Md. Tazwar Islam (C163072)

Salah Uddin Sakib (C163088)

Md. Tazwar Islam (C163072)

Salah Uddin Sakib (C163088)

Thesis Submitted in Fulfilment FOR the degree of  
B. Sc. in COMPUTER SCIENCE AND ENGINEERING

Department of Computer Science and Engineering (CSE)   
INTERNATIONAL ISLAMIC UNIVERSITY CHITTAGONG (IIUC)  
Chittagong, Bangladesh

DECLARATION

We hereby declare that we have read this project report and in our opinion this project is sufficient in term of scope and quality for the award of the degree of B. Sc. in Computer Science & Engineering

|  |  |
| --- | --- |
| 11 June 2021 | Md. tazwar islam  ID: C163072  salah uddin sakib  ID: C163088 |

SUPERVISOR’S DECLARATION

We hereby state that we have reviewed this project report and believe that it is sufficient in scope and quality for the B. Sc. in Computer Science & Engineering degree award.

|  |  |
| --- | --- |
| 11 June 2021 | Dr. shahidul islam khan  Associate Professor, Dept. CSE, IIUC |

DECLARATION OF THESIS / PROJECT REPORT AND COPYRIGHT

THESIS / PROJECT REPORT TITLE:

AUTHORS:

|  |  |  |  |
| --- | --- | --- | --- |
| SL NO. | AUTHOR’S NAME | STUDENT ID | SIGNATURE |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| Name OF SUPERVISOR:  SIGNATURE OF SUPERVISOR: | | | |

I/we declare

1. My/ Our thesis to be published as online open access (full text) at IIUC database or archive.

2. Dept. of CSE, IIUC reserves the right as follows:

i. The thesis is the property of Dept. of CSE, IIUC

ii.The Library of IIUC has the right to make copies for the purpose of research only.

iii. The Library has the right to make copies of the thesis for academic exchange.

ACKNOWLEDGEMENT

First of all, thanks to Almighty Allah for all of his blessings, for providing us patience and good health during this thesis study.

We are very fortunate to have Professor Dr. Shahidul Islam Khan as a research supervisor. Under whose guidance we had the privilege to complete this thesis. We wish to express my deep gratitude towards him for providing me proper guidance and support throughout the Moreover, I am grateful to the faculties and staffs of the Dept. of CSE, IIUC, I would like to thank all.

MD. TAZWAR ISLAM

SALAH UDDIN SAKIB

ABSTRACT

With the elevation of new technologies and innovations, almost all of us in society is surrounded by services and devices with fast internet connectivity. It provides smart lifestyle, the convenience of services and smooth communications to us. At the same time, Web applications help us to recognize the internet's true potential. Anyways, it opens many security defects if we cannot take proper security actions. The main internet-based support for the mechanism of front-end protection is the Web application firewalls (WAF), that is continuously under attack. This project reveals an intelligent Web Application Firewall that detects Unknown, Zero-Day Attacks and Exploits using smart pattern recognition.

TABLE OF CONTENTS

|  | Page |
| --- | --- |

[DECLARATION ii](#_Toc74296333)

[SUPERVISOR’S DECLARATION iii](#_Toc74296334)

[DECLARATION OF THESIS / PROJECT REPORT AND COPYRIGHT iv](#_Toc74296335)

[ACKNOWLEDGEMENT v](#_Toc74296336)

[ABSTRACT vi](#_Toc74296337)

[TABLE OF CONTENTS vii](#_Toc74296338)

[LIST OF TABLES x](#_Toc74296339)

[LIST OF ILLUSTRATIONS xi](#_Toc74296340)

[LIST OF ABBREVIATIONS xiii](#_Toc74296341)

[CHAPTER I Introduction](#_Toc74296342)

[1.1 Research Background 14](#_Toc74296343)

[1.2 Problem Statement 15](#_Toc74296344)

[1.3 Motivation 16](#_Toc74296345)

[1.4 Objective of Research 16](#_Toc74296346)

[1.5 Organization of the Thesis 16](#_Toc74296347)

[CHAPTER II Literature Review](#_Toc74296348)

[1.1 Introduction 18](#_Toc74296349)

[1.2 Scope of Research 18](#_Toc74296350)

[1.3 Similar Existing works 20](#_Toc74296351)

[1.3.1 General 22](#_Toc74296352)

[1.4 Summary 22](#_Toc74296353)

[CHAPTER III Methodology](#_Toc74296354)

[1.1 Introduction 23](#_Toc74296355)

[1.2 Process Model 23](#_Toc74296356)

[1.3 RECENT TRENDS IN SOFTWARE PROCESS MODEL 23](#_Toc74296357)

[1.4 SUITABLE PROCESS MODEL FOR "INTELLIGENT WAF" 24](#_Toc74296358)

[1.5 Reasons behind choosing Agile model: 28](#_Toc74296359)

[1.6 REQUIREMENT SPECIFICATION 28](#_Toc74296360)

[1.6.1 Why Requirement Specification is needed? 29](#_Toc74296361)

[1.7 REQUIREMENT VALIDATION 29](#_Toc74296362)

[1.8 REQUIREMENT VALIDATION IN SOFTWARE PROCESS 29](#_Toc74296363)

[1.9 Application requirement 30](#_Toc74296364)

[1.9.1 Language: 30](#_Toc74296365)

[1.9.2 Database: 30](#_Toc74296366)

[1.10 Feasibility studies 30](#_Toc74296367)

[1.11 FEASIBILITY STUDY OF "INTELLIGENT WAF" 31](#_Toc74296368)

[1.12 Testing 32](#_Toc74296369)

[1.12.1 Unit Testing: 32](#_Toc74296370)

[1.12.2 Testing functionality 34](#_Toc74296371)

[1.12.3 Testing Black Box 35](#_Toc74296372)

[1.13 Summary 37](#_Toc74296373)

[CHAPTER IV System Analysis, Design and Implementation](#_Toc74296374)

[1.1 Introduction 38](#_Toc74296375)

[1.2 ProposedMethodology 38](#_Toc74296376)

[1.3 System Overview Of "Intelligent WAF" 38](#_Toc74296377)

[1.4 Security 39](#_Toc74296378)

[1.5 Version Support 40](#_Toc74296379)

[1.6 System Design 40](#_Toc74296380)

[1.7 Flow Chart 40](#_Toc74296381)

[1.7.1 Security Module Flowchart: 41](#_Toc74296382)

[1.7.2 Admin Module Flowchart: 42](#_Toc74296383)

[1.8 Use Case Diagram 43](#_Toc74296384)

[1.8.1 Use Case Diagram for Security Module 43](#_Toc74296385)

[1.8.2 Use Case Diagram for Admin Module 43](#_Toc74296386)

[1.9 Database Table 44](#_Toc74296387)

[1.10 Entity Relationship Diagram 52](#_Toc74296388)

[1.10.1 ERD for Security Module 53](#_Toc74296389)

[1.11 Data Flow Diagram 53](#_Toc74296390)

[1.11.1 Data Flow Diagram for Security Module: 54](#_Toc74296391)

[1.11.2 Data Flow Diagram for Admin Module: 55](#_Toc74296392)

[1.12 Design Model 59](#_Toc74296393)

[1.13 Graphical Representation 59](#_Toc74296394)

[1.14 Summary 67](#_Toc74296395)

[CHAPTER V Results and Discussion](#_Toc74296396)

[1.1 Introduction 68](#_Toc74296397)

[1.2 Results 68](#_Toc74296398)

[1.2.1 General 68](#_Toc74296399)

[1.3 Discussions 68](#_Toc74296400)

[CHAPTER VI Conclusion and Future Works](#_Toc74296401)

[1.1 Conclusion 69](#_Toc74296402)

[1.2 Contribution of this thesis 69](#_Toc74296403)

[1.3 Future works 69](#_Toc74296404)

[1.3.1 Limitations: 70](#_Toc74296405)

[1.3.2 Future Works of the project: 70](#_Toc74296406)

[References 71](#_Toc74296407)

[Appendices](#_Toc74296408)

[Appendix A CREDENTIALS OF THE DEMO APP 73](#_Toc74296409)

[Appendix B SECURITY MODULE: CLEAN INPUT ALGORITHM 74](#_Toc74296410)

[Appendix C SECURITY MODULE: DATA SANITIZATION ALGORITHM 75](#_Toc74296411)

[Appendix D PATTERNS FOR DETECTING MALICIOUS REQUEST 76](#_Toc74296412)

LIST OF TABLES

|  |  |
| --- | --- |
| Table No. | Page |

[Table 1: Review of most popular existing similar works 20](#_Toc74296413)

[Table 2: Unit Testing 33](#_Toc74296414)

[Table 3: Functional Testing 34](#_Toc74296415)

[Table 4: Black Box Testing 35](#_Toc74296416)

LIST OF ILLUSTRATIONS

|  |  |
| --- | --- |
| Figure No. | Page |

[Figure 1: Basic structure of WAF 18](file:///F:\Importants!!\Project\Defense\155_report.docx#_Toc74296417)

[Figure 2: Agile Approach 24](#_Toc74296418)

[Figure 3: Basic development structure of SCRUM methodology 25](#_Toc74296419)

[Figure 4: SCRUM events 26](#_Toc74296420)

[Figure 5: Three roles of SCRUM methodology 27](#_Toc74296421)

[Figure 6 - WAF CONCEPT 39](#_Toc74296422)

[Figure 7: Security module flowchart 41](#_Toc74296423)

[Figure 8: Admin Module Flowchart 42](#_Toc74296424)

[Figure 9: Use Case Diagram for Security Module 43](#_Toc74296425)

[Figure 10: Use Case Diagram for Admin Module 43](#_Toc74296426)

[Figure 11: Database Table for users 44](#_Toc74296427)

[Figure 12: Database table for logins 45](#_Toc74296428)

[Figure 13: Database table for logs 45](#_Toc74296429)

[Figure 14: Database table for file-whitelist 46](#_Toc74296430)

[Figure 15: Database table for badbot-setting 46](#_Toc74296431)

[Figure 16: Database table for bad-words 47](#_Toc74296432)

[Figure 17: Database table for ip-whitelist 47](#_Toc74296433)

[Figure 18: Database table for bans-ranges 47](#_Toc74296434)

[Figure 19: Database table for bans-other 48](#_Toc74296435)

[Figure 20: Database table for dnsbl-databases 48](#_Toc74296436)

[Figure 21: Database table for proxy settings 49](#_Toc74296437)

[Figure 22: Database table for bans 50](#_Toc74296438)

[Figure 23: Database table for spam-setting 50](#_Toc74296439)

[Figure 24: Database table for adblocker-settings 51](#_Toc74296440)

[Figure 25:Database table for pages-layout 51](#_Toc74296441)

[Figure 26: Database table for bans-country 51](#_Toc74296442)

[Figure 27: Database table for live-traffic 52](#_Toc74296443)

[Figure 28: ERD for Security Module 53](#_Toc74296444)

[Figure 29: 0 Level Data Flow Diagram for Security Module 54](#_Toc74296445)

[Figure 30: 1st Level Data Flow Diagram for Security Module 54](#_Toc74296446)

[Figure 31: 2nd Level Data Flow Diagram for Security Module 55](#_Toc74296447)

[Figure 32: 0 level Data Flow Diagram for Admin Module 56](#_Toc74296448)

[Figure 33: 1st level Data Flow Diagram for Admin Module 56](#_Toc74296449)

[Figure 34: 2nd level Data Flow Diagram for Admin Module 57](#_Toc74296450)

[Figure 35: 3rd level Data Flow Diagram for Admin Module 58](#_Toc74296451)

[Figure 36: Design model 59](#_Toc74296452)

[Figure 37: Malicious Request By Attacker Was Blocked 59](#_Toc74296453)

[Figure 38: An Attacker Was Banned 60](#_Toc74296454)

[Figure 39: A spammer Request was Blocked 61](#_Toc74296455)

[Figure 40: Administrator Authentication 61](#_Toc74296456)

[Figure 41: Admin Module Dashboard 62](#_Toc74296457)

[Figure 42: SQL Injection Protection Module 63](#_Toc74296458)

[Figure 43: Bad Bots - Protection Module 64](#_Toc74296459)

[Figure 44:Proxy - Protection Module 65](#_Toc74296460)

[Figure 45: Spam - Protection Module 66](#_Toc74296461)

[Figure 46: Attack Logs 67](#_Toc74296462)

|  |  |
| --- | --- |
| Photo No. |  |

LIST OF ABBREVIATIONS

WAF Web Application Firewall

IIUC International Islamic University Chittagong

NSS Network Security Services

PWC Price Waterhouse and Coopers & Lybrand

ROI Return on investment

OWASP Open Web Application Security Project

* 1. Introduction
     1. Research Background

In the late 1990s, web application's appearance was a numerous breakthrough when it began to recognize the internet's true potential. They are now a staple for companies of all sizes. The number of websites online, right now, is monstrous 1.75 billion. In this era, they have become the most indispensable part of our lives.

Sadly, for the same purpose, web applications became the main security threat to them [1] A new study reveals that [2] 72 of 100 businesses compromised their websites/applications at least once in 24 months their websites/applications. Most hackers target the application layer (Layer 7 of the OSI model), as the application layer is the weak region. While the lower levels are not stable, network protocols have been hardened to the point that hacking is complicated, and there are few vulnerabilities [3]. The operating systems, web servers, and encryption engines are comparatively stable compared to websites/applications. Since they usually come from only a few manufacturers, everybody has a good user base to find bugs early and patch them fast. However, the case with a Web application is the reverse. They typically are exclusive (although they use common code portions - e.g., a particular framework) and usually only held by or with a specific client. This results in lower average code protection (as specialist expertise is less accessible in such a fragmented sector for each company) and fewer chances of identifying weaknesses before being exploited. That is why web server bugs are more vulnerable than browser/operating systems. The disparity is shown by the 1:10 ratio [4]. Some common attacks include cross-site scripting (XSS)[5], SQL injection, and remote execution of commands. These are the most challenging problems for creators of mobile apps and Internet protection staff [6].

A Web Application Firewall (WAF) is a protection module in an application proxy mechanism that shields the web application server from various attacks in the back end [7]. Security of applications is a worthy safety layer. This is how it will guard against many security vulnerabilities to the application layer. Usually, a traditional network layer intrusion prevention device would not cover this layer. The WAF guarantees that the security of the database server is not affected by resembling the HTTP/HTTPS request packets and the Web traffic patterns. The WAF blocks the attack in any defense, in the setup or the interrupt monitoring method, by blocking an HTTP request or user session, or IP address. The WAF blocks the attack. Logging is a crucial component of every web2 framework. Often it becomes too important to record stuff because they allow certain malicious users to spot an imperfection or behavior later on. Most log processing of the program currently is not performed intelligently. It is just inserted into the logs and managed manually afterward. However, as more inquiries are made, and worthwhile pieces of knowledge may be retrieved from these requests, there is 100% waste. As more attacks are observed, better protection needs to be given to the backend server [8]. The Network IDS attempts to track such attacks by reviewing data and identifying unusual trends. In general, algorithms used in IDS are used to identify intrusions via different techniques. For example, signature-based approaches use hard-coded algorithms from some experts to detect established attacks. Courses in data mining are also used for training. These algorithms more often lag when it comes to the detection of new and unfamiliar types of intrusions. You either need to send a new algorithm or train the machine on a new dataset to fight it.

* + 1. Problem Statement

Web applications are a mixture of client-side scripts to display information to users and server-side scripts to manage the information's storage and retrieval [9]. They require an application server to perform the tasks requested, a webserver to handle requests from the client, and a database to store the information. In 2002, according to the Computer Security Institute and the Federal Bureau of Investigation (United States) survey, 60% of online databases were subject to security breaches every year. Beyond the sessions of 2015, it was determined that over 700 million data records had been negotiated. They match this with the case that the predicted cost of cybercrime is expected to rise to over $2 trillion over 2019 and values around $400 billion yearly now. The freshest records establish that cybercrime is expanding in frequency and cruelty [10], with a prediction to touch $6 trillion by 2021 (up from $3 trillion in 2015) [11] and also take on common crime in number and cost [12]. Around 35,000 to 50,000 websites get hacked every single day. Unfortunately, the numbers are growing fast, and website security's weight is increasing rapidly. No one on the web is protected security risks.

* + 1. Motivation

The increase in safety of the web app is significant, given that it covers all things related to the guarding of confidential data, PII, personal3, intellectual capital, secure health data (PHI), hacker information systems, and the loss of criminals and rivals by national and manufacturers [8]. Cybersecurity risks are growing, propelled by the use of global networking and cloud computing to store confidential data and personal details, such as Amazon Web Services. The pervasive misconfiguration of cloud resources together with increasingly advanced cybercriminals ensures that the possibility of an effective cyber assault or data theft for your company is on the increase [13].

* + 1. Objective of Research

This research concentrates on the development of new techniques….

1. To detects Unknown, Zero-Day Attacks and Exploits
2. To provide more secure and accurate protection to web applications.
   * 1. Organization of the Thesis

The project contains five chapters. These are ordinarily defined as follows:

The first chapter reveals the inception of the project. The issue and study inspiration are mentioned in this chapter. The study goals of the study are therefore illustrated. The following chapters would go into the different facets of this project in depth.

The second chapter analyses related works done by others, focusing on contrasting the functionality and limitations.

Chapter 3 reflects on the methods for system construction and process flow in-depth of this project. It reveals the strategies and the algorithms that will be used in performing the analysis.

Chapter 4 outlines the study, specification, and implementation of the framework in depth.

Chapter 5 concentrates on the specifics of the findings and the discussion of our analysis. The last chapter ends the overall report to explore our study's contribution and potential work.

* 1. Literature Review
     1. Introduction

This chapter casually reviews and discusses existing similar works related to the current project, “An intelligent web application firewall framework to prevent security attacks,” which will be developed. This chapter comprises three sections. The first section describes the scope of the project that contains details about WAF. The second section contains a table comparing three of the most popular WAF available right now. It describes the description, features, and limitations of every specific app. The third section discusses the gap achieved from the review.

* + 1. Scope of Research

WAF protects web applications from malicious traffic movement to the web application by tracking, filtering, and stopping illegal data from leaving the app. It adheres to policies that help decide which traffic is malicious and safe [14].

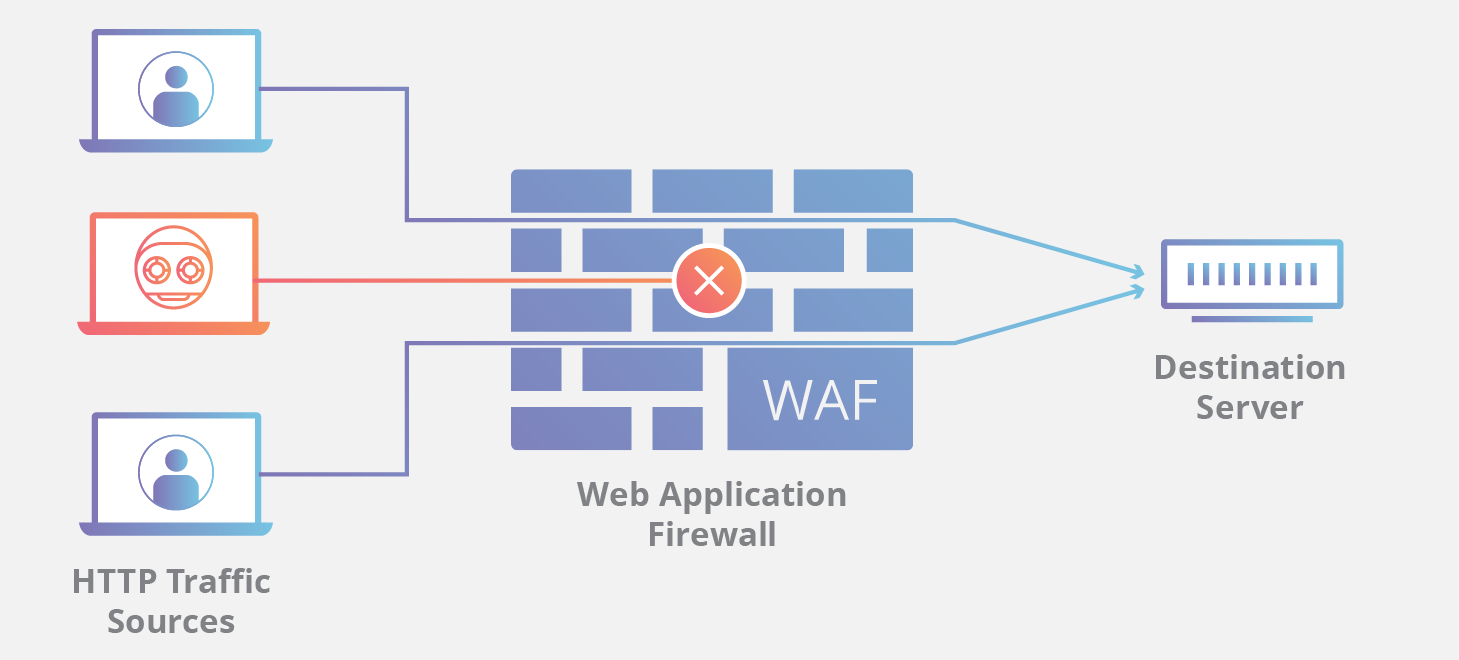


Figure : Basic structure of WAF

WAF has the following methods to block or minimize threats:

Cross-site (XSS): It is one of the most popular assault vectors. This is when an intruder injects malicious code to manipulate session cookies in the client's browser. And then, confidential data is thieved, or even material is rewritten to reveal false facts. WAFs may be designed to apply Content Security Policy to deter XSS assaults.

Misconfigured servers: unsafe configurations - for example, the main intruder priorities are default credentials and guest accounts. These vulnerabilities are created primarily when managers fail to implement best practices in protection. WAFs will enforce protection guidelines and reject unsafe protocols, thus mitigating the effect of malfunctioning devices.

SQL injection: Websites with insufficient input validation are opened in code injection vulnerabilities. In order to obtain unauthorized entry to accounts, attackers attempt to sneak into SQL claims. WAFs will identify and prevent certain attempts.

Known bugs are still open to obsolete applications and databases.

WAFs will block established vulnerabilities before patching is carried out, thereby acting as a stop-gap remedy.

Insufficient logging and surveillance: Comprehensive traffic logging can result in early malicious activity detection. A WAF serves as a central logging point and alerts managers of ongoing threats [15]

Access control and sensitive data exposure: attackers may try to steal sensitive data by scanning a website structure and exploiting unsecured resources [16]. WAFs can be used to lock parts of the website and to only allow trusted parties access.

* + 1. Similar Existing works

There are already many works on WAF that exist. Some of the most popular WAF has been reviewed below:

Table 1: Review of most popular existing similar works

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SL No. | Name | Descriptions | Features | Limitations |
| 1 | Fortinet FortiWeb | FortiWeb is a web application firewall (WAF) that protects hosted web-based applications from exploit-based assaults. It integrates into the Fortinet Security Fabric and shares bidirectional threat intelligence with FortiGate enterprise firewalls, FortiSandbox sandboxing solutions, with automation of security workflows and processes. | Security: Quite excellent. NSS Labs rated it on track with three other leaders in security efficiency, with a 98 per cent blocking rate.  Implementation: Awesome. Its fastest advertised protected WAF throughput is 20 Gbps, though models start at 25 Mbps. | Alliance with other cloud-based DDoS protection assistance such as Cloudflare, Akamai, Arbor, etc.  Doesn't provide protection for Brute Force. |
| 2 | CITRIX NETSCALER | Citrix NetScaler AppFirewall is a WAF that monitors all bi-directional traffic in the fight against protection advisories, including SSL encrypted correspondence. | Security: Very well. NSS Labs also achieved the maximum level of safety efficiency and the second-highest block rate at 99.07%.  Performance: best performance in class. With 149,000 contacts per second and 184,400 transactions per second, NSS Labs ranked the best. | There are restrictions on predictive policy learning, dashboard monitoring, and the capacity to prevent false warnings (false positive rate). |
| 3 | Symantec WAF | Symantec Web Application Firewall is built on the ProxySG principles. Clients can deploy it on-premises or in the cloud with AWS to prevent familiar attack patterns with signature-based engines. | Security: Fair. NSS Labs scored its block rate of only 91.07% and security effectiveness of 92.45%.  Performance: Fair. NSS Labs tested Blue Coat at only 1,905 connections and 1,600 transactions per second. | It takes 4-5 hours to install.  Doesn't share attacking logs with user. |

* + - 1. General

Assuredly, WAFs are not an absolute innovation. For several years they have been available on the market, and there exist many vendors. It is clear from the Comparison of the three popular WAF that none of them is perfectly perfect.

No WAF is dealing with all OWASP Top Ten Web Application vulnerabilities efficiently. Most WAFs cannot Detect Unknown and Zero-Day Attacks, Exploits and do not share attack logs.

* + 1. Summary

Exploring the exact gap is the most critical and time-consuming procedure of the research. We have discovered the way by reviewing the existing similar works in this chapter.

* 1. Methodology
     1. Introduction

This chapter aims to address the project procedure and structure. The technique, methodology, or approach would be utilized throughout the design and implementation of the project. This chapter further explains the process or solution used and the specifications for hardware and software.

* + 1. Process Model

The procedure is a movement, which works on an article and changes its state. Model is the graphical portrayal of an article. In this way, a procedure model shows the exercises of programming graphically. We need a procedure model since the procedure is significant than the item (software). On the off chance that the procedure is excellent, the item likewise is excellent.

* + 1. RECENT TRENDS IN SOFTWARE PROCESS MODEL

Current trends in software process model has been given below:

1. Waterfall model

2. Spiral Model

3. Incremental Model

4. Prototyping

a) Rapid throw away prototyping process model.

b) Evolutionary prototyping process model.

c) Extreme prototyping process model.

d) Incremental prototyping process model.

* + 1. SUITABLE PROCESS MODEL FOR "INTELLIGENT WAF"

To develop this application, we have followed SCRUM methodology of Agile approach.

Agile Approach: The Agile Approach is based on iterative and incremental progress instead of a straightforward approach. It does not formulate a whole system at once but alternatively develops incrementally. More insufficient time is spent upfront for documentation and critique, as consumers are continually seeing and testing the output and providing feedback. The development and feedback process attaches accountability and advances client satisfaction by allowing open-ended input.

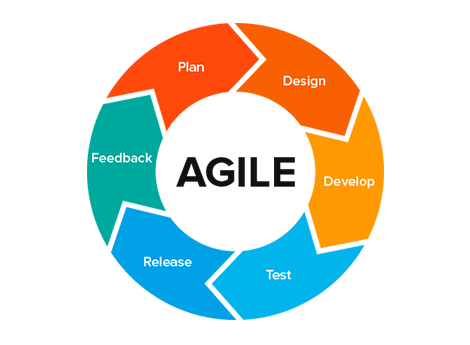
Agile depends on a very high level of consumer engagement completely every phase of the project. The planning, design, development, testing, release, and feedback are constantly in a fixed period. Somewhat of segmenting schemes in stages, agile development leads to address the projects as a whole.

Figure 2: Agile Approach

This approach is more flexible than others. The specifications may change by the way, and the team must modify quickly (the teams are usually more miniature). There is more comprehensive transparency between the consumer and developers, and the schedule and cost are anticipated.

SCRUM methodology:

Scrum methodology is one of the most popular agile approaches for developing, delivering, and transferring complicated products where requirement frequently changes. Scrum's primary goal is to meet the customer's requirements in an atmosphere of communication openness, mutual accountability, and constant change. The creation begins with a broad understanding of what needs to be developed, producing a set of priority characteristics (product backlog) that the product owner wishes to achieve.

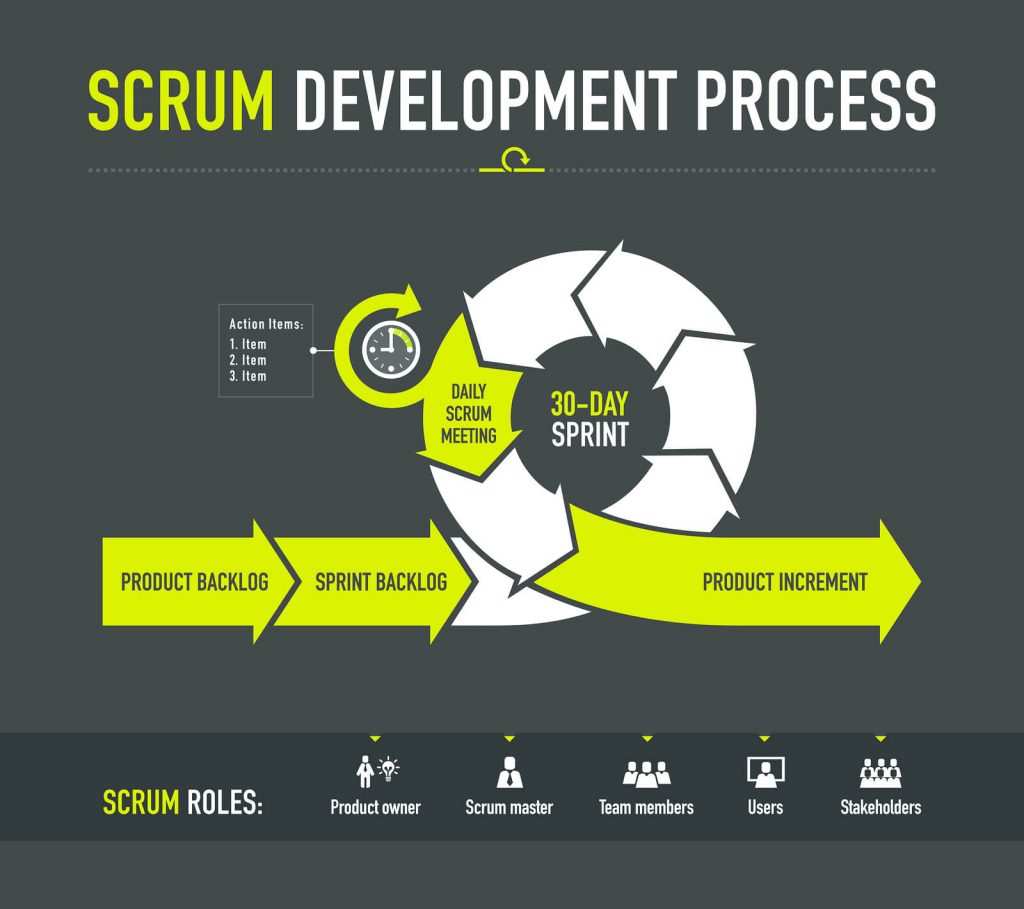


Figure 3: Basic development structure of SCRUM methodology

1. Product backlog: Product backlog is a trendy product management concept for Agile systems and the technique of Scrum.

It contains several concepts, elements, and suggestions for creating collected and assembled into a collection.

2. Sprint Backlog:  The Sprint Backlog is an example of the whole Product Backlog Chart. All the thoughts, demands, and assignments in the context of an extensive list are referred to as items. Product Backlog Items are the product backlog material. They are named objects because they can be somewhat different in condition.

3. Product Increment: It is the latest iteration of the product being created. This latest release includes all the work completed previously, including the job done in the new Sprint.

SCRUM activities:  Events of Scrum are everyday events. Any conditions occur during each case. The official events of Scrum are as follows:

• Sprint

• Sprint Planning

• Daily SCRUM

• Sprint Review

• Sprint Retrospective

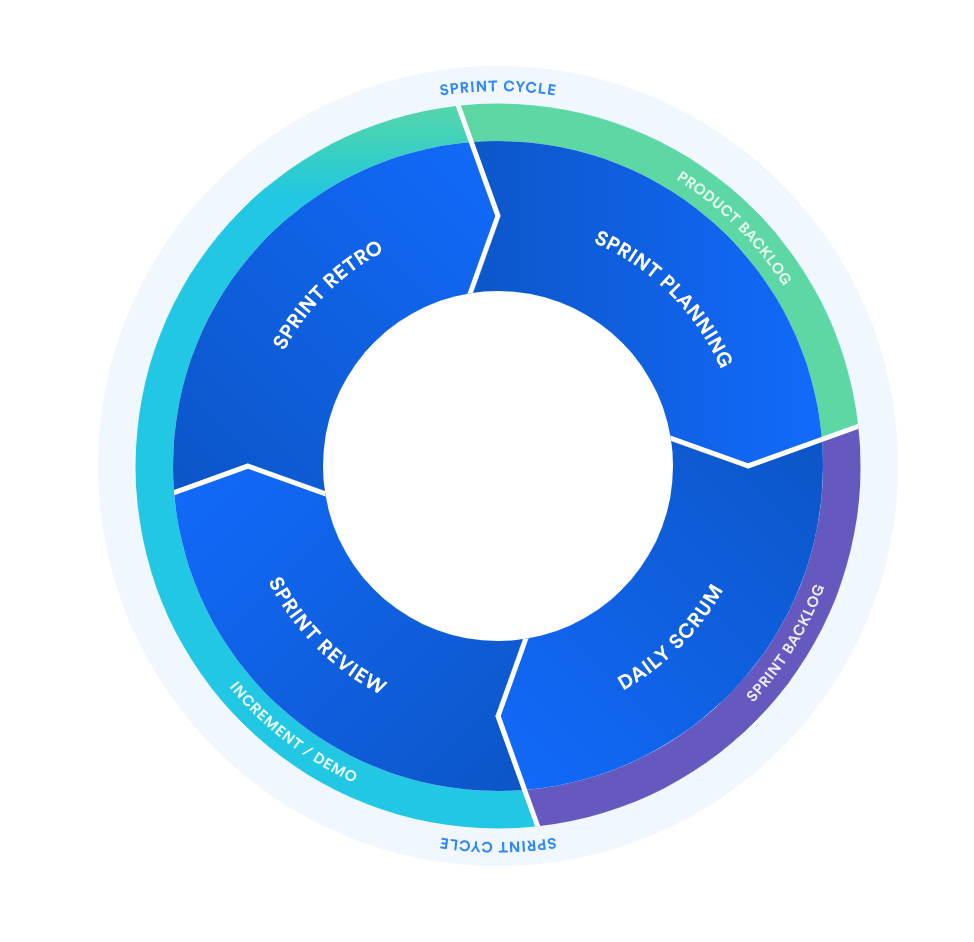


Figure 4: SCRUM events

SCUM technique has three functions. The following functions are:

1. Owner of the product

2. Master Scrum

3. Team Scrum.



Figure 5: Three roles of SCRUM methodology

1. Product Owner: The role of the product owner is crucial and can be performed only by anyone who knows the expectations of end-users thoroughly. They might be anyone from the publicity or product testing department, so they can tell us what users from the project team want to see. The owner of a commodity often has to understand potential niche patterns and what steps or features rivals are taking to win more ground in the business. However, it usually depends on the form of the product is made.

2. Master Role Scrum: The Scrum Master ensures that each part of the squad knows Scrum and its functions. He or she serves as a trainer and mentor and verifies that some squad participants stick to the Scrum principle and practice, leads by example, wear diligence, and pays attention to every part of the project. The Scrum Master works with the product owner to streamline the product backlog and to improve strategies. It encourages the team to shine by ensuring that the sprints are not over-engaged - the primary aim behind routine sprints is to present better variations, given that sprints do not extend four weeks. The purpose of the sprint is to have quality outcomes in a limited period. However, suppose the team is overcommitted to a given period for some purpose. In that case, it will trigger uncertainty and worry, preventing the development of a project rather than speeding up quality outcomes.

3. Scrum Development Team Role: The core work of the Scrum Framework is performed by a devoted Scrum team; the community operates collectively to produce and distribute the product. Ideally, it is a small cross-functional team of about six members (plus, minus three people), comprising consultants for businesses, software testers, and engineers. During regular stand-up sessions, members of the Scrum team must track their daily growth, victories, and obstacles. In the first sprint, no new Scrum team delivers a 100 percent product increase; it usually requires 2-3 weeks to achieve the best success of a new Scrum team.

* + 1. Reasons behind choosing Agile model:

Agile is ideal for applications where the requirements change frequently. Depending on the situation, our pattern matching techniques for the security model may change constantly. Furthermore, it outperforms other existing models in terms of speed. It provides developers implementation flexibility by allowing them to build and test at the same time.

* + 1. REQUIREMENT SPECIFICATION

A specification can be thought of as a contract between users and software developers that specifies the desired (functional/service) properties of the software artefacts and other properties of its achievement, reliability, and so on, without revealing how much functionality will be performed.

* + - 1. Why Requirement Specification is needed?

Requirement Specifications necessitated because:

1. It can be seen as a foundation for contracts between the system developer and customers.

2. It is a more detailed description of the system's functionality and the constraints on its development.

* + 1. REQUIREMENT VALIDATION

Requirement validation is involved with assuring that the requirements determine the system, which the consumer demands. It has much in common with review as it is concerned with discovering problems with the situations. However, they are different processes since validation should be concerned with a complete draft of the requirements document, whereas analysis includes working with incomplete requirements.

* + 1. REQUIREMENT VALIDATION IN SOFTWARE PROCESS

Various sorts of checks must be performed during the software's processing phase. These checks include the following:

1. Validity check: A user may assume that a system is required to do particular tasks. However, more study may reveal the need for new or alternative roles.For example, in the “Intelligent WAF,” the system has diverse users with different needs and requirements. Anyways, the software must be arranged so that the services are provided across the user community by maintaining and checking validity.

2. Completeness Check: The required document should contain specifications that describe the device user's designated roles and restrictions.

3. Control of consistency: text requirements do not clash. For instance, the "Intelligent WAF" does not have inconsistent restrictions or distinct representations of the same machine structure in dealing with unrecorded participants.

4. Realism check: Specifications can be tested using current technical experience to ensure that they can be applied.

* + 1. Application requirement
       1. Language:

1. Backend - PHP

2. Frontend- HTML, JavaScript

* + - 1. Database:

1. MySQL

* + 1. Feasibility studies

Feasibility studies are designed to show pretty and accurately the qualities and weaknesses of existing companies or prospective adventures, openings, and risks, the tools needed to achieve, and at last, to develop possibilities.

Economic feasibility: the monetary benefits of this commodity are investigated. More often known as a cost/benefit analysis, the approach is to assess, contrasting the gains and the reserve capital that is normal in the context of an up-and-coming framework. If the benefits outweigh the expenses, the decision is taken to prepare and implement the system. A business individual should measure the cost and advantages precisely before making a move. Such applications spend our resources.

Feasibility of the operation: The operational feasibility is how well a planned framework responds to the challenges and considers the possibilities recognized during scope definition and how it satisfies the criteria defined in the system implementation specifications review aspect.

Technical Viability: Technical feasibility decides what project work can be accomplished with the available facilities, specialized software, and staff. Technical viability concerns the specification of facilities and applications that meet the user's needs. The machine suggested will operate on any handheld OS, which is a fascinating technology.

Schedule Feasibility: A proposal can take too long to be completed in some manner before it becomes valuable. This usually involves assessing to what degree the system can be created and whether it appears to be finalized in a specific time frame using certain strategies such as payout duration. The probability of calendars is a proportion of how sensitive the schedule is.Our Schedule Feasibility is simply time accommodation of the task. In any case, some of the time political viciousness, terrible climate produced results to present our task due time.

* + 1. FEASIBILITY STUDY OF "INTELLIGENT WAF"

This segment will discuss the possible investigation of the product interfaces plan and framework Functionalities of "Intelligent WAF." This is to recognize whether our website and the idea of the framework meet the potential clients. The attainability study will be surveyed utilizing a few parameters to check the framework prerequisites in various classifications. It is essential to grasp the customer's needs during the arrangement, and sometimes these crucial perspectives are purposely disregarded. The importance of the customer's emotion on an advanced item has also been verified. The main practical characteristics of the research merely emphasize the estimation and provide another element for the operation of an organization. Innovation potential, economic viability, operational practicality, and legal plausibility are the crucial characteristics. Nevertheless, the evaluation of the customers' need for the additional idea is not highlighted here, while customers do not decide whether or not to buy an item. A quality item should be acceptable to use and agree to all possible customer needs.

* + 1. Testing

Software testing is an inquiry administered to present stakeholders with information about the product's quality or service under test. It may also give an impartial, independent assessment of the program to help the company realize and grasp the risks associated with software installation.

Software testing may be confirmed as an authentication and verification process: Software testing may be confirmed as an authentication and verification process:

1. fulfills the design and development requirements of its software program

2. acts as anticipated and

3. may be carried out with the same features.

The testing methodologies are provided below:

* 1. Unit testing
  2. Black box testing
  3. Functional testing
     + 1. Unit Testing:

This is a technique in which individual code modules are tested before incorporating them with other modules. The test unit might be a capacity, a function, a methodology, or a strategy. Units might be generally miniature gatherings of interrelated modules that are executed continuously as a gathering. Unit testing aims to recognize and fix; however, many mistakes would be prudent before modules are consolidated into a bigger programming unit. Mistakes become substantially more troublesome and costly to find and fixed when numerous modules are joined. Unit testing of "Intelligent WAF" are given below:

Table 2: Unit Testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Test Case | Attributes and Values | Expected Result | Result |
| 1 | products?id=678'-- | A GET method malicious request came through HTTP | WAF should detect the malicious request and block the request then log the attacker details | Pass |
| 2 | User = admin'--  Pass = admin'-- | A POST method malicious request came through HTTP | WAF should detect the malicious request and block the request then log the attacker details | Pass |
| 3 | /products | A Blacklisted Spammer IP came through HTTP to client request | WAF should detect the malicious request and block the request then log the attacker details | Pass |
| 4 | /login | A normal user came to login page | WAF should approve this request | Pass |

* + - 1. Testing functionality

Functional testing is classified as a test which ensures that the capability of each product application functions according to the precondition.

This test basically contains discovery testing and does not worry about the application's source code. Functional testing of "Intelligent WAF" is given below:

Table 3: Functional Testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Test Case | Attributes and Values | Expected Result | Result |
| 1 | products?id=678'-- | A GET method malicious request came through HTTP | WAF should detect the malicious request as SQL Injection | Pass |
| 2 | User = admin'--  Pass = admin'-- | A POST method malicious request came through HTTP | WAF should detect the malicious request as SQL Injection | Pass |
| 3 | /products | A Blacklisted Spammer IP came through HTTP to client request | WAF should detect the request as spamming | Pass |
| 4 | /login | A normal user came to login page | WAF should approve this request | Pass |

* + - 1. Testing Black Box

In this form of assessment, constitutional system design is not counted. Requirements and functionality are the basis for tests.

Black box testing of "Intelligent WAF" is given below:

Table 4: Black Box Testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Test Case | Attributes and Values | Expected Result | Result |
| 1 | products?id=678 | A GET method request came through HTTP | WAF should approve the request | Pass |
| 2 | User = admin  Pass = admin | A POST method request came through HTTP | WAF should approve the request | Pass |
| 3 | /products | A Blacklisted Spammer IP came through HTTP to client request | WAF should detect the malicious request and block the request then log the attacker details | Pass |
| 4 | /login | A normal user came to login page | WAF should approve this request | Pass |

* + 1. Summary

In this chapter, the procedure and structure of the project have been explored. Besides method, technique, or approach of the designing and implementing the project has been discussed briefly.

* 1. System Analysis, Design and Implementation
     1. Introduction

This chapter briefly explains the implementation of the project. It reveals the progression of system development methodologies and discusses the functions and skills expected of a systems analyst.

* + 1. ProposedMethodology

Our WAF (Web Application Firewall) can analyze the GET & POST requests that will come through HTTP and HTTPS. If the WAF is appropriately implemented, it will have the power to catch malicious web traffic through HTTP & HTTPS method. To understand more, let us assume - A web-based application is integrated with our WAF. If a malicious request comes to the client application, the WAF will block the request long before it reaches the actual client web application. In other words, every request of the client application will go through our WAF first. The WAF will decide if the request is clean or not. If the request is clean, the WAF will accept the request. Otherwise, it will get blocked.

* + 1. System Overview Of "Intelligent WAF"

System overview of our WAF refers to the whole process of the protection module. Our WAF has two modules, client and admin module. In the client module, every request of the client application is being protected by WAF. And in the admin module an administrator is allowed to manage every rules of the WAF.

The figure of the concepts is given below:

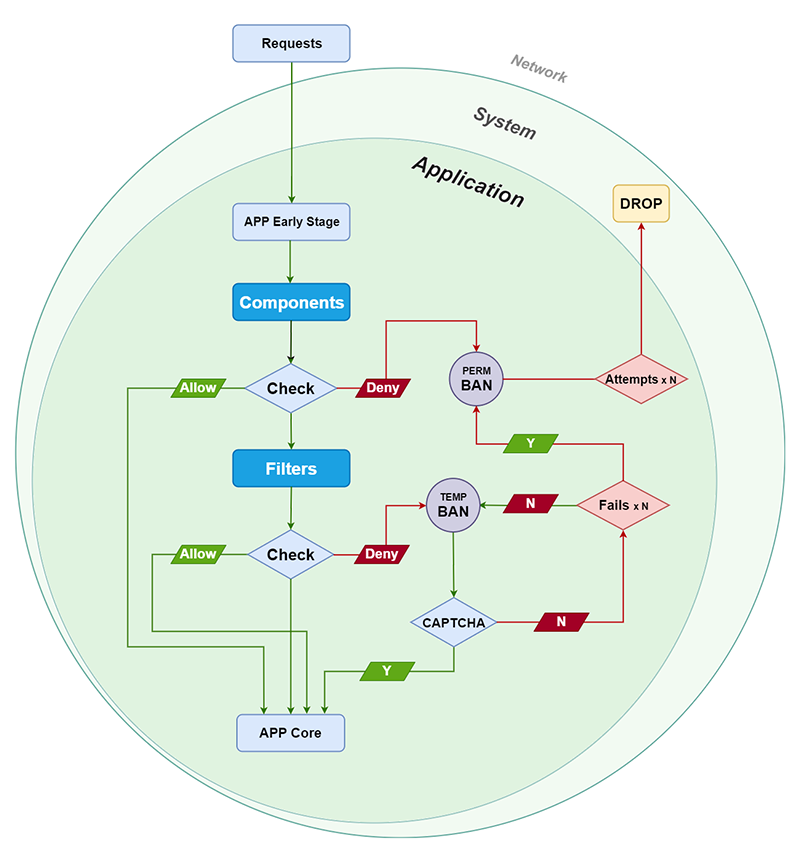


Figure 6 - WAF CONCEPT

* + 1. Security

Intelligent WAF provides an intelligent response to potential threats that can impact client web applications. Our WAF designed to protect client application from potential threats that have yet to be identified, which means that implementing this solution can save an organization from security vulnerabilities, cross-site scripting attacks, SQL injections, zero-day threats and other types of threats.

* + 1. Version Support

The Intelligent WAF build on PHP Version 7.4. This project can be integrated with most of the application build on PHP Version 7.0 +.

* + 1. System Design

# The system design provides an alternate option to meet the research objectives, assess the options, and outline the plans of the selected decision.

# **Objective:** Convert the system specification to a functional system design.

# Input: project objectives, prioritization of user requirements, specification.

# **Process:** Typical transformation with multiple possibilities whose costs and advantages need to be defined and assessed.

# **Output:** Logical system design, planned production, needs for software and hardware.

The system design provides an alternate option to meet the research objectives, assess the options, and outline the plans of the selected decision.

Objective: Convert the system specification to a functional system design.

Input: project objectives, prioritization of user requirements, specification.

Process: Typical transformation with multiple possibilities whose costs and advantages need to be defined and assessed.

Output: Logical system design, planned production, needs for software and hardware.

* + 1. Flow Chart

A flow chart is a symbolic or graphical description of the process. The flow chart figures are linked together with arrows pointing to the process flow direction.

* + - 1. Security Module Flowchart:

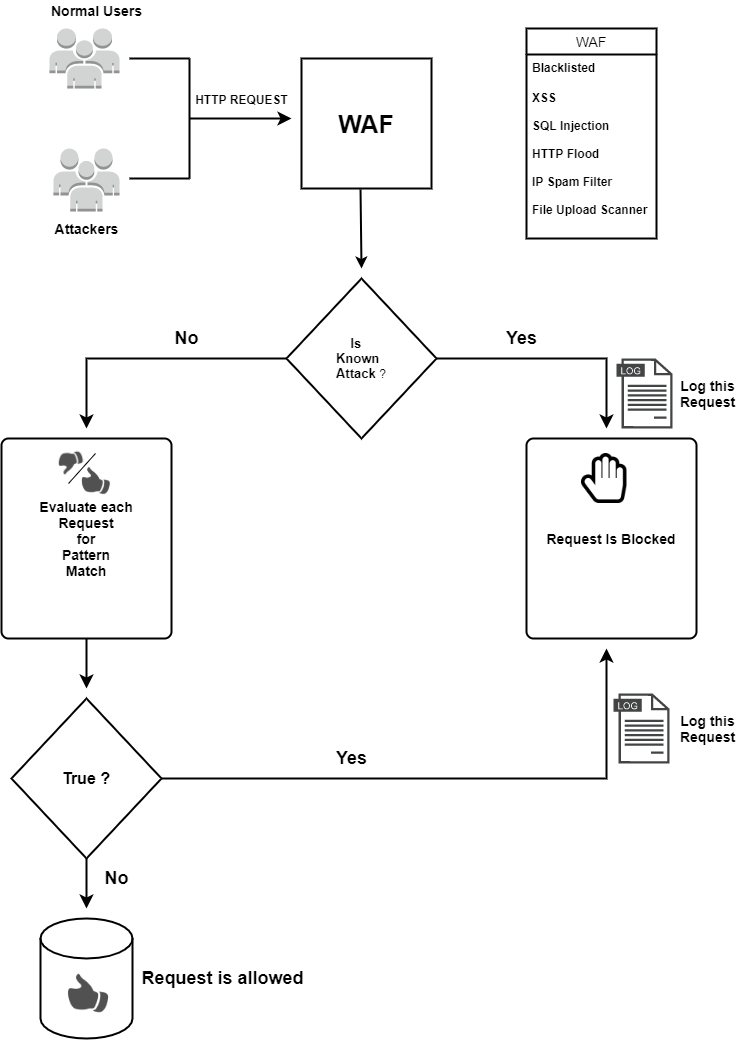


Figure 7: Security module flowchart

* + - 1. Admin Module Flowchart:

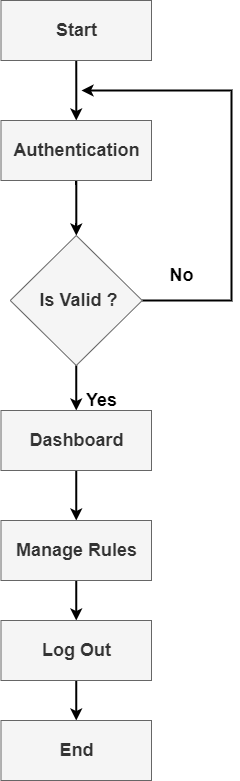


Figure 8: Admin Module Flowchart

* + 1. Use Case Diagram

A use case diagram is the diagrammatic representation of the collaboration between system components. It is the process for identifying, clarifying, and coordinating system requirements used in system analysis.

* + - 1. Use Case Diagram for Security Module

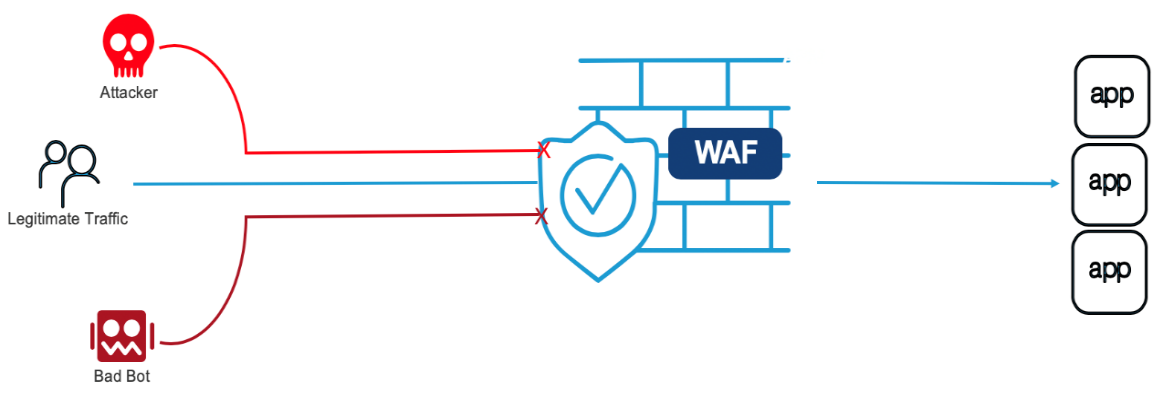


Figure 9: Use Case Diagram for Security Module

* + - 1. Use Case Diagram for Admin Module

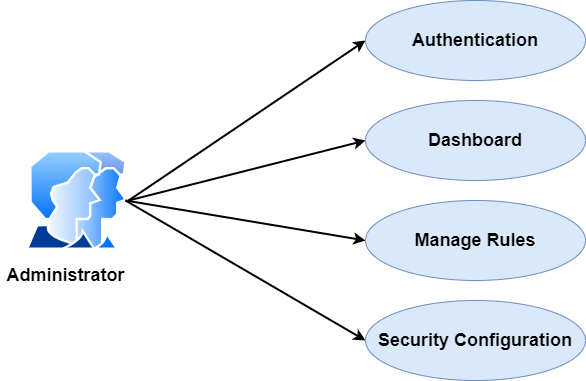


Figure 10: Use Case Diagram for Admin Module

* + 1. Database Table

To fully appropriate MySQL server innovation, it is significant to ensure that the database is very much structured. The documents names picked to mark every one of the tables made inside the database endeavour to mirror the table's motivation and, in this way, add to the thriving plan framework [17]. The in time step in structuring was to choose, as indicated by the needs and particulars of the venture, which tables ought to be made, and what kind of data everyone should hold [18].

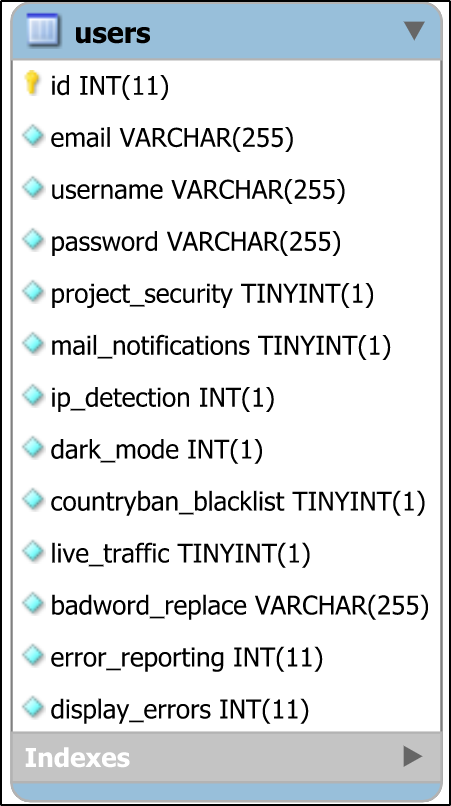


Figure 11: Database Table for users

This users table contains the columns and id holds the primary key.

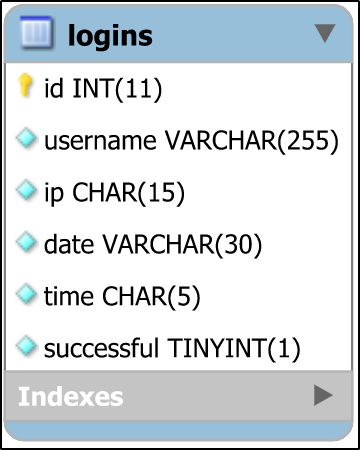


Figure 12: Database table for logins

This is the database table of login where id is the primary key.

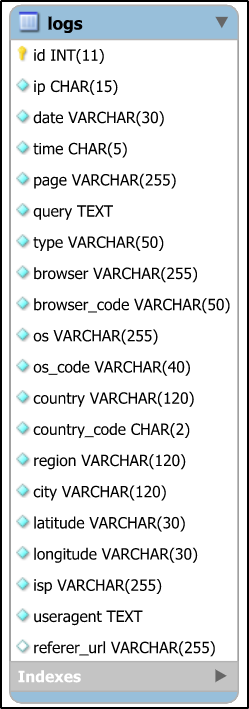


Figure 13: Database table for logs

This is the table for logs where id is the primary key.

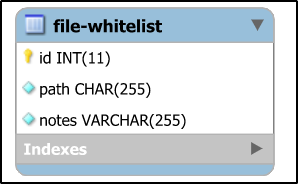


Figure 14: Database table for file-whitelist

It contains the columns of file-whitelist table.

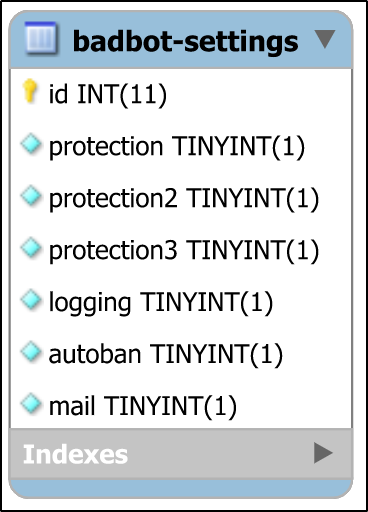


Figure 15: Database table for badbot-setting

In this table id is the primary key and it contains columns for badbot-setting.

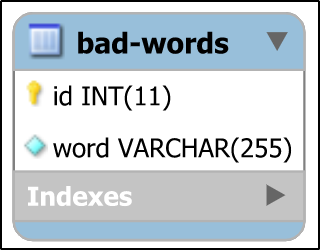


Figure 16: Database table for bad-words

This table is for the ba-words where id is the primary key.

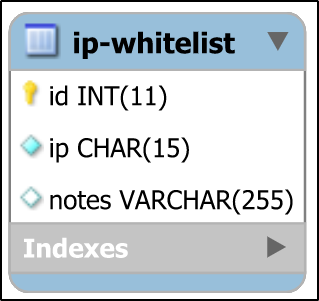


Figure 17: Database table for ip-whitelist

It is the ip-whitelist table and id is the primary key as well.

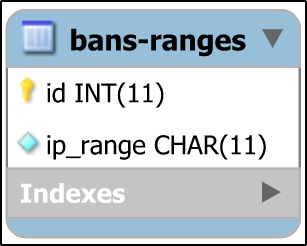


Figure 18: Database table for bans-ranges

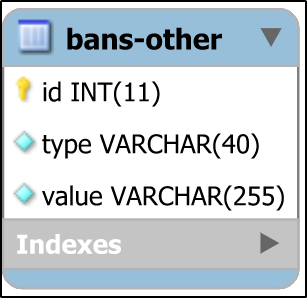


Figure 19: Database table for bans-other

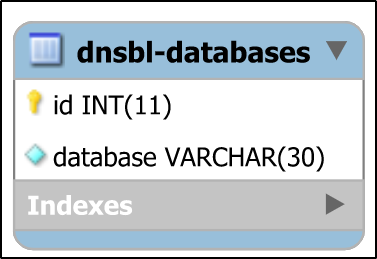


Figure 20: Database table for dnsbl-databases

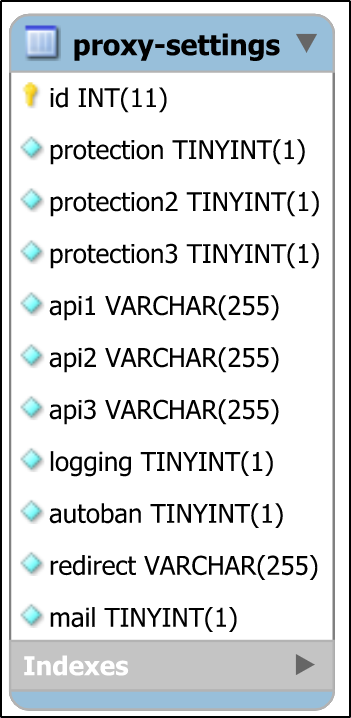


Figure 21: Database table for proxy settings

This database table contains the columns for the proxy settings and id is the primary key.

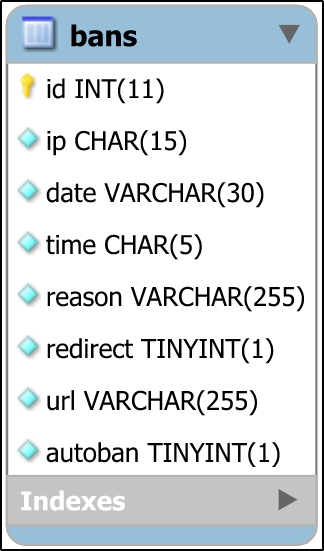


Figure 22: Database table for bans

It is the database table for bans.

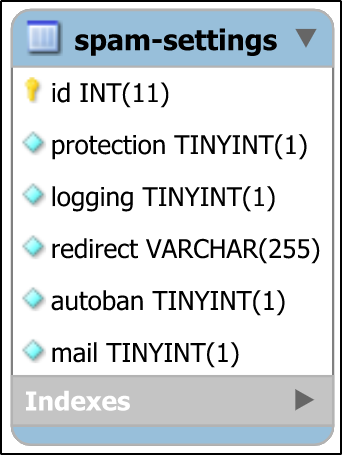


Figure 23: Database table for spam-setting

This is the database table for spam settings and id is the primary key.

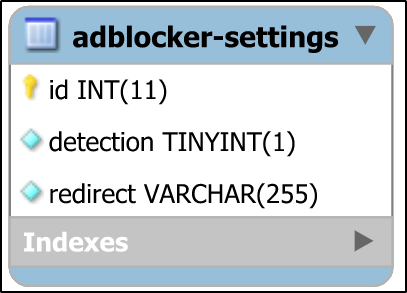


Figure 24: Database table for adblocker-settings

This database table holds the columns for adblocker-settings where id is the primary key.

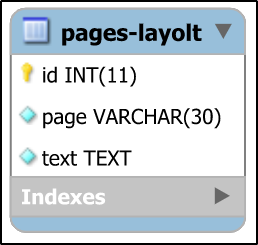


Figure 25:Database table for pages-layout

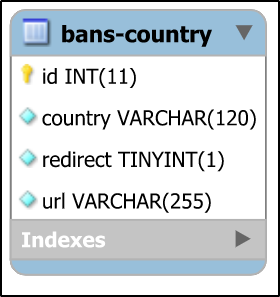


Figure 26: Database table for bans-country

It holds the columns for bans-country and id is the primary key.

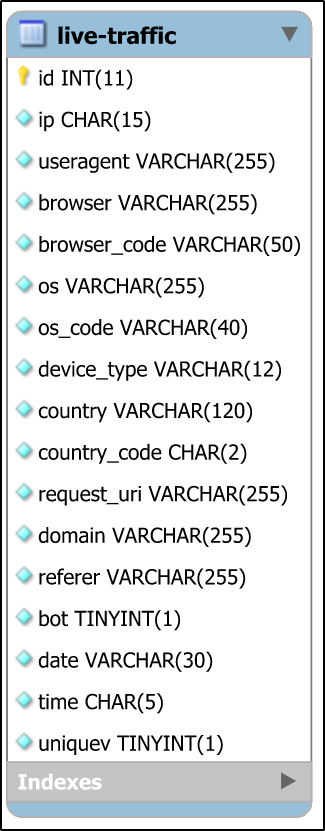


Figure 27: Database table for live-traffic

This is the database table for live-traffic where id is the primary key.

* + 1. Entity Relationship Diagram

The link between entity sets is depicted using an ER diagram. An entity set is a collection of comparable entities that may or may not contain properties. It represents a database's whole logical structure.

* + - 1. ERD for Security Module

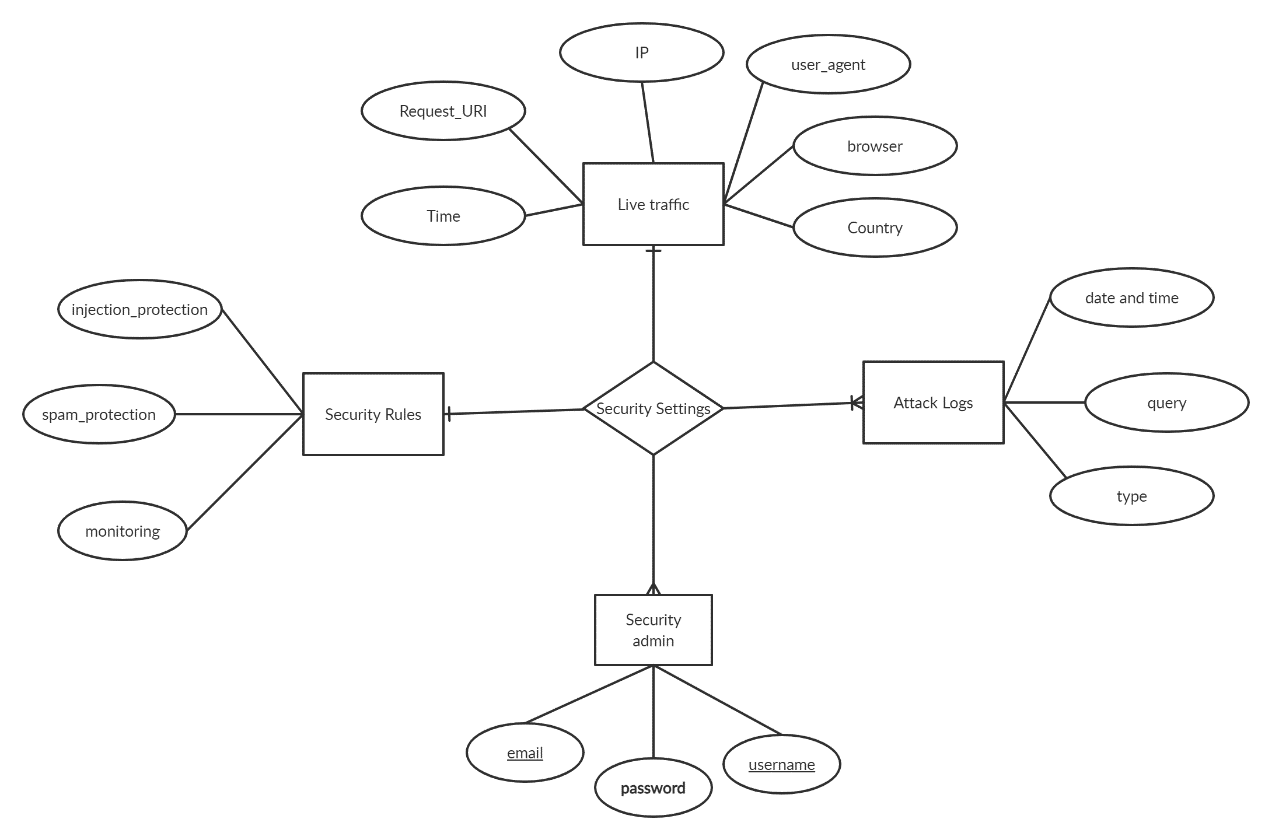


Figure 28: ERD for Security Module

* + 1. Data Flow Diagram

A data flow diagram shows the data flow in a graphical way across an information system. It exposes how Information passes from input to output.

* + - 1. Data Flow Diagram for Security Module:

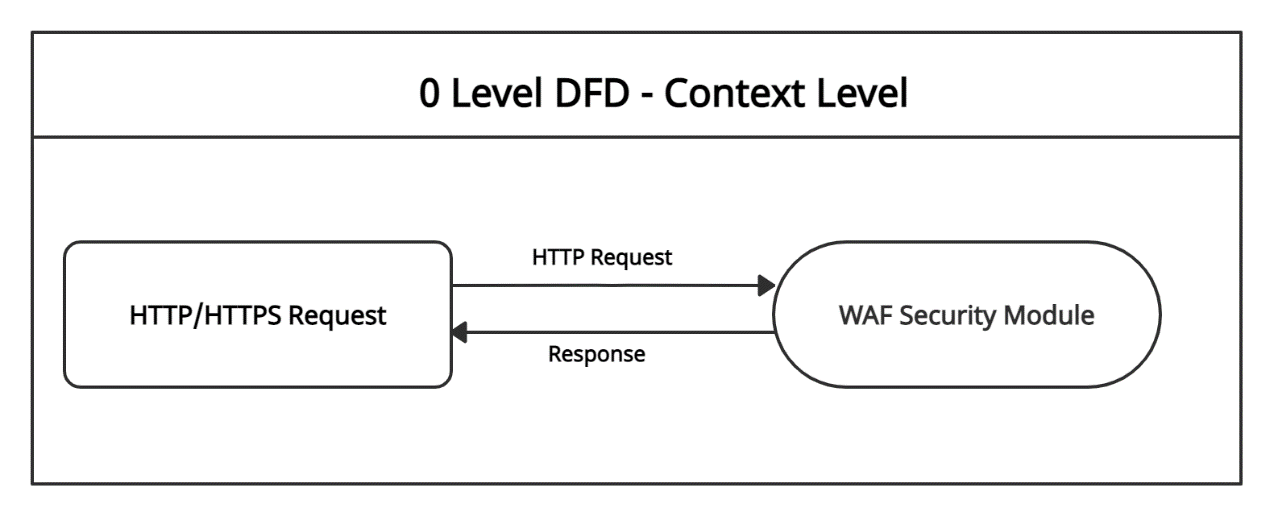


Figure 29: 0 Level Data Flow Diagram for Security Module

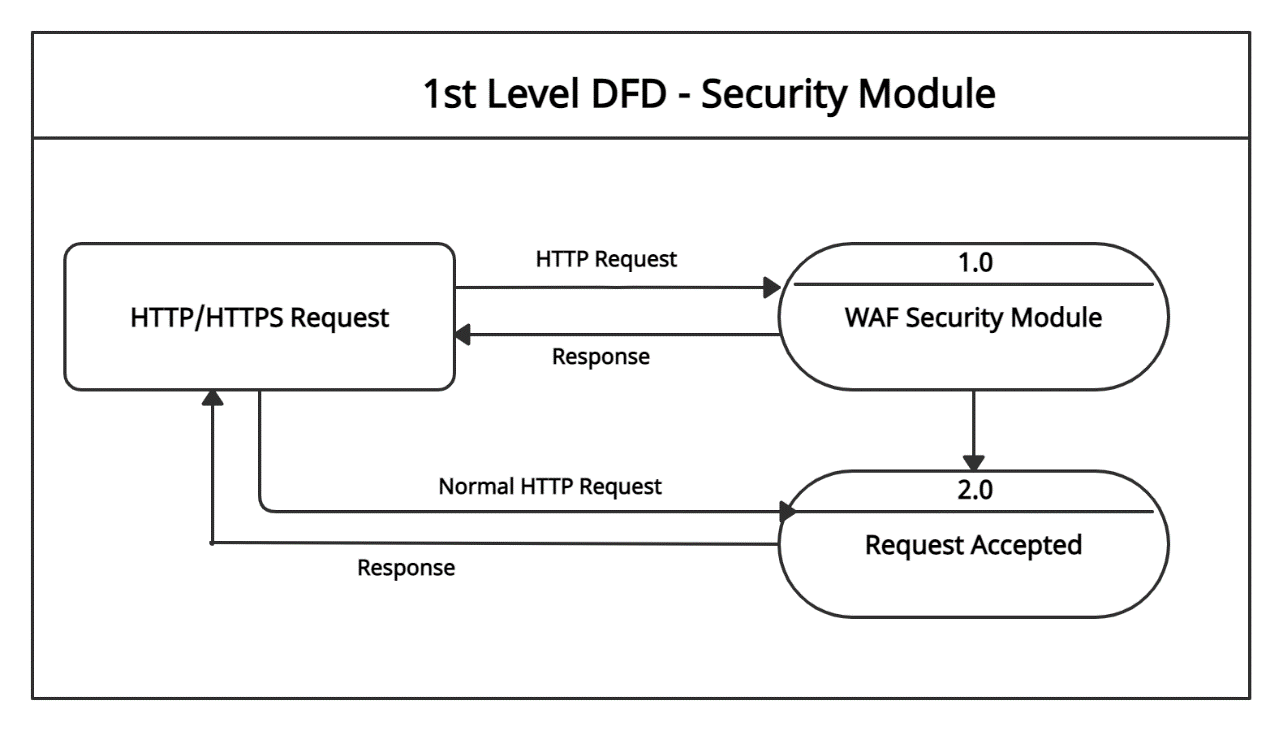


Figure 30: 1st Level Data Flow Diagram for Security Module

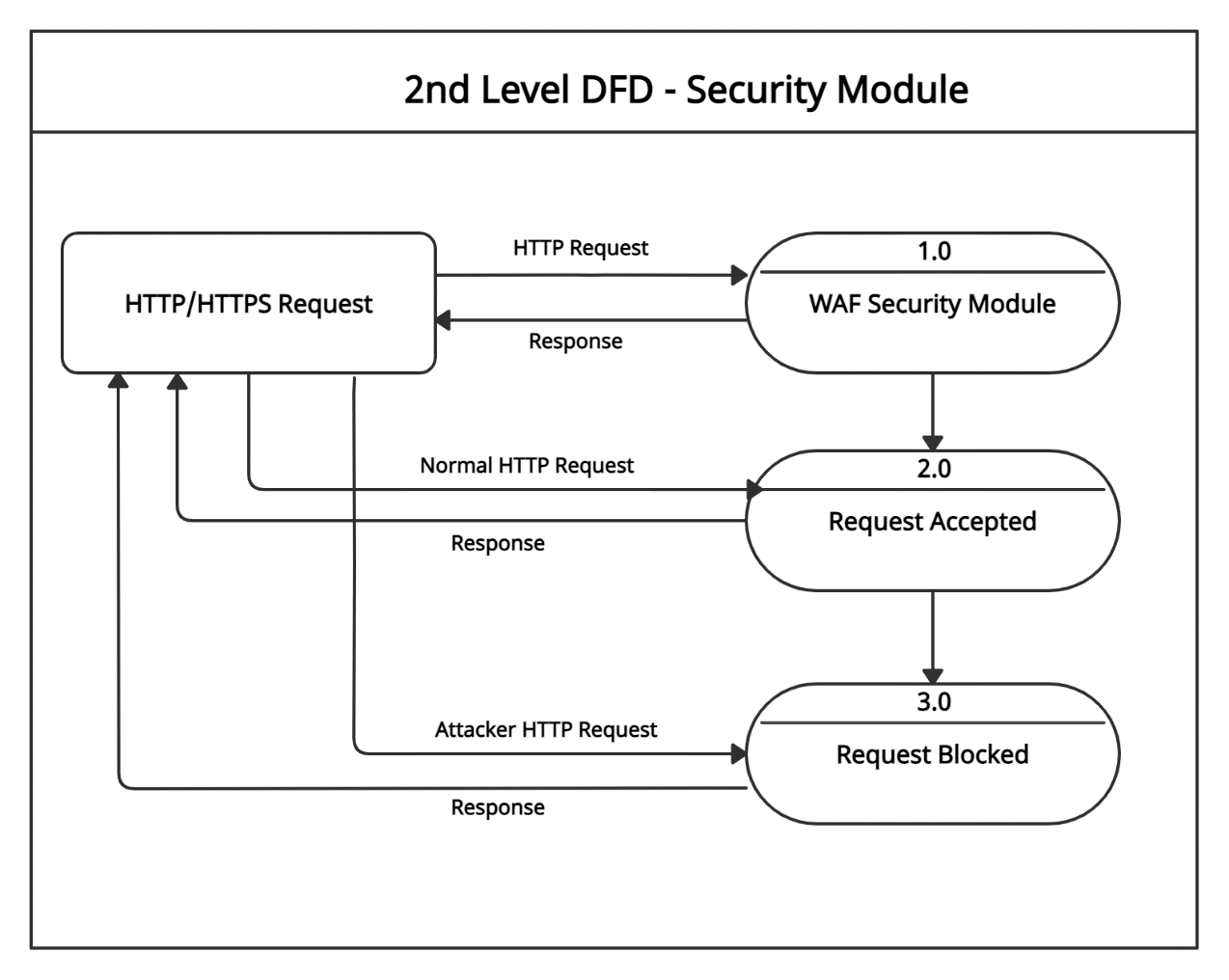


Figure 31: 2nd Level Data Flow Diagram for Security Module

* + - 1. Data Flow Diagram for Admin Module:

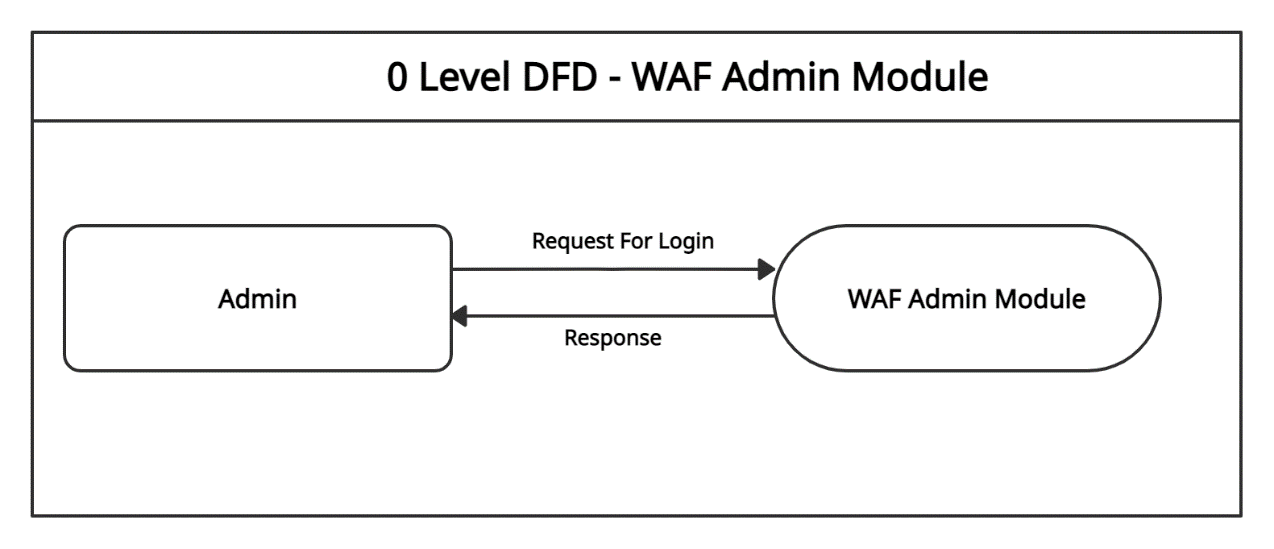


Figure 32: 0 level Data Flow Diagram for Admin Module

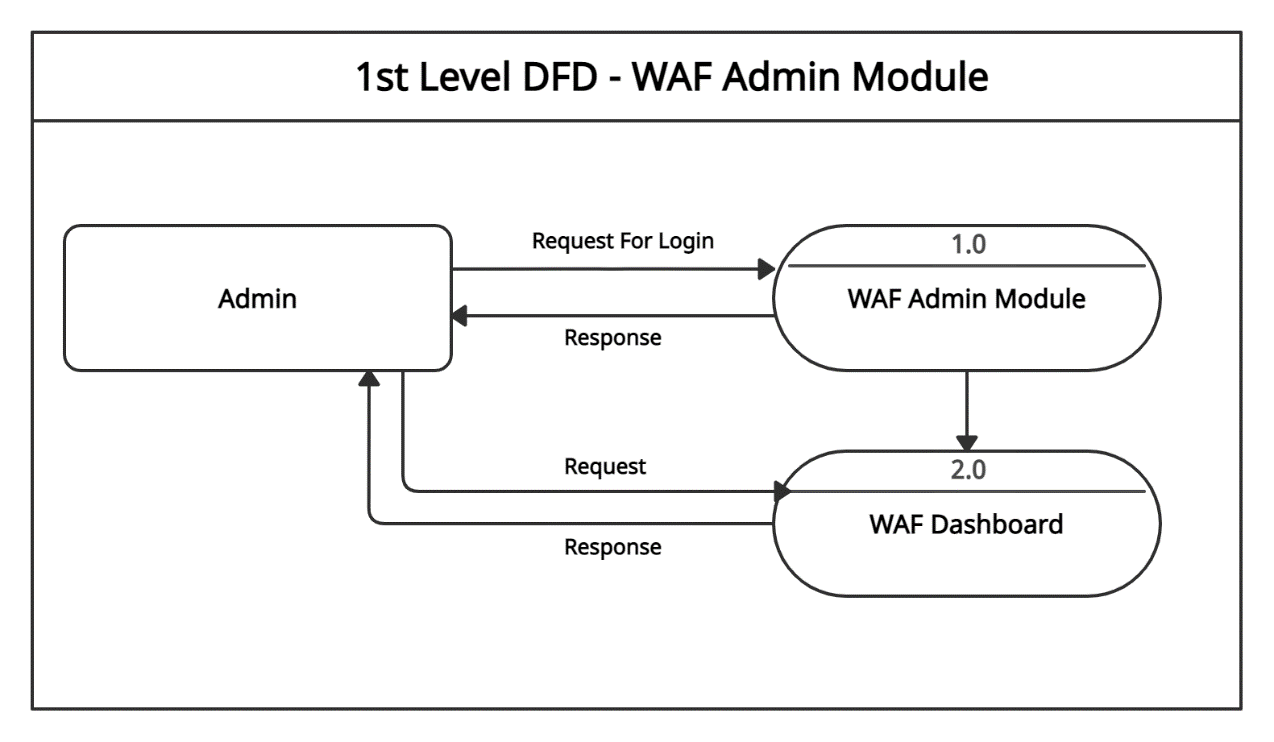


Figure 33: 1st level Data Flow Diagram for Admin Module

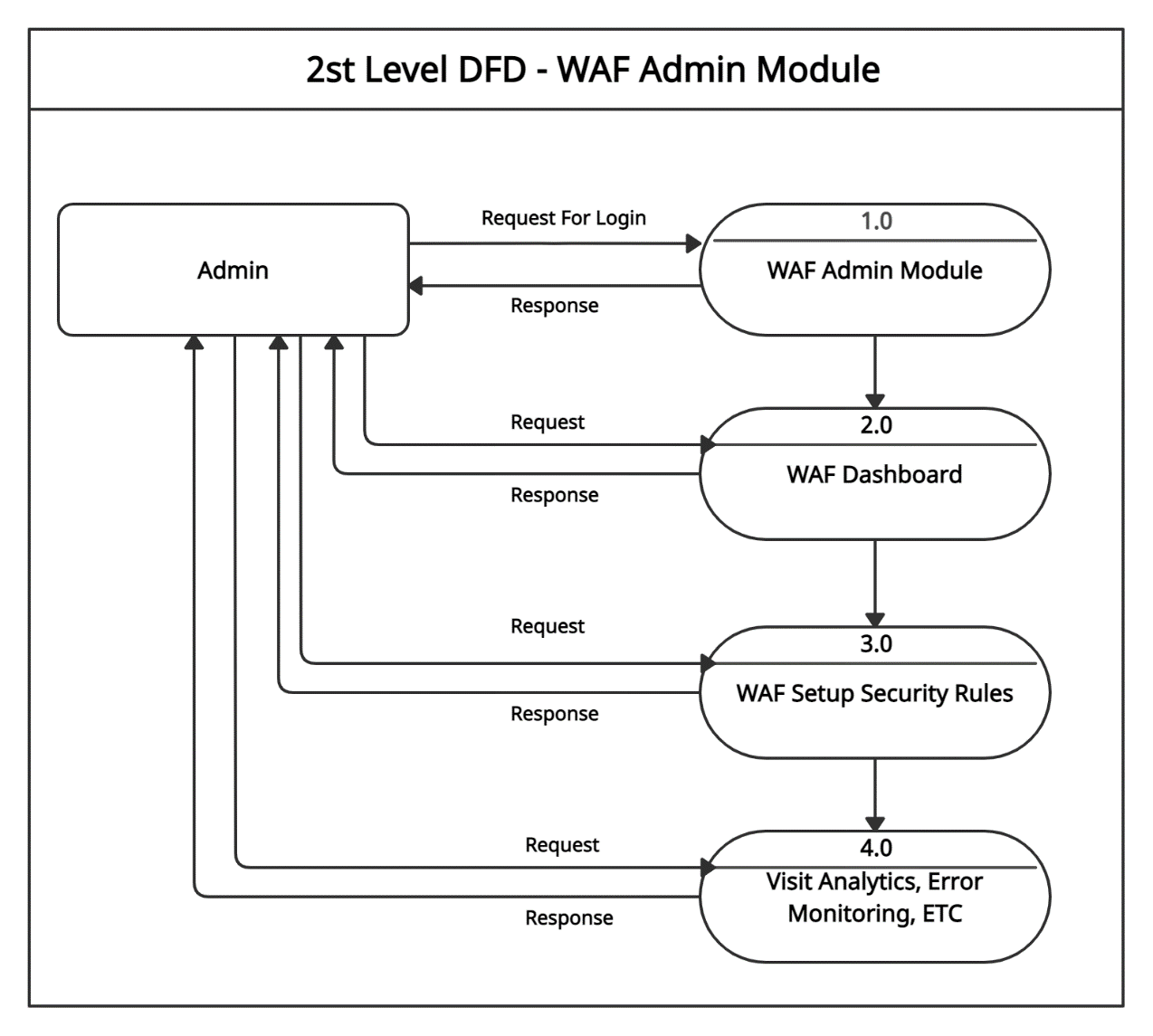


Figure 34: 2nd level Data Flow Diagram for Admin Module

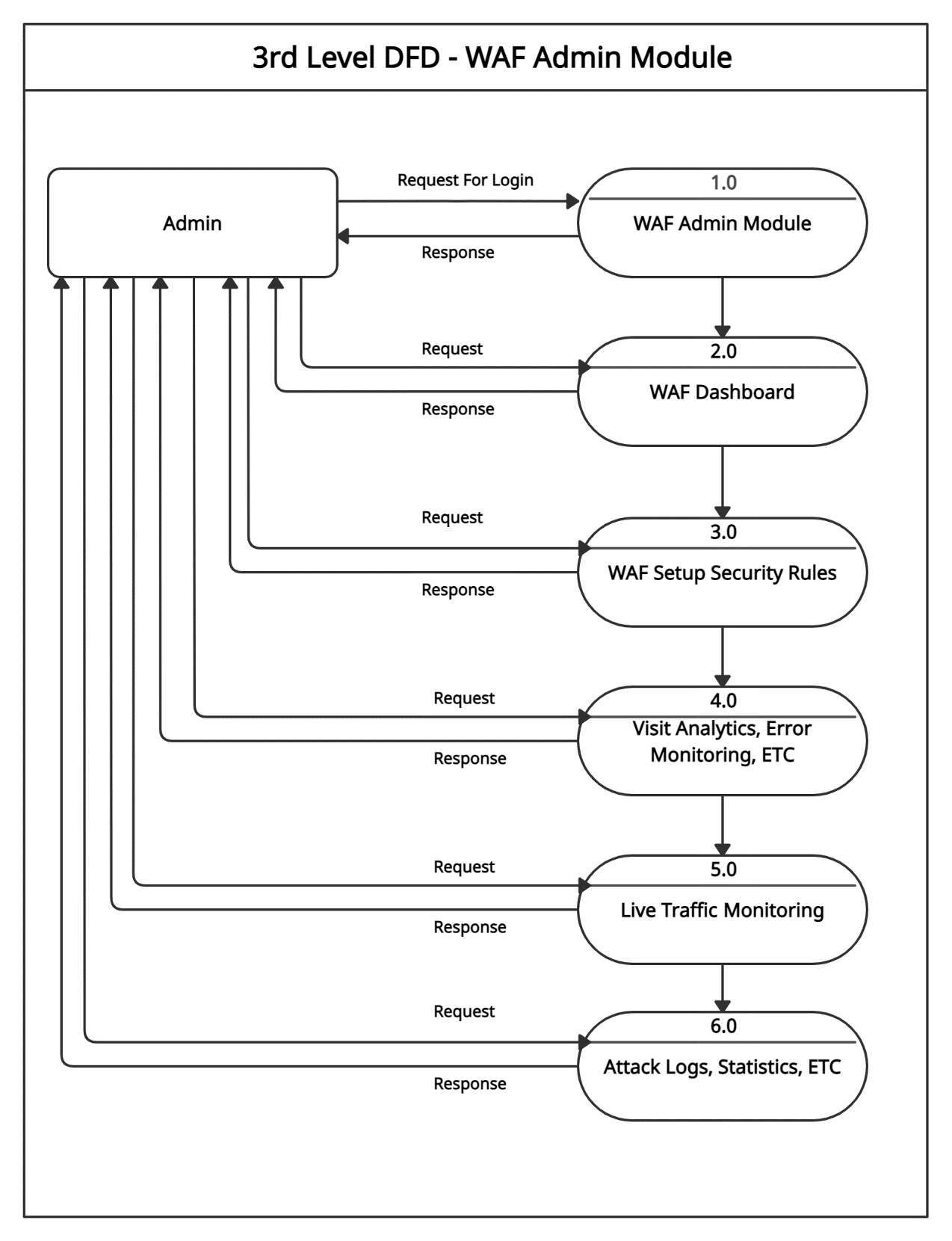


Figure 35: 3rd level Data Flow Diagram for Admin Module

* + 1. Design Model

A design model is an item-based picture or pictures that speak to the utilization cases for a framework. Alternatively, on the other hand, to put it another way, it is the way to depict a framework's execution and source code in a diagrammatic manner.

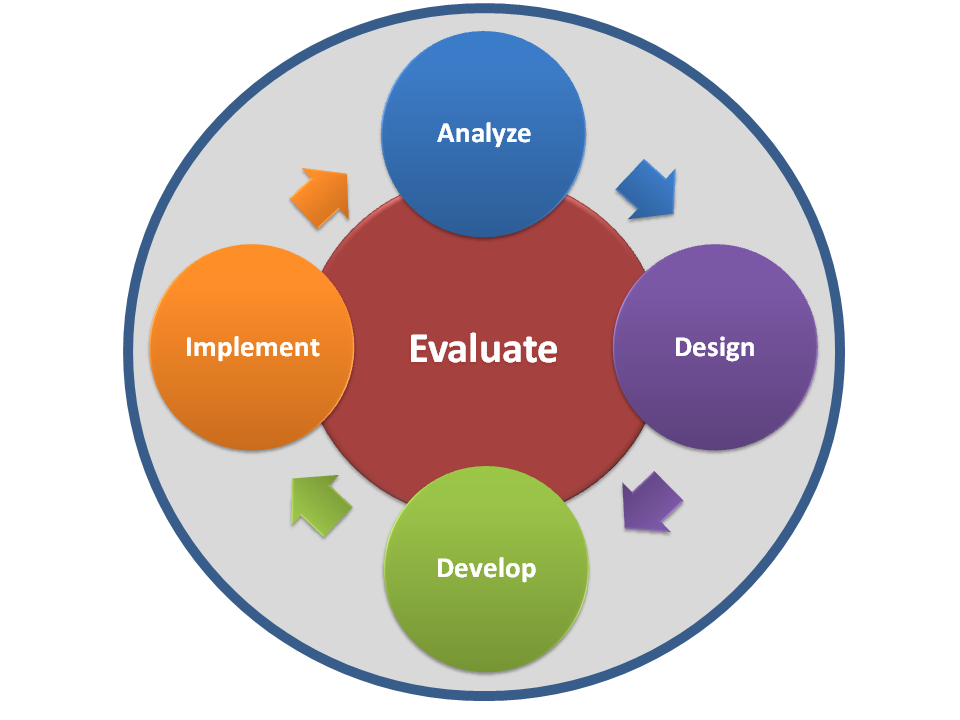


Figure 36: Design model

* + 1. Graphical Representation

The obtained results for WAF Security Module:

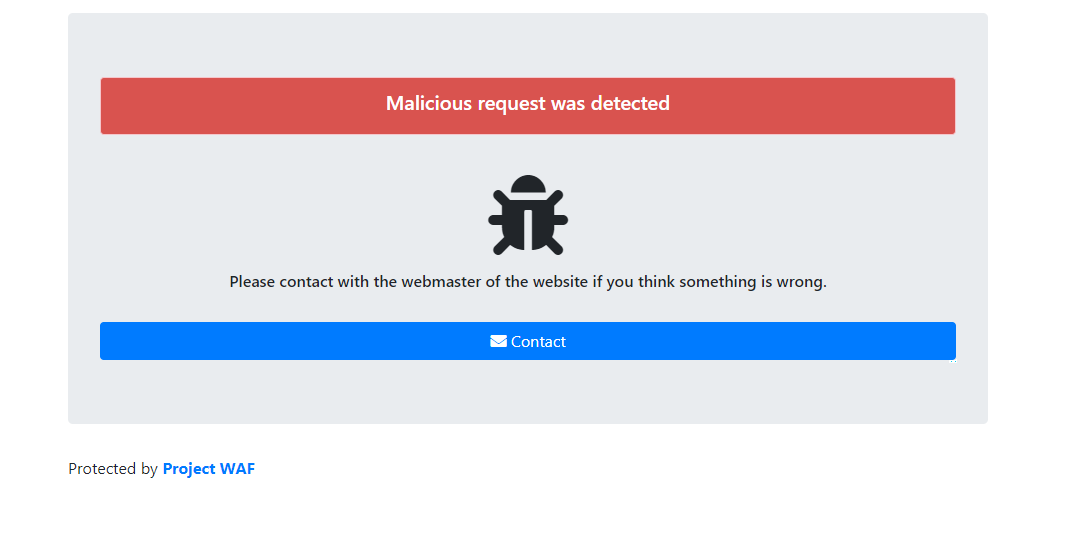


Figure 37: Malicious Request By Attacker Was Blocked

Malicious request was detected and blocked by WAF:

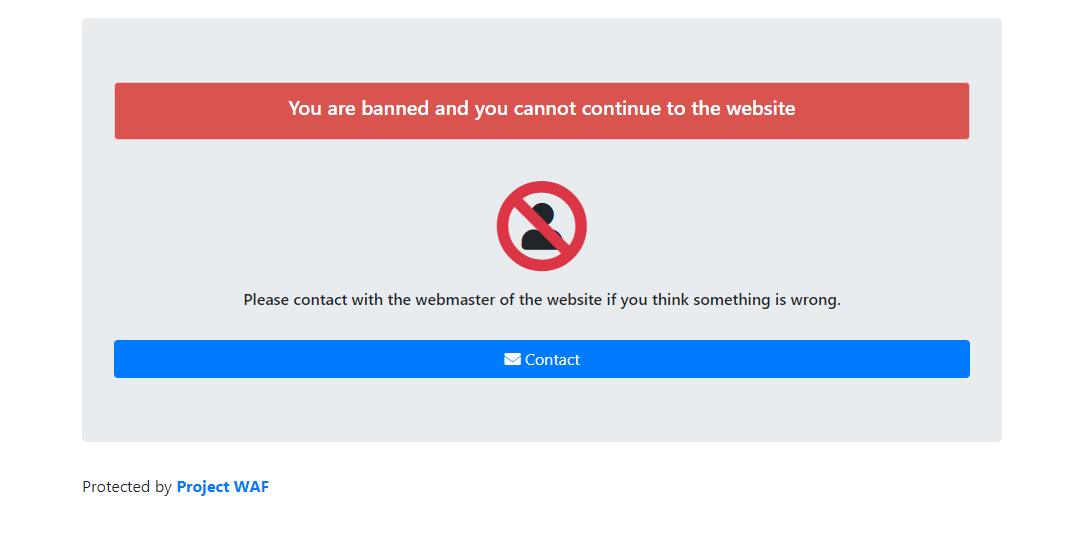


Figure 38: An Attacker Was Banned

For continuous malicious request an attacker was blocked by WAF:

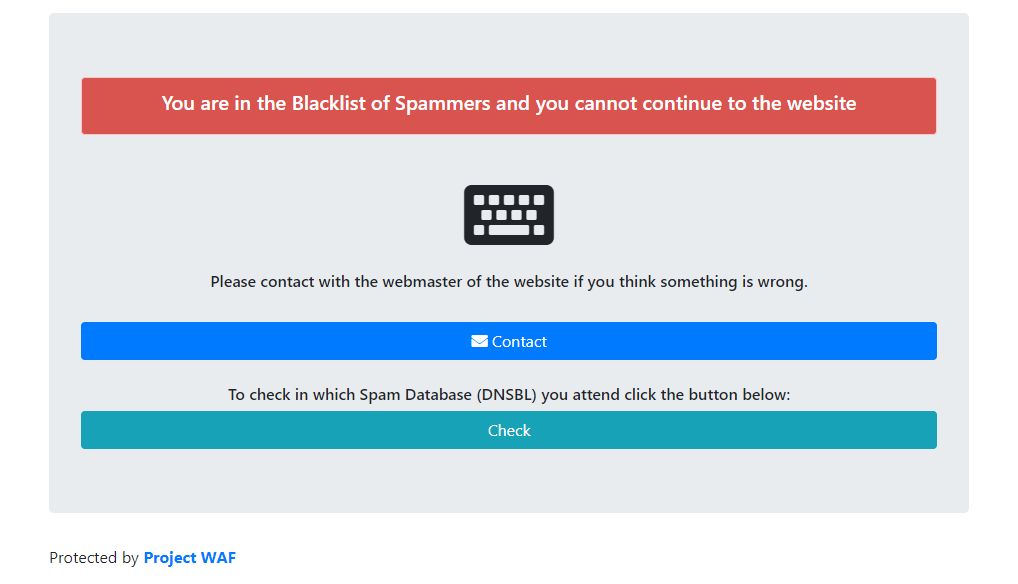


Figure 39: A spammer Request was Blocked

A Blacklisted IP was blocked by WAF for Spamming:



Figure 40: Administrator Authentication

This is the authentication page, where administrator of WAF can Manager security rules:

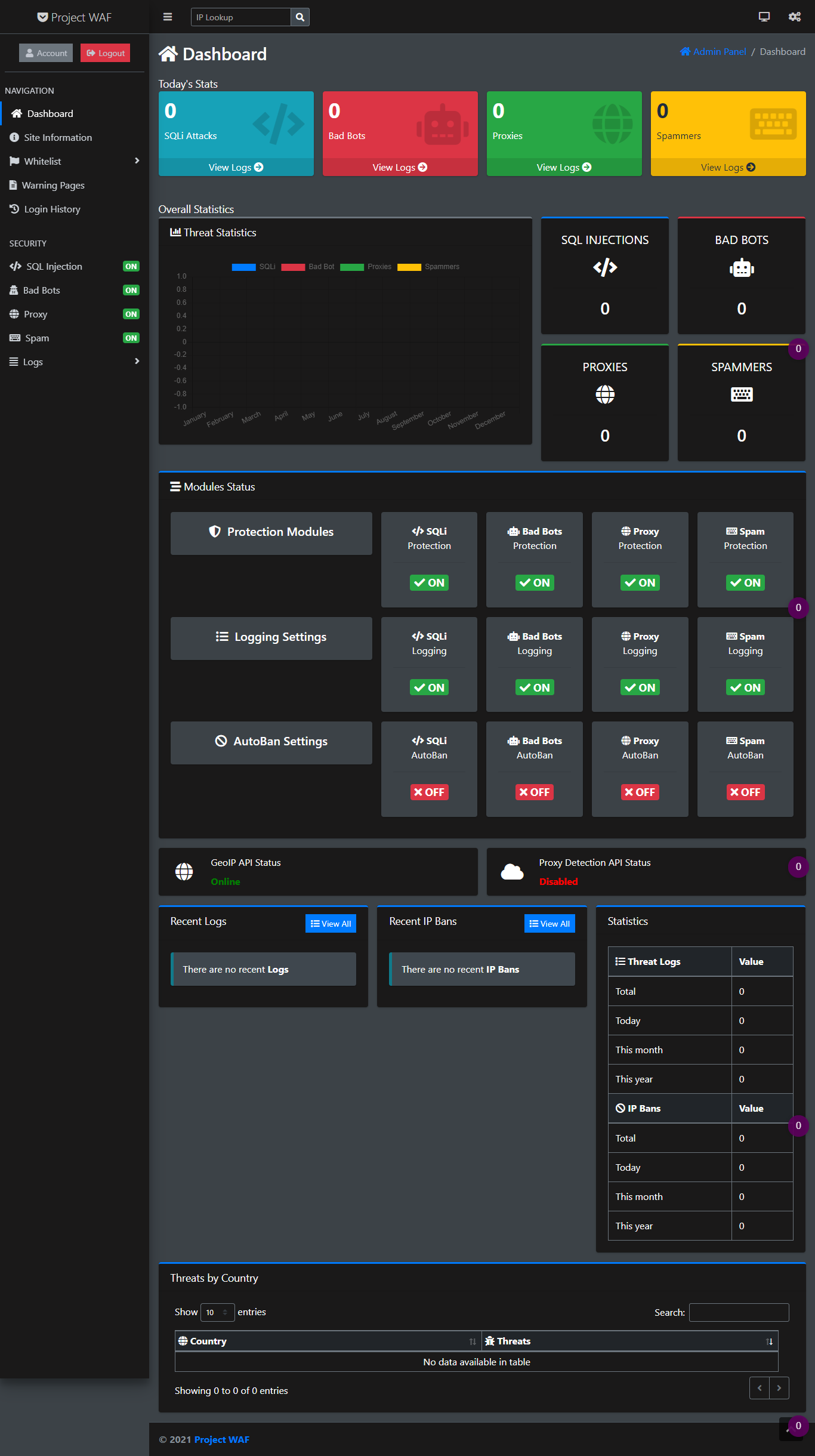


Figure 41: Admin Module Dashboard

This is the admin dashboard:

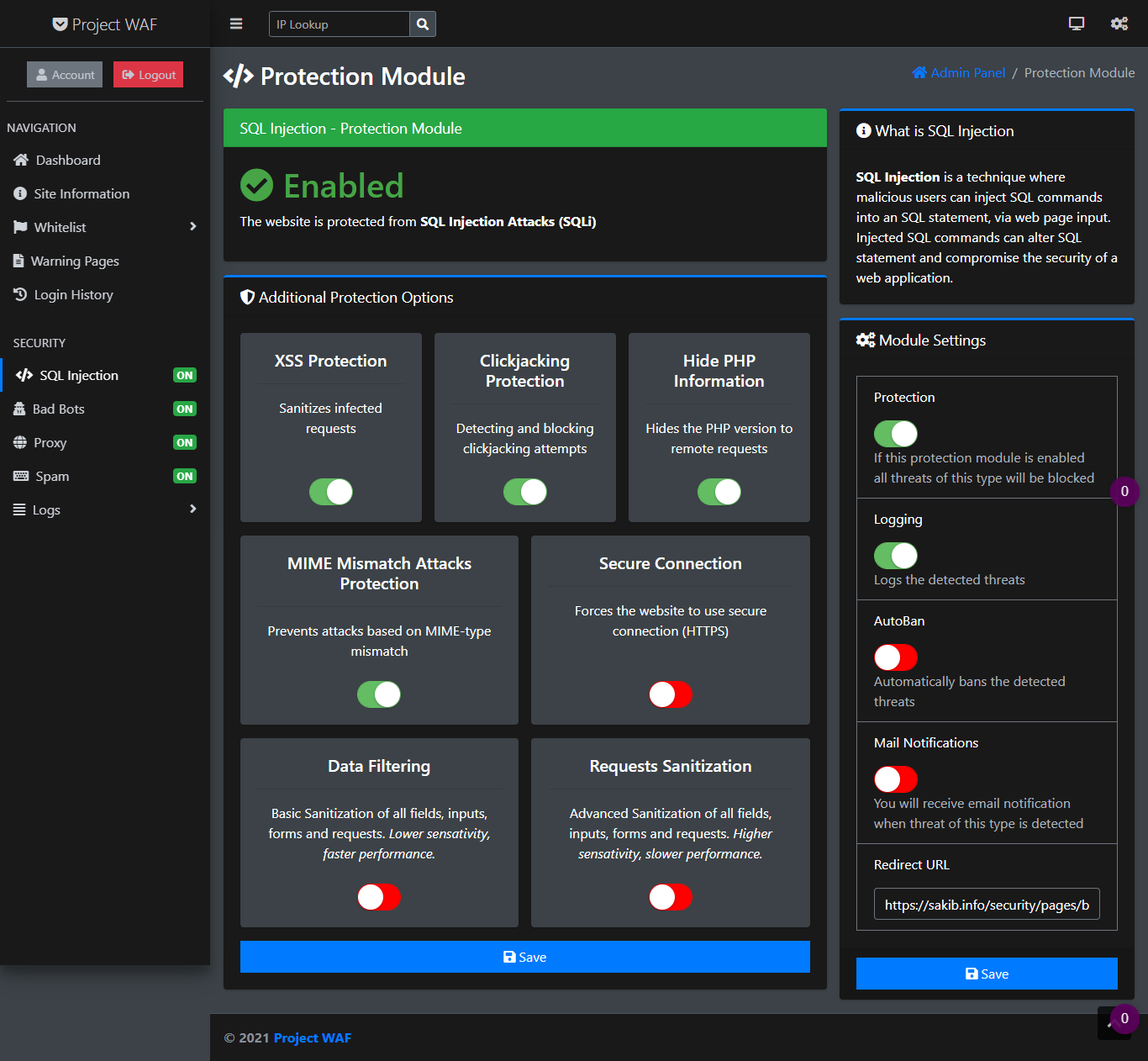


Figure 42: SQL Injection Protection Module

SQL Injection is a method where malevolent individuals may leverage Web page input to insert SQL instructions into SQL statements. Injecting SQL instructions may modify the SQL declaration and jeopardize the web application's security.

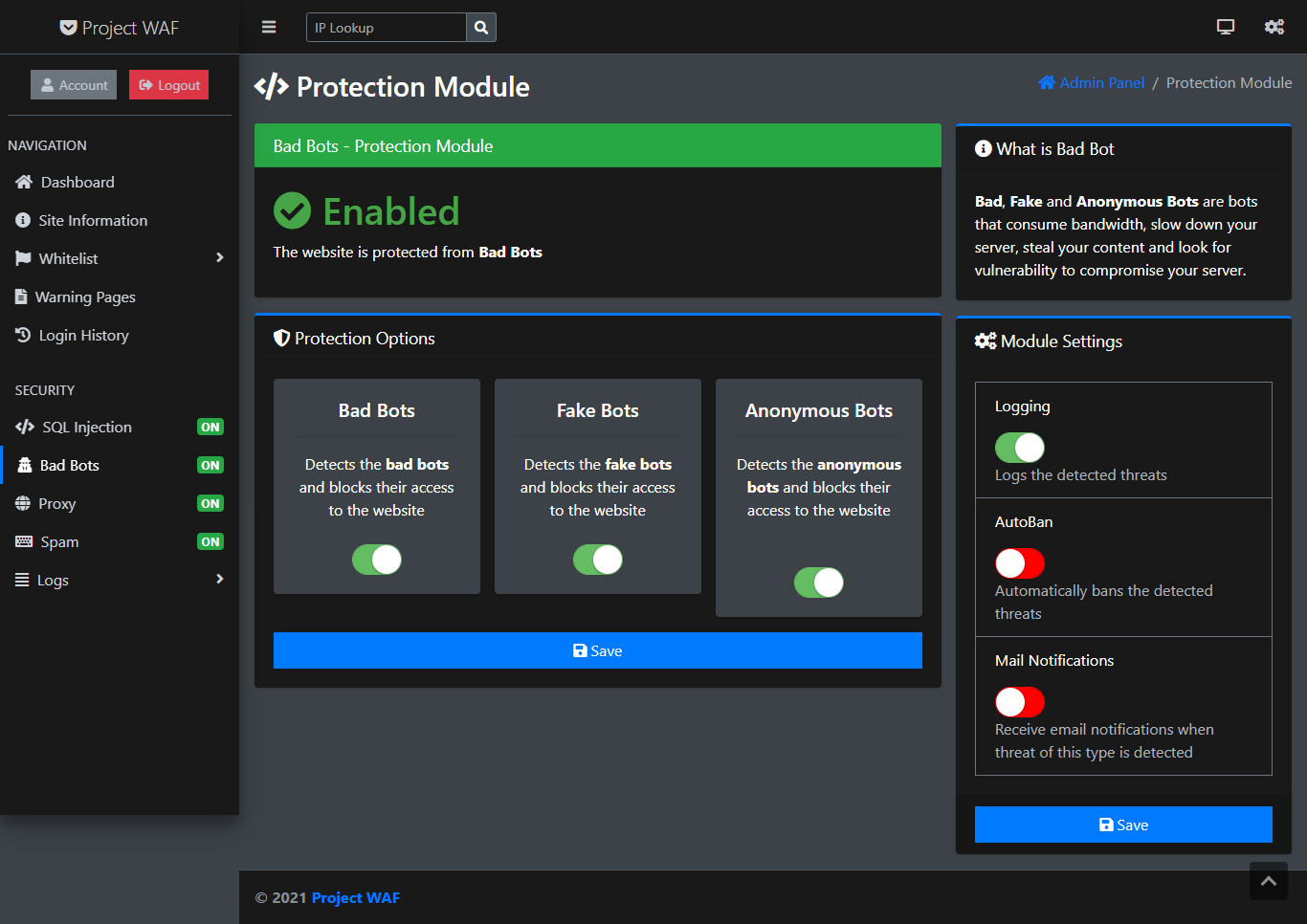


Figure 43: Bad Bots - Protection Module

Bad, Fake and Anonymous Bots are bots that consume bandwidth, slow down your server, steal your content and look for vulnerability to compromise your server.

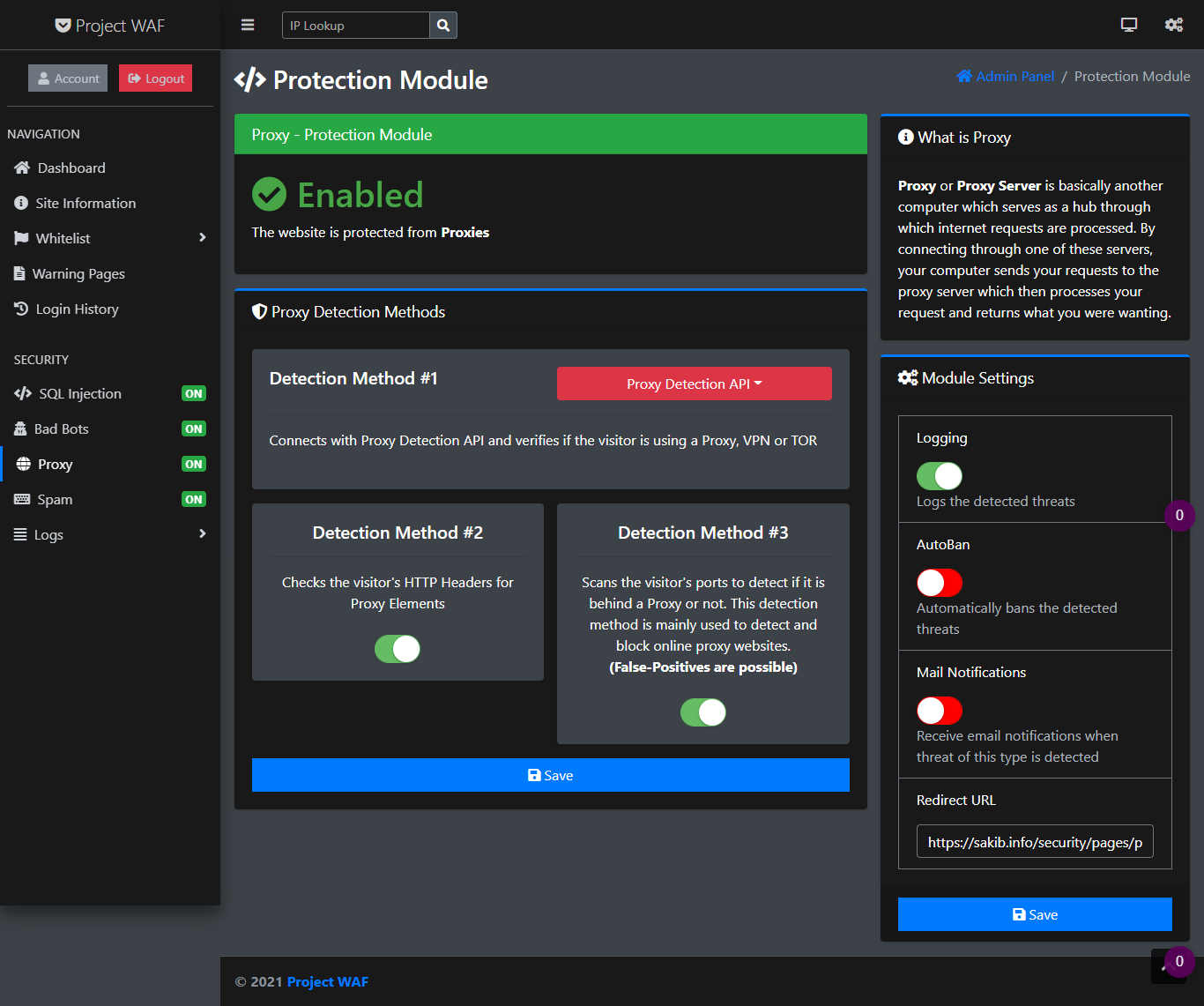


Figure 44:Proxy - Protection Module

Proxy or Proxy Server is simply another computer that acts as a hub for the processing of internet requests [19]. When users connect to one of the servers, the computer sends queries to the proxy server that then processes your application and delivers what user wanted [20].

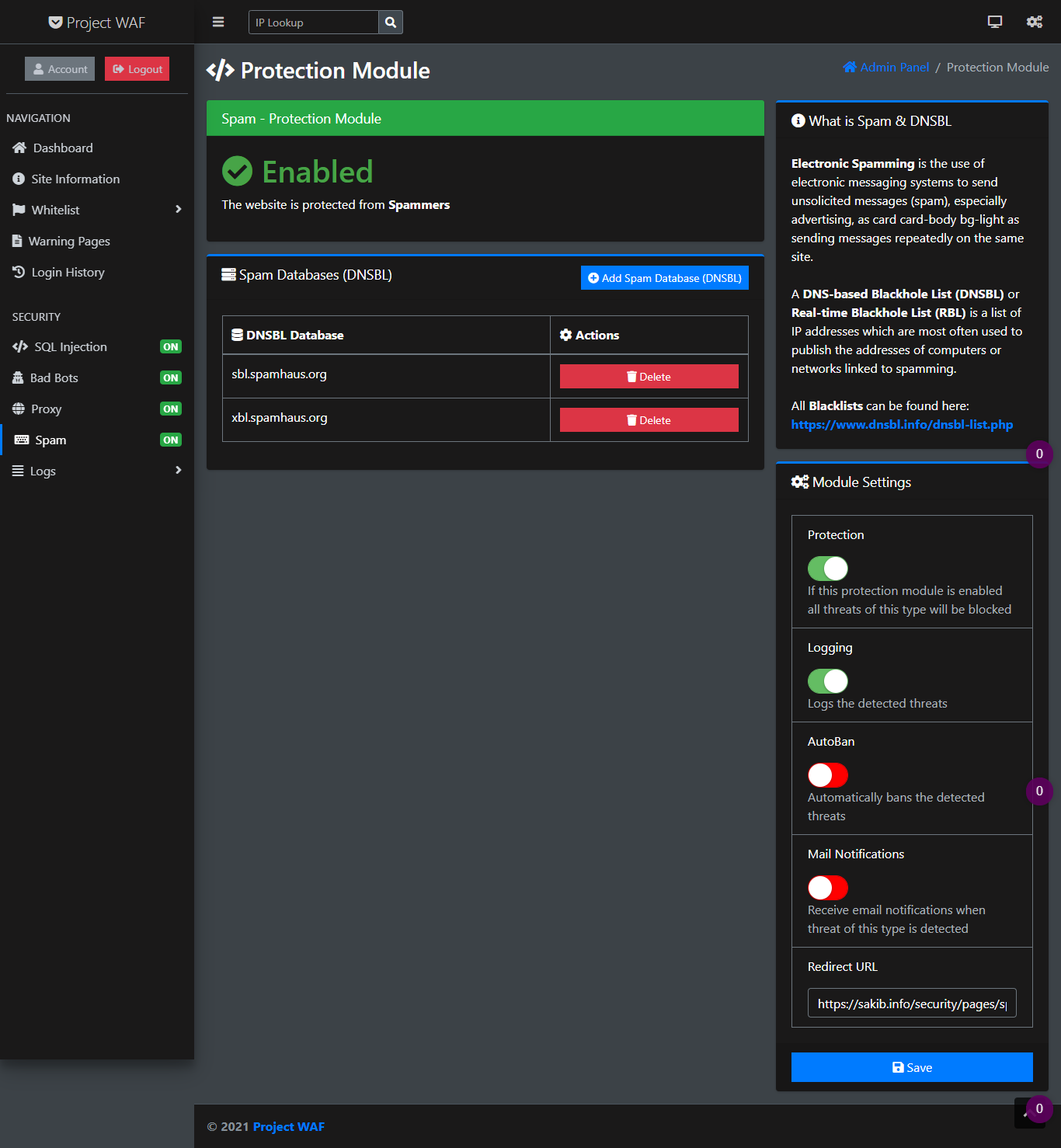


Figure 45: Spam - Protection Module

Electronic spamming employs electronic messaging technologies, notably spamming, to send messages to the same site repeatedly as card-body bg-light [21].

A DNS Blackhole List or real-time Blackhole List records IP addresses that are most commonly used to communicate computer addresses or spamming-related networks.

All Blacklists are available [here](https://www.dnsbl.info/dnsbl-list.php)

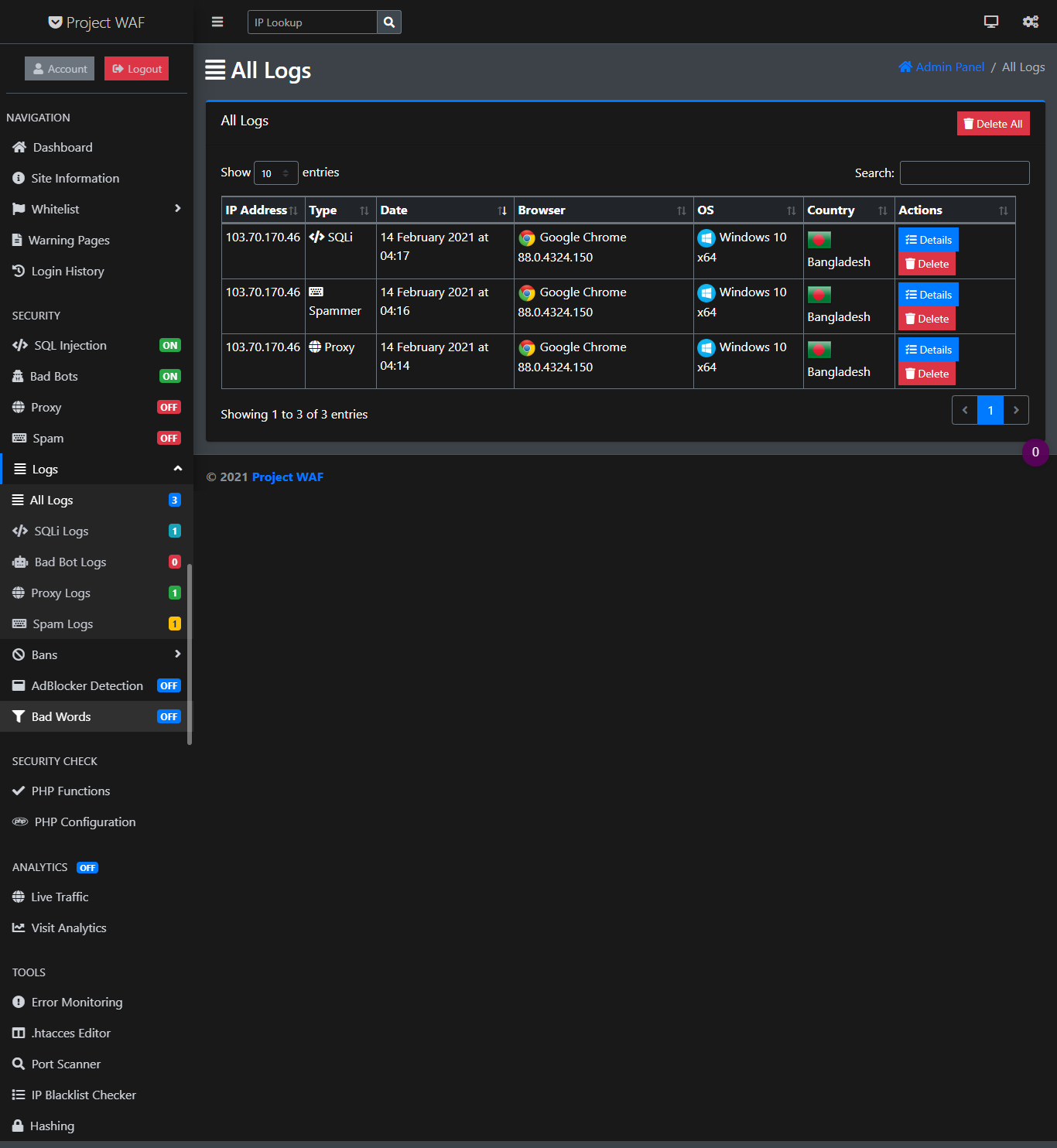


Figure 46: Attack Logs

This is the attack logs from attackers.

* + 1. Summary

This chapter reveals the Fundamental analysis and design of several vital elements. At the same time, it presents the most important diagrams of the system in the proper way.

* 1. Results and Discussion
     1. Introduction

This chapter reveals the result of the project. Which meets the goals properly.

* + 1. Results

a) Perfectly worked in every web application Successfully

b) Blocked malicious HTTP requests

c) Recognized normal HTTP requests

d) Detected Zero-Day attacks and exploits.

* + - 1. General

Finally, we have implemented our WAF (Web Application Firewall) on a cloud server. Then, to test its ability, we integrated it with other web applications also. Our WAF has successfully worked in every web application and blocked all of the malicious HTTP requests. It has successfully recognized all the normal HTTP requests and allowed them to pass through. Apart from it, Intelligent Pattern Recognition can also detect all the unknown and Zero-Day attacks and exploits. The proper detection rate is 100%.

* + 1. Discussions

After analyzing all the tests, we can clearly see that the WAF has passed in all the testing. It has also detected all the malicious requests and has been able to block all of them. The WAF has the ability to approve normal HTTP requests to get through. If the client application is likely to face SQL, XSS, RFL/LFI, or any other kind of attack, our WAF will be able to provide very effective security. The success rate of blocking any kind of malicious request and distinguishing it from normal requests is 100%.

* 1. Conclusion and Future Works
     1. Conclusion

Our goal was to create an intelligent WAF that will prevent security attacks by detecting and blocking the web app's vulnerabilities. We examined the performance of our Web Applications Firewall with several test sets, including hostile and benign traffic. Our goal was to test how efficient our WAF is and whether it is a feasible alternative to commercial solutions.

These objectives have been met by the present application. We had closely followed the specs and improved some of the features when it was necessary. The aims reached have formed the foundation of the application and this project. The construction of this web application was tough and rewarding.

* + 1. Contribution of this thesis

The major contributions in the project are:

Firstly, we have covered all the OWASP Top Ten Web Application vulnerabilities.

Secondly, our WAF has Intelligent Pattern Recognition which detects Unknown and Zero-Day Attacks and Exploits.

* + 1. Future works

Some aspects of the present research work can be further investigated and improved. Based on the literature reviews and studies conducted in this thesis, the following recommendations are proposed:

* + - 1. Limitations:

1) Currently working only for web apps developed using PHP

2) Does not provide quality protections for DDoS attacks

3) It requires installation for every server individually

* + - 1. Future Works of the project:

1) Upgrading the WAF to API so that it will be suitable for other programming languages.

References

[1] A. M. Hasan, D. T. Meva, A. K. Roy, and J. Doshi, "Perusal of web application security approach," in 2017 International Conference on Intelligent Communication and Computational Techniques (ICCT), 2017, pp. 90-95.

[2] "The State of Web Application Security," 2011.

[3] M. Sharma, A. Tandon, S. Narayan, and B. Bhushan, "Classification and analysis of security attacks in WSNs and IEEE 802.15. 4 standards: A survey," in 2017 3rd International Conference on Advances in Computing, Communication & Automation (ICACCA)(Fall), 2017, pp. 1-5.

[4] "Microsoft Security Intelligence Report," 2018.

[5] M. T. B. IVCM156308, "A COMPARATIVE ANALYSIS OF HEALTHCARE SYSTEM IOT AND INDUSTRIAL SCADA IOT FOR CYBERTERRORISM."

[6] A. Vernotte, F. Dadeau, F. Lebeau, B. Legeard, F. Peureux, and F. Piat, "Efficient detection of multi-step cross-site scripting vulnerabilities," in International Conference on Information Systems Security, 2014, pp. 358-377.

[7] J. J. Stephan, S. D. Mohammed, and M. K. Abbas, "Neural network approach to web application protection," International Journal of Information and Education Technology, vol. 5, p. 150, 2015.

[8] M. Ahmed and A. S. B. Ullah, "False data injection attacks in healthcare," in Australasian Conference on Data Mining, 2017, pp. 192-202.

[9] A. Sharma, A. Singh, N. Sharma, I. Kaushik, and B. Bhushan, "Security countermeasures in web based application," in 2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), 2019, pp. 1236-1241.

[10] H. S. Lallie, L. A. Shepherd, J. R. Nurse, A. Erola, G. Epiphaniou, C. Maple, et al., "Cyber security in the age of covid-19: A timeline and analysis of cyber-crime and cyber-attacks during the pandemic," arXiv preprint arXiv:2006.11929, 2020.

[11] C. Ventures, "2019 official annual cybercrime report," 2019.

[12] R. Anderson, C. Barton, R. Bölme, R. Clayton, C. Ganán, T. Grasso, et al., "Measuring the changing cost of cybercrime," 2019.

[13] Z. Su and G. Wassermann, "The essence of command injection attacks in web applications," Acm Sigplan Notices, vol. 41, pp. 372-382, 2006.

[14] J. J. Singh, H. Samuel, and P. Zavarsky, "Impact of paranoia levels on the effectiveness of the modsecurity web application firewall," in 2018 1st International Conference on Data Intelligence and Security (ICDIS), 2018, pp. 141-144.

[15] I. Schmitt and S. Schinzel, "WAFFle: Fingerprinting Filter Rules of Web Application Firewalls," in WOOT, 2012, pp. 34-40.

[16] Y. Wang, Z. Wang, V. Puig, and G. Cembrano, "Zonotopic set-membership state estimation for discrete-time descriptor LPV systems," IEEE Transactions on Automatic Control, vol. 64, pp. 2092-2099, 2018.

[17] A. Varadarajan and J. O. K. Olsson, "Interactions layer in a web application builder framework," ed: Google Patents, 2020.

[18] L. Kuperman and K. Mancevicius, "Analyzing web application behavior to detect malicious requests," ed: Google Patents, 2020.

[19] K. Neupane, R. Haddad, and L. Chen, "Next generation firewall for network security: A survey," in SoutheastCon 2018, 2018, pp. 1-6.

[20] K. Nagendran, S. Balaji, B. A. Raj, P. Chanthrika, and R. Amirthaa, "Web Application Firewall Evasion Techniques," in 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), 2020, pp. 194-199.

[21] H. Mac, D. Truong, L. Nguyen, H. Nguyen, H. A. Tran, and D. Tran, "Detecting attacks on web applications using autoencoder," in Proceedings of the ninth international symposium on information and communication technology, 2018, pp. 416-421.

Appendices

1. CREDENTIALS OF THE DEMO APP

Live Demo: <https://sakib.info/security>

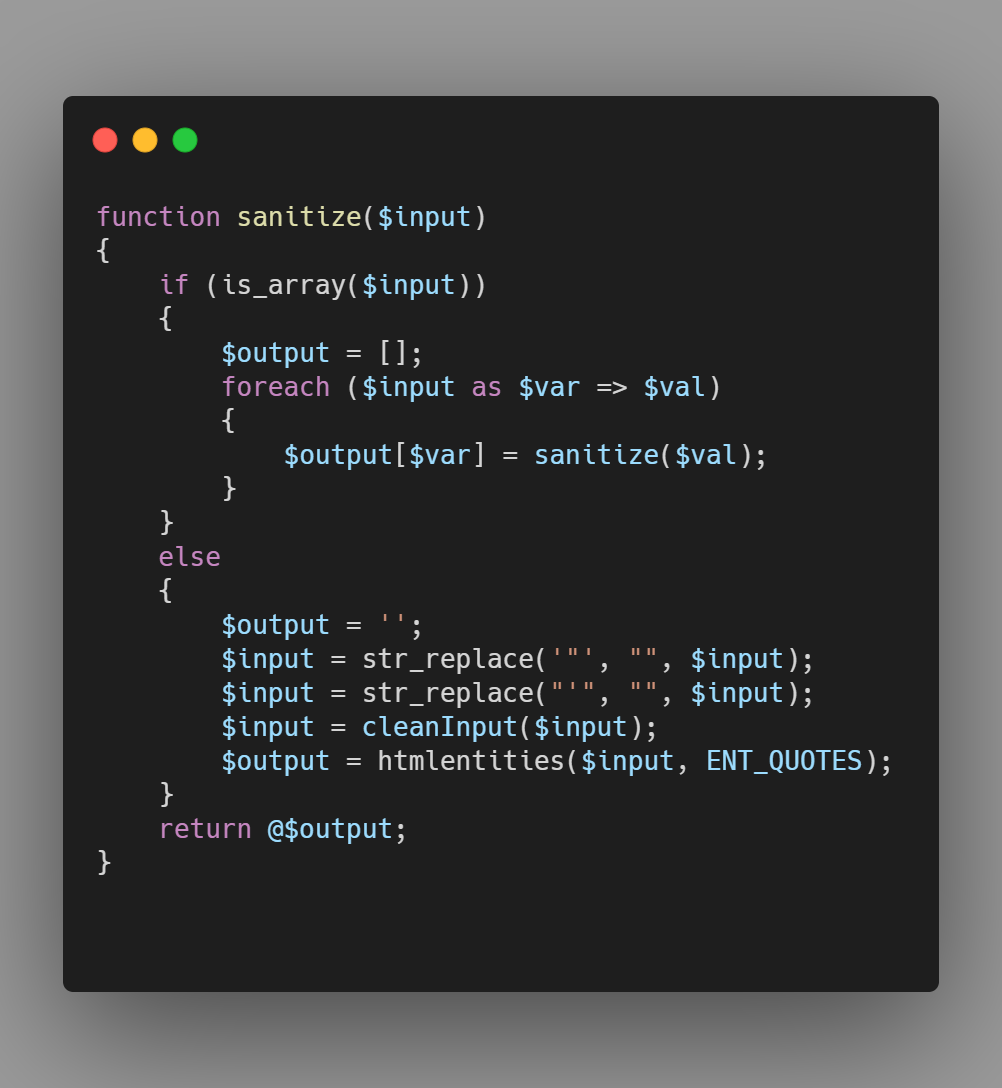
Username: sakib

Password: 123456

1. SECURITY MODULE: CLEAN INPUT ALGORITHM



1. SECURITY MODULE: DATA SANITIZATION ALGORITHM



1. PATTERNS FOR DETECTING MALICIOUS REQUEST

