4. SYSTEM ANALYSIS

4.1 STUDY OF CURRENT SYSTEM

The current system being used in the cyber world is to configure the manual system of WAF on client's web server. For the purpose of implement WAF of the client in between web server and end users, the priWAF admin has to collect the information and necessary details of Core rule sets & Custom rule sets for clients and manage configuration them manually and arrange them according to Customer Id.

The main features of the current system being used are as follows:

- The priWAF admin collects, manages and manipulates the scanner's reports of clients for their provide WAF manually on their web server.
- The priWAF admin collects the information of various vulnerabilities and notifies the clients time to time about the same.
- The priWAF collects and arranges details of clients according to Customer Id and notifies them according to attack reports of daily scan & virtual patch of those attacks.
- If any modifications or corrections are required in the detail of any client, it has to be selected and modified manually.
- A client requiring services of priWAF manually meets the priWAF admin in person to register his/her name.
- The priWAF admin maintains multiple records related to the various WAF configuration like whitelisting client's machine IP, registration of client.re-configure the mod security module for the specific client's URLs /IPs, setting up proxy server using reverse proxy on host configuration , maintain core rule sets and custom rule sets for the specific URLs/IPs.

What is WAF?

Over the past few years, a clear trend has emerged within the information security landscape; web applications are under attack. "Web applications continue to be a prime vector of attack for criminals, and the trend shows no sign of abating; attackers increasingly shun network attacks for cross-site scripting, SQL injection, and many other infiltration techniques aimed at the application layer." (Sarwate, 2008) Web application vulnerabilities can be attributed to many things including poor input validation, insecure session management, improperly configured system settings and flaws in operating systems and web server software. Certainly writing secure code is the most effective method for minimizing web application vulnerabilities. However, writing secure code is much easier said than done and involves several key issues. First of all, many organizations do not have the staff or budget required to do full code reviews in order to catch errors. Second, pressure to deliver web applications quickly can cause errors and encourage less secure development practices. Third, while products used to analyze web applications are getting better, there is still a large portion of the job that must be done manually and is susceptible to human error. Securing an organization's web infrastructure takes a defense in depth approach and must include input from various areas of IT including the web development, operations, infrastructure, and security teams.

One technology that can help in the security of a web application infrastructure is a web application firewall. A web application firewall (WAF) is an appliance or server application that watches http/https conversations between a client browser and web server at layer 7. The WAF then has the ability to enforce security policies based upon a variety of criteria including signatures of known attacks, protocol standards and anomalous application traffic.

I. Web Application Firewall Architecture

WAF Placement

Appliance-based WAF deployments typically sit directly behind an enterprise firewall and in front of organizational web servers. Deployments are often done in-line with all traffic flowing through the web application firewall. However, some solutions can be "out of band" with the use of a network monitoring port. If network based deployments

are not preferred, organizations have another option. Host or server based WAF applications are installed directly onto corporate web servers and provide similar feature sets by processing traffic before it reaches the web server or application.

Security Model

A WAF typically follows either a positive or negative security model when it comes to developing security policies for your applications. A positive security model only allows traffic to pass which is known to be good, all other traffic is blocked. A negative security model allows all traffic and attempts to block that which is malicious. Some WAF implementations attempt to use both models, but generally products use one or the other. "A WAF using a positive security model typically requires more configuration and tuning, while a WAF with a negative security model will rely more on behavioral learning capabilities." (Young, 2008)

Operating Modes

Web Application Firewalls can operate in several distinct modes. Vendor names and support for different modes vary, so check each product for specific details if a particular mode is desired. Each mode offers various pros and cons which require organizations to evaluate the correct fit for their organization.

- **Reverse Proxy –** The full reverse proxy mode is the most common and feature rich deployment in the web application firewall space. While in reverse proxy mode a device sits in line and all network traffic passes through the WAF. The WAF has published IP addresses and all incoming connections terminate at these addresses. The WAF then makes requests to back end web servers on behalf of the originating browser. This mode is often required for many of the additional features that a WAF may provide due to the requirement for connection termination. The downside of a reverse proxy mode is that it can increase latency which could create problems for less forgiving applications.
- **Transparent Proxy –** When used as a transparent proxy, the WAF sits in line between the firewall and web server and acts similar to a reverse proxy but does not have an IP address. This mode does not require any changes to the existing

infrastructure, but cannot provide some of the additional services a reverse proxy can.

- **Layer 2 Bridge –** The WAF sits in line between the firewall and web servers and acts just like a layer 2 switch. This mode provides high performance and no significant network changes, however does not provide the advanced services other WAF modes may provide.
- **Network Monitor/Out of Band –** In this mode, the WAF is not in line and watches network traffic by sniffing from a monitoring port. This mode is ideal for testing a WAF in your environment without impacting traffic. If desired, the WAF can still block traffic in this mode by sending TCP resets to interrupt unwanted traffic.
- **Host/Server Based -** Host or server based WAFs are software applications which are installed on web servers themselves. Host based WAFs do not provide the additional features which their network based counterparts may provide. They do, however, have the advantage of removing a possible point of failure which network based WAFs introduce. Host based WAFs do increase load on web servers so organizations should be careful when introducing these applications on heavily used servers.

Additional Features

WAF appliances are often either add-on components of existing application delivery controllers or include additional features to improve the reliability and performance of web applications. These additional features can help make the case for implementing a WAF for organizations not already taking advantage of such features. Not all WAF solutions have these features and many are dependent upon the deployment mode chosen. Typically a reverse-proxy deployment will support each of these features.

- **Caching –** Reducing load on web servers and increasing performance by caching copies of regularly requested web content on the WAF thus reducing repeated requests to back end servers.
- **Compression –** In order to provide for more efficient network transport, certain web content can be automatically compressed by the WAF and then decompressed by the browser.

- **SSL Acceleration –** Use of hardware based SSL decryption in a WAF to speed SSL processing and reduce the burden on back-end web servers.
- **Load Balancing –** Spreading incoming web requests across multiple back end web servers to improve performance and reliability.
- **Connection Pooling –** Reduces back end server TCP overhead by allowing multiple requests to use the same back end connection.

Non-Commercial Web Application Firewalls

ModSecurity - www.modsecurity.org

The most widely used web application firewall is the open source project ModSecurity. ModSecurity is currently maintained by Breach Security, a company who also sells commercial appliances. "ModSecurity is a web application firewall that can work either embedded or as a reverse proxy. It provides protection from a range of attacks against web applications and allows for HTTP traffic monitoring, logging and real-time analysis" (ModSecurity, 2008). ModSecurity generally uses a negative security model and also has several related projects which help enhance the solution. ModSecurity for Apache is a module for Apache containing ModSecurity. The ModSecurity Core Rules are a collection of rules that will detect the most common web attacks. The core rules are a great starting point for those new to ModSecurity. The ModSecurity console creates a centralized console for individual ModSecurity instances to report logs and alerts to. The ModSecurity profiler analyzes web application traffic and creates application profiles which can then be used to implement a positive security model.

Microsoft URLScan - http://learn.iis.net/page.aspx/473/using-urlscan

Another free option exists for Microsoft IIS called URLScan. "UrlScan v3.1 is an ISAPI filter that reads configuration from an urlscan.ini file and restricts certain types of requests (enumerated in urlscan.ini) from being executed by IIS." (IISteam, 2008) URLScan is a solution that Microsoft IIS shops should strongly consider even if they have a separate web application firewall. First of all; it's free, and second, the application runs on the web server itself and, therefore, can be used as yet another layer of defense.

4.2 PROBLEMS AND WEAKNESSES OF CURRENT SYSTEM

The current system is suffering from the following cons:

- The most important and main drawback of the current system is that it is very Time Consuming. It takes a lot of time for the priWAF admin to collect and manage the details of the clients.
- The manual handling of large data becomes cumbersome and the complexity increases with the increase in the number of clients.
- Analysis of the past attack records and the details of the previous scan report of client URL becomes complex and inefficient.
- It becomes very difficult for the priWAF to maintain the records and preserve them for future reference.
- This system is highly error prone.
- It becomes too inaccurate to correctly predict any information related to the current attack reports and scan reports by analyzing the past records.

4.3 REQUIREMENTS OF NEW SYSTEM

4.3.1 Functional Requirements

The Functional requirements of the new system are as follows:

R1. Client Registration

A client should be able to register himself/herself to the portal by whitelisting by WAF portal admin. The WAF should only activate a client if he/she has register himself/herself by giving personal details about client.

R2. Storage of Client's Details

A WAF admin once logged in should be able to store client's personal details as well as Web application details and also should be able to update the same as and when required.

R3. Configuration to the client's URLs

A WAF admin should be able to apply to a core rule sets for specific client's URLs. Thus, the procedure should be managed on the portal itself in its whole.

The portal is really helpful for signature development & network configuration team to managed WAF (web application firewall) to the specific client's URLs.

R4. Statistical data storage

The portal should store statistical data of multiple client's details, daily scanning reports, virtual patching of those vulnerabilities which will be verified by daily scan, Custom rule sets for a specific requirement of any client, Blacklist IPs which will be managed by WAF to secure by man in middle attack of unauthenticated machine.

R5. Customer Id Allocation

The WAF admin should be able to allocate a unique Customer Id to Client for their project and that is not editable by client himself/herself according to the particular web application.

R5. Update the Signature rule set Management

The WAF admin should be able to add and delete (if required) the signature rule sets which will be developed by WAF signature development team.

R6. WAF (web application firewall) Handling

WAF admin only one who can monitor client's URLs, he is able to manages daily scan on the URLs to find out many more vulnerabilities on URLs this all procedures should be handled by the WAF admin online through the portal.

4.3.2 Non Functional Requirements

The Non Functional requirements of the new system are as follows:

1. **Graphical User Interface**

The user interface of the portal should be compatible to any browser such as internet explorer, Mozilla Firefox, Google chrome etc through which the user can access and communicate with the system.

2. **Hardware Interface**

   As the portal runs on the internet exclusively, the hardware interface required to connect the machine to the internet includes the hardware interface for the system such as WLAN, LAN or a similar internet facility.

3. **Software Interface**

   As the portal requires to be hosted on a web server, it requires the scripting language PHP. The portal also requires MYSQL Database server for storing the user as well as system data. It also needs a DNS (domain name space) for naming the ip address of the portal. At last, the user or a visitor requires a web browser to interact with the system.

4. **Communication Interface**

   The portal running online shall use HTTP/HTTPs protocol for communication over the internet and FTP protocol if required for data handling.

5. **Performance Requirement**

   The portal shall be based on web and has to be run from a web server.

   The portal shall take initial load time depending on internet connection strength which also depends on the media from which the user is running the system. Hence, the performance shall depend upon hardware components of the user.

4.4 FEASIBILITY STUDY

4.4.1 Operational Feasibility

This portal can be used by any novice user because of its user friendly user interface. It will be very helpful to the WAF admin as well as the clients as it provides significant and

important information to the clients and the administrative tasks can also be carried out in a very easy way by the WAF admin.

4.4.2 Technical Feasibility

This project development requires detail knowledge of OWASP top vulnerabilities, attacks which will be following on application layer, OSI model, Apache mod Security, Perl Compatible REGEX writing, Protocol Phases, Network Configuration to setting up apache with mod security, proxy server, manage whole WAF portal on same web server.
This project Development requires detail knowledge of PHP and MYSQL. Apart from this, some knowledge of CSS and Javascript is also required.

4.4.3 Time Schedule Feasibility

Specific time was allotted for the completion of this project and it is important to follow the deadline for any project. This project was completed within the given time period and hence satisfies the time schedule feasibility.

4.4.4 Economic Feasibility

Economic feasibility looks at the financial aspects of the project .i.e. whether it is economically feasible to develop such a system.
This project is developed using the open source technology so no technological cost is associated with its development. Besides the normal web hosting costs, this project is completely cost effective to use and manage.
Also, as it developed by a final semester student, no hidden development cost is associated with the project.

4.5 ACTIVITY/PROCESS IN NEW SYSTEM

Table 4.5 Activity in System

EVENT	TRIGGER	ACTIVITY	RESPONSE
Client wants to Register to the portal.	Click on Register Button	Show a form to enter the necessary details of the new user.	The client has been registered to use the system.
Client wants to login to the website	Navigate to the Login form on the home page.	Show a form to enter the login credentials.	The client is logged in to the system.
WAF admin wants to see client's details	Click on customers details button in navigation.	Display a form to display appropriate details,	The details have been saved.
WAF admin wants to edit/update client's details	Click on Edit customer details buttonin navigation.	Display a form to edit the appropriate details,	The edited fields are updated and the new details saved,
WAF admin wants to setup WAF for the client	Click on rules button in Header.	Display a form to edit the required details,	The edited fields are updated and the new details saved,
WAF admin wants to apply the core rule set on proxy server & reload the apache with mod security module to a client's URL.	Click on Server Reload! button	Display a pop up to confirm WAF action,	The client's URLs has applied to the proxy pass to the mod security module of apache successfully.
WAF admin wants to allocate customer id to a client	Click on Edit Customer & Customer Id input Field.	Display a form to select a particular client & enter customer id.	Customer Id allocated to client & saved in client's details.
TPO want to know Client's daily scan report	Click on Customers tab button from header	Display a table showing the client's details after entering customer id.	Daily Scan report display for specific user by customer id

4.6 FEATURES OF NEW SYSTEM

The main features of the new system are:

1. Online management of WAF (web application firewall) to setting up proxy server for any client's web application.
2. Improvement in the accuracy and performance of the overall procedure undertaken during WAF configuration.
3. Easier storage and navigation of data and other details of Client and their vulnerabilities.
4. User friendly interface having quick authenticated access to reports (daily scanning).
5. Efficient management of resources.
6. Easily scalable to grow with changing system requirement.
7. A cost Effective way of doing the manual processes done in the existing system.
8. Production of Accurate output

4.6.1 Actual Architecture of WAF with priWAF portal implemented for client's Web Server

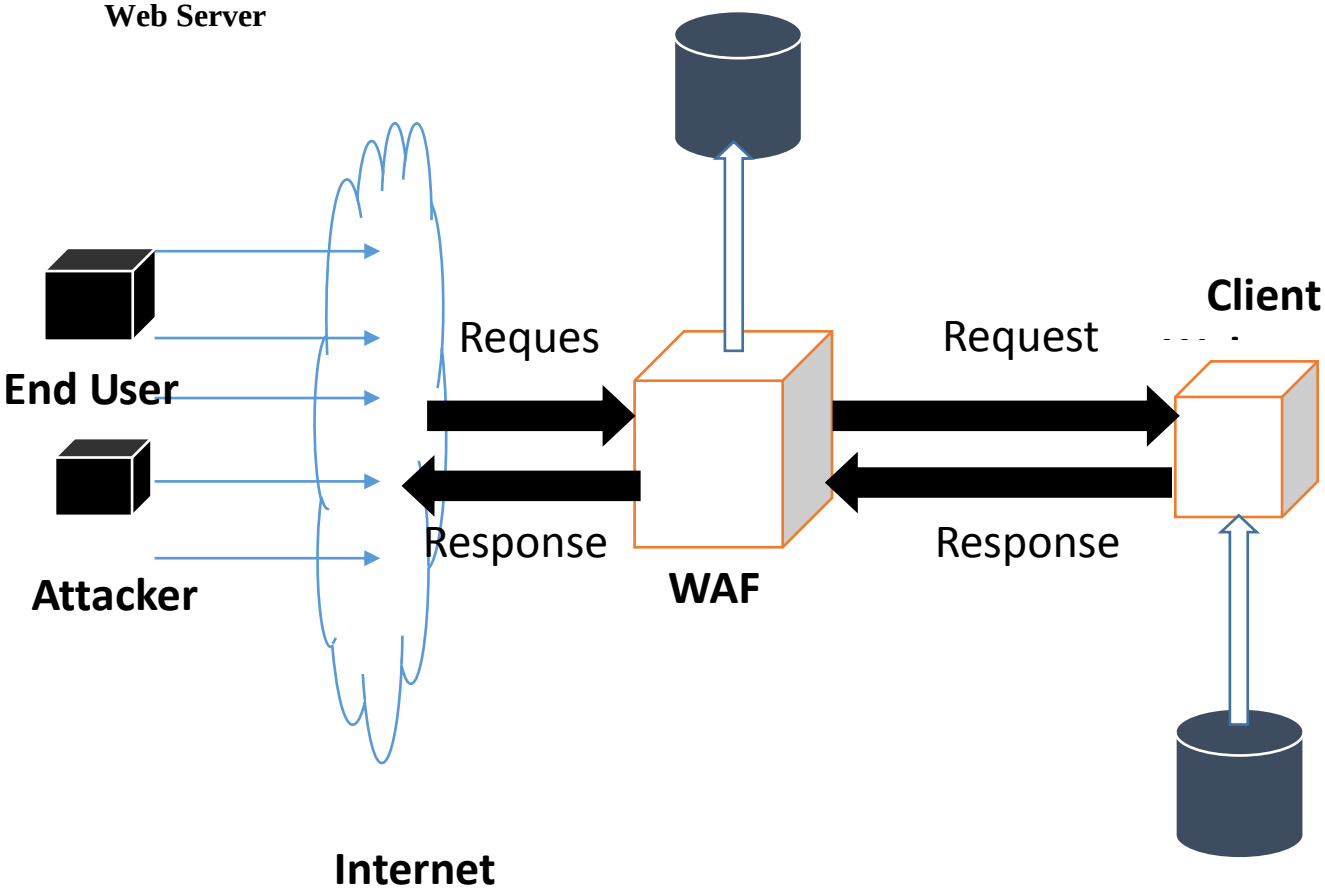


Fig 4.6.1 Architecture of WAF without load balancer

To Handle Number of requests of client's web server have to introduce one more architecture of WAF implemented for client's web server

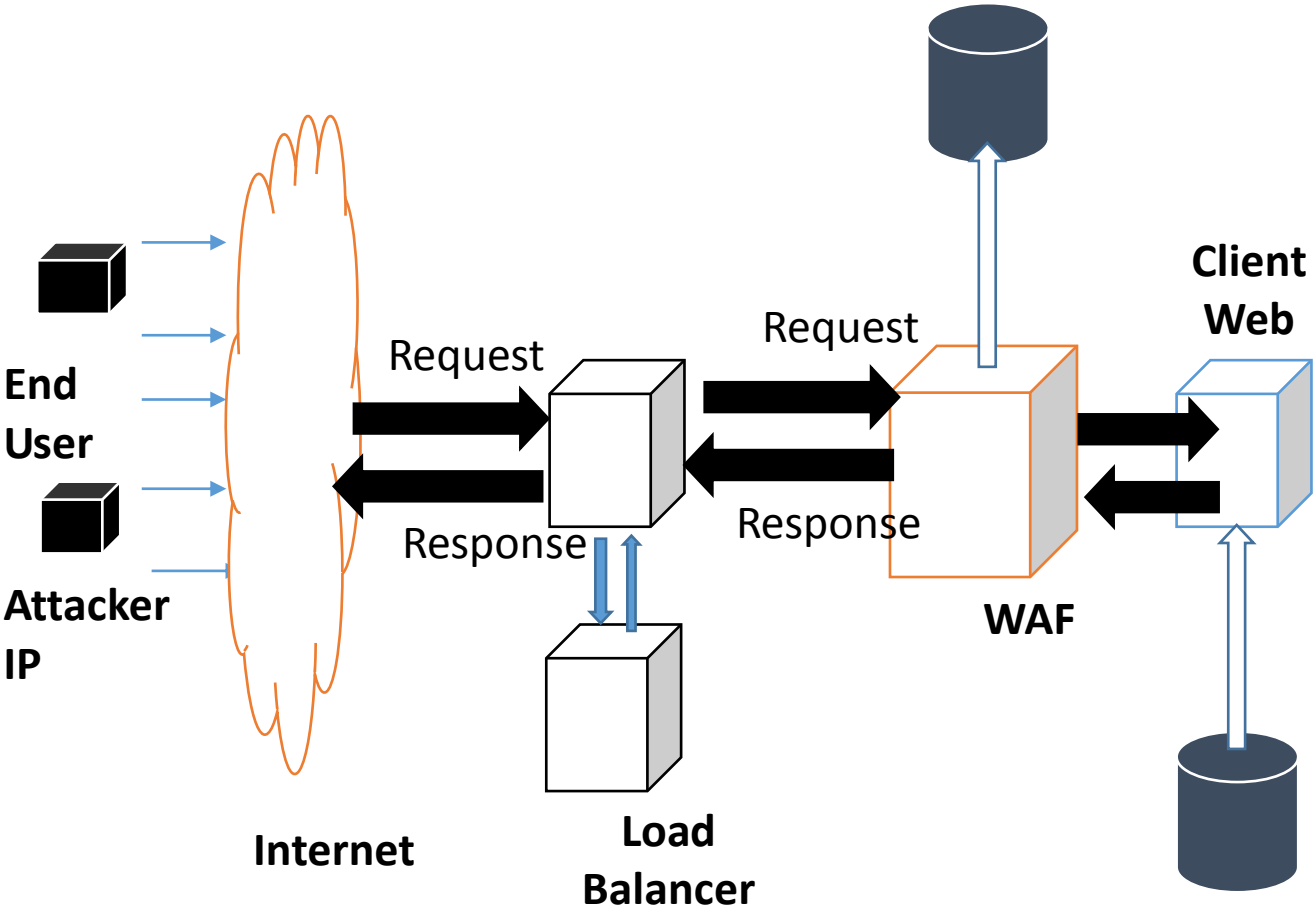


Fig 4.6.2 Architecture of WAF with load balancer

4.7 DATA FLOW DIAGRAMS

- **0 Level DFD(Context Diagram)**

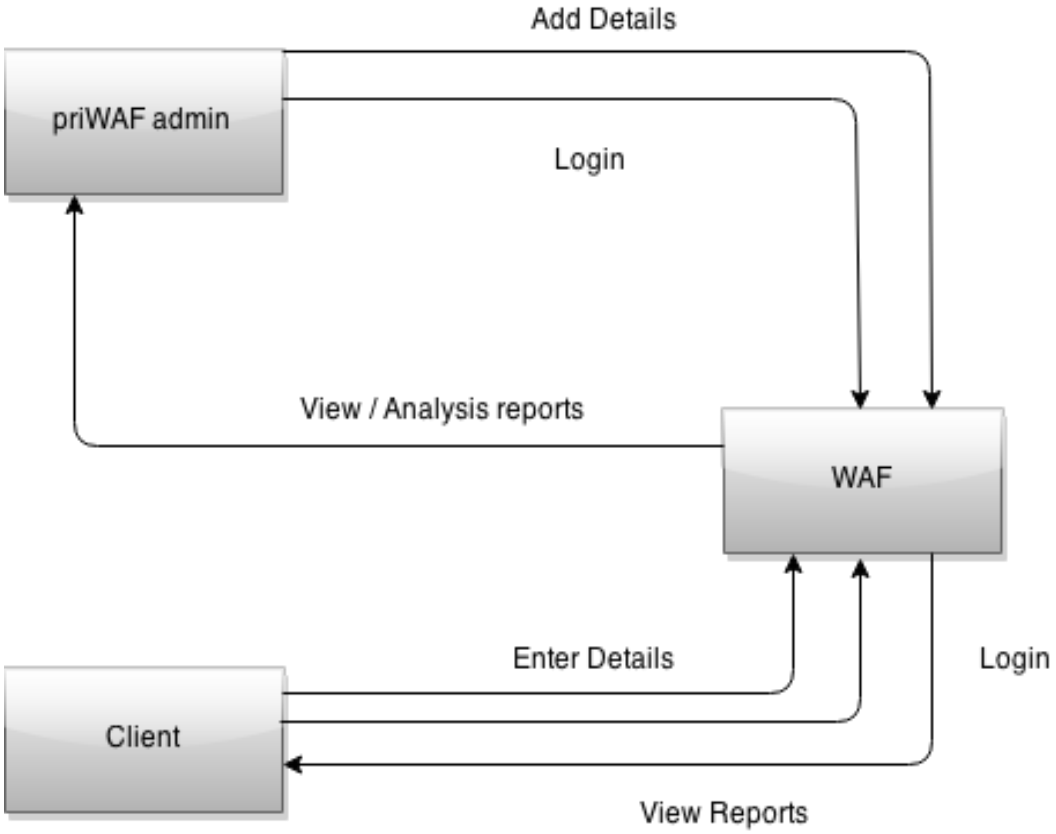


Fig 4.7.1 0 Level DFD

- **Level 1 DFD**

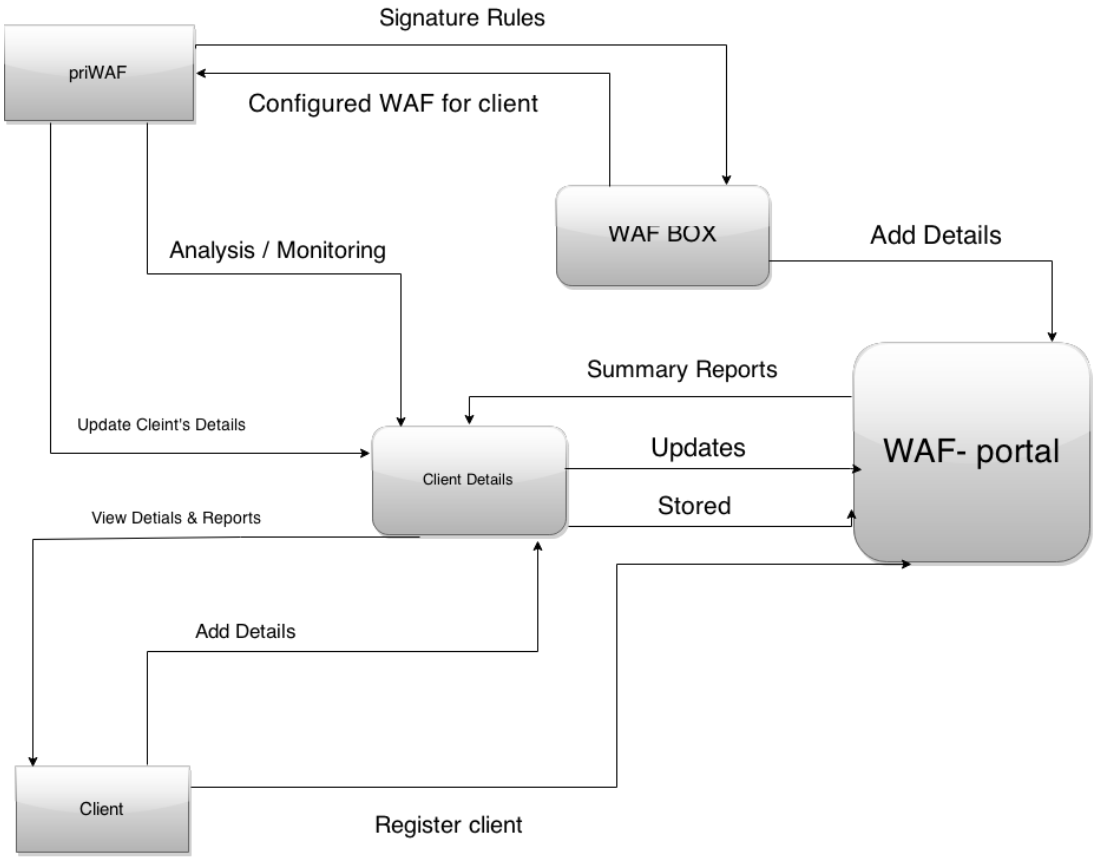


Fig 4.7.2 DFD Level 1

- **Level 2 DFD**

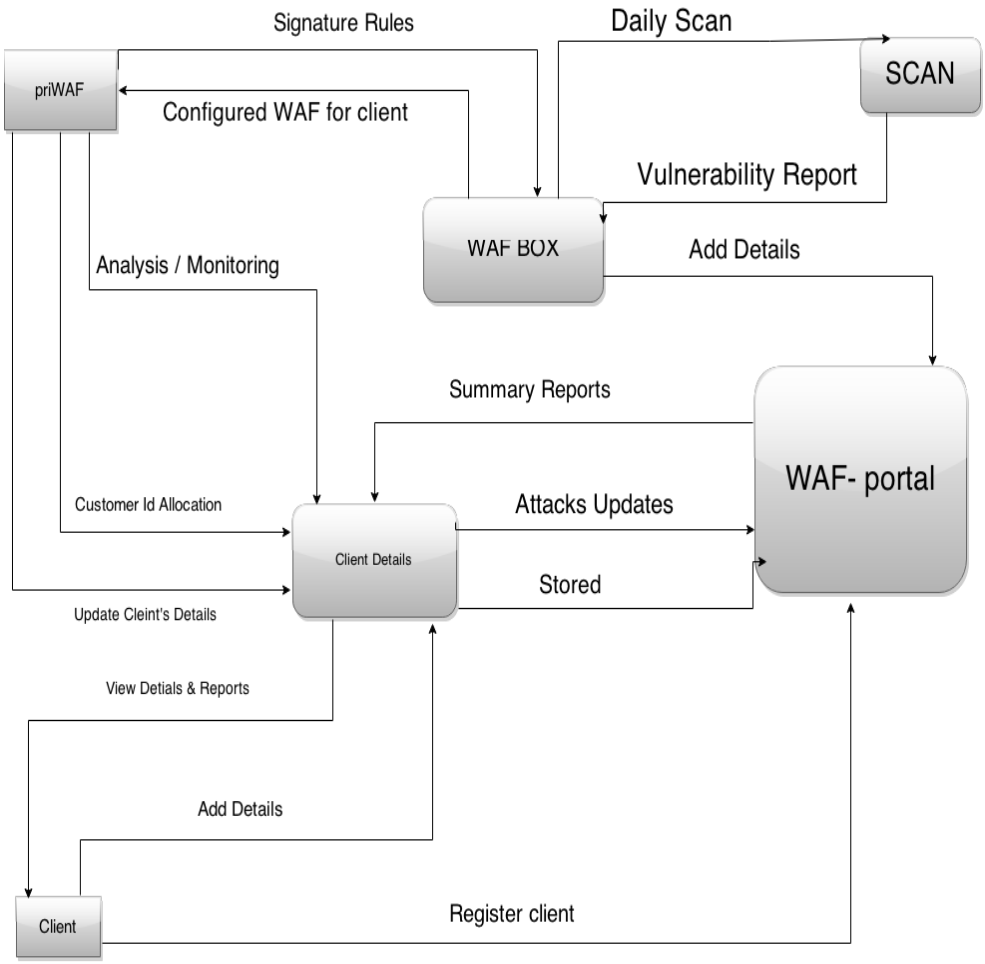
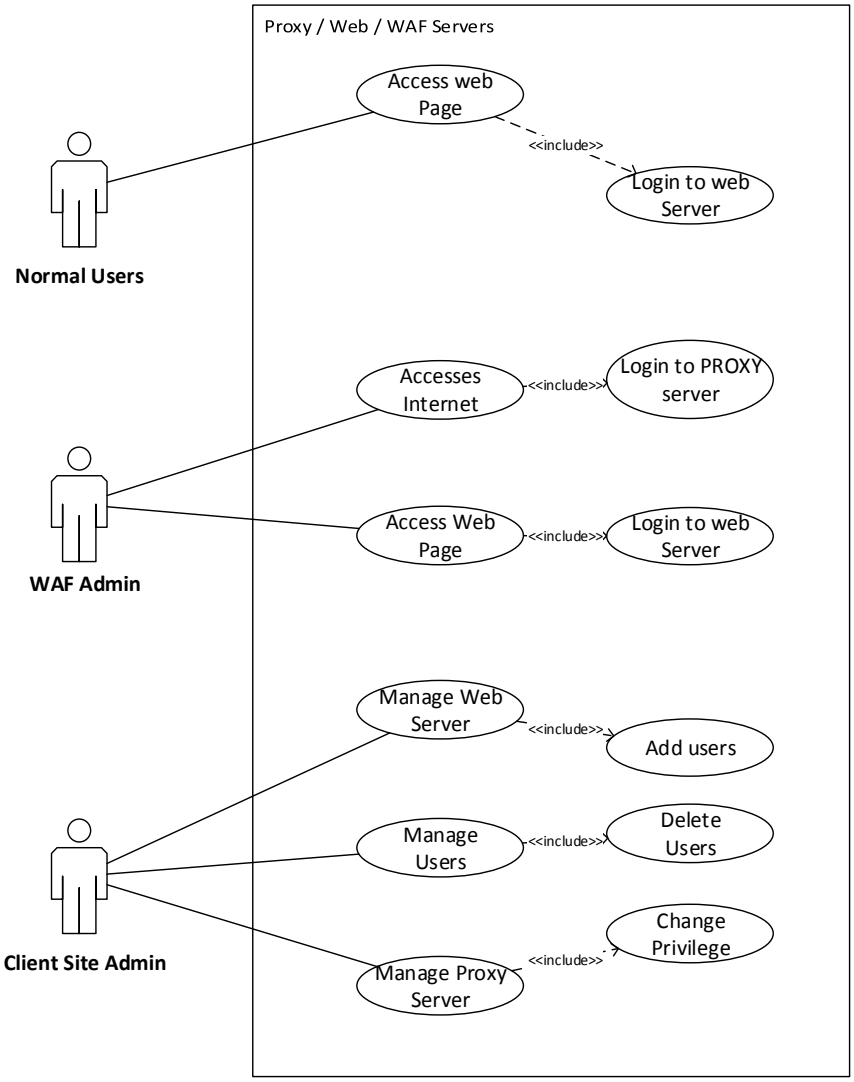


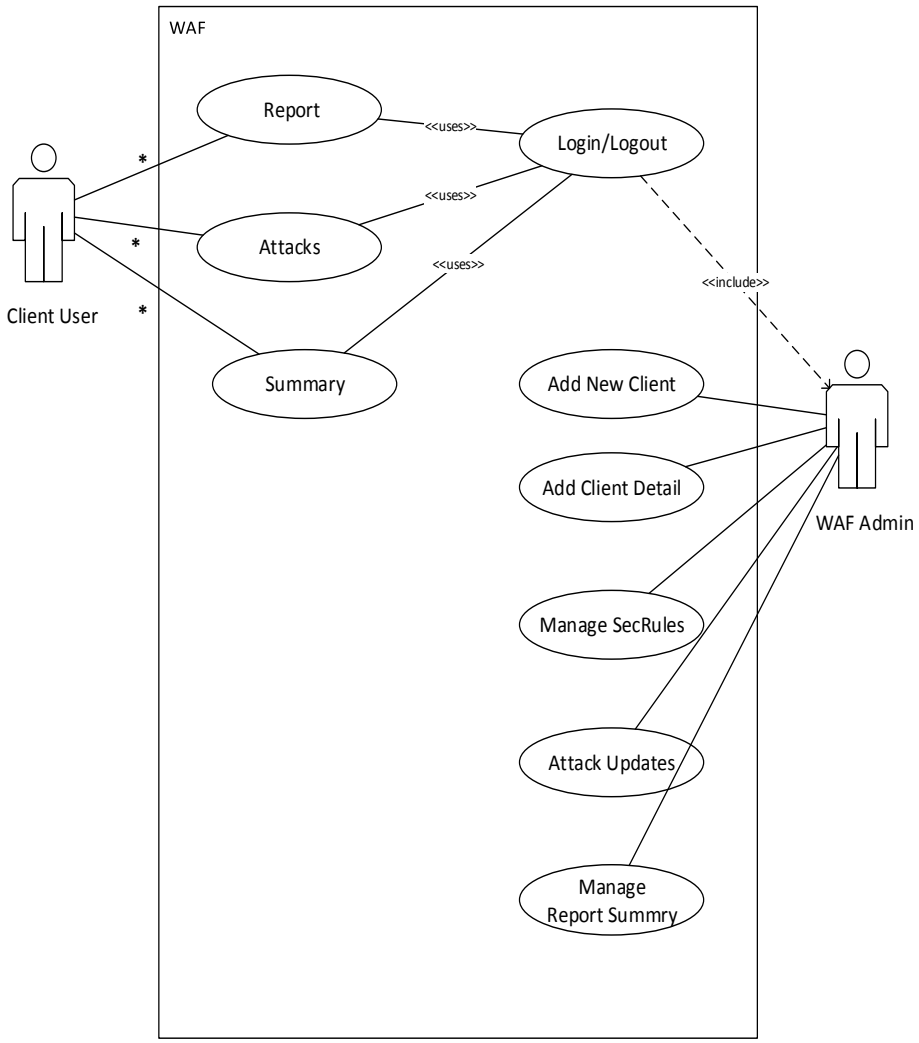
Fig 4.7.3 DFD Level 2

4.8 UML DIAGRAMS



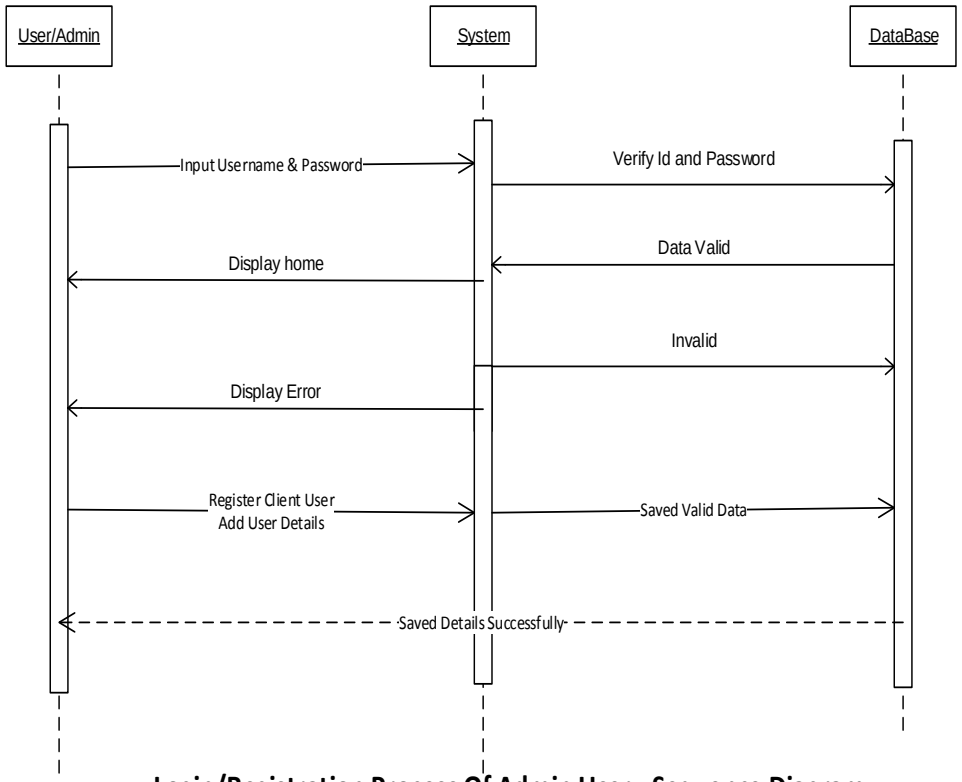
Use Case Diagram – Web Application Firewall

Fig 4.8.1 Use Case

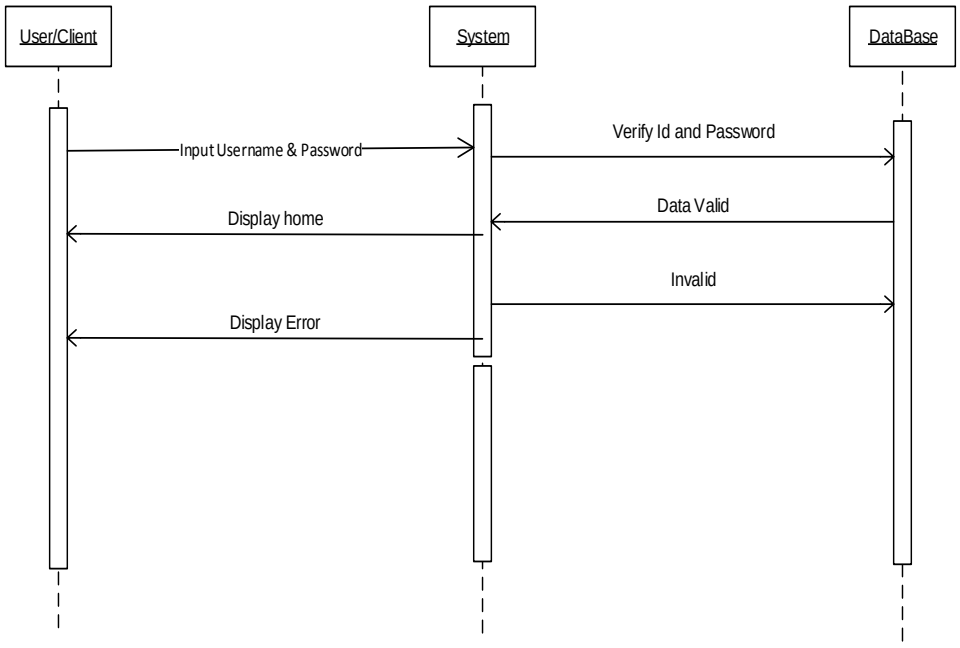


Use case Diagram of WAF For Users

Fig 4.8.2 Use Case

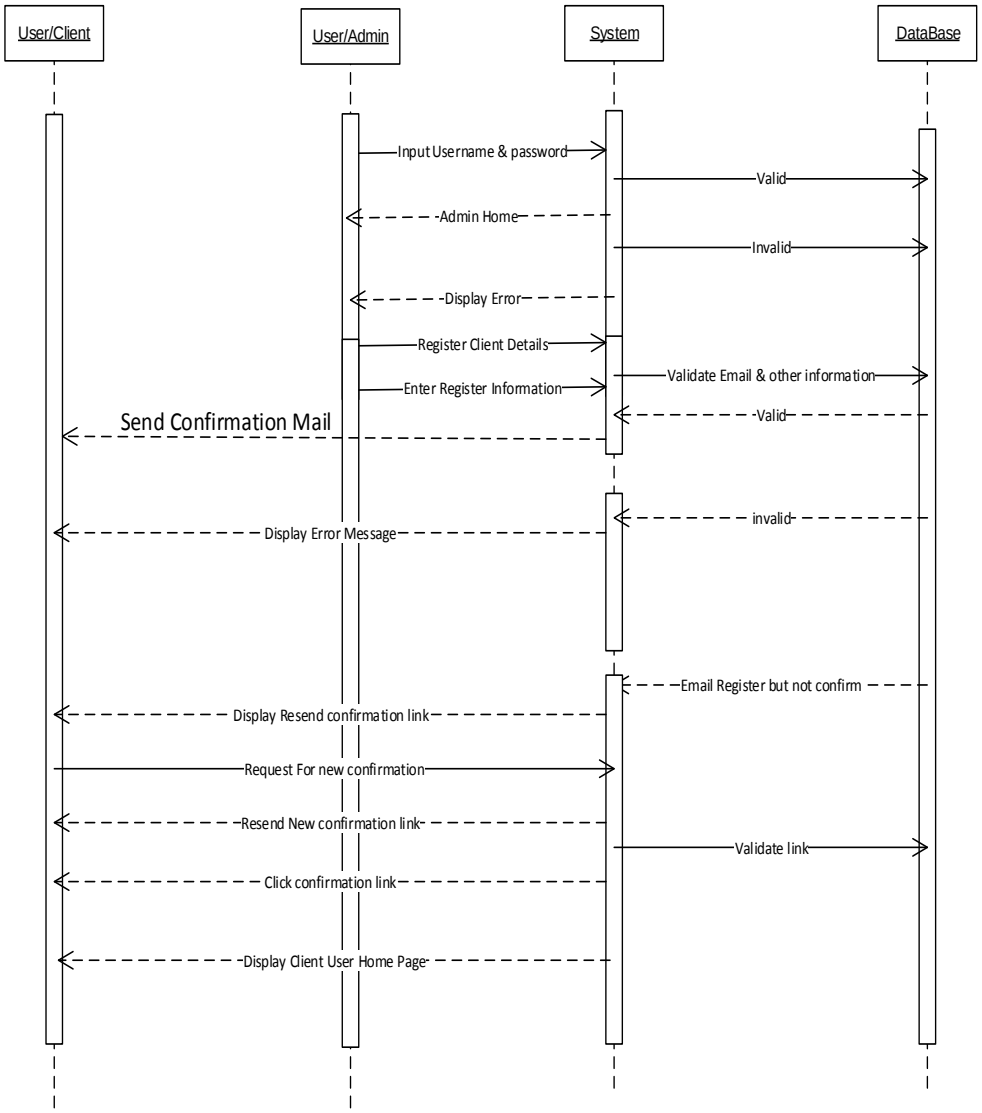


Login/Registration Process Of Admin User– Sequence Diagram



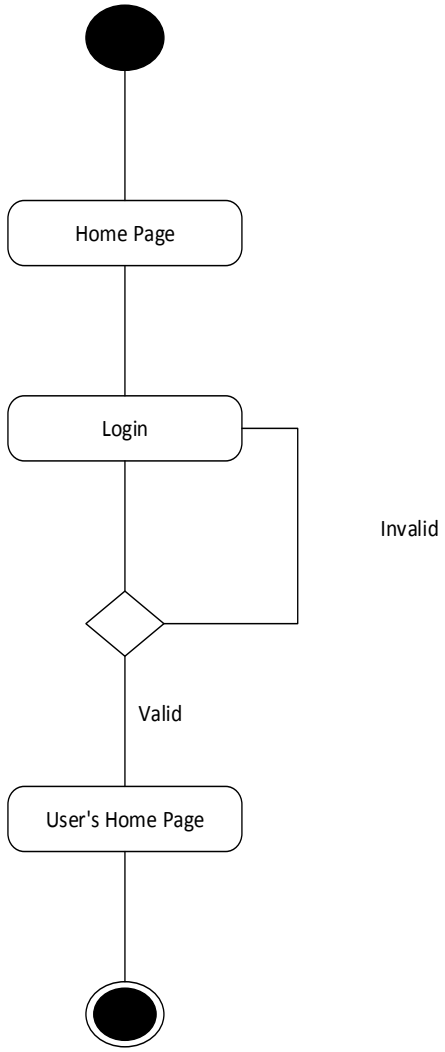
Login Process Of Client User– Sequence Diagram

Fig 4.8.3 Sequence



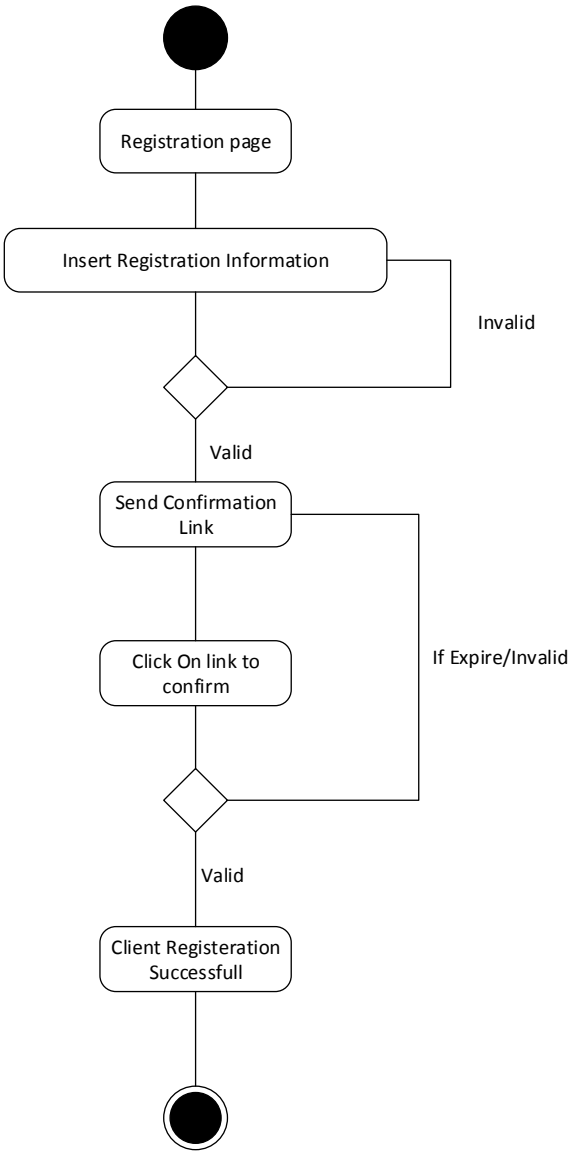
Register Process For Client User – Sequence Diagram

Fig 4.8.4 Sequence



Login Process – Client User – Activity Diagram

Fig 4.8.5 Activity



Activity Diagram – Register Normal User

Fig 4.8.6 Activity

4.9 DATA MODELING

4.9.1 Data Dictionary

The data dictionary of any system is an integral,important and significant component, since data flow diagrams by themselves do not fully describe the subject under investigation about the system. A data dictionary is a catalog – a repository – of the elements in the system. These elements center on data and the way they are structured to meet user requirements during the operation of the system. The step of creating a data dictionary is simultaneous with the process of making data flow diagram(s). Here all the fields in their respective tables are allocated so as to access these data in the system.

The data dictionary consists of different major elements like Data Elements, Data Store [tables used], data flows, Processes and other External entities used in the system. The data dictionary stores details and description of these elements. It is developed during data flow analysis and assists the analyst involved in determining the system requirements.

The data dictionary contains different types of description for the data flowing through the system.

Data elements is the most fundamental level of detail, which is also considered as the building block for all the other data in the system. It refers to all the different data used like fields, data item, etc. to make the system fully functional irrespective to the table used in the system. Here all the different types of fields used to make table are written sequentially without referring to the tables. This process helps in the process of normalization of tables.

Next to Data Elements comes the Data storage which provides the information of where and how each data elements are stored and in which table and it also gives information of any constraints if present. This step also gives detailed knowledge of different fields and their size attributes. All the normalized tables are stored in the data storage.

Next,the Data Flow stage shows all the data in the system. This step can be created in the data flow diagrams above in these documents. This step refers to all the data flow paths where transaction is done in the computerized system or the particular task or function is executed by the user of the portal.

4.9.1.1 Table Design

The database tables used in the functioning of the portal are described below with the necessary details.

1. **Register Client**

Table 4.9.1.1 Register Client

Column name	Data Type	Constraint
userid	Varchar(10)	Primary Key
Email id	Varchar(15)	
password	Varchar(10)	
Confirm password	Varchar(10)	
Security question	Varchar(10)	
Answer	Varchar(10)	
Captcha answer	Varchar(10)	

2. **Client details**

Table 4.9.1.2 Client Details

Column name	Data Type	Constraint
User id	Varchar(10)	Primary Key
CustomerID	Varchar(10)	
Name	Varchar(10)	
Mobile Number	Varchar(10)	
URL	Varchar(10)	
Service Date	Varchar(10)	
Address	Varchar(10)	

3. Scan reports Details

Table 4.9.1.3 Scan reports Details

Column name	Data Type	Constraint
CustomerID	Varchar(50)	Primary Key
AlertID	Varchar(20)	
FoundDate	Varchar(20)	
Description	Varchar(2000)	
URL	Varchar(20)	
Method	Varchar(20)	
Para	Varchar(200)	
ReqHeader	Varchar(500)	
Replay_TIME	Varchar(200)	
HTTP_RESPONSE_CODE	Varchar(100)	
RawLog	Varchar(200)	
Unique_ID	Varchar(200)	

4. Attacks reports Details

Table 4.9.1.4 Attacks reports Details

Column name	Data Type	Constraint
Found Time	Varchar(50)	Primary Key
Error	Varchar(20)	
Pid	Varchar(20)	
Client IP	Varchar(2000)	
Log Description	Varchar(20)	
Pattern	Varchar(20)	
File	Varchar(200)	
Line	Varchar(500)	
Rule ID	Varchar(200)	
Message	Varchar(100)	
URI	Varchar(200)	
Unique_ID	Varchar(200)	

5. Rules Configuration

Table 4.9.1.5 Rules Configuration

Column name	Data Type	Constraint
Customer Id	Varchar(10)	Primary Key
URL	Varchar(15)	
Blacklist_Ip	Varchar(100)	
Rule_conf	Varchar(2000)	

4.9.2 ER Diagram

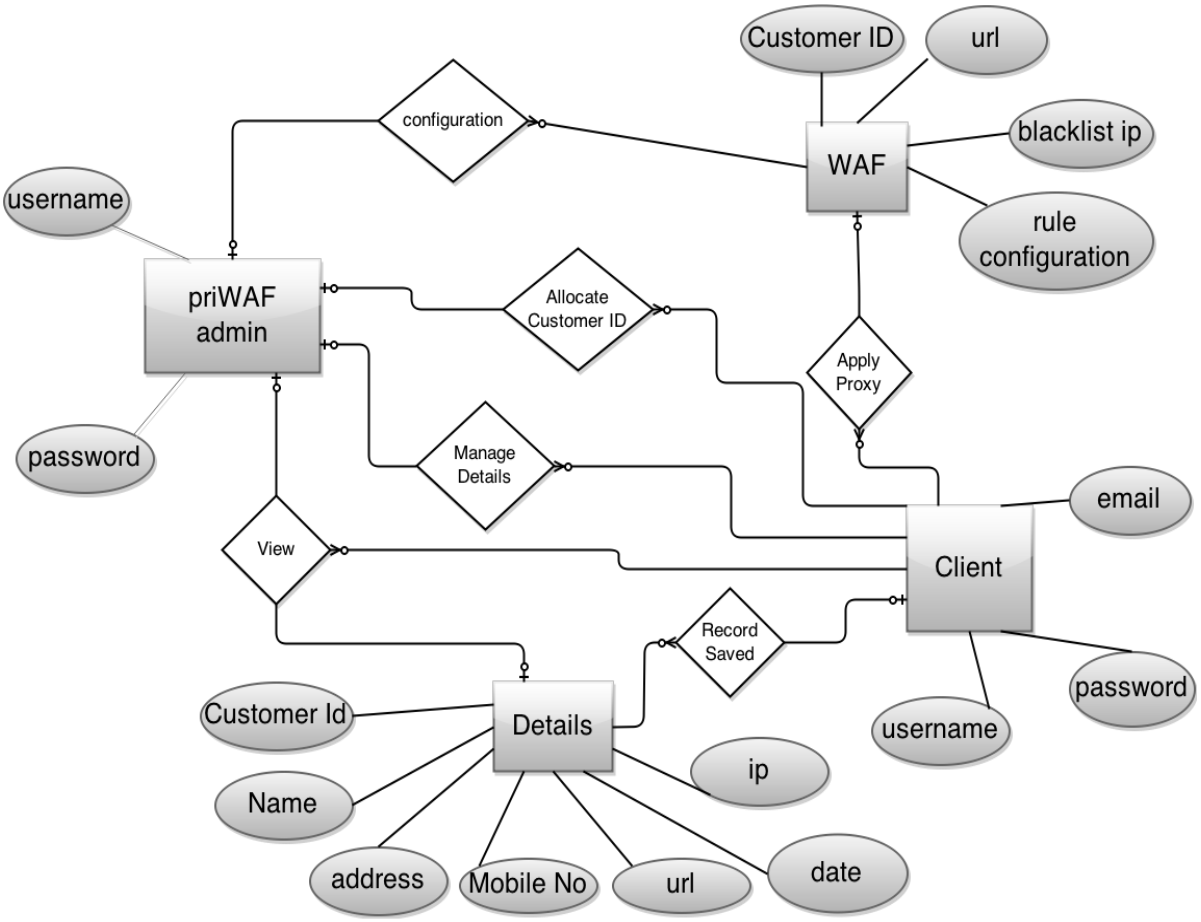


Fig 4.9.2 ER Diagram

4.10 MAIN MODULES OF NEW SYSTEM

The main modules of the system are as follows:

1. Client Registration module.
2. Client, WAF admin Login module.
3. Client Details manipulation module.
4. Content addition and management module.(WAF admin)
5. Customer Id Allocation module.
6. WAF setting up procedure module.
7. WAF maintenance module
8. Monitoring procedure module