EasyChair Preprint

№ 4175

SQL Injection-Biggest Vulnerability of the Era

Harshavardhan Gaddam and M. Maheshwari

September 13, 2020

SQL Injection-Biggest vulnerability of the era

Gaddam Harshavardhan

School of computing

Sathyabama institute of science and technology

Chennai, India.

harshareddy794@gmail.com

Dr.M.Maheshwari

School of computing

Sathyabama institute of science and technology

Chennai, India.

Mahisara10@gmail.com

***Abstract*— the day we started using Database and SQL (Structured Query Language). SQL injection became one of the biggest vulnerabilities. Since 1998 which first SQL injection is discovered by Phrack magazine it becomes critical and affects a lot of web sites, a lot of counter measures are taken but it still exists. It is also ranked No.1 in vulnerabilities by OWASP. Now let us explore briefly the details of SQL injection, methods of SQL injection and ways to protect a web site from SQL injection.**

Keywords*— *SQL Injection, Database, query and Vulnerability

I. Introduction

Two decades from its discovery in 1998, SQL Injection still exists in many Web applications. It is one of the critical vulnerability so that it discloses all the data in data base. According to a security survey around 4 Web applications are hit by SQL injection every month around the world starting from small blogs to Big E-commerce and commercial web sites. It is ranked at top 10 vulnerabilities in 2007 and in 2010 and ranked 1st position in 2013 among all the vulnerabilities by OWASP (The Open Web Application Security Project).

This vulnerability can occur due to improper user input sanitation to unprotected web application with Firewall. Due this vulnerability hacker can access data, modify data, make it unavailable, destroy data and can change privilege of the user to admin. Now let us see what is SQL injection.

SQL Injection:

SQL injection is the process of injecting miscellaneous code with a SQL query so that it is executed by SQL server and give output based on the query.

Fig 1:

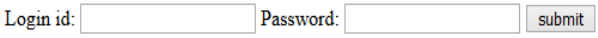

Fig 1: Sample HTML login form

HTML code for this page:

```
<form action="ABCD.com"
method="POST">
Login id:
<input type="text" name="login_id"
id="login_id">
Password:
<input type="password" name="pass"
id="pass">
<button
type="submit">submit</button>
</form>
```

This HTML code is used to display login form on the screen and used to get user input.

PHP code for the page:

<? php

$user=$_POST['login_id'];

$pass=$_POST['pass'];

$result=$Query: SELECT * FROM users WHERE username='$user' AND password='$pass'

if ($result is not blank)

{Echo "Login success";}

else

{Echo "Invalid login credentials";}

?>

PHP code is used to execute all our process

SQL query used for checking login credentials from database is

SELECT * FROM users WHERE username='$user' AND password='$pass'

Database is a storage in which data is stored in the form of tables which has rows and columns. Data we provided is stored in rows and columns. Database is used to access, maintain and update data easily.

Where SQL (Structure query language) is a domain specific language used to search, update, and delete data from database.

Table 1:

Name	username	password
Person1	ABCD	12345
person2	user.123	Pass
Person 3	User	pass@123

Table 1: Sample database

If user gives username as ‘user’ and password as ‘pass@123’ SQL query verifies details by checking in database and if they are correct login is done. If they are wrong it will display invalid login credentials. If the web application having SQL injection vulnerability is found hacker may try to bypass firewall by adding miscellaneous code into SQL query. He uses all his knowledge to trick the server and access critical data or bypass the login process. Let us see how SQL injection vulnerability helps hacker to bypass login authentication.

Fig 2:

Fig 2: Data transfer process

II. Injection process

Hacker enters login id as user and password as ABC’ OR ‘1’=’1 by using this inputs SQL query is modified as

SELECT * FROM users WHERE username=’user’ AND password= ‘ABC’ OR ‘1’=’1’.

Then this query sent to SQL server where this server tries to execute this query. It will go users table and search for username ‘user’ and it will get username as user and try to check password against user input. This password tricks the server in SQL query hacker given password and added ‘1’=’1’ to it.1 is always equal to 1. So the blank condition is false for PHP code. Here login is done and shows dashboard of user.

There are so many types of SQL injections but mainly it is classified into two types. They are

1. Error based
2. Blind based

Error based:

Error based is the process of adding of miscellaneous code into SQL query and based on the error produced by SQL server the input data is modified to get critical data that is present in a SQL server or changing the privilege of a user to admin.

Ex:

When we add a single quote at the end of form it will produce an error stating that error at SQL line ‘X’. Also when we write CONVERT (int,’141’) which is true and it can be converted. server execute the query and produce the output but when we write CONVERT (int,’abc’) it will produce error saying that ABC cannot be converted to integer. By using same technique hackers try to gain all the details about server and get critical information.

To get name of the database we are interacting with we write CONVERT (int,’db_name ()’) server will throw an error showing name of database cannot be converted into integer. By using the same technique hackers will get name of the database, tables, columns and data in columns.

Blind based:

Blind based method is also same as error based but it will not produce output directly, server will produce output in a different manner. This output can be closely monitored and query can be changed accordingly to get output. Blind based also classified into two types

1. Boolean based
2. Time based

Boolean based:

Boolean based is simply true of false. Based on true or false we can fetch the data in a SQL server it can be simply done.

Ex:

If we write SELECT name FROM students WHERE id=121 AND 1=1--+

Where 1 is always equal to 1 it will produce output as true for this statement.

If we write WHERE id=121 AND 1=0--+

It will produce output as false. We can use this method to get lot of data. Attacker will write code to get password of student who is having id 121 as follows. If we write

SELECT name FROM students WHERE id=121 AND (GET_FIRST_CHARACTER_OF (password)) = ’a’--+

If the first character is ‘a’ then it will produce output true else it will produce output as false. Then at place of ‘a’ we replace b, c, d and etc to get output accordingly. By using same technique we can get full password.

Time based:

It is similar to Boolean based SQL injection but Boolean based method shows output as true or false. But in this method it will show true or false by its time of response.

Ex:

We write query as

SELECT name FROM students WHERE id=121 AND (IF_FIRST_CHARACTER_ OF_PASSWORD=’a’_SLEEP_FOR_10_SECONDS)--+

If first character of password is ‘a’ it will show output after 10 seconds that means first character is ‘a’. If first character is not ‘a’ it will show output immediately. By using same method we can get full password of student who is having id 121.

These are all the native types of SQL injection attack. There are also some DBMS specific attacks that they can be applied only on that specific DBMS system. Some types of DBMS systems are MySQL, PostgreSQL, MsSQL, MongoDB, and MariaDB.

Attackers also try to increase the impact of SQL injection by combining SQL injections with some other attacks. They are:

SQL injection+ cross site scripting (XSS)

SQL injection+ DOS (Denial of service)

SQL injection+ XML injection

SQL injection+ DNS hijacking

By combining different attacks to increase the output and to bypass web application filters from user side and server side.

Ex:

SQL injection+ XSS will help to bypass the user side filters and DOS attack will increase number of requests sent to server which will help in bypassing server side filters. Impact produced by SQL injection can be increased very huge by combining attacks with SQL injection.

III. Detection of SQL injection:

SQL injection can be detected in so many ways. Sometimes attacker may try not to get data from SQL server but try to change his privilege from user to admin using SQL injection these types of attacks can be detected by getting the number of administrators of a web application. Some hackers may not try to change privilege instead they dump all data from SQL server. These types of attacks can be detected by knowing that who is accessing the restricted files, tables and data in them. Some hackers can be detected by using log produced by the client. Log of a client will have all the queries used by a client to dump the data.

Automated SQL injection attack can be easily detected because it sends a lot of requests to server to dump lot of tables which will automatically produce lot of load at server. Automated SQL injection also produces some empty tables and produces a bulk log files. By using this files and techniques we can easily detect SQL injection attack on a web application.

IV. Prevention:

Mainly there are 3 methods to prevent SQL injection. They are:

1. Implementation of Block chain
2. AI based web application firewall
3. Standard coding techniques

Implementation of Block chain:

Block chain technology is more widely used to securely store data, prevent from many attacks and prevents manipulation of data. Block chain technology is more secure and prevent from SQL injection.

In block chain blocks are connected by hash codes and form a chain of blocks. A data is stored in multiple blocks which makes manipulation of data impossible. By using all other blocks the data that is manipulated in a single block will get auto corrected. All the data in a block is encrypted by lot of algorithms and techniques which makes reading and changing of data impossible for an attacker.

All the blocks are connected by previous and next block hash codes only a node having a valid IP address and hash code can access data in a block, manipulate data and delete data in a block. A data can be changed only if half of the blocks which have that data accept the change in data. All these properties of block chain make attackers impossible to attack and inject SQL injection into code.

Fig 3:

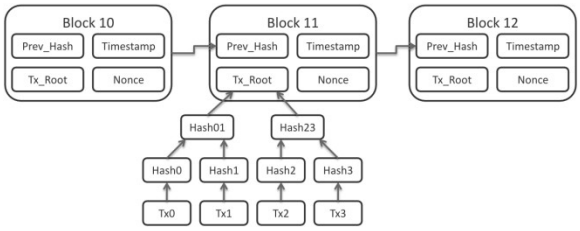

Fig 3: Block diagram for working of a Block chain

Artificial Intelligence based firewall:

A strong firewall powered by machine learning should be implemented. This firewall must be used for all web applications in order to prevent it from lot of attacks.

Train artificial intelligence on the web applications attacks so that it detects SQL injection and lot of attacks and prevents them from happening. Prevent

queries that lead to disclose confidential and critical data and leads to SQL injection vulnerability.

Firewall is used to detect change in input of a client, encryption key to be sent and receive and decryption of data at user interface and execution of process like SQL query at web server and encryption of saved data. Based on the production of log files of a client and queries used to dump data from different tables and columns and queries leads to SQL injection should be filtered and blocked.

All the queries that are sent to server should be audited and queries that disclose critical data should be filtered. Machine learning system can be very useful when attacker tries to test web application using automated tools. Because automated tools produce a large log files. By using log files we can prevent.

Working of Machine Learning:

- Checking the IP of client for valid IP.
- Checking authentication.
- Checking valid input.
- Checking authorization.
- Access page.

Else:

Restrict Access, save IP and inform admin.

Artificial Intelligence based system should be trained by previous data so that it detects latest attacks. Analyzing latest attacks for implementation of artificial intelligence.

Fig 4:

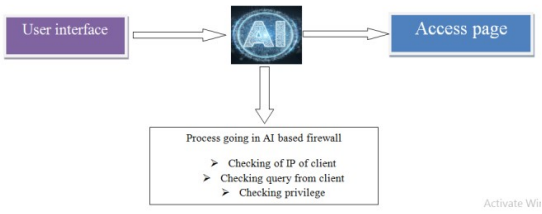

Fig 4: Working of AI based firewall

Standard Coding Techniques:

There are lot of strict rules to be followed in order to prevent SQL injection for a web application minimizing privilege of SQL server from root access check authorization and authentication of access of a user. Audit SQL query which contains some terms like SELECT, WHERE, FROM should be eliminated from query and dropping particular request.

Protection of web application with firewall is must in order to increase security. Second order execution of SQL query is secure because rather executing it directly it is saved and encoded and then it is executed. Maintaining coding standards like rather using traditional single quote using double quote is better. Encrypt data in request sent or received request and data stored in a server. Test all request for type, formal, length and range of a request.

Do not built SQL query based on user input. Should not accept Binary, Boolean, Escape sequences must be eliminated from query checking IP of a client. Before taking it available to all make a full test of SQL injection on server and see how it responds and how it produce output based on input make proper changes to make secure and produce proper output. Create a file for all SQL queries and check request against requests in file. If present execute else drop query.

Use proxy filters and detect change in networks. Do not use stacked queries because some automated tools use stacked queries.

V. Conclusion:

SQL injection is a very critical vulnerability. Through many hackers use SQL injection to steal data, get admin access get products for free which cause a lot of problem gaining critical data like passwords and credit card numbers cause a lot of problem to people who use it and also to owners of association.

But by implementation of some technologies like block chain, artificial intelligence based firewall and some strict rules to be followed to access data very securely and storing of data without any SQL injection vulnerability. Block chain technology is the

best technology to be implemented to access and store data. Maintaining some coding techniques and Using of firewall is best for a web application. All these techniques will give a good security to web application.

VI. Acknowledgment

I thank everyone who motivated me to write this paper. Also I thank M.Maheshwari madam and R.Sethuraman sir for being my mentors to bring this paper to the publication. I also thank my college for giving me this opportunity. Thank you.

VII. Reference

[1] http://joiv.org/index.php/joiv/article/view/144

[2] https://en.m.wikipedia.org/wiki/SQL_injection

[3] https://www.owasp.org/index.php/SQL_Injection