**PRJ2015CE044**

**Software Requirements Specification Document (SRS)**

**Table of Contents**

1. Introduction 3

1.1 Purpose 3

1.2 Scope 3

1.3 Definitions, Acronyms, and Abbreviations 3

2. Project Management 4

2.1 Project Development Approach and Justification 4

2.2 Project Effort, Time and Cost Estimation 5

2.3 Project Scheduling 7

3. System Analysis 9

3.1 Users of System 9

3.2 Requirements of System 9 3.2.1 Functional Requirement 9

3.2.2 Nonfunctional Requirement 9

3.3 Feasibility Study 9

3.3.1 Technical Feasibility 9

3.3.2 Economic Feasibility 10

3.3.3 Optional Feasibility 10

3.4 UML Diagrams 10

4. System Design 16

5. Implementation & Testing 19

6. Conclusion 28

1. **INTRODUCATION**

priWAF is a web application firewall to patch a vulnerable site using mod\_security. It will work as a Layer 7 firewall for securing a website against application layer threats.

**1.1 Purpose**

The purpose of this document is to give a detailed description of the requirements for the “priWAF”. It will illustrate the purpose and complete declaration for the development of the system. It will also explain system constraints, technologies, interface and interactions with other external applications. This document is primarily intended to be proposed to a customer for its approval and a reference to develop the first version of the system for the development team.

**1.2 Scope**

The “priWAF” is a web application firewall that helps secure web applications against OWASP top vulnerabilities and threats while providing SSL certificate. Rights & privileges are under Indusface Pvt. Ltd to serve the services to their clients.

Users should provide their Server host IPs & other details to the WAF portal admin-user. This information will be added by WAF admin manually. Then after the clients IPs are given to the audit team, a team will scan those IPs and also exploited vulnerabilities manually attacked on web pages that given by client. Audit team will forward report to that client, if the client is satisfied with their report and wants to secure their web application then their vulnerability report forwarded to research & signature development team, they will try to patch those vulnerability using CRS (core rule set) / or develop (custom rule sets) a specific rules for them & tested on virtual machine using reverse proxy for that specified web application. The team will then push those rule sets to the WAF portal admin. Admin can manage them in WAF portal.

* 1. **Objective**

To make the web application safer using mod\_security. priWAF using mod\_security. It is a really useful technology in today’s cyber world to secure web application against threats.

1. **Project Management**

Project planning is unquestionably one of the most significant works in developing any project. Before the project can begin, it is best to estimate the work to be done, what resources will be required and how much time will elapse from start to the finish of a project. Planning helped us to prepare a framework that enables us to make a reasonable estimate of all such things.

**2.1 Project Development Approach and Justification**

I have chosen the spiral model of software development for my project. The diagrammatic representation of this model appears like a spiral with many loops. The exact number of loops in the spiral is not fixed. Each loop of the spiral represents a phase of the software process. For example, the innermost loop might be concerned with feasibility study. The next loop with requirements specification, the next one with design, and so on. Each phase in this model is split into four sectors (or quadrants). The following activities are carried out during each phase of a spiral model.



Figure 2.1 Spiral Model

**First quadrant (Objective Setting)**

* During the first quadrant, it is needed to identify the objectives of the phase.
* Examine the risks associated with these objectives.

**Second Quadrant (Risk Assessment and Reduction)**

* A detailed analysis is carried out for each identified project risk.
* Steps are taken to reduce the risks. For example, if there is a risk that the requirements are inappropriate, a prototype system may be developed.

**Third Quadrant (Development and Validation)**

* Develop and validate the next level of the product after resolving the identified risks.

**Fourth Quadrant (Review and Planning)**

* Review the results achieved so far with the customer and plan the next iteration around the spiral.
* Progressively more complete version of the software gets built with each iteration around the spiral.

The spiral model is called a Meta model since it encompasses all other life cycle models. Risk handling is inherently built into this model. The spiral model is suitable for development of technically challenging software products that are prone to several kinds of risks. However, this model is much more complex than the other models.

**JUSTIFICATION**

This model is best suited for this system as the customer mostly has clear idea about the requirements. As well as there are very less possibilities for major changes in requirements. It makes the implementation easier in terms of simplicity. There are no risks present in developing the system and step by step development is desirable.

* 1. **Project Effort, Time and Cost Estimation**

For any software project, it is again equally important to estimate the project effort, time and cost. I have used the concept of Function Point (FP) Analysis and COCOMO Model to implement the same.

**Function Point (FP) Analysis**

For a software project, the project size is a measure of the problem complexity in terms of the effort and time required to develop the product. Function point (FP) is one the metrics used to estimate this size of the problem. We are interested primarily in estimating the cost and duration of the project. Function point metric was proposed by Albrecht [1983]. This metric overcomes many of the shortcomings of the LOC metric used also as a metric.

One of the important advantages of using the function point metric is that it can be used to easily estimate the size of a software product directly from the problem specification. This is in contrast to the LOC metric, where the size can be accurately determined only after the product has fully been developed.

The conceptual idea behind the function point metric is that the size of a software product is directly dependent on the number of different functions or features it supports. Each function when invoked reads some input data and transforms it to the corresponding output data. A computation of the number of input and the output data values to a system gives some indication of the number of functions supported by the system. Also, in addition to the number of basic functions that a software performs, the size is also dependent on the number of files and the number of interfaces. Function point is computed in two steps. The first step is to compute the unadjusted function point (UFP).

**UFP =** (Number of inputs)\*4 + (Number of outputs)\*5 +

(Number of inquiries)\*4 + (Number of files)\*10 +

(Number of interfaces)\*10

**COCOMO Model**

It is said that any software development project can be classified into one of the following three categories based on the development complexity:

* + - Organic
    - Semidetached
    - Embedded

Roughly speaking, these three product classes correspond to application, utility and system programs, respectively. My project is a well understood web portal and the size of the development team is extremely small i.e. only one person. Due to all these reasons this software is considered to be **ORGANIC**.

The COCOMO model gives an approximate estimate of the project parameters. The COCOMO estimation model is given by the following expressions:

**Effort =** a1 х (KLOC)a2 PM

**Tdev =** b1 x (Effort)b2 Months

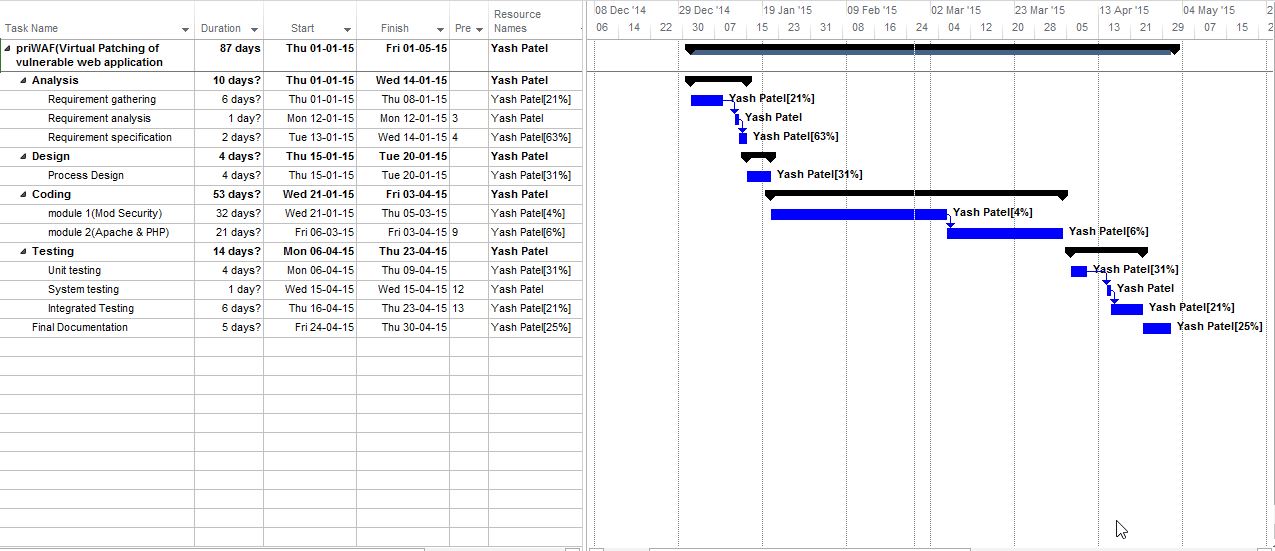
Where

* KLOC is the estimated size of the software product expressed in Kilo Lines of Code,
* a1, a2, b1, b2 are constants for each category of software products,
* Tdev is the estimated time to develop the software, expressed in months,
* Effort is the total effort required to develop the software product, expressed in person months (PMs).

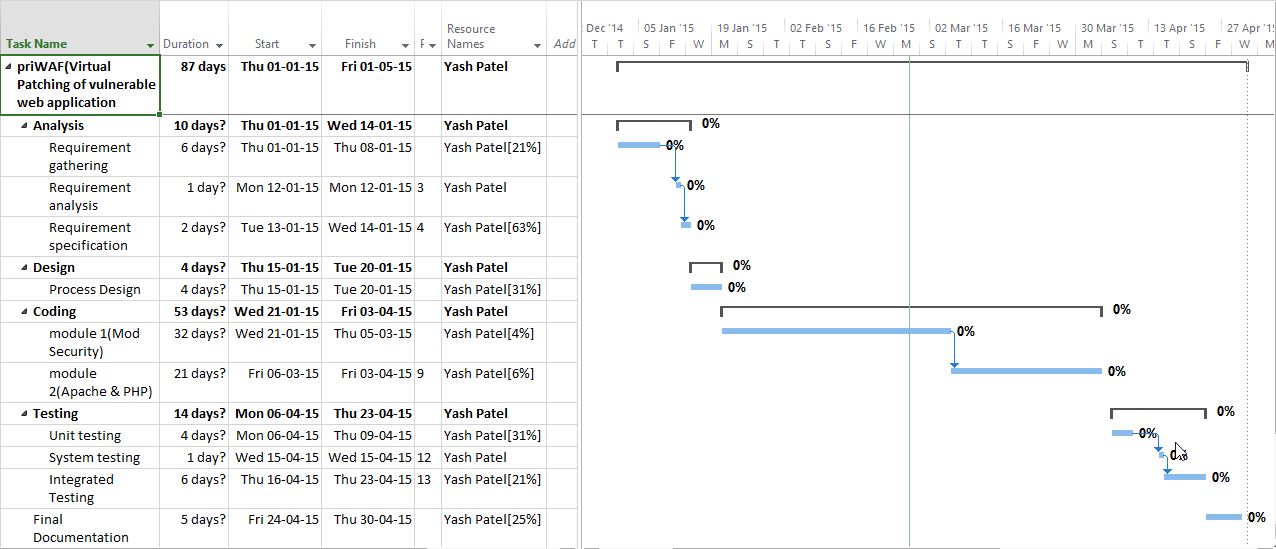
|  |  |
| --- | --- |
| **Phase** | **Time Duration** |
| Feasibility Study | 5 days |
| Requirement Gathering and Analysis | 9 days |
| Design | 4 days |
| Coding | 53 days |
| Testing and Maintenance | 14 days |
| Documentation | 5 days |

**2.3 Project Scheduling (Gantt chart/ Tracking Gantt / Network Chart)**

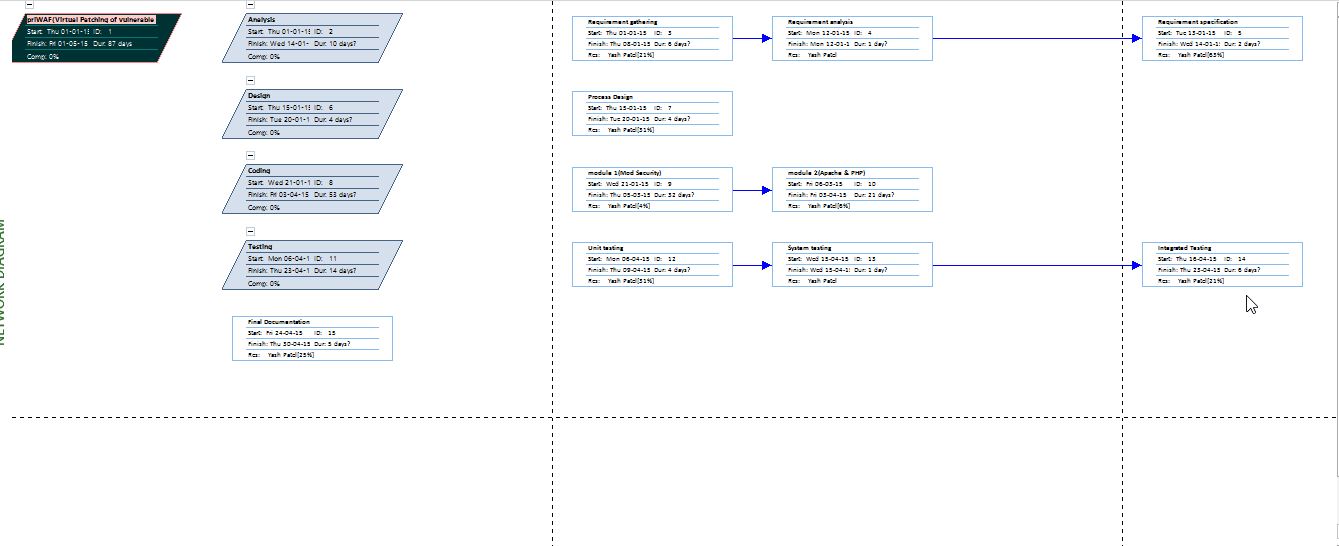
**Gantt chart**



**Tracking Gant Chart**

****

**Network Chart**

****

1. **System Analysis**

**3.1 Users of System**

There are two type of user: **(1) WAF-admin** user **(2) Client** (Server-Admin)

**3.2 Requirements of System**

**3.2.1 Functional Requirement**

There are two type of integration in mod\_security to secure web application

1. Customer side WAF & deployment
2. WAF & deployment using reverse proxy at WAF server side

**In first option** an engineer checks & scans client’s web application and for patching found vulnerabilities. He will configure mod\_security on client’s web-server & handle the WAF remotely.

**In another option** the team will use reverse proxy using proxy server & configure virtual host file on their WAF server.

**3.2.2 Nonfunctional Requirement**

* Client should configure their web- server Apache for a specific mod\_security modules.
* As per clients configuration the WAF portal will be changed with mod\_security.

**3.3 Feasibility Study**

**3.3.1 Technical Feasibility**

* Client can see report of vulnerabilities & attacks on WAF-portal using their unique login.
* Client can see summary on web application issues like errors, bugs, exploitation with attacker’s machine IPs and time stamp also.
* WAF- portal admin can manage any IPs using white list of IPs, he will able to block any Man-in-Middle attacker’s IPs.

**3.3.2 Economical Feasibility**

As it is a software its cost would be more

* + - Will provide Free Scanning
    - The client should be charged based on their web application
    - Charges for issuing SSL certificate from Entrust.

**3.3.3 Operational Feasibility**

* + - We have developing our product & testing on 3 different web application and also developed Custom Rule Sets for that websites, its response is good.
    - There is quiet a possibility that it may support all web servers on different platform. But we cannot be sure about it
  1. **UML Diagrams**













1. **System Design**

**4.1 Data Dictionary**

**Register Client**

**Path: /Register.php**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr.No. | Fields | Data Type | Size | Constraints |
| 1 | username | String | 6 | Not Null |
| 2 | Email id | String | 20 | Not Null |
| 3 | password | String | 10 | Not Null |
| 4 | Confirm password | String | 10 | Not Null |
| 5 | Security question | string | 50 | Not null |
| 6 | Answer | String | 10 | Not null |
| 7 | Captcha answer | String | 10 | Not null |

**Add Client’s Details**

**Path: /detailscustomer.php**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr.No. | Fields | Data Type | Size | Constraints |
| 1 | User id | String | 6 | Not Null |
| 2 | CustomerID | String | 20 | Not Null |
| 3 | Name | String | 10 | Not Null |
| 4 | Mobile Number | String | 10 | Not Null |
| 5 | URL | string | 50 | Not null |
| 6 | Service Date | String | 10 | Not null |
| 7 | Address | String | 10 | Not null |

**Login page**

**Path: /index.php**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr.No. | Fields | Data Type | Size | Constraints |
| 1 | username | string | 6 | Not Null |
| 2 | password | string | 10 | Not Null |

**Dashboard page (for registered clients)**

**Path: /index.php->dashboard.php->export.php**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr.No. | Fields | Data Type | Size | Constraints |
| 1 | CustomerID | String | 2000 | Not Null |
| 2 | AlertID | String | 2000 | Not Null |
| 3 | FoundDate | String | 10 | Not Null |
| 4 | Description | String | 1000 | Not Null |
| 5 | URL | string | 50 | Not null |
| 6 | Method | String | 20 | Not null |
| 7 | Para | String | 10 | Not null |
| 8 | ReqHeader | String | 50 | Not null |
| 9 | Replay\_TIME | String | 10 | Not null |
| 10 | HTTP\_RESPONSE\_CODE | String | 100 | Not null |
| 11 | RawLog | String | 500 | Not null |
| 12 | Unique\_ID | String | 50 | Not null |

**WAFadmin**

**Path: /admin/attacks.php**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr.No. | Fields | Data Type | Size | Constraints |
| 1 | Found Time | String | 2000 | Not Null |
| 2 | Error | String | 2000 | Not Null |
| 3 | Pid | String | 10 | Not Null |
| 4 | Client IP | String | 10 | Not Null |
| 5 | Log Description | string | 5000 | Not null |
| 6 | Pattern | String | 2000 | Not null |
| 7 | File | String | 1000 | Not null |
| 8 | Line | String | 50 | Not null |
| 9 | Rule ID | String | 10 | Not null |
| 10 | Message | String | 100 | Not null |
| 11 | URI | String | 500 | Not null |
| 12 | Unique\_ID | String | 500 | Not null |

**WAFadmin**

**Path: /admin/rules.php**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr.No. | Fields | Data Type | Size | Constraints |
| 1 | CustomerID | String | 2000 | Not Null |
| 2 | IP | String | 2000 | Not Null |
| 3 | URL | String | 10 | Not Null |
| 4 | RulesConfig | String | 1000 | Not Null |

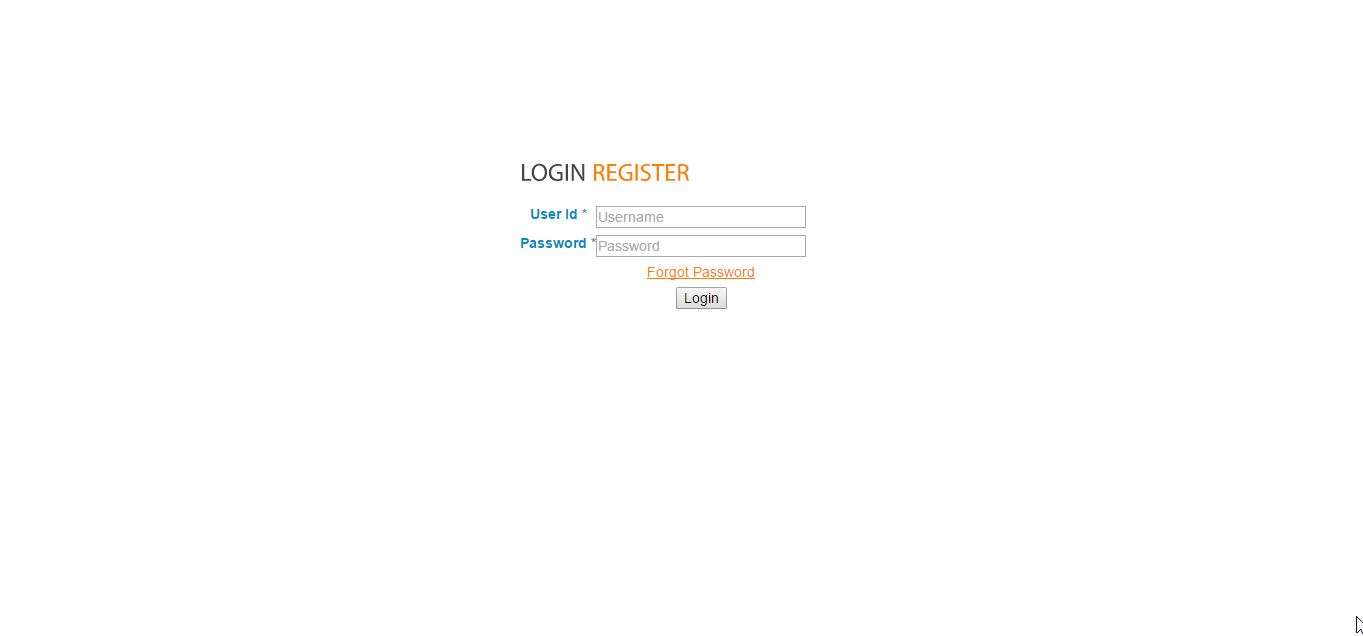
**4.2 Table Relationship**

* There are two different tables and in between them a “CustomerID” common key to communicate with each other.
* For userdetails & user both different tables the only one userid is the key to communicate with each other.
* For userdetails & rules the key for communication is customer ID.
  1. **User Interface Design**
* We have used php/html as an up design part
* Different search fields for client’s report
* Client can easily export their URLs vulnerability & patching report
* WAF admin has every right to manage the whole portal at backend side and client side also.
* WAF admin has to setup Core Rules & Custom Rules on WAF using this portal

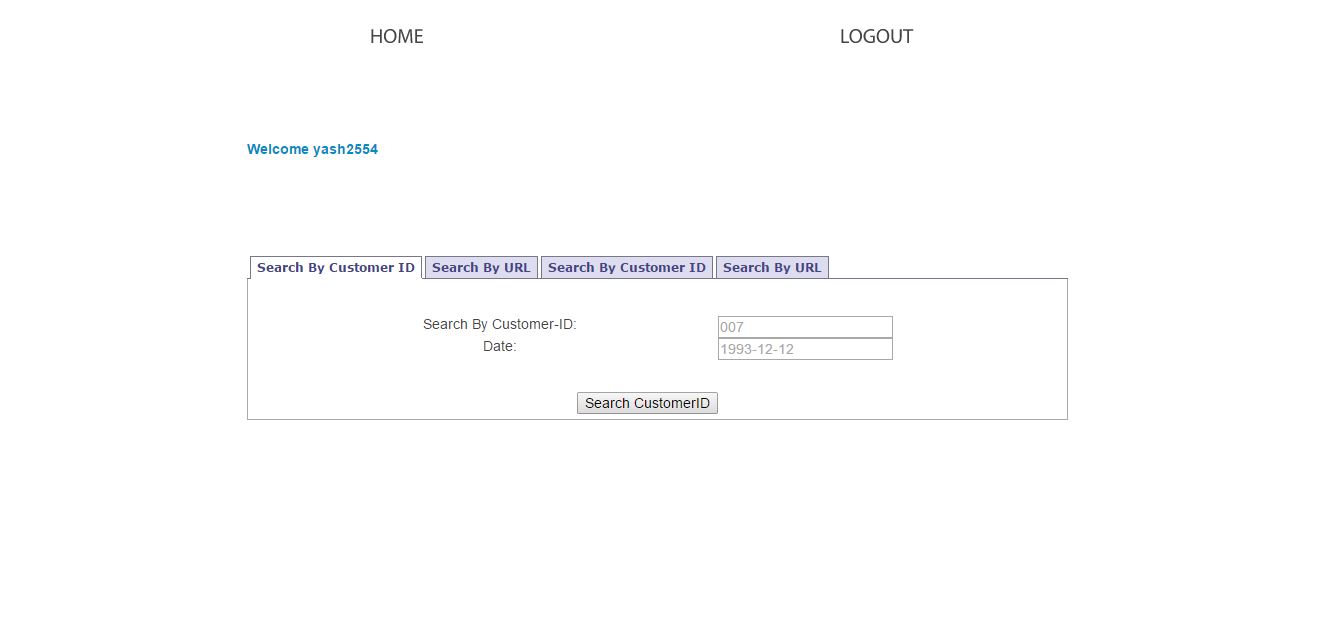
1. **Implementation & Testing**

**5.1 GUI of Form and Reports**

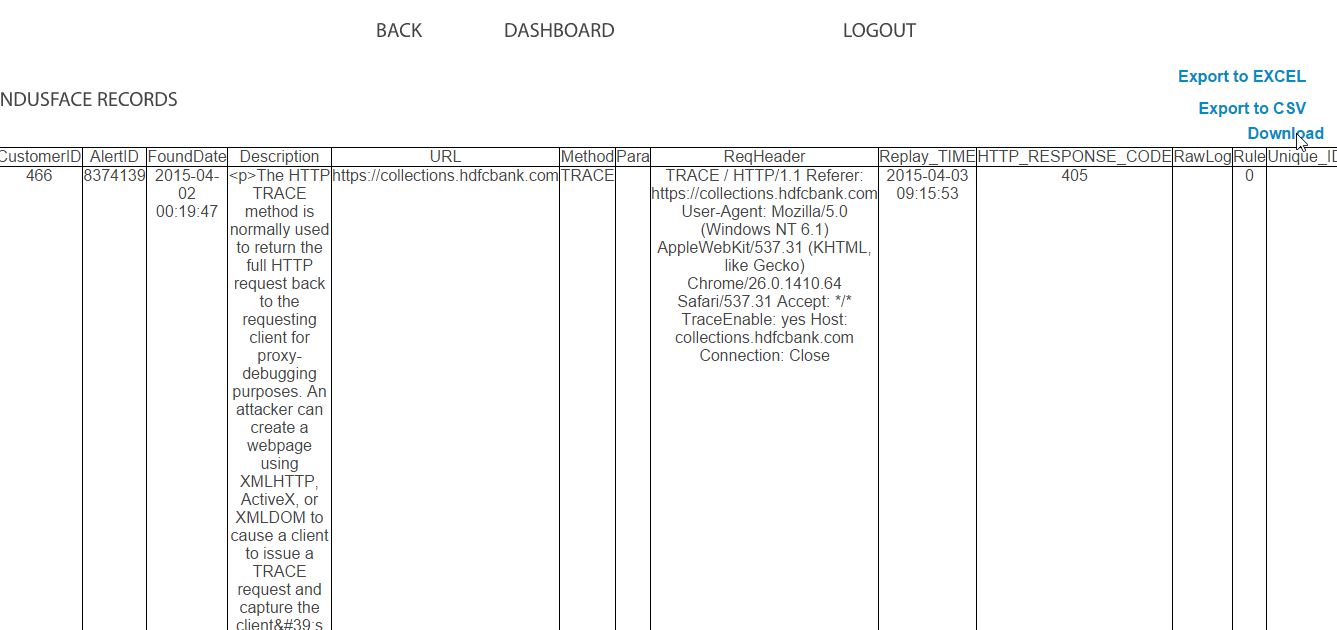
**Login page**



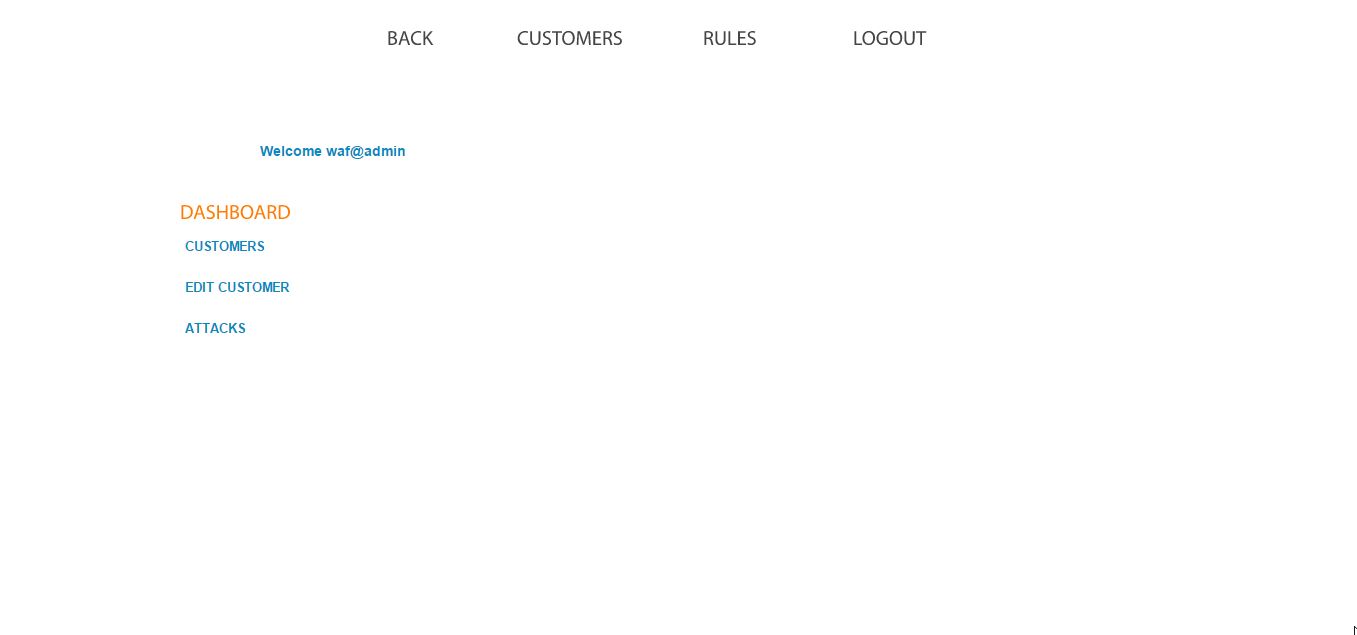
**Client Login**

****

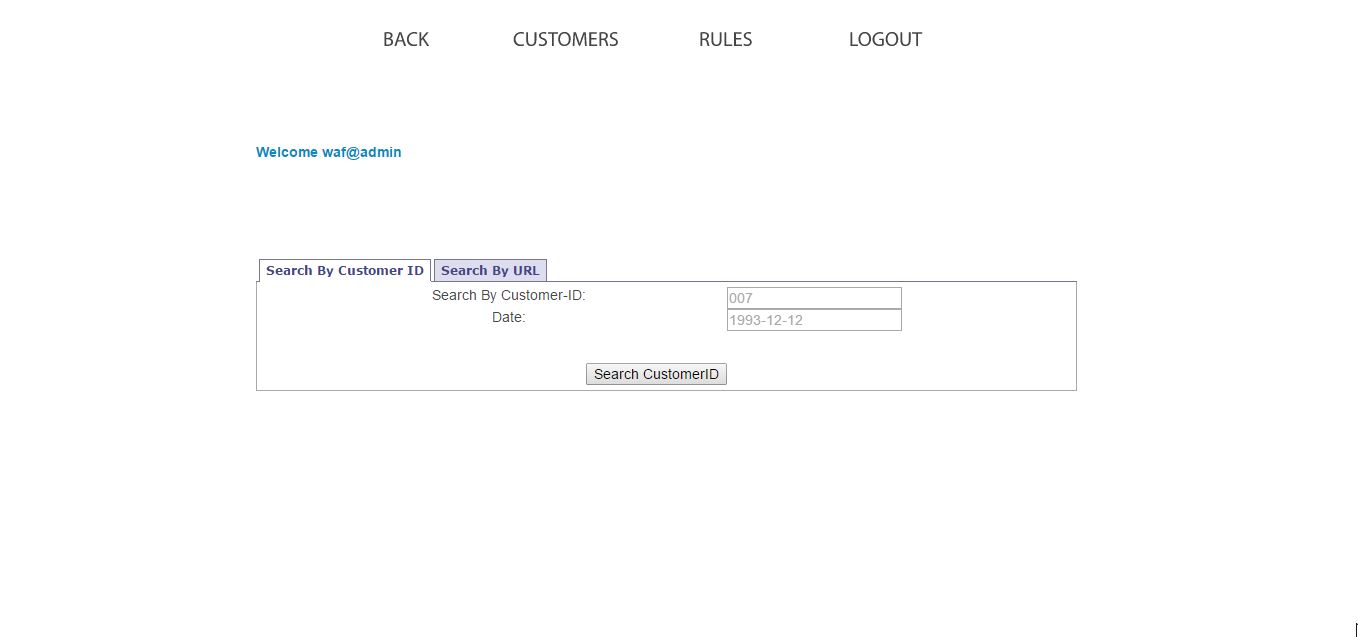
**Vulnerability Report**

****

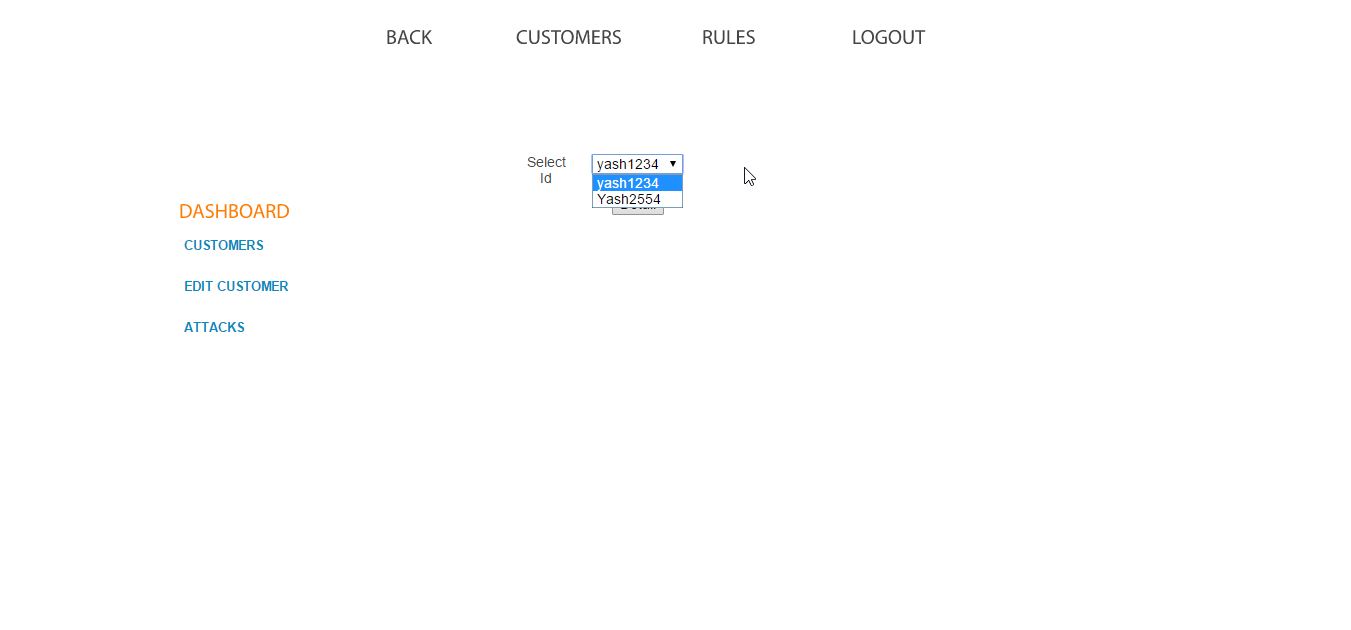
**priWAF Admin Home**

****

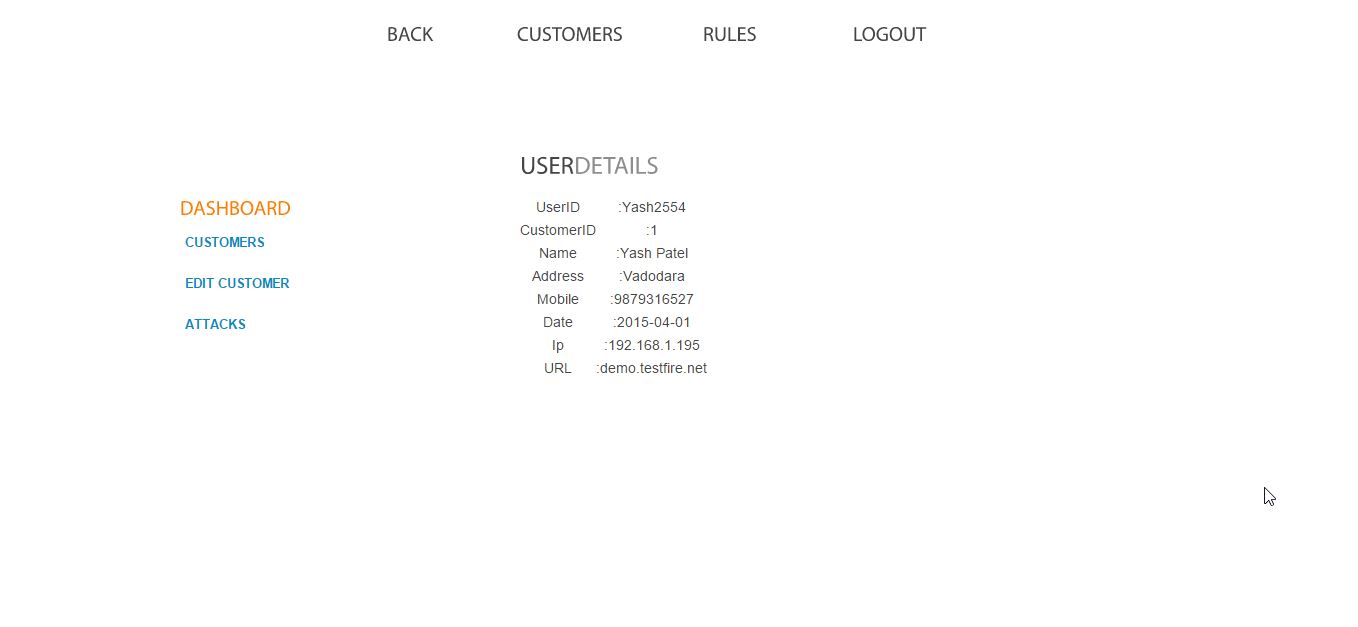
**Customer Vulnerability Reports**

****

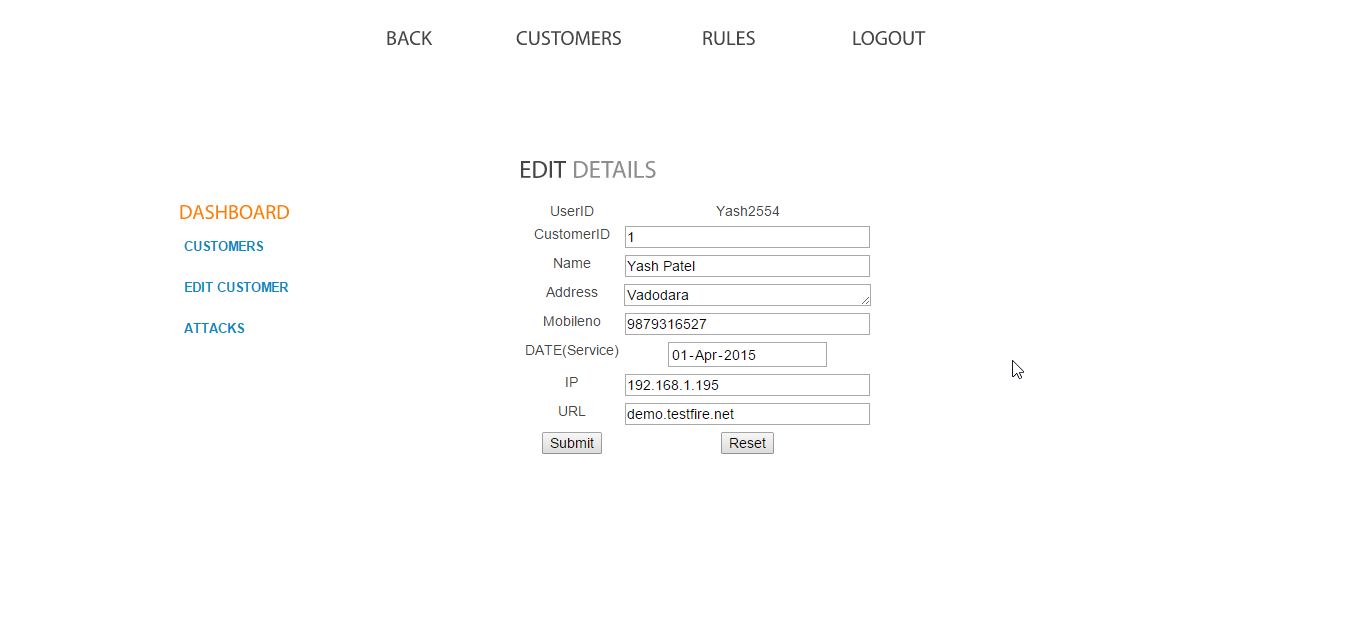
**Maintain Customer details**

****

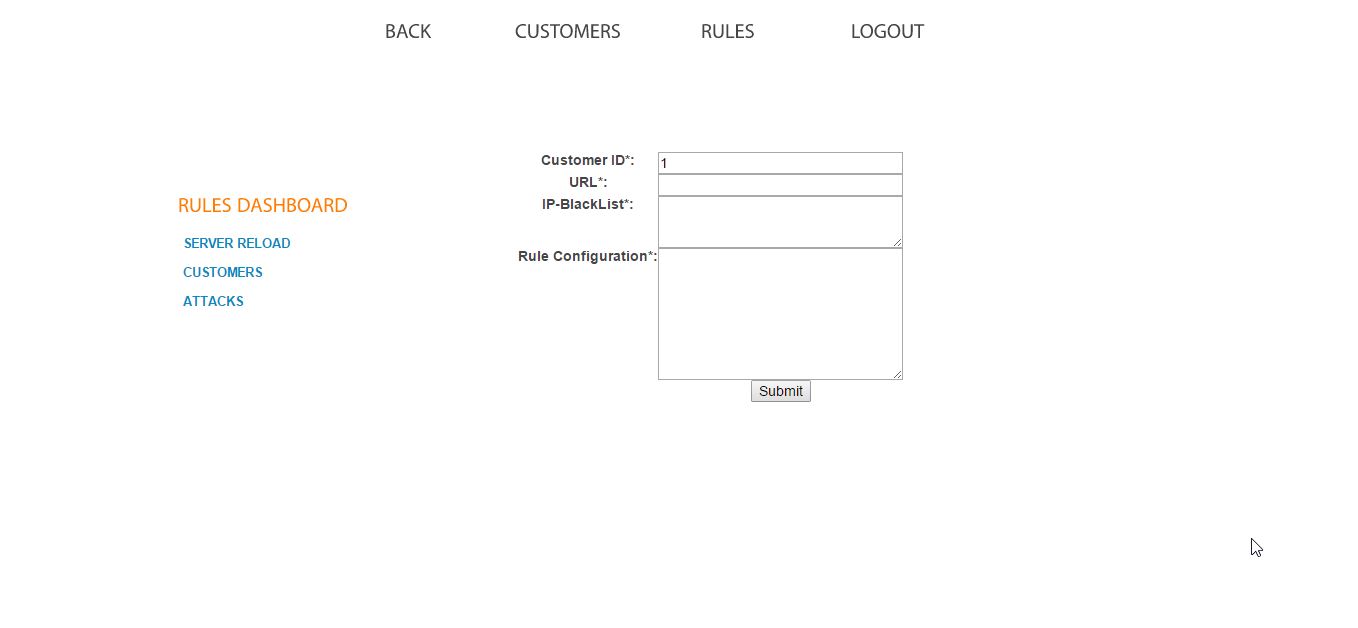
**Added Details**

****

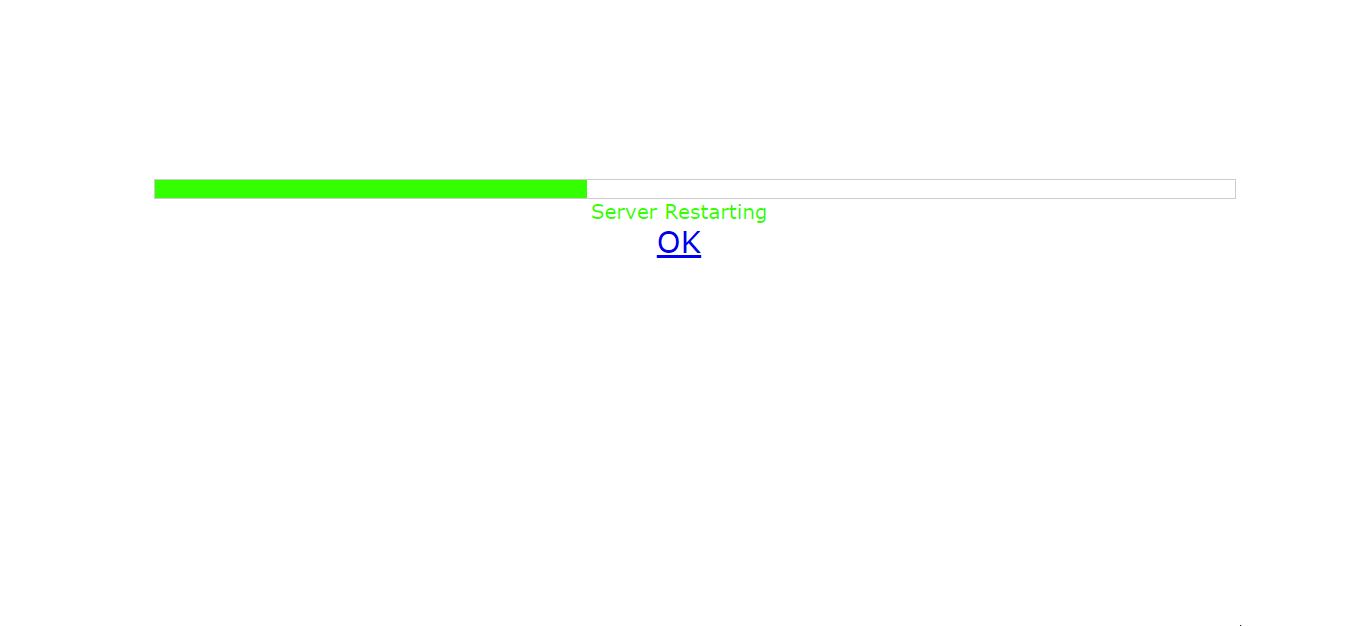
**Edit Details**

****

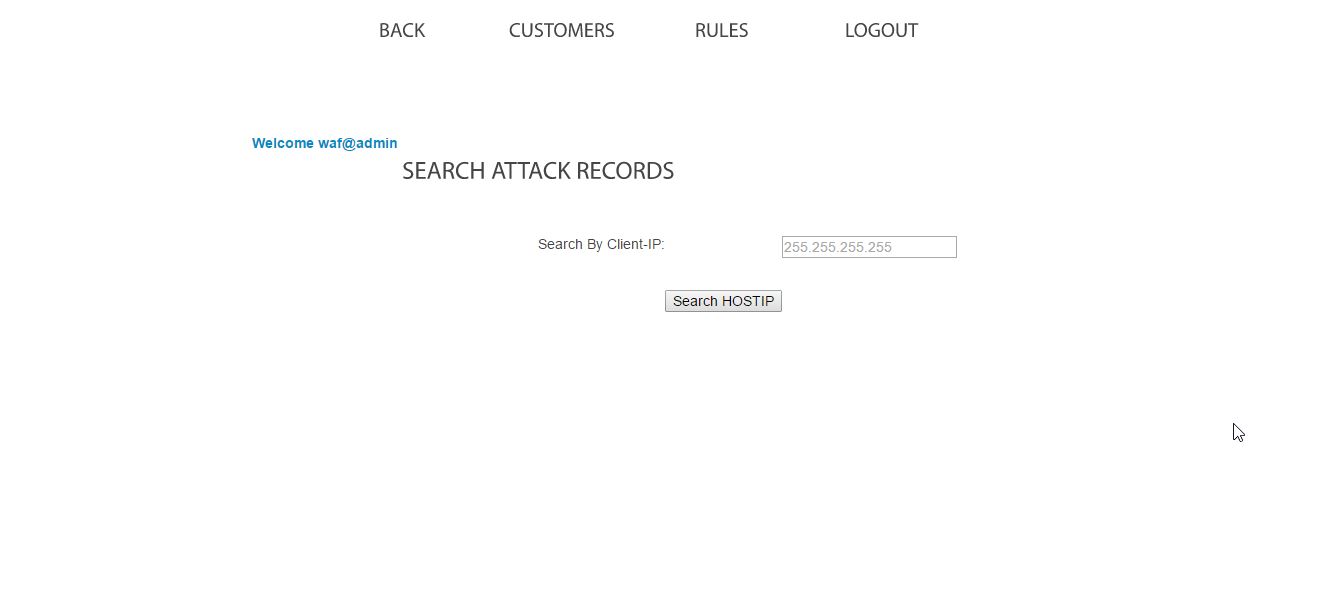
**Push Rule-Configuration File to Apache**

****

**Reload Apache Server**

****

**Error-Log Search By clients IP**

****

**5.2 Test Cases of System**

**Test Suites No: 1**

Test Suite Detail: For Registration of client

Add details to register client to access client’s dashboard

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Function Name** | **Input** | **Expected Output** | **Actual Output** | **Pass/Fail** |
| **1** | Register Client’s Userid | User id:  Yash1234 | Next Register user’s userid To Add details | Saved successfully | pass |
| **2** | Enter client’s email id | Email:  [Yash2554@gmail.com](mailto:Yash2554@gmail.com) | Next Register user’s emailid To Add details | Saved successfully | pass |
| **3** | Enter Client’s  Password | Password:  \*\*\*\*\*\*\*\*\*\* | Next Register user’s password To Add details | Saved successfully | Pass |
| **4** | Security Question | Question:  My first name? | Next Register user’s question To Add details | Saved successfully | pass |
| **5** | Answer | Answer:  Patel | Next Register user’s answer To Add details | Saved successfully | pass |

**Test Suites No: 2**

Test Suite Detail: Maintain Clients Details by WAF admin

Valid Customer Id, Name, Address, Contact number, Date for Service, IP, URL.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Function Name** | **Input** | **Expected Output** | **Actual Output** | **Pass/Fail** |
| 1 | Customer ID | 1 | Saved to database  Next add other details | Saved Successfully | Pass |
| 2 | Name | Yash Patel | Saved to database  Next add other details | Saved Successfully | Pass |
| 3 | Address | Valid - Address | Saved to database  Next add other details | Saved Successfully | pass |
| 4 | Contact Details | Mobile number:  9879316527 | Saved to database  Next add other details | Saved Successfully | Pass |
| 5 | Date | 1-April, 2015 | Saved to database  Next add other details | Saved Successfully | Pass |
| 6 | IP | Internal IP:  192.168.1.199 | Saved to database  Next add other details | Saved Successfully | Pass |
| 7 | URL | Demo.testfire.net | Saved to database  Next add other details | Saved Successfully | pass |

**Test Suites No: 3**

Test Suite Detail:

Add rules for customer

Add rules Configuration file assign a valid URL to proxy pass & IP list to black list those IPs for a specific URL/IPs.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Function Name** | **Input** | **Expected Output** | **Actual Output** | **Pass/Fail** |
| 1 | Customer Id | 1 | Saved to database  Next add other details | Saved Successfully | Pass |
| 2 | URL | Demo.testfire.net | Saved to database  Next add other details | Saved Successfully | pass |
| 3 | Blacklist IP | 192.168.1.100 | Saved to database  Next add other details | Saved successfully | pass |
| 4 | Rule | Rule configuration file | Saved to database  Next add other details | Saved successfully | pass |

**6.0 Conclusion, Limitation & Future Enhancement**

**Conclusion:**

 PriWAF is the box by which we can secure any website using reverse proxy to their IPs

Provide them more protection against Malicious & harmful attacks which are listed on OWASP 10 list of vulnerabilities

**Limitation:**

 maintain database of whole box is too difficult when number of clients are there to be serve for.

**Future Enhancement**

 Platform independent whatever web server or database server client using & what version they are using is not important.

we should configured network settings for reverse proxy & assign them one unique port that will assign to them on public IP | private IP for use mod security.