Research Article

An Integrated Web Security Application: Integration of Nginx Reverse Proxy, Fail2ban, WAF, Postgresql and Laravel

Raif Sime, Necmettin Sezgin and Fikri Aggun

Abstract— **Recently, the increase in network-connected devices and the ability to run every application over the web has made web application security an issue that needs to be seriously considered. Although firewall solutions are used to protect networked systems and users, it seems that they are insufficient to ensure application security, especially in today's conditions. In this context, WAF (Web Application Firewall) systems have been developed and continue to be developed, especially to ensure the security of web applications. While the firewall filters traffic at the network layer, which is a lower layer, WAF protects at the application layer closest to the user. Network administrators intensively use WAF applications and the systems they create with new technologies integrated into these applications in order to maximize security.**

In this study, the WAF application, which is used together with Laravel, File2ban and Postgresql, is discussed, which we compiled and ran to protect the corporate network we manage from attacks and application vulnerabilities. In addition, it is thought that this study will guide other researchers working in this field and aims to open doors to produce more effective solutions.

Index Terms— **WAF, File2ban, Laravel, Postgresql.**

I. Introduction

THE INCREASING use of the web and the proliferation of web applications in every field makes it necessary to ensure application security and it is becoming increasingly difficult to ensure security. In addition, developers use risky tools and techniques to detect such attacks, threats and vulnerabilities present in the application [1]. In order to ensure security in corporate networks, problems arising from weak points should be solved as much as possible.

This is possible with security systems that complement each other and work interactively together. Such a structure is called multi-layered security and defense in depth [2].

In addition, even people who are authorized to use the systems may unknowingly damage the system as a result of unconscious use. For this reason, information systems must be protected against both internal threats and possible malicious threats from outside [3].

Firewalls have been used for general protection, port and protocol-based filtering, network layer security, etc. to ensure web security in the use of applications, but providing only network security is insufficient to prevent recent attacks. Web applications have recently become the primary target of attacks due to apathy, lack of awareness, lack of secure software development techniques and disregard for web application security. Although traditional firewalls successfully prevent network layer attacks, they are not effective in web-based attacks on web applications. Therefore, web applications need security in order to prevent information loss and vulnerability on the Internet, which is not a secure environment [4].

In this context, in addition to firewalls, which alone are insufficient to ensure the security of web applications, application layer protection, http-based filtering, web application security and detailed content review, applications called web application firewall (WAF) have started to be used extensively. WAF is likened to a security shield for applications accessed using HTTP [5]. In summary, a firewall is designed for general network security, while a WAF is designed to secure web applications. A firewall filters traffic at the network layer level, while a WAF filters traffic at the application layer level. When both security mechanisms are used together, it creates a holistic security strategy and protects your network and web applications more effectively. In addition to these features, they can bring additional protection in areas that were not taken into account when the software was developed [6]. The complexity of web applications, the increase in cyber-attacks, the difficulty of detecting unknown threats, the ability to perform security controls at the application level, the protection of sensitive data, and the adaptation of standards that must be followed for data security to web applications reveal the need for WAF applications. WAF can have two types of security models, positive or negative, depending on the type of policy. A positive security model only allows traffic that

Raif Sime, Master Student, Graduate School of Education, Batman, University 72100 Batman, Turkey,(e-mail: raifsime@yahoo.com).

https://orcid.org/ 0009-0008-4292-2456

Necmettin Sezgin, Department of Computer Engineering, Batman University, Batman 72100, Turkey, (e-mail: necmettin.sezgin@batman.edu.tr).

https://orcid.org/0000-0002-4893-6014

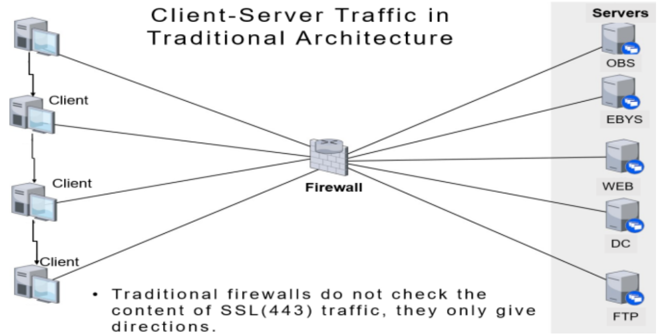
Fikri Ağgün, Department of Computer Technologies, Adilcevaz Vocational School , Bitlis Eren University, Bitlis 13100, Turkey, (e-mail: faggun@gmail.com).

https://orcid.org/0000-0001-9550-1462

Manuscript received Sep 11, 2024; accepted March 06, 2025.
DOI: 10.17694/bajece.1547456

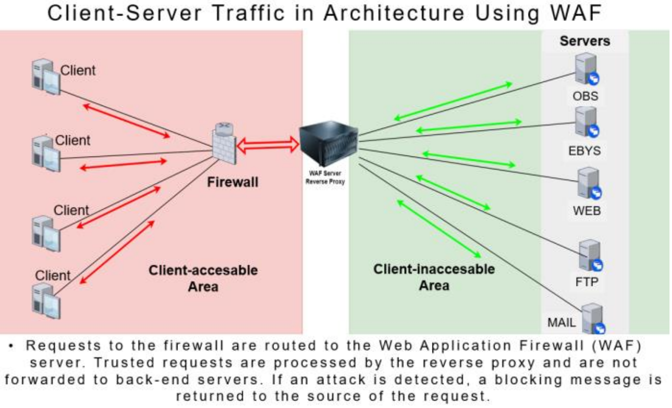
matches the policies to pass through. All other traffic is blocked. A negative security model allows all traffic through and only tries to block traffic represented by malicious rules. If the negative model alone is used in a WAF, hackers can bypass the inspection. Therefore, it is recommended to use a combination of both negative and positive model with better protection [7]. In teheir work, the authors emphasize the importance of enhancing web application security against SQL injection attacks and present their findings as a promising solution for improving the effectiveness of existing WAFs. The solution proposed in the article demonstrates robust detection capabilities, enhanced performance with higher generalization rates, and competitive advantages over existing solutions, indicating its potential to serve as a valuable tool in web application security [8]. In conclusion, WAF is a critical component to enhance the security of modern web applications and protect against cyber-attacks.

With the increasing security risks associated with web applications, the need for a WAF is also increasing. As shown in figure (Fig.1a), all SSL traffic passes directly through without content checking in networks with traditional firewalls, but in networks with WAF architecture (Fig.1b), all requests for user-to-server traffic are answered at the reverse proxy server, security checked, and if an attack is detected, it can be rejected before the request reaches the original servers.


Client-Server Traffic in Traditional Architecture

• Traditional firewalls do not check the content of SSL(443) traffic, they only give directions.

Fig.1a. Client-Server traffic in traditional architecture


Client-Server Traffic in Architecture Using WAF

• Requests to the firewall are routed to the Web Application Firewall (WAF) server. Trusted requests are processed by the reverse proxy and are not forwarded to back-end servers. If an attack is detected, a blocking message is returned to the source of the request.

Fig.1b. Client-Server traffic in WAF-enabled architecture

Web applications, which enable many transactions to be performed, become the target of attacks because they contain various information such as personal, banking and corporate information. Ensuring the security of these environments is directly related to the security of web applications. Web application security refers to all measures taken to ensure the confidentiality, integrity and accessibility of the data it contains [9]. In their article the authors aim to develop an affordable and user-friendly WAF framework. They employ proxy techniques to block and sanitize malicious user requests and implement two machine learning models for detecting malicious requests and classifying attack types. As a result, they propose a more accessible, effective, and user-friendly WAF solution that could significantly benefit organizations with limited resources. The proposed framework addresses common security vulnerabilities and provides real-time monitoring, assisting organizations in enhancing their security measures. Additionally, focusing on interface development through user feedback, as expressed for future work, is among the outcomes of our study [10]. The security of web applications is of increasing importance today. Therefore, developers and system administrators try to protect web applications by implementing various security measures. This paper will focus on a web application security system integrated with Nginx reverse proxy, its structure, components and benefits.

II. MATERIALS AND METHODS

In our application, a complex and effective structure has been created to increase web application security. In this system, an efficient and effective network performance is achieved by combining different technologies such as Nginx reverse proxy, WAF, PostgreSQL, Laravel and Fail2ban. Nginx and WAF are used for traffic routing and filtering, PostgreSQL for logging and data storage, File2ban for automatic IP blocking mechanism, and Laravel for security rules management and application environment.

A. Nginx Reverse Proxy and WAF Integration

Nginx is an open source web server software that can be used as an HTTP and reverse proxy server. Nginx is stable, secure and very easy to configure, and has performance and efficiency advantages over Apache [11]. A reverse proxy takes requests and redirects them to another server. That is, clients make requests to the Nginx server and Nginx receives the request, forwards it to the destination server and forwards the answers to the clients. This is useful in situations where the target servers are not directly accessible and provides several advantages. The client (for example a browser) sends a request to the Nginx server, like an HTTP request to a URL. Nginx examines the incoming request and routes it according to certain rules. These rules ensure that certain URLs or domains are directed to certain destination servers. In the next step, Nginx forwards the request to the destination server for processing. The response from the destination server is received by Nginx and forwarded to the client. This is in response to the client's original request.

In this way, we can list the benefits of using Nginx as a reverse proxy as follows:

1. *Load Balancing*
   Nginx balances the load by distributing traffic between multiple destination servers. This improves the performance and availability of the target servers.
2. *Provide Cache Support*
   Nginx reduces the load on target servers and speeds up response times by caching static content.
3. *Security*
   Nginx provides security by filtering incoming requests, validating them, and applying firewall rules if necessary.
4. *URL Routing*
   Provides a flexible structure by routing specific URLs or domains to different destinations. This is especially useful when multiple web applications or microservices are used.
5. *SSL Termination*
   Nginx manages SSL/TLS certificates by managing HTTPS traffic and can forward HTTPS traffic as HTTP to destination servers. This does not require destination servers to deal with SSL/TLS and simplifies the structure.

Using Nginx as a reverse proxy makes web servers and applications more secure, efficient and scalable in many cases. For this reason, many large-scale websites and applications prefer Nginx.

B. PostgreSQL Integration with Laravel and Criteria Based Transaction Execution

Laravel is a PHP-based open source web application framework that aims to create fast and efficient web applications, is based on MVC (Model-View-Controller) architecture and offers developers a set of useful tools and libraries. In the development of the project, Laravel's features and components such as Eloquent ORM, Blade Template Engine, Migrations and Seeding, Routing and Controls, and Artisan Command Line Tool were used extensively for efficiency.

It is important to record the operation of the system in question and the movements during the operation process, both in order to identify problems that may arise in the system and for legal obligations. In order to keep the event records of the system developed for this purpose, integration with a powerful database such as PostgreSQL was provided. PostgreSQL is an open source and relational database management system. PostgreSQL is known for its strong ACID compatibility, wide data type support and advanced features. In the application, PostgreSQL's ACID Compatibility, Wide Data Types, Advanced Extension Support, JSON Support and high performance features were utilized to maximize the recording capability.

C. Mod Security Integration

Web application firewalls identify, monitor and block HTTP traffic to and from a web server. By controlling HTTP traffic, application vulnerabilities such as SQL injection, cross-site scripting (XSS), file injection and protection failures can be prevented [12]. ModSecurity is an open source WAF method. Version 3 (ModSecurity3) is the latest and most actively available version of this application. ModSecurity3 includes a number of improvements, new features and updates compared to previous versions. ModSecurity3 has abandoned the single-threaded architecture in favor of a Multi-Threaded Architecture. This provides better performance and scalability, allowing the WAF to work more effectively, especially in high-traffic environments. Another feature, the New Low-Level Engine, uses a more flexible and efficient engine compared to previous versions. This allows to process requests faster and apply more complex security rules. ModSecurity3 offers better performance and efficiency compared to previous versions. The multi-threaded architecture and new low-level engine enable ModSecurity to consume fewer resources and run faster. ModSecurity3 includes new security rules, predefined rules, cyber threat signatures and updates. This enables the WAF to more effectively protect against current threats. It improves compatibility with popular web servers such as Apache HTTP Server and Nginx. It also provides better integration with various security products and services. ModSecurity3 improves debugging and diagnostics. This enables faster detection and resolution of errors and problems, which makes the WAF work more reliably. In addition, ModSecurity3 provides enhanced features, performance and updates to protect web applications against security threats. In today's world of growing security awareness and increasing complexity of web applications, security tools like ModSecurity play an important role.

D. Fail2ban Integration and Automatic Blocking

PostgreSQL and Fail2ban integration is an effective method to increase the security of a database system. This integration ensures database security by automatically responding to malicious activity and allows system administrators to quickly respond to security threats. In our application, File2Ban and PostgreSQL integration works as follows: PostgreSQL offers various logging levels. One of the logging levels, which is important for security, enables logging of certain types of queries, failed login attempts or other critical events. Fail2ban monitors specific log files and detects certain patterns in those files (for example, failed login attempts) and temporarily or permanently blocks IP addresses that match those patterns. Fail2ban manages blocked IP addresses dynamically. IP addresses blocked for a certain period of time can be automatically released or permanently blocked according to a specific set of rules.

Thanks to PostgreSQL and Fail2ban Integration, automatic security is provided by monitoring and blocking events recorded for web security in the running system. In addition, thanks to Fail2ban, malicious activities are quickly detected and intervention is provided. Automatic blocking helps prevent threats without requiring manual intervention and with less hassle for system administrators. By recording and archiving events, system administrators can analyze the security status and take faster precautions against new vulnerabilities that may arise.

When these effects are analyzed, the PostgreSQL-Fail2ban integration is an effective method to increase the security of a system. This integration ensures database security by automatically responding to malicious activity and provides system administrators with a rapid response to security threats.

III. Results and Discussion

When the studies and the operation of the implemented system were analyzed, it was observed that the integrated Nginx reverse proxy, WAF, PostgreSQL, Laravel and Fail2ban system increased the security of web applications. Thanks to the enhanced security provided by the system, a more efficient structure has been achieved by ensuring that each job is done by a different module instead of uploading all jobs to a system.

In our study, as a security measure, the Nginx reverse proxy filters incoming requests and blocks malicious traffic, while the WAF detects vulnerabilities by performing more in-depth examinations. Thus, it provided a more effective protection for the web application against cyber-attacks.

Thanks to the Laravel application, security policies can be defined and customized more flexibly. Security checks are performed according to the criteria determined by the event logs stored in PostgreSQL, so that each application can determine and implement specific security requirements. This is an ideal solution that facilitates the implementation of security policies and determines the security level specific to each application. Event logs obtained as a result of the application provide an important data source for monitoring, analyzing the security situation and responding when necessary. Thus, it has been observed that security measures can be taken more easily and earlier. It is also clear that analyzing this data will enable continuous improvement and strengthening of security policies.

Another benefit emphasized in the system is the caching mechanism. This mechanism and the accompanying Nginx reverse proxy improve the performance of the web application running on the system. Thanks to this system, the necessary protection measures are taken for the security of the web application, while performance loss is minimized and the user experience is prevented from being exposed to negative effects.

As a result of our study, the system established with our application works in the active corporate network and responds to many requests as seen in the graph and faces a large number of attacks.

As can be seen from the graph (Fig.2), 2% of the requests directed to the servers are attack requests. Although this rate may seem low, it means a serious attack attempt in a network with very heavy traffic and can pose significant security risks.

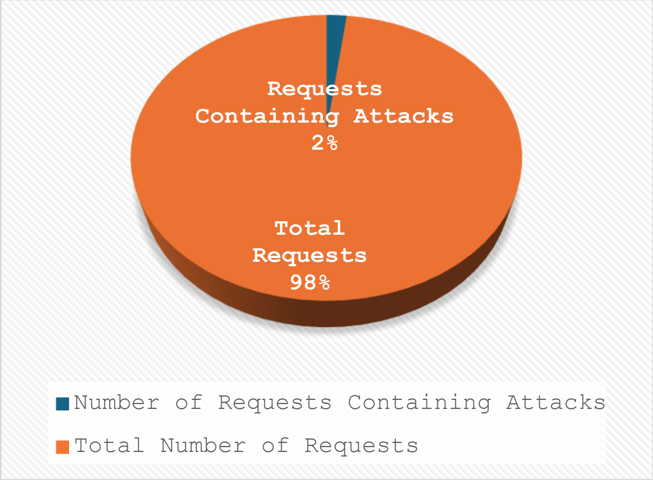


Fig.2. Total / Harmful Requests for corporate network traffic

For this reason, detecting and blocking these attack requests is of great importance in terms of ensuring system continuity.

The graph below shows the number of attacks on our organization's network for 12 days and the increases in these numbers on some days.

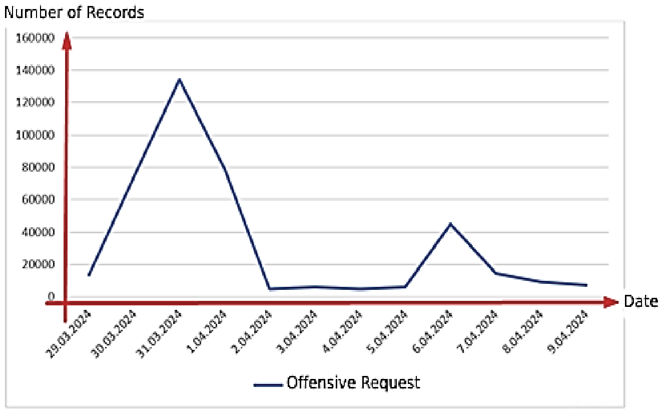
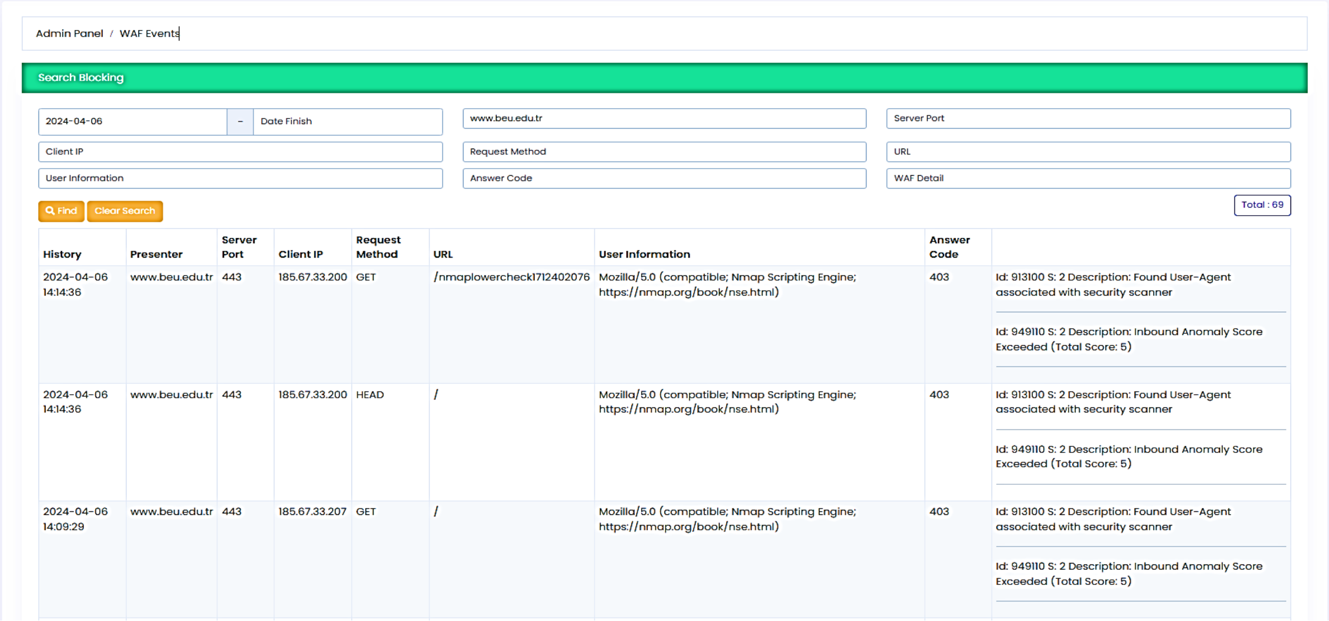


Fig.3. Number of traffic records containing attacks between 29.03.2024 - 09.04.2024

When the graph in Figure 3 is analyzed, it is seen that servers are attacked on some days and in this context, the number of attacks increases significantly. That is, the daily distribution of requests clearly shows that there are more attacks on certain days than others. This is an important consideration for the security of the servers and requires measures to be taken to detect and prevent attacks more effectively. The increases in the event logs generated by the application and analyzed with the help of PostgreSQL and Laravel show that attacks are being made at certain intervals and that there is a search for open doors to infiltrate the system. An example of attack record table that have been stored in PostgreSQL is showed in Figure 4.



Admin Panel / WAF Events

Search Blocking

2024-04-06 – Date Finish

www.beu.edu.tr

Server Port

Client IP

Request Method

URL

User Information

Answer Code

WAF Detail

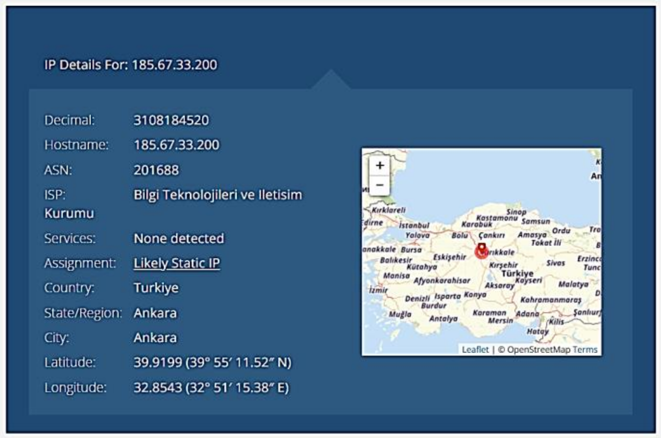
Total : 69

History	Presenter	Server Port	Client IP	Request Method	URL	User Information	Answer Code	
2024-04-06 14:14:36	www.beu.edu.tr	443	185.67.33.200	GET	/nmaplowercheck1712402076	Mozilla/5.0 (compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)	403	Id: 913100 S: 2 Description: Found User-Agent associated with security scanner Id: 949110 S: 2 Description: Inbound Anomaly Score Exceeded (Total Score: 5)
2024-04-06 14:14:36	www.beu.edu.tr	443	185.67.33.200	HEAD	/	Mozilla/5.0 (compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)	403	Id: 913100 S: 2 Description: Found User-Agent associated with security scanner Id: 949110 S: 2 Description: Inbound Anomaly Score Exceeded (Total Score: 5)
2024-04-06 14:09:29	www.beu.edu.tr	443	185.67.33.207	GET	/	Mozilla/5.0 (compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)	403	Id: 913100 S: 2 Description: Found User-Agent associated with security scanner Id: 949110 S: 2 Description: Inbound Anomaly Score Exceeded (Total Score: 5)

Fig.4. Attack records that stored in database

Some useful information can be obtained as a result of examining these attempts, which are perceived as attacks on systems, with reverse engineering methods and identifying the sources of attack.

When the data and IP addresses in the figure are checked, it is seen that the attempts that appear to be attacks are the penetration tests performed by the Information and Communication Technologies Authority on our organization's network (Fig.5). Since penetration tests are controlled attack and vulnerability scanning procedures performed to assess the security of an institution's or an organization's information systems, such tests help to take preventive steps to detect and correct vulnerabilities of information systems. Therefore, the identification that the IP addresses examined were used during the penetration tests shows that we have taken an important step to increase the effectiveness of security measures and strengthen the network's defenses.



IP Details For: 185.67.33.200

Decimal:	3108184520
Hostname:	185.67.33.200
ASN:	201688
ISP:	Bilgi Teknolojileri ve Iletisim Kurumu
Services:	None detected
Assignment:	Likely Static IP
Country:	Turkiye
State/Region:	Ankara
City:	Ankara
Latitude:	39.9199 (39° 55′ 11.52″ N)
Longitude:	32.8543 (32° 51′ 15.38″ E)

Fig.5. Location analysis of an attack

When we examine the communication requests and attacks in network traffic on the same graph(Fig.6); Requests directed to servers and attacks are independent of each other.

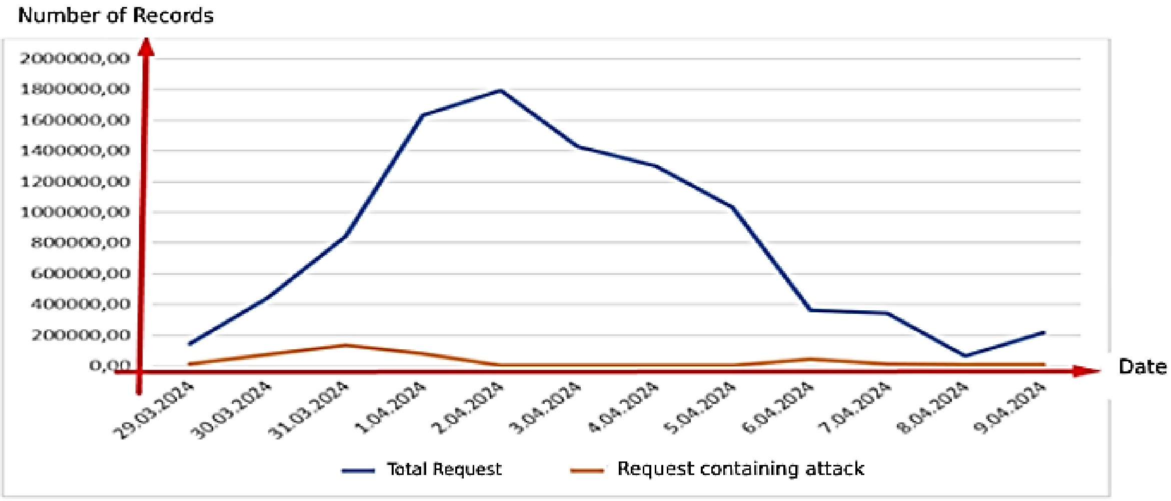


Fig.6. All requests and attacks

That is, there is no direct correlation between the number of requests directed to servers and the number of requests containing attacks. For example, on any day, the number of requests may increase, but this increase does not affect the number of requests containing attacks. This may indicate that attacks are carried out independently of the request traffic and that attackers are not focused on density at certain times. This detail is an important consideration in the design of security measures and intrusion detection mechanisms.

IV. Conclusion

Our web application security system, integrated with Nginx reverse proxy, offers an effective solution to increase the security of web applications. Considering the benefits, it provides, the security of web applications is increased while performance and efficiency are also ensured. This is a gain that increases both the security of users and the operational efficiency of the business. It is thought that our system will strengthen the hands of system developers and administrators in areas such as providing web security, cyber security measures, etc. and offers both fast and effective solutions. Since the system has an open source structure, it can be said that it is a system open to development. In addition, it is thought that the system is a work that can be a source of inspiration for system developers working in this field and experiencing security problems.

References

[1] D. Mairaj Inamdar and S. Gupta, "A Survey on Web Application Security," Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol., vol. 3307, pp. 223-228, 2020, doi: 10.32628/cseit206543.

[2] E. Karaarslan, T. Tuğlular, and H. Şengonca, "Enterprise web security structure," in Akademik Bilişim, 2008, pp. 1-9.

[3] M. Baykara, R. Daş, and G. Tuna, "Web-based log analysis platform for detection of web attacks from web server access logs," Firat University Engineering Sci. Derg., vol. 28, no. 2, pp. 291-302, 2016.

[4] A. Tekerek, C. Gemci, and O. F. Bay, "Development of a hybrid web application firewall to prevent web based attacks," in 8th IEEE International Conference on Application of Information and Communication Technologies, AICT 2014 - Conference Proceedings, 2014, pp. 1-4, doi: 10.1109/ICAICT.2014.7035910.

[5] R. A. Muzaki, O. C. Briliyant, M. A. Hasditama, and H. Ritchi, "Improving Security of Web-Based Application Using ModSecurity and Reverse Proxy in Web Application Firewall," 2020 Int. Work. Big Data Inf. Secur. IWBIS 2020, pp. 85-90, 2020, doi: 10.1109/IWBIS50925.2020.9255601.

[6] H. Tan and A. Z. Aktas, "An approach for an organization's information system security," in Network and Information Security Symposium, 2011, pp. 34-39.

[7] V. Clincy and H. Shahriar, "Web Application Firewall: Network Security Models and Configuration," in Proceedings - International Computer Software and Applications Conference, 2018, vol. 1, pp. 835-836, doi: 10.1109/COMPSAC.2018.00144.

[8] F. Omar, D. Ahmed, O. Elnakib, et al., “Towards a User-Friendly Web Application Firewall.,” In: Proceedings - 11th IEEE International Conference on Intelligent Computing and Information Systems, ICICIS 2023. pp. 483–488. IEEE (2023).

[9] D. Aydogdu and M. Gündüz, "A Research on Web Application Security Vulnerabilities and Security Solutions," Uluslararası Uluslararası Bilgi Güvenliği Mühendisliği Dergisi, vol. 2, no. 1, pp. 1-7, 2016, doi: 10.18640/ubgmd.56836.

[10] A. Coscia, V. Dentamaro, S. Galantucci, A. Maci, and G. Pirlo, “PROGESI: A PROxy Grammar to Enhance Web Application Firewall for SQL Injection Prevention.,” IEEE Access. vol. 12, no. August, pp. 107689–107703, 2024.

[11] Nginx: the High-Performance Web Server and Reverse Proxy, https://dl.acm.org/doi/fullHtml/10.5555/1412202.1412204.

[12] T. D. Sobola, P. Zavarsky, and S. Butakov, "Experimental Study of ModSecurity Web Application Firewalls," Proc. - 2020 IEEE 6th Intl Conf. Big Data Secur. Cloud, Big Data Security 2020, 2020 IEEE Intl Conf. High Perform. Smart Comput. HPSC 2020 2020 IEEE Intl Conf. Intell. Data Secur. IDS 2020, pp. 209-213, 2020, doi: 10.1109/BigDataSecurity-HPSC-IDS49724.2020.00045.

BIOGRAPHIES

Raif Sime was born in Ağrı in 1984. He graduated from Atatürk University Ağrı Vocational School, Department of Computer Technologies and Programming in 2006. He graduated from Anadolu University, Faculty of Business Administration and Ağrı İbrahim Çeçen University, Faculty of Education, Department of Religious Culture and Ethics Education in 2014. Since 2022, he has been working as the Branch Manager at Department of Information Technologies in Bitlis Eren University.

Necmettin Sezgin graduated from Hacettepe University, Faculty of Engineering, Department of Electrical and Electronics Engineering in 2001. He received MSc degree at Dicle University, Institute of Science, Department of Electrical and Electronics Engineering in 2003 and PhD at İnönü University, Institute of Science, Department of Electrical and Electronics Engineering in 2011. In 2011, he started working as an assistant professor at Batman University at Department of Electrical and Electronics Engineering. In 2014, he received the degree of associate professor in the field of Electrical and Electronics Engineering and started working as an associate professor at Batman University, Department of Electrical and Electronics Engineering in the same year. He was appointed as a Professor at Batman University at Department of Computer Engineering in 2019.

Fikri Ağgün was born in Ahlat, Bitlis in 1980. He received the B.S. in computer engineering from the University of Selçuk, Konya in 2001 and M.S. degree in in Biometry-Genetic from Yuzuncu Yil University, Van in 2011 and the Ph.D. degree in electrical and computer engineering from Ankara Yildirim Beyazit University, Ankara, in 2020. His research interests are wireless sensor networks, Vanets and Mac protocol design. He still works as an Assistant Professor at Bitlis Eren University.