International Journal of Advances in Engineering and Management (IJAEM)
Volume 4, Issue 1 Jan 2022, pp: 579-583 www.ijaem.net ISSN: 2395-5252

SQL Injection Attack Prevention System

Saroj Hiranwal[1], Ajay Sharma[2], Ayush Saini[3]
Head of CSE Department, Rajasthan Institute of Engineering & Technology, Jaipur, India
Computer Science & Engineering, Rajasthan Institute of Engineering & Technology, Jaipur, India
Computer Science & Engineering, Rajasthan Institute of Engineering & Technology, Jaipur, India

Submitted: 02-01-2022 Revised: 09-01-2022 Accepted: 12-01-2022

ABSTRACT:-
Existing Web system vulnerabilities jeopardize the regular operation of information systems. SQL injection is the most prevalent Web system vulnerability. In the article, you'll get information on how to secure Web applications from SQL injection attacks. To increase the security of Web software, adefence mechanism that protects Web resources against SQL injection has been developed. PHP, JavaScript, and the formal language theory known as regular expression were utilized to create this software.Hackers can acquire access to confidential and private information by exploiting flaws in most web apps. Structured query injection is one of the most prevalent and widely used information theft strategies, posing a substantial threat to web services. Where hackers profit from flaws in system design or existing gaps by failing to filter user input for special characters and symbols contained within structural query sentences, or failing to check the quality of the information, whether text or numerical, resulting in unpredictability in the outcome of its implementation.As a result, a software tool that can defend Web software from SQL injection vulnerabilities has been developed. The developed software solution allows the user to use SQL to secure his own Web application from an attack.

KEYWORDS:- SQL Injection , Database Security, WEB application, SQL Injection Attack prevention methods etc.

I. INTRODUCTION

In today's technology-driven world, website applications are an integral component of daily living. Websites are used for a variety of activities, such as online shopping, banking, and chatting with friends. On the backend, many websites use databases to store user data [1]. Malicious hackers frequently target such files because they store sensitive information such as passwords, credit card numbers, and social security numbers [2-4].

Based on an analysis of various hacking events, any time operating system safety boosts and safety protection software in addition to hardware solutions come to be more widespread,systemattacks directly brought on by operatingprogram vulnerabilities decrease 12 months after year, asthe use rate associated with WEB application program vulnerabilities increases. Due to its easy syntax plus high development overall performance, PHP has turn out to be the language associated with choice for building all types associated with portal websites plus Web application applications. LAMP architecture (Linux + Apache & MySQL + PHP) is a PHP language development atmosphere with fast velocity, high compatibility, totally free open source, plus other benefits [5]. In line with the present traffic levels, regarding 70% from the LIGHT architecture depends on the particular present-day network problems [6]. Given that the PHP earlier design process will be over-efficient and clear, some necessary protection specifications in the particular programming language are usually not strictly limited [7]. Including delicate data will be needed. It will be counterproductive because the particular programmer lacks their own security knowledge whilst the machine features are jeopardized [8].

The operational protection of some parts of the system is unrestricted [9]. Because there are so many data points that are sensitive, the system can pose major security issues [10]. SQL injection vulnerabilities are particularly frequent among them. The SQL injection was 10 years old and had been in people's minds for the first time since 1999. While a thorough preventative approach is already in place, its potential should not be overlooked [11]. SQL injection is one of the most hazardous flaws [12]. SQL injection was recognized as the top challenge to Web application systems three times [13,14] in the OWASP Top 10 for 2010, OWASP Top 10 for 2013, and OWASP Top 10 for 2017.

The likelihood of SQL injection and the danger it poses are determined by the database type and its conditions. It usually allows the attacker to

International Journal of Advances in Engineering and Management (IJAEM)
Volume 4, Issue 1 Jan 2022, pp: 579-583 www.ijaem.net ISSN: 2395-5252

run any database query (e.g., read the contents of any table, delete, alter, or add data) as well as read and/or write local files and execute arbitrary instructions on the server side, potentially leading to more destructive assaults inside a network or DMZ. Due to the processing of input used in SQL queries without verification or/and filtering, a SQL injection attack is conceivable.

The operational protection of some parts of the system is unrestricted [9]. Because there are so many data points that are sensitive, the system can pose major security issues [10].

Individual websites, as well as the entire database architecture and network system that hosts relevant applications, are vulnerable to SQL injection flaws [15].

II. SQL INJECTIONS ATTACKS TYPES

In today's world, many Web systems employ databases to store data such as user preferences, personal data, sensitive financial information, and so on in a variety of industries ranging from banking to e-government to social networking. Developers may make Web products exciting and valuable for clients by incorporating a number of technologies (e.g., e-commerce, online banking) [12-15].

SQL injection is a sort of code injection attack in which a portion of the user's input is interpreted as SQL code [9]. This is a way for inserting SQL queries or orders using web pages as input. It occurs when the user's information isn't double-checked and isn't expressly utilized in the SQL questionnaire. By exploiting these flaws, an attacker can get direct access to the database. According to Sharma, there are two main SQL injection tactics (2005). (There are two ways to get access: through the login page and through the URL.)

1.1 Properly filtered escape character

This type of SQL injection happens when client input isn't sifted for get away from characters and is then passed into a SQL explanation.This is a type of injection attack in which the user is not tied to prevent them from writing anything they don't want to write, and the information is instead sent to a SQL query. If the end-user is responsible for the SQL statements, this can only cause problems for the user. A code piece that highlights this vulnerability is a good example of the problem:

Statement: (“SELECT * FROM users WHERE name=“” + 17username + “;”)

1.2 Failure to handle sort

This assault pattern comes in a variety of forms. This attack can be carried out if an attacker uses basic types in a nonconfirmed field. Until you submit it to the database, it is not complete (whether it is numeric or not). Consider the following situation:

Statement: (“SELECT * FROM data WHERE id = + a_variable+”;)

1.3 Database server vulnerabilities

Frequently, database system programs have vulnerabilities, such as the vulnerability in the MYSQL server's actual _escape_ string () function. Using erroneous unified character encoding, an attacker will be able to conduct a sophisticated SQL injection attack

2.4 Blind SQL injection attack

This is referred to as a "blind SQL injection." An application is vulnerable to attack, but the attacker is unaware of the consequences. When a SQL injection is used, it becomes a "blind" SQL injection. The data might be enlarged without being displayed, but utilizing legitimate statements and logical processes entered into the system, numerous created data outcomes will be produced.

When a unique string is checked, new bytes must be added for each byte added to the database. When you know where the fault is, you can utilize Absinthe to look for the target information that corresponds to it.

2.5 Out-of-band SQL injection attack

Out-of-band methods, offer an aggressor an option in contrast to inferential time sensitive procedures, particularly assuming the waiter reactions are not entirely steady (making an inferential time sensitive assault problematic).

The aggressor can complete this type of assault when certain elements are empowered on the information base server utilized by the web application. This type of assault is essentially utilized as an option in contrast to the in-band and inferential SQLinjection methods.

Out-of-band SQLinjection is performed when the aggressor can't utilize a similar channel to send off the assault and assemble data, or when a waiter is excessively sluggish or shaky for these activities to be performed. These strategies rely on the limit of the server to make DNS or HTTP solicitations to move information to an aggressor.

III. SQL INJECTION PREVENTION SYSTEM

The main sure method for forestalling SQL Injection assaults is input approval and parametrized inquiries including arranged articulations. The application code ought to never utilize the info straightforwardly. The engineer should clean totally enter, not just web structure data sources, for example, login structures. They

International Journal of Advances in Engineering and Management (IJAEM)
Volume 4, Issue 1 Jan 2022, pp: 579-583 www.ijaem.net ISSN: 2395-5252

should eliminate potential malignant code components like single statements. It is likewise really smart to switch off the perceivability of information base blunders on your creation destinations. Data set mistakes can be utilized with SQL Injection to acquire data about your information base.

We have provided protection at performs three functions to safeguard the system against SQL injection attacks. When a hacker tries to perform SQL injection on a vulnerable system, Converting special characters to HTML entities is the first way. Regular expressions and exceptions are two alternative options.

Assuming you find a SQL Injection weakness, for instance utilizing an Acunetixscan, you might not be able to fix it right away. For instance, the weakness might be in open source code. In such cases, you can utilize a web application firewall to disinfect your feedback for a brief time.

1.4 Escape of output Because server-side web applications frequently interact with webpages, URLs, and databases, there are several methods available to manage this information. The majority of the data is handled as a string, although different validation using filters is required in various contexts (database or server). A space in a site URL, for example, is defined as 20%, but a single quotation (') is specified as '. PHP includes a number of conversion functions by default. As a result, the Output escaping approach is frequently used.

By deleting HTML elements and extracting meta tags, PHP methods may turn special characters in a string into their entities. The htmlspecialchars() function in PHP transforms preset characters into entities (omitting the space character). To create appropriate HTML, this method translates the fewest amount of entities feasible.

- For example:

```
$string = htmlspecialchars("It is known that 30 > 10, but < 50");
echo $string;
// $output is 'It is known that 30 &amp;gt; 10, but &amp;lt; 50'.
```

As a result, the server side might apply a filter that uses the htmlspecialchars() method to guard against basic SQL injection that uses single or double quotation characters. Furthermore, the strip tags() method, which is a security option forHTML forms, may be used. All HTML tags in the provided data are ignored by this function.

- For example:

```
$input = '<p>&amp;quot;Hello&amp;quot;</p>';
$out = strip_tags($input);
// $out is '&amp;quot;Hello&amp;quot'.
```

By inspecting type, length, format, and range data, it is important to check the type or obligatory data type assignment, as well as all submitted data. It's critical to think about architecture and script execution while creating safeguards against malicious input. For example, if a variable has been put into the request and is needed to keep numeric data, it must employ the forced reduction derived type to a numeric. The following is the code:

```
$id = settype($_GET["id"],"integer");
$email = settype($_GET["id"],"varchar");
```

1.5 Regular expressions can be used to check the structure of supplied data. When entering data into a website, users may not always follow the developer's instructions. Users make errors all the time. However, there are attackers that aim to take advantage of popular Web platforms. It is critical for both sets of users to ensure that the data they transmit is correct. A regular expression that is accessible might be utilised for this purpose.

If the developer understands that the input data has a structure, such as a customer's e-mail address, phone number, or date, it makes sense to use regular expressions to "drive the meanings." As a result, the correctness of the inputted data was confirmed, and any undesired characters and changes to the sent input were cut off at the same time. In the case of basic queries, however, regular expressions can be employed. For example, the input parameter may be a four-digit integer or all lowercase letters with a length of 5 to 10 characters. Here's how to use it:

```
<?php
class data_validation {
   public function validate_string($data) {
      // Remove excess whitespace
      $data = trim($data);
      if ( preg_match("/^[0-9A-Za-z \.\-\'\"]+$/", $data) ) {
         return true;
      } else {
         //return 'Not a valid string';
         return false;}
   }}} ?>
```

1.6 Exceptions are used. To avoid services being stopped when a Web application stated fault occurs, the regular flow of code execution must be changed. Exception management is the term for such a circumstance. Sometimes software fails to perform as expected, resulting

International Journal of Advances in Engineering and Management (IJAEM)
Volume 4, Issue 1 Jan 2022, pp: 579-583 www.ijaem.net ISSN: 2395-5252

in an error. Exceptions can be made for a variety of reasons, including:
- the Web server has run out of disc space;
- a user enters an invalid value in a form field;
- the file or database record does not exist;
- theprogramme does not have authorization to perform anything;
- a service is temporarily or permanently unavailable.

Developers must deal with mistakes in instances like these. As a result, any programming system should have exception management. In addition, the whole occurrence of the exception should be documented. Notifications should be issued to other systems/teams based on the severity of a mistake.
It should be possible to save the current code state when an exception is thrown. Then, for exception management, the execution of the code should be outsourced to a designated function. And, depending on the scenario, this function either ends the code, continues it with a different condition, or restarts it from a previously stored state.
There is an example of error handler with exceptions:

```
<?php
//create function with an exception
   public function checkData($data) {
      if ($data>1){
      throw new Exception (“Value must be validate”);
       } return true;
      } //trigger exception;
       try {
  //code, where error can be occured
} catch( Exception $e ) {
  echo "Error text: {$e->getMessage()}";}
?>
```

When an error happens in this situation, the script will terminate. Additionally, the user will receive a personalised error message that contains no information about the source file, stack trace, or code.

Exceptions are a component of almost every programming language, therefore knowing how to utilise them appropriately is essential. Where a webpage interacts with a database, exception management is crucial for Web system security.

In reality, the Rewrite is a rather simple to use SQL injection protection module. When utilising the Rewrite module, it must be enabled in the Apache HTTPd.conf settings. The replacement rules can be set up in two different ways [5]. When adding a replacement rule, the first step is to configure the virtual host, and the second is to create an html access file to replace the rules file in the site directory. The first benefit is that Apache loads rule information automatically when it starts up, allowing it to run effectively and replacing the rules defined in the Apache Configuration File. However, you must restart Apache after altering the replacement rules, which disrupts your operation. The second method is more adaptable.The Apache setup file and the replacement rules are kept separate. They go into effect immediately when the regulations are added or modified. Although there is a low running efficiency, you do not need to restart Apache.

REFERENCES

[1]. Maulud DH, Ameen SY, Omar N, Kak SF, Rashid ZN, Yasin HM, et al. Review on natural language processing based on different techniques. Asian Journal of Research in Computer Science.
[2]. Thiyab RM, Ali M, Basil F. "The impact of SQL injection attacks on the security of databases," in Proceedings of the 6th International Conference of Computing & Informatics. 2017.
[3]. Kaur N, Kaur P. SQL injection–anatomy and risk mitigation. Cover Story What, Why and How of Software Security 7 Cover Story Developing Secure Software. 2014.
[4]. Yasin HM, Zeebaree SR, Zebari IM. "Arduino based automatic irrigation system: Monitoring and SMS controlling," in 2019 4th Scientific International Conference Najaf (SICN). 2019.
[5]. Salih AA, Ameen SY, Zeebaree SR, Sadeeq MA, Kak SF, Omar N, et al. Deep learning approaches for intrusion detection. Asian Journal of Research in Computer Science. 2021.
[6]. Kindy DA, Pathan ASK. "A survey on SQL injection: Vulnerabilities, attacks, and prevention techniques," in 2011 IEEE 15th international symposium on consumer electronics (ISCE). 2011.
[7]. Al Janaby AO, Al-Omary A, Ameen SY, AlRizzo H. Tracking and controlling highspeed vehicles via CQI in LTE-A systems. International Journal of Computing and Digital Systems. 2020.
[8]. Mohammed K, Ameen S. Performance investigation of distributed orthogonal space-time block coding based on relay selection in wireless cooperative systems; 2019.

[9]. Fawzi LM, Alqarawi SM, Ameen SY, Dawood SA. Two levels alert verification technique for smart oil pipeline surveillance system (SOPSS). International Journal of Computing and Digital Systems.
[10]. Zeebaree S, Yasin HM. Arduino based remote controlling for home: Power saving, security and protection. International Journal of Scientific & Engineering Research. 2014.
[11]. Vulnerability statisticsweb applications.
[12]. A.S. Markov, A.A. Fadin - Systematics vulnerabilities and security defects in software resources.
[13]. Halfond, G. William, J. Viegas, A. Orso. "A classification of SQL-injection attacks and countermeasures."
[14]. L-K. Shar, H-B Kuan. "Predicting SQL injection and cross site scripting vulnerabilities through mining input sanitization patterns." Information and Software Technology, 2013.
[15]. Sadeghian, Mazdak Zamani, Azizah Abd Manaf. "A taxonomy of SQL injection detection and prevention techniques. International Conference on. IEEE Informatics and Creative Multimedia.
[16]. MauludDH, Ameen SY, Omar N, Kak SF, Rashid ZN, Yasin HM, et al. Review on natural language processing based on different techniques. Asian Journal of Research in Computer Science. 2021;1-17.
[17]. Thiyab RM, Ali M, Basil F. "The impact of SQL injection attacks on the security of databases," in Proceedings of the 6th International Conference of Computing & Informatics. 2017;323-331.
[18]. Kaur N, Kaur P. SQL injection–anatomy and risk mitigation. Cover Story What, Why and How of Software Security 7 Cover Story Developing Secure Software. 2014;9:27.
[19]. Yasin HM, Zeebaree SR, Zebari IM. "Arduino based automatic irrigation system: Monitoring and SMS controlling," in 2019 4th Scientific International Conference Najaf (SICN). 2019; 109-114.
[20]. Salih AA, Ameen SY, Zeebaree SR, SadeeqMA, Kak SF, Omar N, et al. Deep learning approaches for intrusion detection. Asian Journal of Research in Computer Science. 2021;50-64.
[21]. MauludDH, Ameen SY, Omar N, Kak SF, Rashid ZN, Yasin HM, et al. Review onnatural language processing based on different techniques. Asian Journal of Research in Computer Science. 2021;1-17.
[22]. Thiyab RM, Ali M, Basil F. "The impact of SQL injection attacks on the security of databases," in Proceedings of the 6th International Conference of Computing & Informatics. 2017;323-331.
[23]. Kaur N, Kaur P. SQL injection–anatomy and risk mitigation. Cover Story What, Why and How of Software Security 7 Cover Story Developing Secure Software. 2014;9:27.
[24]. Yasin HM, Zeebaree SR, Zebari IM. "Arduino based automatic irrigation system: Monitoring and SMS controlling," in 2019 4th Scientific International Conference Najaf (SICN). 2019; 109-114.
[25]. Salih AA, Ameen SY, Zeebaree SR, SadeeqMA, Kak SF, Omar N, et al. Deep learning approaches for intrusion detection. Asian Journal of Research in Computer Science. 2021;50-64.