

STRENGTHENING WEB APPLICATION SECURITY:
USING OPEN-SOURCE WAF EVALUATION
FRAMEWORK

ZHOU HAO

UNIVERSITI KEBANGSAAN MALAYSIA

STRENGTHENING WEB APPLICATION SECURITY: USING OPEN-SOURCE
WAF EVALUATION FRAMEWORK

ZHOU HAO

PROJECT SUBMITTED IN PARTIAL FULFILMENT FOR THE DEGREE OF
MASTER OF CYBER SECURITY

FACULTY OF INFORMATION SCIENCE AND TECHNOLOGY
UNIVERSITI KEBANGSAAN MALAYSIA
BANGI

2025

**MEMPERKUAT KESELAMATAN APLIKASI WEB: MENGGUNAKAN
RANGKA KERJA PENILAIAN WAF SUMBER TERBUKA**

ZHOU HAO

**PROJEK YANG DIKEMUKAKAN UNTUK MEMENUHI SEBAHAGIAN
DARIPADA SYARAT UNTUK MEMPEROLEH IJAZAH
SARJANA KESELAMATAN SIBER**

**FAKULTI TEKNOLOGI DAN SAINS AND MAKLUMAT
UNIVERSITI KEBANGSAAN MALAYSIA
BANGI
2025**

DECLARATION

I hereby declare that the work in this thesis is my own except for quotations and summaries which have been duly acknowledged.

12 February 2025

ZHOU HAO
P128640

LIBRARY FETSM

ACKNOWLEDGEMENT

First and foremost, I am immensely and sincerely grateful to my parents for providing financial support for my study abroad. Their unwavering encouragement and support have been my greatest strength in overcoming life's challenges and pursuing my education at UKM.

I would like to express my deepest and most loyal thanks to my supervisor, Dr. Ravie Chandren Muniyandi, for his invaluable advice, guidance, and patience throughout my project. His insights and expertise have been instrumental in the completion of this work.

I am also grateful to my classmates for their help and companionship. They offered valuable suggestions and assisted me in adapting to life in Malaysia. Their support and shared experiences have made my journey enriching and fulfilling.

Finally, I would like to extend my heartfelt gratitude to everyone who has contributed, directly or indirectly, to the success of this project. Thank you for being a part of this meaningful journey.

LIBRARY UKM

ABSTRAK

Tembok Api Aplikasi Web (WAF) sangat penting untuk mengurangkan ancaman siber seperti suntikan SQL, skrip lintas tapak (XSS), dan kerentanan muat naik fail. Kajian ini menilai tiga WAF sumber terbuka—ModSecurity, WebKnight, dan SamWAF—dengan menggunakan rangka kerja DFPUE yang baru dibangunkan, yang menilai ketepatan pengesanan, positif palsu, prestasi, kebolehgunaan, dan pengalaman pengguna. Penyelidikan ini melibatkan simulasi serangan dunia nyata dalam persekitaran terkawal, menggunakan teknik pintasan seperti pengekodan dan pengaburan beban untuk menguji keupayaan pengesanan. Data eksperimen dikumpul dan dianalisis untuk membandingkan kekuatan dan kelemahan setiap WAF. Hasil kajian menunjukkan bahawa WebKnight cemerlang dalam menangani serangan suntikan SQL yang dikodkan tetapi menghadapi batasan kebolehgunaan disebabkan oleh konfigurasi yang kompleks. SamWAF, dengan antara muka yang mesra pengguna, menunjukkan prestasi baik dalam menyekat ancaman asas tetapi menghadapi cabaran dalam senario serangan lanjutan seperti XSS tersimpan. ModSecurity, walaupun menawarkan keupayaan pengesanan yang kuat, memerlukan pengetahuan konfigurasi yang mendalam, yang menjadi cabaran bagi pengguna yang kurang berpengalaman. Penemuan ini mengesahkan keberkesanan rangka kerja DFPUE dan menyediakan pandangan praktikal untuk meningkatkan reka bentuk dan prestasi WAF. Kajian ini menyumbang kepada peningkatan pertahanan keselamatan siber dengan mengenal pasti kekurangan dalam pelaksanaan WAF semasa dan mencadangkan strategi yang disasarkan untuk pengoptimuman masa depan, membolehkan pentadbir melindungi aplikasi web dengan lebih baik daripada ancaman yang semakin berkembang.

ABSTRACT

Web Application Firewalls (WAFs) are essential for mitigating cyber threats such as SQL injection, cross-site scripting (XSS), and file upload vulnerabilities. This study evaluates three open-source WAFs—ModSecurity, WebKnight, and SamWAF—using a newly developed DFPUE framework, which assesses detection accuracy, false positives, performance, usability, and user experience. The research involves simulating real-world attacks in a controlled environment, employing bypass techniques like encoding and payload obfuscation to test detection capabilities. Experimental data is collected and analyzed to compare the strengths and weaknesses of each WAF. Results reveal that WebKnight excels in handling encoded SQL injection attacks but faces usability limitations due to complex configurations. SamWAF, with its user-friendly interface, performs well in blocking basic threats but struggles with advanced attack scenarios like stored XSS. ModSecurity, while offering strong detection capabilities, demands extensive configuration knowledge, presenting challenges for less experienced users. The findings validate the DFPUE framework's effectiveness and provide actionable insights for improving WAF design and performance. This study contributes to enhancing cybersecurity defenses by identifying gaps in current WAF implementations and proposing targeted strategies for future optimization, enabling administrators to better safeguard web applications against evolving threats.

TABLE OF CONTENTS

		Page
DECLARATION		iii
ACKNOWLEDGEMENT		iv
ABSTRAK		v
ABSTRACT		vi
TABLE OF CONTENTS		vii
LIST OF TABLES		x
LIST OF ILLUSTRATIONS		xi
LIST OF ABBREVIATIONS		xiii
CHAPTER I	INTRODUCTION	
1.1	Introduction	1
1.2	Research Background	2
1.3	Problem Statement	4
1.4	Research Questions	5
1.5	Research Objective	5
1.6	Research Scope	6
CHAPTER II	LITERATURE REVIEW	
2.1	Introduction	7
2.2	Method	8
	2.2.1 Literature Criteria	8
	2.2.2 Search Process	8
	2.2.3 Screening	9
2.3	Bypass Techniques Overview	10
	2.3.1 SQL Injection Attacks	10
	2.3.2 Cross-Site Scripting Attacks	12
	2.3.3 WebShell Transformations	13
	2.3.4 File Upload Vulnerability	15
2.4	Waf Evaluation Methods Overview	17
2.5	The Necessity of The Development of Wafs	18
2.6	The Future of Waf Overview	19
2.7	Data Privacy and Legal Considerations	21

2.8	Research Gaps	22
2.9	Chapter Summary	23
CHAPTER III METHODOLOGY		
3.1	Introduction	24
3.2	Research Design	25
	3.2.1 Data Survey - Step 1	25
	3.2.2 Simulated Attack - Step 2	26
	3.2.3 Data Analysis - Step 3	27
	3.2.4 Evaluation and Suggestion - Step 4	28
3.3	Experimental Setup	28
	3.3.1 WAF Selection, Version and Configuration	28
	3.3.2 Virtualized Environment	29
	3.3.3 Virtual Machines and Operating Systems	30
3.4	Data Collection and Analysis	33
3.5	Evaluation Framework Design	34
	3.5.1 WAF Evaluation Scope	35
	3.5.2 WAF Evaluation Weight Allocation Design	36
	3.5.3 WAF Evaluation Methodology	37
	3.5.4 WAF Criteria Determination and Acceptance	43
3.6	Ethical Considerations and Limitations	44
3.7	Chapter Summary	45
CHAPTER IV EVALUATION AND RESULTS		
4.1	Introduction	47
4.2	Experimental Procedure	48
	4.2.1 Attack Detection Rate Testing	55
	4.2.2 False Positive Rate Testing	73
	4.2.3 Performance Metrics Testing	75
	4.2.4 Usability and User Experience Testing	79
4.2	Data Collection, Induction and Analysis	81
4.3	Result Analysis and Evaluation	84
4.4	Comparison with The Existing Evaluation Methods	87
4.5	Chapter Summary	89
CHAPTER V CONCLUSION AND RECOMMENDATION		
5.1	Introduction	91
5.2	Finding and Discussion	92

5.3	Limitations of The Study	93
5.4	Recommendations for Improving Waf Performance	94
5.5	Recommendations for Future Research	95
5.6	Final Conclusion	96

REFERENCES	98
-------------------	-----------

APPENDICES

Appendix A	Screenshot of Feedback to Waf Developers and Related Communities	102
------------	--	-----

LIBRARY FTSM

LIST OF TABLES

Table No.		Page
Table 2.1	Code Snippets for Trojan Horse Attacks	15
Table 3.1	Kali-linux _Attack Machine	31
Table 3.2	Windows Server 2019 IIS 10_Target Machines (Modsecurity)	31
Table 3.3	Windows Server 2019 IIS 10_Target Machines (WebKnight)	31
Table 3.4	Windows Server 2019 IIS 10_Target Machines (SamWAF)	32
Table 3.5	Windows 10_Host_Attack Machine	32
Table 3.6	Windows 10_Attack Machine	32
Table 3.7	WAF Criteria of Attack Detection Testing	38
Table 3.8	WAF Criteria of Performance Testing	40
Table 3.9	WAF Criteria of Performance Testing (Configuration)	41
Table 3.10	WAF Criteria of Performance Testing (Management)	42
Table 3.11	WAF Criteria of Performance Testing (Monitoring)	43
Table 4.1	Usability and User Experience Test Results	80
Table 4.2	Detection Rate Results	83
Table 4.3	False Positive Rate Results	83
Table 4.4	Performance Results	84
Table 4.5	Usability and User Experience Results	84
Table 4.6	Overall Evaluation Results	86
Table 4.7	Comparison of WAF Evaluation Methods: Strengths and Limitations	89

LIST OF ILLUSTRATIONS

Figure No.		Page
Figure 1.1	How Does a WAF Work? (2023)	1
Figure 2.1	How SQL Injection Attack Works. (Hlaing and Khaing, 2020)	11
Figure 2.2	A brief overview of the four categories of cross-site scripting vulnerabilities. (Sonkarlay, 2022)	13
Figure 2.3	Web Application Firewall vs Network Firewall. (A10,2024)	19
Figure 3.1	Process About This Study	25
Figure 3.2	Flow Chat of WAF Evaluation Framework	35
Figure 4.1	Index Interface Source Code	49
Figure 4.2	SQL Test Interface Source Code	50
Figure 4.3	Upload Test Interface Source Code	50
Figure 4.4	XSS Test Interface Source Code	51
Figure 4.5	Database Connection Source Code	51
Figure 4.6	Database Data File Content	52
Figure 4.7	Webknight Log and Configuration File Changes (Header)	53
Figure 4.8	Webknight Log and Configuration File Changes (Path)	53
Figure 4.9	Restart IIS Server	54
Figure 4.10	SamWAF Configuration Page	55
Figure 4.11	Basic SQL Injection Testing Results	56
Figure 4.12	URL-Encoded SQL Injection Results	57
Figure 4.13	Double-layer URL Encoding Results	58
Figure 4.14	Single-layer IBM037-Encoded SQL Injection Results	59
Figure 4.15	Index Interface Source Code	59
Figure 4.16	Dynamically Generated Payloads Results	60

Figure 4.17	Time-Based Blind SQL Injection Results	61
Figure 4.18	Error-Based Information Disclosure Injection Results	62
Figure 4.19	Basic XSS Results	63
Figure 4.20	URL-Encoded XSS Results	63
Figure 4.21	Advanced XSS Bypass Results	64
Figure 4.22	DOM-Based XSS Script	65
Figure 4.23	DOM-Based XSS Result	66
Figure 4.24	Basic File Upload Result	67
Figure 4.25	Double Extension Bypass Result	67
Figure 4.26	Hidden Code Fragments Result	68
Figure 4.27	MIME Type Spoofing Result	69
Figure 4.28	Base64-Encoded Upload Result	70
Figure 4.29	Directory Detection Script	71
Figure 4.30	Directory Detection Result	71
Figure 4.31	Directory File Download Script and Results	72
Figure 4.32	WebShell Directory Enumeration Result	72
Figure 4.33	False Positive Rate Test Python Code	74
Figure 4.34	False Positive Rate ModSecurity Results	74
Figure 4.35	False Positive Rate SamWAF Results	75
Figure 4.36	Performance Metrics Test Python Code	76
Figure 4.37	Performance Metrics ModSecurity Results	77
Figure 4.38	Performance Metrics WebKnight Results	78
Figure 4.39	Performance Metrics SamWAF Results	79

LIST OF ABBREVIATIONS

UKM	Universiti Kebangsaan Malaysia
SQL	Structured Query Language
XSS	Cross-Site Scripting
DOM	Document Object Model
WAF	Web Application Firewall
HTTP	HyperText Transfer Protocol
OWASP	Open Web Application Security Project
WASC	Web Application Security Consortium
DFPUE	Detection Rate, False Positive Rate, Performance Metrics, Usability, and User Experience
MIME	Multipurpose Internet Mail Extensions
URL	Uniform Resource Identifier

CHAPTER I

INTRODUCTION

1.1 INTRODUCTION

With the current rapid development of technology, digital has become the hallmark of this era, and the Internet has become a fundamental part of our daily lives, linking people around the globe and making the exchange of information, ideas and services possible. However, there are two sides to everything, and where there are advantages, there are disadvantages. With the increasing interconnectivity of the network, cyber-attacks and security breaches have become one of the current pain points of the Internet. Currently, the most important means to defend against these attacks is Web Application Firewalls (WAFs). WAFs is used to mitigate common web-based attacks such as cross-site scripting and SQL injection. WAFs may come in the form of a server plugin, filter, or appliance (2023). But are WAFs always secure? Figure 1.1 below shows how WAFs work.

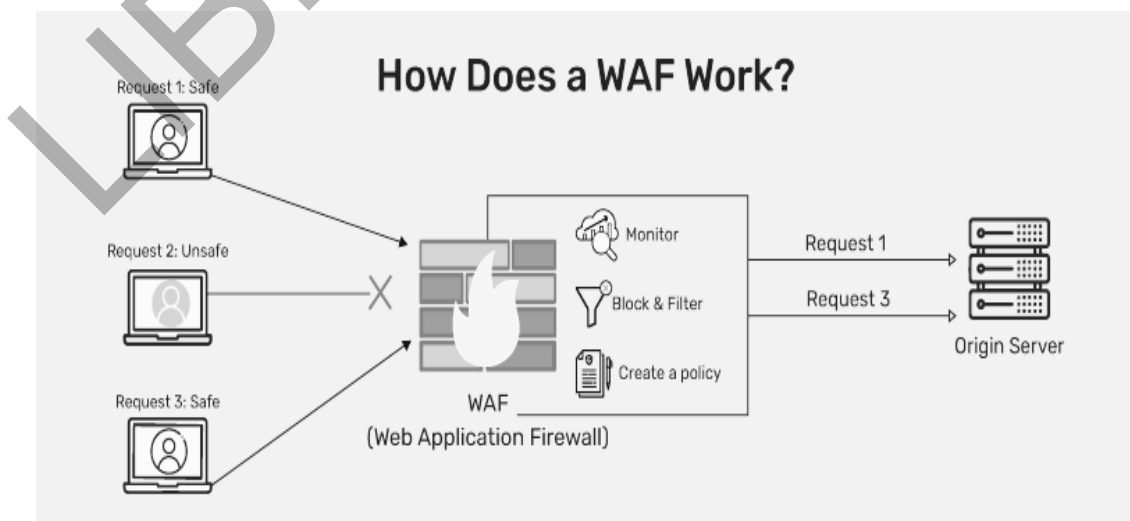


Figure 2.1 How Does a WAF Work? (2023)

From the figure, we can see that WAFs intercept malicious traffic by monitoring, blocking, filtering and creating policies, that is, the insecure request 2 in the figure, but if the request 2 bypasses the WAFs security policy by means of WAFs by replacing, splicing, and morphing, then the WAFs will become meaningless the user's data will leaked, causing irreversible losses. So, what is the security performance of existing WAFs? How effective is the blocking of popular Injection Attacks, Cross-site scripting attacks, and File Upload Vulnerabilities?

As a security officer, if security is only made in terms of safety, then it must be unsafe. The literal meaning of this statement may be a bit contradictory, but to think of it in another way, if a body armour is only wannabe material secure without considering and testing the penetration effect of various bullets, then this body armour can't be used in the real world after all. Therefore, this research aims to explore ways to bypass existing WAFs and identify their limitations against advanced cyber threats in order to protect valuable data and systems from unauthorised access. This investigation targets techniques for escaping detection executing injection attacks, Cross-site scripting attacks, and file upload vulnerabilities. We plan to do this in a variety of ways, having a profound look into WAF setups currently in use, detecting their unorthodoxy's as well as identifying methods to evade their defences. Its confirmation will display task boundaries in web application firewalls and be the first step towards a clearer vision of used techniques.

1.2 RESEARCH BACKGROUND

The modern era of cybersecurity faces increasing challenges due to the rising frequency and sophistication of cyber-attacks. Studies indicate that malicious traffic floods the internet alongside legitimate searches, with an estimated 4% probability of encountering malicious content. Firewalls, a traditional method of defence, enhance security by blocking harmful requests at the network's entry point. This firewall, the traditional method of protection, raises the bar for the security level by dropping malicious requests at the gate. Although protective measures are in place such as filtering, some attackers optionally bypass the firewall stage by utilizing the evasion mechanism (2020). On top of this, digital technology is changing at a fast pace, and

media has become a thing of the past. Thus, reliable security systems must be implemented to prevent rapid development and theft from unauthorized personnel.

Web Application Firewalls (WAFs) are of vital importance when it comes to the toolbox of safety measures to be used by web-based attacks, such as cross-site scripting (XSS) and SQL Injection. Attachments provided by servers are often, but not exclusively, plugins, filters, and appliances.

On the other hand, the efficiency and reliability of WAFs, mentioned by experts, come into our discourse. Apart from that, wrongdoers are set to be always seeking to devise alternative means of making users unaware of their objectives. This development entails the disposal of security checks and the exploration of the weak points in the system. We have recently seen some instances of attackers using such complex methods. They were able to perform WAF bypasses as well as Webshell detections, uploads, and injections, and a few more. For example, Bypass Frameworks, a tool that hackers were using in 2016, is a hacking tool that allows them to bypass a web application firewall. Different researchers showed in 2017 that there were vulnerabilities, which in turn let them bypass various Web Application Firewalls using some charset encoding techniques for different characters. Hazlitt Threat Intelligence (Zscaler) in 2018 discovered an APT attack named Phantom Lance. This attack was the case of an application's devices being targeted in a web application firewall bypass for evil purposes. Akamai Technologies published a report in 2019 about a new type of attack targeted against web application firewalls. The attack exploited HTTP control characters located in the header to restrict the firewall from discovering and denying dangerous demands. In 2020, the vulnerability was disclosed for the first time when security researchers found the tool called BoltWire that uses some of the Web application firewall anti-bypass rules.

These situations remind us that even if conditions are relatively secure, they can still be circumvented or attacked. Hence, regular safety check-ups and system upgrades are critical to guarantee HTTP WAF security for web applications. The typical practice of most companies is to carry out vulnerability management by employing information security experts to carry out penetration tests. The primary purpose of a penetration test

is to identify defects within a computer network, system, or web application to pinpoint the loopholes that might be exploited by an unauthorized user (Margaret Rouse, 2011). However, it differs since vulnerability assessment is done when the WAF is deployed for dynamic resources of the test to improve the testing conditions and have a direct impact on the real test results (Stefan Schumacher 2019). Consequently, overcoming the WAF in the scope of the assessment is the most difficult point in determining the true pass or fail results.

1.3 PROBLEM STATEMENT

The COVID-19 pandemic in 2020 had a profound impact on global network communications, exemplified by well-known vulnerabilities such as the Log4j flaw. The emergence of such vulnerabilities during critical periods significantly disrupted business communications and underscored the importance of Web Application Firewalls (WAFs) in cybersecurity. Since then, the implementation and development of WAFs have become a key strategy for leading players in the cloud security and infrastructure market, such as Alibaba Cloud, Tencent Cloud, and Cloudflare, to protect their networks from modern cyber threats.

Web application firewalls (WAFs) work to find and prevent harmful activities in web applications that are mostly injection attacks, Cross-site scripting (XSS) attacks, and file upload vulnerabilities. These issues have stood long-term, and many appear frequently in the OWASP Top 10. The major reason for their persisting is the frequent development of new variations by hackers. A case in point, a SQL injection, a type of injection attack has been in the first place of the list for years, proven demonstrating the fact that it is hard to fix such vulnerabilities. The complexity and diversity of web applications provide further ammunition for cyber theft that takes place, by which WAFs can hardly be successful in the identification of cyber threats and protection from them.

File upload vulnerabilities, such as, allow attackers to upload illegal files to web servers, which means an extra layer of complexity is added to the security of web applications. Even though WAFs have been used for many years, there is a restriction on their ability. In addition, WAFs themselves may also have vulnerabilities that can be

exploited. For instance, in the year 2019, the attack was identified by Akamai Technologies, which inserted HTTP control characters in the headers to pass through the WAF security system controls, which is a clear demonstration of the necessity to fix the vulnerabilities within the WAF implementation.

Being faced with these challenges, one must check and test the current WAF bypass strategies to give web applications a higher level of security so that they will be well covered from the worsening cyber-attacks.

1.4 RESEARCH QUESTIONS

1. How can the DFPUE framework be systematically developed to comprehensively assess the performance of open-source web application firewalls?
2. How effective is the proposed DFPUE framework in evaluating the performance of open-source web application firewalls based on Detection Rate Testing, False Positive Testing, Performance Testing, Usability Testing, and User Experience Testing?
3. What are the key differences in interception effectiveness among different open-source web application firewalls when tested using the proposed DFPUE framework?

1.5 RESEARCH OBJECTIVE

1. To develop an DFPUE evaluation framework for assessing the comprehensive performance of open-source web application firewalls.
2. To evaluate the open-source web application firewalls, conducting rigorous bypass testing and assessing intercept data for the proposed DFPUE framework.

1.6 RESEARCH SCOPE

1. The existing WAFs tested are all open source and free.

2. Test the current mainstream injection attacks, Cross-site scripting attacks and file upload vulnerability bypass methods.?
3. The test environment is limited to local or authorised sites only.

LIBRARY FTSM

CHAPTER II

LITERATURE REVIEW

2.1 INTRODUCTION

A Web Application Firewall (abbreviation: WAF) is a specific form of application firewall used to filter, monitor and block HTTP traffic through web services (Wikipedia). Through the analysis of HTTP traffic, it can defend against most of the known vulnerabilities such as SQL injection, cross-site web scripting, file inclusion and other common types of WEB vulnerability attacks. Because of the power of WAF, and the constant iterations and innovations of researchers in the field, it plays a crucial role in today's web security. Figuratively speaking, WAF is like a city wall, all the time to resist the hacker's various forms of attack, but the wall is not impregnable, it may also have some dark or special ways to reach the inner city, which is often referred to as the WAF vulnerability and WAF bypass. To reduce such vulnerabilities to be discovered and exploited by hackers, regular WAF scanning, testing, evaluation and repair is a crucial part. How to properly and efficiently test and evaluate WAF is the purpose of this chapter.

As we enter the hallowed halls of the literature review, we should be humble enough to appreciate the depth of the old wisdom, "We all stand on the shoulders of giants." The statement emphasizes the significant association of knowledge and scholarship being the result of the hard work and the research of the former on which we are appreciative of. This paragraph first investigates the sources of scholarship, then relegates the works that have been studied, and culminates with researchers' own contributions.

We will then delve deeply into the literature, with primary concentrations on bypassing techniques, WAF evaluation methods, and the searches. The main objective of this chapter is to obtain the existing research findings, pinpoint the most crucial things, and give a critical evaluation of open-source WAFs, marking their preferences and restrictions in comparison with other solutions.

2.2 METHOD

2.2.1 Literature Criteria

For the purpose of guaranteeing trustworthiness, power, and hence the wholeness of only the selected literature, all the literature in this section is supposed to go through the screening criteria list:

All or any studies that are the whole of literature, articles, books, and websites must have been published and should be publicly available. To guarantee that the research process is visible and can be repeated, all sources should be in the public domain. Anything that is included in the literature review should be directly related to this study. The critical factor in the selection of literature was the justification of the source material; this led to a focus on papers from well-established academic databases such as Google Scholar, IEEE Xplore, and a variety of other academic repositories. These sites are well-known for their intensive peer review process, which serves as a means of making research more authentic and reliable. Even though we are concentrating on the academic databases that are most relevant, we are including other platforms that provide a more comprehensive view thus enriching the dialogue. This will probably contain references to industry reports, government publications, and serious websites with a focus on cybersecurity and web application development. In order to ensure that the selected literature is timely and relevant to this study, we will give priority to the latest publications and articles in the past five to ten years. However, seminal works and basic literature that have made great contributions to the topic can also be included in this literature review.

2.2.2 Search Process

The literature search process will employ a system of stages that crucially guarantee that literature encompassing a vast number of the most pertinent research is found. We will be searching for various online literature databases including Google Scholar, IEEE Xplore, SpringerLink, etc. Multiple online authority websites such as <https://owasp.org/>, <https://learn.microsoft.com/>, <https://m.freebuf.com/>. A few offline books are of course to be mentioned: "White Hat on Web Security", and "Web Security Attack and Penetration Testing Practical Guide".

Refinement of literature search through the utilization of keywords, tags, and citations and its relevance ensured. The keywords are "SQL injection", "cross-site scripting attack", "web application firewall", "WebShell ", "File Upload Vulnerability", "WAF Evaluation" "WAF Bypass" and "Cybersecurity Laws ". The tags include "site", "-", "... " and "in the title".

2.2.3 Screening

Using both internet and offline sources, we went through a great number of studies that might be interesting to us in our research. To limit our choice and confirm a high level of relevance to the purpose of the research, we painstakingly checked every single study by a series of strict criteria, consisting of publication date, source credibility, and conformity with the research questions.

By using the thematic approach, we arranged and summarized the literary insights in a meticulous way. This thorough classification was achieved by warehousing articles of recurring themes and topics, such as the WAF theory, WAF evaluation methods, and the other kinds of bypassing techniques. The purpose of organizing the review in this way is to provide the reader with a coherent and exhaustive account of the contemporary situation surrounding the assessment of open-source WAFs and the vulnerabilities they address.

For the reader's convenience and reference, we have made an exhaustive bibliography in Appendix reference to go with the most condensed literature review

section. This trouble-shooting tool consists of academic works, research papers, and articles. This is a selection we have painstakingly studied and skimmed throughout the study.

2.3 BYPASS TECHNIQUES OVERVIEW

2.3.1 SQL Injection Attacks

Infiltration attacks using SQL injections include the insertion of SQL queries into application input data, allowing attackers to access sensitive information, tamper with database content and even execute arbitrary commands. These attacks act as a grave danger towards data confidentiality, authenticating mechanisms, authorizing systems and data integrity. Despite the prevalence of SQL injection vulnerabilities, their risks can be mitigated through proper input validation, parameterized SQL statements and, to some extent, the use of stored procedures. However, it must be recognized that no single solution is completely immune to SQL injection attacks and therefore a layered approach to security implementation is required (Kingthorin, 2024).

SQL injection attacks are a significant threat to database-driven web applications, posing a risk of exposing critical confidential information to hackers. Hackers exploit the absence of input validation for disadvantages in the way dynamic queries are processed, which leads them to take over the execution process of malicious SQL queries. If the updater is not careful about it, injected SQL queries endanger the execution of malicious code. This is because attackers are missing the injection of SQL code into input panels. If the SQL code is designed for Modification and Select operations, tautologies, and UNION queries can be introduced to bypass default system commands and retrieve unauthorized data. Comment hopping, string parameters, and tautologies like 'OR 1=1' are the methods of offenders who misuse the vulnerability to access key data without permission. To prevent potential SQL injection vulnerabilities, developers need to either turn off error messages or harden the server error message setup and use secure coding practices (Sadeghian et al., 2013).

Aliero et al. (2020) examined the effectiveness of Web Application Firewalls (WAFs) in mitigating SQL injection attacks. Their study assessed a range of tools and

techniques designed to detect and prevent such attacks. The findings suggest that while WAFs play a crucial role in enhancing web security, they should not be regarded as a standalone solution. Instead, they should be complemented by secure coding practices, proper database configuration, and rigorous vulnerability assessments. The study also highlighted limitations in existing security tools, emphasizing that WAFs primarily serve as an additional layer of protection rather than a foolproof security measure.

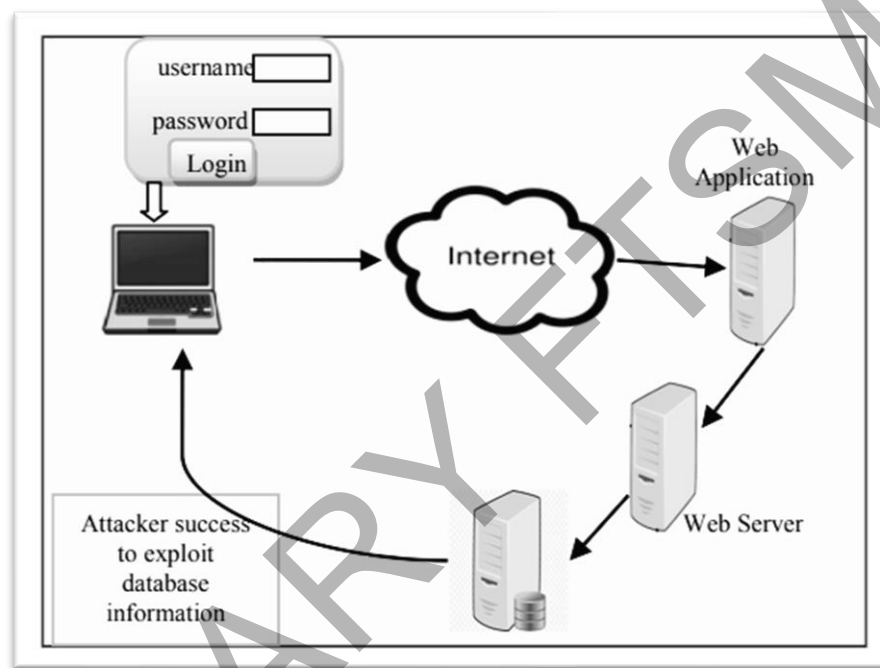


Figure 2.1 How SQL Injection Attack Works. (Hlaing and Khaing, 2020)

The study by Hlaing and Khaing (2020) focuses on the necessity of addressing vulnerabilities associated with SQL injection attacks in web applications. The authors present an approach involving the creation of a lexicon and tokenizing input query statements to detect and prevent SQLIA. The proposed technique utilizes reserved word-based lexicons to identify malicious SQL statements. Their experiments with detection and prevention technologies for SQL injection attacks yield satisfactory results. Various related works have proposed different techniques, such as ASCII-based string matching, tokenization, and dynamic analysis, to mitigate SQLIA risks. Addressing SQL injection vulnerabilities is essential for preventing unauthorized access, data extraction, and ensuring data integrity in backend databases.

SQL injection vulnerability is one of the most common Web attacks, in the book *Web Security Attacks and Prevention - Penetration Testing Practical Guide*, the author introduces a kind of SQL injection to sweep through the WAF method, different methods corresponding to different strengths of the WAF. These seven bypass methods are case bypass, replace the keyword bypass, encoding bypass, bypassing convergent annotations, HTTP parameter contamination, chunking transfer and SQLMap bypass. Keyword bypass can be used to adopt the keyword bi-writing, the same price word replacement and special number splicing. Encoding bypass can take URL encoding and Base64 encoding.

Keyword doubling is mainly to use the incompleteness of WAF to verify the string only once or the filtered string is incomplete, the code is as follows:

```
xxx.com/index.php?id=-3 UNION ON SELselectECT 1,2,3
```

2.3.2 Cross-Site Scripting Attacks

Cross-site scripting attacks (XSS) are a serious security threat that falls under the category of injection attacks. In an XSS attack, malicious script is injected into an otherwise trusted website and delivered to the end user via a web application. Vulnerabilities for successful attacks usually exist in web applications, where malicious code is successfully injected when the application does not validate or encode user input. An attacker exploiting an XSS vulnerability can send a web link or content containing malicious code to a user, and once the user views the content, the browser executes the script, leading the attacker to steal sensitive user information, session tokens, etc., or even tamper with the HTML page content's attacks consist of two main types, reflective and stored, depending on how the malicious code reaches the user's browser, respectively. In addition, XSS attacks can lead to serious consequences, including session hijacking, information leakage, and malware installation, posing a serious threat to both individual users and organizations. Effective protection against XSS attacks requires a combination of appropriate security measures and tools, such as input validation, output encoding, and HTTP TRACE support disabling (KirstenS, 2024).

The server-side procedures of some Web applications may have done part of the security filtering work or used security products, but the security filtering in many scenarios is not perfect enough, and a simple deformation of the XSS Payload may bypass the defence mechanism (Wu hanqing and Ye min, 2023).

Various techniques for bypassing security measures, such as cross-site scripting (XSS) filters and firewalls, have been explored in previous research. A study published in 2020 examines methods including obfuscation, double-coding, and file inclusion, as well as sub-techniques such as null byte injection, path truncation, and PHP wrappers. These insights help security researchers understand vulnerabilities in firewalls and develop more effective firewall policies (2020).

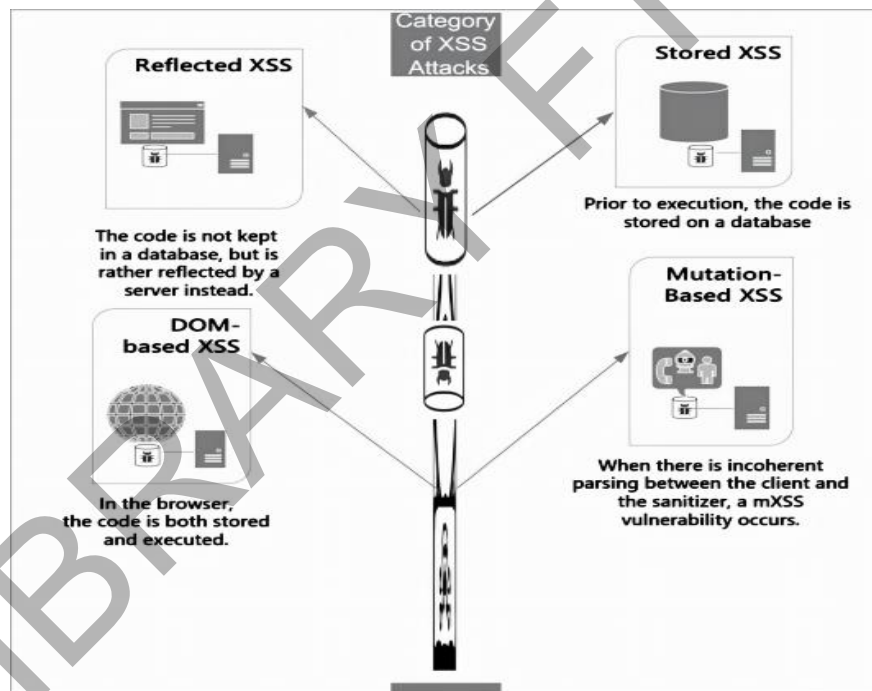


Figure 2.2 A brief overview of the four categories of cross-site scripting vulnerabilities. (Sonkarlay, 2022)

2.3.3 WebShell Transformations

Common WebShell backdoor functions: eval, assert, exec, shell exec, passthru, popen and other functions, but at present, many security products have been sensitive to WebShell detection, common WebShell Trojans have not been able to directly over the

WAF and so on, so D Shield as a detection of the medium to test how high the level of WebShell detection is under constant change (vFREE, 2022).

A web page may contain thousands of code lines or more in each file; therefore, it is very difficult and sometimes impossible to review the source code manually to detect malicious file such as webshell. So, finding and detecting webshell inside webpage application are necessary for securing websites (Truong Dinh Tu, 2014).

WebShell deformation may be our most common bypass means, but it is still a good means of dynamic detection, PHP comes with a large number of strings to deal with deformation and encoding functions, we can use this to operate on our webshell (Anon, 2024).

Yihuai Cao and Wei Chen et al. used specific models and datasets to analyze and detect encrypted obfuscated web shell traffic in crypto-obfuscated WebShell detection for high-speed web traffic. The experiments highlight the importance of distinguishing features in encrypted obfuscated web shell traffic for accurate detection. For example, using dataset B enriched with different sample types enhances the model's recognition capability. The results show that precision, recall and F1 scores change when changing the training and test datasets, emphasizing the need for well-structured training data for optimal detection performance (Yihuai Cao et al, 2022).

Table 2.1 Code Snippets for Trojan Horse Attacks

TYPE	basic Trojan horse (type of computer virus)
PHP	<?php @eval(\$ POST['key']);?>ect
ASPX	<%@ Page Language="Jscript"%><%eval(Request.Item["g"],"unsaf");%>ect
ASP	<% eval request("cmd") %>ect
JSP	<pre data-bbox="456 517 1390 808"> <%!class U extends ClassLoader{ U(ClassLoader c) { super(c); }public Class g(byte [lb){ returnsuper.defineClass(b,0,b.length); }}%><% String cls- request.getParameter("ant"); if(cls!=null){newU(this.getClass0.getClassLoader().g(new sun.misc.BASE64Decoder(). decodeBuffer(cls)).newInstance(.equals(pageContext);)%>etc (2018) </pre>

2.3.4 File Upload Vulnerability

A file upload vulnerability is when a Web server allows a user to upload a file to its file system without sufficient verification of information such as file name, type, content, or size. Failure to enforce these restrictions correctly can mean that even basic image upload functions can be used to upload arbitrary and potentially dangerous files. This may even include server-side script files that enable remote code execution (PortSwigger).

The unrestricted file upload vulnerability poses a significant threat to applications and could lead to system crashes, client-side attacks, or simple tampering. The risk factor for this vulnerability is extremely high, easily discovered and exploited by attackers, and widespread. To mitigate this vulnerability, appropriate handling of the file metadata by the user and a complete audit of the file handled by the application are the experts' suggestions (OWASP).

Because of the persistently high rate of digital offenses and faults in web applications, Imam Riadi and Eddy Irawan Aristianto, as they did their postgraduate studies, discovered a loophole in the system of bug free image file unrestricted uploading in applications such as Oscommerce and put forth the solution that patching should be done. The procedure includes identifying the code, implementing, and then testing the patches to strengthen security and launching of attacks. Predicting

vulnerability and methods of security testing, for example, penetration testing is very important to limit the number of possible attacks. The critical problem the article addresses is the difficulties in auditing PHP source code and the use of security testing to fight threats in dynamic web services (Iman Riadi and Eddy Irawan Aristianto, 2016).

Lack of security awareness and delay in security testing by developers are the reasons for insecure applications. Collaboration between client and server is vital to prevent attacks. Various studies have highlighted common vulnerabilities such as remote code execution and SQL injection. File upload vulnerabilities can occur due to lack of authentication or bypassing client-side filters (Karishma Pooj and Sonali Patil, 2016).

Xu et al. (2018) discuss four key factors contributing to file upload vulnerabilities, including insufficient file type validation, lack of strict file size limitations, and the ability to manipulate file paths and filenames. They also propose five methods for bypassing WAFs: line break bypass, multiple equals sign bypass, file name plus ";" bypass, null byte (00) truncation bypass, and file name plus "" bypass. Their analysis highlights the need for stricter validation mechanisms to mitigate these vulnerabilities effectively.

2.4 WAF EVALUATION METHODS OVERVIEW

Web application firewalls have emerged in response to the increasing prevalence of endpoint-oriented web attacks. Watson (2007) discusses various attack techniques and vulnerabilities associated with web applications in *The Evolution of Web Application Attacks*. These include topics such as drive-by downloads, XSS attacks, third-party content vulnerabilities, and the rise of web application botnets. The article highlights a shift in web attack targets, focusing on end-user data and the growing threat of malicious web servers to web clients.

Evaluating the effectiveness of firewall defenses is crucial to this research. Mukhtar and Azer (2020) confirm the effectiveness of ModSecurity in mitigating SQL injection attacks through a three-phase test in *Evaluating the ModSecurity Web Application Firewall Against SQL Injection Attacks*.

Firewalls play a vital role in network security by regulating traffic between networks of different trust levels. Firewalls should be integrated throughout the network to enhance security. Evaluations and comparisons of various firewalls such as Cisco ASA, Checkpoint SPLAT and OpenBSD PF have shown differences in performance, cost and security features. Firewalls are critical to compliance standards such as ISO 27002, PCIDSS and COBIT that emphasise efficient configuration, management and auditing. Despite the increasing deployment of firewalls, there is no standardised methodology for evaluating their performance due to the critical role of IT security in protecting business information and processes (Sheth C and Thakker R, 2011).

The identification of the operating system by the Nessus and Nmap programs is affected by the accuracy and precision as reported by Sun-young Im and her fellow students in their paper. It was observed that only three Windows 7 Enterprise SP1 systems were correctly captured by the Nessus when the firewall was up, but meanwhile, Linux 3.2 was not identified by Nmap. The accuracy and precision of identification results are a crucial factor in evaluating the performances of these tools (Sun-young Im et al, 2016).

The Web Application Firewall Evaluation Criteria (WAFEC) project, which is a joint effort of The Web Application Security Consortium (WASC) and OWASP, sets the standard for web application firewalls (WAFs) through the WAFEC project. WAFs act as a critical element in safeguarding the sites against the advancements of the attacks and giving support in addition to the traditional network firewalls and intrusion detection systems. The WAFEC serves as a tool for informing the stakeholders, as they can make decisions on WAF selection which is very important for security improvement in web applications. Work commenced in 2006, the project under the aegis of Tony Turner saw an increase of its activities in 2015 as it was the solution to the issues identified in the 2013 OWASP. WAFEC, by designing full-fledged, accurate, and impartial rating standards, address these emerging threats in the area of web application security. They are actually the pioneer in the sequence of cybersecurity improvements in the field as per the statement (Anon, 2016).

2.5 THE NECESSITY OF THE DEVELOPMENT OF WAFS

In *Network Security Models and Configuration* (2018), the authors explore the necessity of WAFs, discuss the advantages and disadvantages of positive and negative policy-based attack detection models, and highlight that web servers using default configurations may remain vulnerable despite the presence of firewalls.

Holm and Ekstedt (2013) estimate the effectiveness of WAFs in preventing injection attacks conducted by professional penetration testers, considering four key factors: the availability of an experienced operator to monitor the WAF, whether an automated black-box tool was used when adjusting the WAF, the expertise level of WAF personnel, and the effort invested in fine-tuning the WAF. By summarizing the assessments of 49 domain experts, they derived effectiveness estimates for WAFs under 16 operational scenarios. The results indicate that when all measures were implemented, the median prevention rate of WAFs was 80%, whereas if no measures were taken, the median prevention rate dropped to 25%. These findings suggest that WAFs provide substantial protection against common web attacks but require skilled personnel for maintenance and policy adjustments to maximize their effectiveness.

In a study by Sahin and Sogukpınar (2017), the GET, POST, and PUT sections of the HTTP protocol structure were examined in detail to enhance web attack prevention. The method employed an anomaly-based approach to attack detection, identifying key expressions used in known attack methods and their relationship to attack patterns. Accordingly, the gain value of the selected properties was calculated, with the magnitude of these values helping to distinguish between normal and anomalous requests.

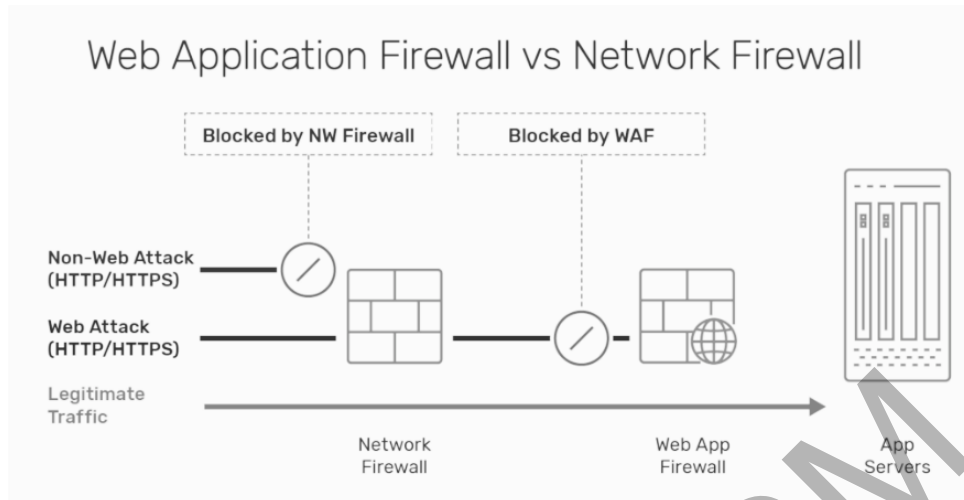


Figure 2.3 Web Application Firewall vs Network Firewall. (A10,2024)

The market for free WAF solutions in 2024 offers a wealth of options, from NGINX ModSecurity to Open Application Security and Cloudflare. These solutions not only protect your website from common web threats, but also offer features such as machine learning capabilities, customized rule sets and integration with existing technology stacks. Some of these solutions also offer features such as real-time threat intelligence, DDoS protection and easy-to-use dashboards. When evaluating these solutions, it's important to consider their functionality, ease of use and fit with your specific needs (Eyal Katz, 2023).

2.6 THE FUTURE OF WAF OVERVIEW

Then over the years, firewalls have evolved into many different kinds to provide better protections. However, traditional firewalls are not catching up with the evolving threats, and the industry is rapidly adopting the Next-Generation Firewall (NGFW) that offer higher performance, better protection, and simplicity (Liang J and Kim Y, 2022).

Webshell remains a significant challenge in bypassing security measures. Deng et al. (2016) analyze the gap between the latest web attack techniques and existing security protection systems. Their study proposes a novel approach—lexical analysis—to enhance website protection against malicious code injection.

A study by Ines Jemal and Haddar Mohamed amine introduced a Smart Web Application Firewall (SWAF) based on a Convolutional Neural Network (CNN) to detect and stop malicious HTTP requests with high accuracy. The SWAF achieved an attacks detection rate Two-thirds 98% and can catch and stop a malicious request in 2.3ms. Advanced deep learning techniques like CNN are crucial in enhancing security measures against web attacks, and the SWAF is designed to reduce attack detection overhead time, making it an effective solution for web server protection (Ines Jemal et al, 2022).

The future of WAFs must be iterative and continuously upgraded. Tekerek, Gemci, and Bay (2014) investigated and developed a hybrid web application firewall to prevent attacks by utilizing signature-based inspection and anomaly detection methods to analyze HTTP requests. In Development of a Hybrid Web Application Firewall to Prevent Web-Based Attacks, they implemented a system that detects anomalies based on request frequency, request count, and request length. The experimental results highlight a direct correlation between request character length and the likelihood of an anomaly, demonstrating the importance of analyzing request patterns in improving WAF effectiveness.

Free and open-source WAFs are ideal subjects for experimentation at the student level due to cost constraints. Jadhav (2023) explores the use of a free web application firewall to enhance the security functions of web applications in An Analytic Approach to Improve Security Features of Web Application Using Freeware WAF. The study examines WAF and reverse proxy methods, implementing and comparing their effectiveness using different security scanners to assess the level of protection provided to web applications. The experimental results indicate that Shadow Daemon WAF plays a significant role in mitigating various web-based attacks.

2.7 DATA PRIVACY AND LEGAL CONSIDERATIONS

Currently, most of the schemes use digital signature-based methods to achieve data integrity and data completeness. Digital signature, or signature for short, is an algorithm based on symmetric or asymmetric ciphers (public key cryptography). Most signature algorithms are based on asymmetric cryptography. Common digital signature methods

such as RSA, DSA and so on. Signature has the following characteristics: the signature is trustworthy, non-forgable, non-reusable; the signed document is not tamperable, the signature is non-repudiation. Data integrity uses these characteristics of digital signatures to ensure that the query results come from the real original data without any tampering (Tian Xiuxia et al, 2010).

Tian Xiuxia and Wang Xiaoling et al. explore cybersecurity, privacy and the future development trend of the Internet in a study on cybersecurity and privacy. By analyzing the business case of Tencent QQ and Qihoo 360, the authors reveal the strategies used by Internet companies to consolidate resources and expand their market share through cybersecurity issues. It is found that Internet giants' attempts to provide exclusive services have made the Internet architecture more closed, and control has gradually returned to the center from the periphery, creating restrictions on Internet innovation. The article provides an in-depth analysis of the legal issues in Internet business competition, such as privacy and unfair competition, and offers new perspectives for understanding the future development of the Internet (Hu ling, 2012).

Erica Wiking Hager conducted an in-depth study of the field of cybersecurity and privacy and found that countries, while differing in their cybersecurity legislation, place importance on the protection of critical infrastructure and data privacy. Cybersecurity legislation addresses aspects of domestic security, national interests and criminalized cyber activities. In addition, the scrutiny of foreign investment into sensitive areas was becoming increasingly stringent. In his study, he emphasized that cybersecurity and privacy protection were intertwined issues of technology and law, and that governments, enterprises and individuals needed to work together to build a secure and trustworthy cyberenvironment (Erica Wiking Hager and Carolina Dacko, 2017).

2.8 RESEARCH GAPS

Existing frameworks for evaluating Web Application Firewalls (WAFs) present significant challenges. Some are excessively complex, making them impractical for general users without specialized technical knowledge, while others are overly simplistic, focusing solely on detection rates and system performance without

addressing usability and user experience. Furthermore, many evaluation frameworks rely on a single tool or method, lacking a comprehensive and standardized approach. This fragmentation makes it difficult to compare WAF effectiveness accurately and hinders the development of a unified testing standard that balances security, usability, and overall system efficiency.

Another limitation lies in the inconsistency of testing methodologies. Many studies use isolated tools and predefined attack scenarios, failing to replicate the diverse threats encountered in real-world web environments. Additionally, there is no widely accepted benchmark for assessing WAFs, resulting in inconsistent findings across different evaluations. Some assessments focus only on specific types of attacks, such as SQL injection or XSS, while neglecting other critical vulnerabilities. Without a holistic and structured approach, it is difficult to determine the overall resilience of WAFs against modern attack techniques.

The actual performance of WAFs in practical deployment remains another unresolved issue. While many evaluations are conducted under controlled laboratory conditions, these do not always reflect the complexities of real-world applications, where attack patterns and network environments are highly dynamic. Attackers continuously refine evasion techniques, such as encoding manipulation and traffic obfuscation, making it challenging for WAFs to maintain consistent effectiveness. Current research lacks longitudinal studies that assess how well WAFs adapt to evolving security threats over time, leaving a gap in understanding their long-term reliability.

Additionally, usability and accessibility concerns remain largely overlooked in WAF research. Many open-source solutions require intricate manual configuration, creating barriers for organizations or individuals without extensive cybersecurity expertise. While some commercial WAFs offer user-friendly interfaces, they may still demand significant tuning to function optimally. A comprehensive evaluation framework should not only assess security effectiveness but also consider operational practicality, ensuring that WAF solutions are both robust and accessible to a broader range of users.

2.9 CHAPTER SUMMARY

This chapter covered the basic principles, bypass methods, assessment methods, and the dynamic changes in regard to Web Application Firewalls (WAFs). It was composed of the definition of WAFs and their main function of preventing web applications from vulnerabilities such as SQL injection, cross-site scripting (XSS), and file upload attacks.

The discussion explored present provisions of bypass pathways, enlightening elaborate methods hackers use to get around WAF defenses. It also looked at recent appraisal frameworks, where the emphasis was on the standards for WAF effectiveness and the identification of shortcomings in practical usage, automation integration, ML-based categorization, and usability research.

A review of relevant literature indicated that constant improvement of WAFs is necessary in order to keep pace with new threats. This chapter serves as the background for future research by identifying major challenges and providing a structure for assessing open-source WAFs, not only adding to theoretical knowledge, but enabling the practical enhancement of web application security.

CHAPTER III

METHODOLOGY

3.1 INTRODUCTION

This section describes the approach taken for the evaluation of the performance and effectiveness of open-source Web Application Firewalls (WAFs) by the well-defined framework design. Through the WAF framework construction, scope determination, methodology use, weight design allocation, and finally the specification and formulation of standards, the open-source network application firewall is evaluated in all aspects through the framework system.

The research includes four main steps: data survey, experimental setup, data collection, and analysis, as well as evaluation with actionable insights. The data survey is the central core that scrutinizes the WAF types, capacities, weaknesses, and illicit bypass mechanisms.

Data collection involves machines and human beings who use various tricks to collect data for attack detection, system performance, and user feedback. Later, they employ statistical modeling and visualization techniques to catch developments and find valuable information. Python is the main service which is responsible for an automated data analysis, with manual adjustments to ensure the contextual accuracy.

The research is the premise for DFPUE method (Detection Rate, False Positive Rate, Performance Metrics, Usability, and Experience) and introduces weighted scoring as the way of dealing with WAFs. This framework is in line with security standards such as ISO/IEC 27001 and OWASP guidelines, securing the objective and actionable evaluations.

One ought to ponder over conducting experiments in a virtualized environment, which is free from the touch of a third party and keeping a private and secure working model. The methodology aims to overcome some of the limitations of modeling simulation experiments and deed to some principles for practical improvements of WAF solutions to be applied in real-world contexts.

3.2 RESEARCH DESIGN

This study is my final project and follows a mixed-method design, combining quantitative and qualitative approaches to evaluate the performance indicators of open-source web application firewalls (WAFs). To assess the effectiveness of WAFs, key indicators such as interception rate, false positive rate, performance metrics, usability, and user experience will be the focus of the study. The research will include several components, such as a literature review (Chapter 2), data collection and analysis, and result evaluation, followed by feedback dissemination and recommendations. Figure 3.1 represents the complete research process well.

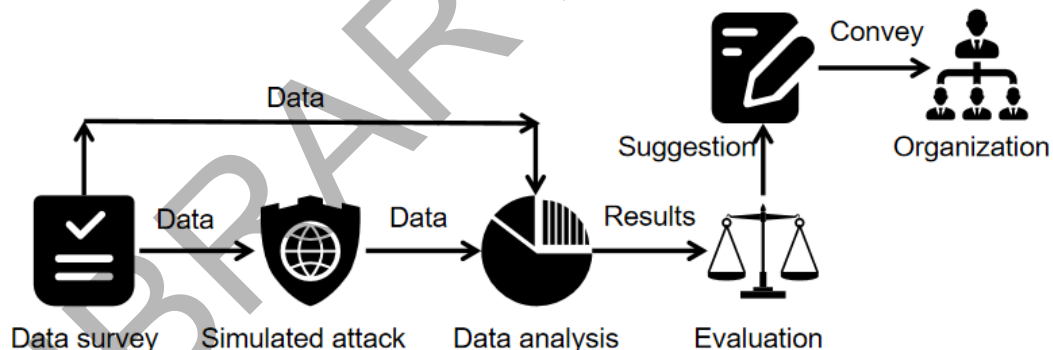


Figure 3.1 Process About This Study

3.2.1 Data Survey - Step 1

As the first stage of this experiment, data investigation mainly summarizes and counts the current types of attacks on web application firewalls and websites, as well as the advantages and disadvantages of the current web application firewall evaluation framework through literature, white papers, technical reports, and other related materials (as shown in the chapter), so that we can recognize the advantages and