RESEARCH ARTICLE OPEN ACCESS

DESIGN AND IMPLEMENTATION OF WEB APPLICATION USING FIREWALL MODEL

Asst. Prof Vishakha Akhare*, ROHAN RATHI**,PORNIMA KOLHE***, PIYUSH ADASE****

*Asst. Prof CSE Department (Guide), GH Raisoni University, Saikheda
Email: vishaka.akhare@ghru.edu.in)
**(B.Tech CSE Department , GH Raisoni University, Saikheda
Email: rohanrathi333@gmail.com)
***(B.Tech CSE Department , GH Raisoni University, Saikheda
Email: dhanshreekolhe03@gmail.com)
****(B.Tech CSE Department , GH Raisoni University, Saikheda
Email: Piyushadase19@gmail.com)

----------**********----------

Abstract:

One of the Intrusion Detection System (IDS) methods for preventing web servers from HTTP attacks is Web Application Firewall (WAF). WAF is a security solution that can detect and block a variety of threats, including XSS, Proxy, Bad Bot, and SQL-injection. As the number of Web apps grows, security becomes more exposed to a range of attacks. The majority of these attacks are aimed at the web application layer, and a network firewall alone will not be able to protect them.

The effectiveness of these attacks is mostly attributable to the ineptitude of application developers when it comes to building online apps and flaws in existing technologies. Web application attacks are the newest trend, with hackers attempting to target web applications using a number of methods.

In this paper, we propose a centralised For web application security, use a web firewall solution.that will provide a new type of synchronised system with for an online application, the capacity to identify and avoid a number of web application assaults large number a large number of hosts at the same time using a centralised command and control system. The information is subsequently sent to a server by the infected client. centralised command and control server, which will distribute the attack information to all of the integrated hosts.

Keywords —**Web application firewalls (WAF), web application solutions, SQL Injection, XSS, DDoS Attack, Suspicious User Behaviour.**

----------**********----------

I. INTRODUCTION

A Web Application Firewall (WAF) secures web applications by screening and monitoring HTTP traffic between them and the outside world rest of the Internet, as well as preventing bad HTTP traffic, malicious web service requests, and automated botnet attacks. It can prevent attacks that leverage known vulnerabilities in a web application, such as SQL injection, cross-Site Request Forgery (CSRF), cross-Site Scripting (XSS), DDoS assaults, cookie

poisoning, file inclusion, wrong system setup, and more, by examining traffic.

While proxies protect clients, WAFs protect servers from the majority of web application threats. A web application firewall (WAF) is used to safeguard a single web application or a group of web apps. A WAF may be thought of as a reverse proxy that protects servers from being exposed by requiring clients to pass through it before contacting the server.

II. LITERATURE REVIEW

Before we go into the specifics of our Identification of unauthorised user to prevent website, let's have a look at how it works. We take a look at some of the current systems. The following literature is research that has been done and has a relevance or connection to web application firewall and design for prevent the website.

Alexander Endraca et al. was formed by us (2013) The Web Application Firewall may compare the Access Control List, which is established by the administrator using any text editor, with the arriving HTTP packets from the traffic before it reaches the web server. Basic pattern matching with regular expressions is used to compare the payload of the packet. The findings show that the Web Application Firewall is accurate in detecting and rejecting various types of attacks based on OWASP's top 10 web application attacks.

Marek Zachara and Dariusz Palka (2014) The article goes over some of the challenges surrounding the installation and configuration of a Web Application Firewall, which secures the target application by evaluating incoming requests and their parameters.by comparing them to previously documented use patterns. However, there are certain issues about the categorization of data utilised in the learning process, which can, in some situations, impede the firewall's capacity to appropriately categorise traffic. On the basis of the authors' reference implementation, several issues are highlighted.

"In this assessment of the literature, we offer a centralised The web firewall system for web application security will provide a new type of synchronised system that will be able to detect and prevent a variety of web application attacks for a large number of hosts at the same time, using a centralised command and control system, the attacked client then sends the information to a centralised command and control system,

Aliero Muhammad Saidu (2020) One of the most serious threats against web database-driven systems is SQL injection attack (SQLIA). SQLIA is used by attackers to gain unauthorised access and modify data without permission. According to an analysis, these tools and procedures were created to avoid a subset of SQLIAs, and only a handful of them may be used to test various injection settings while investigating SQLIAs. It was also discovered that none of the tools or approaches could be used to defend against attacks that used the second order (server side SQLIA) SQLI vulnerability.

Finally, the paper underlines the significant difficulties that developers and researchers must address right away in order to avoid being attacked using SQLIAs. (IP)

III. METHODOLOGY

We proposed a method for cyber security for the website and how many visitors and hackers were trying to hack and the site blocked their ip address and notified us in our research paper where we explain about the details of different attacks where many different hackers are present in the outside world and become threats for the websites where they can hack their personal information's and other details so we proposed a method for cyber security for the website and how many visitors and hackers were trying to hack and the site blocked their ip address and notified us. Its techniques are divided into three basic stages, which are as follows:

Planning

1. Gather Information

Following some initialization, we get all of the data from the user for their security foundation, as well as from the client in the case of a firewall configuration review, and this data will comprise

- IP Address / URL for the firewalls in scope
- We want to see all the necessary information regardng configuration setting without the ability to modify means read only administrator - level credentials
- Any required acces information i.e. for VPN or MFA credentials for thr internal network
- Any best practce standards preferred

Execution

1. We will begin the device review by analysing the current configuration, looking for issues, and attempting to resolve them one at a time, as well as attempting to minimise the issues and vulnerabilities from the practises, as well as attempting to solve the realistic risk perspective and attempting to identify its issues by their rank and threats one at a time, and attempting to configure all of the settings that are categorised and important for the device.

- Authentication
- Authorization
- Alerting
- Firmware Patching
- Administrative Access
- Enable Security Add-Ons and Configurations

2. Firewall rule - set Review

The examination of the access control list (ACL) is the most important part of this sort of inspection. Our engineers will assess your rules against best practises, indicating potentially unsafe or risky rules, probable misconfigurations, and too permissive rules, among other things. Things like duplicate objects/rules, poorly documented rules, underused object/rules, and temporary rules may all be detrimental to your security posture over time.

Post-Execution

1. Reporting

Triaxiom will formally document the findings after finishing the major phase of the examination. An executive-level report and a technical findings report are usually included in the deliverables. The technical findings report, on the other hand, will include a list of all vulnerabilities, along with instructions on how to reproduce the problem, a risk summary, recommended remedial methods, and any useful reference links.

2. Quality Assurance

All evaluations are subjected to a thorough technical and editorial quality assurance process. Follow-ups with the customer to confirm or refute environmental information may also be necessary.

3. Presentation

The final task in any evaluation is to provide all documentation to the customer. Triaxiom will walk the customer through the content, make any required changes, and answer any questions regarding the assessment findings. Following this, we'll submit updated documentation and, if necessary, arrange for official retesting.

The WAF Engine is the heart of the Web Application Firewall. On the same machine, both the Web Application Firewall and the Web Server are installed. The Web Application Firewall only protects the Apache HTTP Web Server. The Packet Analyzer Module and the Configuration Module are the two components that make up the WAF Engine.

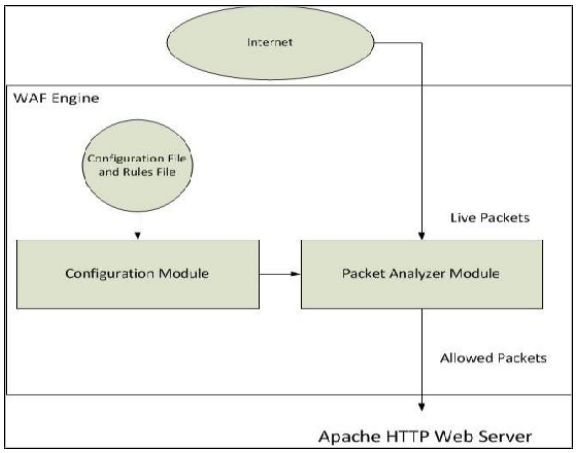

Fig 1. Packet Analyzer Module

```
iptables -t mangle -N PktEcho
iptables -t mangle -A PktEcho -j QUEUE
iptables -t mangle -A PREROUTING -p tcp --dport 8080
-j PktEcho PktEch
```

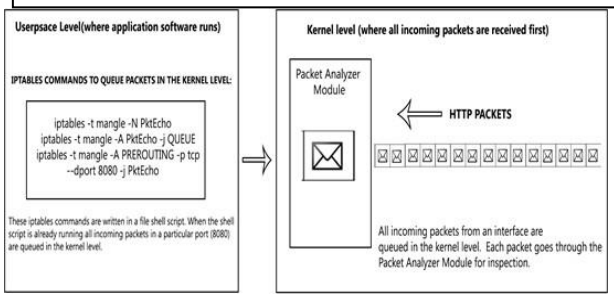

Fig 2. Packet analyzer.

Fig. 2 The diagram shows how packets are queued at the kernel level and where they are examined. Iptables is a userspace command for queuing packets at the kernel level and directing them to the Packet Analyzer first. The following commands are used to use ip tables:

Tables of IP commands A file shell script is used to write these commands. So that all inbound packets do not go immediately to the web server, this shell script should execute first before the other WAF modules. When the shell script runs, all incoming packets to a web server's specific port (for example, 8080) are queued at the kernel level. At the kernel level, each queued packet passes via the Packet Analyzer Module, which inspects the packet.

The Packet Analyzer sniffs packets that reach the Apache HTTP Web Server for data analysis. The data from the packets that were evaluated is utilised to determine whether the packets should be allowed or rejected. The information about the rejected packets is logged so that it may be analysed later. The Web Server's response is not monitored since it is presumed that the protected Web Server is secure.

Configuration Module

The Web Application Firewall's settings and the Web Application Firewall's Access Control List are applied via the Configuration Module. Text files are used to customise the settings and Access Control List. Both the settings and the Access Control List of the Web Application Firewall have a defined syntax. Both configuration files are checked by the Configuration Module to see if there are any input or syntax issues. The Configuration Module applies the setting and the Access Control List in the Web Application Firewall if no mistakes are identified in the configuration files.

Fig 3. Iptables commands

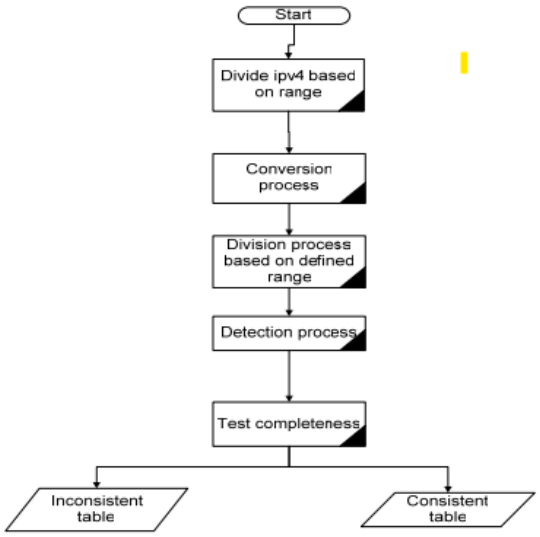

Fig 4. Flowchart

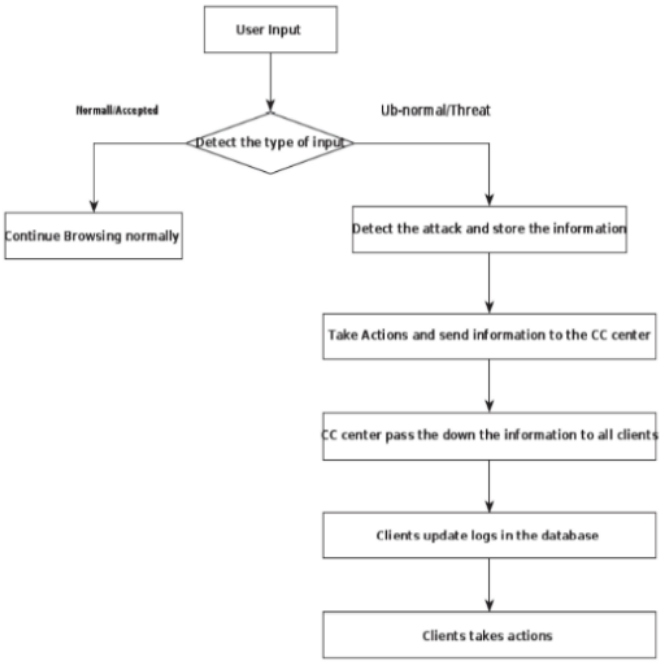

Fig 5. Centralized web firewall system

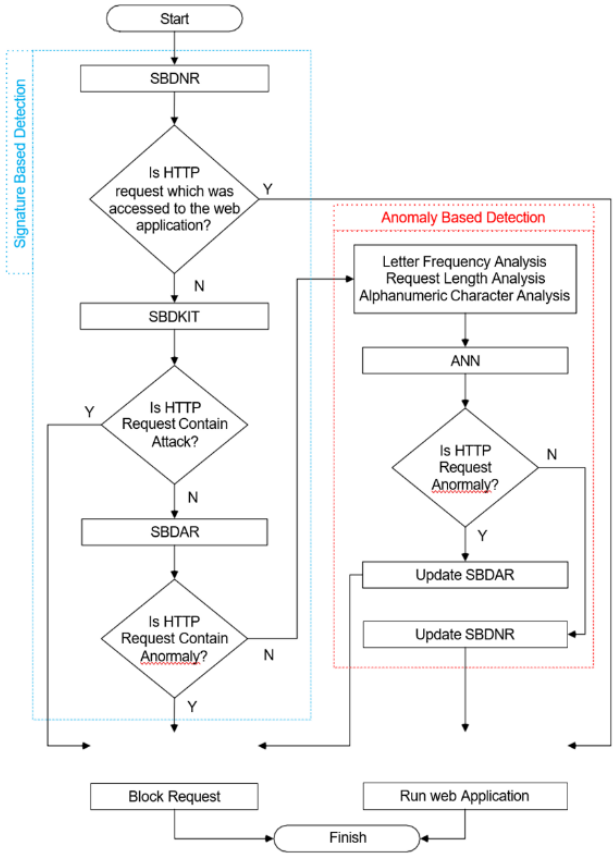

Fig 6. Proposed model flow diagram

IV. Result

The study's conclusion is a framework in which a user can log in from anywhere and prevent their website from being hacked, as well as see all of their website's activity in real time. The user may quickly ban or block the nation or IP addresses that they choose. The user can create his or her own massege to display the attacker or block IP addresses. The user may protect himself from sql injection, bad bot attacks, and proxy attacks with this security framework.

Fig 7. Dashboard

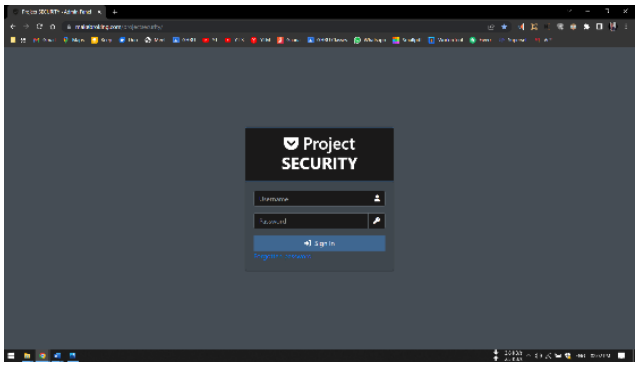

Fig 8. Admin login page

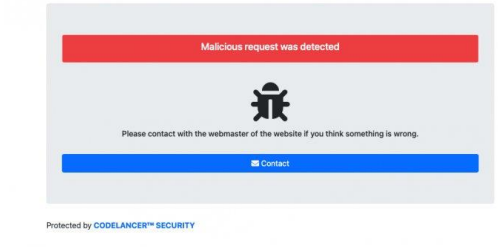

Fig 9. Request detection

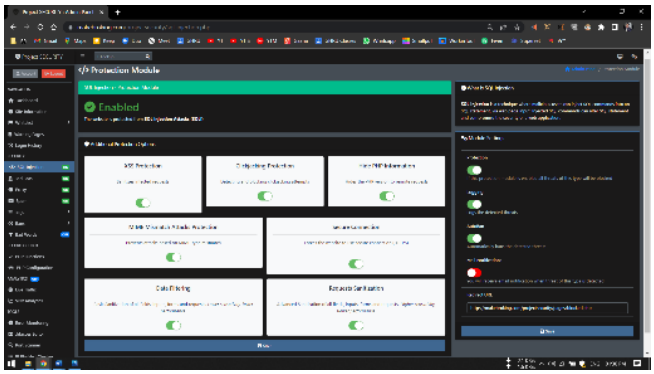

Fig 10. Protection Module

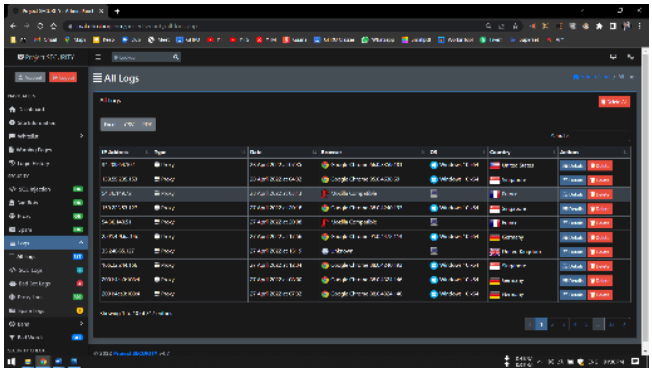

Fig 11. Log page

V. CONCLUSION

This research examined numerous methodologies and tactics for detecting and preventing SQL injection attacks as a web firewall, as well as the backdrop of web application threat and firewall.

The proposed system for web firewall centralization improves the process of detecting and preventing web application-based attacks by allowing standard standalone web firewalls to work as one fully integrated system, simply by updating and distributing the attack log to all firewalls connected to the system. Although each Web Firewall has its own log attack and may operate independently, this will improve the functionality and lower the risk of an attack across the entire integrated system. And it improved the chances of decreasing and averting a variety of assaults.

References

[1] Critical Analysis on Web Application Firewall Solutions, Abdul Razzaq, Ali Hur, Sidra Shahbaz, Muddassar Masood, H Farooq Ahmad

[2] Detecting inconsistent firewall configuration rules using range algorithm, Ahmed Farouk , Hamdy N.Agiza , Elsayed Radwan

[3] Centralized Web Application Firewall Security System Saher Manaseer1 & Ahmad K. Al Hwaitat

[4]Jim Beechey, "Web Application Firewalls:Defense in Depth for Your Web Infrastructure" March 2009

[5] A Mayer, A Wool, E Ziskind, 2000. "Fang: A Firewall Analysis Engine." IEEE SYMPOSIUM ON SECURITY AND PRIVACY

[6] WEB APPLICATION FIREWALL: REVIEW Muhammad Saidu Aliero1 , Bilyaminu Isah Shamaki2 , Ibrahim abubakar3 , Bello shamsudden kalgo4 , Abdul-azeez Muhammad Bello

[7] Aliero MS, Ardo AA, Ghani I, Atiku M. "Classification of Sql Injection Detection And Prevention Measure". IOSR Journal of Engineering (IOSRJEN), ISSN (e). 2016:2250- 3021