IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 20, Issue 5, Ver. II (Sep - Oct 2018), PP 73-83
www.iosrjournals.org

New Technique For Preventing SQL Injection Attack Based On Normal Use Model

Anjali Gupta[1], Amita Dhankhar[2] , Kamna Solanki[3]
M.Tech Student , Assitant Professor ,Associate Professor
Department Of C.S.E. Uiet, M.D University, Rohtak , Haryana India
Corresponding Author: Anjali Gupta

Abstract: *Online applications have turned into an essential piece of our day by day lives, yet in the meantime, security of digital data put away in the web databases has been a developing concern. SQL injection attacks have been an overwhelming and inescapable security risk on web applications in the course of the last one and half decades. The helplessness has been abused by attackers bringing about enormous information break episodes. Advanced and mechanized attack devices have made the circumstance disturbing than at any other time. In spite of broad research on relief of SQL injection attacks, the threat has persevered with almost a similar force. Over some time, SQL injection attack vectors have developed with new structures and highlights, which have made the issue considerably all the more difficult. This paper introduce a new Prevention mechanism of SQL injection attacks at the application level; then continue into propose solution for server level insurance which is particularly valuable on shared facilitating situations. Aside from exactness and execution, it additionally emphasize on simplicity of usage, platform freedom, and versatile learning in the studies.*

Keywords: *SQL, Injection Attack, Normal Use Model.*

Date of Submission: 02-10-2018 Date of acceptance: 19-10-2018

I. Introduction

Now day's online applications have turned into an indispensable piece of our everyday lives. The Internet and web applications are the cutting edge working environment and business ground adding to drive the world economy. In the meantime, programmers and attackers have built up a parallel underground economy of hacking into web applications and taking huge measure of delicate business-basic data with malignant expectation. Occurrences of information break have turned out to be visit making gigantic monetary misfortunes associations and genuine effect for the clients in different levels.

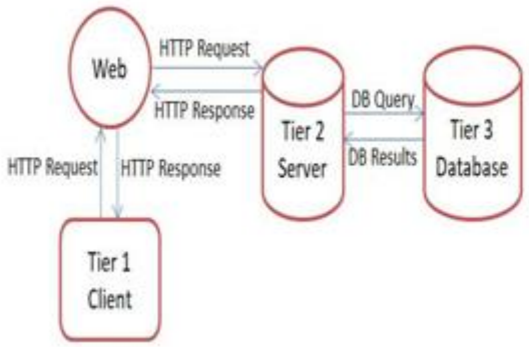


Fig 1 -Tier Architecture of Web
Application

Securing web applications according to attackers has turned into an essential concern . With the help of SQL injection attack, an attacker can remove, change, or crush the backend database of web applications. The effortlessness of attacking a web application utilizing SQL injection and plenitude of helpless applications on the Internet has generally added to across the board information rupture occurrences.[1][3] 3-level design of web application is spoken to in the Figure 1 The essential motivation behind the content of a database-driven dynamic site page is tocreate the HTML content in light of the sources of info got from the client. For instance, a website page on a news webpage may demonstrate articles identified with a specific classification, for example, Sports, Politics, and so on. Based on the value in URL string, an inquiry page would demonstrate the outcomes in view of the watchwords entered by the client in an info box.[4][5] For the most part, a site page gets such contributions through the URL question string parameters or potentially shape fields. Qualities from Cookies and other HTTP headers incorporated into the demand are additionally different types of information sources and might be utilized as a part of the program rationale as required. The server-side content or program of the website page utilizes these contributions to question the backend database and recover the outcomes powerfully, which are then used to produce the HTML content on the travel to be sent back to the program. At the point when these sources of info are misused by an attacker by infusing deformed information into a web application to cause the web application execute contrastingly or in a startling way, it is named as injection attack **[5-8].** This class incorporates following Attacks:

1. XSS attacks
2. Header injection
3. Log injection
4. SQL Injection

II. Literature Review

Borrajo et.al. - A Review Of Machine Learning For Automated Planning (2009)
This paper surveys late systems in machine learning for the programmed meaning of arranging learning. It has been sorted out as indicated by thsse objective of the learning procedure: programmed meaning of arranging activity models and programmed meaning of arranging control learning.[1]

Kotsiantis - Supervised Machine Learning: A Review of Classification Techniques (2007)
This paper depicts different directed machine learning grouping methods[2]

Seghal et.al. - A Review Of Improving Software Quality Using Machine Learning Algorithms (2017)
In this paper programming expectation display used to control the classes of programming which are regularly to change. Machine learning calculations are utilized for anticipating programming.[7]

Portugal et.al. - The Use of Machine Learning Algorithms in Recommender Systems: A Systematic Review(2010)
This paper exhibits an efficient audit of the writing that examines the utilization of
Machine learning calculations in recommender frameworks and distinguishes look into open doors for programming building research.[12]

Barua1 - Review on Machine Learning Algorithms in Handling EEG Artifacts(2006) This paper gives a survey of machine learning calculations that have been connected in EEG antiquities taking care of, for example, relics ID and expulsion. Furthermore, an examination of these strategies has been accounted for in light of their execution.[6]
Carrio et.al. - Review of Deep Learning Methods and Applications for Unmanned Aerial Vehicles(2017)
In this paper, an intensive audit has been performed on late announced employments what's more, uses of profound learning forUAVs, including the most applicable advancements and also their exhibitions and restrictions.[14]

Dhage et.al. - A Review On Machine Learning Techniques(2009)
The fundamental objective and commitment of this audit paper is to exhibit the outline of machine learning and gives machine-learning strategies. Additionally paper audits the benefits and negative marks of different machine learning calculations in distinctive methodologies[15]

Khan et.al. - A Review of Machine Learning Algorithms for Text-Documents Classification (2010) This paper gives an audit of the hypothesis and strategies for record grouping and content mining, concentrating on the current writing.[18]

Das et.al.- Survey on Machine Learning: Concept, Algorithms and Applications (2017)
This paper centers on clarifying the idea and development of Machine Learning, a portion of the famous Machine Learning calculations and endeavor to look at three most well-known calculations in view of some essential thoughts

Domingos - A Few Useful Things to Know about Machine Learning(2011)
This article compresses twelve key lessons that machine learning specialists and

professionals have learned. These incorporate traps to dodge, critical issues to center around, furthermore, answers to normal inquiries.[13]

Pintelas et.al. - Machine learning: a review of classification and combining techniques (2007)
This paper portrays different order calculations and the current endeavor for enhancing arrangement precision—gatherings of classifiers.[4]

Bellinger et.al. - A systematic review of data mining and machine learning for air pollution epidemiology (2017)
Author led a precise writing survey on the utilization of information mining and machine learning strategies in air contamination the study of disease transmission. Author completed our hunt procedure in PubMed, the MEDLIE database and Google Scholar.[16]

Singh - A Review Of Studies On Machine Learning Techniques(2008)
This paper introduces the most regularly utilized machine learning methods, for example, neural systems, case based thinking, order and relapse trees, manage acceptance, hereditary calculation and hereditary programming for master estimation in the field of programming advancement[25]

Ortiz et.al –A Review of Classification Problems and Algorithms in Renewable Energy Applications(2016)
The paper additionally gives an exhaustive writing survey and talk on various characterization strategies in particular RE issues, including wind speed/control forecast, blame analysis in RE frameworks, control quality unsettling influence arrangement and different applications in elective RE frameworks.[17]

Dey et.al.-Machine Learning Algorithms: A Review (2016)
In this paper, different machine learning calculations have been talked about. These calculations are utilized for different purposes like information mining, picture handling, and prescient examination, and so on to give some examples.[24]

Bhatt et.al. - A Review Paper on
Machine LearningBase Recommendation System (2014)
This paper arranges community oriented separating in two kinds: Memory based and Model based Recommendation .The paper explains these methodologies and their strategies with their constraints. This study demonstrates the guide for esxamine around there.[20]

Catesa et.al. - A Machine Learning Approach To Research Curation For Investment Proces (2017) This volume presents a test for venture experts who need to utilize bits of knowledge from scholarly research to create productive speculation methodologies[21]

Pahwa - Stock Prediction using Machine Learning a Review Paper (2017)
This paper reviews the machine learning calculations reasonable for such an application; also it examines what are the present apparatuses and methods suitable for its execution.[22]

Cihan –A Review Of Machine Learning Applications in Veterinary Field (2017)
In this survey, it was watched that the neural system, calculated relapse, straight relapse, different relapse, guideline segment examination and k-implies techniques were much of the time utilized in inspected Distributions and machine learning application in veterinary field upward energy.[19]

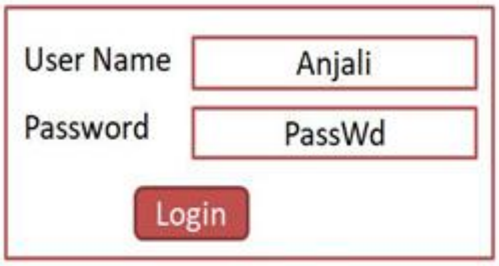
3.SQL Injection Attacks

SQL Injection Attack is a subset of the unsubstantiated information and happens when an attacker endeavors to change the rationale, semantics or punctuation, and conduct of an honest to goodness SQL statement by including extra SQL statement or administrators into the announcement through the URL inquiry string parameters or shape fields, typically with a noxious purpose. By infusing uniquely made contributions with SQL delimiters, keyword, administrators, and so on. The attacker endeavors to change the consequences of the SQL question, in this waadjusting the HTML content returned by the server. Utilizing this injection procedure, the attacker can increase unapproved access to limited regions of a web application, and furthermore have the capacity to recover, adjust, or harm the data in the backend database.

3.1 Mechanism of SQL Injection Attacks

In this software engineer has composed the code as show in figure 3 in programming dialect with a specific end goal to confirm whether the client is approved to sign in or not, by checking the provided client name and secret key against the information in the backend database. On the off chance that an honest to goodness and legitimate client visits the sign in page and enters 'Anjali' in the client name box and 'passwd' in the secret word box, at that point the subsequent.

Fig 2 Basic Login Form

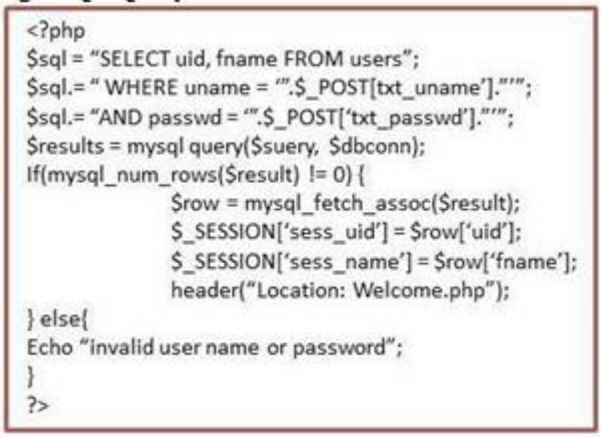


User Name	Anjali
Password	PassWd

SQL question will be as in figure 4. The question will restore an outcome if there is a record with the given client name and keyword in the 'clients' database table. It might be watched that the POST esteems got from the shape fields have been utilized straightforwardly in the code to build the dynamic SQL inquiry.

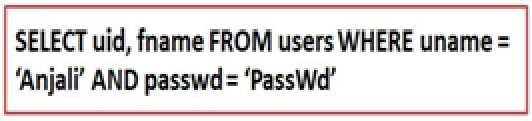
It might be watched that, the single-quote character entered in the client name field fills in as a string delimiter in the SQL Let attacker visits the sign in page and enters the qualities as appeared in Fig. 5 not surprisingly, these qualities will likewise be submitted to the page on the server, and the PHP code will utilize the qualities to build the dynamic SQL question by string link. The SQL question would be as show in figure 6

Fig 3 SQL Query



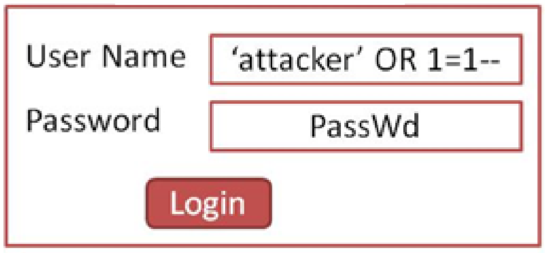
```
<?php
$sql = "SELECT uid, fname FROM users";
$sql.= " WHERE uname = '".$_POST[txt_uname'].""";
$sql.= "AND passwd = '".$_POST['txt_passwd'].""";
$results = mysql query($suery,  $dbconn);
If(mysql_num_rows($result)  != 0) {
                    $row = mysql_fetch_assoc($result);
                    $_SESSION['sess_uid'] = $row['uid'];
                    $_SESSION['sess_name'] = $row['fname'];
                    header("Location: Welcome.php");
} else{
Echo "invalid user name or password";
}
?>
```

Fig 4 Attacker login



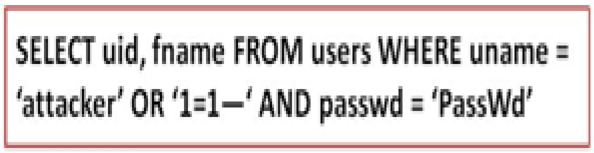
```
SELECT uid, fname FROM users WHERE uname =
'Anjali' AND passwd = 'PassWd'
```

Fig 5 attacker login query



User Name	'attacker' OR 1=1--
Password	PassWd

Fig 6 Attacker SQL



```
SELECT uid, fname FROM users WHERE uname =
'attacker' OR '1=1—' AND passwd = 'PassWd'
```

The twofold dash toward the finish of client name is a remark character according to SQL language structure, which successfully remarks out whatever is left of the question. Whenever executed, the outcome set will contain all records from the clients table in light of the fact that the condition "OR 1 = 1" assesses to valid for all columns. The record pointer will be situated at the main record, which could be some other client. Since the quantity of columns isn't zero, the if() condition will assess to genuine, and the attacker will obtain entrance into the limited zone of the web application with the character of the principal client account in the clients table. intermediary set between the web application and database server de-randomizes the inquiries by expelling the arbitrary key after the question effectively parses against a changed SQL language. Likewise, AMNESIA and CANDID utilize static investigation to discover hotspots in the application's code, instrument it with extra code to perform observing at runtime. SQLProb works at the database firewall layer however requires learning of every conceivable question a web application can produce. Swaddler requires change of the PHP mediator to catch runtime data, which no facilitating supplier would permit on a common server.

4. Existing Techniques

Being a genuine danger to web applications for almost 15 years, magnificent research on SQL infusion avoidance and recognition exists in the writing. In spite of the fact that numerous methodologies have been recognized as recognition or avoidance strategies, just some of them have been actualized from imminent of common sense. Furthermore, due to non-accessibility of any standard dataset of SQL infusion assaults, the majority of the methodologies have been approved utilizing engineered ones or

straightforward applications without mirroring true situations. A portion of the methodologies have been approved against a couple of CVE IDs of some open source application like phpBB, phpNuke, and so on. A few scientists have utilized the AMNESIA test bed, yet the applications included are very little and a long way from true applications. We pick 6 very refered to works for making a correlation. These are: SQL-DOM **[19]**, Swaddler [20], AMNESIA [21], CANDID [22], SQLProb [23] and SQLRand [24]. Every one of these methodologies has their advantages and additionally deficiencies under particular circumstances and frameworks. For instance, the SQLRand approach requires the software engineers to randomize the SQL inquiries in the code by annexing an arbitrary key to each SQL catchphrase. An

5.New Techniques for Preventing SQL Injection Attacks

5.1 Introduction

New technique for prevention of SQL injection attacks will compresses of following two steps:

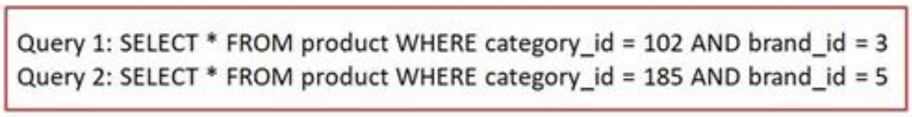
1. Creating a vault of the real queries produced by the web application amid its typical use inside a secured domain
2. Comparing the queries created at runtime against in the web application with the achieved queries.

Whenever; Runtime query does not match with achieved queries than suspicious activity is found. An abnormal query is hindered by not sending it to the database server for execution, along these lines disposing of the likelihood of a SQL injection attack. In spite of the fact that the approach is actually straightforward, simple to execute, and very compelling in avoiding SQL injection attacks, there are two noteworthy difficulties to overcome:

1. By decreasing the size of achieved query repository
2. Performing quick and productive comparison at runtime

Difficulties with put away SQL questions are:

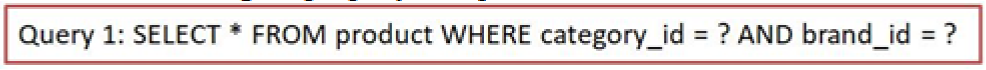
1. The size of the authentic inquiry store must be limited.



```
Query 1: SELECT * FROM product WHERE category_id = 102 AND brand_id = 3
Query 2: SELECT * FROM product WHERE category_id = 185 AND brand_id = 5
```

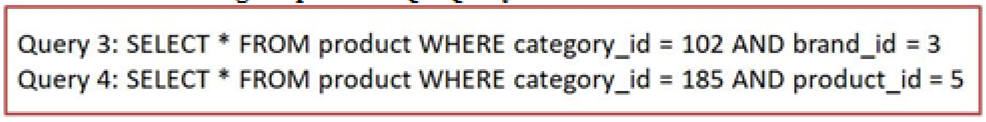
2. The second test is the handling overhead forced at runtime by the abnormality finder.

Fig 7 SQL Query example for Normal Use Model



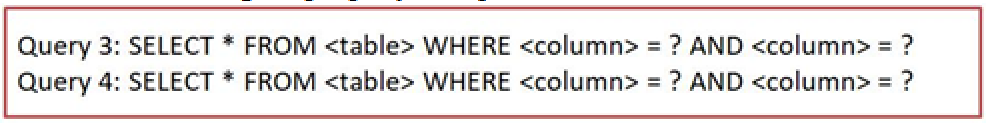
```
Query 1: SELECT * FROM product WHERE category_id = ? AND brand_id = ?
```

Fig 8 Optimize SQL Query for Normal Use Model



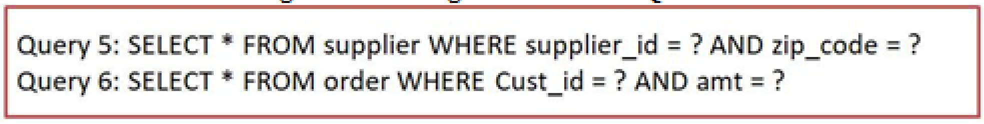
```
Query 3: SELECT * FROM product WHERE category_id = 102 AND brand_id = 3
Query 4: SELECT * FROM product WHERE category_id = 185 AND product_id = 5
```

Fig 9 SQL Query example for Normal Use Model



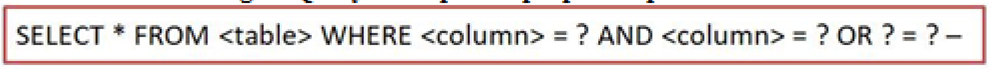
```
Query 3: SELECT * FROM <table> WHERE <column> = ? AND <column> = ?
Query 4: SELECT * FROM <table> WHERE <column> = ? AND <column> = ?
```

Fig 10 Re-Writing Parameterized Queries



```
Query 5: SELECT * FROM supplier WHERE supplier_id = ? AND zip_code = ?
Query 6: SELECT * FROM order WHERE Cust_id = ? AND amt = ?
```

Fig 11 Query Example for proposed optimization



```
SELECT * FROM <table> WHERE <column> = ? AND <column> = ? OR ? = ? –
```

Fig 12 Proposed Structural SQL Query in Tautological Attack

A typical approach that has been utilized as a part of the writing in such manner is to speak to the true blue property estimations by conservative consistent expression articulations or totally remove values and store just the skeletal type of the inquiries. For instance, consider the two SQL inquiries in figure 7. These two questions vary just in the estimations of "category_id" and "brand_id" traits in the WHERE provision. The skeletal type of the query is acquired by substituting the value with a placeholder ―?‖. It can be seen that the two query decrease to an indistinguishable shape from in figure 8

This shape is likewise called as the parameterized type of the query. Rather than putting away every query, just the parameterized frame can be put away in the archive. This decreases the pursuit space with a factor. In any case, consider the two queries in figure 9 which are as of now in the parameterized shape. Both of these parameterized frames must be put away on the grounds that one of them utilizes the brand_id property while alternate uses the supplier_id trait in the WHERE statement. In this way, however parameterized shape helps in space lessening, it can't limit the extent of the genuine query vault.

5.2Optimization performed

It is sheltered to expect that each unique question produced by a web application is similarly defenseless against some type of injection attack. An attacker infuses SQL code portions keeping in mind the end goal to modify the structure of the dynamic question, subsequently driving it to restore an alternate outcome set than the software engineer initially proposed. In this way, how a query is organized is more critical than the semantics of the query. We recommend that, aside from the keywords, extraordinary characters and keywords, the identifiers and exacting esteems in a SQL inquiry, are immaterial towards the basic creation of the query. By disregarding the identifiers from the two parameterized queries Q3 and Q4 in figure 7, queries can be re-composed as in figure 10.It might be watched that both the queries now lessen precisely to the same basic shape. Thusly, putting away the auxiliary frame in the vault is more useful than putting away the parameterized shape, yet at the same time satisfactory to show the real queries. This is a straightforward yet intense thought that understands a negligible model of queries. For instance consider two more queries in figure 11 issued for execution on various tables.By the rule of overlooking thetable and section names, it is anything but difficult to see that both Q 5 and Q 6 will likewise deliver an indistinguishable auxiliary shape from Q1 to Q4. This basically implies these queries have the very same basic arrangement. In this way, by summing up the identifier tokens, a wide range of queries having same basic synthesis, will deliver a similar inquiry display, and the span of the true blue query vault can be limited. Presently let, Queries succumbs to tautological SQL injection attack like "OR 1 1 - " as considered in past segment. At that point the auxiliary type of the infused inquiry in new proposed shape will be as in figure 12.

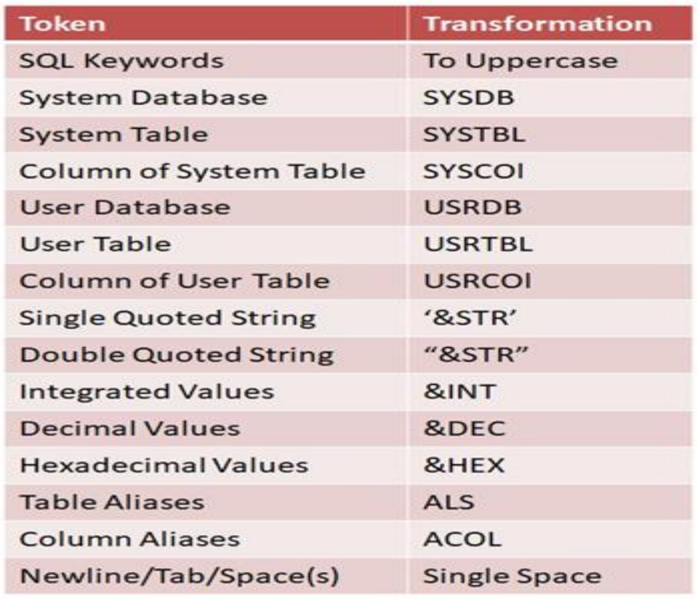
Obviously, this is not the same as the basic type of the genuine queries; thus it can be effortlessly decided as bizarre. The real favorable position with this approach is:

1. Number of queries which are semantically unique can possibly decrease to the same basic shape, this makes repository of legitimate query size is minimal.
2. Secondly it will improve the performance.

5.3 The query Transformation Scheme Proposed approach is to change the legal queries into their basic frame by summing up the identifier tokens alongside exacting esteems. For this, a transformation conspire is proposed which is spoken to in figure 13 the transformation plot is intended to deal with every single conceivable assortment of queries, including any query that references objects from the framework databases.
The greater part of the above transformations will be finished by substring substitution utilizing str_replace() elements of PHP in well-ordered way. Following the query transformation process, the queries create an indistinguishable basic shape from in figure 14. It might be watched that the changed inquiry saves the basic structure of the first query alongside the information sorts of characteristic esteems and how they are delimited. Thusly, if an attacker endeavors to include string esteem where an integer is normal, it will at present trigger an irregularity even with no SQL infusion.

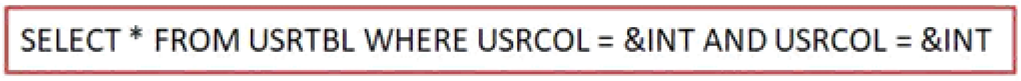
5.4 Hashing Transformed Queries

The query transformation delivers an insignificant arrangement of legal SQL queries in their basic shape, which can be put away as normal use model. In spite of the fact that changing the queries into basic shape limits the extent of the store, it doesn't encourage quick and effective looking at runtime. This is on account of, putting away the auxiliary types of the legal queries will require direct search at runtime, which will be costly. Along these lines, it is proposed to apply a reasonable hashing capacity to create extraordinary hash-keys for each real basic frame. Producing hash-keys of the changed queries offers many accompanying preferences The hash-keys can be considered. For best execution of the framework, it is however fundamental that the hash capacities is computationally quick and has low impact rate. PHP form 5 or more, has built in bolster for more than 40 hash calculations through its hash() work which create hash-keys extending from 8 to 128 characters. Because of littler key size, these are more inclined to impacts, in this manner for the most part utilized for checksums. Then again, calculations delivering bigger keys are exceedingly impervious to impacts, yet would require more storage room. Thinking about these certainties, a key size of 32 characters is a perfect tradeoff for our approach. Hash keys of 32 characters are broadly utilized for record and secret word hashes in numerous applications. Despite the fact that MD4 calculation has the best execution over MD5 as MD5 calculation is around 20% slower than MD4, yet it has a high crash protection of 220.96 and uniform dispersion of produced hash esteems. Accordingly, it is chosen that MD5 hashing is most reasonable for creating hash-keys of the changed queries. There is likewise a different md5() work accessible in PHP 5, which is barely quicker.



Token	Transformation
SQL Keywords	To Uppercase
System Database	SYSDB
System Table	SYSTBL
Column of System Table	SYSCOl
User Database	USRDB
User Table	USRTBL
Column of User Table	USRCOl
Single Quoted String	‘&STR’
Double Quoted String	“&STR”
Integrated Values	&INT
Decimal Values	&DEC
Hexadecimal Values	&HEX
Table Aliases	ALS
Column Aliases	ACOL
Newline/Tab/Space(s)	Single Space

Fig 13 Proposed Transformations Scheme



```
SELECT * FROM USRTBL WHERE USRCOL = &INT AND USRCOL = &INT
```

Fig 14 SQL Query after Transformations Scheme

5.5 System Architecture
The design of the framework is spoken to in Figure 15. The framework works in two phases:

Learning Phase
In the learning phase, the normal use model for query is prepared. The model is used at runtime to distinguish atypical queries.

1. The initial step comprises of gathering all conceivable authentic queries which can be produced by the web application amid typical use inside a secured domain. Genuine information esteems are given wherever outside contributions from the client are required.
2. The arrangement of real queries gathered is contribution to the Query Transformation module. It changes the SQL queries into basic shape by applying the change. The yield of this progression is the basic type of every single lawful query which can be produced by the web application.
3. The hash-keys of the one of kind basic structures are figured by applying MD5 hashing on each auxiliary frame. The hash-keys connote the marks of the real queries amid typical utilization of the framework.
4. The hash-keys are put away in the database server in a different table. This table requires just a single section of CHAR(32) information compose. Since the hash-keys are remarkable, an essential file is made by making the section the primary key of the table.

Runtime Phase
Amid runtime, the framework works in the accompanying way:
1. At the point when a client gets to a dynamic page, SQL queries are created by the application utilizing the outer sources of info given by the client assuming any.
2. Each SQL query is first changed into its basic frame by applying a similar change conspire.
3. The MD5 hash estimation of the basic shape is registered. The hash esteem value signifies the mark of the runtime query.
4. The hash esteem is turned upward in the database table utilizing the essential list. There are two conceivable results of this progression. Either the hash esteem exists in the legitimate queries table or it doesn't exist.
5. In the event that the hash key is found in the legitimate queries table, it implies that the mark of the runtime query matches with that of a query amid ordinary utilization of the application. In this way, the SQL query is affirmed to be bona fide and permitted to execute by sending it to the database server.
6. In the event that the hash key isn't discovered, at that point the query is abnormal. For this situation, the query is rejected and a mistake is activated back to the web application.

Along these lines, any SQL query that is created at runtime is checked against normal use model. A dynamic query that is delivered utilizing attacker client inputs is recognized by hash-coordinating of its basic shape. Probability of SQL infusion assaults is wiped out by keeping strange queries from execution.

6 Comparisons with Existing
For new characterized method, contrasting it and announced experimental outcomes would be misleading as it won't give anobjective correlation. We in this way describe the accompanying criteria with practical approach in reality, for making a systematic correlation of characterized with alternate methodologies:

1. Specific Coding: Whether the approach depends on a particular coding technique or utilization of a proposed system to avoid SQL infusion assaults.
2. Source Code Access: Whether the approach expects access to the source code or alters it for static examination, retrofitting, or enlarging with extra code.
3. Platform Specific: Whether the approach is material or reasonable just for a particular programming dialect as well as database stage.
4. Normal Use Method: Whether the approach requires building a model of the SQL inquiries produced by the web application with considerate information sources keep running in an anchored domain.
5. Multiple Websites: Does the approach apply to just a single web application? At the end of the day, would it be able to secure various web applications facilitated on a common server?

6. Adaptive: Whether the approach is versatile and self-figuring out how to have the capacity to identify beforehand concealed assaults.

7. Time Complexity: Whether the general time multifaceted nature of the framework is high or low.

8. Usability: How well the framework is for all intents and purposes implementable and usable in a genuine generation condition.

In view of the above attractive criteria, the logical examination of our approach with the chose approaches is appeared in figure

16. Unmistakably, New Defined Technique

performs superior to existing methodologies. New Defined Technique was likewise contrasted and these methodologies in view of the capacity to forestall or identify different kinds of SQL infusion assaults as appeared in figure 17. New Defined Technique recognized the greater part of the tautological attack vectors in the tests, it is still demonstrated as 'fractional'. Hypothetically, tautological attack can be framed in interminable number of ways, and such a claim would be improper.This paper compresses twelve key lessons that machine learning specialists andprofessionals have learned. These incorporate traps to dodge, critical issues to center around, furthermore, answers to normal inquiries.

III. Conclusions

A singularity based framework was exhibited for forestalling SQL injection attacks on a web application. The approach depends on setting up a model or archive of all conceivable SQL questions that the application can produce amid its normal use inside a secured situation. The model is then used to identify irregularities in runtime questions to counteract SQL injection attacks. In spite of the fact that the strategy is fundamentally straightforward, size of the normal query store and performing proficient seeking at runtime were two difficulties to overcome.

The principal challenge was tended to by planning a query transformation plan to change over SQL inquiries into their basic frame. This presented an original thought that, the identifiers and literals in a SQL query are unimportant towards the basic type of a query, which is the way to limit the measure of the normal query model.. As the approach depends on a normal-use query model, it offers 100% security against SQL injection attacks with irrelevant effect on execution. The main downside of the framework is that, at whatever point the web application is altered with changes at the SQL query level, the normal query model must be recovered.

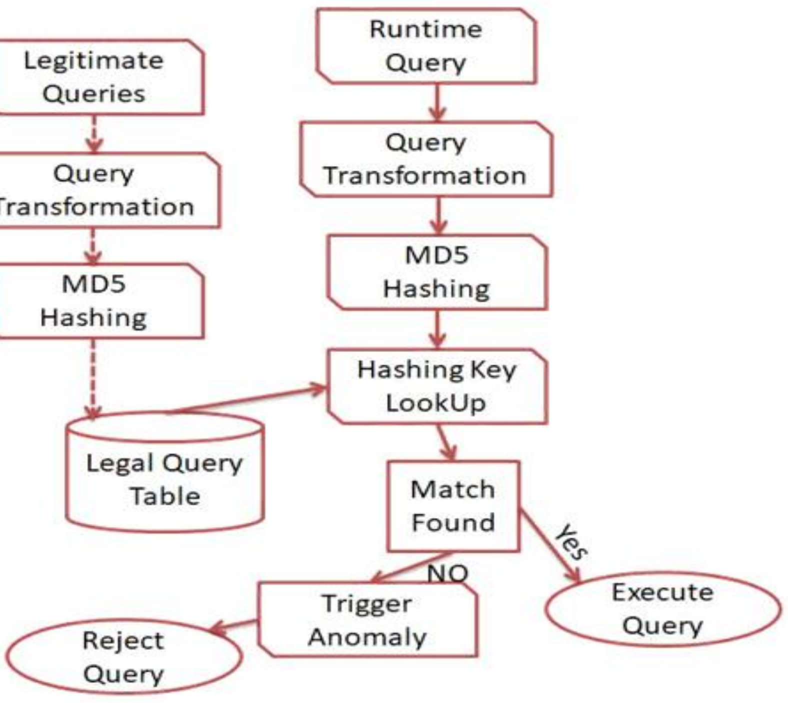
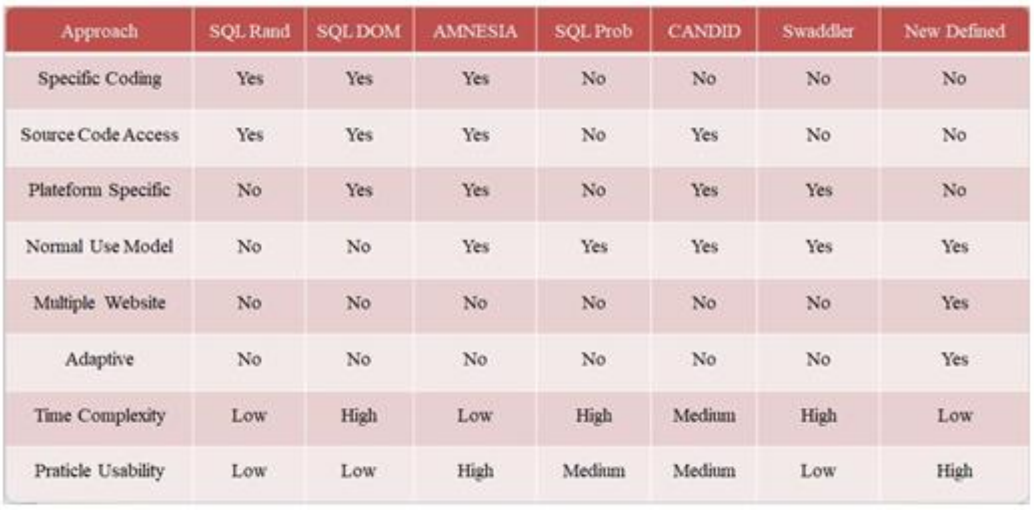
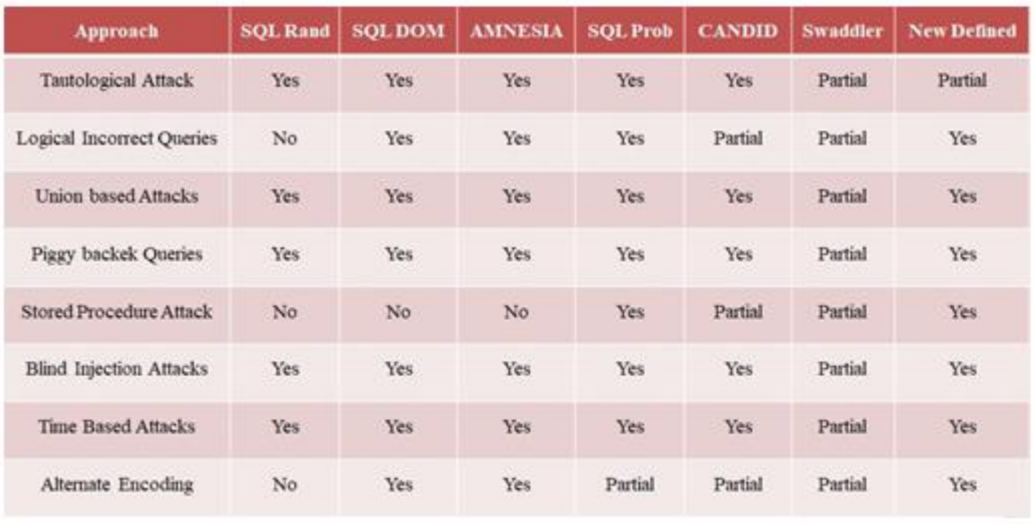


Fig 1.5: New System Architecture



Approach	SQL Rand	SQL DOM	AMNESIA	SQL Prob	CANDID	Swaddler	New Defined
Specific Coding	Yes	Yes	Yes	No	No	No	No
Source Code Access	Yes	Yes	Yes	No	Yes	No	No
Plateform Specific	No	Yes	Yes	No	Yes	Yes	No
Normal Use Model	No	No	Yes	Yes	Yes	Yes	Yes
Multiple Website	No	No	No	No	No	No	Yes
Adaptive	No	No	No	No	No	No	Yes
Time Complexity	Low	High	Low	High	Medium	High	Low
Praticle Usability	Low	Low	High	Medium	Medium	Low	High

Fig 16: Comparison with existing



Approach	SQL Rand	SQL DOM	AMNESIA	SQL Prob	CANDID	Swaddler	New Defined
Tautological Attack	Yes	Yes	Yes	Yes	Yes	Partial	Partial
Logical Incorrect Queries	No	Yes	Yes	Yes	Partial	Partial	Yes
Union based Attacks	Yes	Yes	Yes	Yes	Yes	Partial	Yes
Piggy backek Queries	Yes	Yes	Yes	Yes	Yes	Partial	Yes
Stored Procedure Attack	No	No	No	Yes	Partial	Partial	Yes
Blind Injection Attacks	Yes	Yes	Yes	Yes	Yes	Partial	Yes
Time Based Attacks	Yes	Yes	Yes	Yes	Yes	Partial	Yes
Alternate Encoding	No	Yes	Yes	Partial	Partial	Partial	Yes

Fig 17: Comparison with existing

References

[1]. Jim,S., and Borrajo,k.(2009) "**A Review Of Machine Learning: For Automated Planning**", The knowledge engineering review, cambridge university press, vol. 01,pp. 1–24.

[2]. Kotsiantis,B.(2007) "**Supervised Machine Learning: A Review Of Classification Techniques**", Department of Computer Science and Technology University of Peloponnese, Tripolis GR ,Vol .5,pp.45-67.

[3]. McClure,R. and Kruger,I.(2005) "**SQL DOM: Compile Time Checking of Dynamic SQL Statements**" International Conference on Software Engineering ,vol31, pp. 88–96.

[4]. Cova,M. and Pintelas,H.(2007) " **An Approach for the Anomaly-based Detection of State Violations inWeb Applications**," International Workshop on Recent Advances in Intrusion Detection,vol. 14, pp. 63–86.

[5]. Halfond, W.G. and Orso,W. (2005)"**AMNESIA: Analysis and Monitoring for Neutralizing SQL-Injection Attacks,**" International Conference on Automated Software Engineering ,vol. 3 , pp. 174–183.

[6]. Bisht,P.(2010) "**CANDID: Dynamic Candidate Evaluations for Automatic Prevention of SQL Injection Attacks**," ACM Transactions on Information and System Security , vol. 13, no. 2, pp. 14.

[7]. Liu,A., and Sehgal,.N. (2009) "**SQLProb: A Proxy-based Architecture towards Preventing SQL Injection Attacks**," ACM Symposium on Applied Computing ,vol. 15, pp. –2061.

[8]. Boyd,S.W. .and Keromytis,A.D.(2004) "**SQLrand: Preventing SQL Injection Attacks**," International Conference on Applied Cryptography and Network Security,vol. 6, pp. 292–302.

[9]. Baum,L.E.(1970) "**A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of MarK Chains,"**
The Annals of Mathematical Statistics,vol. 13, pp. 164–171 .

[10]. Rabiner,L.R.(1989) **"A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition"**, IEEE, vol. 77, no. 2, pp. 257–286.

[11]. Juang,B.H. and Rabiner,L.R.(1991) "**Hidden Markov Models for Speech Recognition**," Technometrics, vol. 33, no. 3, pp. 251–272.

[12]. Hu,J. and Portiugal,S.(1996) “**HMM based Online Recommender system**,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 18, no. 10, pp. 1039–1045 .

[13]. Lee,H.K. and Kim,J.H.(1999) “**An HMM-based Threshold Model Approach for Gesture Recognition**,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 21, no. 10,S., pp. 961–9739(1999.)

[14]. [14] Ivon,Y. and Carrio,S.(1999)“**Real-time Context-based Gesture Recognition using HMM and Automaton**,” International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems, vol. 15 ,pp. 127–134.

[15]. [Kotsiantis,S.B. and Dhage,L.(2007) “**Machine learning: a review of classification and combining techniques**”, international workshop on machine learning techniques,vol.15,pp.345-387.

[16]. Bellinger,C.(2017) “**A systematic review of data mining and machine learning for air pollution epidemiology**”, BMC Public Health ,vol. 32,pp.465-467.

[17]. Pérez ,S. and Ortiz,S.(2016)“**A Review of Classification Problems and Algorithms in Renewable Energy Applications**”, Academic Editor,vol.14,pp.342-356.

[18]. Dey,A. and Khan,L.(2016)”, International Journal of Computience and Information Technologies, Vol. 7 (3) , pp. 1174-1179.

[19]. Cihan,P.(2017) “**A Review of Machine Learning Applicats in Veterinary Field**”, Journal of research,vol.14 ,pp.471-473.

[20]. Bhatt,B.and Portugal ,S.(2014) “**A Review Paper on Machine Learning Based Recommendation System**”, International journal of research, Volume 2, pp. 2321-9939 .

[21]. Catesa,S.(2017) ”**A Machine Learning Approhach To Research Curation For Investment Process**”, Journal Of investment Management, Vol. 15, No. 1 pp. 39–49 .

[22]. Pahwa,N.S.(2017) “**Stock Prediction using Machine Learning a Review Paper**”, International Journal of Computer Applications,Vol. 163,pp.234-239.

[23]. Halfond,W.and Domingos,S.(2011) “**A Classification of SQL-injection Attacks and Countermeasures**”,International Symposium on Secure Software Engineering ,vol.13, pp. 12–23.

[24]. Sun,S.T and Dey,S(2016). **Attacks**”, Electrical and Computer Engineering, University of British Columbia ,Vol. 13,pp.435-437.

[25]. Singh,N(2011) “**Machine learning techniques**”international journal of machine learning,vol. 2,pp.345.

* Anjali Gupta. " New Technique For Preventing SQL Injection Attack Based On Normal Use Model." IOSR Journal of Computer Engineering (IOSR-JCE) 20.5 (2018): 73-83