WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Pythonic Shield: Fortifying Your Database Against SQL Injection Attacks

Swetha. S [1], Jenani. S [2], Devadharshini. D[3], Mrs. Umadevi Venkat G[4]

[1 2 3] Student, Computer Science and Engineering, Agni College of Technology, Chennai-600 130, Tamil Nadu, India

[4] Assistant Professor, Computer Science and Engineering, Agni College of technology, Chennai-600 130, Tamil Nadu, India

ABSTRACT –

Existing vulnerabilities in the Web system threaten information systems. Organisations must take measures to secure their systems and protect their data. SQL injection is a cyberattack where an attacker uses malicious code into web apps to access the database unofficially. In 2022, SQL Injection attacks are the most consequential security risk. To access the unauthorized databases, Hackers uses SQL injection type.

In the article, various techniques for safeguarding web applications against SQL injection assaults are examined. Additionally, measures to protect these same web apps from such malicious attacks are also discussed in detail. The software gives permission to user so that they can protect their web apps from SQL injection.

Keywords- Web resource, SQL Injection, Prevention system, Vulnerabilities, Queries, Database.

I NTRODUCTION

There are many web resources that don't fulfil modern security requirements. This may cause complications to the particular organization. An attacker uses SQL Injection to manipulate a web application's database by injecting malicious SQL code into its input fields. Attacker gains unauthorized access to data, modify or delete data, and can even access the entire application. SQL injection is a cyber-attack where an attacker uses malicious SQL code into a web applications input field to manipulate its database. Hackers often exploit a common vulnerability in websites and applications that use databases, by injecting malicious code through fields where users enter data such as HTML forms. This data can be sent to the web server in the form of POST or GET request. Then the server takes, verifies, processes the data and send back the responses to its client. Insufficient data filtering can result in SQL injection attacks. These attacks occur because the data is processed without any validation, allowing malicious code to be executed. For example (Fig. 1), for authentication basically users enter their username and passwords, those username and passwords are used to create the SQL query (e.g., a SELECT query), which was fetched to the database. If the entered values are valid as expected then the user access is allowed else access gets denied. If the users enter data with some special characters that damages the structure of the database. To avoid this, it is important to validate user input and make sure that special characters are not accepted. To safeguard against the injection of malicious code, it is crucial to utilize appropriate filtering and escaping method .To ensure the secure environment, it is predominant to enforce strict validation and the filtering in order to protect the malicious code injection. Without proper protection SQL injection can be dangerous and can result in a wide range of catastrophic. This paper gives a background study about SQLI attacks, the process of SQLI attacks, and the introductory types of attacks along with the sources. In section three, there will be a literature review of the SQLIA discovery and forestalment system proposed by numerous experimenters. In section four, During our analysis, we have reviewed and compared the different styles or approaches suggested, and evaluated the limitations of each.

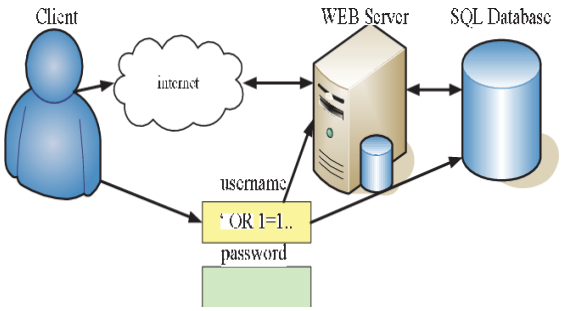

Fig.1: SQL Injection Scheme

SQL INJECTION ATTACK TYPES

In the current era, the prevalence of daily attacks surpasses the staggering number of 40,000, highlighting the urgent need for effective solutions to address security concerns across diverse web applications. Today, many web-based applications use databases to store the data needed for the website's activity. In order to create web products that are interesting and useful for the client, developers need to use various technologies. For example, online stores, internet banking and etc,.[06-09].

There are three major categories in SQL Injections:

- In-band SQLi
- Blind SQLi
- Out-of-band SQLi

In-band SQLi :The attacker uses the same channel of communication to gather results by attacking a web. There are two common types, they are Error-based SQLi and Union-based SQLi.

Blind SQLi: In this Blind SQLi, It doesn't involve direct transfer of data through the web application. There are two most common types, they are Boolean and Time-based.

Out-of-band SQLi: This type of SQL injection occurs when the attacker faces limitations in using the same communication channel to initiate the attack and obtain the desired results.

For example,

```
#          Injection          work          using          cursor.execute(sql)
#id                 =                 '1                OR                 1=1'
#cursor.execute("SELECT * FROM accounts WHERE username = %s" % (username,))
```

The above example comes under Blind SQLi. This SQL Injection attack known as 'Boolean-based SQL injection.
Here the attacker handles the input value (username) of SQL query to extract the information from the database. Since the condition '1=1' is always true ,so this query will return all records in the 'accounts' table which allow the attacker to authorize our Database without any permission and thus it can lead into hacking.

BACKGROUND STUDY

SQLIA Definition

Utmost web- grounded operations have a three- league design a donation league that serves as the stoner interface, a business league that handles all logical processes, and a data league that stores all structured data. A SQLIA uses all three categories for a successful attack. The donation subcaste is used to shoot vicious SQL commands to the business league to alter the being SQL queries, which exploits the database league with the intention of gaining access to means. The lack of confirmation of both donation and business categories in web operations leads to a successful SQLIA. [12]

SQLIA Procedure

The procedure of the SQLIA starts with the web interface which is used to shoot the stoner's inputs to the backend garçon. By relating the shy attestations in the input fields, the bushwhacker recognizes the possibility of aSQLI.SQLIA is constantly used to gain access to a system by avoiding security mechanisms. This can be fulfilled by fitting vicious query into the stoner login inputs that, when reused, always returns a true condition.(12). 1. Inserts vicious SQL queries through the stoner input fields 2. Alters the being SQL queries by adding the vicious query entered from the web interface(bushwhacker) through HTTP request. 3. Executes the new vicious SQL statement in the database .

SQLIA Consequences

The goods of SQLIA range from data exposure to full remote control of the database and include a number of different issues. which is an authorization honour and allow for SQLIA- grounded changes to vital data. also, authentication enables a login to the system indeed when the stoner doesn't have the correct username or when the authentication runner's log credentials aren't duly controlled. Confidentiality is also lost when sensitive data, similar as credit card figures or particular information, is bared. Integrity's, still, has the capability to modify or abolish this non-public dated.

SQLIA Attack Types

SQLI attacks can in a variety of forms. In this section, the most prevalent forms of SQLIA will be discussed, along with how they are executed, effects and consequences.

Tautology: A tautology attack is where an attacker exploits an injectable field in a SQL query and inserts a malicious value to it in the intention of always making conditions true. This attack is primarily done by reforming the WHERE condition of a SQL statement. As an example, this method can be used by an attacker to bypass the authentication mechanism by manipulating the SQL query to return true for any username and password provided.

Blind SQL injection: An instance of a SQL Injection attack AKA blind SQL injection implies asking the database true or false questions in according to figure out the response sent on the action taken by the application. Blind SQL attacks are often exploited when the allowing code is not alleviated during a web app is configured to exhibit common error messages.

Union query: This is kind of an attack where the intruder combines a malicious query to the original query applying the UNION SQL operator. The attacker will be able to access other SQL tables by combining the outcomes of the malicious query and the outcomes of the original query.

PREVENTION SYSTEM

When the hacker is trying to access our database he performs,

1) Initially the hacker asks for something specific that is not in the list .
2) Verifying user-provided information is crucial as the input data can pose risks if utilized without thorough validation
3) As a result, requests will be rejected if the injection is attempted, further protecting the transaction information from malicious hackers.
4) Thus, blocking injection to the Database
5) Furthermore, the forestalment system also has a built-in system for identifying and analysing errors.

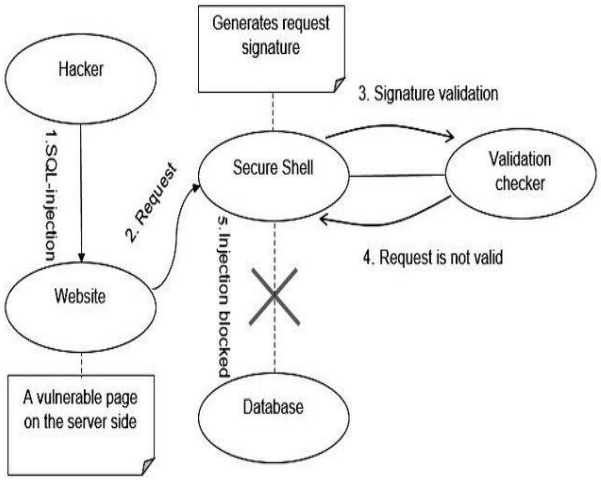

Fig 2. Prevention Scheme

The above given fig (fig.2) represents the technique under done in our system. When the hacker tries to inject malicious code into our website by requesting to the server, the server checks whether the given input is valid or not using a validation checker. If the input is valid further steps will be carried on. If the input is not valid then the validation checker sends an acknowledgement message to the Server informing about the input validation, thus by doing that malicious code can be rejected and our database can be secured.

Defences:

1. Prepared statement should be used
2. Implement well-constructed procedure
3. Employ encryption and data protection
4. Conduct regular vulnerability assessments and penetration testing
5. Enforce the principle of least privilege
6. Implement intrusion detection and prevention systems:
7. Conduct regular security audits and reviews

These are measures can be under taken to prevent SQL injection.

CONCLUSION

The protection system we propose consists of a series of scripts that are integrated into the webpage's infrastructure. These scripts are designed to provide defence against SQL injection attacks by operating within the webpage's framework. This system effectively prevents SQL injection attacks, making it a cost-effective for protecting web attack on SQL databases .SQL injection attacks are continuously getting a serious trouble to programmers and web operations. There are numerous ways to launch these attacks, but there are also numerous ways to descry them and help them. In previous studies, researchers established discovery methods for SQL injection attacks, but these have become ineffective due to the increasing intensity of the attacks. To address this issue, they conducted a comprehensive literature review on various SQL injection techniques. They also discussed multiple approaches to detect and prevent these attacks, including input validation, AES and DES encryption, machine learning algorithms, among others. This exploration could be helpful to the nonprofessional in understanding SQL and its pitfalls, and this study also helps programmers and experimenters who want to know about

all the issues that still affect web operations and which ways can be used to descry and help SQL injection attacks. It's anticipated that if the inventors follow the approaches bandied in this paper, the web operations will be free from similar destructive attacks.

As a result, users can be confident that their data is secure and defended from vicious pitfalls.

REFERENCES

1. Vulnerability statistics web applications (09.06.2016) http://ptsecurity.ru/download/analitika_web.pdf (in Russian)
2. SQL Injection – OWASP (19.06.2016) https://www.owasp.org/index. php/SQL Injection. (In Russian).
3. Top ten most critical Web Application Security Risks, OWASP-Open Web Application Security Project (19.06.2014) https://www.owasp.org/ index. Php/Category:OWASP_Top_Ten_Project (in Russian).
4. Halfond, G. William, J. Viegas, A. Orso. "A classification of SQL-injection attacks and countermeasures." Proceedings of the IEEE International Symposium on Secure Software Engineering. Vol. 1. IEEE, 2006, pp. 13-20.Li Qian, Zhenyuan Zhu, lun Hu, Shuying Liu “Research of SQL Injection Attack and Prevention Technology 2015” International Conference on Estimation, Detection and Information Fusion (ICEDIF2015), pp. 303-306.
5. T. Atefeh, M. Massrum, M. Zaman. "Comparison of SQL injection detection and prevention techniques" 2nd International Conference on Education Technology and Computer, Vol. 5, 2010, pp.348-359.
6. A design review: Concepts for mitigating SQL injection attacks." 4thInternational Symposium on Digital Forensic and Security (ISDFS)».2016, 169 p.
7. J. Clarke. “SQL Injection Attacks and Défense. Second Edition”,Syngress, 2012, 576 p.
8. M. Al Rubaiei, T. Al Yarubi, M. Al Saadi and B. Kumar, ”SQLIADetection and Prevention Techniques,” 2020 9th International Conference System Modelling and Advancement in Research Trends (SMART),2020, pp. 115-121, doi: 10.1109/SMART50582.2020.9336795