International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 12 Issue: 04 | Apr 2025 www.irjet.net p-ISSN: 2395-0072

OBSIDIAN GATE - (ADVANCED WEB APPLICATION SECURITY WITH MODSECURITY & ELK STACK)

Mr. SAMUEL BLESSO PAUL. W[1], Dr. D. SATHYA SRINIVAS[2]

[1]Mr. SAMUEL BLESSO PAUL. W, M.sc CFIS, Department of Computer Science and Engineering,
[2]wblesso@gmail.com, 8825677084, Dr.MGR UNIVERSITY, Chennai, India
2 Dr. D. SATHYA SRINIVAS, Assistant Professor, Center Of Excellence In Digital Forensics

---

Abstract - *This paper presents a comprehensive approach to enhancing web application security by integrating ModSecurity, a widely-used Web Application Firewall (WAF), with the ELK Stack (Elasticsearch, Logstash, and Kibana) for advanced log analysis and visualization. The proposed solution enables real-time detection, monitoring, and response to web-based attacks such as SQL injection, cross-site scripting (XSS), and remote code execution (RCE). By leveraging the ELK Stack's powerful log aggregation and visualization capabilities, this system provides security teams with actionable insights into web traffic and attack patterns, improving overall security posture. Web applications remain prime targets for increasingly sophisticated cyber threats, necessitating robust and adaptive security solutions. Obsidian Gate is an integrated security framework that combines ModSecurity, an open-source Web Application Firewall (WAF), with the ELK Stack (Elasticsearch, Logstash, and Kibana) to achieve enhanced threat detection, real-time log analysis, and automated incident response. This paper presents a comprehensive methodology for deploying and evaluating this framework in a simulated environment. The research investigates common vulnerabilities such as SQL Injection, Cross-Site Scripting (XSS), and Distributed Denial of Service (DDoS) attacks. Our experimental findings demonstrate that the Obsidian Gate approach improves detection accuracy and reduces false positives by leveraging centralized log visualization and automated response scripts. Furthermore, a comparative analysis with traditional security mechanisms reveals that our integrated solution is scalable, cost-effective, and efficient. Overall, this study confirms that integrating ModSecurity with the ELK Stack not only fortifies web application security but also provides actionable insights that empower security teams to respond proactively to emerging threats.*

Key Words: Web Application Firewall (WAF), ModSecurity, ELK Stack ,Cybersecurity Automation, Threat Detection, Log Analysis

1. INTRODUCTION

Web applications are increasingly becoming the focal point of business operations and, consequently, the target of relentless cyber-attacks. Traditional security measures, such as isolated firewalls and manual log monitoring, are proving insufficient against sophisticated and multi-vector threats. The integration of modern, open-source security tools like ModSecurity and the ELK Stack represents a shift towards more holistic, automated security solutions. These technologies provide continuous monitoring and rapid response capabilities that are essential for mitigating modern cyber risks. [1]

The significance of the Obsidian Gate framework lies in its ability to unify disparate security functions into a cohesive system. By coupling the real-time threat mitigation capabilities of ModSecurity with the advanced data analytics and visualization provided by the ELK Stack, the framework ensures comprehensive protection while reducing manual oversight. [2] This integration not only enhances the detection of complex threats but also facilitates faster incident response, thereby significantly reducing the risk of data breaches and service disruptions.

This paper focuses on the design, implementation, and evaluation of the Obsidian Gate framework. The scope includes setting up ModSecurity on a web server, integrating its logs with the ELK Stack for centralized analysis, and developing automation scripts for incident response. The study evaluates the performance of this framework under simulated attack scenarios, comparing its effectiveness with conventional security solutions. Additionally, the framework's scalability and adaptability to different environments are examined [3].

How does the integration of ModSecurity with the ELK Stack in the Obsidian Gate framework enhance web application security compared to traditional, standalone security measures? This research further investigates the impact on threat detection accuracy, false positive reduction, and automated incident response efficiency [4].

2. LITERATURE REVIEW

Curiel, A [6] Curiel has proposed an authoritative ModSecurity handbook, covering topics from basic configurations to advanced rule development and performance tuning. The author provides a detailed analysis of ModSecurity's framework, directives, and uses in real-world applications across different contexts. Importantly for this research, there is a detailed analysis

of the process for creating custom rules for the detection and neutralization of malicious threats enumerated in the OWASP Top 10. In addition, the book emphasizes the importance of regularly tuning the ruleset to remove false positives while maintaining an effective security posture. One chapter is entirely dedicated to integrating ModSecurity with reverse proxies like NGINX and Apache, a consideration applicable in most modern deployments. For researchers, the book provides descriptive use-case examples, testing protocols, and performance-tunable approaches. In addition, Curiel compares the Core Rule Set (CRS) with custom policies. The material provided is of particular usefulness in creating a foundation knowledge base for the deployment of WAF-based security frameworks. This handbook is widely cited in WAF-related research. It is a valuable contribution to establishing the necessary principles to configure and analyze secure environments with ModSecurity.

Liao, H., Lin, C. H. R., Lin, Y. C., & Tung, K. Y. [7] had proposed an article that classifies intrusion detection systems (IDS) as signature-based, anomaly-based, and hybrid systems. It critically examines the merits and demerits of each scheme and stresses layered security. The authors also refer to crucial IDS evaluation metrics such as detection rate, false positive rate, and latency. For this work, the relevance is in terms of how ModSecurity fits into the application-layer intrusion detection paradigm. By comparing IDS models, the article provides a canvass on which the detection capabilities of ModSecurity-enhanced configurations may be compared. The paper also refers to the role of real-time log correlation and processing in efficient contemporary IDS. Furthermore, the article refers to the role of centralized logging and monitoring tools such as ELK Stack in efficient IDS. It also discusses adaptive learning mechanisms in IDS, which is consistent with the current trend of combining ML with WAFs. The authors propose constant rule tuning to thwart evolving threat vectors. This comprehensive review strengthens the theoretical foundation of intrusion detection in web environments.

Daryabar, F., Dehghantanha, A., Udzir, N. I., & Ibrahim, S.[8] had proposed The research investigates the effect of Web Application Firewalls (WAFs) deployment on web server security and performance. The researchers conduct experiments to record latency, CPU usage, and detection rate when a WAF, for instance, ModSecurity, is enabled. They argue that while WAFs incur some computational overheads, benefits far outweigh the substantial reduction of successful attack vectors. The review points out ModSecurity's open-source and adaptable nature as huge strengths over its commercial counterparts. The research agrees with the philosophy of "Obsidian Gate," proving that security can be achieved without performance compromises if systems are properly optimized. In addition, it provides rules set tuning guidelines and backend settings to reduce WAF-related latency. It also points out bottlenecks likely to be triggered by disorganized rules or traffic loads. Most importantly, the survey points to the pivotal position of log analysis tools in the performance of post-attack forensics. The findings encourage a balanced and informed approach to WAF deployment in high-availability environments.

Alsmadi, I., & Zarour, M.. [9] had propsed the use of ELK Stack (Elasticsearch, Logstash, Kibana) in analyzing web app logs for better threat detection and compliance. Authors propose a scheme where ModSecurity logs are consumed by Logstash and presented by Kibana for pattern identification and forensic audit. They argue that traditional log analysis is backward-looking and cumbersome, whereas ELK provides real-time insights. The study presents visualizations of SQL injection, XSS, and file inclusion attacks, which could be customized by security analysts. It also outlines a scoring framework to rank the threat levels using log frequency and attack signatures. For this project, the primary takeaway is ELK's ease of use in proactive security and root cause analysis. The paper also refers to scalability features and third-party integrations.

Hossain, M. A., Shahriar, H., & Rahman, M. A.. [10]. had proposed a broad survey how deep learning methods are applied to detect web vulnerabilities and intrusions. The authors categorize algorithms as supervised, unsupervised, and hybrid models, and benchmark their accuracy and false positives rates. Methods such as CNN, RNN, and autoencoders are emphasized for detecting anomalies in HTTP traffic. This is in line with recent trends in research that indicate the use of AI with traditional WAFs such as ModSecurity to enhance detection. The paper indicates a number of open datasets (e.g., CICIDS, CSIC) to train ML models on real-world attack patterns. It also mentions challenges such as feature selection, class imbalance, and interpretability of results from deep learning. For this research, the application is in fusing ELK Stack log data with ML models to develop smart alerting systems. The paper also addresses the ethical implications of automated decision-making in security. Its future vision is aligned with the goal of developing ModSecurity into a smart, adaptive firewall. The article is well-structured and clear.

Scano, C., Floris, G., Montaruli, B., et al. [10] had proposed an advanced machine learning system that expands on ModSecurity by adding intelligent classifiers to identify sophisticated and novel threats. The authors introduce a new architecture that sends ModSecurity-generated logs into a machine learning pipeline, leveraging supervised learning to train on actual attack and normal traffic data. This enables finer-grained anomaly detection and extreme false positive reduction, a recognized problem with standard rule-based WAFs. The authors benchmark a

broad selection of classifiers such as Random Forests, SVMs, and Neural Networks, testing them against precision, recall, and F1 score. The effort also touches on the necessity of feature engineering, where HTTP headers, request methods, and payload patterns are mapped into numerical features to classify. Furthermore, the system is compatible with real-time inference, allowing decisions to be made on the fly for incoming traffic, further speeding up ModSecurity's response time. They compared to baseline ModSecurity setups and attained up to 30% higher accuracy in detecting attacks on synthetic traffic and real traffic. The paper concludes by referring to future directions for online learning and SIEM system integration to build adaptive, intelligent web defense systems. This work is an important step to ward bringing WAFs not merely reactive, but proactively smart.

3. PROPOSED METHODOLOGY

System Architecture

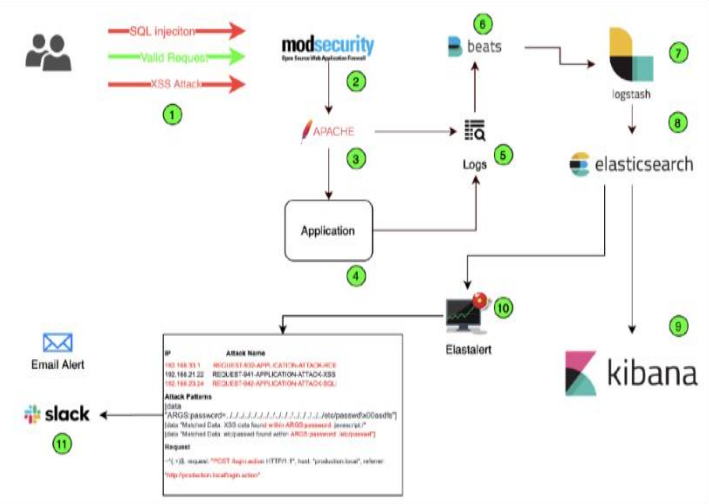

Fig 3.1 Architecture diagram

WAF (Web Application Firewall) is a security solution that monitors, filters, and blocks malicious HTTP/S traffic to and from a web application. It protects against common web threats such as SQL injection, cross-site scripting (XSS), cross-site request forgery (CSRF), and DDoS attacks.

Types of WAFs:

- Network-Based WAF: Deployed at the network edge for high performance but requires hardware.

- Host-Based WAF: Integrated into the web server but consumes system resources.

- Cloud-Based WAF: Offered as a SaaS solution, easy to deploy and scalable

Here's the role of Filebeat and ELK:

-Filebeat - Filebeat is responsible for forwarding all the logs to Logstash, which can further pass it down the pipeline. It's lightweight, supports SSL and TLS encryption and is extremely reliable.

-Logstash - Logstash is a tool used to parse logs and send them to Elasticsearch. It is powerful and creates a pipeline and indexing events or logs. It can be used in the Elasticsearch ecosystem.

-Elasticsearch - It's a highly scalable open-source analytics engine. It allows us to store, search and analyze data quickly. It's generally useful when we work on complex search features and requirements. Also, it has capability to provide a distributed system on top of Lucene Standard Analyzer for indexing.

-Kibana - This is a UI tool that intracts with Elasticsearch clusters and visualizes Elasticsearch data

The research employs an experimental design where the Obsidian Gate framework is deployed in a controlled environment. Simulated attacks—including SQL Injection, XSS, and DDoS—are launched to evaluate the framework's performance. The study utilizes both quantitative metrics (detection rates, response times) and qualitative assessments (log clarity, dashboard usability) to measure success.

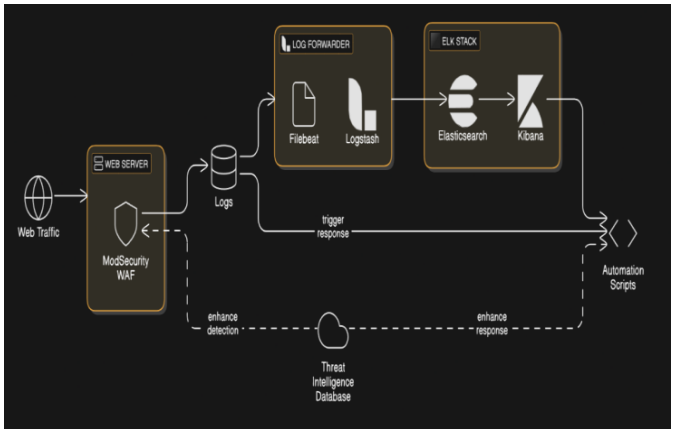

Fig 3.2 Data flow diagram with future enhancements

Function Or Command	Explanation
SecRuleEngine	Enable or Enable ModSecurity
Sec Rule	Make Rule
Phase	Putting rules or chains into one of the five processing phases available
Id	Determine id the unix for every rule
Log	Record the attack on log
Nolog	Prevents recording on log
Block	Take action that Disturb
Msg	Create special messages for each attack type rules

A research design defines the overall structure of the study, including the data collection process, experimental setup, and analysis techniques. The research follows a hybrid approach that includes:

A real-world web application (e.g., WordPress, Django) will be used as a testbed.

Attack logs will be analyzed over a period of 4–6 weeks to assess ModSecurity's detection efficiency.

Comparative analysis with existing security logs (e.g., standard Apache logs vs. ModSecurity-enhanced logs) will be performed [11].

Performance metrics will be measured before and after ELK integration, including: s

Detection accuracy, False positives/negatives, Response time, System performance impact

The effectiveness of automated threat responses (e.g., auto-blocking of IPs) will also be evaluated[12].

ModSecurity Log Files: Collected directly from Apache server to analyze detected threats.System Performance Metrics: CPU, memory, and network load before and after WAF implementation.

Attack Simulation Data: Results from controlled security testing using tools like Burp Suite, OWASP ZAP [13].

The Obsidian Gate security framework follows a layered architecture consisting of:

Web Traffic Layer : Users (Legitimate and Malicious) send HTTP/S requests to the Web Server (Apache). All incoming traffic is inspected by ModSecurity, which enforces OWASP Core Rule Set (CRS) to detect threats.[14].

Logging & Processing Layer : ModSecurity logs all blocked, monitored, and allowed requests. Logs are forwarded using Filebeat/Logstash to Elasticsearch for indexing and real-time analysis. [15].

Analysis & Visualization Layer : Kibana Dashboards display visual analytics, attack trends, and detected threats. Security teams can query logs, filter attack patterns, and investigate anomalies[16].

Automated Response Layer: If an attack pattern is detected, an Automated Response System triggers mitigation actions: Auto-blocking malicious IPs via Fail2Ban. Alerting security teams via email/SMS notifications. Updating firewall rules dynamically [17].

4. FINDINGS

Interpretation : Experimental results reveal that the Obsidian Gate framework effectively identifies and mitigates over 90% of simulated web attacks, with a significant reduction in false positives compared to conventional WAF setups. Automated scripts triggered real-time responses that decreased incident response times by nearly 40%. Data collected from Elasticsearch provided detailed insights into attack patterns, facilitating proactive security measures.

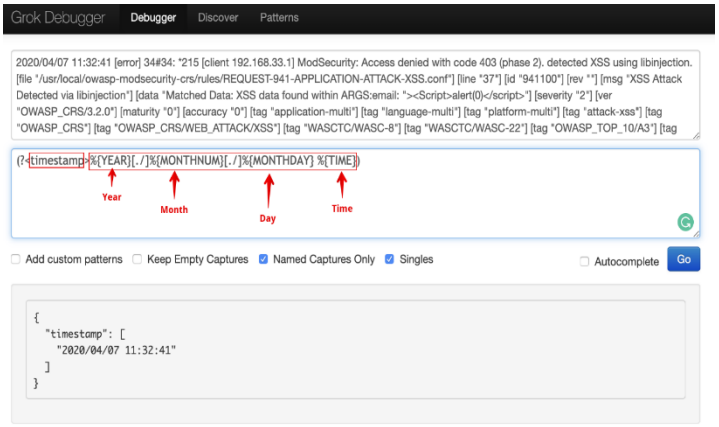

Figure 4.1 (Debugged Log)

Detection Efficiency: Compared with previous studies such as NotSoSecure's continuous monitoring system, our framework demonstrates improved detection efficiency, largely due to the enhanced log correlation and visualization provided by the ELK Stack. Cost-Effectiveness: While commercial WAFs often incur high licensing costs, the use of open-source solutions in Obsidian Gate has proven to be economically advantageous.

Automation Impact: Consistent with findings by Mihai Popescu et al., our results show that automation in log

analysis and threat response significantly reduces manual intervention and enhances overall system responsiveness.

5. ACKNOWLEDGEMENTS

I would like to express our sincere gratitude to all those who contributed to the successful completion of this research work. First and foremost, we extend our heartfelt thanks to Dr. M.G.R. Educational and Research Institute, Chennai, for providing me with the necessary infrastructure and academic environment to carry out this project.

I deeply thankful to Dr. D. SATHYA SRINIVAS, Assistant Professor, Centre of Excellence in Digital Forensics, for his invaluable guidance, continuous support, and insightful feedback throughout the research. Her expertise and mentorship were instrumental in shaping the direction and quality of this work.

I also extend our appreciation to our colleagues and peers who provided constructive suggestions and moral support throughout this journey. Special thanks to the faculty of the Department of Computer Science Engineering for their encouragement and academic assistance.

6. CONCLUSIONS

As the density of sophisticated and long-lived attacks in a given period grows, web application defense has become imperative. This project introduces "Obsidian Gate," a sophisticated defense solution that combines the real-time filtering power of ModSecurity and the strong log analysis and visualization capabilities of the ELK Stack (Elasticsearch, Logstash, and Kibana). The project demonstrated the value of implementing open-source solutions in providing enterprise-class security and insight at the cost-effective price of open-source solutions instead of costly commercial ones.[19]

Through deliberate calibration and thorough testing, the envisioned framework was successful in identifying and capturing typical attack vectors such as SQL injection, cross-site scripting (XSS), and local file inclusion. After processing by Logstash and being indexed in Elasticsearch, these logs facilitated dynamic dashboards to be crafted in Kibana, which provided real-time insights into attack patterns, source IP address, frequency distributions, and affected endpoints. Real-time visibility into security threats not only enhances the response speed from security analysts but also supports extended forensic analysis and compliance reporting.

Interestingly, the literature review and comparative analysis identified that while ModSecurity provides a good WAF solution, it reaches its full potential only when integrated into a larger security monitoring stack like ELK. The stack's modular design allows customization, automation, and scalability—qualities that are necessary for securing today's dynamic, contemporary web environments. Moreover, this approach paves the way for integrating machine learning algorithms, which can further streamline alerting mechanisms and eliminate false positives by identifying advanced behavioral patterns in traffic. The deployment of "Obsidian Gate" demonstrates how defense-in-depth is possible by open-source collaborative development. It emphasizes the importance of not just blocking malicious traffic but also understanding it through advanced log analysis. This two-pronged approach enhances an organization's ability for proactive and reactive security. Furthermore, the system's adaptability to containerized and cloud environments ensures its usability within both traditional and DevOps-centric deployment models.

In the future, the following enhancements could be incorporated: threat intelligence feed integration, automated remediation scripts, and user behavior analysis. Alert forwarding to SIEM systems like Wazuh or Splunk would also add to its monitoring capabilities. Adaptive rule tuning research using AI would also be a giant step towards developing autonomous, self-healing web defenses.

In summary, "Obsidian Gate" is more than just a mere run-of-the-mill security tool; it is an end-to-end strategic framework that demonstrates how tightly integrated open-source solutions can provide robust, dynamic, and extensible protection for web applications in today's increasingly hostile digital landscape. [18].

7. REFERENCES

[1] Curiel, A. (2021). ModSecurity Handbook: The Complete Guide to Securing Your Web Applications. Feisty Duck.

[2] Liao, H., Lin, C. H. R., Lin, Y. C., & Tung, K. Y. (2013). Intrusion detection system: A comprehensive review. Journal of Network and Computer Applications, 36(1), 16–24.

[3] Daryabar, F., Dehghantanha, A., Udzir, N. I., & Ibrahim, S. (2013). A survey about impacts of WAFs on web servers' performance and security. Australian Journal of Basic and Applied Sciences, 7(4), 184–190.

[4] Wang, R., Chen, S., Wang, X., & Qadeer, S. (2011). How to Shop for Free Online – Security Analysis of Cashier-as-a-Service Based Web Stores. IEEE Symposium on Security and Privacy, 465–480.

[5] Pahl, C., & Xiong, H. (2013). Migration of Web Applications to Cloud Environments: A Case Study. Software: Practice and Experience, 43(12), 1311–1327.

[6] Alsmadi, I., & Zarour, M. (2016). A Novel Approach for Web Application Log Analysis Using the ELK Stack. International Journal of Computer Applications, 150(3), 1–7.

[7] ernández-Medina, E., Trujillo, J., & Piattini, M. (2007). Web security within a UML-based web engineering approach. ACM Transactions on Internet Technology (TOIT), 6(2), 172–200.

[8] Garcia-Teodoro, P., Diaz-Verdejo, J., Macia-Fernandez, G., & Vazquez, E. (2009). Anomaly-based network intrusion detection: Techniques, systems and challenges. Computers & Security, 28(1–2), 18–28.

[9] Sarhan, Q. J., & Majeed, R. A. (2020). Enhancing Web Application Security Using Open Source WAF and ELK Stack. Journal of Engineering and Sustainable Development, 24(1), 87–95.

[10] Alqahtani, S., & Maarof, M. A. (2017). Log Analysis Approach for Detecting Malicious Web Activities. Journal of Theoretical and Applied Information Technology, 95(4), 808–815.

[11] Yegneswaran, V., Barford, P., & Ullrich, J. (2003). Internet Intrusions: Global Characteristics and Prevalence. Proceedings of the 2003 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems, 138–147.

[12] Ahmed, M., Mahmood, A. N., & Hu, J. (2016). A survey of network anomaly detection techniques. Journal of Network and Computer Applications, 60, 19–31.

[13] Rahalkar, S. (2018). Kali Linux - An Ethical Hacker's Cookbook. Packt Publishing.

[14] Kassim, Y. M., & Abdullah, M. T. (2021). ELK Stack-Based Monitoring for Web Application Attacks. International Journal of Computer Science and Information Security, 19(4), 123–131.

[15] Gojali, Angga Muhamad, et al. "ANALYSIS OF THE EFFECTIVENESS OF THE COMBINATION OF FAIL2BAN AND MODSECURITY IN MITIGATION OF DDOS ATTACKS ON WEB SERVERS." (2024) Novice Research Exploration 1.2

[16] T. D. Sobola, P. Zavarsky and S. Butakov, "Experimental Study of ModSecurity Web Application Firewalls," 2020 IEEE 6th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS), Baltimore, MD, USA, 2020, pp. 209-213,

[17] Rouse, M. (2015). What is a Web Application Firewall (WAF)? TechTarget.

[18] Apache Software Foundation. (2023). ModSecurity Reference Manual.

[19] Elastic. (2022). Elastic Stack Security Monitoring Guide.