Faculty of Electrical and Electronic Engineering Technology

WEB APPLICATION FIREWALL USING PHP AND MYSQL

MUHAMMAD AIZAT BIN KHAMIS

Bachelor of Computer Engineering Technology (Computer Systems) with Honours

2023

WEB APPLICATION FIREWALL USING PHP AND MYSQL

MUHAMMAD AIZAT BIN KHAMIS

A project report submitted
in partial fulfillment of the requirements for the degree of
Bachelor of Computer Engineering Technology (Computer Systems) with Honours

Faculty of Electrical and Electronic Engineering Technology

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2023

UNIVERSITI TEKNIKAL MALAYSIA MELAKA
FAKULTI TEKNOLOGI KEJUTERAAN ELEKTRIK DAN ELEKTRONIK

BORANG PENGESAHAN STATUS LAPORAN
PROJEK SARJANA MUDA II

Tajuk Projek :

Sesi Pengajian :

Saya Muhammad Aizat bin Khamis mengaku membenarkan laporan Projek Sarjana Muda ini disimpan di Perpustakaan dengan syarat-syarat kegunaan seperti berikut:

1. Laporan adalah hakmilik Universiti Teknikal Malaysia Melaka.
2. Perpustakaan dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan laporan ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. Sila tandakan (✓):

☐ **SULIT*** (Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)

☐ (Mengandungi maklumat terhad yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)

☐ **TIDAK TERHAD**

Disahkan oleh:

(TANDATANGAN PENULIS)
Alamat Tetap: JA 6251, Jalan Seroja 4/2, Taman Maju, Jasin, Melaka

(COP DAN TANDATANGAN PENYELIA)
AHMAD FAIRUZ BIN MUHAMMAD AMIN
Pensyarah
Jabatan Teknologi Kejuruteraan Elektronik & Komputer
Fakulti Teknologi Kejuruteraan Elektrik & Elektronik
Universiti Teknikal Malaysia Melaka
76100 Durian Tunggal
Melaka

Tarikh: 8/2/2023

Tarikh: 15 Feb 2023

*CATATAN: Jika laporan ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali tempoh laporan ini perlu dikelaskan sebagai SULIT atau TERHAD.

DECLARATION

This project report, titled "Development of Web Application Firewall using PHP and MySql," is the result of my own research, with the exception of what is indicated in the sources. I hereby declare this to be the case. The report on the project was not approved for any degree, and it is not being considered for any other degree either.

Signature : 

Student Name : Muhammad Aizat bin Khamis

Date : 15/01/2023

APPROVAL

I officially declare that I have read over this project report, and after doing so, I have come to the conclusion that it meets the requirements, both in terms of its scope and its quality, to be awarded the degree of Bachelor of Computer Engineering Technology (Computer Systems) with Honours.

Signature : 

Supervisor Name : Ts. Ahmad Fairuz bin Muhammad Amin

Date : 15/01/2023

Signature :

Co-Supervisor :

Name (if any)

Date :

DEDICATION

I'd want to dedicate my research project to my parents, who have assisted me in completing it. Without them, it will be incredibly impossible to complete your courses and final year project. Not to mention my boss, who accepted and allowed me to finish this project. Also, thanks to my pal for his support and assistance in coming up with project ideas.

ABSTRACT

Web Application Firewalls, also known as WAFs, are deployed to protect websites and web applications and, when properly configured, offer a comprehensive level of security. When they are used excessively, these tools eventually cause problems. The deployment of a WAF may give users a mistaken impression of their level of security. In this article, we provide an overview of the traffic filtering models currently available as well as some recommendations for making the most of the advantages offered by a web application firewall. A WAF protects your online applications by classifying, analysing, and rejecting any dangerous HTTP/S traffic that reaches them. It also keeps any unauthorised data from exiting the app. It accomplishes this by adhering to a set of rules that assist it in determining which types of traffic are harmful and which are safe. A WAF, like a proxy server, acts as an intermediary to protect a client's identity while protecting the web app server from a potentially malicious client. This is known as a reverse proxy. WAFs can be software, hardware, or services delivered as a service. Policies can be modified to meet the unique requirements of your web app or group of web apps. Even though many WAF policies must be updated on a regular basis to address new vulnerabilities, machine learning has enabled some WAFs to update themselves. This automation is becoming increasingly important as the threat landscape becomes more complicated and unclear.

ABSTRAK

Tembok Api Aplikasi Web, juga dikenali sebagai WAF, digunakan untuk melindungi tapak web dan aplikasi web dan, apabila dikonfigurasikan dengan betul, menawarkan tahap keselamatan yang komprehensif. Apabila ia digunakan secara berlebihan, alat ini akhirnya menyebabkan masalah. Penggunaan WAF mungkin memberi pengguna gambaran yang salah tentang tahap keselamatan mereka. Dalam artikel ini, kami menyediakan gambaran keseluruhan model penapisan trafik yang tersedia pada masa ini serta beberapa cadangan untuk memanfaatkan sepenuhnya kelebihan yang ditawarkan oleh tembok api aplikasi web. WAF melindungi aplikasi web anda dengan menapis, memantau dan menyekat sebarang trafik HTTP/S berniat jahat yang tiba di aplikasi web. Ia juga menghalang sebarang data yang tidak diluluskan daripada meninggalkan apl. Ia mencapai ini dengan mematuhi satu set peraturan yang membantunya dalam menentukan jenis lalu lintas yang berbahaya dan yang selamat. WAF, seperti pelayan proksi, bertindak sebagai perantara untuk melindungi identiti pelanggan sambil melindungi pelayan aplikasi web daripada klien yang berpotensi berniat jahat. Ini dikenali sebagai proksi terbalik. WAF boleh berupa perisian, perkakasan atau perkhidmatan yang dihantar sebagai perkhidmatan. Dasar boleh diubah suai untuk memenuhi keperluan unik apl web atau kumpulan apl web anda. Walaupun banyak dasar WAF mesti dikemas kini secara tetap untuk menangani kelemahan baharu, pembelajaran mesin telah membolehkan beberapa WAF mengemas kini diri mereka sendiri. Automasi ini menjadi semakin penting apabila landskap ancaman menjadi lebih rumit dan tidak jelas.

ACKNOWLEDGEMENTS

I would like to begin by expressing my gratitude to my supervisor, Ts. Ahmad Fairuz Bin Muhammad Amin, for the invaluable direction, wise advice, and patience he provided during the entirety of the process of putting this project into action.

In addition, I would like to express my appreciation to Universiti Teknikal Malaysia Melaka (UTeM), as well as to my parents, for the financial support they provided during the completion of the project.

My gratitude extends to my parents, my in-laws, and other members of my family for their support, love, and prayers while I was pursuing my education.

In conclusion, I would like to express my gratitude to each and every one of my fellow students and colleagues, as well as the members of the teaching staff and any other individuals who aren't mentioned here for their aid and cooperation.

b 2023

TABLE OF CONTENTS

PAGE

CHAPTER 1

INTRODUCTION

1.1 Background

Over a million business and personal websites have been created in recent years. Many had to defend their websites from hackers. Many use a Content Delivery Network as a website firewall, but it's expensive. This plugin provides website security with PHP, MySQL, and other features. This prototype plugin has password and sensitive data encryption security features. This plugin may protect self-coded website systems from attackers exploiting site flaws. This plugin will provide CSRF token, (brute force, SQL Injection, XSS Injection, LFI, XXE Injection) protection, malicious HTTP headers from user-agent, web shell detection, email validation from user input, managing file download to avoid Local File Disclosure, and other security. This plugin may reduce cyber attacks that archive sensitive data, deface websites, or take control of them. The plugin will also check the SSL expiry date and recommend a Content Delivery Network (CDN) if one is not available on the site. Website protection eliminates cyberthreats. Due to its client-side defence, this plugin may struggle to control the internal webserver. DDOS protection requires a CDN. This plugin emails the administrator about potential attacks. This framework may help website coders avoid hackers and protect their sites.

1.2 Problem Statement

The world has gotten more advanced due to the expansion of technological knowledge and the introduction of new technologies made by humans. The technology is mostly based on chip microarchitecture, and depending on the architecture's design, there must be a code inside. The code is then converted from a low-level language to a higher-level language using another form of code program. Humans, like the technology on which they are based, are not perfect. Because nothing is perfect, technology becomes vulnerable as a result of vulnerabilities in some of the programmers that have been created. Because no system is completely secure, some technology becomes a target for hackers seeking information that could be harmful to all users. However, there must be a solution to prevent it by developing security software that can detect harmful conduct in any user. This project focuses on web application technologies and is intended to identify malicious scripts within web servers.

Because of the detection from the program that has been deployed inside the systems, a cyber-attack becomes less likely as technology for cyber security improves. Although today's sophisticated security appears to be successful in preventing most hackers from obtaining information, this does not mean that they all stop there. This is because they also studied how the security program works and how it flows to detect any harmful activity. As a result, they employ the security program to get around their nefarious operations and get what they want. As a result, this project will be coded to match the malicious script in order to identify hacking. Almost every website on the planet uses PHP as its backbone, accounting for 77.6% of all websites with a server-side programming language that we are aware of. As the major backend language, PHP will be used to build this project.

1.3 Project Objectives

The objective of this project:

- To detect Malicious script and malicious user input using the regex method.
- To log attacker's IP, user-agent and location.
- To block any attempts of attack and display them on a block page.
- To display a banned page for IP address and country location code.

1.4 Scope of Project

The scope of this project is as follows:

- Using the regex method for matching the unusual input from users using PHP regex.
- Display a block page with firewall detecting notice to the attacker and logging their information like IP address, user agents, requests and location.
- Notification of recent attacks through email.
- Sanitize every input of the user to make sure that although there is a firewall bypassing, they still cannot do the attack.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

A Web Application Firewall, often known as a WAF, is used to safeguard online applications by filtering and monitoring the HTTP traffic sent between a web application and the rest of the internet. It protects online applications from usual vulnerabilities such as cross-site scripting (XSS), file inclusion, and SQL injection. A WAF (in the OSI model) is a protocol layer 7 protection that is not designed to fight against all forms of assaults. This sort of attack mitigation is often part of a bigger group of technologies that work together to provide comprehensive protection against a wide range of attack vectors.

When a WAF is deployed in front of a web application, it acts as a barrier between it and the Internet. A web application firewall (WAF) is similar to a reverse proxy in that it prevents server hacking by requiring clients to pass through it before contacting the server. WAF works similarly to a reverse proxy in that it protects the server from potential threats by requiring clients to connect through it first. When a client connects to a proxy server, the client's identity is hidden since the proxy server acts as an intermediary between the client and the server. A web application firewall (WAF) is a type of reverse proxy that protects the server by routing client connections through the WAF first.

A WAF is governed by policies, which are a set of restrictions. These policies strive to defend against application vulnerabilities by limiting potentially dangerous messages. It is possible to implement policy changes quickly and easily, allowing for rapid responses to different attack vectors. It's possible, for example, to swiftly limit bandwidth during a DDoS attack by changing WAF regulations. This is made possible by the speed and ease with which policy updates can be made.

2.2 Related Works

The purpose of Web Application Firewall are to detect and match any malicious input from users while accessing web servers.

2.2.1 PHP Languages

In The PHP programming language (hypertext preprocessor) is a widely used open-source general-purpose programming language that can be embedded in HTML. Instead of a long list of commands to generate HTML, PHP websites have embedded code that does "something" (in this case, print "Hi, I'm a Muhammad Aizat!") (like in C or Perl). You can enter and exit "PHP mode" by using special start and stop processing instructions in the PHP code. In contrast to client-side JavaScript, PHP code is executed on the server, which then generates HTML for the client to view. The client would receive the results of the script's execution, but the client would be uninformed of the underlying code. You might even set up your web server to process all of your HTML files using PHP, making it practically impossible for users to figure out what you're up to.

The best thing about PHP is that it is simple to learn for a novice, but it also contains a lot of advanced features for experienced programmers. Don't be afraid to look over PHP's huge feature set. In just a few hours, you'll be able to jump right in and start building simple scripts.

Although PHP was created with server-side scripting in mind, it has a lot more capabilities. Continue reading to find out more about PHP's capabilities, or jump to the beginning course if you're only interested in web programming.point.

2.2.2 MySQL

MySQL is a relational database management system (RDBMS) based on structured query language developed by Oracle (SQL).

A database is a structured collection of information. It could be anything from a simple grocery list to a photo gallery or a huge business network's data repository. A relational database is a sort of digital storage that uses the relational paradigm to collect and organize data. In this architecture's tables, which are made up of rows and columns, the relationships between data objects are all logically organized. RDBMS stands for a relational database management system. It is a collection of software tools for creating, managing, and querying databases.

Popular software stacks employ MySQL to build and run a wide range of applications, including customer-facing web apps and complicated B2B services that rely heavily on data. Databases powered by MySQL are found on many of the most important websites on the internet. This is owing to the open- source nature of the software as well as its stability, comprehensive feature set, and Oracle's ongoing development and support.

2.2.3 Top ten Vulnerabilities According to OWASP

a) Broken Access Control. 94 percent of applications were found to have some form of failed access restriction, a significant improvement over the previous ranking of fifth. Broken Access Control is the category having the most instances in applications, with 34 Common Weakness Enumerations (CWEs) associated with it.

b) Cryptographic Failures, Sensitive Data Exposure, which was a symptom rather than a primary cause, dropped to second place. The emphasis in this article is on encryption failures, which commonly result in sensitive data leakage or system compromise.

c) Injection falls to third place. The 33 CWEs mapped into this category had the second-highest number of occurrences in applications, accounting for 94 percent of the apps analyzed for injection. This category now includes cross-site scripting in this edition.

d) Insecure Design is a new 2021 category that focuses on dangers induced by design faults. More threat modeling, safe design patterns and principles, and reference architectures will be required if we truly want to "move left" as an industry.

e) Safety Misconfiguration has increased from sixth place in the previous edition; 90 percent of applications were examined for misconfiguration in some way. It's not unexpected to see this sector rise as more individuals turn to highly adaptable software. This category has been expanded to include the old XML External Entities (XXE) category.

f) Using Vulnerable and Outdated Components was initially known as Vulnerable and Outdated Components, and it is now ranked #2 in the Top 10 community survey, but it also has enough data to make the Top 10 through data analysis. This sector has increased from ninth place in 2017 and is a well-known concern that we find challenging to identify and estimate risk. Because no Common Vulnerabilities and Exposures (CVEs) have been mapped to CWEs in this category, their rankings are based on a default exploit with a weight of 5.0.

g) Identification and Authentication Broken Authentication difficulties have slid to second place and now constitute CWEs that are more directly tied to identity failures. Despite remaining in the Top 10, the rising availability of common frameworks appears to be helping.

h) Software and Data Integrity Failures is a new category for 2021 that focuses on assuming the integrity of software updates, vital data, and CI/CD processes without checking them. The weighted impact of the Common Vulnerability and Exposures/Common Vulnerability Scoring System (CVE/CVSS) data on the ten CWEs in this category was one of the greatest. This wider category now covers Insecure Deserialization, which was established in 2017.

i) Logging and Monitoring of Security Before being included in the industry research (#3), Failures stood for Insufficient Logging & Monitoring, jumping up from #10. This category

has been expanded to encompass a broader range of failures, is difficult to test for, and is underrepresented in CVE/CVSS statistics. On the other side, failures in this area can have a direct impact on visibility, incident alerting, and forensics.

j) Server Side Request When a web application retrieves a remote resource without first validating the URL that was supplied by the user, this is an example of forgery. An adversary can still coerce a programme to send a forged request to an unexpected location, even if the programme is protected by a firewall, a Virtual Private Network (VPN), or some other type of network Access Control List (ACL).

2.2.4 Half of total websites in the world are vulnerable!

Companies all over the world are still struggling to defend themselves against an increasing number of web application and application-specific attacks. According to a report that was recently made public by NTT Application Security, by the year 2021, approximately half of all websites will have at least one critical flaw that can be exploited.

This report is the result of an in-depth analysis of data generated by more than 15 million application security scans performed by organizations in the year 2021. This will probably be remembered as one of the most important years for cybersecurity in the long run. The goal of the report is to give security and development teams who are in charge of securing the web applications that run their companies steps they can take right away.

The events that took place over the course of the past year, including the attack on the Colonial Pipeline, President Biden's Executive Order on "improving the nation's cybersecurity," and the ongoing fallout from the Log4j breach, have brought application security to the forefront of all discussions. There is evidence that the increased effort to fix critical vulnerabilities in applications used in both the public and private sectors has resulted in an unintended negative outcome. This is due to the fact that "fire-drill" remediation initiatives appear to occur as a tradeoff with existing

remediation efforts rather than as an addition to the latter. The market has reached a tipping point in how we approach application security testing as a result of the rapid adoption of modern practices that enable developers to rapidly build and deliver valuable functionality. In addition, the explosive growth in web applications that was accelerated by the COVID-19 pandemic has also contributed to this market shift.

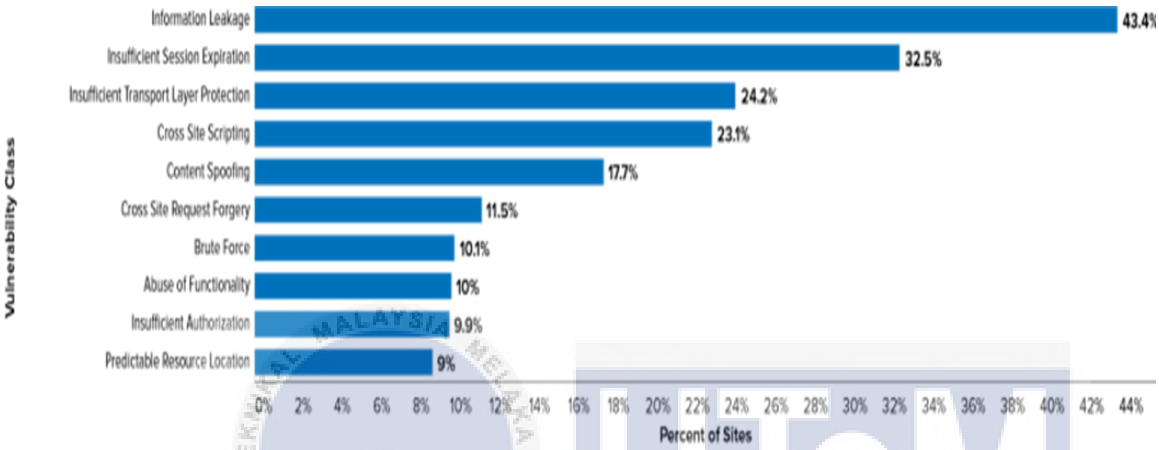

Figure 2.0. Percent of site with its common vulnerabilities. [25]

The industry of professional, scientific, and technical services had the highest percentage of permanently exposed sites in 2021, with 65%, while the industry of finance and insurance had the lowest percentage, with 43% of their sites permanently exposed.

Since the beginning of the year 2021, there has been a 1.7-day decrease in the typical amount of time required to patch a critical vulnerability (193.1 vs 194.8). Despite the fact that this data point indicates a positive trend, the reduction in time-to-fix is insignificant when compared to the reported increase in time-to-fix for all other risk categories throughout the year. Education had the longest time to fix a critical vulnerability (523,5 days), which is nearly 335 days longer than Public Administration, which maintained the shortest timeframe throughout the year. Education also had the highest number of critical vulnerabilities (188,6 days).

NTT Application Security discovered that the vulnerability classes that were most likely to be detected throughout the course of the year remained relatively consistent and that applications were plagued by vulnerability classes that are well-known in the industry. It is obvious that attackers

benefited in 2021 from a target-rich environment, given the low amount of effort and skill required to discover and exploit these vulnerabilities.

2.2.5 Eight(8) most important security headers!

Web applications employ security headers to configure security defences in web browsers. Browsers can make it more difficult to attack client-side vulnerabilities such as Cross-Site Scripting or Clickjacking based on these directives. Headers can also be used to set the browser to only allow valid TLS communication and to enforce the use of valid certificates, or even to mandate the usage of a specific server certificate.

I. X-Frame Options.

For the first time, X-Frame Options, developed by Microsoft to prevent script injection and cross-site scripting attacks, is enabled by default in Microsoft Internet Explorer. Protecting your website's iFrames is as simple as adding this HTTP security header and telling browsers how to handle iFrames. It protects primarily against all clickjacking attacks, where an attacker adds extra layers to a link or button in order to redirect users to a different page and steal their important data.

Htaccess code syntax to be applied as follows:

```
X-Frame-Options: DENY
X- Frame-Options: SAMEORIGIN
X-Frame-Options: ALLOW-FROM <em>URL</em>
```

Code explanation:

DENY: With this directive, iFrames will not be displayed.

SAMEORIGIN: Only iFrames will be rendered if this directive is in place.

ALLOW- FROM: With this directive, you can restrict iFrame rendering to a certain URL.

II. Strict-Transport-Security.

Strict-Transport-Security, or HSTS, is a secure online protocol. When enabled, the Strict Transport Security header protects against MIM attacks and cookie hijacking. This directive forces the browser to communicate via HTTPS rather than HTTP. Let's look at how an HTTP website that has been converted to HTTPS works. Your previous visitors will continue to try to access the old URL through HTTP. Because your site is already HTTPS, the old URL will automatically redirect to the new one.

However, before to being redirected to the new encrypted URL, your visitors will continue to have access to the non-encrypted version of your website. Hackers can carry out MIM, or Man in the Middle, attacks in the middle of an operation. When Strict-Transport-Security is set, the browser is encouraged to forgo loading HTTP pages in favour of HTTPS communication.

Htaccess code syntax to be applied as follows:

```
Strict-Transport-Security: max-age=”expire-time”
Strict-Transport-Security: max-age=”expire-time”;
includeSubDomains
Strict-Transport-Security: max-age=”expire-time”; preload
```


Code explanation:

max-age=<expire-time>: You can limit a browser's HTTPS access for a given period of time with this directive (in seconds).

max-age=<expire-time>;includeSubDomains: To indicate that the previous rule applies to all subdomains of the website, this directive must be used.

max-age=<expire-time>;preload: If you see this directive, it means your site is now included in the worldwide index of secure HTTPS resources.

III. Content Security Policy.

With the help of this HTTP security header, the browser will only be allowed to load the content that conforms to the policy. This means you will have control over your website's resources and only allow whitelisted resources to be loaded by browsers.

It directs the browser to the correct location while loading resources like images, scripts, and CSS. If you correctly implement this HTTP security header, your website will be protected from Clickjacking, Cross-Site Scripting (XSS), and malicious code injection.

To be sure, it's not bulletproof, but it can help prevent or at least lessen the severity of any harm that could otherwise occur. Most browsers have acknowledged and begun to support this key issue as well.

Htaccess code syntax to be applied as follows:

```
Content-Security-Policy:<policy-directive>;<policy-directive>
```


Code explanation:

<policy-directive>: It is permissible to load policy directives such as script-src (CSS), img-src (Images), and style-src (Stylesheet).

IV. X-Content-Type Options.

You can minimise or eliminate MIME type sniffing by informing the browser that MIME types are explicitly declared on the server. By monitoring the contents of incoming messages via MIME sniffing, hackers might potentially introduce any kind of malicious code that can be executed.

Any malicious resource that altered the response of a benign resource, such as an image, could have been injected by an attacker. MIME sniffing prevents the browser