SECURITY OF WEB APPLICATION DATABASE FROM SQL INJECTION ATTACKS

Project report submitted in partial fulfilment for the requirement for the award of the Degree of Bachelor of Information Systems Management

by

A.MOHAMED AJMEL

19BIS023

Under the Guidance of

Ms.B.Preethi

Assistant Professor

Department of Management Science

Department of Management Science

Sri Krishna Arts and Science College

(Autonomous)

Coimbatore 641 008

May 2022

Sri Krishna Arts and Science College

(An Autonomous Institution)

Affiliated to Bharathiar University

Kuniamuthur, Coimbatore -641008

CERTIFICATE

This is to certify that the Project work entitled **" SECURITY OF WEB APPLICATION DATABASE FROM SQL INJECTION ATTACKS "** in partial fulfilment of requirements for the degree of **Bachelor of Information Systems Management** to Bharathiar University, Coimbatore, is a record of bonafide work carried out by **A.MOHAMED AJMEL Reg No:19BIS023** and that no part of this has been submitted for the award of any other degree or diploma and the work has not been published in popular journal or magazine.

GUIDE HOD

PRINCIPAL

Viva voce conducted on:

Place: Coimbatore

Date:

Internal Examiner External Examiner

Sri Krishna Arts and Science College

(An Autonomous Institution)

Affiliated to Bharathiar University

Kuniamuthur, Coimbatore -641008

DECLARATION

I hereby declare that the Project work entitled **" SECURITY OF WEB APPLICATION DATABASE FROM SQL INJECTION ATTACKS "** submitted to Bharathiar University, Coimbatore, in partial fulfilment of the requirements for the award of degree of **Bachelor of Information Systems Management** is an original work and it has not been previously formed the basis for the award of any Degree, Diploma, Associate ship, Fellowship or similar titles to any other university or body during the period of my study.

Place: Coimbatore

Date:

Signature of the Candidate

ACKNOWLEDGEMENT

First and foremost, I thank the almighty for endowing his immense blessing that helped in each step, towards the completion of the report.

I express my heartfelt thanks to our CEO **Dr. K. Sundararaman M.Com., M.Phil., Ph.D.** and **Principal** for providing me the facilities needed to complete this project.

I take this opportunity to express my deep profound gratitude to our Head of the departments **Prof. P. Rajan M.Sc., MBA, (Ph.D.) and Prof. Julian Gnana Dhas C B.E., MBA NET** and for all their encouragement and providing his healthy co-operation throughout the project.

I express my thanks to my **Guide Prof. Ms. B. Preethi M.Com IB., M.Phil., (Ph.D.) -,** for her guidance, engagement and aspiring support to complete this project.

Finally, I express my thanks to my parents who have given support and encouragement in doing this project.

A.Mohamed Ajmel

19BIS023

LIST OF CONTENTS

ABSTRACT

The internet users have been rapidly increased so as the web applications are in much command. These can be accessed over the web with the help of any web browser on any operating system. To serve large number of users, huge volume of data is stored in web application database servers all around the world. Web applications have been becoming the comprehensive disclosure of commercial online industry. Swift development in web technologies has been step up the rate of embracing the database driven web application. With online activities like e-banking, e-shopping, e-booking, e-trading, e-payments etc. the web applications have become the essential tool of our daily routine activities. Web applications interact with the back-end database to retrieve data as and when requested by the user. Web applications are the backbone of today's online business industry. For various activities such as paying of bills & merchandize information must be kept safe & secure with these web applications but unfortunately there is no guarantee of integrity and confidentially of information. The hackers and assailants are likewise running an equal economy by hacking the application information and taking the individual and important data identified with people or with associations. The global exposure of such applications makes them prone to the attacks due to presence of vulnerabilities. These security vulnerabilities continue to contaminate the web applications through many injection attacks.

SQL injection attacks (SQLi attacks) are one of the top most threats in database centric web application and SQL injection vulnerabilities (SQLi Vulnerabilities) are the most serious vulnerability types now a days. The SQLi attacks could be solemn threats to web application database. It permits the attacker to take control over application ensuing -leak of confidential data, financial frauds, denial of services, database finger print, network hacking, destroying database and many more to count. Security threats in web application are becoming a challenging role for the web developers and security professionals. In this research work a paramount security mechanism for web application database is discussed to prevent SQLi attacks. This work recommends a risk analysis approach based on probability and impact of SQLi attacks executed on web applications. This analysis estimates the risk associated with a particular web application. Furthermore, it also helps in computing the risk level and risk range of a web application. These obtained results would be much useful to the web developers, web administrators, security analysts and organizations which are serving in the domain of information and web application security.

LIST OF TABLES

Chapter-I

Web Application Security

Preview

The World Wide Web (WWW) has known as the web- global repository mechanisms that provide the information to users via computers. Initially, the web was planned for sharing data all the way through with one another and furthermore with machines. It was first developed in 1989 as a vehicle for the transmission of material which can be read only from vigorously stacked corporate workers to the mass of web associated customers. A group of innovators like Tim Berners Lee who brought HTML, HTTP and URL together and made the World Wide Web. The web comprises pages that are utilized by a browser. A browser is just like software or a computer program that allows users to view the retrieved documents. The web provides clients admittance to an immense range of documents that are associated with one another by methods for hypertext or hypermedia links. Hyper Text Markup Language (HTML) is used for writing web pages and these works with Hyper Text Transfer Protocol (HTTP).

The universal acceptability of the web has changed the lifestyle of mankind. Today electronics files are replacing paper files in institutions like banking, healthcare system, insurance sector, education and many more to count. Nowadays, this web is turning into a significant apparatus for many small associations, families and even individual data frameworks. At times the term web is frequently erroneously utilized as an equivalent word for the internet itself yet the web is a service that works on the internet like: an e-mail service. A web is also known as the information superhighway whereas the internet is the huge and largest interconnected network in this world. Generally, the web works on the basic client-server design of the internet. A server is just like a program that broadcasts & stores data to different machines and while clients are also computer programs that demand data from the server as the client requests them.

Now the people are demanding more and more access to the web for faster and constantly available information. Initially, the web was not created with so much security, as it was meant for only sharing information and ideas in academics but in the modern era, it is available to everybody even with dubious moral values. User's desire

to retrieve information anytime and anywhere is satisfied by means of web-based applications. With the growth of this web traffic, users are exposing themselves to a greater risk of having vital personal information intercepted by attackers. A system that contains this type of sensitive information must be safeguarded prior to the attack. Today people even don't want their cell number to be disclosed, there is a need to strengthen the security of the web applications which uses the database for information retrieval and exchange.

1.1 Information Security

Information (data) security, now and then well-known as Infosec, is the act of averting unofficial admittance, utilization, disturbance, alteration, examination, recording, or devastation of data on the World Wide Web.

Information can be processed through a computer system from anywhere in the world. Information is a collection of meaningful data i.e. an asset to all individuals and organizations and security means protection. Information Security refers to the protection of these assets in order to achieve CIA tried - confidentiality, integrity and availability i.e. maintaining the core principle of information security intact. These objectives ensure that sensitive and personal information about the users or organization is just unveiled to authorized users (termed as confidentiality), prevention of unlawful alteration in information (termed as Integrity), and assurance the information must be right to use by certified clients when demand (termed as Availability). Numerous organizations hire a group of professionals for managing the security of crucial data of the parent organization. This group is normally liable for directing the risk related processes. The organization's status is labelled by the security of assets.

1.1.1 Principle of Information Security

Information security holds the CIA paradigm acknowledged as the central part of Infosec.The three essential components are discussed here.

a) Confidentiality- The ability of the system to make its resources accessible only to authorized users. Like an unauthorized user can't perform any transactions from genuine user credit/debit/cash or any other card details. This property is essential but sufficient for preserving the privacy of user information.
b) Integrity- It is the ability of the system to allow the authorized user to modify data only in specified ways defined by the system. Like an employee can't modify his salary or an authorized user can't vandalize their own web application. Integrity

means the information can't be modified without the proper authorization of the user

c) Availability- It is the ability of the system to provide access to information to rightfully request by the user whenever needed in a specified time. Like an authorized user can check his bank account detail at any time. The information must be available to the authorized user whenever they required i.e. services can't be denied to any user at any point of time. The graphical representation of the core principle of information security is shown in the figure given below.

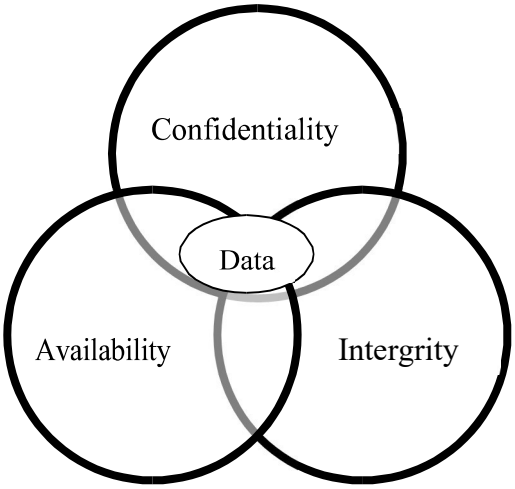

Fig1.1 CIA paradigm

From the above paradigm, it refers that when confidentiality is breached, integrity is violated & availability is denied i.e. the principles of information security are at stake. Then it concludes that someone (attacker) has exploited the weakness and gains the control of the system further "An action has exploited vulnerabilities in a controlled manner" i.e. the system is being attacked by some unauthorized person. Information security is a progressing the cycle of exercising to guard information and information systems from unlawful admission. This cycle involves continuous preparation, appraisal, security, checking, incident reaction and fixing, documentation and survey too. Information security assignments are certainly identified with reducing the risk due to the utilization of information. It provides protection of data and information from a scope of threats in order to guarantee to limit the risk value and enhance the rate of profit.

1.2 Formal Terminology

Before moving further, there is a need for some formal definitions in context with the domain of information security. These terminologies will outline concepts for better understanding of organizational risk management process

i) Vulnerability- The weaknesses or susceptibility in form ambiguity, loopholes, deficiency or defects of any system.

ii) Attack- A technique to take advantage of weakness of the system.

iii) Threat- A progression of actions to harm the system in an unapproved manner to harm

iv) Risk-Risk is the impact of the threat or combination of events & its respective impacts

v) Risk analysis- a process of identifying and analyzing the threats with their potential impact. It is a systemic method for the estimation of risk value associated with web applications.

1.3 Web Application Security

Web-based application/web application or web apps are the same sides of a coin. A web- based application is any program that is accessed over a network connection using HTTP, rather than existing within a device's memory. Web-based applications often run inside a web browser. However, web-based applications also may be client-based, where a small part of the program is downloaded to a user's desktop, but processing is done over the internet on an external server. Web-based applications are also known as web apps. Web applications include online forms, shopping carts, word processors, spreadsheets, video and photo editing, file conversion, file scanning, and email programs such as Gmail, Yahoo mail, Hotmail etc. The internet is a medium for exchanging data or information. For effective communication or exchange of information there is a need a web application (WebApps) that receives data and sends it back to the web server or database server. Technically, a Web application is like any other computer program, accessible only through the web that performs dynamic changes in data and executes a specific function. Web applications are placed on web servers which are accessed through a web browser on a computer that hosts the application by end-user for all communications.

In fact, the web application owner could perform data updation and modification with minimum effort at any time. As the core parts of web application stored on the server-side within the application server. The programming languages such as PHP, ASP, CGI used for developing web applications are also browser friendly. A browser is responsible for running these applications and helps in exchanging the data from client to server and vice versa. The principle purpose behind the prevalent acceptance of web apps in the present associations is permeability and global admittance. The accessibility is not limited to a computer system but also through many communication devices such as mobile phones, tablets, PDAs, etc. The organizations are moving towards web applications as with little investment, an organization can reach its potential clients globally and open up a marketing channel for business and advertisement. Common web applications such as "Google" for online searching," Flipkart or Amazon" for online shopping," Stacksoverflow" for open source projects and many more for e-mails, Blogs, Social networks, E-commerce, online Auctions, online Banking, Online Trading etc. The working model of the web application is depicted below.

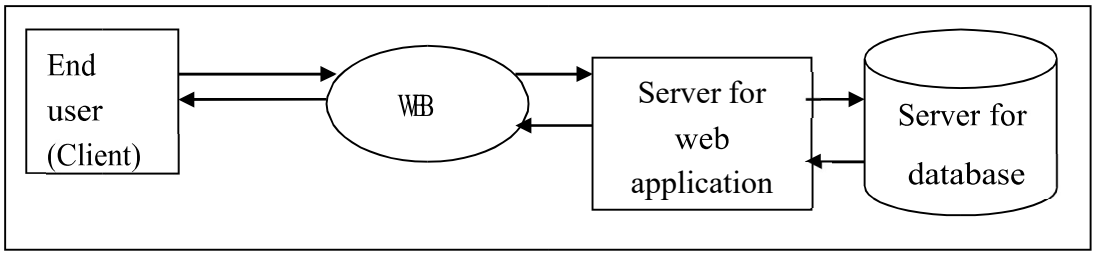

Fig 1.2 Web application model

The recent development in internet-based applications along with various new techniques has been changing methods of information communication and doing business. Thus, a lot of organizations have changed their business to online mode. This helps, firstly the people of the organization working in the far-flung workplace and secondly, trade associates of various nations may be able to share crucial information in actual moments to achieve their aims. With the emergence of Web 3.0 and HTML5.0, user's demands for information sharing have increased via social engineering and acceptance of the web for delivering services anytime (24*7). This requirement leads to the development of web applications for making the data available online like purchasing any items online or making transactions in a bank account without visiting the bank are examples of web applications by organizations. This development came with a dark side; some hackers exploited these applications for easy and unlawfully monetary gain. This is the start of the new-age industry: security of web applications. Now information security has a new branch that deals with securing the data placed online, services

used for web applications, threats to application data security i.e.- web application security. It only deals in securing the data and services required for web applications avoiding malicious users like hackers and attackers for exploiting vulnerabilities presents in an application's code. It is the process of preventing and detecting unauthorized use of users and organizations vital information. A prevention measure helps to prevent information from attackers and a detection measure helps to determine whether someone has tried to break into the system or not. Most attacks are performed directly on applications like- content management with a backend database, software as a service application and data administration. Therefore, organizations are taking measures for securing their application by securing the system and its network. If any organization fails in this attempt, then the risk of attack on application increases. The attack can result in data loss, loss of clientele and reputation too.

1.3.1 Web Application Architecture

Web applications are a set of web pages and programs which reside on a web server. The inputs provided by the user are sent to the server in the form of a parameter string. These inputs are used to engender the SQL query to retrieve information from the database. An authorized user can access it over the cyber world or over a public network and store the data in the database. A web application utilizes a web browser as an interface to extract the data from database server to accommodate the queries placed by the users. Every web application is composed of 3 tier architecture consisting of three layers. Each layer can run potentially on a different machine and each layer should be independent of other layers. The three-tier architecture incorporates a database server alongside client and server architecture. An internet browser on the client-side process all information and grant users to connect with elements of internet applications by and large developed in CSS, HTML and JavaScript. The server-side code written in PHP, Java and Python responds to user requests made through a browser. The database server on server-side manage, store, retrieve and provides information as and when required. In this section, the function of each layer and basic architecture is briefed with the help of figure given below

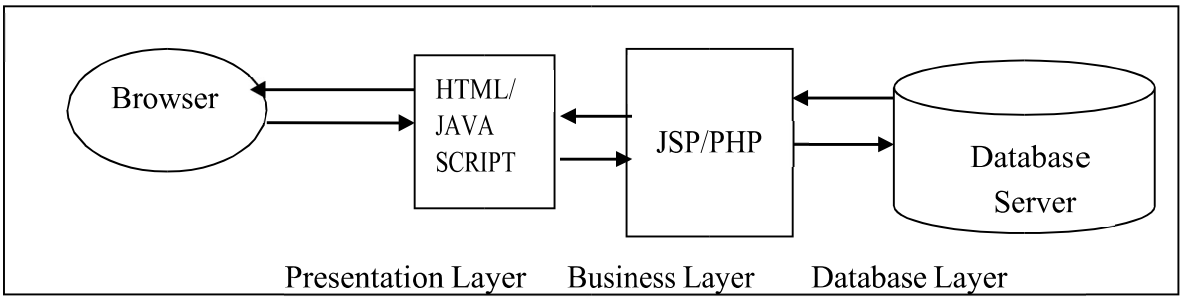

Figure 1.3: Basic architecture

a) Presentation Layer-It contains presentation logic. It is the topmost level of application and handles the interaction with users. Its main function is to receive input from the user and provide the result back to user in a convenient way that the user can easily understand.

b) Business Layer-This layer acts in between the presentation layer and the database layer. It is a logic layer that consists of a set of rules for processing the information between two layers. It contains application process commands which retrieve the data from the database and sends it to the presentation layer for viewing the data. This layer can be programmed in any server scripting language like JSP, PHP and ASP etc.

c) Database Layer-This is a physical storage layer for data persistence. It manages all access to the database and file system. Information is stored and retrieved from the database. It is then passed back to the presentation layer for processing and eventually back to the user. The main function of this layer is to provide access to the authorized users and restrict the maleficent user.

The basic architecture of a web application explains that web browser sends the request to presentation layer, the request gets processed and dynamic part of the request is sent to next layer i.e. business layer. This business layer processes the server scripting languages. All requests for database access are passed to the database server. The result is then dispatched to web browser as web pages. This architecture is easy to maintain and all the components are reusable. For the faster and smooth functioning, all the layers are governed and managed by different groups of experts. Web designer looks after the presentation layer. Software engineer does the logic and database administrator manages the database servers.

1.3.2 Threats to web application security

The major kind of attack on web applications are injection attacks. These are ranked in the first position among the top ten most dangerous web application attacks in various surveys conducted by (OWASP) Top-10. The community of web application security is managed by an organization called OWASP and WASP. These global organizations put a spotlight on the enhancement of securing the data of web applications by providing research papers and articles, providing free tools and methodologies and other documents. Open web application security project community provides a list over a period of three years of Top 10 attack categories from which any web application suffers. According to OWASP, the most attempted attacks on web applications are referred in a table depicted here.

Sl.No.	Name of Attacks	Description
1	A1:Injection Attacks	At the point when a piece of untrusted information is sent to the user as a query and attacker gets unapproved admittance to the web application. For example, SQL, LDAP, OS and so forth.
2	A2:Broken Authentication	At the point when application capacities identified with verification and sessions are not actualized accurately. This permits the attackers to bargain session, passkeys and passwords
3	A3: Sensitive Data Exposure	Some applications and interfaces secure critical information week by week. So it becomes easy for attackers to steal or modify conduct financial fraud, unauthorized access, or other crimes.
4	A4:XML External Entities (XXE)	Numerous weakly designed XML processors assess other references which may be present in reports of XML. The other references are used to extract codes with the help of remote command, internal references etc.

5	A5:Broken Access Control	These types of vulnerabilities are exploited by attackers to take out the potential & relevant data. The type of data includes- viewing & editing of other client's records.
6	A6:Security Rearrangement	At point, when great security having a safe design characterized and conveyed for the application, systems and servers, information base servers are not applied. This incorporates staying up with the latest software, including all code libraries utilized by the application.
7	A7:Cross-Site Scripting (XSS)	At the point when untrusted information is sent by an application to the internet browser without appropriate approval then XSS permits attackers to execute contents that can capture client sessions, closing the application or even divert to other malevolent applications.
8	A8:Insecure Deserialization	It's simply prompting remote command implementation. Once in awhile deserialization imperfections don't bring about remote command however can be utilized to injection/infusion and benefit acceleration attacks
9	A9:Using Components with Known Vulnerabilities	Numerous segments run with similar rights and benefits as the application like libraries, frameworks and software modules, on the off chance that if a segment gets defenseless like information misfortune or server takeover then entire applications gets attacked.
10	A10:Insufficient Logging & Monitoring	This attack is linked with inadequate reconciliation with occurrence reaction that permits attackers to additionally assault frameworks to alter separate or devastate information.

Table 1.1- Portrayal of attacks

1.4 Injection Attacks

Injection attacks are key concerns in internet-based application security. The OWASP has also ranked at the top of the list among the top ten attacks. Injection attacks, particularly SQL injection (SQLi) are very dangerous as their attack surface is enormous. Furthermore,

these types of attacks can be performed by inexperienced attackers very well to exploit the vulnerabilities present in the application. Injection attacks mean attacks that are caused by processing invalid data. The invalid data is routed by a query that changes the actual execution of the program. Usually, the injection of code is performed by SQL query, Linux command and operating system commands etc. Since its inception, the injection attacks are considered as very risky attacks. These can results in permanent loss of all data, defacement of application, fingerprinting of database etc. even in some cases may take total control of the application. The reasons for these attacks are performed given as under.

i) Randomly alter values in an information base through a sort of code injection called SQL injection. The effect of this can extend from site disfigurement to genuine trade- off of crucial information.
ii) Performing pernicious code or execution of malware on a backend data server by writing scripts in any scripting languages.
iii) High accessibility to all accounts can be the source reason for misusing injection weaknesses on any operating system may be windows or Linux/Unix

1.4.1 Types of injection attacks

The simple queries or commands can be used for attacking the user's code by any kind of attack. Even inside the resource of application code also plays a part in executing these attacks. Table 1.2 summarizes the various types of injection attacks with their description and potential impacts on the application.

Sl.No.	Type of Attack(s)	Description	Potential Impact
1	SQL Injection	This injection attack accomplishes the advantage of the syntax utilized for composing a SQL query to inject malicious query/queries that can execute or change the information resides in a backend data set.	Bypass authentication, disclosing information, denial of service or full system/application is compromised

2	LDAP Injection	This injection is caused by modifying LDAP (Lightweight Directory Access Protocol) statements i.e. granting permissions to unauthorized queries from data provided by users.	Bypass authentication, Privilege escalation and disclosure of information
3	Xpath Injection	This particular type of infusion misuses the web applications that permit attackers to infuse information into an application so as to execute Xpath commands.	Bypass authentication and disclosure of information
4	OS Commanding	In this kind of attack, the OS commands are injected via HTTP for guessing the passwords and misusing the application	Full system compromise
5	SMTP injection	when code insertion is not directly possible then this attack is executed by to a mail server via SMTP statements	Revel of information

Table 1.2 Depiction of injection attack

1.4.2 Consequences of injection attacks

The SQLi is executed when a piece of a query is deliberately inserted into some applications code via any input field like login field, feedback form, search bar etc. The results of this insertion depend on the vulnerabilities present in the application. The consequences of SQLi directly affect the principal of information security. The following steps will describe the details:

a) Confidentiality: After the successful execution of SQLi, all the personal and crucial information is accessed by unauthorized users. The secrecy of data is no more intact. Hence confidently is lost.

b) Integrity: When an attacker enters the system, the fate of data stored in the backend database is in the attacker's hand. The data can be edited, new data can be inserted, and a complete database may be also deleted. Hence the integrity of data is tempered.

c) Availability: The authorized user has the right to access information from anywhere and at any point of time. The SQLi denies the right to access for an authorized user by an unauthorized attacker. Hence the availability is denied.

Some other consequences include- authentication and authorization in which database is get affected duo to improper data filters and extra privileged accounts

1.5 SQL Injection attacks

Open web application security project found that 80% of web applications are defenseless against injection attacks and SQLi is a category of injection attacks. These attacks can get and control any application whose administration inhibited to an authorized person/group. It is depicted as the most serious risk for an internet-based application that can be accomplished effortlessly with no proficient guidance. The attention on executing the web application functionality relative to security from the inner and outer surroundings, are the reasons for vulnerabilities. The capability of the attacker perforates the application to dig out delicate data. The misuse of vulnerabilities in the plan of web application infringes the primary standards of data security. These assaults are application-level assaults that are not blocked by standard detection systems like firewalls and intrusion detection systems. It is difficult to identify SQL infusion assaults before its effect. By and large, illegal actions are carrying out via valid user permission for getting into the vital segment of stored information of the web application. This stored information centers are persuaded that infused code is grammatically as correct as a genuine SQL query. The information supplied through user, structure the query to get stored information. On the off chance that information supplied by the end-user is not appropriately purified, they can make the application create unintended outputs. To understand the working principle considers the following SQL query.

SELECT * FROM TABLE Emp WHERE Eid= 'OPS99';

The query mentioned above supply the data regarding a specific employee (with Eid= OPS99), however a SQL query with injected code will act distinctively as the query original relevance is modified in unauthorized manner by an assailant utilizing query-

such as Use Select *(select all column) from TABLE Emp (table name) WHERE (condition) Eid= 'OPS99' OR '7'='7'; (Always true condition 1=1, a=a etc.)

This query will be translated as "SELECT * FROM TABLE Employee" because of injection of always true condition of injected code (OR '7'='7').

The output of this query represents the data for all employees from the table in lieu of a desired output that exposes the whole database. This is exceptionally fundamental and basic SQLi attack. These natures of vulnerabilities prompt the expansion of much more refined attacks. The professional attacker can utilize very logical and resourceful keywords to extract the data from database servers. SQLi attacks are the on the whole are hazard to database- driven web applications

1.5.1 Chronology of SQL injection attacks

This section portrays the advancement of SQLi attacks from the time of existence. An organization named OWASP founded in 2001, is an open community which provides free & open tools, documents and chapters for everyone to develop, operate, maintain and improve the application security in all areas, provides the list of threats (named Top 10 project) to web application security from time to time in accordance with changes in the intensity of the attack. The main objective is to enhance the alertness regarding the security of web applications by means of recognizing a number of threats. The table below discusses only the evolution of SQL injection attacks since the time of its existence.

Sl. No	Resource Collection Period	Release Year	Position in Top 10 Threat(s)	SQLi attacks as threat
1	2001-2004	2004	VI	The malicious codes in the form of queries are injected which are executed by applications
2	2004-2007	2007	II	At this point of time these SQLi becomes famous and frequent. Attacker uses malevolent query through user input to inject code.
3	2007-2010	2010	I	Tremendous increases in SQL injection attacks along with OS command injection and LDAP injection

4	2010-2013	2013	I	The new addition in the list is injection through web application URL along with previous type of injection to get access to crucial data.
5	2013-2017	2017	I	Another addition is XXE injection for attain limitless and unofficial admission to the backend data storage
6	2017-20202	2020	I	SQL injections are becoming dangerous to the security of application program interface (API) and No SQL technology used by Big data

Table 1.3 Chronologies of SQLi attacks

1.5.2 SQL Vulnerabilities

Those applications whose code is not tested or reviewed before propelling the web applications may have programming vulnerabilities. This sort of indiscreet conduct puts the application at risk. A risk is a potential cause that can exploit the vulnerabilities. The programming errors/flaws introduced during the design of the application got the consideration of the attacker to persuade them to penetrate the backend database of web applications. This sort of practice uncovered the syntax structure requirements. The basic idea of PL/SQL is likewise motivating and responsible for SQL injection attacks. Utilizing these vulnerabilities/syntax constraints in the user input tab or URL bar of the application, a database can be pulverized. The table depicted below shows the corresponding risk to respective vulnerabilities.

SQL Vulnerabilities	Risk to web application
Weak data type	Code execution without verification
Extra privilege account	Easy execution of sensitive data
Non sanitized inputs	Bypass the standard authentication
Detailed error messages	Provides DB schema
Buffer overflow	Consumes extra storage

Table 1.4 SQLi vulnerabilities with associated risk

Susceptibility (vulnerability) is the shortcoming, escape clauses or weakness in the plan of an application. Each coding language has a few imperatives. The poorly developed web applications are easy targets for these vulnerabilities which direct to form risks for a web application. As a whole these kinds of exercise permit the aggressor to design the assaults as needs be. By embeddings some specific watchwords the ideal yield can be extracted accordingly. The assault relies upon the weakness present in the application. The most seasoned, normal and perilous susceptibility are –SQLi susceptibilities. The attackers use the easy implementation of SQL queries and commands to discover the weakness presents in the applications. It affects any website/web application having a backend SQL-based database. The attackers can bypass the authentication and authorization mechanisms to access to the content of the backend database affecting data integrity. The common vulnerabilities exploited by attackers are as follows

i) Mismatching of data type used
ii) More rights to general account
iii) Excess functions in application
iv) In appropriate filtering of input values
v) Bypass Authentication
vi) Informative Error Messages

1.5.3 Injection methods

The injection of malicious SQL statements into a vulnerable application can have different input mechanisms i.e. the different ways an attacker can use SQL injection to transverse through a vulnerable application. The outcomes of this unauthorized penetration by attackers are the breach of three basic information security principles. The following table displays the most common mechanisms.

Injection through	Attack vector
User input	Suitable crafted SQL commands
Cookies	Details extraction from the user browser
Server variables	HTTP, network headers and environmental variables

Order injection	Indirect triggers
Blind injection	True/false query statements
Database focused	SQL query manipulation

Table: 1.5 Injection methods

1.5.4 Motive behind SQLi

Attackers are dependably distinctly inspired in taking advantage of vulnerabilities present in web applications to get unlimited and unauthorized access. In the accompanying table the purpose behind the injection attack and the attacker's expectation in arranging that particular attack are listed

Reason for attacks	Attacker's motive
Data extraction	To extract sensitive and personal Information
Data modification	Modification of backend data accordingly
Database fingerprinting	Obtain vital information about DB schema
Bypass Authentication	Availing same privileges as Admin
Identify inject able parameters	Helps in planning attack according to weakness in web application
Perform Denial of service	Shutdown database, Services not provided to authorized user
Executing Remote commands	To gain control of the whole system

Table: 1.6 Intent of attacker

1.6 Types of SQLi attacks

There are a few sorts of attacks executed by the attacker to capture the backend database. It is very difficult to classify every single existing attack in light of the fact that different attacks can have a comparative name in various circumstances. There are endless varieties of attack orders, for example, to discover malicious constraints, to determine vulnerabilities, to extricate particular data and so on. As the distinctive sorts of attacks are for the most part not performed freely, huge numbers of them are utilized together or

successively or relying upon the particular objectives of the attacker. In this section, the taxonomy of SQL injection attacks are discussed. These attacks are portioned as under

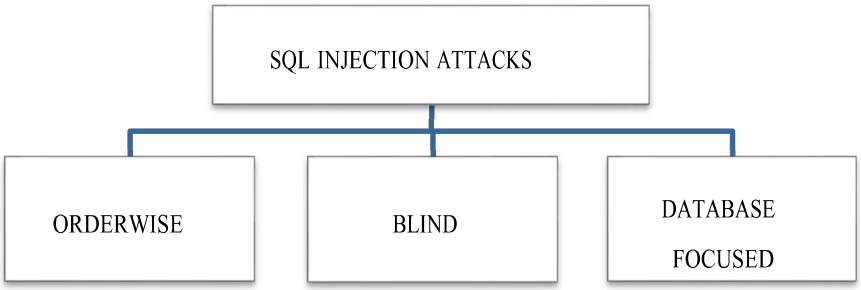

Figure 1.4 Classification of SQLi

A) OW (Order-wise attacks) —these attacks are actualized when a malevolent code is infused into the client's code directly/by implication for boundless gain and unlawful admittance. These are classified as FO-first-order, SO-second-order & LI- lateral injection attacks.

i) FO-First-order -The order of execution of the first-order attack is the insertion of the malicious string by attacker, results in the implementation of customized code instantaneously. The attack implementation includes the attackers specifically collaborate with the application.

ii) SO-Second-order -The execution of second-order attacks are initiated by other activities when an attacker tries to penetrate into a dependable source of data storage. In these injection attacks, the malicious code is not promptly initiated upon injection. Rather, it is put away in the client framework/application and executed by the client amid resulting in subsequent activities.

iii) LI-Lateral attack –The execution of lateral is operational when the attacker controls the variable elements i.e. implicit functions of a query language to alter data that is pertinent to clients.

B) BI -Blind attacks -the attackers get access to perceptive data by soliciting an arrangement from right/wrong or true/false inquiries through SQL statements. Important information is extricated on the principle of answers given by the application. In this situation, as opposed to getting an error message, the attacker gets the required information.

C) DF-Database focused (against database of web application) - majority of attacks are normally utilized for extricating information from back end data storage. Improper

filtration of values inputted is major cause for this attack type. Proving security to application from this type is particularly troublesome and even IDS's (intrusiondetection systems) and firewalls can't distinguish these on the grounds that these attacks are made from SQL statements and keywords. The essential sorts of attacks against the database are clarified as underneath:

i) Tautologies -Tautology is a statement that assesses constantly valid for any expression. It enables an attacker to sign in to an application without providing substantial and valid user identification. The broadest type of SQL tautology is comparators like OR, < and > and so on

SQLi Attack Type	Damage to backend database	Example/Sample query
First order	Access to data as authorized user	Inserting a true condition like 1=1 or a=a, with input query
Second order	Whole data can be targeted	By manipulating temporarily-stored or dynamically-cached search requests.
Lateral order	Access to system settings	Access to system restricted functions like date and time
Blind injection	Access to whole system via URL	http://www.sampleportal.com?id=3 http://www.sampleportal.com?id=3+ and 3=2
Against Database		
Tautology	Usually affects individual identity	SELECT Staff_id from S_Record Where SID= 'qwsx' OR 'k=k' and pass= ' --- ';
Logically incorrect queries	Information about backend DB revealed	SELECT salary from S_Record where S_ID ='@#$%^&*' and S_No ='VPK+=79%';
Union Query	Access to sensitive information and fingerprinting database	SELECT Staff_id from S_Record where user SID= 'qwsx' and password=-----; UNION SELECT Salaryinfo from Salary where SID = 12';

Piggy backed	Whole data can be deleted	SELECT * From S_Record Where =' S_ID; DROP TABLE S_Record;
Stored procedure	Access to the system by remote command execution	SELECT S_ID from S_Record Where S_Name='wxyz'; SHUTDOWN; and password='psrs42';

Table 1.7 SQLi types with sample query

ii) Logically Incorrect -also acknowledged as pre-arrangement attack because the attacker infuses a query which causes a language syntax or logical error with some valuable data about the underlying database. On the premise of the subsequent error message, the attacker extricates the information with respect to the points of interest of database being utilized.

iii) UNION Queries -Likewise called as statement injection attack. An attacker can specifically get too sensitive data from a database as most SQLi are performed for this reason. In this sort, the attacker utilizes union queries between two SQL statements that contain set operators. The yield of UNION attack depends on combination of genuine & infused query.

iv) Piggy bagged Queries -This inquiry is the merger of two queries one certified and one malevolent i.e. a supplementary/ malevolent query is added with core query. The malevolent query is utilized to alter or erasing sensitive data put away in a database. These queries are executed to harm severe damage to the back-end database.

v) Stored Procedures -As we know each database utilizes the inbuilt stored procedures to avoid injection attacks but the attackers can still penetrate into the system by embedding commands with a certified query with the goal that the two queries keep running as a single query.

There also exist various kinds of attacks that are executed on applications with the same intentions to steal crucial and personal data, defacement of application, modification in a database and many more to count. Accompanying table 5 clarifies SQLi attack types with sample query and with maximum damage placed on applications by each type when a SQL injection attacks become successful.

1.7 SQLi- Still a Breathing Threat

The dominant attack category of year 2019 was injection attack and surveys reported in year 2020 stated SQLi are on the top among injection attacks. The rate of increase in SQL injection attacks is 267% as compared to year 2018. This overview on web application attacks has shown different components of SQL injection attacks. The attack insight explains that SQL injections are on first spot on of the list of attacks with an attack level of 27%. As the development of web applications; information /data communications, server maintenance, etc are related to the IT industry. The percentage rate increments to 40 % i.e. SQL attacks are the most believed weapon utilized by attackers to extract information from the IT business centre. Indeed, even the companies related to transport like air tickets, railway or bus booking are likewise victims of SQL injection attacks with a portion of 32%. The public authority or the government organizations are not distant by this SQLi attack. With a share of 16% higher than any other classification of attack SQLi attacks are second after the information leakage.

Attacks on web applications published on June 26, 2019 shows that SQLi are

i) Top of the list on web application attacks (27%)

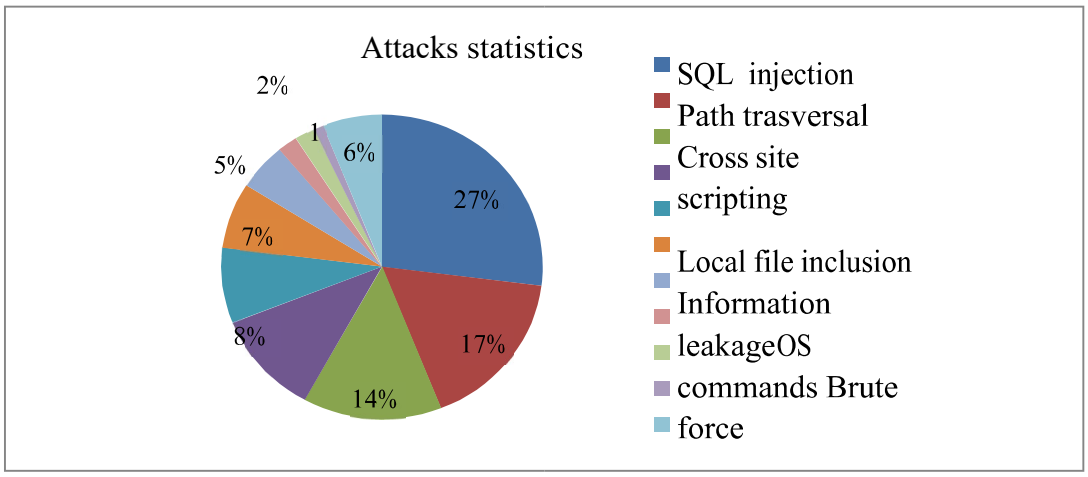

Fig 1.5: Top attacks on web application

ii) Top attack on IT industry (40%)

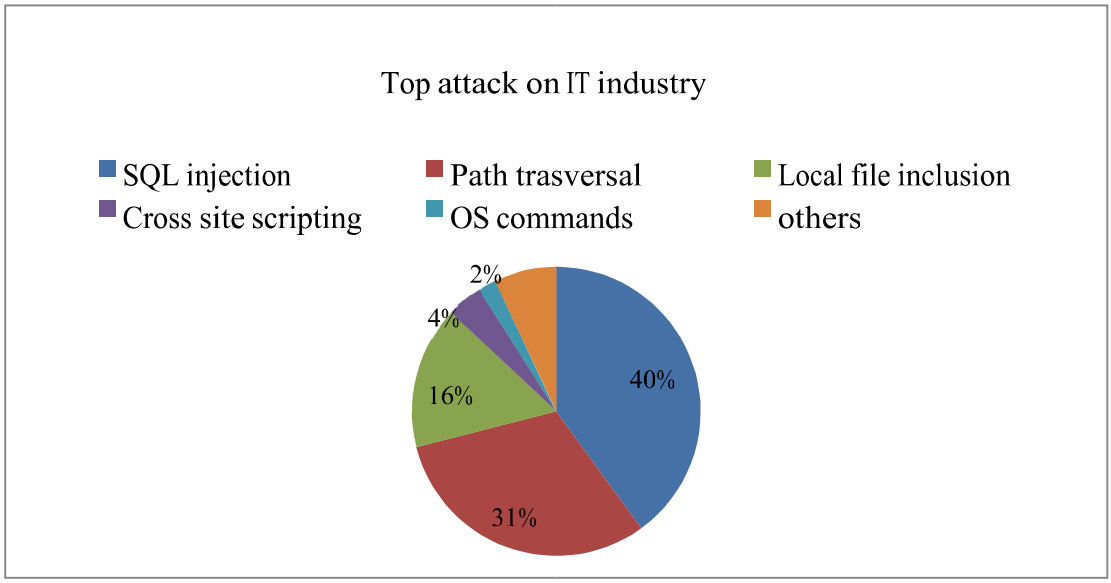

Fig 1.6: Top attacks on IT industry

iii) Top attack on transportation industry (32%)

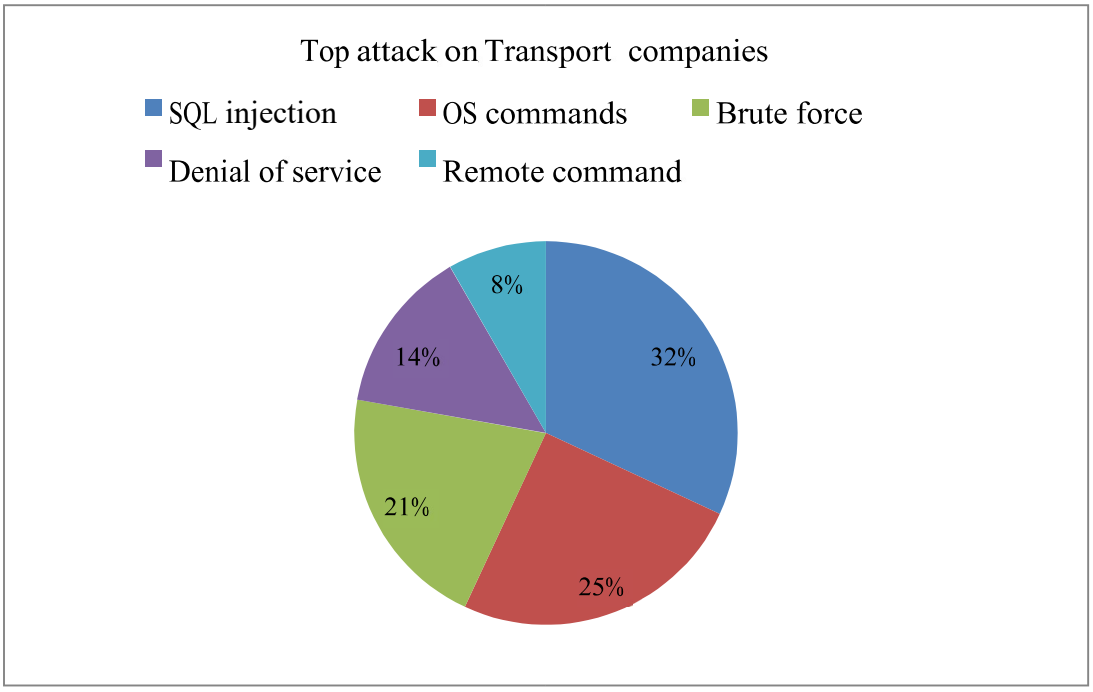

Fig 1.7: Top attacks on transport companies

iv) Second attack in Government (16%) after information leakage

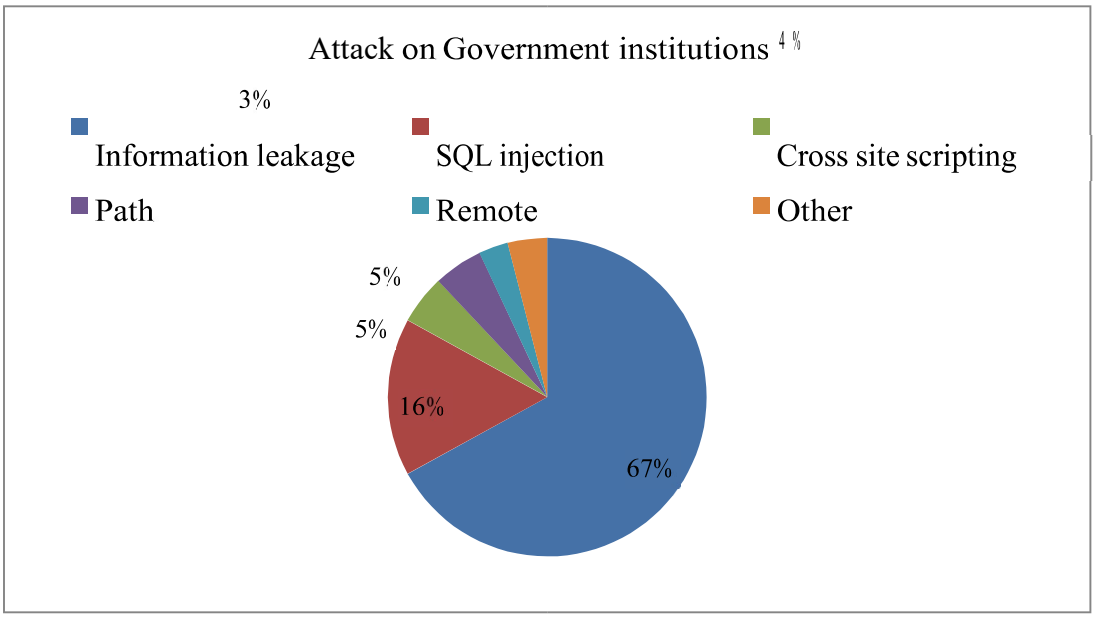

Fig 1.8: Top attacks on Govt. organizations

The intricacy of web application security is often underestimated. Knowing how to prevent and detect SQLi attacks is just a large portion, not the whole of the web application security combat. A large number of components become an integral factor with regards to secure the application, numerous of them are out of the application developers and security professional's direct control which makes the application vulnerable to attacks and SQLi continues on plaguing the present applications. Here are the best purposes/ reasons behind SQLi attacks are as yet widespread. In following subsection, some common practices are listed that describes why SQL injections are still valid and applicable for penetrating the database.

1.7.1 Validation Grounds

A) Challenges face by Developer team

This section lists some of the challenges faced by developer's team that creates issues like extraction or manipulation of data, insertion of malicious code etc. by SQL injection attacks.

i) A mammoth idea: SQL is naturally susceptible due to the syntax of SQL queries used for extracting data from the backend database whenever required. In this manner, to know about every SQL prevention and detection mechanism is a mammoth idea- As attackers are using new methods every day to invade the web application.

ii) Not possible to plug all weak links: Almost every web applications have numerous pages with various constraints that may be assaulted in many means. An application can have enormous numbers of probable loopholes/weaknesses. Every time it is not possible for the developers to secure and plug all loopholes presents. A solitary error can become the path of destruction.

iii) Aware of input: The most common reason for successful of SQL injection are vulnerabilities related to non sanitization of input fields. Irrespective of the developer - Either experienced or naive developers both must comprehend the application inputs that access the application's database. This process reminds us to be aware from SQLi for each input.

iv) Team effort: The role and responsibility of application developers and application security persons are very much different and well defined in an organization. It is not possible for application developers to stay aware of regularly changing the attacker's strategies. So a team effort is required to accumulate strategies to keep away applications from issues like SQL injection attacks,

v) Managing old application: Some applications take long time to reach in final stage and many organizations are still managing some of which applications developed long ago. The situation got worse when a developer left the job or original code is not located then vulnerabilities present in that application are very hard to fix for developer team working presently.

1.7.2 Challenging Grounds

B) Challenges face by Security team

This section lists some of the challenges faced by security team that accumulate strategies to keep check on latest prevention and detection mechanisms from SQL injection attacks.

i) Assets lacking: A large amount of money is required for research and development of applying security measures in an organization which organizations don't do. This leads to lack of essential manpower & techniques to fix vulnerabilities. Finally it weakens the level of security

ii) Overburden in a day: Applying different Security measures to different applications makes the security team overburdened in an organization. The regularly managing an

overabundance of several applications and need to discover, organize and propose a security solution to cure them is just like a race with time as the opponent.

iii) Different opinions: For some application both manual and mechanized testing are required. Sometimes a situation occurs when the difference in finding the vulnerability by two testers varies, then it becomes very difficult for security programmers to propose a security measures for that application.

iv) Dependency: After proposing a security measure, it is the responsibility developer team to execute them but as security programmer's team have little control over the application development team so they don't need to confront the challenge alone, the developer team needs to help. Both teams should function as a group to avoid potential vulnerabilities and destroy the present one.

C) Innumerable Factors

With such a large number of offices, agencies, scientists, researchers, developers and designers are cooperating to look for insurance from SQL injection attacks yet it is as yet going strong. Numerous challenges are faced by many researchers, application and security developers, and other professionals etc. are looking for insurance from SQL injection attacks. Here some challenges are listed which can't be included under specific terms yet are extremely fundamental and basic reasons why SQLi attacks still works

i) No professional training required to perform SQL injection attacks. Creative guesswork about database schema and knowledge of SQL quires are required for injecting attacks.

ii) The advantage of flaws in the input validation logic of web components.

iii) Problem due to the extreme heterogeneity of the attack vectors.

iv) SQL injection attack is interpreted differently on different databases.

v) Lack of proper planning in the design of an application

vi) A vulnerable and outdated code samples used by programmers.

vii) No privacy in sharing important information

1.7.3 Recent SQLi Attacks

SQLi attacks are in news due to some unauthorized successful penetration in the database in the last few years (2011 to 2020) :

i) In June 2020, the IBM Emptoris Strategic Supply Management Platform is vulnerable to SQL injection.

ii) In March 2020, CISCO-Data Center Network Manager vulnerable to SQL injection.

iii) In Nov 2019, Oracle E-Business suite PAYDAY was critically vulnerable to SQL injection twice in the same year.

iv) In Sept 2019 Amazon Alexa application hacked by SQL injections.

v) In Aug 2019, an SQL injection flaw opened the doorway to the accounting database of Starbucks. vi) In March 2018, SQLi was the reason at IIT-Delhi for accessing student's data by a karela student.

vii) In November 2017, over 40,000 customer's details including bank account are leaked at Hetzner, South Africa.

viii) In August 2016, a hacker shows how you can steal bitcoins using SQLi ix) In December 2016, a group of Russian hackers accessing the US election voting machine using SQL vulnerabilities.

x) In May 2015, the personal and sensitive information of 53000 clients of the world trade organization was stolen.

xi) In October 2015, around four million customer's information on credit cards and their passwords of a British company was revealed by hackers using SQLi attacks.

xii) In August 2014, SQL injection exposes the database of wall street journals xiii) In October 2013, around $100,000 were looted by hackers from Sebastian ISP / Banks via SQLi attacks. xiv) In July 2012, the very famous hacking scenario, confirmation of four million accounts of yahoo

In June 2011, the Citigroup database was exposed by an SQL injection attack revealing personal details of more than 200000 customers.

1.7.4 Threat to Emerging Technologies

The presence of flaws in these applications and regular attacks by attackers to penetrate the backend database decreases the efficiency and security of organizations. As per Symantec's and akamai's latest Internet security risk report says that in excess of 400000 assaults are actualized every day against webapplications. Further, it additionally reasons that 76% of sites are defenseless against various assaults. The yearly analysis report by Akamai's security shows that 84% of the live web applications are prone to SQLi attacks. Akamai's security third-quarter report wraps up the information that SQLi attacks & XSS

(cross-site scripting) are the most perilous danger at present. The new advances like BigData which utilize NoSQL technology likewise succumb to Injection attacks. Indeed, even in recent research, the SQLi attacks are likewise turning into a threat to cloud computing [25] environments also.. These attacks persist to dominate since 2007. In nutshell, the SQLi still holds on as it exists on its first day. The analysts/ researchers are still in chase of its countermeasures. The following graph 44% of total web attacks are SQL injection attacks only

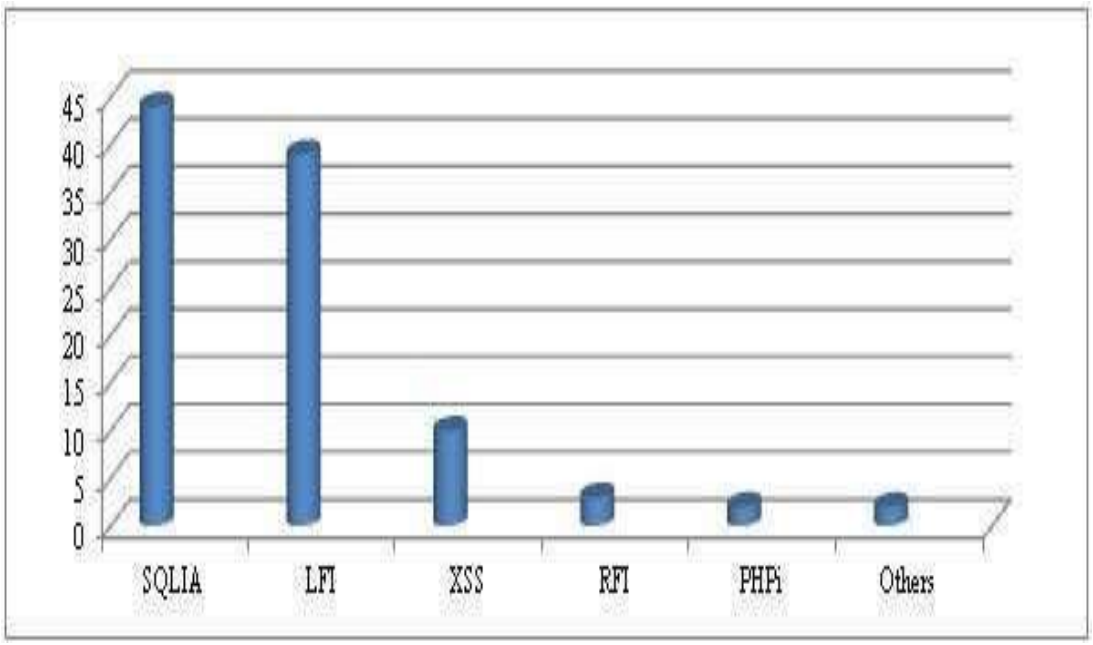

Fig. 1.9 Web Application Attack Frequency

1.8 Motivation for the Research

Every organization communicating or providing services owns a web application for information sharing with its consumers. Even individual people have a web application for business and personal use. Now a day's web applications (web apps) have been becoming a powerful tool for web users over the internet. Whether it is daily routine bank work or sending some messages over social media network? It provides the ease of data communication. Over the internet, these friendly web applications are exposed to many risks every day. With the change in type or content/data of web application, the risk associated with web application also varies- as these applications include sensitive data and information. Web applications are feeble on the way for many attacks. The attackers are keenly interested in a rudimentary component of this these applications i.e. the backend database. Thus, the security of web application databases is a hot research area. For

providing security many methods can be applied like preventing and detecting mechanisms for keeping attackers away from stealing sensitive, personal & vital information of users.

1.9 Problem Definition

SQL injection attacks (SQLi attack) exist as a noteworthy threat to a web-based application with their underlying databases. These attacks achieve the right of entry to web application database servers by means of the assistance of SQL queries, placing applications, confidential data, business strategies and financial records at risk. SQLi attacks are rewarding for attackers as there exists a flourishing ebony market (dark web) that exchanges an ample range of purloined advanced digital information as credit/debit/cash card information, individual points of interest in Bank-accounts, PAN and UID records. It is demonstrated by the surveys conducted every once in a while by two esteemed organizations Open Web Application Security Projects (OWASP) & Web Application Security Consortium (WASC) working in the web application security domain that SQLi attacks are a practical risk to a web application. SQLi attacks have stayed in the rundown of the best three attacks since 2000. Numerous analysts and researchers have proposed methodologies to counter SQLi attacks, including code reviews, tools for detection and prevention, vulnerability scanners and positive tainting and loads of mechanisms. The entire staff well-thought-out, SQLi attacks remains an intense threat. The threat is not constrained to conventional databases like RDBMS but rather additionally exists for new era Big data's NoSQL technology. These SQLi attacks also provide a way to attackers to get into Big data technologies like Hadoop, Hive and Map Reduce. The challenge before the application developers and researchers is to analyze the risk postured by these applications. An appropriate approach to conquer with SQL injection attacks is to conclude the risk level prior to the execution of web applications on the internet. Generally, risk analysis is the estimation of the degree of threat. It is applied as a preventive measure that estimates the risk level due to the presence of vulnerabilities and their potential impact. This detection of risk level helps in taking further decisions in mitigating, accepting or transferring the risk. Web applications can be secured by computing risk level prior its execution on the internet. It must be gained that to play out the risk analysis for the security of web application is an urgent need of great importance. This research work is driven to a risk analysis approach which is an imperative stride towards the security of the web application database.

1.10 Main research contribution

The following are the main contributions of the research work:

- Experimentally computed values are provided for the probability of attack.
- Introduction of naval metrics for analyzing some features like complexity of attack, data on web application, resources compromised & security features applied by web applications.
- Final estimation of risk value associated with the web application is evaluated by means of modified formulation.

New formulation is proposed based on risk value for determining risk level and risk ranking of a particular web application.

Chapter II

Review of Literature

Introduction

The review of literature is a basic and deep assessment of research already made. In other words- the research in a particular area permits the researchers to analyze the motives for seeking in a specific area. A decent survey grows up to the purpose of choosing a research material. It is a collection of the manuscript which intends to survey the basic purposes of current information including meaningful discoveries just as hypothetical and practical offerings on a specific area. These reviews are derived sources so can't be accounted for unique or new contribution. It is just helpful to gather more data for the investigation of the problem that could prompt an ideal framework arrangement. This part incorporates a short acquaintance of nature utilized with build up the exploration matter and connected effort completed in the area of SQL injection prevention and detection mechanisms and analysis of risk associated with web applications.

2.1 Review of SQLi Prevention and Detection Mechanisms

A decade ago numerous researchers had given the answers about the solutions for SQL injection attacks. The security mechanism includes both Prevention methods (static & Dynamic) and Detection methods (static & Dynamic) are suggested in manual and automated form by researchers. This section reveals the significance of some security mechanisms. Here a very recent period is considered for the year 2010 to the year 2020 for discussion. An inclusion or exclusion criteria is applied over 72 research papers searched. After the execution of this criterion only 52 studies were supposed to be fit for reviewing. The following table depicts the review of selection

Constraints	To Include	To Exclude
Language	Only English language article	Other than the English language
Article since	2004 to 2019	Past to 2004

Research Type	Research articles, white-papers and extract from books	Policy documents and not directly related research papers
Range	Manuscripts that discuss previous work mainly focused on SQL injection prevention and detection mechanisms	Manuscripts other than SQL injection prevention and detection mechanisms

Table: 2.1 Constraints of inclusion and exclusion

2.1.1 SQLi prevention mechanism

Prevention mechanism blocks the opportunity of executing SQL commands for exploiting backend database by identifying loopholes/vulnerabilities present in the code of web application. The best prevention mechanism includes sanitization of user inputs, parameterized queries by using prepared statements, query transformation, hashing techniques, best defensive coding practices. The prevention analysis techniques are listed with mechanism used in following table.

Analysis Technique	Mechanism
Static Analysis	AFGSSS[30],Java Static tainting[28],Pixy [29],ADMIRE[31]
Dynamic analysis	Clustering technique[34]
Hybrid Analysis	WebSSARI[27],Hybrid method[33]
Grammar based violation	CANDID [26],
Program based	MDWAA[35] D-wav[32]

Table: 2.2 Prevention analysis techniques

Mechanism	Principle	Advantage	Limitation
AFGSSS[30]	Analysis based on secure programming	No modification in code base and also provides code suggestion report	Server side application and works only at run time

Java Static tainting[28]	Negative tainted Data flow	Provides report of code suggestion	Server side application and works only at run time
Pixy [29]	Negative tainted Data flow	Provides report at coding time	Static analysis method
ADMIRE[31]	Grammar based violation	Provides report of code suggestion	Static analysis method
WebSSARI[27]	Negative tainted Data flow	No modification in code base and also provides code suggestion report	Server side application and works only at run time
CANDID [26]	Grammar based violation	Dynamically provides the response and works at run time	Modification in original code is required
MDWAA[35]	Input signature	Programmed based anomaly detection	Provides report only after testing
D-wav[32]	Grammar based violation	Client side application with dynamic analysis method	Provides report only after testing

Table: 2.3 Advantage and limitations of prevention mechanism

2.1.2 SQLi detection mechanism

The detection mechanism detects the code vulnerabilities present in the web application. The detection is done in both a static and dynamic manner. The detection analysis techniques are listed with mechanism used in following table.

Analysis Technique	Mechanism
Static Analysis	CVSID[45],ADMIRE[49]
Dynamic analysis	AProSec[41],IAAT[47]
Black box testing	WAVES[50],SecuBat[40]
White box testing	MUSIC[43]

Grammar based violation	Tautology checker[48], Sania[42], Prepared Statement[51], SQLInjectionGen[46]
Signature based violation	Viper[44]

Table: 2.4 Detection analysis techniques

The best detection mechanisms include testing the code before compilation; other methods include black-box testing with dynamic analysis. The mechanism with report generation at run time and no modification in existing code are highly recommendable. The detection mechanism with principle, advantage and respective limitation are given in following table.

Mechanism	Principle	Advantage	Limitation
CVSID[45]	Grammar based violation	Server side application	Escape the response
ADMIRE[49]	Grammar based violation	Server side application	Escape the response
AProSec[41]	Input signature	Client and server side proxy application	Escape the response
IAAT[47]	Output signature	Black box testing and dynamic analysis	Detection only at testing time
WAVES[50]	Output signature	Black box testing and dynamic analysis	Detection only at testing time
SecuBat[40]	Output signature	Client side application	No report generation
MUSIC[43]	Learning based anomaly detection	White box testing and dynamic analysis	Detection only at testing time
Tautology checker[48]	Grammar based violation	Server side application with code suggestion	Detection at run time only
Sania[42]	Grammar based violation	Client and server side proxy application	Response time is very high

Prepared Statement[51]	Grammar based violation	Follow secure programming	No report generation
SQLInjectionGen[46]	Grammar based violation	Static and dynamic analysis	Modification in base code required
Viper[44]	Output signature	Black box testing	Detection only at testing time

Table: 2.5 Advantage and limitations of detection mechanism

Researchers and engineers are continuously working for developing the security mechanism for these injection attacks. The following table depicts that SQL injection attacks are still the hot topic of research as the prevention and detection from these attacks are proposed for the year 2017 to year 2020.

Sl.No.	Feature	References
1	SQL injection Literature review & survey	[57][58][60][67][69]
2	SQL injection Prevention mechanism	[59][64][66][70][74][75]
3	SQL injection Detection mechanism	[55][56][61][62][63]
4	SQL injection Prevention & Detection mechanism	[52][65][68][71][72][73][76][77]

Table: 2.6 SQLi prevention & detection mechanisms

From the above-reviewed literature, it is concluded that the development of prevention and detection mechanisms are not sufficient for blocking and mitigating these SQL injection attacks. As web applications are attacked every day with new versions of attacks. But the devolvement process and usage of web applications can't be stopped by fearing of SQL injection attacks. One solution to this problem is the computation of Risk associated with a web application. This should be computed in advance prior to the web application being made available on the internet. After the quantity of risk labeled with a web application, the owner of a web application can decide, among the three options available i.e. first one is, go with the risk, the second is to mitigate the risk and the third one is to have insurance with the third party and then proceed with the risk. This all depends on the quantity of risk associated and the will of the web application owner. So in the following section, the review of risk analysis of web application is presented.

2.2 Risk analysis

Risk is constantly comprehended as an opportunity to an attacker or we can state the risk is the chances of an attack on an application. Risk analysis is the identification & assessment of various levels to some threat under consideration. Here SQL injection attacks are under scanner and regarded as the risk. Emblematically a risk is a damage that can happen to our assets (here web application database). The magnitude /extent of a threat is calculated as an impact. Risk analysis is a preventive measure for finding the bona fide portrayal of threat. Increasing no of attacks each hour makes the risk assessment essential for backend database security. It is a procedure for considering possible risks and figuring out which is a good number noteworthy for a specific attempt. Formative which type of risk needs to deal & finding the ideal procedure for alleviating risks is regularly a natural and qualitative process? A target perspective of the risk analysis exertion requires the expansion of a risk analysis methodology.

2.2.1 Risk analysis methodology

The analysis of risk value or risk estimation has two approaches. One provides the risk value in a numeric form related to each type, generally known as Quantitative analysis. On the other hand, other provides risk value in the subjective form, normally called as qualitative analysis. Both are very complex processes.

However, the risk value in numerical form provides better results in making final decisions and comparison analysis. That's why quantitative risk assessment methodologies are picking up the pace in recent times. There are numerous strategies created by analysts and vendors both in qualitative and quantitative nature. Some methodologies are also proposed that uses both qualitative and quantitative risk analysis at a time like. These method provides the results in both subjective as well as in numeric manner. In this section, we will break down the current work. A brief description about the previously proposed approaches used for risk analysis, are tabled in this here.

Sr.No.	Approach	Property	References
1	Quantitative analysis	Provides the results in numeric values	[84,85,86,87,88,89,90,91,92,93]

2	Qualitative analysis	Provides the results in subjective values	[94,95,96,97,98,99,100,101,102]

Table: 2.7 Approaches for risk analysis

2.2.2 Risk computation

Numerous theories about models and framework are tabled point by point no theory is by all accounts consummate. Each has its own particular leverage and detriments. Some models are in nature which produces data in a subjective form. Be that as it may, for better analysis, we require numerical information which can be just be given by other methods. In 2001, National Institute of Standard and Technologies (NIST-USA) has given general formulation for Risk analysis in information security domain, symbolically (R= P*I)
Risk = Probability of threat event* Impact of that event, These two variables probability (likelihood) and impact are defined by researchers now and then in their own manner and compute the value of risk associated with applications. Considering the research based on above mentioned computational method. The table depicted below shows some methodologies\models that uses standard risk computation formulation computational approach provided by NIST. These methodologies uses various computing method based on standard formulation and computed the risk associated with the web application in their own manner. In this section detailed examination is done to consider and recognize how to apply or broaden the current existing model/ framework and different metrics to think about and develop new risk assessment strategies.

Reference No.	Authors	Computational approach	Methodology
[104]	H C Juh et al. (2010)	Risk = probability of (single vulnerability) * exploitation threat	Likelihood is calculated from markov modelling and Impact values are taken from CVSS dataset

[105	]Hui Guan et al.(2012)	SV= (A2 *Wa *T 2 *Wt *V 2 *Wv)1/2	Potential risk will be evaluated based on the results of asset, threat and vulnerability analyses using proposed security vector.
[90]	Hossain Shahriar et al.(2014)	R=(importance of vulnerability*occurrence of vulnerability)*(payload type* severity of payload)	Tools are used for find the value of probability and impact
[106]	Mouna Jouini et al.(2015)	M(s, D) = Vs ◦ PFRs ◦ Cs ◦ Ps, w, 4 step process based on orthogonal dimension of security threat	Risk level is based on three matrices and vector probability theory
[107]	Sharma et al (2016)	R= probability of a particular attack * Impact of that particular attack	The status of risk is measured in subjective values- low to very high.
[108]	Yuandong Cheng et al. (2016)	Risk= function of (Risk probability and risk effects)	The probability and impact of the risk estimate the risk factor. In this estimation entropy weight coefficient provides the weight
[109]	Daljit Kaur et al.(2017)	Risk = Likelihood of an adverse event * Impact of the adverse event	Value of probability is assumed and impact value is taken from CVSS
[92]	Umesh Kumar	R= Frequency of attack found* impact on CIA, Values are derived from CVSS to compute frequency and impact	Tools are used for determine severity level. CVSS is used calculate the value of impact

	Singh et al.(2017)		
[110]	Mansour Alali et al (2018)	R= Probability of success * impact of successful attack is computed through Fuzzy	Four factors namely vulnerability, threat along with likelihood and respective impact are the basis of assessment
[111]	Stephen Hart et al.(2020)	R= Likelihood of attack* impact of attack, values are computed from analyst opinion	LINDDUN threat categories are used for probability and impact. The assessment is subjective to the expertise of the analysts

Table: 2.8 Risk computation methodologies

The methodologies discussed above, provides analysis in both Qualitative and Quantitative mode. These are used for decision-making procedures and can be practically applied for risk estimation. Some methods are introduced at the design stages of the application whereas some focus on asset identification after the development of the application. Few have stressed the involvement of staff like network and infrastructure for risk analysis purposes. On the other side of the coin, most of the existing methodologies are based on already existing topology/too/s/framework/methodology i.e. just the modification of already existing qualitative and quantitative approaches. For executing some methods qualified, prepared and experienced individuals are required for operation. Most of the techniques have not covered all the types & subtypes of SQL injection attacks for computing risk value.

2.3 Challenges in Risk Analysis

2.3.1 Probability computation

The major problem with probability computation is, it is computed with respect to vulnerability. Like presence of vulnerability, existence or frequency of vulnerability, existence of vulnerability on other vulnerabilities and many more but what if probability is

to be computed for an unknown system. Also some consideration to attack types and related severity of attacker must be given. Every attack type has different level of severity. Some are used for extracting the data and others are used for injecting the code only. So these attack types can't be treated same in probability computations.

2.3.2 Impact computation

The issues with impact computation is, it is estimated only by damage potential to backend data of web application or in terms financial losses to the organization. It must be understood that not all attacks are performed for extracting data – some are attempted for education purposes or for fun, sometimes attackers want to leave their mark and many more. The query used for attacking any application depends on the knowledge and experience of attacker. In nutshell consideration to capability of attack and intention of attacker must be given for the computation of impact.

2.4 Research Gap

To overcome the limitations of existing methods for risk analysis, this research work will design an efficient and practical methodology for estimating the risk associated with a web application. Some identified research gaps as follows:

1. Probability is evaluated in terms of vulnerability only – Consideration to attack type is limited.
2. Impact is estimated in terms of financial loss and damage potential to assets only-- Consideration of factors like capability of attack & intention of attacker are not included in computing impact.
3. Most of the researchers have used the already derived value of probability (likelihood) and impact from previous computed dataset and tables.
4. Some researchers have also assumed the value of probability and impact depending on the nature of attack and web application

2.5 Research objective

In this research work, new risk analysis methodology is proposed and performance is analyzed by different set of values. After analyzing the research gaps, four research objectives are formed as follows:

1. To propose a new Risk Analysis methodology for estimation of risk associated with a web application.
2. To propose modification in existing risk analysis formulation.
3. To provide an experimental value for probability computation.
4. To provide a new mechanism for impact evaluation of a web application.

Chapter III

Security of Web Application Database

Introduction

The focus of this research work is security of web based applications with backend database by estimation of the risk value associated with web application.

Internet based web applications are part of our daily routine from paying bills to transferring money or from booking movie tickets to air tickets; not only organizations uses these applications to provide facilities for their clients but individuals are also using these applications for their personal use like writing their blogs, recording their videos for education/coaching, selling their home made product etc. While registering or signup for a web application-personal details like name, address, date of birth, mobile no etc are required. Even some application that provides facility for online payment transaction, offers to store the details of credit/debit card for next time fast transaction. This crucial and sensitive information is stored in backend database of web application for verification of the credentials. This stored information has to be secured, otherwise personal and confidential data of individual or the organizations is at the risk of exposure. The internet based applications are becoming the foundation for online businesses enterprises by providing effortless access to information. These types of applications save precious time, vital cash and considerably efforts. These web applications are feeble on the way for injection attacks. The attackers are keenly interested in a rudimentary component of these applications i.e. backend database. SQL injection attacks (SQLIA) exist as a vital danger to the web based application. These injection threats accomplish the privilege of passage to information servers by methods of SQL queries, putting applications, classified information, business systems, and monetary records in danger. Injection attacks made by SQL queries steal the digital data and sell on the dark web by the attackers. The digital data like payment cards issued by banks along with personal bank details, individual card used for income tax purposes and card used for unique identity numbers like Adhar card etc.Thus, security of web application databases is a hot research area by preventing attackers from stealing sensitive, personal & vital information of users. An appropriate approach for securing web application database from SQLi is to estimate the risk value prior to the execution of applications on internet. Hence it concludes that analyzing the

risk of web based application for providing security is an urgent call for of enormous significance.

3.1 Risk Analysis

The risk is continually understood as a chance to attacker or it is the odds to attack on web based application. Symbolically a risk is a harm that can happen to data storage systems or backend servers of web application. The analysis of risk is the assessment of different degrees of threats. For finding the true blue depiction of threat, it is applied as preventative measures. The expanding number of attacks on web applications every hour makes the risk estimation necessary for web application security. It is a system for taking into consideration of probable risks and making sense of that is a decent number essential for a particular attempt. The requirement for arranging and finding a perfect method for analyzing risks is ongoing cycle. This research work is driven for analyzing the risk i.e. risk analysis methodology - a basic step in the direction of the security of backend database of web application. When web applications are developed, some risks are inherited due to some factors like-copying of code, lack of programming knowledge, lack of technical staff etc. carries a risk value associated with it. After that, some additional risk factors are also associated like- complexity of attacks faced by applications, non-applicability of security measures, heterogeneity of data present on applications etc. So the computation of these associated risk values is essential before the application is available for the end-user. The risk that resides in the application makes it vulnerable and allows the attacker to penetrate into the web application for data extraction. This strategy is anticipated as enchasing the security of web applications. The risk analysis methodology has three steps.

In first step: Inherent risk present in the web application is computed i.e. basic structure of application is exploited using vulnerable features like: login field, search field, comment field, URL of web application or any other input field.

In second step: After computing inherent risk, additional risk is computed i.e. analyzing some features like complexity of attack, data on web application, resources compromised & security features applied by web applications.

In third step: Final estimation of risk value associated with the web application is evaluated by means of modified formulation. Risk level and risk ranking are also computed during this step. The research contribution of this work is as follows:

i) Experimental value is provided for the computation of attack wise probability.

ii) New metrices are derived for efficient impact evaluation.

iii) Generalized Risk range and risk ranking of web applications are specified according to the risk associated with them.

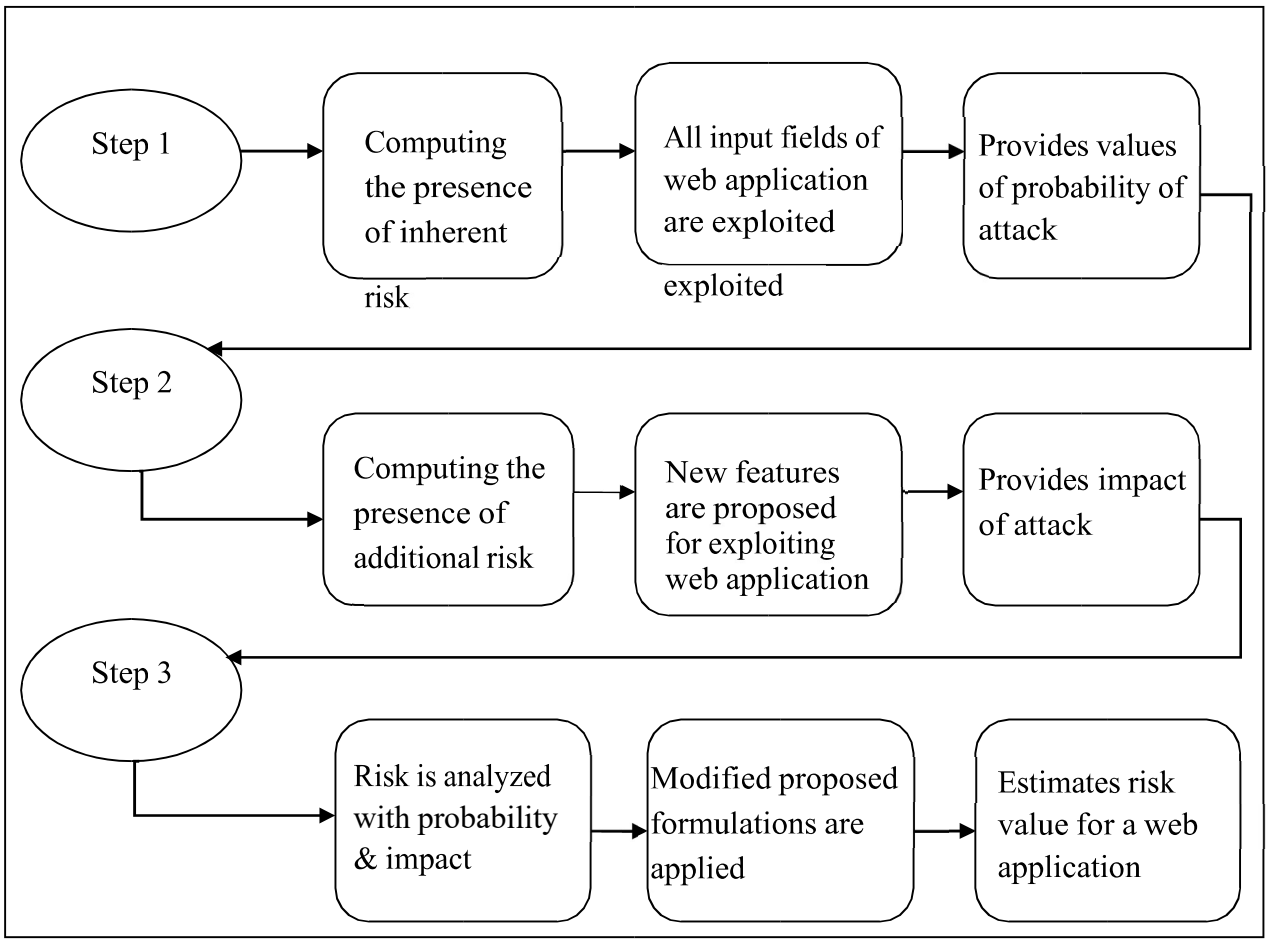

Fig 3.1: Steps for risk analysis

3.2 Proposed methodology

Risk analysis is a preventive measure that assesses threat. It is the function of probability (likelihood) and impact.

o Probability: a particular attack is successfully performed from a pool of attacks. o Impact: measure of damage to assets by that particular attack.

Hence, it is the estimation of risk due to any given threat, Here SQL injection is the identified threat and a risk analysis is a method to compute this threat. In 2001, National Institute of Standard and Technologies (NIST-USA) has given general formulation for Risk analysis in information security domain [103].

Risk = Probability of threat event (attack) * Impact of that event

Symbolically, R= P*I

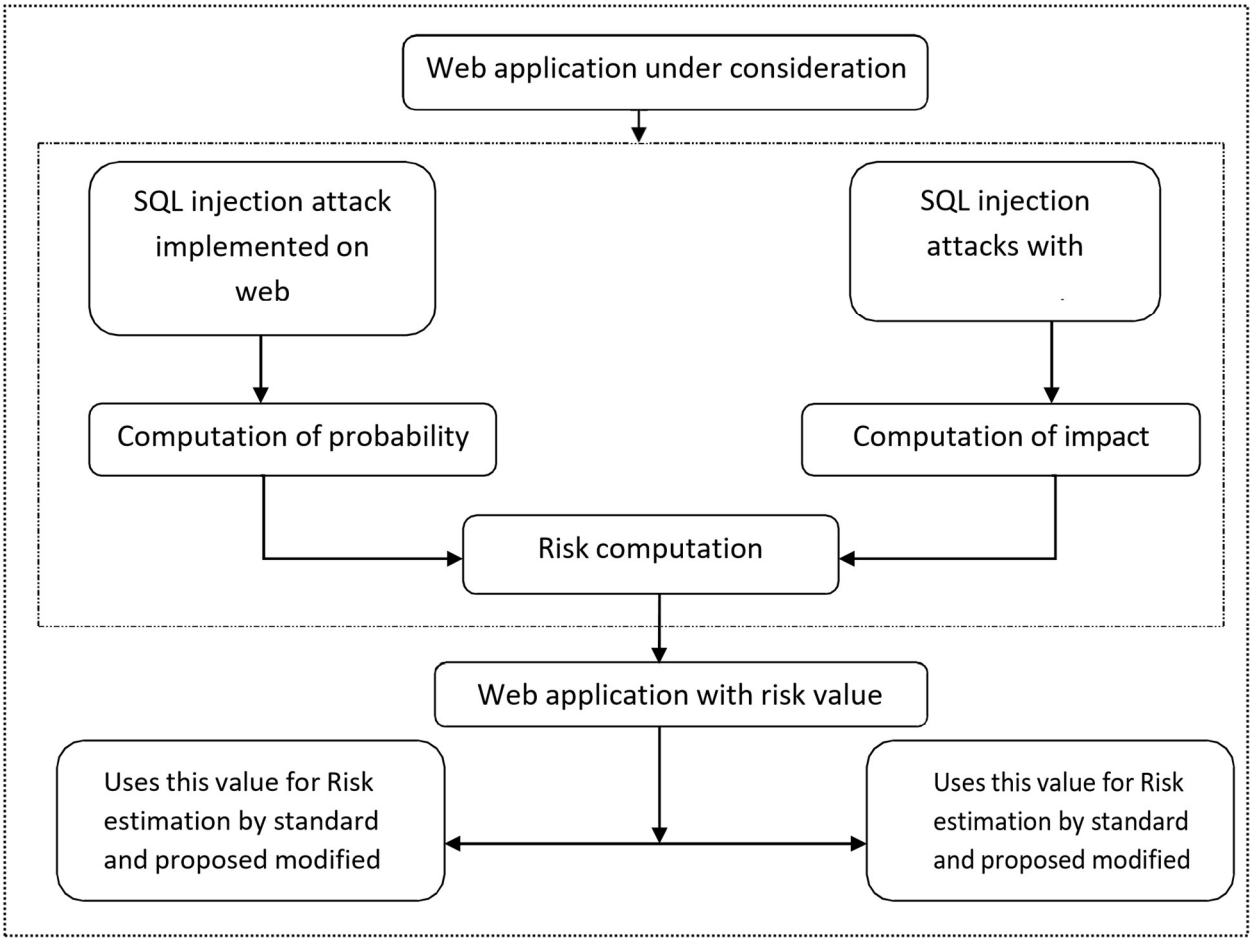

Fig 3.2: Proposed methodology

These two variables (probability & Impact) are defined by researchers now and then in their own manner and compute the value of risk associated with applications. This research support the statement that SQL injection has wide range of attacks and each attack poses a different stage of loss to crucial data stored at server of web based application that depends upon severity of individual attack. The formulation is modified in accordance with various types of SQL injection attacks. Proposed modified expression:

$$R= \sum 1^{N} (Pi * Ii)$$

i is ith type of SQLi attack, R= Computed risk, N= Types of SQLi attack

3.3 Computation of probability

From literature review it is observed that the value of probability used by researchers are based on vulnerability only, like- ease of discovery of vulnerability, ease of exploit of

vulnerability, awareness about vulnerability etc.The value of probability recommended by vulnerability scoring systems are based on severity of vulnerability relative to other vulnerability and attacker must have knowledge about weakness of the target system/application. What if attack is to be executed on unknown web application or target system and no previous knowledge of vulnerability present? This indicates the need of discussion and here are some analyzed points:

i) Vulnerability remains as it is until it is exploited -It depends on the nature and severity of attack that how vulnerability is treated? ii) It also depends on the attacker's capability and choice of query used for attack. So there is need for probability computation from attacker point of view that is based on attack type and their relative severity

In web programming technology, many technologies are used for developing web sites and applications. As on today most of the web is in PHP, ASP and JSP.Some other programming languages (Python, ruby etc.) and technologies (Ajax, word press etc.) are also used but we are considering only the former ones. Although unauthorized access (Manually or with tools) made on any web applications is illegal and punishable act but for the sake of experimental analysis attacks were performed. Most attacks which were based on SQL query manipulation were performed manually with the knowledge of SQL queries. An experimental analysis is performed on a single PHP application (say abc.net) & a JSP application (say xyz.com) .Each SQL injection attack is performed 10 times on as single web application. Web applications developed in PHP can be of different categories. The most viewed/popular categories of web applications are E-commerce, Social media, Portfolio, Informational, blogs and Photo sharing. The web applications selected for SQL injection attacks are from these categories such as E-commerce and gift gallery are from E-commerce category, social networking is from social media type, portfolio web applications include career guidance and IT organizations, photo sharing includes a picture gallery and informational display comprises educational institute web applications. The following table shows the success rate of these attacks performed. The success rate may vary with the ability of attacker and the type of query selected for attack.

Sl.No	SQLIA Types	Total attempt performed	Success rate	Success rate (In %age)
1	First order	10	6	60

2	Second order	10	4	40
3	Lateral Injection	10	2	20
4	Blind injection	10	6	60
5	Database focused	10	5	50

Table 3.1: Reference table for PHP application

Sr.No	SQLIA Types	Total attempt performed	Success rate	Success rate (In %age)
1	First order	10	01	10
2	Second order	10	----	----
3	Lateral Injection	10	02	20
4	Blind injection	10	04	40
5	Database focused	10	---	----

Table 3.2: Reference table for JSP application

This process was repeated on 180 web applications (80- PHP, 50-JSP & 50-ASP) developed in different programming languages over a period of one year and following table shows the experimental result of attacks. The attacks were based on PL/SQL query, performed manually with previous knowledge and experience.

Sr.No	SQLIA Types	Success rate on PHP technology	Success rate on ASP technology	Success rate on JSP technology
1	First order	48%	30%	10%
2	Second order	30%	15%	-------
3	Lateral Injection	16%	30%	15%
4	Blind injection	50%	58%	40%
5	Database focused	62%	25%	--------

Table 3.3: Success rate on different technologies

Success rate of attacks means that an application (Say coded in PHP) is vulnerable to a particular type of attack. The table depicted above analyzed that lateral injection has only success rate of 16% whereas against database it is 62%. So the range of damage to web application varies with the nature of attack. Also, PHP applications are found more vulnerable as compared to ASP & JSP applications because the success rate of attack of PHP applications are more than others in all type of attacks.

3.4 Method of probability computation

As 70% of the web is still in PHP, so here focus is on applications programmed in PHP Only. Extending our experiment to next step, some specific types of web application in PHP are attacked manually to find out their probability of attack.

Eighty (80) web applications are of PHP technologies having eight (8) different categories i.e. we have ten (10) applications of each category.

A randomly selected web application is attacked ten times for a single attack to compute the probability of attack. The table displays the probability of attack according to web application type.

Web applications	First order	Second order	Lateral order	Blind injection	Database focused
E-commerce	----	----	----	0.1	0.2
Social networking	0.8	0.5	----	0.7	0.5
IT organizations	----	0.8	0.6	----	0.8
Government deptt	0.8	----	----	0.2	0.6
Career guidance	----	----	----	0.2	0.8
Education institute	0.8	0.1	----	0.8	0.7
Gift gallery	0.8	0.2	----	0.6	0.8
Picture gallery	0.2	0.1	0.2	0.8	0.4

Table 3.4: Attack wise probability values

Now the 10 web applications of a single category are attacked with all attack types and the process is repeated for all categories. The average value for all the application against attacks are tabled below

Web applications	First order	Second order	Lateral order	Blind injection	Database focused
E-commerce	----	----	----	0.2	0.2
Social networking	0.8	0.7	----	0.7	0.4
IT organizations	0.1	0.8	0.4	0.1	0.8
Government deptt	0.6	----	----	0.4	0.6
Career guidance	0.2	----	----	0.3	0.7
Education institute	0.8	0.1	----	0.8	0.7
Gift gallery	0.8	0.2	----	0.6	0.8
Picture gallery	0.6	0.2	0.2	0.8	0.6

Table 3.5: Category wise probability values

The values of probability are almost same in these two tables which conclude that the method of computing probability by manually attacking the web applications is providing results. Applying the above computation method to some vulnerable applications used previously in research paper. In this research paper SQLi attack types are not properly considered. So considering the same vulnerable applications and computing the probability of attack according to the redefined SQL injection attacks classification. The following table depicts the results

Applications	First order	Second order	Lateral injection	Blind injection	Database focused
PHP address book (Address & contact manager)	0.6	0.4	0.1	0.7	0.4
Serendipity (Blog management)	0.4	0.2	0.2	0.7	0.3
PHP- fusion (Content management)	0.3	0.3	0.1	0.5	0.3

Table 3.6: Probability of vulnerable application

Some observations are made during the experimental analysis of attacking the web applications. These observations lead to form new guidelines, analyzing results, defining role & responsibilities for end users and organizations.

3.4.1 Minimum measures for securing web application

Some standard coding measures must be followed while coding/programming a web application. Like

i) Input query filtration
ii) Code sanitization
iii) Restrict authorization to access database
iv) Error handling mechanism
v) Concrete defined schema of database

Similarly, some standard security measures must be followed while applying security mechanism for a web application. Like

i) Enable firewall
ii) Login audit enable
iii) Prevent defacement of web application by infinite requests
iv) Protected resources available to authorized persons/system only
v) Prevention of SQL arbitrary command execution & system level SQL command execution

After performing all types of SQL injection attacks on various web applications developed in different technologies. Based on observations some guidelines are prepared for application programmer for secure development of web applications.

Attack types	Minimum measures for secure web application		
	Security measures	Coding measures	Remark
First order	-----	i)Input query filtration	Whatever security measures may be applied or not applied at all, if input query filtration is not done – First order injection attacks are always possible.
Second order	i) Execution of arbitrary command execution stored in tables.	i) Sanitization of inputs ii) User inputs as SQL string not to be saved.	These minimum measures must be considered
Literal injection	i) Less use of stored procedures ii) Restriction in violation of system level commands	i)No dynamic schema generation	These minimum measures must be considered
Blind Injection	-----	i)Sanitization of URL ii)Malicious query filtration	All security measures are applied or none, if coding measures are not followed-Blind injection attacks are always possible.

Database focused	i) Enable firewall ii)Login audit enable ii) Prevent defacement of web application by infinite requests iii) Protected resources available to authorized persons/system only iv) Prevention of SQL arbitrary command execution & system level SQL command execution.	i) Prevention from system level attack ii) Prevention from runtime file/ procedure creation iii) No dynamic schema generation	As most of SQL injections are performed to extract information from database. So all standard security measures and coding measures must be applied.

Table 3.7: Suggestive guidelines for programmer

3.4.2 Behavior of web applications

During this analysis, a behavioral study is also carried out. This study concludes that, it is almost impossible to attack high level organization's web applications as they have very effective and efficient security mechanisms/technology/manpower. The same set of web applications which were attacked during experimental analysis was attacked again after a period of 30 days to study the behaviour of organizations.

Sr. No	Technology	Web application attacked	Success in attack	Web application closed	Web application improved	Available for attack again
1	PHP	80	57	7	2	48

Table 3.8: Behaviour study

It is observed that applications that were attacked belonged to medium, small and very small sized organizations. The medium size organizations have improved web application security mechanisms and were not available for attacks any more. For small and very small organizations, few had shown interest in securing applications by closing

more vulnerable applications but most of the applications are still available for attacks. The table below shows the study results. Based on behavioral study, some guidelines are suggestive for organizations: Progressing familiarity with risks and urging all team members to play a functioning part in keeping up security is a primary and necessary advance:

i) Resistance to excess: An audit system must be deployed which can alarm, on any new transaction in database.
ii) Role & responsibility: Role & responsibility of office/remote employee handling database must be very well structured and clearly defined. The authorization to database servers must be restricted.
iii) Code your code: In order to meet project delivery deadlines, sometimes already written code is used. Automatic scanner can be installed at organizations to avoid the downloading of already written code (which can be vulnerable),
iv) Realization: Simulation of some attacks with their consequences can be revealed to educate and test the knowledge of employees about the security.
v) Enforcing policy: Strong password policy must be enforced after sharing the awareness of password generation techniques and tool.

3.4.3 Security status

The success rate of attacks can be used to find out the security status of web application. When an application is prone to attacks, it means it is not 100% secure. So security status can be derived as, Security status = 100% - [percentage of successful attack]

Sr.No.	Web application types (PHP)	Success rate of attack	Security status
1	E-commerce	34%	66%
2	Social networking	26%	74%
3	IT organization	23%	77%
4	Government applications	16%	84%
5	Career guidance	12%	88%
6	Educational Institutes	24%	76%
7	Gift gallery	22%	78%
8	Pictures gallery	40%	60%

Table 3.9: Security statuses

Applications like E-commerce, Social N/w and IT organizations are bound to be more secure than other application but the above table shows some contradictions which displays that large percentage of security in career guidance and gift gallery than former applications ones because

i) By attacking web applications like E-commerce and social N/w, personal and sensitive data can be extracted which is more profitable to attackers.

ii) As these applications are combination of various services, probability of attack increases with different attack payload. Attack on any service makes the whole application insecure

Hence security status is less as compared to other applications.

3.4.4 Deliverables from probability computation

As the web designers/developers are aware of all coding practices, so can easy correlate with this method of probability computation. The web designers/developers can

1) Directly references the computed values
2) If not referenced, can compute probability by self using this method.
3) Suggestive guidelines for securing web applications.
4) Behavior of organization
5) Security status after the application is successfully attacker.

3.5 Computation of Impact

A lot of inconvenience is involved in measuring impact as so many factors are to be considered like: Same treatment to each attack, different value assets, dependency of one attack on another and many more, it is also not possible to include all the factors. Some selected factors are quantified with some inputs to compute impact. Four metrices are proposed to measure the impact due to these attacks.

1. Effort level of attack: This metric estimate the level of effort required by an attacker i.e. whether only little information is adequate or various skills is necessary. It can be measured in three values- Low, Medium and High. The value of effort for attack under consideration is fixed. Attack type like first order can be

effortlessly performed so its value is always low whereas lateral orders are very hard to implement so its value is always high.

2. Popularity of attack: This metrics estimate which attack is profitable to attacker i.e. whether an attack is competent of extracting various types of data that the application is carrying. It can be measured in three values- Low, Medium and High. The crucial data like Payment related or user information shows has high popular value as attackers are more interested in it, where any information providing web application gets low value.
3. Damage capability of attack: This metric identify the damage done to backend database i.e. whether the attacker has just delayed the outputs or deleted the whole database. It can be measured in three values- Low, Medium and High. Mostly attacks are done to steal the crucial information or shutdown the database but there are some other dimensions also to consider like attacks are also done for fun, learning/education purpose, to warn administrator or just to mark the presence etc. So the value of damage capability varies with the intent of attacker.
4. Damage control strategy from attack: This metric classifies the capability of damage control strategy i.e. whether the attack can enter all safety methods or none. It can be measured in three values- Low, Medium and High. If an attack can enter all safety methods applied at server end like: data encryption, system privileges and security audits along with firewall and malicious code filter then the value of this metrices is low. The medium value indicates that attack has just passed firewall and malicious code filter only. Finally high value signifies that attack is able to penetrate firewall only. The following table provides the description of metrices in tabular form

Metrices Explanation	Effort level of attack (ELA)	Popularity of attack (POA)	Damage capability of attack (DCA)	Damage control strategy (DCS)
Refer	Estimate the level of effort required by an attacker	Estimates which attack is profitable for attacker	Estimate the damage done to backend database	Estimates the resources compromised by a particular attack.

Description	Whether only little information is adequate or various skills is necessary	Whether attack under consideration can extract crucial and personal information from database	whether the attacker has just delayed the outputs or deleted the whole database	whether the applied security measures are sufficient as compared to attack performed
Assigned Values	Fix- for particular type of attack	Any value- Low, Medium or High	Any value- Low, Medium or High	Any value- Low, Medium or High

Table 3.10: Description of metrices

Now, how the Low, Medium and High value are assigned to each metrices?

The following table demonstrates the criteria of assigning values based on the description of the proposed metrices.

Metrices / Assigned values	Effort level of attack (ELA)	Popularity of attack (POA)	Damage capability of attack (DCA)	Damage control strategy (DCS)
Low	A Simple PL/SQL query is utilized and login form is available	Any informatory data	Provides limited access	If penetrated all Data encryption, system privileges and along with firewall and malicious code filter
Medium	Few queries are utilized & Http request manipulation	Encrypted data, product data & E-commerce, files and multimedia	Access to authorized functionality/ resources	If passes though firewall and malicious code filter

High	Combination of PL/SQL queries are executed as a whole, writing procedures etc.	Payment related or user information	Full access as authorized user	Passes through Firewall only

Table 3.11: Assigning values to metrices

To compute impact a mechanism is applied, after placing all types of SQL injection attacks with the proposed metrices.

The value for the metrics Effort level of attack (ELA) is fixed for each type of attack. Whereas the other three metrices can have any values from Low, Medium or High.

The resultant impact values must be output in terms of Low (L), Medium (M) and High (H). The following example table contains some arbitrary values chosen for demonstration purpose that displays some random set of values for metrices and impact.

Metrices SQLi types Example table	Effort level of attack (ELA)	Frequency of attack (POA)	Damage capability of attack (DCA)	Damage control strategy (DCS)	Impact
First order(FO)	L	M	M	H	L
Second order(SO)	H	H	L	H	M
Lateral injection(LI)	H	L	L	H	H
Blind injection(BI)	M	H	H	L	M
Database focused(DF)	H	H	H	L	L

Table 3.12: Example table with subjective values

The significance of L/M/H of impact values is evaluated in the table given below

Impact	Evaluation
Low	No damage to assets(backend database)
Medium	Some damage to DB/Damaged database can be recovered from backup database.
High	Database is exposed/deleted, can't be recovered again

Table 3.13: Evaluation of impact

From here impact can be computed in two bearings by Qualitative assessment and Quantitative assessment. Qualitative assessment has subjective list like low–med–high and Quantitative assessment hold numeric values. Sometimes subjective values do not reflect the conclusion in great manner, whereas numeric values are better in analysis and reveal better results. Considering these aspects a mathematical method is used for impact computation.

Let's consider a numerical range for Low, Medium & High values. For Low, the value is between 0.01 - 0.36, for Medium 0.37 - 0.65 and for High 0.66 – 0.99. [These values are referenced from CVSS table]. The numerical ranges of chosen values are referenced from the CVSS framework for communicating the characteristics and severity of software vulnerabilities. Likewise, the numerical range is based on proposed metrics with different characteristics that collectively form the rating and severity score. The proposed metrices that includes- effort level of attack and popularity of attack form the basis of computing severity while damage capability of attack adjusts the severity based on access to resources and damage control strategy checks the security environment of a web application. Effort level of attack reflects the ease and technical means by which vulnerability can be exploited. Popularity of attack reflects the characteristics that changes with attack type and with web application too. The damage capabilities are a sign of the consequences of a successful exploit. The damage control strategy reflects the presence of security controls relative important to environment of web applications. The severity score is computed on the basis of values assigned based on their definitions.

Placing the corresponding range of numeric value in place of subjective values as discussed above in example table. The numeric value will replace the subjective value and new table is depicted as under:

Metrices	ELA	POA	DCA	DCS	Impact

SQLi type	A1	A2	A3	A4	B
FO	0.01-0.36	0.37-0.65	0.37-0.65	0.66-0.99	0.01-0.36
SO	0.66-0.99	0.66-0.99	0.01-0.36	0.66-0.99	0.37-0.65
LI	0.66-0.99	0.01-0.36	0.01-0.36	0.66-0.99	0.66-0.99
BI	0.37-0.65	0.66-0.99	0.66-0.99	0.01-0.36	0.37-0.65
DF	0.66-0.99	0.66-0.99	0.66-0.99	0.01-0.36	0.01-0.36

Table 3.14: Example table with numeric values

In above table, metrices ELA is symbolically represented as A1 and other metrices as A2, A3 & A4 respectively. These four metrices along with five SQL injection types collectively form a matrix of (5*4), say matrix A. The impact is also represented as a matrix of (5*1), say matrix B.

[Note: Here the matrix is presented in rectangle box instead of using matrix symbol, as the name of rows and columns has to be mentioned for better understanding]

As the value of Matrix A lies in some range; hence it will compute the impact in range too. To particularly define that range, the values for matrix A is chosen in two scenarios i.e. two different set of values for matrix A. For standardizing the values of matrix, instead of considering all range values, just two extreme values are chosen. The two Low values are (0.01 & 0.36), for Medium the values are (0.37 & 0.65) & for High the values are (0.66 & 0.99). This will compute the range of impact with lower value and higher value.

Lower values for matrix A		Higher values for matrix A	
Low	0.01	Low	0.36
Medium	0.37	Medium	0.65
High	0.66	High	0.99

Table 3.15: Value of matrix A

The sample matrix A (values from table no. 3.14) displays the lower and higher values of matrix

Metrices Attack type	A1 ELA	A2 POA	A3 DCA	A4 DCS
FO	0.01	0.37	0.37	0.66

SO	0.66	0.66	0.01	0.66
LI	0.66	0.01	0.01	0.66
BI	0.37	0.66	0.66	0.01
DF	0.66	0.66	0.66	0.01

Metrices / Attack type	A1 ELA	A2 POA	A3 DCA	A4 DCS
FO	0.36	0.65	0.65	0.99
SO	0.99	0.99	0.36	0.99
LI	0.99	0.36	0.36	0.99
BI	0.65	0.99	0.99	0.36
DF	0.99	0.99	0.99	0.36

Table 3.16: Sample matrix A with lower and higher values

3.6 Method of impact computation

The impact computation is proposed with the help of matrix multiplication. To find the value of Impact a general matrix Equation, A.X=B is applied.

HERE matrix A contains metrices value (A1, A2, A3&A4) w.r.t. attack types and B is the desired impact. The values of both X & B are unknown. The steps of this computation are:

i) First, a matrix X of additional features is introduced. ii) Secondly, it is placed with matrix A for the computation of matrix B.

3.7 Parameters of X

Matrix X has four parameters X1, X2, X3 & X4.All these parameters has one to one mapping with the proposed metrices. Like X1 has mapping with ELA and X2 has with POA. Similarly X3 is related to DCA and X4 has mapping with DCS. A feature is also associated with each metrices which characterizes these metrcies. This feature will help us in evaluating the values of parameters of X.

Parameters	Metrices	Features
X1	ELA	Complexity of attack

X2	POA	Profit from attack
X3	DCA	Resources compromised by attack
X4	DCS	Security measures faced by attack

Table 3.17: Parameters of X

3.7.1 Placing value of matrix X

Value of X lies in some range and that range of X is provided on the basis of experimental analysis (manually attacking web applications) which includes nature of attack, Severity of attack, injection method and some other observations.

The values of X are not absolute values. These are based on analysis and not on any empirical formulation. So any numeric value can be chosen for matrix X say between (0 & 1).Consider the range of matrix X with three value- Min(minimum), Max(maximum) & greater than Min but less than Max.

For the value of X1- The effort level of attack is mapped by parameters X1 with feature complexity of attack. As X1 varies with the attack type (i.e. query/queries used for injection), it will affect the range of X1.The FO injection attack are simple to execute i.e. complexity of attack is less so range is minimum. Second order, literal and database focused are very hard to implement means more complex as compared to first order, so their range is maximum. As blind injection are based on hit and trail method so its value is greater than minimum but less than maximum. The following table displays the range of X1.

Attack type	Complexity of attack	Range of X1
FO	Simple queries used to implement	Min
SO,LI& DF	Complex queries used to implement	Max
BI	Hit & Trail implementation	Greater than Min but less than Max

Table 3.18: Range of X1

Some conventional values are used for the providing the numerical values to range of X1.Here the values for X1 are chosen between 0 & 1.(End users are free to choose any values).The table below shows the numeric value of X1 with respect to the range.

	FO	SO	LI	BI	DF
X1	0.10	0.90	0.90	0.50	0.90

Table 3.19: Value of X1

For the value of X2- The popularity of attack is mapped by parameter X2 with feature profit from attack. As X2 varies with attack type i.e. the attacks that are able to extract meaningful data are popular among attackers. A first order injection attack provides informatory data so they are less profitable to attacker with minimum range. Whereas second order, lateral and database focused are more profitable as compared to other attacks. Extraction of data from blind injection attacks are based on trails. The following table displays the range of X2.

Attack type	Profit from attack	Range of X2
FO	Usually informatory data	Min
SO,LI& DF	Very crucial & personal data	Max
BI	Extraction varies on execution	Greater than Min but less than Max

Table 3.20: Range of X2

The table below shows the numeric value of X2 with respect to the range

	FO	SO	LI	BI	DF
X2	0.10	0.90	0.90	0.50	0.90

Table 3.21: Value of X2

For the value of X3- The damage capability of attack is mapped with parameter X3 with feature resources compromised. X3 varies according to severity of attack type i.e. the amount of damage done by a particular attack to backend asset: like database of web application. A first order injection attack can do little damage with minimum damage capability. The potential of blind injection varies with attack execution. Remaining types of attack can completely damage the assets so these are tagged with highest range. The following table displays the range of X3.

Attack type	Resources compromised	Range of X3
FO	Some damage	Min
SO,LI& DF	Complete damage	Max

BI	Damage depends on execution of attack	Greater than Min but less than Max

Table 3.22: Range of X3

The table below shows the numeric value of X3 with respect to the range

	FO	SO	LI	BI	DF
X3	0.10	0.90	0.90	0.50	0.90

Table 3.23: Value of X3

For the value of X4- The damage control strategy is mapped with parameter X4 with feature security measures.X4 varies according to specific characteristics of web application like: E- commerce application bound to have more security than a gift gallery application. To prevent the execution of first order injection attacks, little measures are needed. As other category of attacks are complex attacks prepared with multiple queries, requires complete measures. The measures rate for blind injection attacks are between first order and other attacks. The following table displays the range of X4.

Attack type	Security measures	Range of X4
FO	Some measures	Min
SO,LI& DF	Complete measures	Max
BI	Between some and complete measures	Greater than Min but less than Max

Table 3.24: Range of X4

The table below shows the numeric value of X4 with respect to the range

	FO	SO	LI	BI	DF
X4	0.10	0.90	0.90	0.50	0.90

Table 3.25: Value of X4

Finally placing all the values of X1, X2, X3 & X4, the matrix X with their respective values in accordance with the range is depicted in table given below

Parameters	FO	SO	LI	BI	DF

X1	0.10	0.90	0.90	0.50	0.90
X2	0.10	0.90	0.90	0.50	0.90
X3	0.10	0.90	0.90	0.50	0.90
X4	0.10	0.90	0.90	0.50	0.90

Table 3.26: Values of matrix X

3.7.2 Placing value of matrix A

After placing value of matrix X, it's time to place the value in matrix A. For A1, the values of ELA (effort level of attack) are fixed for every type of attack. The values of POA (popularity of attack), DCA (damage capability of attack) & DCS (damage control strategy from attack) depend on probability of attack.

Attack type	Probability	POA	DCS	DCA	Remarks
FO & BI	0.1 -1.0	0.66	0.01	0.66	If even only once, attacker enters the system, any attack can be possible.
SO,LI & DF	0.1-0.29	0.01	0.66	0.01	After entering the system, it's attacker capability to what extent data can be extracted
	0.30-0.59	0.37	0.37	0.37	
	0.60-1.0	0.66	0.01	0.66	
All types- FO & BI, SO,LI & DF	0.0	0.01 (Min.)	0.66 (Max.)	0.01 (Min.)	No attack is performed i.e. security measures applied doesn't allow the attack to execute

Table 3.27: Values for matrix A

For FO (first order) & BI (Blind injection)- if even once these attacks are executed successful out of many attempts, the attacker can enter the system and can perform any operation. In case of SO (second order), LI (lateral injection) & DF (database focused) - value of the attacks with low probability values are less popular among attackers so their POA is also low. The attack with more probability of attack value signifies weak security measures are applied for web application. The value of DCA depends on DCS, if more security measures are applied then it naturally weakens the attack capability and in other

words if less security measure are applied, more attacks bound to execute that makes the value of DCA high. Considering all these aspects the table above supplies the respective values for matrix A.

3.7.3 Quantification of impact

For matrix A, consider an example of social networking web application with lower set of values. The values in matrix A are placed according to the probability of each type of attack on this application (discussed earlier in probability computation). This will be multiplied with matrix X, having standardized values for each type of attack.

To compute the impact values. The product of matrix A (5*4) and matrix X (4*5) is to be carried out.

As per matrix multiplication rule, the first Row of matrix A (w.r.t FO) will be multiplied with first column of matrix X (w.r.t FO). Similarly second row of matrix A (w.r.t SO) will be multiplied with second column of X (w.r.t. SO).

In this computation, the values of our interest are only diagonal elements that provide the values related to each attack type. Computing the multiplication of matrix A with matrix X

	A1	A2	A3	A4
FO	0.01	0.66	0.66	0.01
SO	0.66	0.37	0.37	0.37
LI	0.66	0.01	0.01	0.66
BI	0.37	0.66	0.66	0.01
DF	0.66	0.37	0.37	0.37

Table 3.28: Matrix A with lower set of values

FO	SO	LI	BI	DF
0.10	0.90	0.90	0.50	0.90
0.10	0.90	0.90	0.50	0.90
0.10	0.90	0.90	0.50	0.90
0.10	0.90	0.90	0.50	0.90

Table 3.29: Matrix X with values The resultant matrix shows the bold elements of interest

0.06	0.62	0.62	0.34	0.62
0.13	1.20	1.20	0.67	1.20
0.13	1.20	1.20	0.67	1.20
0.17	1.53	1.53	0.85	1.53
0.13	1.20	1.20	0.67	1.20

Table 3.30: Lower impact values

Similarly placing higher set of values in matrix A and multiplying with same values of matrix X, the resultant matrix is

0.27	2.43	2.43	1.35	2.43
0.29	2.64	2.64	1.47	2.64
0.29	2.43	2.43	1.35	2.43
0.29	2.69	2.69	1.49	2.69
0.32	2.95	2.95	1.64	2.95

Table 3.31: Higher impact values

The impact computation started with subjective values, now not only quantification but a range is also by provided with numeric values. The impact values with range for web application under consideration, shown in following table:

Attack type	Lower range	Higher range
First order (FO)	0.13	0.27
Second order (SO)	1.59	2.64
Lateral injection (LI)	1.20	2.43
Blind injection (BI)	0.85	1.49
Database focused (DF)	1.59	2.95

Table 3.32: Quantification of impact

The lower and higher value of impact against all types of attacks, for all web applications are computed by placing appropriate values in matrix A and multiplied with matrix X, are depicted in table given below

	Lower values					Higher values				
Web applications	FO	SO	LI	BI	DF	FO	SO	LI	BI	DF
E-commerce	0.06	1.20	1.20	0.85	1.79	0.27	2.43	2.43	1.49	2.99
Social networking	0.13	1.59	1.20	0.85	1.59	0.27	2.64	2.43	1.49	2.95
IT organizations	0.19	1.79	1.79	1.17	1.79	0.33	2.64	2.43	1.81	2.95
Government deptt	0.13	2.37	2.37	1.17	1.79	0.27	3.56	3.56	1.81	2.99
Career guidance	0.19	2.37	2.37	1.17	1.79	0.33	3.56	3.56	1.81	2.99

Education institute	0.13	2.37	2.37	0.85	1.79	0.27	3.56	3.56	1.49	2.99
Gift gallery	0.13	2.37	2.37	0.85	1.79	0.27	3.54	3.54	1.49	3.56
Picture gallery	0.19	2.37	2.37	0.85	1.79	0.27	3.56	3.56	1.49	2.99

Table 3.33: Values of impact for each web application

The matrix A is the input provided by end-user. For easy computation of Impact, Matrix B is quantified with some range. The computed value of impact can be referenced for PHP web applications

3.7.4 Deliverables from Impact computation

1) Informed/assistance decision: The proposed mechanism can compute the impact for any attack type.
2) Aid: Acts as an aid to web administrator/Organization to compute impact without hiring any security professionals.
3) Quantification: The value of X is quantified for end user to facilitate the computation of Impact.

3.8 Risk computation

Risk computation is a function factors Probability (P) and Impact (I) are discussed in earlier sections. The Proposed modified expression for risk computation is:

$$R = \sum^{N} (P * I) \quad R=$$

Estimated risk value, i is i^{th} type of SQLi attack and N=
Types of SQLi attack

This expression will provide the numeric value which shows the range of minimum and maximum value of the risk associated with a particular web application. The illustration by proposed modified formulation is given in next chapter.

3.8.1 Proposed methodology for Risk level and Ranking

On the basis of this risk computation, what decision is to be taken by end-user, it is discussed in this section. The end-user has three options:

1. Accept the risk: if application is attacked and whatever the outcome of these attacks may be, will go with this risk value.
2. Transfer the risk: if the application is exposed to some attacks, third party (insurance) will be responsible for the losses if the application is attacked.
3. Mitigate the risk: some prevention or detection mechanism must be applied to secure the application from incoming attacks.

A Generalized formula is proposed to navigate the end-user in taking decision out of these three options for the security of web application. The steps to determine the risk level and risk ranking are discussed as under:

In first step, Step size (smallest part of risk range) is computed by subtracting lower value of risk from higher values of risk & then it is divided by three (because we have three options) i.e. first step is determination of step size

I step: Determine the Step size = (Higher value – Lower value)/3

After the determination of step size, all the options can be explored in step II, III & IV

Steps	Formulation	Risk level	Preference
II Step	Lower value + Step size value	Risk level1 (RL1)	Accept risk
III Step	Value between (RL1) Risk level 1 & (RL3) Risk level 3	Risk level 2 (RL2)	Transfer risk
IV Step	Higher value – Step size value	Risk level 3 (RL3)	Mitigate risk

Table 3.34: Preferences from risk level

The above table describes the list of preferences with respect to the risk levels. For computing the range of risk levels another methodology is proposed to provide the generalized risk level and risk range. The gain from this methodology is it can be used for any type of attack and not limited to SQL injection attacks only. It can be also be utilized in analyzing the computational results in qualitative and quantitative form because the values of attributes- risk rating & risk range can be either in subjective or in numeric format.

The description attribute explains the value of risk according to the risk level and risk range. Like for RL1, computed risk value is in tolerable range and can be made available to end- users but with very little modification. The RL2 suggests that application can be made available on internet but need to be mitigated very soon to avoid attacks. RL3 signifies that web application is at high risk i.e. prone to attacks, so it is recommended to apply some mitigation mechanisms immediately. The following table displays the view of proposed methodology:

Risk level	Risk Rating	Risk range	Description
RL1	Low	Lower value + Step size	Risk is acceptable, some changes in coding which requires very less time
RL2	Medium	Between lower value and higher values	Risk is acceptable for some time only, need to plan remedy very soon
RL3	High	Higher value- Step size	Risk is not acceptable, requires remedy immediately

Table 3.35: Generalized risk level and range

Finally we can conclude that, the proposed methodology not only computed the risk value associated with a web application but also helps in deciding, what preference is to be followed. The decision of mitigating, accepting or transferring the risk depends on the end-user.

3.8.2 Deliverables from risk computation

This risk computation methodology is a maiden attempt to provide the protection of application's backend database from SQLi assaults. This attempt can be useful in many ways:

i) Can work as an input for a designing new security mechanism.

ii) Small organizations can use methodology- Saving significant money & resources. iii) Researchers can extend this work.

3.9 Risk index number (RIN)

Although the proposed methodology is simple, practical and efficient enough to compute the risk value of web application. But if any small organization or a start-up doesn't have any budget to hire technical or security professionals to know about the risk associated with their application, for that situation a risk estimation method is proposed. After the development of the web application and before placing that application on internet, the risk value can be computed by the proposed risk index number.

The behaviour of web application was discussed earlier which reveals that the SQL injection attacks are easily executed on web applications, that belongs to small size organization (SO) and medium size organizations (MO). The high level organization web applications are very difficult to attack as they have plenty of resources in terms of technology and manpower to secure their application i.e. they can easily secure their web applications from SQLi attacks. Here a mechanism in terms of risk index number (RIN) is proposed for web applications developed and maintained by small & medium sized organization. These organizations don't have separate budget for hiring security personal or purchasing tools for providing security to their web application. Keeping this in mind some general and standard checkpoints based risk index number is derived. This index number is helpful in deciding whether a particular application is secure to be uploaded on internet or not. RIN is a score based method in which some attributes are checked. For each attribute a score is added based on TRUE & FALSE. Finally the sum of attribute is computed to get the final Risk index number of each application. This index number has some range with minimum and maximum risk value. As already said the difference between small & medium level organization is the presence of resources. In this section two different mechanisms (one for small level organization & other for medium level organization) are proposed for evaluating RIN respectively.

3.9.1 Risk index number (RIN) for small organization

A score-based method is proposed for calculating the risk index number (RIN). For this, some assumptions are made considering values as 0.36 for Low, 0.65 for medium & 0.99 as high as in our proposed methodology. The range of risk will fall from 1.80 to 2.97, where 1.80 as minimum risk and 2.97 as maximum risk.

Step1: Check the application code for standard checking methods like Testing, filtering and sanitizing; as these must be applied on before executing the application live, if any above-

mentioned techniques are not properly executed then assume the output is FALSE otherwise TRUE. Some abbreviation of terms used in pseudo code of this method, Like CDT -code tested, CFS -code filtered and sanitized, AAR- admin access restricted and FW - firewall is enabled. If these methods are applied then value is 0 otherwise 0.36. Step 2: Partitioning of data into various categories which are based on causes of attacks i.e. in which attackers are keenly interested.

Data category	Description
PR Payment Related	Payment Related (information about user account no's, Credit/debit card etc),
UI User Information	User Information (information related to username, E-mail Id's, saved passwords etc.),
EL Encryption Logic	Encryption Logic (information related to encryption keys, lock hashes, recovery codes etc.)
FM Files & Multimedia	Files & Multimedia (information related to data files, audio/Video/image files etc.)
PD Product & Data	Product & Data (information related to E-commerce, word press, blue port etc.)

Table 3.36: Various data category

Here PR, UI, EL, is considered into high priority risk factor with value 0.99 and FM, PD is of medium priority risk factor with value 0.65. Data other than these categories are considered as low priority risk factor with value 0.36.

Attributes	TRUE	Score	FALSE	Score
Is code tested?	True	0	False	0.36
Is code filtered and sanitized?	True	0	False	0.36
Is admin access restricted?	True	0	False	0.36
Is firewall enabled?	True	0	False	0.36
Type of data	PR & UI	0.99		
	FMM ,ED & E-Commerce	0.65		

	Any other	0.36		
Are attacks manual? (AM)	True	0	False	0.36
No. of attempts for attack(NOA)	0 to 2	0.99		
	3 to 4	0.65		
	5 and more	0.36		
Consequences of attack(COA)	Low	0.36		
	Medium	0.65		
	High	0.99		

Table 3.37: Risk index for SO

Step 3: The attack method (AM) will decide the number of attempts (NOA) required for web application breach i.e. attacking the application at own end before made accessible by manual or automated method. For manual its value (0) is true otherwise false (0.36). When attack average varies from 0-2, just increase previous value of risk index by 0.99.When range reaches to 3-4 add 0.65 and when more than 5 attempts are required to break into database then add 0.36 to risk index value.

Step 4: The consequences of attack (COA) value varies from low, medium & high, i.e. from 0.36, 0.65 & 0.99 respectively. This value depends on damage done to assets of application. It will be decided by developer or programmer after seeing the consequences of attacks like: some data is breached or full data is compromised etc. Finally the level of security is checked in terms of Low, Medium and High and value is assigned accordingly.

The four steps are suggested for finding the final Risk index number. The values of attributes in terms of true & false are easily recorded in a table to compute the final risk index number and for better understanding the pseudo code for the same is also provided.

3.9.2 Pseudo code

Set RIN: =0

Input CDT, CFS, AAR and

FAD if CDT is not

true RIN: = RIN

+0.36 if CFS is not

true

RIN: = RIN
+0.36 if AAR is
not true RIN: =
RIN +0.36 if
FAD is not true
RIN: = RIN
+0.36
Input DC

If DC is

PR|UI|EL

RIN: = RIN

+0.99
Else If DC is
FM|PD RIN: =
RIN +0.65

Else

RIN: = RIN +0.36

Input AM

If Am is

automated

RIN: = RIN

+0.36

Input NOA

If NOA is between 0 to 2
RIN: = RI +0.99 Else If
NOA is between 3 to 4
RIN: = RI +0.65 Else
RIN: = RI +0.36

Input COA If
COA

is low

RIN: =

RI +0.36

Else If COA is

medium RIN: =

RIN +0.65

Else

RIN: = RIN +0.99

Print the RIN

3.9.3 Risk index number for Medium organizations

Medium sized organizations are more financially and technically sound in terms of available resources than small organizations. The application development team is capable of handling things like, testing and filtering of developed code, enabling firewalls etc. This team must take care of administrative rights for storing critical data, interactions with other systems, monitoring of log files, writing procedures and many more to count. As in these organizations time is more important, as every project has some deadlines and increase in development time of application can become a financial burden. Here a quick score based method is proposed based on YES & NO for computing final risk index number (RIN). The range will fall from 1.5 to 5.5, where 1.5 as minimum risk and 5.5 as a maximum risk. The steps are for above said method are explained below

Step 1 Check to see if admin access is enabled with proper rights.

ADMIN_RST_ACCESS, it can be either a yes or no

Step 2 finds out the system to system interaction

type SYS_INTR, which can be below listed

values IP_BASED = IP based limited access

SRV_BASED = only few services are allowed to

interact AUTH_BASED = authentication is required

to access NO = anyone one can access

Step 3 what type of procedures/sequences written in the database for querying?

PROC_SHAPE, it can be below listed values

NOT_WRT = procedures are not written

SECURE = written securely with proper sanitization/filtration without accessing lowlevel calls

CRET_SQL_DYN = procedures generate SQL queries dynamically (harmful)

ACC_SYS_CMD= procedures to call/access system commands such as shutdowns/file generation etc.

Step 4 the critical data we are storing such as credit-cards/banking or passwords are encrypted?

CRT_DATA_ENC, it can be either true or false

Steps 5 are monitoring of other system such as logs, login audits, and other activities are being in surveillance?

LOG_MNT_AUDIT, either it is true or not

Step 6 checks to see if backups or alternate systems are enabled in case of failure, data breach or any other unwanted exploit?

BKP_ALT_SYS, this value can be either true or false

Step 7 the other important factor is to check for third party services/ applications integration in our system. There are several ways for giving some control/access to them.

THRD_PRT_ACCESS, these values can be

LIM_CERT = limited resources access is given. They are certified and reliable UNK_SRC = they are from unknown sources, may or may not be authorized

NOT_AUTH = they are not authorized and access/privileges are not managed

Step 8 checks to see if our application is analyzed or tested by the security expert

(manually or with tools/framework)

TST_AUDIT, this value can be either true or false

The eight steps are suggested for finding the final Risk index number. The values of attributes in terms of YES & NO are easily recorded in a table to compute the final risk index number and for better understanding the pseudo code for the same is also provided.

Attributes	YES	Score	NO	Score
Admin access restricted	Yes	0	No	0.5

Limited sys to sys interaction	Yes	Authorized	0	No	1
	Yes	IP based	0.5		
	Yes	Service based	0.5		
SQL procedures written	Yes	Securely written	0.5	No	0
	Yes	Dynamically created	1		
	Yes	Dynamically created	1.5		
Critical data encrypted	Yes		0	No	0.5
Monitoring /logging audits enabled	Yes		0	No	0.5
Are backup system enabled	Yes		0	No	1
Third party access	Yes	Limited	0	No	1
	Yes	Unknown source	0.5		
	Yes	Not authorized	1		
Tested by security expert	Yes		0	No	1

Table 3.38: Risk index for MO

3.9.4 Pseudo code

1. Set RIN = 0
2. Input ADMIN_RST_ACCESS
3. If ADMIN_RST_ACCESS is not true RIN = RIN + 0.5
4. Input SYS_INTR
5. If SYS_INTR is AUTH_BASED RIN = RIN + 0
6. If SYS_INTR is IP_BASED | SRV_BASED RIN = RIN + 0.5

7. IF SYS_INTR is NO RIN = RIN + 1
8. Input PROC_SHAPE
9. If PROC_SHAPE is NOT_WRT RIN = RIN + 0
10. If PROC_SHAPE is SECURE RIN = RIN + 0.5
11. If PROC_SHAPE is CRET_SQL_DYN RIN = RIN + 1
12. If PROC_SHAPE is ACC_SYS_CMD RIN = RIN + 1.5
13. Input CRT_DATA_ENC
14. If CRT_DATA_ENC is not true RIN = RIN + 0.5
15. Input LOG_MNT_AUDIT
16. If LOG_MNT_AUDIT is not true RIN = RIN + 0.5
17. Input BKP_ALT_SYS
18. If BKP_ALT_SYS is not true RIN = RIN + 1 19. Else RIN = RIN + 0.5
20. Input THRD_PRT_ACCESS
21. If THRD_PRT_ACCESS is LIM_CERT RIN = RIN + 0
22. If THRD_PRT_ACCESS is UNK_SRC RIN = RIN + 0.5
23. If THRD_PRT_ACCESS is NOT_AUTH RIN = RIN + 1
22. Input TST_AUDIT
23. If TST_AUDIT is true RIN = RIN + 0

24. Else

    RIN = RIN + 1

25. Print RIN

Chapter-IV

Results and Analysis

Introduction

The scope of experimental results focuses on risk value associated with a web application under consideration. This consists of manually attacking the web application and computing the metrics mathematically. The biggest part of this experiment is to focus on securing the web application before executing it on the internet for end-users. The experiment results of this thesis are compared and analyzed with the results of already computed values. The overall objective of all experimental work is to evaluate the probability and impact of attack for efficient risk value estimation with the following goals:

i) Attacking the unknown web applications.

ii) Applying the maximum number of attacks for each attack type to get optimal value of probability.

iii) Applying all possible queries for extracting data from web applications.

iv) Attacking the web applications to get the probability and impact of an attack.

Applying metrics to compute the additional risk associated with a web application.

4.1 Experimental results and analysis

In this thesis web applications are attacked by all types of SQLi attacks. For this experimental work, a lot of unknown applications are attacked whose risk value is not already known or accessible in any dataset. Performing any type of attack on any web application is illegal to attempt. To figure out the risk associated with an unknown application, attacks are performed by SQLi attacks but solely for education and research purposes. The applications whose risk value is already known are used for comparing and analyzing the results. The name of all these web applications used in this experiment, that are attacked is not disclosed due to security reasons.

4.1.1 Elucidation by standard formulation

The computation of risk value depends of two factors, probability of attack (P) and impact of attack (I).The details of these are provided in previous chapter.

The standard formulation for risk computation is

Risk =Probability of attack * Impact of attack i.e. P*I

The table below depicts the values for probability of attack for application under consideration:

Web applications	First order	Second order	Lateral order	Blind injection	Database focused
E-commerce	----	----	----	0.1	0.2
Social networking	0.8	0.5	----	0.7	0.5
IT organizations	----	0.8	0.6	----	0.8
Government deptt	0.8	----	----	0.2	0.6
Career guidance	----	----	----	0.2	0.8
Education institute	0.8	0.1	----	0.8	0.7
Gift gallery	0.8	0.2	----	0.6	0.8
Picture gallery	0.2	0.1	0.2	0.8	0.4

Table 4.1: Attack wise probability

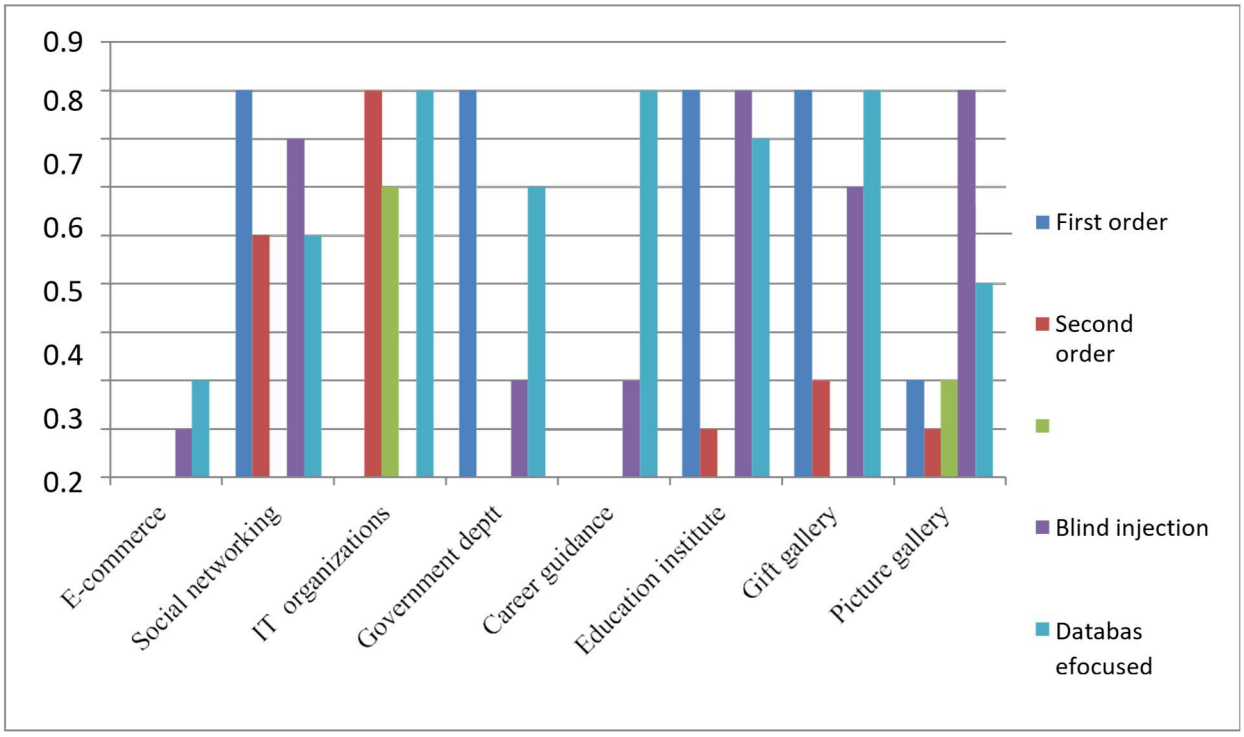

Fig 4.1: Respective probability of attack

Values for impact of attack are derived from quantification of impact. The following table displays the lower (L) and higher (H) value of impact computed for every web application under consideration.

Attack type →	First order		Second order		Lateral order		Blind injection		Database focused	
Web applications ↓	L	H	L	H	L	H	L	H	L	H
E-commerce	0.06	0.33	1.20	3.56	1.20	3.56	0.85	1.49	1.79	2.99
Social networking	0.13	0.27	1.79	2.99	2.37	2.99	0.85	1.49	1.79	2.99
IT organizations	0.19	0.33	1.79	2.99	1.79	2.99	1.17	1.81	1.79	2.99
Government dept	0.13	0.27	2.37	3.56	2.37	3.56	1.17	1.81	1.79	2.99
Career guidance	0.19	0.33	2.37	3.56	2.37	3.56	1.17	1.81	1.79	2.99
Education institute	0.13	0.27	2.37	3.56	2.37	3.56	0.85	1.49	1.79	2.99
Gift gallery	0.13	0.27	2.37	3.54	2.37	3.54	0.85	1.49	1.79	3.56
Picture gallery	0.19	0.27	2.37	3.56	2.37	3.56	0.85	1.49	1.79	2.99

Table 4.2: Lower & Higher value of impact

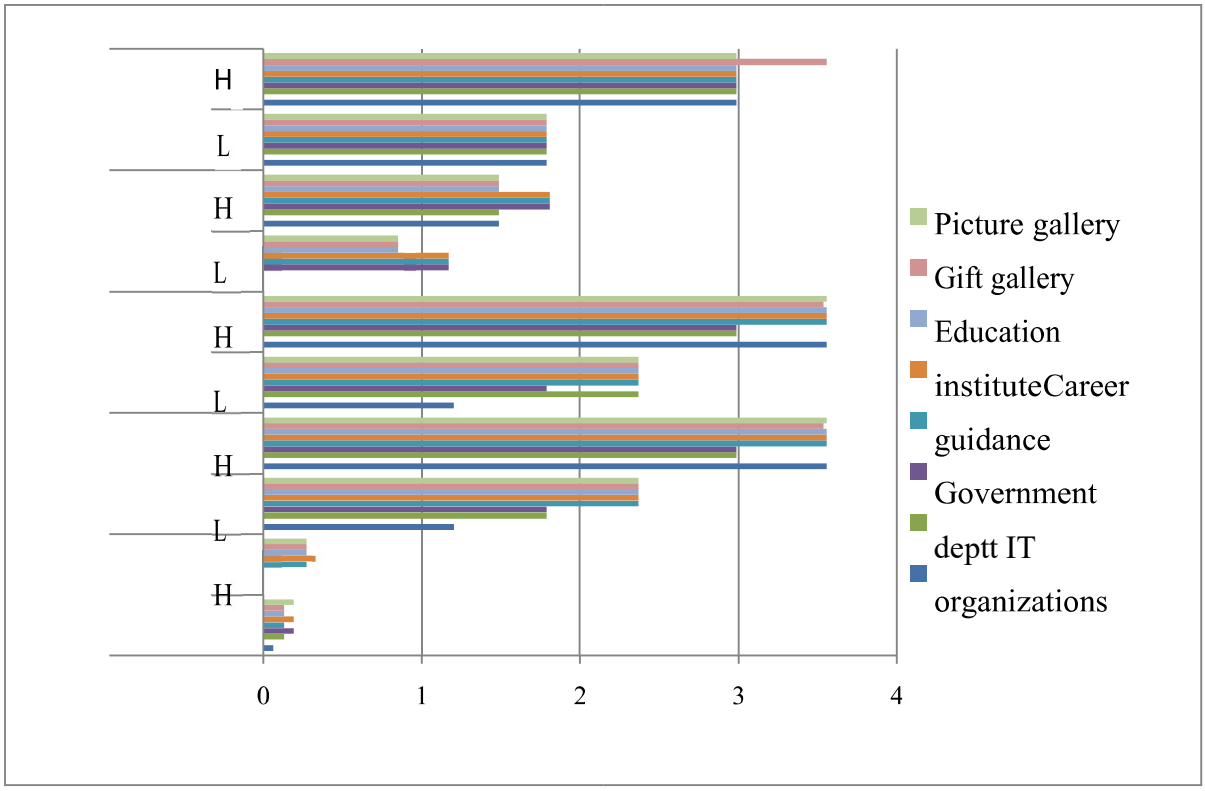

Fig 4.2: Range of impact

A graph from above table can be derived and used to compute the range of impact. The range of impact varies from 0.06 to 3.56.It also analyzes that second order, lateral injection and database focused has higher range of impact as compared to other attacks.

Risk is the function of probability and impact, so extracting the values of probability and impact from respective tables and calculating the risk associated with E-commerce web application. The following table depicts the values of risk:

Attack Type	Probability (P)	Lower Impact(I) value	Risk (P*I)
First order	--	0.06	--
Second order	--	1.20	--
Lateral order	--	1.20	--
Blind injection	0.1	0.85	0.08
Database focused	0.2	1.79	0.35

Table 4.3: Risk value with lower value of impact

This calculation provides the risk value according to the attack executed on web application. It also helps in deciding for which attack type remedy or security measures are required for the securing the web application backend database. The computed risk value varies with the values of probability and impact. The risk associated with higher impact values are tabled below:

Attack Type	Probability (P)	Higher Impact(I) value	Risk (P*I)
First order	--	0.33	--
Second order	--	3.56	--
Lateral order	--	3.56	--
Blind injection	0.1	1.49	0.14
Database focused	0.2	2.99	0.59

Table 4.4: Risk value with higher value of impact

Hence, the risk value is estimated by standard formulation. This computation helps in the analysis that the security measures applied at the application end are not enough to stop blind injection and database-focused injection attacks i.e. the security measures applied for first order, second-order and lateral injections are sufficient, now if any mitigation strategy or tools are required then it is only for former attacks. To conclude, this analysis helps in saving a significant amount of human effort and monetary budget requirements for the organizations.

4.1.2 Elucidation by proposed formulation

The proposed formulation is the extension of standard formulation. The methodology in this thesis considered SQL injection attacks as a set of attacks. Accordingly the changes have been proposed and the modified expression is given as below. The proposed modified expression:

$$R = \sum_{i}^{N} (P_i * I_i)$$

R= Computed risk i is i^{th} type of SQLi attack and N= Types of SQLi attack

To illustrate the computation by proposed modified formulation, consider the example of Social networking web application, the values of probability and impact for the same are

derived from previously calculated tables. The following table depicts the respective range of risk associated with this application.

Attack type	Probability (P_i)	Lower value Impact(I_i)	Higher value Impact(I_i)	Risk ($P_i * I_i$) Lower range	Risk ($P_i * I_i$) Higher range
First order	0.8	0.13	0.27	(0.8*0.13) 0.10	(0.8*0.27) 0.21
Second order	0.5	1.79	2.99	0.89	1.49
Lateral order	--	2.37	2.99	--	--
Blind injection	0.7	0.85	1.49	0.59	1.04
Database focused	0.5	1.79	2.99	0.89	1.49
	Total risk value	Risk $\Sigma(Pi*Ii)$		2.47	4.23

Table 4.5: Risk range of Social networking application

Similarly, the risk range of all web applications under experimental observations can be computed and are depicted in table given below:

Web applications	Risk range (Risk $\Sigma(Pi*Ii)$)	
	L	H
E-commerce	0.43	0.73
Social networking	2.47	4.23
Government deptt	3.93	6.57
Education institute	1.40	2.36
Career guidance	1.66	2.75
Education institute	2.26	5.48
Gift gallery	2.51	4.64

Picture gallery	2.12	3.49

Table 4.6: Risk range of web applications

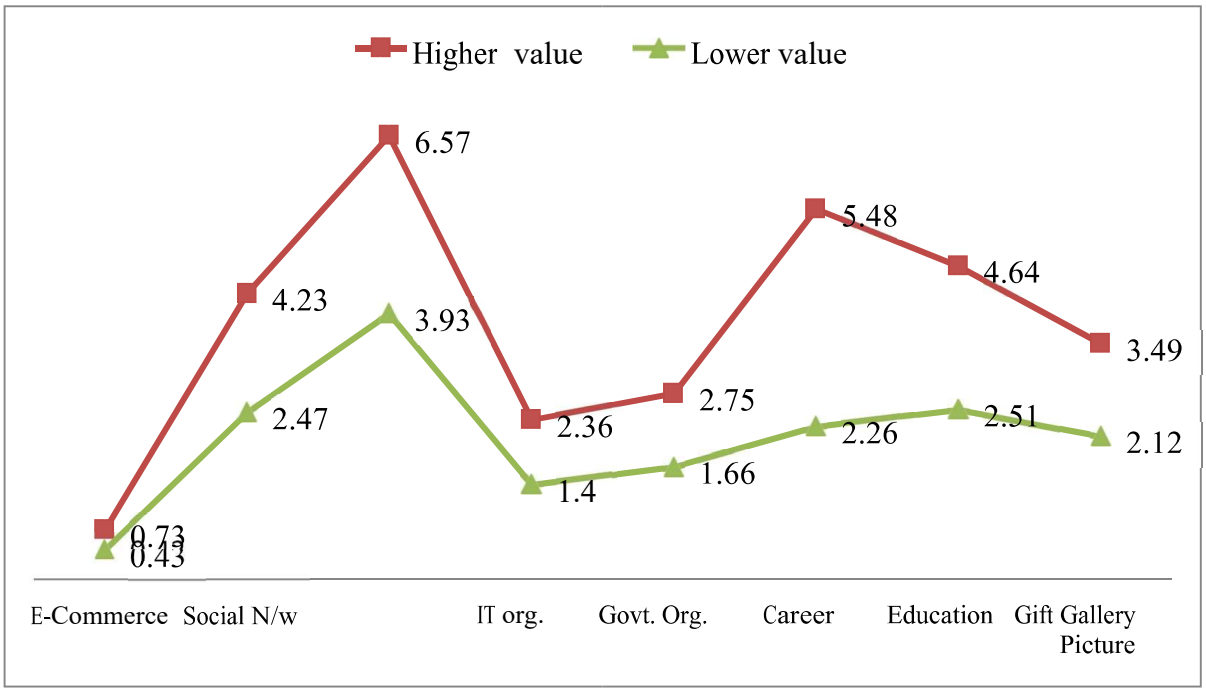

Figure 4.3: Risk range of web applications

The graph drawn from above computed value shows that IT organization application is badly attacked by SQLi than any other applications. The computed risk value of all applications shows that these values are in accordance to the standard statistics of SQLi attacks on various organizations that studied in chapter 1 i.e. IT related web applications are number one victim of SQL injection attacks.

This proposed methodology is an attempt to benefit the end user in many ways like- Along with analyzing the need for security remedies for a particular type of SQLi attack; it also provides the lower and higher end of the risk range associated with web application. The computed minimum and maximum risk value will help in evaluating the risk level and risk ranking in both numeric and subjective form i.e. computed risk value be analyzed in either in qualitative or in quantitative manner.

4.2 Risk level and Risk ranking

A Generalized formula was proposed in previous chapter is to navigate the end-user in taking decision out of accepting, transferring or mitigating the risk for the security of web application. The illustration of steps to determine the risk level and risk ranking are discussed as under

For an E-Commerce web application, the higher and lower values of risk are 0.73 & 0.43respectively. According to first step -determination of step size is

I step: Determine Step size = (Higher value – Lower value)/3 i.e. (0.73-0.43)/3= 0.1

II step: Lower value + Step size value, i.e. (0.43+0.1) =0.53

III step: Value between Risk level 1 & Risk level 3

IV step: Higher value – Step size value, i.e. (0.73-0.1) =0.63

The following table displays the all the preferences according to risk level & risk range.

Risk Level	Formulation	Range	Preference
RL1	0.43+0.1=0.53	0.43 to 0.53	Accept the risk
RL 2	Range between RL1 & RL 3	0.54 to 0.62	Transfer the risk
RL 3	0.73-0.1=0.63	0.63 to 0.73	Mitigate the risk

Table 4.7: Risk level and Range

The above table describes the list of preferences with respect to the risk levels. The range is evaluated from proposed formulation. This methodology can be used to provide the generalized risk level and risk range which can be used for any type of attack not limited to SQL injection attacks only. Now if the end user changes in the value of Matrix A, depending upon the probability of attack and then the newly computed range of risk will lie between 0.43 & 0.73 i.e. the minimum risk computed for this application is 0.43 and maximum risk computed is 0.73 by the proposed methodology for this application. So the final decision of uploading the web application on internet depends upon the range of computed risk value.

4.3 Risk computation of vulnerable applications

For evaluating the efficiency of proposed methodology and results analysis, the computed results must be compared with already existing results. For this, risk values of three PHP applications are exported from previous results. These applications are vulnerable and their respective risk value was already computed in research paper. The computed risk value is registered in the table given below:

Applications	Risk value
PHP address book (Address & contact manager)	0.68
Serendipity (Blog management)	0.75
PHP- fusion (Content management)	0.92

Table 4.8: Risk value of vulnerable application

Now in this section, risk value will be computed by proposed methodology on three vulnerable web applications that were already evaluated. First step is computation of probability of attack, all web applications are attacked with each type of SQLi attacks and each attack is performed 10 times with different query every time. The following table depicts the probability of attack for application under consideration:

Applications	First order	Second order	Lateral injection	Blind injection	Database focused
PHP address book (Address & contact manager)	0.6	0.4	0.1	0.7	0.4
Serendipity (Blog management)	0.4	0.2	0.2	0.7	0.3

PHP- fusion (Content management)	0.3	0.3	0.1	0.5	0.3

Table 4.9: Application wise probability of attack

4.3.1 The estimated risk values for PHP address book

Depending upon the value of probability, the values for Matrix A with both higher and lower values are placed accordingly as depicted as under:

0.01	0.66	0.66	0.01
0.66	0.37	0.37	0.37
0.66	0.01	0.01	0.66
0.37	0.66	0.66	0.01
0.66	0.37	0.37	0.37

Table 4.10: Lower value of matrix

0.36	0.99	0.99	0.36
0.99	0.65	0.65	0.65
0.99	0.36	0.36	0.99
0.65	0.99	0.99	0.36
0.99	0.65	0.65	0.65

Table 4.11: Higher value of matrix

The values of Matrix X are already derived, so its values can be referenced directly. Now impact of attack is computed by matrix multiplication of matrix A (with both lower and higher values) with matrix X and the values of interest are only diagonal values. So computed impact matrix for PHP address book application are shown below:

0.13	1.20	1.20	0.67	1.20
0.17	1.59	1.59	.88	1.59
0.13	1.20	1.20	0.67	1.20
0.17	1.53	1.53	0.85	1.53

0.17	1.59	1.59	0.88	1.59

Table 4.12: Lower impact values

0.27	2.43	2.43	1.35	2.43
0.29	2.64	0.264	1.47	2.64
0.27	2.43	2.43	1.35	2.43
0.29	2.69	2.69	1.49	2.69
0.29	2.64	2.64	1.47	2.64

Table 4.13: Higher impact values

The computation of risk value by modified proposed formula, the range of risk values associated with PHP address book is evaluated as under:

PHP address book					
Attack Type	Probability of attack	Lower value of impact	Higher value of impact	Risk range lower value (R=P*I)	Risk range Higher value (R=P*I)
FO	0.6	0.13	0.27	0.6*0.13= 0.07	0.6*0.27= 0.16
SO	0.4	1.53	2.64	0.61	1.05
LI	0.1	1.20	2.43	0.12	0.24
BI	0.7	0.85	1.49	0.59	1.04
DF	0.4	1.59	2.64	0.63	1.05
Total risk value		Risk Σ(Pi*Ii)		2.02	3.54

Table 4.14: Risk range of PHP address book

Similarly, the impact matrix is computed for other applications and computing the respective risk value. The risk range of each application after computing are shown as below.

4.3.2 The estimated risk values for Serendipity

Serendipity (Blog management)					
Attack Type	Probability of attack	Lower value of impact	Higher value of impact	Risk range lower value	Risk range Higher value
FO	0.4	0.11	0.23	0.04	0.09
SO	0.2	1.20	2.43	0.24	0.48
LI	0.2	1.20	2.43	0.25	0.48
BI	0.7	0.85	1.49	0.59	1.04
DF	0.3	1.59	2.64	0.47	0.47
Total risk value		Risk Σ(Pi*Ii)		1.59	2.56

Table 4.15: Risk range of Serendipity (Blog management)

4.3.3 The estimated risk values for PHP fusion

PHP Fusion(Content management)					
Attack Type	Probability of attack	Lower value of impact	Higher value of impact	Risk range lower value	Risk range Higher value
FO	0.3	0.11	0.23	0.03	0.06
SO	0.3	1.59	2.64	0.47	0.79
LI	0.1	1.20	2.43	0.12	0.24
BI	0.5	0.74	1.30	0.37	0.65
DF	0.3	1.59	2.38	0.47	0.71
Total risk value		Risk Σ(Pi*Ii)		1.46	2.45

Table 4.16: Risk range of PHP Fusion (Content management)

4.4 Result Analysis

The values computed from above calculations will provide the attack wise risk associated with web applications. These calculations also helps in deciding for which attack security measures or remedy is required to stop data extraction by attackers.

4.4.1 Risk value analysis

To compare and analyzes the risk estimated value results, the computed lower and higher risk range values from proposed methodology is to be converted from range 0 to 1. For standardizing the range, if the converted range value is increasing by 1, then the upper value is truncated and maximum value remains 1.

After converting the lower risk range and higher risk range to between 0 & 1. It is analyzed that the proposed methodology is capable of computing more risk value as compared to the previous one. The table below illustrates the risk range values and converted range values for all web applications:

Application	Lower risk value	Higher risk value	Values in range from 0 to 1
PHP address book	2.02	3.54	0.0-1.0
Serendipity (Blog management)	1.59	2.56	0-0.97
PHP Fusion(Content management)	1.46	2.45	0-0.99

Table 4.17: Rearranging the risk range

The following table displays the maximum value of the risk computed by previous and proposed methodologies.

Application	Max Risk value by previous methodology	Max Risk value by proposed methodology
PHP address book	0.68	1.0

Serendipity (Blog management)	0.75	0.97
PHP Fusion(Content management)	0.92	0.99

Table 4.18: Maximum risk value

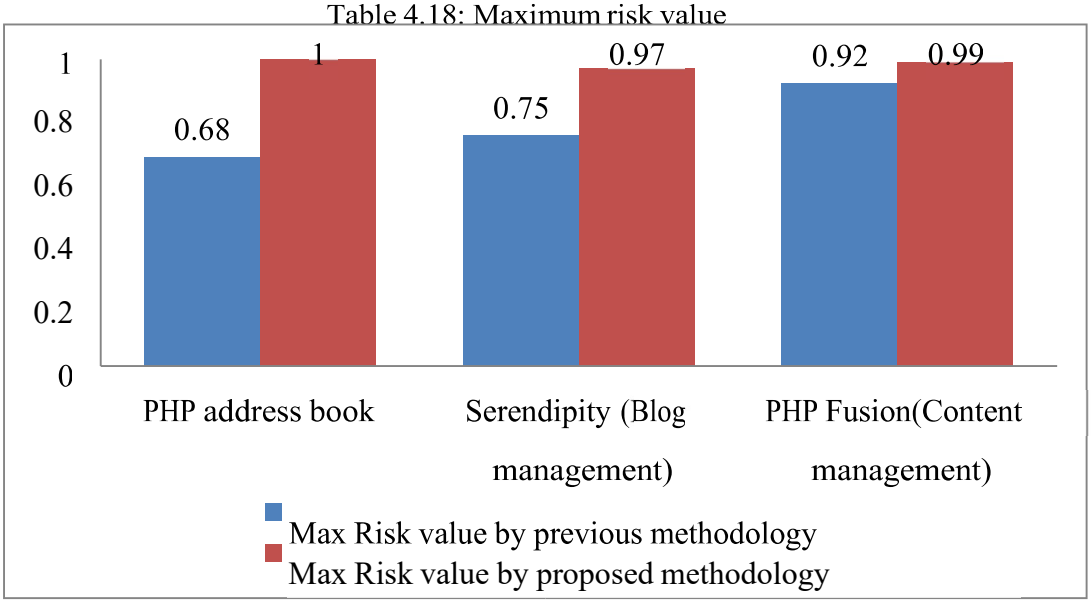

Fig 4.4: Comparison of risk values

The results from above graph shows that proposed methodology is capable of computing the maximum value of risk from a vulnerable application than the previous methodology used for comparision.On analyzing all the results, it is also concluded that the efficiency of evaluating risk by proposed methodology is also high for all unknown and vulnerable applications. This is the evidence that proposed methodology is efficient in computation and worthwhile for practical implementation in organizations. The following table and graph shows the efficiency of methodology:

Application	Proposed methodology is efficient by
PHP address book	47%
Serendipity (Blog management)	29%
PHP Fusion(Content management)	7%

Table 4.19: Efficiency table

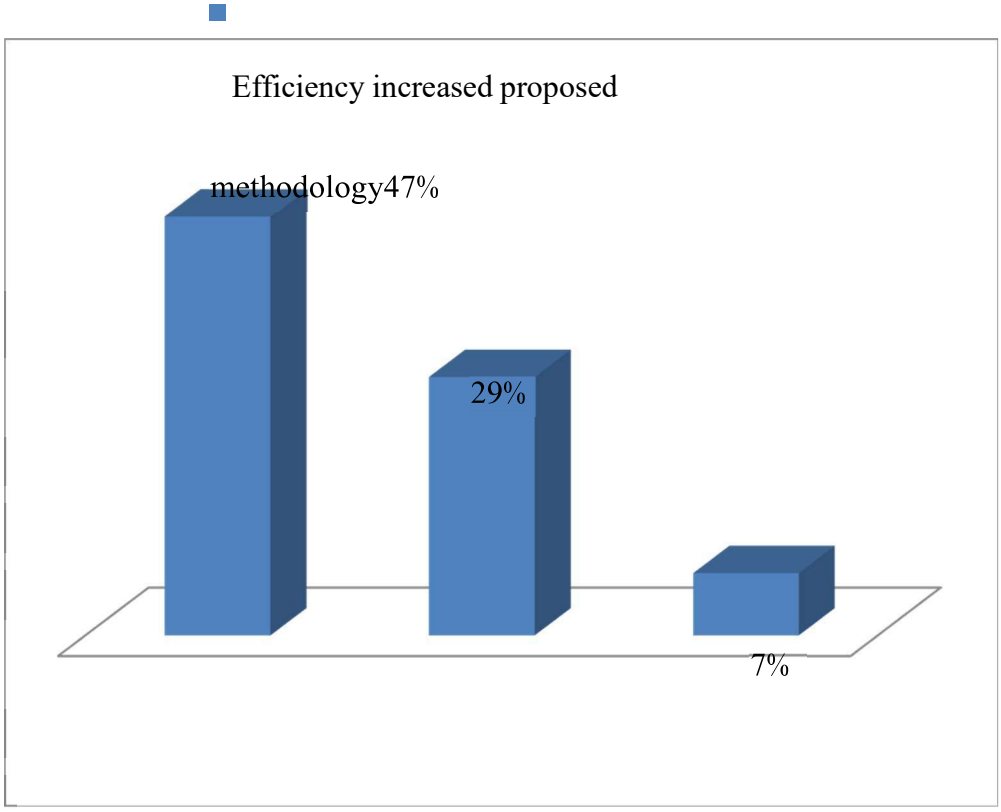

Fig 4.5: Efficiency graph

4.4.2 Risk range analysis

In the above analysis, it is demonstrated that the proposed risk analysis methodology is efficient in estimating risk values. Now another parameter is employed for testing the effectiveness by risk range analysis. In this section, the risk range of vulnerable applications is evaluated by the proposed formulation.

Previous researcher has considered three vulnerable applications from open source. First, the risk value of the vulnerable versions (as present on open source) of these applications is computed. After that some standard security measures and code correction techniques are applied to those vulnerable applications to make them less vulnerable or less prone to attacks i.e. the good version of applications. Then for this good version, the risk value is again computed. The following table displays both risk values for the same applications.

Application	Risk value	
	Good version	Vulnerable version

PHP address book (Address & contact manager)	0.25	0.68
Serendipity (Blog management)	0.36	0.75
PHP- fusion (Content management)	0.24	0.92

Table 4.20: Both version risk value

Now the risk value computed in two versions will work as lower and upper limit of risk estimation done by previous methodology i.e. risk value in vulnerable version means it can estimate the maximum value associated with the application and same is case for the good version. Applying the methodology proposed in thesis for risk level and risk range on these applications. The comparisons of the risk range estimated by both methodologies are placed in tables given below:

Risk level	Preferences	PHP address book	
		Previous	Proposed
RL1	Accept the risk	0.25-0.39	0.0-0.33
RL2	Transfer the risk	0.40-0.53	0.34-0.65
RL3	Mitigate the risk	0.54-0.68	0.66-1.00

Risk level	Preferences	Serendipity –Blog management	
		Previous	Proposed
RL1	Accept the risk	0.36-0.49	0.0-0.32
RL2	Transfer the risk	0.50-0.61	0.34-0.64
RL3	Mitigate the risk	0.62-0.75	0.65-0.97

Risk level	Preferences	PHP Fusion – Content management

		Previous	Proposed
RL1	Accept the risk	0.24-0.46	0.0-0.33
RL2	Transfer the risk	0.47-0.69	0.34-0.65
RL3	Mitigate the risk	0.70-0.92	0.66-0.99

Table: 4.21 Comparison of Risk range

After comparing the estimates range, it is analyzed that proposed methodology is providing a wide range for decision making preference of accepting, transferring or mitigating the risk for any application.

Chapter-V

Conclusion and Future scope

5.1 Conclusion of Research Work

In this thesis, an attempt has been made to rectify the problems pertaining to computing the risk value associated with web applications. These applications are manually attacked for the computation of the probability of attack. All attacked applications are unknown and chosen from most attacked categories of applications. The evaluation of the probability of an attack is done by considering the specific attack payload. New metrics are proposed for the computation of the impact of an attack. The evaluation for impact is performed mathematically. The newly proposed metrics for impact computations can be applied to any type of attack. The proposed solution is provided in accordance with the level of attacks. As SQL injection attack is a set of attacks. The proposed solution helps in analyzing for which particular type of attack security measures is required. Usually, security measures are financially heavy, so there will be no need for applying/purchasing a full security package i.e. whatever is required depends on associated risk value. Secondly, it not only estimates the risk value but also helps in making decisions for end-user:- to what inclination is to be followed after computing risk value for a web application. Hence it is the right kind of solution for providing security to web application database.

With regard to the goal of the thesis, considering the SQL injection attack as one attack and not considering all subtypes of these attacks can increase the financial budget for providing security to any organization's web application. Either purchasing of tools or designing mitigating solutions will depends on attack type and the severity posed by that particular attack. Usually, security measures are financially heavy. Every organization cannot afford security expenses like- appointing a specialist person for a specific task or hiring security professionals for maintaining security of application. This proposed methodology is a maiden attempt that can be used by risk analysts and security mangers to perform reliable and repeatable risk analysis in an authentic and affordable manner.

In conclusion; risk analysis is a general estimation of risk associated with an application. The decision of mitigating, accepting or transferring the risk depends on the web application owner/organization. This attempt can be a useful input for developing a new security mechanism. Small and medium size organizations can use this methodology in saving

their limited resources and significant money. Even researchers can extend this piece of work for experiment work on other types of attacks.

5.2 Future Scope of Research Work

Looking at all of the experiments, the overall goal of the thesis has been more than fulfilled. A risk analysis methodology performs well in terms of estimating risk and also improves the security of web applications. There are many possibilities of proximity in other research areas for future research.

This research work has focused on securing web applications from SQL injection attacks. There still many related problems that need further investigation. Securing web applications from other types of injection attacks like OS command injection, XPATH injection, SMTP injection, etc. are also desired. The attacks like cross-site scripting, sensitive data exposure, denial of services are also needed to be filtered. Some directions for future extension of this work are given below.

Experimental probability of attack computation for web applications developed in ASP and JSP technology.

i) Proposing general metrics for probability computation.

ii) To extend range values for proposed metrics from very low to very high.

iii) To include new parameters in Matrix X for computations.

iv) Defining new metrics for impact computation.

5.3 Summary of Research Work

Risk analysis is a methodology that estimates the risk associated with a web application. In this thesis research work, the prime focus is to provide the security of web application backend database. For this a risk estimation based on two functions of risk i.e. probability and impact is proposed. All the values used in this estimation are computed either experimentally or mathematically. The probability of attack is computed experimentally by executing the attacks. This step will compute the inherent risk associated with a web application. For impact computation, four metrices are proposed & a matrix method is also proposed to evaluate the impact. This step will figure out the additional risk other than computed in previous step. Finally the risk value is computed and a general risk range and risk rank formulation is also proposed. Usually risk measures are financially heavy- Every organization can’t afford security expenses like: appointing security specialist persons or

purchasing tools but this methodology is very much suitable for startup, small & medium sized organizations. As the method of evaluating risk value is very much close to the daily routine of developers and programmers. It can be used by risk analysts and security managers to perform reliable and repeatable risk estimation in authentic and affordable manner. For the small and start-up organizations, where resources in terms of technical staff and security persons are not available- a direct method is proposed for computing the risk value associated with the web application. Also for medium level organization the quick evaluation of risk value is also proposed. Initially the effort was to just quantify the risk value associated with a web application. Now this proposed methodology can be a helpful tool not only to estimate the risk value but also helps in decision making with computed risk value. It provides the freedom to analyze the risk in both qualitative mode i.e. subjective values or in quantitative mode i.e. numeric values. It is complete transparent mechanism to check security of web application.

References

[1] Karwan Jacksi and Shakir M. Abass, "Development History of the World Wide Web," International Journal of Scientific & Technology Research, Vol. 8, Issue 9, PP.75-79, 2019

[2] Santosh Kumar, "A review on client-server based applications and research opportunity," International Journal of Recent Scientific Research, Vol. 10, Issue 07(H), pp.33857-33862, 2019

[3] W. Jackson, "The past and future of HTML,"pp.1-4,Springer,doi:10.1007/978-1- 4302_4302-6536-8_1

[4] Mohamed Abomhara and Geir M. Køien, "Cyber Security and the internet of things: Vulnerabilities, Threats, Intruders and Attacks," Article no.4, pp. 65-88, 2015, doi: 10.13052/jcsm2245-1439.414

[5] Rossouwvon Solms and Johanvan Niekerk, "From information security to cyber security," Computers & Security, Vol. 38, pp.97-102, 2013

[6] Chandershekhar Sharma, S.C.Jain and A.K Sharma, "Explorative Study of SQL injection Attacks and Mechanisms to Secure Web Application Database," International Journal of advanced computer science and application, Vol.7, No.3, pp.79-87, 2016,doi:10.14569/IJACSA.2016.070312

[7] P. Tonella and F. Ricca, "Dynamic model extraction and statistical analysis of Web applications," Proceedings. Fourth International Workshop on Web Site Evolution, pp. 43- 52, 2002, doi: 10.1109/WSE.2002.1134088

[8] Xiaowei Li and Yuan Xue, "A Survey on Web Application Security," Journal of Nashville, Vol.25, Issue 5, pp.1-14, 2011

[9] Chandershekhar Sharma, S.C.Jain, "Analysis and classification of SQL injection vulnerabilities and attacks on web applications", Proc. of Int. Conf. on advances in engineering & technology research (ICAETR-2014), pp.1-6, 2014, doi: 10.1109/ICAETR.2014.7012815.

[10] D.Stttard and M.Pinto, "The Web Application Hacker's Handbook:Discovering and Exploiting Security Flaws,"(Wiley publication, 2007)

[11] Dimitris Mitropoulos and Diomidis Spinellis, "Fatal injection: a survey of modern code injection attack counter measures," PeerJ computer science, Vol.3, Edition 136, pp.1-40, 2017, doi:10.7717/peerj-cs.136

[12] Bikesh Shrestha, "Code Injection in Web Applications," Thesis, Helsinki Metropolia University of Applied Sciences, Finland, 2016

[13] D. Mitropoulos, P. Louridas, M. Polychronakis and A. D. Keromytis, "Defending Against Web Application Attacks: Approaches, Challenges and Implications," in IEEE Transactions on Dependable and Secure Computing, Vol. 16, No. 2, pp. 188-203, 2019, doi: 10.1109/TDSC.2017.2665620.

[14] Chad Dougherty, "Practical Identification of SQL Injection Vulnerabilities," Report produced for US-CERT,pp.1-15,2012

[15] Cenzic vulnerability report, Dark readings,2013

[16] M.A.Lawal, Abu Bakar, Md. SultanShuai Liu and Ayanoye.O.Shakiru, "Systemic literature review on SQL injection Attack," International journal of soft computing Vol.11, No.1, pp.5279–5293, 2016

[17] William G.J. Halfond, Jeremy Viegas, and Alessandro Orso, "A Classification of SQLinjection attacks and countermeasures," IEEE Int. Symp. on secure software engineering, 2006

[18] CyberEdge-2019-CDR-Report-v1.1.pdf, (available at www.imperva.com)

[19] Cyber security threatescape 2019 report, (available at www. Ptsecurity.com)

[20] Dan Kuykendall, "Why SQL Injection Vulnerabilities Still Exist," A report for database zone, 2016 (available at www. dzone.com/articles/whysqlinjectionvulnerabilitiesstill-exist)

[21] Akamai internet security report of first quarter, 2017(available at www.akamai.com/state-of-the-internet/q1-2017-state-of-the-internet-security)

[22] SQL-injection hall of shame (available at www.codecurmudgeon.com)

[23] Symantec security threat report of 2019, (available at www.symantec.com/securitycenter/threat-report)

[24] White paper on top ten database security threats, "How to Mitigate the Most Significant Database Vulnerabilities," (available at www.globalsecurity.com)

[25] Tsu-Yang Wu, Chien-Ming Chen, Xiuyang Sun, Shuai Liu and Jerry Chun-Wei Lin, "A Countermeasure to SQL Injection Attack for Cloud Environment," Journal of wireless personal communications Vol.96, pp- 5279-5293,2017,doi:10.1007/s11277-016-3741-7

[26] P.Bisht, P.Madhusudan and V.N. Venkatakrishan, "CANDID: Dynamic Candidate Evaluations for Automatic Prevention of SQL Injection Attacks," ACM Transaction on Information System Security, pp.1-39, 2010

[27] Y. W. Huang, F. Yu, C. Hang, C. H. Tsai, D. T. Lee and S. Y. Kuo, "Securing web application code by static analysis and runtime protection," Proc. of the 13th international conference on World Wide Web, pp.40–52, 2004

[28] V. B. Livshits and M. S. Lam, "Finding security vulnerabilities in java applications with static analysis," Proc. of the 14th conference on USENIX Security Symposium, Vol. 14, pp. 18-25,2005

[29] N. Jovanovic, C. Kruegel and E. Kirda, "Pixy: A static analysis tool for detecting web application vulnerabilities," Proc. of IEEE Symposium on Security and Privacy, pp. 258– 263, 2006

[30] S. Thomas and L. Williams, "Using automated fix generation to secure sql statements," Proc. of the Third International Workshop on Software Engineering for Secure Systems, pp. 9-16,2007

[31] S. Madan and S. Madan, "Shielding against sql injection attacks using admire model," Computational Intelligence, Communication Systems and Networks, International Conference, pp. 314–320, 2009

[32] L Zhang, Q. Gu, S. Peng, X. Chen, H. Zhao and D. Chen, "D-wav: A web application vulnerabilities detection tool using characteristics of web forms," Software Engineering Advances, International Conference, pp.501–507, 2010.

[33] Jalal Omer Atoum and Amer Jibril Qaralleh, "A hybrid technique for SQL injection attacks detection and prevention," International Journal of Database Management Systems (IJDMS) Vol.6, No.1, pp.21-28, 2014

[34] Garima Singh, Dev Kant, Unique Gangwar and Akhilesh Pratap Singh, "SQL Injection Detection and Correction Using Machine Learning Techniques," Proceedings Society of India, Vol. 1, pp.435-442, 2015

[35] A. Razzaq, A. Hur, N. Haider and F. Ahmad, "Multi-layered defence against web application attacks," Proc. of Sixth International Conference on Information Technology: New Generations, pp. 492–497,2009

[36]Srinivas Avireddy, "Random4: An Application Specific Randomized Encryption Algorithm to prevent SQL injection," XI Int. Conf. on Trust, Security and Privacy in Computing and Communications (TrustCom), IEEE, pp. 1327-1333, 2012

[37]Sruthy Mamadhan, Manesh T, and Varghese Paul, "SQLStor: Blockage of Stored Procedure SQL Injection Attack Using Dynamic Query Structure Validation," 12th Int. Conf. on Intelligent Systems Design and Applications (ISDA), IEEE, pp. 240-245, 2012

[38] OWASP Top 10 attacks, www.Owasp.org

[39] Xiaowei Li and Yuan Xue, "A Survey on Web Application Security," journal of Nashville, Vol.25, Issue 5, pp.1-14, 2011

[40] S. Kals, E. Kirda, C. Kruegel and N. Jovanovic, "Secubat: web vulnerability Scanner," Proceedings of the 15th international conference on World Wide Web, pp. 247–256, 2006

[41] G. Hermosillo, R. Gomez, L. Seinturier and L. Duchien, "Using aspect programming to secure web applications," 2007

[42] Y. Kosuga, K Kono and M. Hanaoka, "Sania: Syntactic and semantic analysis for automated testing against sql injection," Proc. of Twenty-Third Annual Computer Security Applications Conference, 2007

[43] H. Shahriar and M. Zulkernine, "Music: Mutation-based sql injection vulnerability Checking," Proc. of Eighth International Conference on Quality Software, pp. 77–86, 2008

[44] A. Ciampa, C. A. Visaggio, and M. D. Penta, "A heuristic-based approach for detecting sql-injection vulnerabilities in web applications," Proc. of Workshop on Software Engineering for Secure Systems,pp. 43–49,2010

[45] Z. Zhang, Q. Zheng, X. Guan, Q. Wang and T. Wang, "A method for detecting code security vulnerability based on variables tracking with validated-tree, "Frontiers of Electrical and Electronic Engineering Vol. 3, pp.162–166, 2008

[46] M. Junjin, "An approach for sql injection vulnerability detection," Proc. of Sixth International Conference on Information Technology: New Generations, pp. 1411–1414, 2009

[47] A. Anchlia and S. Jain, "A novel injection aware approach for the testing of database applications," Proc. of International Conference on Recent Trends in Information, Telecommunication and Computing, pp. 311–313, 2010

[48] Gary W. and Z. Su, "An analysis framework for security in web applications," Proc. of the FSE Workshop on Specification and Verification of Component-Based Systems, pp.70– 78, 2004

[49] S. Madan and S. Madan, “Shielding against sql injection attacks using admire Model,” Proc. of international conference computational intelligence, pp. 314–320, 2009

[50] Y. W. Huang, S. K. Huang, T. P. Lin, and C. H. Tsai, “Web application security assessment by fault injection and behaviour monitoring,” Proc. of the international conference on World Wide Web, pp. 148–159,2003

[51] S. Thomas, L. Williams, and T. Xie, “On automated prepared statement generation to remove sql injection vulnerabilities,” Inf. Software Technology, Vol. 51 pp. 589–598, 2009.

[52]Mohammad Abu Kausar & Mohammad Nasar, “An Effective Technique for Detection and Prevention of SQLIA by utilizing CHECKSUM Based String matching,” International Journal of Scientific & Engineering Research, Vol. 9, Issue 1,2018 [53] Types of injection attacks,www.ibm.com/wap_injection_attacks

[54] Web application security, Top _10 attacks, www. Wasc.com

[55] Haifeng Gu,Jianning Zhang, Tian Liu, Ming Hu, Junlong Zhou,Tongquan Wei and Mingsong Chen, “DIAVA: A Traffic-Based Framework for Detection of SQL Injection Attacks and Vulnerability Analysis of Leaked Data,” in IEEE Transactions on Reliability, vol. 69, no. 1, pp. 188-202, March 2020, doi: 10.1109/TR.2019.2925415.

[56] Peng Tang, Zheng Huang, Huijuan Lian and Guozhen Liu, “Detection of SQL injection based on artificial neural network,” Knowledge Based Systems, Vol .190, 29, ELSEVIERFebruary, 2020

[57] K. Varshney and RL Ujjwal, “LsSQLIDP: Literature survey on SQL injection detection and prevention techniques”, Journal of statistics and management systems, Vol 22, 2, Taylor & Francis, March 2019

[58] Kirti Sharma and Shobha Bhatt, “SQL injection attacks - a systematic review,” International Journal of Information and Computer Security, Inder science Vol. 11, Issue 4/5, pp. 493-509, Jan 2019

[59] Reynaldo E. Castillo, Jasmin A. Caliwag, Roxanne A. Pagaduan and Aira Camille Nagua, “Prevention of SQL Injection Attacks to Login Page of a Website Application using Prepared Statement Technique,” Proc. of 2nd Int. Conf. on Information Science and Systems, pp.171–175, ACM digital library, March 2019 ,https://doi.org/10.1145/3322645.3322704

[60] Cagri Cetin, Dmitry Goldgof and Jay Ligatti, “SQL-Identifier Injection Attacks”, IEEE Conference on Communications and Network Security (CNS), pp.151-159, August 2019, doi: 10.1109/CNS.2019.8802743

[61]Peng Tang, Weidong Qiu, Zheng Huang and Guozhen Lie, “Detection of SQL injection based on artificial neural network,” Knowledge Based Systems, Vol.190, 29, ELSEVIER, February, 2020,doi.org/10.1016/j.knosys.2020.105528

[62] Muhammad Saidu Aliero, Imran Ghani, Kashif Naseer Qureshi and Mohd Fo’ad Rohani, “An algorithm for detecting SQL injection vulnerability using black-box testing,” Journal of Ambient Intelligence and Humanized computing” Springer,Vol.11,pp 249-266, February, 2019

[63] Oliveira Batista Lucas, Adriano de Silva Gabriel, Souza Araújo Vanessa, Silva Araújo Vinícius Jonathan,Silva Rezende, Thiago,Junio Guimarães Augusto and Vitor de Campos Souza Paulo, “Fuzzy neural networks to create an expert system for detecting attacks by SQL Injection,” The International Journal of forensic computer science, E-forensic press, Vol-13 no. 1, pp. 8-21, January 2018

[64] Shaji N. Raj and Elizabeth Sherly, “An SQL Injection Defensive Mechanism Using Reverse Insertion Technique,” Int. Conf. on Next Generation Computing Technologies, pp. 335-346, June 2018

[65]Ahmad Ghafarian, “A hybrid method for detection and prevention of SQL injection attacks,” IEEE Computing conference, July 2017, doi: 10.1109/SAI.2017.8252192

[66] Q. Temeiza, M. Temeiza and J. Itmazi, "A novel method for preventing SQL injection using SHA-1 algorithm and syntax-awareness," 2017 Joint International Conference on Information and Communication Technologies for Education and Training and International Conference on Computing in Arabic (ICCA-TICET), pp.1-4, Khartoum, 2017 doi: 10.1109/ICCA-TICET.2017.8095285.

[67]Daniel Thomas Loughran, Mayar Kefah Salih and Vinitha Hannah Subburaj, “All about SQL Injection Attacks”, Journal of the Colloquium for Information System Security Education (CISSE) Edition 6, Issue 1 pp.1-24,September 2018

[68]MA Kausar, M Nasar and A Moyaid, “SQL Injection Detection and Prevention Techniques in ASP.NET Web Application,” International Journal of Recent Technology and Engineering, Volume-8 Issue-3, pp. 7759-7766, September 2019

[69] Nada Basit, Abdeltawab Hendawi, Joseph Chen and Alexander Sun, "A Learning Platform for SQL Injection," Proc. of the 50th ACM Technical Symp on Computer Science Education, pp. 184-190, Feb. 2019,doi:10.1145/3287324.3287490

[70] K. Rohini, K. Kasturi and R. Vignesh, "Method for Simulating SQL Injection and DOS Attack," Intelligent Computing and Innovation on Data Science, Lecture Notes in Networks and Systems, Vol. 118, pp.793-801, Springer, May 2020

[71] Md. Emon Hossain and Sabbir Ahmed, "An approach to secure multi-tier websites through SQL-Injection detection and prevention," Proc. of the Int. Conf. on Computing Advancements, Article No: 14, pp. 1–7, January 2020,doi:10.1145/3377049.3377096

[72] Muhammad Saidu Aliero, Kashif Naseer Qureshi, Muhammad Fermi Pasha, Imran Ghani and Rufai Aliyu Yauri, "Systematic Review Analysis on SQLIA Detection and Prevention Approaches," Journal of Wireless Personal Communications,112, pp.2297–2333,Springer, January 2020, doi: 10.1007/s11277-020-07151-2

[73] Rajesh Vemulakonda, Ketha Venkatesh, "SQLIADP: A Novel Framework to Detect and Prevent SQL Injection Attacks," Smart Intelligent Computing and Applications, Springer, Vol.160, pp. 41-50, October 2019

[74] Nabeel Salih Ali ,Abdul Samad Bin Shibghatullah ,Ahmed Hazim Alhilali ,Salam AlKhammasi ,Mohammed Falih Kadhim and Hayder K. Fatlawi, " A comparative analysis and performance evaluation of web application protection techniques against injection attacks," International Journal of Mobile Communications ,Inderscience, Vol.18,Issue 2,pp.196- 228,March 2020

[75] Makera M. Aziz and Dina Rafaa Ahmed, "Using the Behavioural Biometrics to Prevent SQL Injection Attack," Multi-Knowledge Electronic Comprehensive Journal for Education and Science Publications, Issue 28, pp.1-10, January 2020

[76] Doaa E.Nofal and Abeer A. Amer, "SQL Injection Attacks Detection and Prevention Based on Neuro-Fuzzy Technique," International Conference on Advanced Intelligent Systems and Informatics, Springer, Vol. 1058, pp.722-738, October 2019

[77] Zar Chi Su Hlaing and Myo Khaing, "A Detection and Prevention Technique on SQL Injection Attacks," IEEE Conf. on Computer Applications (ICCA), pp.1-6, March 2020, doi: 10.1109/ICCA49400.2020.9022833

[78] A. (Tony) Cox Jr, Some Limitations of Risk = Threat × Vulnerability × sequence for Risk Analysis of Terrorist Attacks, Risk Analysis: an official publication of the society for Risk Analysis, 28(6), pp. 1749-1761, 2008, doi: 10.1111/j.1539-6924.2008.01142.x

[79] Chandershekhar Sharma,S.C. Jain, A.K. Sharma, "A quantitative risk analysis methodology for the security of web application database against SQL injection (SQLi) attacks utilizing fuzzy logic system as computational technique," The international journal of electrical engineering & education, pp.1-20,June 2019, doi: 10.1177/0020720919847542

[80] National cyber security centre UK, Critical risk methods report-2018

[81] Dawid czagan,"Quantitative risk analysis report, 2018

[82] Rot A, "IT risks assessment: quantitative and qualitative approach," Proc. of the World Congress on Engineering and Computer Science, 2008

[83] "A practical approach for managing risk," Software Engineering Institute, Carnegie Mellon University, Available at www.sei.cmu.edu/risk

[84]Yoon Jung Chung, In Jung Kim, NamHoon Lee, Taek Lee and Hoh Peter, "Security Risk Vector for Quantitative Asset Assessment," Int. Conf. on Computational Science and its Applications, Vol. 3481, pp. 274-283, book series,2005

[85] J. Spears, "A Holistic Risk Analysis Method for Identifying Information Security,"

International Federation for Information Processing, Springer, Vol.193, pp. 185- 202, 2006

[86] I.Mkpong-Ruffin, D. Umphress, J. Hamilton, and J. Gilbert, "Quantitative software security risk assessment model," Proc. of ACM workshop on Quality of protection,pp.31–33, ,2007, USA

[87] L. Grunske and D. Joyce, "Quantitative risk-based security prediction for component based systems with explicitly modelled attack profiles," Journal of Systems and Software, Vol. 81(8), pp.1327-1345, 2008

[88] W. Zhihu and W. Xin, "Research on technologies in quantitative risk assessment and forecast of network security," 3rd Int. Conf. on Advanced Computer Theory and

Engineering pp. V6-524-V6-528, 2010,doi:10.1109/ICACTE.2010.5579190

[89]Chenmeng Sui, Yanzhao Liu and Yun Liu," A Software Security Assessment System Based On Analysis of Vulnerabilities", Journal of Convergence Information Technology Vol.7, No.6, pp. 211-219, 2012

[90] Hossain shahriar and Hisham haddad, “Risk assessment of code injection vulnerabilities using Fuzzy Logic based system,” Proc. of the 29th Ann. ACM Symp on Applied Computing, pp.1164–117,2014,doi:10.1145/2554850.2555071

[91] D. Reyes Fernandez de Bulnes, “Redmine plugin: Risk Quantitative Analysis,” IEEE Latin American Transactions, Vol.13, No.7, pp. 2423-2429,doi:10.1109/TLA.2015.7273808

[92] U. K. Singh and C. Joshi, “Quantitative Security Risk Evaluation using CVSS Metrics by Estimation of Frequency and Maturity of Exploit,” Proc. of the World Congress on Engineering and Computer Science,Vol.I,2016

[93] U. K. Singh and C. Joshi, “Information security assessment by quantifying risk level of network vulnerabilities,” International Journal of Computer Application, Vol.156, pp.37-44, 2016

[94] S. Halkidis, N. Tsantalis, A. Chatzigeorgiou, and G. Stephanides, “Architectural Risk Analysis of Software Based on Security Patterns,” IEEE Transactions On Dependable and
Secure Computing, Vol.5, Issue 3,pp.129-142, July 2008

[95] S. H. Houmb, V. N. L. Franqueira and E. A. Engum, “Quantifying security risk level from CVSS estimates of frequency and impact,” Journal of Systems and Software,Vol.83,pp. 1622-1634,2010

[96]Feng N. and Yu, X., A “Data-driven assessment model for information system security risk management,” Journals of computer, Vol.7, No.12, pp.3103-3109, 2012

[97]Hui Guan,Wei-Ru Chen,Lin Liu and Hong-Ji Yang, “Estimating Security Risk for Web Applications using Security Vectors,” Journal of Computers,Vol.23, No.1,2012

[98]Jignesh Doshi and Bhushan Trivedi, “A Framework for Analyzing Risk of Web Application Vulnerabilities,” International Journal of Emerging Trends & Technology in Computer Science, Vol.3, Issue 4, pp.233-237, 2014

[99]Abdul Razzaq, Zahid Anwar, H.Farooq Ahmad, Khalid Latif and Faisal Munir, “Ontology for attack detection: An intelligent approach to web application security,” Computers & Security, Vol. 45, pp.124-146, 2014

[100]Mohammed Alhomidi and Martin Reed, “Attack Graph-based Risk assessment and Optimization Approach,” International Journal of Network Security & Its Applications, Vol.6, No.3, pp.31-43, 2014

[101]Ennan Zhai, Liang GU and Yumei Hai, "A Risk-Evaluation Assisted System for Service Selection," Proc. of 2015 IEEE Int. Conf. on Web Services (ICWS), pp.671 – 678,2015,doi:10.1109/ICWS.2015.94

[102]Yazar Z, "A Qualitative risk analysis and management tool – CRAMM," SANS Institute Infosec Reading Room, 2011

[103] National Institute of Standards and Technology (NIST), Report on risk management guide for information technology systems, 2001, Special publication 800-30

[104] H C Juh and Y K Malaya, "A framework for software security risk evaluation using vulnerability life cycle and CVSS metrices," Proc. of 5th Int. Conf. on risks and security of internet and system, pp.430-434, 2010

[105] Hui Guan, Wei-Ru Chen, Lin Liu and Hong-Ji Yang, "Estimating Security Risk for Web Applications using Security Vectors," Journal of computer, Vol.23, No.1, pp. 54-69, 2012

[106] Mouna Jouini, Latifa Ben Arfa Rabai and Ridha Khedri, "A Multidimensional Approach towards a Quantitative Assessment of Security Threats," Procedia Computer Science, 52, pp. 507-514, 2015

[107] Chandershekhar Sharma, S.C.Jain and Arvind K.sharma, "Risk based quantitative analysis of SQLIA on web application database," Proceedings of the III International
Conference on Computing for Sustainable Global development (INDIACom-2016), pp 748 – 752, 2016

[108] Yuan-dong Cheng, Ji-dong He and Fa-gang Hu, "Quantitative risk analysis method of information security-combining fuzzy comprehensive analysis with information entropy," Journal of discrete mathematical science & cryptography, Taylor & Francis, Vol.20, Issue 1, pp.149-165, 2016

[109] Daljit Kaur, Parminder Kaur and Hardeep Singh, "Insecurity Status and Vulnerability Density of Web Applications: A Quantitative Approach," International Journal of Computer Science and Information Security, Vol.15, No.1, 2017

[110]Mansour Alali, Ahmad Almogren, Mohammad Mehedi Hassan, Iehab A.L. Rassan and Md Zakirul Alam Bhuiyan, "Improving risk assessment model of cyber security using fuzzy logic inference system," Computers & Security,74,pp.323–339,2018

[111] Stephen Hart, Anna Lisa Ferrara and Federica Paci, "Fuzzy-based approach to assess and prioritize privacy risks, "Springer Methodologies and application,24,pp.1553-

1563,2020,doi:10.1007/S00500-019-03986

[112] www.wikipedia.org/wiki/Information_security

[113] https://www.incapsula.com/web-application-security/application-security.html

[114] https://whatis.techtarget.com/definition/Confidentiality-integrity-andavailability-CIA

[115]https://www.csoonline.com/article/2130877/data-breach/the-biggest-databreaches-ofthe21stcentury.html