Defense Mechanism Using Multilayered Approach and SQL Injection Methods for Web Based Attacks

Khalid Akram[1],Gulzar Banu[2], Mustafa Basthikodi[3], Ahmed Rimaz Faizabadi[4],
[1,2, 3,4]BEARYS INSTITUTE OF TECHNOLOGY, Mangalore

Abstract: ***Web application security is a major concern in dependent E-Business environment. It's crucial to keep cyber thieves and hacker away from sensitive information. To understand how things work behind the scene we shall study secure software development and its impact. Provided with some of the attacks and its prevention I have merged two approaches to deal with SQL injection and cross - site scripting .This defense mechanism shall filter potential injections and possess high ability to detect vulnerabilities and handle them .With the help of control flow graph it will reduce the time of sequential search.***

KEYWORDS: Sql injection, cross-site scripting ,Malicious injection ,input validation.

INTRODUCTION:

Software security is idea of engineering software which makes system work under malicious attack .This dept aims to explore software security best practices .Software problem is critical and central aspect of computer security problem . Using software defects hackers can exploit software defects. With growth in extensibility and complexity the problem of software development becomes more fierce. Good software development leverages when developer knows and understands common threats including language based flaws and pitfalls[1]. Its better to handle something which's defect free rather than something riddled with vulnerabilities. Two aspects of secure software development are application security and software security. Application security deals with fixing and finding known security problems after they are exploited where as software security deals designing ,building and testing software for security problem is that vulnerabilities in the software let malicious hackers skirt standard security technologies with impunity. If this were not the case, then the security vulnerability problem would not be expanding the way that it is .Clearly, this emphasizes the need to get builders to do a better job on the software in the first place .In short run we should desperately make progress on both the fronts but in long run we should figure out easier ways to defend-code. Software security deals with leveraging builders to do better job so that operators end up with easier job. To develop such software often developers focus on functionalities. Adding security features like SSL(for cryptographically protecting the communication) doesn't give complete solution to the problem. Software security is a system-wide issue that takes into account both security mechanisms (such as access control) and design for security (such as robust design that makes software attacks difficult.) The commercial vendors and technologist acknowledge that software security problem exist but no discreet initiative towards the solution has been implemented. The solving of problem is implemented in late life cycle activities like penetration testing, patch management, firewalling.

Application security-Penetration testing

Assessing network vulnerabilities and IT systems in current interconnected digital world is a daunting endeavor[2]. By embracing penetration testing's best practices and procedures, we can affordably and proactively address security loopholes before hackers penetrate into the system. In today's connected digital ecosystem, applications are center stage, influencing all aspects of interaction and communication. Such digitalize applications contain sensitive data delivering business - critical information services resulting in even smallest security is exploited by cybercriminals wanting to wreak havoc. Hacking threat forced Yahoo and Google to partner up creating an encrypted email system that cannot be decrypted by company themselves. Our life's are completely reliant on digital devices which are prone to security hacks. As a result there's grave security concern reinforced by recent event

• In January 2016, a large Belgian bank was attacked by cybercriminals that cost the bank 70 million Euros, although no customers were affected by the breach. This type of attack is called a whaling attack or spear-phishing.[3]

In August 2015, the U.S. Internal Revenue Service reported that about 300,000 taxpayers' personal information was compromised when hackers cracked the agency's multi-step authentication process and were able to make fraudulent claims for tax refunds using stolen identities.[4].

In November 2015, a Switzerland-based encrypted e-mail provider's Internet connection was held for ransom by hackers in what could be described as a distributed denial of service (DDOS) attack.[5] .

In February 2015, a large U.S. health insurer's database was breached, and sensitive information that affected about 80 million customer records was stolen. This was described as a sophisticated advanced persistent threat (APT), where a malicious user gains access to internal networks primarily to steal data.[6].

In October 2015, a UK phone and broadband provider's website was hacked by cybercriminals who may have pilfered confidential banking details and personal information. This type of attack could be described as a sequential attack or SQL injection[7].

Different techniques:

1.Dictionary attack-*Sensitive information is accessed using passwords for security reasons, these passwords are created using dictionary words which are stored in database after they are encrypted instead of plaintext using one way hash function. Other methods like brute-force, rainbow table or dictionary attack can be used to attack but most commonly used method to crack password hashes is dictionary attack .Certain existing policies consider few passwords as strong but they are easily vulnerable for dictionary attack[8].*

Implementation: Most of possible combination of passwords of various dictionary words are present in dictionary file. The password entered by user is stored along with user name in text file with particular format. Set of code is written based on cryptography which on processing gives the required password with the help of dictionary[9].

Different approaches of dictionary attack-

1.Offline dictionary attack*: This attack is considered to be the most damaging threats which reveals smart card information. To access victims smart card information, secret parameters should be known from the smart card. So once attacker gets access to victims card, he extracts those secret parameter and then sends it back to the victim without his knowledge. As victim uses this card, hacker performs malicious attack leading to leaking of password[10].*

2.Pattern based dictionary attack*: Its improvised version of dictionary attack. It differs from ordinary dictionary attack as dictionary file is used for the attack. This method uses huge files containing different patterns of password in contradict to ordinary dictionary attack which has large file containing millions of simple possible dictionary words..Software named pattern based password generator is used to crack password which generates a pattern based dictionary[9].*

Web based attacks*: Our daily life is significantly effected by web based applications. They contain sensitive data which can be exploited in order to personate the user or to steal information[11].These attacks can also decrease the system performance.*
Different approaches of web based attack-Input validation attacks like inspect element trick, cross- site scripting, buffer overflow, SQL injection are most extensive forms of vulnerabilities on web applications.

1.Cross site scripting(XSS)attack*: This type of attack occurs when harmful data is sent from web application by the user. The gathered data is in the form of hyperlink which contain harmful data in it. The injection into HTML output generated by web application enables hackers to inject script into web page which's viewed by others[11]. Cross site scripting is further classified into two categories namely Persistent and Non-Persistent.*

Persistent: *In persistent method XSS script resides permanently in database and this script is executed every time user visits web app that is affect. Its more devastating variant of cross-site scripting flaw. An example of this is online message boards where users are allowed to post HTML formatted messages for other users to read.*

Non-persistent: *This method is more oftenly used nowadays in which script is not present in server, and it affects the user running the code directly[12]. It causes victims browsers to navigate to URLs on vulnerable site*

automatically leading to scenarios like picking up their contact information which performs in the background

Prevention technique for XSS:

1.Using tool called XSS Me for detection of XSS.
2. Making text boxes non executable wherever possible
3. Filtering all special characters like semicolons, slash, backslash, less than and greater than symbol ,quotation marks.
4. Validation for length of input.
5. Restriction for user and file access in all kind of application environment.
6.Using proactive approach to input validation attacks.

2.Buffer overflow attack: *This attack occurs when program outruns the buffer size and overwrites into adjacent memory resulting in memory related errors, violation of system security ,crash. It is caused due to inputs given to execute the code and software liablity is exploited .Systems with fixed amount of memory are more vulnerable to this type of attack[12.].It comes under input invalidation and poor memory management as well as sloppy programming category. Such attacks take place on C and C++ programs. There are four types of buffer overflow attacks-Heap Overflow, Stack Overflow, Arithmetic Overflow , Format Overflow[13].*

Prevention technique for Buffer overflow:

1.Bound checking on input[14].
2.User permission and file access have to be taken into account while programming [15].
3.Using sandboxing to prevent attackers from injecting malicious code in vulnerable application.

3.SQL Injection: *It is technique which has malicious code that exploits security in database. Generally SQL injection works by inserting harmful code into program by modifying the SQL query structure. This results in attacker aquisition of sensitive information such as business related data or account details. Another way to perform SQL injection is insert malicious strings into data value arguments/form in URL. The main reason for this attack is improper validation. SQLIA-SQL injection attack gives direct access to database and culminates in targeting database by extracting sensitive information.*

Classification of SQLIA:

In this section we discuss different kinds of SQLIA. Each type of attack shall be provided with descriptive name, attack intent, description of attack, example, and set of references that discuss the attack technique in detail. These attacks are not performed separately. It can be performed in sequence of attackers will with any number of variation which can be countless.

Tautologies: *Injecting of malicious code in one or more conditional statements which after evaluating turns to be true. Common usage is to extract data by bypassing authentication. The attacker uses WHERE condition for injecting code. This way tautology causes retrieval of all rows from database. Attacker must consider not only vulnerable parameters but also coding constructs that evaluate the query results. Attack is successful when result displays one of the record or all the records are returned*

Example: The attackers inputs "1=1-- or ' " in the login field. The resulting query is:

SELECT account FROM admins WHERE
login=' ' or 1=1 AND pass=' ' AND pin=

Here the conditional OR 1=1 transforms the whole WHERE clause into a tautology. The database checks conditional as a base for evaluating each row and decides which row to return to the application .As the conditional is a tautology, the query evaluates to true for each row and returns all of them .In the above example the returned set evaluates to non null value which causes successful authentication of admin. The application invokes method called displayAccounts() and shows all the accounts in the database[16]

Logically/Illegal Incorrect Queries:

Using this attack the attacker can get to know the structure and type of database of application. Its considered as information gathering step for other attacks. The vulnerability leveraged by this attack is that the default error page returned by application servers is often overly descriptive. When the error messages is generated, injectable or vulnerable parameters revealed to the attacker. Instead of helping the programmer To debug their application it helps the attacker to gain the information about the schema of back end database. While performing this attack an attacker tries to inject statements that cause a type conversion ,logical error ,syntax into database. Logical error gives the name of the column and the table. Syntax errors are used to identify injectable parameters and type errors are we used to get extract data or data type.

Example: Using type conversion error to reveal relevant data by injecting the following text into input field pin: "convert(int,(select top 1 name from sysobjects where xtype='u'.

SELECT accounts FROM users WHERE login='' AND pass='' AND pin= convert (int,(select top 1 name from sysobjects where ytype='z'))

The attacker attempts to Extract the first user table (ytype ='z') from the database. The query tires to convert the table name into an integer. As it is not Illegal type conversion the database throws the following error "Microsoft OLE DB Provider for SQL Server (0x80040E07) Error converting nvarchar value 'DebitCards' to a column of data type int." Using this error message the attacker gets to know the database is an SQL server database. Secondly error message reveals the value of the string that cause the type conversion to occur. In this scenario, this value is also the name of the first user defined table in the database "DebitCards". From the gathered information attacker creates further attacks that targets the specific area.

Union query:

By exploiting over vulnerable parameter to manipulate the data set returned for a given query. Using this technique attacker can't retrieve data from different table than the intended one by the developer. Attacker injects the statement of the form. Union SELECT <.rest injected query> . As second query is under the control of attacker the query is used to retrieve information required from the table. So the database returns a dataset that's Union of injected query and the original query

Example:Attacker injects the text in the login field with the following "' UNION SELECT card number from Debitcards WHERE account_number= 70722 --".

query generated is:

SELECT accounts FROM users WHERE login='' UNION SELECT card_number from DreditCards where acctNo=70722 -- AND pass='' AND pin=.

with the assumption that there is no login to """", first query returns null set and second query return data from "Debitcards" table. The database would return card_number for account "70722". the following result of two queries are Union and are returned to the application.

Piggyback queries

In Piggy back queries additional queries are injected into the original query. It is different type of attack compared to other attacks as original query is not modified instead new and distinct query called "piggyback" are included in the original query resulting in multiple SQL queries sent to the database. The first query execute normally followed by the injected query which are executed in addition to the first .This attack is considered to be the most extremely harmful attack because if attackers can insert any sort of stored procedure, SQL command into the additional queries along with the original query the vulnerability becomes high.

Example: If the attacker inputs "'; drop table users - -" into the pass field, the application generates the query:

SELECT accounts FROM users WHERE login='khalid' AND pass='';

drop table users -- ' AND pin=689

After execution of the first query the database come across query delimiter ";" Execute the second query which would drop the table users, destroying valuable information.

Stored procedures: *Stored procedures in the database are violated. Standard database contains set off of stored procedure which extend the functionality of the database increasing the interaction with the operating system. Once the hacker has the information of backend database in use the SQL injection attack can be crafted to execute stored procedures provided by that specific database .Stored procedures are as vulnerable as normal attack on the application. Stored procedures are written in special scripting language they contain vulnerability such as buffer overflow.*

Example: stored procedure for checking credentials

```
CREATE PROCEDURE DBO.isAuthenticated
@userName varchar2, @pass varchar2, @pin int
AS
EXEC("SELECT accounts FROM users WHERE login='" +@userName+ "' and pass='" +@password+ "' and pin=" +@pin);
GO
```

The launching of SQL injection attack is done by entering " ' ; SHUTDOWN; -" into either the userName or password fields.This causes the stored procedure to generate the following queen

SELECT accounts FROM users WHERE login='doe' AND pass=' '; SHUTDOWN; -- AND pin=

The attack looks like piggy-back attack.The first query execute normally and second query contains malicious query which results in breakdown of database.

Inference:

In this type of attack, based on the value true or for false the query is modified to recast it in the form of action. The attacker tries to you target a secured website so that there is no usable feedback via database error message , change database error messages are unavailable to provide the attacker with feedback an attacker must opt different method to obtain a response from database. After injecting commands observe how they respond and function of website varies. By observing the variation in the response the attacker comes to know additional information about the database and parameters which

are vulnerable. Inference is classified into two categories based on the response

Timing attacks: *information is gathered by observing timing delay of the database to perform this attack the injected query is structured in the form of if then statement who branch predicate response to an unknown about the content of database one of the branches that uses SQL construct that takes a loan amount of time to execute .You know which branch was taken into injection the increase or decrease in response time of database is calculated .*
*Example :SELECT * FROM Table WHERE id='101' or pass= _1'='0'; 2.*

Blind injection*: The information is inferred by asking the server true or false questions .If statement evaluates to be false the page significantly differs from normal functioning page and if the injected statement evaluates to be true then site continues to functions normally.*
*Example: SELECT * FROM Employee WHERE empid='45' & _1'='0';*

RELATED WORK: *Lot of research work is done in the area of SQL injection .Studies discuss attacking mechanism around solution of the problem list of name approach used of researchers:*

S.no	Name	Approach
1.	E.Ramaraj, Indrani[17].	filtration technique known as text based key generator is used to detect and prevent SQL injection attack from accessing database
2	Hee Beng Kuan and Lwin Khin Shar[18].	Authors of this paper proposed use of active attributes to balance static quality in prediction of vulnerabilities
3.	Sonam Panda[19].	RSA and rabin used for prevention of injection
4.	Sampada Gadgil[20].	This paper covered current technique of SQLIA and prevention of attacks.
5.	Anuj Kumar Singh,Kanchan Choudhary[21].	Effective and efficient scheme is developed for SQL injection attacks situated between web application and database.
6.	D.R. Ingle and Dr. B. B. Meshram[14].	This paper explain the need of hybrid tool to prevent attacks
7.	Joshi Padma N,Joshi Padma N[23].	It proposes pattern locking model for validaion of queries passed to the database server.
8	StephenW. Boyd andAngelos D. Keromytis[24].	Propose SQLrand For preventing SQL Injection attacks to the server
9	Harti Nagpa[25].	This paper sheds light on increase in hacking of weakness and security across all geography and types of web technologies.

Prevention for SQLIA*:*

1.SANIA: *Syntactic and semantic analysis for automated Testing against SQLI*
It performs syntactical analysis of SQL query generated by web application. SANIA exploits the syntactical knowledge of SQL query to generate attack request. It should be used during debugging and development phases by web application developer to avoid SQL injection by intercepting SQL query between database and application as well as HTTP request between application and client. Then it checks for any vulnerablities using following steps.

Step1-A web application developer sends innocent HTTP requests to web application to Identifying vulnerable spots.

Step2-Sania generates attack requests that attempt to exploit vulnerable spots where SQLIA may occur by sending attack requests generated from second step and thereby checks if vulnerablities in web application[26].

2.SBSQLID:*Securing Web Application With Service Based SQL Injection Detection[27].Here independent web service is placed to parse the SQL statement. The SQL statement passes through Web Service from the Application server in which there are different modules. These modules prevent SQL injection vulnerabilities. Following are methodology consisting of set of modules to be considered.*

Filtering of Vulnerable characters: *Query validator and Query analyzer is has critical component called SQL statement. The SQL statement follows general*

specifications and rules describing the statement. These statements follow relational algebra which describes statements syntactic structure. Query analyzer module is responsible for Syntactic and semantic structure verification thereby defending SQL injection in web application. While processing the SQL statement in the database server corresponding error message returned from the database server to application server.

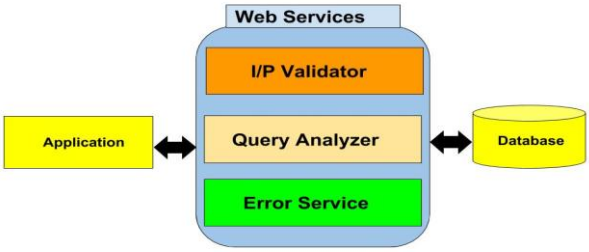

Figure:Service based Architecture

PROPOSED MODEL: *To avoid attacker from breaking in the website, security system Is proposed which can detect SQL injection and cross- site scripting. In simple words we can say that it shall not allow attackers to penetrate into the website.*

Filtration

1: Take User Generated Query SPL[ ] Pattern List with n Anomaly Pattern.
2: Set I equal to 1 & make increment in I by 1 until it is not equal to n repeat & following steps during this loop.
3: Compare all values query length & pattern values if both are same then calculate anomaly value.
4. If there is any anomaly Score Value Anomaly greater or equal to Threshold then query would be rejected.
5. Otherwise return query is accepted.
6: Stop[24].

Detection module: *This module consists of three components. a. Anomaly Detection, b. Positive Security c. Negative Security. These components run parallely for processing .if any one of the threads find the result, other two threads will be immediately stopped. if there are no vulnerabilities the authentication is done otherwise exception is thrown. To overcome the drawback of performance parallel processing is necessary.*

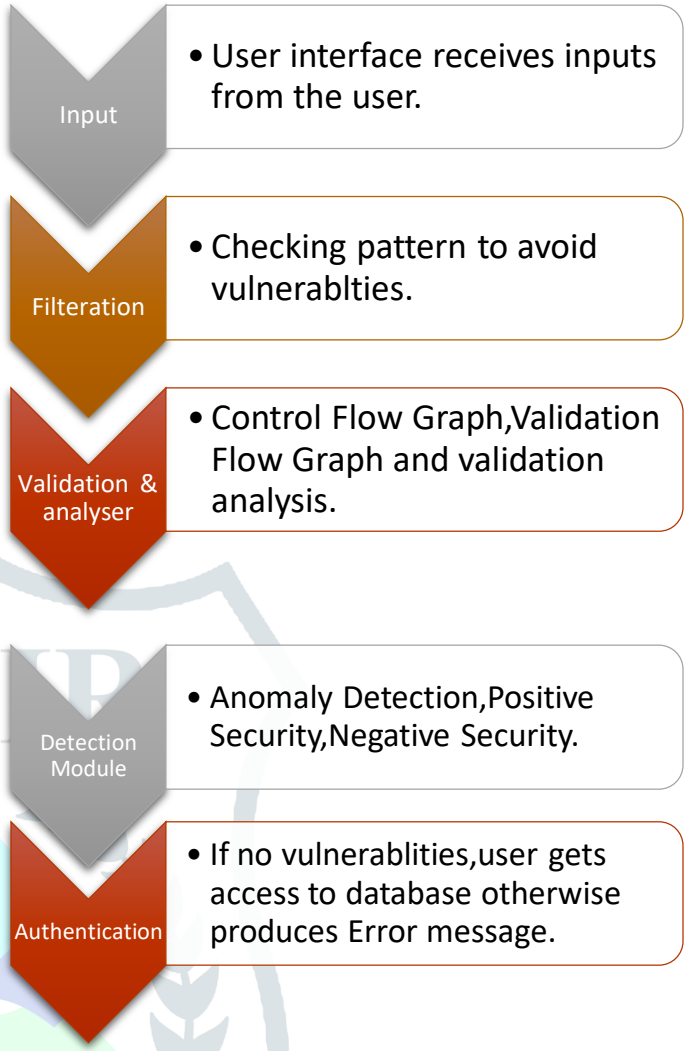

Figure : Process flow diagram.

a. Anomaly Detection: in the learning phase anomaly detection component comes across different attribute values of log entry different statistical models like length attribute, character distribution, case based, reasoning and structural inference.

b. Positive Security :Non binign input which has signatures of allowed tags SQL queries for URL in the form of of regular expression. Example list items<li>, ordered list<ol>, unordered list<ul>.

c. Negative Security: Security component contains signature of of variety of attacks like cross site scripting Example: "< img src=javascript>" and "< script> alert(document. cookie)</ script>".Regular expression "((\%3C)|<)[^\n]+((\%3E)|>)" can catch almost any remote attempt to attack XSS with very few false positives. It checks for occurrence of "<" or hex-equivalent, zero or more non-new line characters and then ">" or hexequivalent. Similarly Regex for SQL Injection

Attack "\w(\')|(\s((%6F)|o|(%4F))((%72)|r|(%52)))|((\%3D)|(=))[^\n]*((\%27)|(\'))|((\)|(\%3B)|(\%23)|(#)|(exec))" can mitigate the tautology, comments, exe and apostrophic SQL Injection attacks.*

Validation and Analyzer Module: *This module analyses the provided information that develop control flow graph and then validation flow graph from the source for input validation. It takes care of semantic and syntactical validation. Validation report is generated for the developer.*

CONCLUSION:

In this paper we discuss about secure software development and its devastating impact on different groups. We have discussed about Validation attacks like cross site scripting(XSS), SQLI, buffer overflow attack And their prevention techniques. I came to a conclusion that no system is full proof Defensive system.SQL injection occurs when Unreliable content is sent through the queries. XSS hijacks sessions of users and redirects to malicious sites.BOF occurs when process or Program tries to store more than intended data. Counter attackers malicious injection the proposed model filters the input for SQL injection and Cross - site scripting using detection module for Signature for different attacks and validation and analyzing module for control flow graph and validation flow graph and also semantic and syntactical validation.

FUTURE SCOPE : The proposed model has security loop and is vulnerable to other attacks like Buffer overflow attack. There is a need of more advanced and sophisticated tool which can prevent intruders . The scope of this research can be extended by developing such tool.

AKNOWLEDGEMENT: *I would like to extend my sincere gratitude to Dr.Mustafa bastikodi-Hod of CSE department and Dr.Rimas faizabad-faculty for their guidance and support throughout my research work*

REFERENCES

1. Building Security In Editor: Gary McGraw, gem@cigital.com "Software Security".

2.Cognizant-"Application Security: Safeguarding Data, Protecting Reputations".

3. Zorz, Zeljka, "Belgian bank Crelan loses €70 million to BEC scammers, " January 26, 2016, www.helpnetsecurity.com.

4. Ashford, Warwick, "More than 300,000 US taxpayers affected by data breach," August 18, 2015, www.computerweekly.com.

5. Thielman, Sam, "ProtonMail: encrypted email provider held ransom by hackers," November 5, 2015, www. theguardian.com.

6. Riley, Charles, " Insurance giant Anthem hit by massive data breach" (www.cnn.com, February 6, 2015.

7. BBC UK, "TalkTalk cyber-attack: Website hit by 'significant' breach" (www.bbc.co.uk, October 23, 2015.

8. Jacob Jose, Tibin T. Tomy, Vibin Karunakaran , Anjali Krishna V, AnoopVarkey, Nisha C.A.''Securing Passwords from Dictionary Attack with Character-Tre".

9. Dictionary attack sample program. [Online]. Available: https://github.com/npapernot/dictionary-attack.

10. Ding Wang, and Ping Wang, "Offline dictionary attack on password authentication schemes using smart card," 16th Information security conference (ISC 2013) SpringerVerlag., Vol. 7807, pp. 1-16, 2013.

11. Ossama B. Alkhurafi, and Mohammad A. AlAhmad, "Survey of web application vulnerability attacks," 4th International conference on advanced computer science applications and technologies., pp. 154-158 , 2015.

12. Yaashwanth.C., and Dr.R.Ramesh, "Attacks in web based Embedded Applications," IJCSI International journal of computer science Issues., vol. 7, pp. 116-119 , November 2010.

13. D.R. Ingle and Dr. B. B. Meshram," ATTACKS ON WEB BASED SOFTWARE AND MODELLING DEFENCE MECHANISMS", International Journal of UbiComp (IJU), Vol.3, No.3, July 2012.

14. Liang Guangmin(2008), "Modeling Unknown Web Attacks in Network Anomaly Detection" Computer Engineering Department Shenzhen Polytechnic, Shenzhen 518055, China Email: gmliang@oa.szpt.net Third 2008 International Conference on Convergence and Hybrid Information Technology

15. Open Web Application Security Project. The ten most critical Web application security vulnerabilities. http://umn.dl.sourceforge.net/

sourceforge/owasp/OWASPTopTen2004.pdf, 2004, visit on 2005/10/05.

16. C. Anley. Advanced SQL Injection In SQL Server Applications. White paper, Next Generation Security Software Ltd., 2002. , M. Howard and D. LeBlanc. Writing Secure Code. Microsoft Press, Redmond, Washington, second edition, 2003.

17. Indrani B., E. Ramaraj (2012) "An Efficient Technique for Detection & Prevention of SQL Injection Attack using ASCII Based String Matching" International Conference on Communication Technology & System Design .

18. Lwin Khin Shar & Hee Beng Kuan (2013)Tan Mining SQL Injection & Cross Site Scripting Vulnerabilities using Hybrid Program Analysis.

19. P. Sonam,"Protection of Web Application against Sql Injection Attacks", International Journal of Modern Engineering Research Vol.3, Issue.1, Jan-Feb. 2013 pp-166-168 .

20.Sampada Gadgil 2015 SQL injection attacks & prevention techniques.

21.Kanchan Choudhary,Anuj Kumar Singh (2016) "A Modified Scheme for Preventing web Application against SQL Injection Attack", International Journal of Computer Applications (0975 – 8887) Volume 141 – No.10, May 2016 [11] Manju Khari, Parikshit Sangwan (2016).

23. Joshi Padma , Dr. N. Ravishankar ,Dr. M. B. Raju ,N.CH. Ravi "Encountering SQL Injection in Web Applications", Proceedings of the Second International Conference on Computing Methodologies and Communication (ICCMC 2018) IEEE Conference Record # 42656; IEEE Xplore

24 StephenW . Boyd and Angelos D.Keromytis ." SQLrand : Preventing SQL Injection Attacks".

25.Bharti Nagpal Naresh Chauhan Nanhay Singh1(2016) "security engine for CSRF, SQL injection & XSS attacks", Division of Operation & Maintenance, Lulea University of Technology, Sweden 2016 .

26. Jin-Cherng Lin and Jan-Min Chen(2006), "An Automatic Revised Tool for Anti-malicious Injection", in Proceedings of The Sixth IEEE International Conference on Computer and Information Technology.

27. Politecnico di Milano(2009), "Integrated Detection of Attacks Against Browsers", European Conference on Computer Network Defense,.