International Journal of Electronics Communication and Computer Engineering
Volume 8, Issue 6, ISSN (Online): 2249–071X

Getting Down to Brass Tacks of Prevention Mechanism of Sql Injection in Php based Web Application

Joshi Padma N[1*], Dr. N. Ravishankar[2], Dr. M. B. Raju[3] and N. Ch. Ravi[4]
[1]Associate Professor, Sreyas Institute of Engg. & Technology, Hyderabad
[2]Professor, Dept of CSE, LKRBREC, Vijayawada
[3]Professor, Dept CSE KIET, Hyderabad
[4]Associate Professor, Dept of CSE, SIET, Hyderabad
email id: padmajoshi2015@gmail.com[1*]; ravish00@yahoo.com[2]; drrajucse@gmail.com[3] and ravi@saimail.com[4]
[*]Corresponding author

Date of publication (dd/mm/yyyy): 01/11/2017

Abstract **– SQL Injection Attacks are comparatively recent threat to privacy, integrity & accessibility of all online requests & their technical infrastructure, secretarial for practically fourth of internet vulnerabilities. This review paper is fully based on a master thesis, & numerous references in that, we presented this study on anticipation of SQL Injections. Overview of future approaches & accessible way outs, & recommendations on defensive coding techniques for PHP-powered web applications & other situations. Then, analysis of McClure's SQL DOM approach is for safety of SQL Injections in object-oriented applications. Solution for PHP-based online applications, SQLDOM4J, which is generously depends upon SQL DOM but tries to address a few of our condemnations toward it, & also evaluated its performance.**

Keywords **– Attack, PHP, SQL, Web Server.**

I. INTRODUCTION

SQL is Structured Query Language, which is a computer language for storing, manipulating and retrieving data stored in a relational database.

SQL is the standard language for Relational Database System. All the Relational Database Management Systems (RDMS) like My SQL, MS Access, Oracle, Sybase, Informix, Postgres and SQL Server use SQL as their standard database language.

Also, they are using different dialects, such as

1. MS SQL Server using T-SQL,
2. Oracle using PL/SQL,
3. MS Access version of SQL is called JET SQL etc.

Fig. 1. SQL Injection

By leveraging SQL Injection vulnerability, it is given right situations in which an attacker could use it to bypass web applications authentication & endorsement mechanisms & take back contents of entire database. SQL Injection is used to add, adjust & delete records in main database, disturbing data integrity. And extents like this, SQL Injection is also provides an attacker within illegal access to susceptible data including, clients information, personally identifiable information (PII), intellectual assets, trade secrets & other sensitive information.

SQL Injection Working

To run unbearable SQL queries close to database server. Attacker had been to find an input surrounded by web application which is also included inside of an SQL query.

In order to an SQL Injection attack to take place, susceptible websites needs to straightly include clients input within an SQL statement. Then attacker had to insert a payload which would also include part of SQL query & compile against database server.

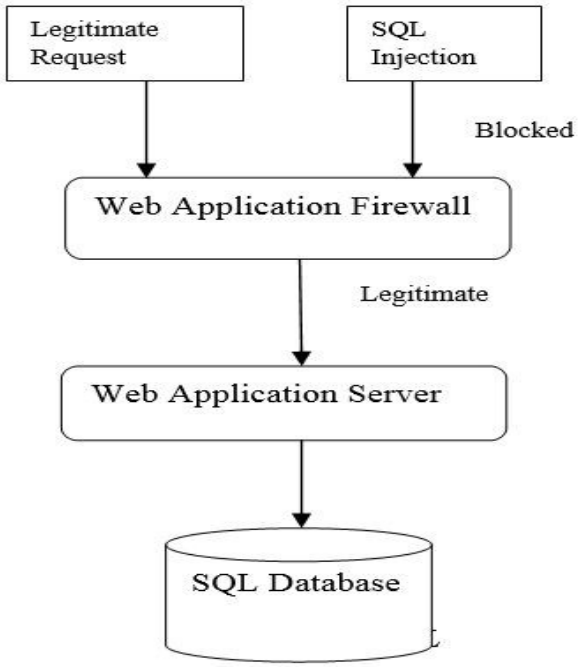

Fig. 2. SQL

II. SQL ATTACK

SQL is programming language proposed for administrating information stored in RDBMS, So that SQL could easily be used to access, adapt & delete information. And also, in exact cases, an RDBMS could also compile commands on OS from an SQL statement.

According to above observation, bearing in mind following, it's very simple to appreciate how profitable a successful SQL Injection attack could be for an attacker.

An attacker could also use SQL Injection to get around verification or even imitate specific users.

One of all SQL's main functions is to choose information based on a problem & output consequence of that query. SQL Injection vulnerability allows full revelation of information residing on a database server. Since web applications uses SQL to modify information within a database, & attacker could also use SQL Injection to modify information stored in a database. Modifying data concerns data integrity & also origins denial issues, for instance, problems such as voiding transactions, modifying balances & other records.

SQL is also used to delete records from a database. An attacker could also use SQL Injection vulnerability to delete information from database. Even if proper backup strategy is in working, deletion of information could affect whole applications accessibility until database is restored.

III. Attack Types

The attacks are different from each other, & they are classified in to two categories:

Data Exhilaration

Data exhila ration via SQL Injection is what had donated to some of biggest data breaches to date. And attackers also find a vulnerability that allows them to list all tables & dump all clients' accounts, emails & passwords.

Code Injection

We don't see code vaccination just commonly they often rely on a few opening vulnerability pre-tests that we block it automatically via our Website Firewall making that much harder to trace & attempt. A good instance of nasty code injection happened within old Liza moon type of attacks against Internet Information Services web sites.

IV. Related Work

Stephen W. Boyd SQL Rand: Preventing SQL Injection Attacks 2003 [1]

In this review paper we presented a practical safety mechanism against SQL injection attacks. Such attacks aim databases that are easily reached all way through a web front-end, & take advantage of errors in input substantiation logic of Web components for example CGI scripts. We as well relate thoughts of training - set randomization to SQL, making instance of language that are random to attacker. Queries inserted by attacker would be jammed & ended by database parser. In this review paper we also showed how to use this method within My SQL database using an agent proxy that decodes random SQL to its normal language. Our mechanism inflicts negligible presentation overhead to problems processing & could also be simply retrofitted to accessible systems. We offered SQL rand, a structure for preventing SQL injection attacks next to web servers. Main instinct of SQL is that by using a randomized SQL query language, exact to a exacting CGI application, it is likely to notice & terminate queries that also contain injected code.

Sonam Panda 2013 Protection of Application against SQL Injection Attacks [2]

In this review paper, different types of SQL injection attacks as well as predefined avoidance methods are conversed. Then hybrid encryption technique is used which consists of AES encryption & Rabin's cryptosystem. Main reason behind use of two layer of encryption is that it would be secured. SQL query is created & encrypted by Rabin‟s cryptosystem because even if hackers hack information & crack AES encryption part, it would still be trickier for them to be familiar about encrypted query. Between Rabin & RSA, it is more complex to say which cryptosystem is better.

Sampada Gadgil 2015 SQL Injection Attacks & Prevention Techniques [3]

In this review paper, we presented a survey of current techniques of SQL injection as well as a result methodology for avoiding attacks. To carry out this evaluation, we firstly identified different types of SQL Injection attacks. And we also studied various mechanisms from end to end which SQL Injection Attacks could also be initiated into an application & identified methods that are able to switch mechanisms. Many of methods have troubles in handling attacks that take benefit of poorly coded stored methods & SQL problems cannot handle attacks. This dissimilarity could be easily explained by fact that prevention-focused techniques always try to slot in defensive coding best practices into their attack avoidance mechanisms.

Parveen Sadotra 2017 SQL Injection Impact on Web Server & their Risk Mitigation Policy Implementation Techniques: [4]

In SQL injection attacks, hackers could take benefit of poorly coded Web application software to bring in nasty code into organization's systems & network. Susceptibility exists when a Web application do not correctly filter or validate entered information by a user on a Web page. Big Web applications have thousands of places where clients could input information, each of which could offer a SQL injection attack opportunity. Attackers could edit confidential & dangerous information of association within these attacks resulting loss of market price of organization. This paper offered an effective analysis of SQL Injection attack, detection & avoidance techniques. We also analyzed some accessible techniques to notice attack & alleviate risk connected within these attacks.

Pankaj Sharma 2012 Integrated Approach to Prevent SQL Injection Attack & Reflected Cross Site Scripting Attack [5]

The Internet & web applications are playing very important role in our today's modern day life. Several activities of our daily life like browsing, online shopping and booking of travel tickets are becoming easier by use of web applications. As volumes of web applications are increasing security of web applications becomes a major concern.

Harti Nagpa SECSIX: Security Engine for CSRF SQL Injection & XSS Attacks [6]

With increase in human-web interaction, Vulnerabilities had surfaced various networks. Within rapidly growing technology, ease of accessibility through web applications had revolutionized traditional view of an office or a company completely. Web application carries sensitive data & they are accessible 24 9 7.

Web site hacking continue to gain popularity as hackers are exploiting vulnerabilities across all geographies & across various types of web technologies.

Amir Mohammad Sadeghian 2014 SQL Injection Vulnerability General Patch using Header Sanitization [7]

SQL injection is one of well-known web application vulnerabilities. SQL injection is a type of attack which attacker attempts to insert malicious SQL query through none sanitized variables into web application. Consequently web application would concatenate variable within legitimate query & would send it to database for execution. In result of a successful SQL injection attack, attacker could read from database or modify entities of database (Insert, Delete, Update). Currently different types of defense systems are available to defeat this vulnerability.

Swapnil Kharche Preventing SQL Injection Attack using Pattern Matching Algorithm [8]

SQL injection attacks, a class of injection flaw in which specially crafted input strings leads to illegal queries to databases, are one of topmost threats to web applications. A Number of research prototypes & commercial products that maintain queries structure in web applications have been developed. But these techniques either fail to address full scope of problem or have limitations.

Nabeel Salih Ali Protection Web Applications using Real-Time Technique to Detect Structured Query Language Injection Attacks [9]

At present, Web applications have been used for most of our life activities increasingly, & they affected by Structured Query Language Injection Attacks (SQLIAs). This attack is a method that attackers employ to impose database in most of web applications, by manipulate SQL queries, which sent to Relational Database Management System (RDBMS). Hence, change behavior of applications. In this paper, developing web application SQLI Protector (WASP) tool in real-time web application to detect SQL injection attacks in stored procedures. Then, evaluated & analyze developed tool respect to efficiency & effectiveness in practices.

Rathod Mahesh Pandering 2015 a Mapping-based Podel for Preventing Cross Site [10]

Scripting & SQL Injection Attacks on Web Application & its impact analysis web applications provide vast category of functionalities & usefulness. As more & more sensitive data is available over web, crackers are getting attracted in such data revealing which could root immense harm. SQL injection is one of such type of attack. This attack could be used to infiltrate back-end of any web application that might lead to modification of database or disclosing significant information. Attacker could obfuscate input given to web application using Cross site scripting attack that might direct to distortion in web page view. Three tier web applications could be categorized into static & dynamic web application for detecting & preventing these types of attacks. Mapping model in which requests are mapped on generated queries could be used productively to detect such kind of attacks & prevention logic could be applied for attack removal. Impact measurement of container based approach on web server is measured using auto bench tool, parameters used are network throughput & response time.

Lwin Khin Shar & Hee Beng Kuan Tan Mining SQL Injection & Cross Site Scripting Vulnerabilities using Hybrid program analysis [11]

We proposed a set of static attributes that characterize input validation & input sanitization code patterns. We showed that some of proposed static attributes are significant predictors of web application vulnerabilities related to SQL injection & cross site scripting. Static attributes have advantage of reflecting general properties of a program. Yet, dynamic attributes collected from execution traces might reflect more specific code characteristics that are complementary to static attributes. Hence, to improve our initial work, in this paper, we propose use of dynamic attributes to complement static attributes in prediction of vulnerabilities. Furthermore, since existing work relies on supervised learning, it is dependent on availability of training data labeled within known vulnerabilities.

V. TOOLS AND TECHNOLOGY

Hardware & Software Requirement

Hardware Requirement

CPU (More than 1 GHz)
RAM (2 GB) Recommended
Hard disk (10 GB free space)
Monitor (High resolution)
Keyboard
Mouse

Software Requirement

Windows 7
SQL
PHP
SQL

Structured Query Language (SQL), pronounced "sequel", is set of commands that all programs & users must use to access information within Oracle database.

The language, Structured English Query Language (SEQUEL) was urbanized by IBM Corporation, Inc. to use Codd's model. SEQUEL later became SQL.

Benefits of SQL

This part describes many of causes for SQL's widespread acceptance by relational database vendors as well as end users. Strengths of SQL benefit all ranges of users including application programmers, database administrators, management & end users.

Non-Procedural Language

SQL is a non-procedural language because it: Processes sets of records rather than just one at a time; Provides automatic navigation to data.

Unified Language

SQL provides commands for a variety of tasks including: Querying data; Inserting, updating & deleting rows in a table;

Databases

Because all major relational database management systems support SQL, you could move all skills you have added within SQL from one database to another. In addition, since all programs written in SQL are transportable, they could often be moved from one database to an alternate within very little modification.

PHP

- PHP stands for PHP: Hypertext Preprocessor.
- PHP is a server-side scripting language, like ASP.
- PHP scripts are executed on server.
- PHP supports many databases (MySQL, Informix, Oracle, Sybase, Solid, Postgre SQL, Generic ODBC, etc.).
- PHP is an open source software.
- PHP is free to download & use.

PHP Syntax

PHP code is executed on server, & plain HTML result is sent to browser.

A. Basic PHP Syntax

A PHP scripting block always starts within **<? php** & ends within **?>**. A PHP scripting block could be placed anywhere in document.

On servers within shorthand support enabled you could start a scripting block within <? & end within?>.

For maximum compatibility, we recommend that you use standard form (<? php) rather than shorthand form.

A PHP file normally contains HTML tags, just like an HTML file, & some PHP scripting code.

Below, we have an example of a simple PHP script which sends text "Hello World" to browser:

Each code line in PHP must end within a semicolon. Semicolon is a separator & is used to distinguish one set of instructions from another.

There are two basic statements to output text within PHP: echo & print. In example above we have used echo statement to output text "Hello World".

Note: File must have a .php extension. If file had a .html extension, PHP code would not be executed.

Select Query in SQL Could be used as SQL Injection

The basic syntax of the SELECT statement is as follows

Here, column 1, column 2... are the fields of a table whose values you want to fetch. If you want to fetch all the fields available in the field, then you can use the following syntax.

SELECT column 1, column 2, column N FROM table_name;

The following code is an example, which would fetch the ID, Name and Salary fields of the customers available in CUSTOMERS table.

SQL > SELECT ID, NAME, SALARY FROM CUSTOMERS;

If you want to fetch all the fields of the CUSTOMERS table, then you should use the following query.

SQL > SELECT * FROM CUSTOMERS;

VI. SQL - WILDCARD OPERATORS

We have already discussed about the SQL LIKE operator, which is used to compare a value to similar values using the wildcard operators.

SELECT FROM table_name WHERE column LIKE 'XXXX%'
or
SELECT FROM table_name WHERE column LIKE '%XXXX%'
or
SELECT FROM table_name WHERE column LIKE 'XXXX_'
or
SELECT FROM table_name WHERE column LIKE '_XXXX'
or
SELECT FROM table_name WHERE column LIKE '_XXXX_'

N number of conditions can be combined using the AND or the OR operators. Here, XXXX could be any numeric or string value.

During attack operation user could insert %% instead of actual password. The sql query processor would allow the access of data as %% means any password. Thus the restriction of wild character like % or _ at the interface level could check the un authentic access of web based application.

VII. SCOPE FOR RESEARCH

The security form sql injection has major concern. The scope of such research is significant in case of web development. The user could make attack on web application using sql injection. Thus the study of threats and their prevention is must. The conservative mitigations techniques were evaluated based on parameters like answer time & decrease in number of injections possible. Job converged to a conclusion that by usage of conservative mitigation techniques, we could reduce number of injections considerably. Techniques which focus on prevention of attack should incorporate defensive coding. Dynamically built SQL statements should be created properly within conventional mitigation techniques mentioned rather than using simple concatenation technique. This would make sure that only legitimate queries are passed to database server.

Using validation along within other two techniques proved to be a reasonable solution for this attack. Study of effectiveness of conventional fixes showed their inability for ultimate eradication & this opens scope for further research work.

REFERENCES

[1] Stephen W. Boyd SQL rand: Preventing SQL Injection Attacks 2003.
[2] Sonam Panda 2013 Protection of Web Application against Sql Injection Attacks.
[3] Sampada Gadgil 2015 SQL injection attacks & prevention techniques.
[4] Parveen Sadotra 2017 SQL Injection Impact on Web Server & their Risk Mitigation Policy Implementation Techniques:
[5] Pankaj Sharma 2012 Integrated approach to prevent SQL injection attack & reflected cross site scripting attack.
[6] Bharti Nagpa SECSIX: security engine for CSRF, SQL injection & XSS attacks.
[7] Amir mohammad Sadeghian 2014 SQL Injection Vulnerability General Patch using Header Sanitization.
[8] Swapnil Kharche Preventing sql injection attack using pattern matching algorithm.
[9] Nabeel Salih Ali Protection Web Applications using Real-Time Technique to Detect Structured Query Language Injection Attacks.
[10] Rathod Mahesh Pandurang 2015 A Mapping-based Podel for Preventing Cross Sit.e
[11] Lwin Khin Shar & Hee Beng Kuan Tan Mining SQL Injection & Cross Site Scripting Vulnerabilities using Hybrid Program Analysis.
[12] Stephen W. Boyd SQLrand: Preventing SQL Injection Attacks 2003.
[13] Sonam Panda Protection of Web Application against Sql Injection Attacks International Journal of Modern Engineering Research Vol.3, Issue.1, Jan-Feb. 2013 pp-166-168.
[14] Sampada Gadgil SQL injection attacks & prevention techniques International Journal on Recent & Innovation Trends in Computing & Communication ISSN 2321 – 8169 Volume: 1 Issue: 4
[15] Parveen Sadotra SQL Injection Impact on Web Server & Their Risk Mitigation Policy Implementation Techniques: An Ultimate solution to Prevent Computer Network from Illegal Intrusion Volume 8, No. 3, March – April 2017 International Journal of Advanced Research in Computer Science.
[16] Kindy, D.A., and Path an, and A.K.: A Detailed survey on various aspects of SQL injection in web applications: vulnerabilities, innovative attacks & remedies. In: International Journal of Communication Networks & Information Security, vol. 5, no. 2, pp. 80–92 August 2013.
[17] Bono, S.C., Domangue, E.: SQL Injection: A Case Study, Whitepaper Oct 2012.
[18] Shar, L.K., Beng, H., Tan, K.: Defeating SQL Injection. IEEE Comput. Soc. 46(3), 69–77 (2013) (IEEE).
[19] Ahmad, K., Shekhar, J., Yadav, K.P.: Classification of SQL injection attacks. In:
[20] VSRD-TNTJ, vol. I, no. (4), pp. 235–242(2010).
[21] Bisht, P., Madhusudan, P., Venkatakrishnan, and V.N.: CANDID: Dynamic candidate evaluations for automatic prevention of SQL injection attacks. In: ACM Transactions on Information & System Security, vol. 13, no. 2, p. 139. ACM (2010).
[22] Jane, P.Y., Chaudhari, M.S.: SQLIA: Detection & prevention techniques: a survey. IOSR J. Comput. Eng. 2, 56–60. IOSR J. (2013).
[23] Halfond, W.G.J., Orso, A.: AMNESIA: analysis & monitoring for neutralizing SQL injection attacks. In: Proceedings of 20th IEEE/ACM International Conference on Automated Software Engineering, pp. 174–183. ACM, New York (2005).
[24] Clarke, J.: SQL Injection Attacks & Defense. Elsevier Inc (2009).