an attacker floods the system with TCP ACK packets in response to valid requests, the server resources will ultimately run out; (c) HTTP/HTTPS flood: The web server's resources are depleted due to the attacker's excessive HTTP or HTTPS queries; (d) DNS: Attackers that amplify their attack traffic via open DNS servers overwhelm the target with improved DNS replies, a practice known as DNS amplification; (e) RST Flood: This attack involves sending many TCP (Transmission Control Protocol) packets with the RST (Reset) flag set to break up already-established network connections and prevent devices or services from communicating with one another. Forcefully serving connections can disrupt services and networks.

Figure 1 presents DDoS attack vector percentages during the year 2023.

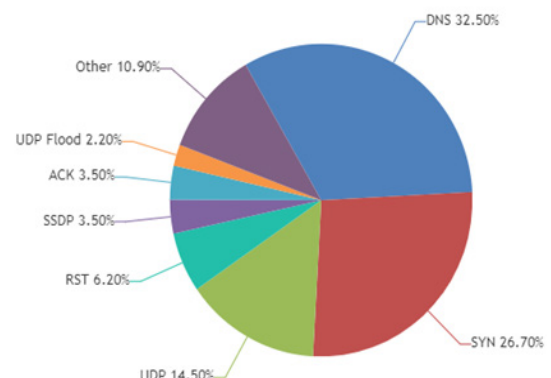


Figure 1. DDoS attack vectors in 2023

Traditional methods for mitigating DDoS attacks have primarily employed rate-based thresholds and signature-based matching to identify and diminish attack traffic [14]. These solutions provided some initial defense but are ineffective in dealing with DDoS assaults' constantly changing attack paths and obscuring strategies. As a result, there is a rising demand for more flexible and wise strategies to counter DDoS assaults successfully.

2.2 DDoS Attacks Frequency and Trend

Web DDoS system attacks are evolving and growing more prevalent. The frequency and variability of DDoS assaults will be discussed below.

Cybercriminals now have easy access to botnet rentals and DDoS for hire services on the underground internet.

The entry hurdle has been decreased, enabling more people and organisations to carry out assaults. DDoS assaults can occur for several reasons, including hacktivism or financial gain.

DDoS assaults have also become more prevalent due to a broader spectrum of targets, such as corporations, governmental institutions, and online gaming sites, as presented in Figure 2.

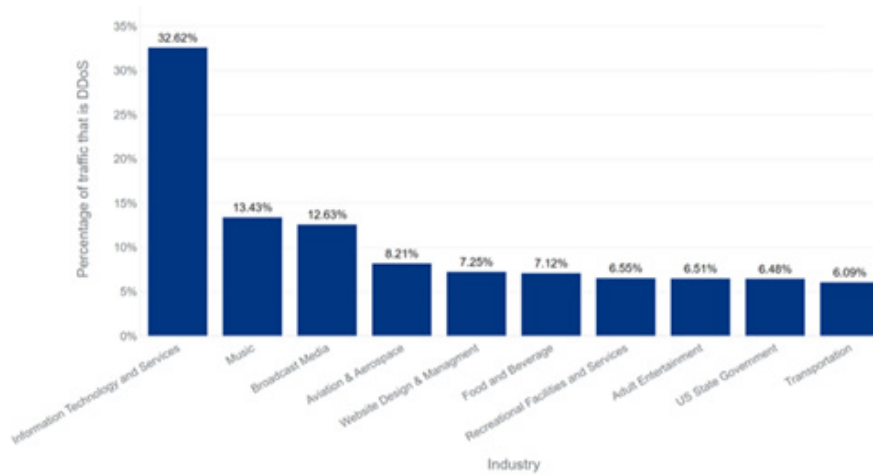


Figure 2. DDoS attacks distributed by industry [13]

In response to the evolving DDoS threat landscape, organizations are implementing sophisticated mitigation strategies such as:

Cloud-Based Protection: Many companies are increasingly relying on DDoS protection services that can intercept malicious traffic before it reaches the targeted network.

Machine Learning and AI: Security systems can detect unusual traffic patterns and instantly react to new attack vectors using machine learning and artificial intelligence.

Traffic Scrubbing: Network traffic undergoes cleaning to remove malicious packets, allowing legitimate traffic to pass through with minimal disruption.

The frequency and characteristics of DDoS attacks continue to evolve as fraudsters adjust to new technologies and security measures. To defend against these attacks, organisations must remain vigilant, implement modern mitigation techniques, and stay informed about emerging risks associated with DDoS assaults.

3. Detecting DDoS Attacks with ML and Blockchain Applications

This section examines how machine learning can enhance detection effectiveness and explores the potential applications of blockchain technology in cybersecurity.

3.1. Machine Learning for DDoS Detection

Recently, there has been considerable interest in the application of machine learning (ML) techniques for detecting DDoS attacks, as illustrated in Figure 3. Various ML algorithms capable of analyzing network traffic data and identifying patterns associated with DDoS assaults include (a) Support Vector Machines (SVMs); (b) Decision Trees; (c) and Neural Networks [15], [16]:

Support Vector Machines are a subclass of machine learning models that are advantageous for classification issues because they divide the data points using a hyperplane with the most significant degree of significance.

Decision trees are a versatile, intelligible, non-linear machine-learning technique for classification and regression issues. They employ a tree structure to make decisions depending on the qualities of the input; *The human brain* influenced the neural network family of deep learning models, sometimes called “neural networks.” These models comprise interconnected layers of nodes that execute various tasks, including image recognition and natural language processing, exceptionally well.

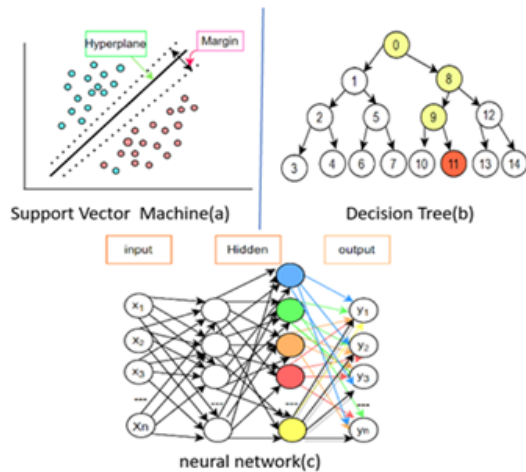


Figure 3. Artificial Intelligence techniques for detecting DDoS attacks

The ability of ML-based detection systems to dynamically alter their judgement bounds, in contrast to conventional rule-based techniques, enables adaptation to new and evolving attack strategies. Incoming traffic may be classified according to annotated valid and malicious traffic datasets using supervised ML models, which can identify odd patterns. Contrarily, unsupervised ML algorithms analyse traffic without prior categorization to detect outliers or irregularities [17]. Hybrid models, that combine supervised and unsupervised learning, enhance detection accuracy by leveraging the strength of both approaches [18].

Many researchers have used machine learning algorithms to detect DDoS attacks [19], [21]. One study [23], presents a model for detecting remote access network attacks using supervised ML methods.

Another system proposed in [24] utilizes Stochastic Gradient Descent and Support Vector Machine algorithms to evaluate malicious activities.

Those approaches improve scalability, manageability, and performance of attack detection but may introduce a single point of failure. Additionally, the paper proposes implementing machine learning-based algorithms to detect malicious traffic and offer real-time detection. Additionally, the integration of blockchain technology supports a decentralised approach to enhance responsiveness to emerging botnet attacks.

3.2. Blockchain-Based Security

Blockchain technology, the underline framework of cryptocurrency, offers various applications in cybersecurity. Its decentralised and tamper-proof characteristics provide potential solutions to numerous cybersecurity challenges [25]. In particular, the adaptation of blockchain in security applications facilitates decentralised threat intelligence sharing, secure data sharing, and authentication [26].

The Ethereum blockchain is particularly well-suited for developing decentralized security solutions due to its smart contract capabilities. Smart contracts are automated programs that enforce predefined rules and conditions, ensuring secure interactions without intermediaries [20]. Written in a language such as Solidity, these contracts run on the Ethereum Virtual Machine, ensuring they execute as intended. Once deployed, smart contracts are immutable and can interact with one another, making them ideal for automated, reliable security applications [22]. Figure 4 illustrates the workflow diagram of smart contracts.

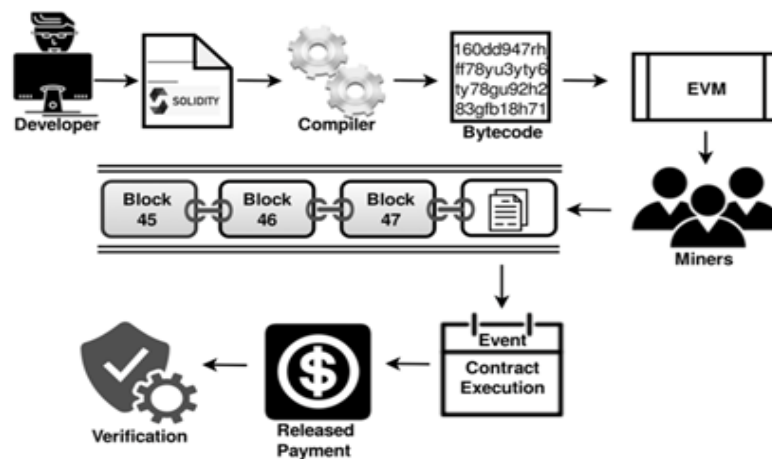


Figure 4. The workflow diagram of Smart Contract [19]

4. Proposed Web Application Firewall Architecture

The proposed Web Application Firewall (WAF) integrates Blockchain and ML technologies to enhance the detection and mitigation of botnet-based DDoS assaults. This section outlines the projected WAF’s architectural framework of the WAF and

details the essential components and their interactions. The architecture consists of three main elements: the Ethereum blockchain, the Backend Server, and the Frontend Web Application. Together, these elements establish a robust security system to combat botnet-based DDoS attacks. Figure 5 presents the workflow diagram illustrating the proposed WAF architecture.

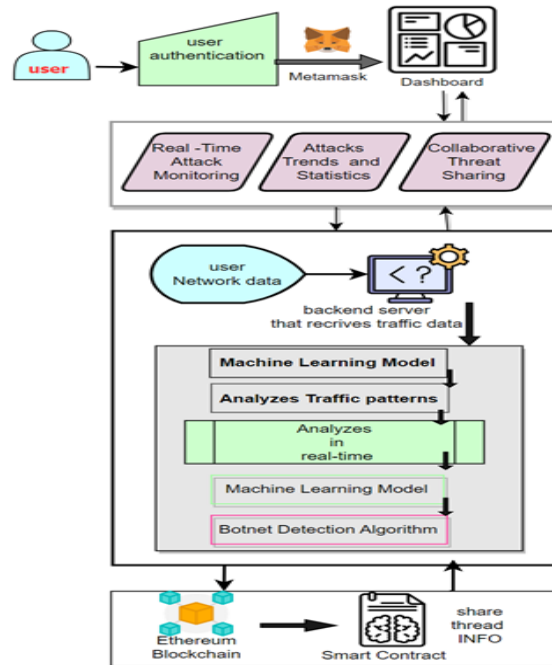


Figure 5. The workflow diagram architecture

The workflow we propose follows the steps as presented below:

1. User Authentication: When a user views the front-end web application, MetaMask integration is used to carry out Ethereum-based authentication. The user has access to the WAF dashboard after successful authentication.
2. Traffic Monitoring and Analysis: The backend server receives data about network traffic from the user. Real-time traffic analysis using a machine learning model identifies abnormalities and deviations from expected traffic patterns.
3. Botnet Detection: The machine learning model assesses whether the traffic displays signs of DDoS attacks powered by botnets. The ML model notifies the backend server whenever a potential attack is found.
4. Threat Reporting to the Ethereum Blockchain: The backend server communicates with the Ethereum blockchain to alert an intelligent contract to an identified threat. Information about dangers is safely kept on the blockchain.
5. Decentralised Threat Sharing: The Ethereum blockchain is accessible to several WAF instances on the network. The smart contract provides participating instances with threat information, such as attack patterns and characteristics.
6. Collaborative Defense: The centralised WAF instances employ shared threat information for real-time analysis. The WAF instances can change their defenses and stop new attacks. The partnership improves the WAF network’s overall resilience.
7. User Interaction and Monitoring: Users view the WAF dashboard using the front-end web application. The dashboard displays real-time attack status, trends, and statistics to show the collective efforts to mitigate assaults.
8. Ongoing Learning and Adaption: The WAF system can increase its ability to recognise emerging attack techniques due to the machine learning model’s ongoing learning from new traffic data and dynamic assault patterns. The blockchain also allows for the safe storage of intelligence regarding potential threats.

5. Implementation Details

Architecture encompasses real-time network traffic analysis, machine learning-based DDoS attack detection, and blockchain-backed threat sharing. The architecture of the proposed WAF framework presented in the previous session visually conveys how data flows through the system and undergoes various processing stages. This high-level representation illustrates the core components and their interactions.

5.1. Data Collection and Analysis

The foundation of DDoS detection capabilities is built on data gathering and analysis. The following procedures are applied to handle network traffic data efficiently: First, packet capture techniques are utilized to compile network traffic data from various sources. This raw data is continuously monitored to provide insights in real-time.

Second, extracting key characteristics from the raw data is essential for executing the machine learning-based detection technique properly. To optimize the training of the models, emphasis is placed on critical factors such as packet rates and the distributional patterns of packet sizes. The architecture employs machine learning methods to identify DDoS attacks. By training and integrating diverse models, such as Support Vector Machines (SVMs) and Convolutional Neural Networks (CNNs), precision in detection is enhanced.

The pseudocode provides a concise overview of the standard procedure for using a training dataset in machine learning. The procedure encompasses data loading, partitioning into distinct features and labels, selecting and training a model, optionally evaluating the model, optionally serialising the model, generating predictions on test data, and performing post-processing on the predictions. Algorithm 1 presents the pseudocode of workflow training dataset in machine learning.

Algorithm 1. Pseudocode for using a train dataset in machine learning

```
// Load the training dataset from a data source
training_data = load_training_data("path/to/nsl_kdd_train.csv")

// Split the dataset into features (X) and labels (Y)
X_train = training_data.drop(columns=["target_column"])
y_train = training_data["target_column"]

// Choose a machine learning algorithm (e.g., decision tree, neural network)
selected_model = select_machine_learning_model()
trained_model = train_model(selected_model, X_train, y_train)

// Load a validation dataset for evaluating model performance (if available)
validation_data = load_validation_data("path/to/validation_dataset.csv")
X_val = validation_data.drop(columns=["target_column"])
y_val = validation_data["target_column"]

// Evaluate the trained model's performance on the validation dataset
evaluation_metrics = evaluate_model(trained_model, X_val, y_val)
save_model(trained_model, "path/to/trained_model.pkl")
test_data = load_test_data("path/to/test_dataset.csv")
X_test = test_data.drop(columns=["target_column"])

// Use the trained model to predict on the test dataset
predictions = trained_model.predict(X_test)
post_processed_predictions = post_process_predictions(predictions)
save_predictions(post_processed_predictions, "path/to/predictions.csv")
```

The NSL-KDD dataset has become a prominent standard for assessing network intrusion detection systems in the scientific community. This dataset is essential for training, evaluating, and appraising machine learning models to enhance cybersecurity. After careful consideration, a machine learning model was selected and trained using the training dataset, emphasising its robustness and ability to adjust to the intricate patterns in network traffic data.

Additionally, several further steps have been considered, including evaluating the model's performance on a validation dataset (if available) preparing the model for future use, and the crucial procedures of generating predictions and conducting post-processing on the test dataset. The utilisation of the NSL-KDD dataset and its comprehensive methodology provides a solid foundation for addressing challenges in network intrusion detection and advancing the field of cybersecurity.

5.2. Real-Time Analysis and Mitigation

The algorithm activates quick-response measures upon the discovery of a potential DDoS attack. Mitigation techniques, such as traffic filtering, rate limiting, and resource scaling, are implemented to absorb attack traffic.

Algorithm 2. Practical implementation pseudocode for DDoS attack detection

```

// Define data structures
Initialize attackRecords as empty list
Initialize trafficLogs as empty list

// Define parameters and thresholds
Set attackThreshold = 1000
Set analysisInterval = 60 seconds
Set detectionWindow = 5 minutes

// Function to collect and preprocess traffic data
Function collectAndPreprocessTraffic():
    Initialize trafficData as empty list
    Append preprocessedData to trafficData
    Return trafficData

// Function to analyze traffic logs for DDoS attacks
Function analyzeTrafficLogs(trafficLogs, threshold,
detectionWindow):
    Initialize detectedAttacks as empty list
    Initialize attackWindow as empty list
    Initialize currentTime as current time

// Iterate through traffic logs
    For each logEntry in trafficLogs:
        If logEntry.timestamp >= currentTime -
detectionWindow:
            Append logEntry to attackWindow

// Calculate the total requests in the attackWindow
totalRequests = sum(requests for logEntry in
attackWindow)

// Check if the total requests exceed the threshold
If totalRequests > threshold:
    attack = {
        "start_time": currentTime - detectionWindow,
        "end_time": currentTime,
        "total_requests": totalRequests,
        "source_ip_addresses":
extractSourceIPs(attackWindow)
    }
    Append attack to detectedAttacks
// Return the list of detection attacks
Return detectedAttacks

// Main loop for continuous operation
While true:
    trafficData = collectAndPreprocessTraffic()
    Append trafficData to trafficLogs
    If timeElapsed(analysisInterval):

        detectedAttacks = analyzeTrafficLogs(trafficLogs,
attackThreshold, detectionWindow)
        Append detectedAttacks to attackRecords
        Clear trafficLogs
// Sleep for a short duration before the next iteration
Sleep for 10 seconds

```

This paper presents a practical approach for identifying DDoS attacks to enhance cybersecurity. The strategy highlights continue data gathering and analysis to detect potential DD0S attacks by monitoring incoming web traffic. The system is structured around two fundamental data structures: **attackRecords** for storing detailed attack information and **trafficLogs** for housing incoming traffic data. Key parameters, including the **attackThreshold**, **analysisInterval**, and **detectionWindow**, are defined to tailor the system's sensitivity to attacks. A critical component of this system is the **collectAndPreprocessTraffic** function, which gathers raw traffic data from network sources and preprocesses it to extract pertinent information. The core of the system lies in the **analyzeTrafficLogs** function, which evaluates the traffic logs to ascertain the presence of a potential DDoS attack. When an attack is identified, an attack record is meticulously constructed and added to the **attack Records**. This comprehensive approach provides practical and continuous DDoS detection capabilities. Practically, actual data sources and advanced algorithms would be integrated to enhance the accuracy and speed of detection.

5.3. Blockchain Integration

The Ethereum blockchain is utilized to securely store data and facilitate cooperative threats sharing. Smart contracts simplify the process of sharing critical information among the participating instances of the system. Algorithm 3 presents the pseudocode of the web application firewall smart contract.

Algorithm 3. The pseudocode of the “Web Application Firewall” smart contract

```

// Smart Contract

contract WebApplicationFirewall {
    address owner;
    mapping(address => bool) authorizedUsers;
    uint256 totalAttacks;
    uint256 blockedAttacks;

    constructor() {
        owner = msg.sender;
        totalAttacks = 0;
        blockedAttacks = 0;
    }
    // Function to authorize a user
    function authorizeUser(address user) public onlyOwner {
        authorizedUsers[user] = true;
    }
    function deauthorizeUser(address user) public onlyOwner
    {
        authorizedUsers[user] = false;
    }

    // Function to check if a user is authenticated
    function isUserAuthorized(address user) public view
    returns (bool) {
        return authorizedUsers[user];
    }
    // Function to record a detected attack
    function recordAttack() public {
        require(authorizedUsers[msg.sender], "You are not
        authorized to perform this operation");
        totalAttacks++;
    }

    // Function to record a blocked attack
    function recordBlockedAttack() public {
        require(authorizedUsers[msg.sender], "You are not
        authorized to perform this operation");
        blockedAttacks++;
    }

    // Function to restrict access to the contract owner
    modifier onlyOwner() {
        require(msg.sender == owner, "Only the owner can
        perform this operation");
    }
}

```

The smart contract “WebApplicationFirewall” serves as a foundational component for enhancing the security of web applications by encapsulating critical functionalities necessary for detecting and mitigating security threats. At its core, this contract maintains state variables, including the contract owner’s address, a mapping of authorised users, and counters for total detected and blocked attacks. Upon deployment, the constructor initialises the contract, designating the deploying entity as the owner and initializing the attack counters.

The functions ‘*authorizeUser*’ and ‘*deauthorizeUser*’ allow the owner to grant or revoke authorisation for specific users to perform designated operations, while the ‘*isUserAuthorized*’ function enables any party to verify their authorisation status.

Additionally, the smart contract logs detected and blocked attacks through the ‘*recordAttack*’ and ‘*recordBlockedAttack*’ functions ensuring that only authorised users can invoke these operations. The ‘*onlyOwner*’ modifier restricts access to certain methods, allowing only the contract’s owner to perform critical management functions. This smart contract establishes a robust foundation for secure web application management and can be deployed across various cybersecurity scenarios to strengthen defenses against cyber threats.

Before detailing the specific results, a brief overview of the performance measures used to evaluate WAF system is presented. These metrics provide a numerical assessment of the system’s effectiveness and efficiency. Among our evaluation criteria are:

- The term “*detection accuracy*” in a WAF refers to accurately identifying malicious traffic while lowering false positives.
- **Response Time:** The duration required by the Web Application Firewall (WAF) to react to identified threats, aiming to minimize their impact on the performance of web applications.
- **Scalability:** The WAF’s ability to handle increasing traffic loads without significantly impacting performance.
- **Adaptability:** The ability of the system to adjust to evolving attack patterns while maintaining security.
- **Resource utilization:** The efficient use of a computer’s CPU and memory.

Continuous improvement and adaptation of the WAF to address the changing web security requirements remains a top priority.

6. Results

The research identified a comprehensive strategy to enhance cybersecurity against botnet-based DDoS attacks. This approach combines advanced technologies such as machine learning and blockchain. This study demonstrates that an integrated Web Application Firewall (WAF) system can effectively identify and counteract botnet-powered DDoS attacks through experimentation and analysis. Machine learning models have demonstrated an exceptional capability in detecting abnormal patterns in incoming traffic indicative of botnet activities.

The application of blockchain technology, in conjunction with machine learning, has improved the security of our WAF system by enhancing the integrity of attack logs and establishing tamper-resistant audit trails.

Based on the initial evidence gathered, the combination of these technologies has resulted in a notable decrease in false positive and false negative identifications of events, thereby improving the overall precision of attack detection. Additionally, the research clarified the inherent benefits of utilizing a decentralised and immutable ledger, such as blockchain, for documenting and verifying network occurrences. This functionality improves the capacity to oversee and identify attack efforts while strengthening the system's resistance against attempts to modify or delete crucial log entries.

Although our findings are positive, they also highlight the complex nature of cybersecurity and the ongoing development of botnet-based DDoS strategies. Consequently, this research establishes a foundational framework that represents a significant advancement toward stronger DDoS mitigation solutions.

The immediate actions involve implementing and testing this integrated WAF system in real-world scenarios to evaluate its effectiveness in dynamic and hostile settings. It is asserted that these discoveries present new opportunities for integrating machine learning and blockchain in cybersecurity, moving toward a more secure digital environment capable of withstanding the ever-changing of botnet-based DDoS attacks.

7. Conclusion

The theoretical exploration into the design and principles of a Web Application Firewall (WAF) has laid the groundwork for innovative advancements in web application security. While the project remains theoretical, it offers a fresh perspective on WAF architecture that emphasizes adaptability and proactivity in addressing evolving threats. The contributions presented encompass a robust theoretical framework that redefines how web applications can be safeguarded. As this phase of theoretical exploration, the focus will shift toward practical implementation and further theoretical development. The next steps involve transforming these theoretical constructs into functional prototypes, validating the principles in real-world scenarios, and refining our security models. The evolving nature of cyber threats necessitates a proactive and forward-thinking approach, positioning this project to contribute to the ongoing mission of fortifying web applications in a dynamic digital landscape.

References:

- [1]. Aldhyani, T. H., & Alkahtani, H. (2023). Cyber security for detecting distributed denial of service attacks in agriculture 4.0: Deep learning model. *Mathematics*, 11(1), 233.
- [2]. Avcı, İ., & Koca, M. (2023). Predicting DDoS Attacks Using Machine Learning Algorithms in Building Management Systems. *Electronics*, 12(19), 4142.
- [3]. Saeed, M. M., Mohammed, H. N. R., Gazem, O. A. H., Saeed, R. A., Morei, H. M. A., Eidah, A. E. T., ... & Al-Madhagi, M. G. Q. (2023, October). Machine Learning Techniques for Detecting DDOS Attacks. In *2023 3rd International Conference on Emerging Smart Technologies and Applications (eSmarTA)*, 1-6. IEEE.
- [4]. Alieyan, K., Almomani, A., Abdullah, R., Almutairi, B., & Alauthman, M. (2021). Botnet and Internet of Things (IoTs): A definition, taxonomy, challenges, and future directions. In *Research Anthology on Combating Denial-of-Service Attacks*, 138-150. IGI Global.
- [5]. Prasanna, I. P., & Maraiappan, S. (2020). Detection of distributed denial of service attack using NSL-KDD dataset – A Survey. In *Proceedings of the International Conference on Computer Networks, Big Data and IoT (ICCBi – 2019)*. Springer.
- [6]. Buterin, V. (2014). A next-generation smart contract and decentralized application platform. *White paper*. Retrieved from: <https://ethereum.org/en/whitepaper/> [accessed: 18 April 2024].
- [7]. Leka, E., Hoxha, E., & Rexha, G. (2023). Security and privacy concerns associated with the Internet of Things (IoT) and the role of adapting blockchain and machine learning - A systematic literature review. In the *46th MIPRO ICT and Electronics Convention (MIPRO)*, Opatija, Croatia, 1690-1696. IEEE.
- [8]. Mirkovic, J., & Reiher, P. (2004). A taxonomy of DDoS attack and DDoS defense mechanisms. *ACM SIGCOMM Computer Communication Review*, 34(2), 39-53.
- [9]. Zargar, S. T., Joshi, J., & Tipper, D. (2013). A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks. *IEEE communications surveys & tutorials*, 15(4), 2046-2069.
- [10]. Rahamuthullah, U., & Karthikeyan, E. (2021). Distributed denial of service attacks prevention, detection and mitigation – A Review. *Proceedings of the International Conference on Smart Data Intelligence (ICSMDI 2021)*. Springer.
- [11]. Sambangi, S., & Gondi, L. (2020). A machine learning approach for DDoS (Distributed Denial of Service) attack detection using multiple linear regression. *Proceedings of 14 International Conference on Interdisciplinarity in Engineering – INTER-ENG 2020*, 63(1), 51.
Doi: 10.3390/proceedings2020063051.
- [12]. Vanitha, K. S., Uma, S. V., & Mahidhar, S. K. (2017, December). Distributed denial of service: Attack techniques and mitigation. In *2017 International Conference on Circuits, Controls, and Communications (CCUBE)*, 226-231. IEEE.

- [13]. CloudFlare Radar. (2023). *DDoS attack*. Home page. Retrieved from: <https://radar.cloudflare.com/press> [accessed: 20 April 2024]
- [14]. Adedeji, K. B., Abu-Mahfouz, A. M., & Kurien, A. M. (2023). DDoS attack and detection methods in internet-enabled networks: Concept, research perspectives, and challenges. *Journal of Sensor and Actuator Networks*, 12(4), 51.
- [15]. Ali, T. E., Chong, Y. W., & Manickam, S. (2023). Machine learning techniques to detect a DDoS attack in SDN: A systematic review. *Applied Sciences*, 13(5), 3183.
- [16]. Arshi, M., Nasreen, M. D., & Madhavi, K. (2020). A survey of DDOS attacks using machine learning techniques. *E3S Web of Conferences*, 184, 01052. EDP Sciences.
- [17]. Kumar, K. & Mrunalini, M. (2022). Detecting Denial of Service attacks using machine learning algorithms". *Journal of Big Data*, 9(1), 56. Springer Open.
- [18]. Garg, U., Kaur, M., Kaushik, M., & Gupta, N. (2021). Detection of DDoS attacks using semi-supervised based machine learning approaches. In *2021 2nd International Conference on Computational Methods in Science & Technology (ICCMST)*, Mohali, India, 112-117. IEEE.
- [19]. Sueshkumar, S., Prasanna, G. K.D., Venkatesan, R. (2023). Detection of DDOS attacks on cloud computing environment using altered convolutional deep belief networks. In *International Journal of Computer Network and Information Security (IJCNIS)*, 15(5), 63-72. MECS Press.
- [20]. Leka, E., Selimi, B., & Lamani, L. (2019). Systematic literature review of blockchain applications: Smart contracts. In *Proceedings of the Digest of the 33rd International Conference on Information Technology (InfoTech-2019)*, 1-3, Varna, Bulgaria. IEEE.
- [21]. Dasari, K. B., & Devarakonda, N. (2022). Detection of DDoS attacks using machine learning classification algorithms. *International Journal of Computer Network and Information Security (IJCNIS)*, 14(6), 89-97. MECS Press.
- [22]. Leka, E., Lamani, L., & Hoxha, E. (2023). Using blockchain technology for ID management: a case study for Albania. *Industry 4.0*, 7(6), 213-218. STUME Journals.
- [23]. Ndichu, S., McOyowo, S., Okoyo, H., & Wekesa, C. (2023). Detecting remote access network attacks using supervised machine learning methods. *International Journal of Computer Network and Information Security (IJCNIS)*, 2, 48-61. MECS Press.
- [24]. Umamaheswari, K., Subramanian, N., & Subramanian, M. (2023). Distributed Denial of Service Attack Detection Using Hyper Calls Analysis in Cloud. *International Journal of Computer Network and Information Security*, 15(4), 61-71.
- [25]. Habib, G., Sharma, S., Ibrahim, S., Ahmad, I., Qureshi, S., & Ishfaq, M. (2022). Blockchain technology: benefits, challenges, applications, and integration of blockchain technology with cloud computing. *Future Internet*, 14(11), 341.
- [26]. Deshmukh, A., Sreenath, N., Tyagi, A. K., & Abhichandan, U. V. E. (2022). Blockchain enabled cyber security: A comprehensive survey. *2022 International Conference on Computer Communication and Informatics (ICCCI)*.