International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 09 | Sep 2020 www.irjet.net p-ISSN: 2395-0072

STUDY ON SQL INJECTION TECHNIQUES & MITIGATION

Ravindra Auti[1] and Prof. Indira Bhattachariya[2]

[1]Student, Department of MCA, Vivekanand Education Society's Institute of Technology (VESIT), Mumbai, India
[2]Associate Professor, Department of MCA, Vivekanand Education Society's Institute of Technology (VESIT), Mumbai, India

---

Abstract - *In modern days, cyber threats and attacks are triggered to corrupt or steal the knowledge of an individual in huge volume of knowledge from different lines of companies. Across the world, nowadays it became mandatory to guard the database from security related attacks. SQL injection is a High risk and most vulnerable threat which may exploit the entire database of any organization whether it is a private organization or a government sector, where code is injected in a web page. This code injection technique is employed to attack data-driven web applications or applications. A SQL statement will be changed in such a manner, which goes with always true as constraint. This study paper is ready to offer a comprehensive information about topics like basics of SQL Injection, types, recent attacks and mitigation as a case study. This survey won't be complete, if we miss out to learn the algorithms, which used as a base to identify vulnerability in this internet connected world which in turn exploits the database and discover top secrets. Tautology SQL injection – one of the code injection attack is widely used as a data – driven attack and gaining unauthorized access as per the security related literatures and causes severe damage to the organizational data banks and government sector.*

***Key Words*: SQL Injection, Tautology, security, detection, prevention**

1. INTRODUCTION

In this era, websites have become the most essential part in our lives. Among the top most security threats SQL Injection attack ranks top based on OWASP Top 10 security vulnerability report. Through these websites we insert number of personal data which gets stored in the database. We can access it from anywhere using network. This opened the gate for the attackers to grab those data from vulnerable web pages. To find those vulnerable web pages the attackers can find many efficient tools like botnet which generate the list of vulnerable web pages. Once the webpage is detected the attacker start to steal the data using SQL Injection attack. Web page detection is done to intrude inside the Database. So they target the webpage which is connected with back end database.

In this paper we try to answer the following review questions

1. What does SQL Injection means?

2. Why SQL Injection attack is done ?

3. How this attack is processed ?

4. What will be the consequences of this SQL Injection attack?

5. In how many ways these attacks are grouped?

6. List out the types of attack.

1.1 What is SQL injection (SQLi)?

SQL injection may be a web security vulnerability that permits an attacker to interfere with the queries that an application makes to its database. It generally allows an attacker to see data that they're not normally rights to retrieve. This might contain data belonging to other users, or the other data that web application itself is able to access. In many cases, an attacker can change or delete this data, causing persistent changes to the application's content or data

2. Overview of SQL Injection

2.1 Terms

SQL stands for structured query language which must be pronounced as se-qual. This language is mainly developed for interacting with the relational database. For data manipulation, Query is used to insert data, modify the database, to access the required data alone. Here comes the SQL injection which is done through SQL query under data manipulation

2.2 Purpose of attack

This attack is done based on two tasks. One is to gain benefit by grabbing others sensitive data and another one is to test the knowledge i.e. curiosity in learning new tasks and try to prove them.

2.3 Processing steps

Attacker finds out the vulnerable web pages with the help of some predefined tools. Through those webpage malicious HTTP request is send to the database where injected query try to get privileges for data manipulation

2.4 Classification

The attack is classified based on the attackers intention, vulnerabilities and asserts. Based on intention of the attacker we can have a classification in their goals.

- ***Goals***

i) **To extract data** – Sensitive data will be grabbed by the attacker. Suppose if admin database is hacked the entire database becomes vulnerable.

ii) **To access data** – They try to break the privileges and get access to the entire database and try to manipulate the data.

iii) **Finger print the database**- In this attack, database version and its type will be derived out by the attacker. This attack help them to try different type of queries in different application.

iv) **Injectable parameters are found** – using some of the automatic tools the vulnerable parameters will be found for attack.

v) **Authentication Bypass** –application authentication mechanisms will be bypassed to enter inside the database.

vi) **Database schema identification** - From the database table name, data type of each field, column name, etc. will be retrieved to gather information successfully

vii) **To perform denial of service** – Dropping table and system shutdown falls under this category. Attacker tries to intrude inside the system to perform some specific instruction within the database.

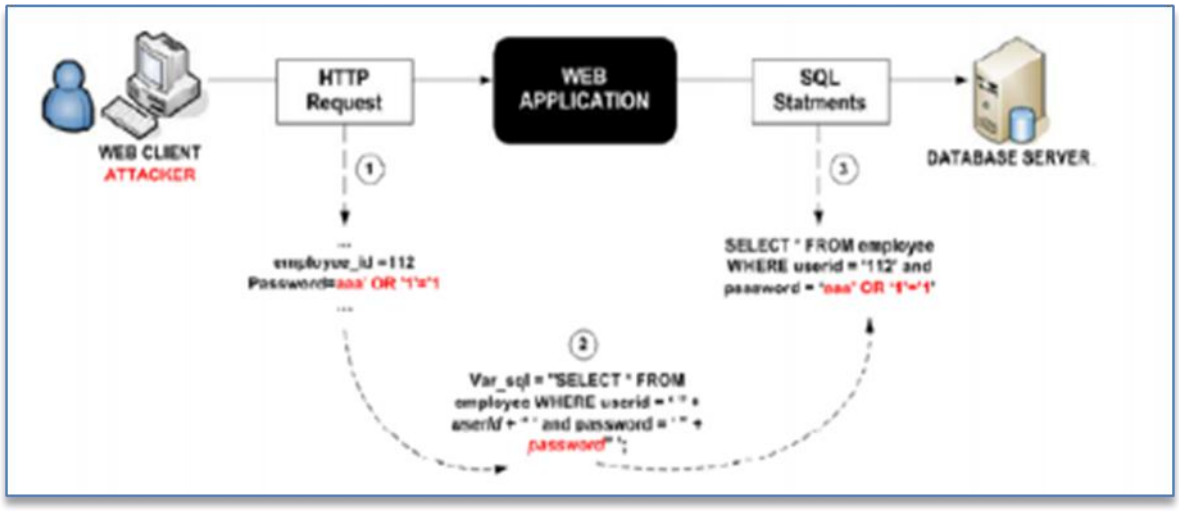
SQL Injection Attacks Process

SQL Injection Attack is a hacking technique which the attacker adds SQL statements through a web application's input fields or hidden parameters to access to resources. Improper input validation in web applications causes hacker to be successful by doing malicious activity. For the following examples we will assume that a web application receives a HTTP request from a client as input and generates a SQL statement as an output for the back end database server. For example an administrator will be authenticated after typing: employee id=123 and password=admin. Below figure describes a login by a malicious user exploiting SQL Injection vulnerability. Basically it is structured in three phases:

1. An attacker sends the malicious HTTP request to the web application

2. Creates the SQL statement

3. Submits the SQL statement to the back end database



3. Types of SQL Injection Attacks

There are numerous SQL Injection attacks and it is performed sequentially or in combinatorial.

1) Tautological attack:

Result - authentication page is bypassed for data extracting Tricks applied – 'where' clause in SQL tokens is injected to make the conditional query remains true

2) Union Query:

Result – Different dataset is retrieved from the Database

Tricks applied – SQL Injected query remains safe by joining the keyword 'union'

3) Illegal/Logically Incorrect Queries:

Result – Error message with useful debugging information

Tricks applied – By cause injects query with type mismatch, syntax error, logical errors

4) Piggybacked Queries:

Result – multiple queries are executed without the knowledge of the user which may lead to Database exploitation

Tricks applied – Many injected queries are added to the normal executable query

5) Inference:

Result – different responses from database is cross checked by changing its behavior.

Tricks applied – True/False questions using SQL statements is asked in serious (Blind attack). Based on time delay injected SQL queries are executed using if/then statement (Timing attack)

6) Stored procedure:

Result – remote commands, denial of service is performed

Tricks applied – Injection attack is done to the stored procedure present in the Database

Table 1: Types of SQL Injection with Example

Sr.No.	SQL Injection Attack Types	Purpose	Example Code
1	Tautologies	Bypass authentication	Select * from userdet where uid='abcd' and pwd ='admin' or '1'='1'
2	Union	Extracting or retrieved Data	Select * from userdet where uid='' union select * from details -- and pwd='x';
3	Illegal/ logical incorrect queries	Identify injectable parameters	SELECT * FROM students WHERE username = 'admin"' AND password =
4	Piggybacked Queries	Extract different datasets	SELECT Rno FROM St WHERE login = 'xyz' AND pass = ''; DROP table St --'
5	Inference	Defining Database Schema	SELECT name, email FROM members WHERE id=1; IF SYSTEM_USER='test' SELECT 1/0 ELSE SELECT 5
6	Stored Procedure	Execute remote commands	SELECT Eid, Ename FROM Employee WHERE Ename LIKE '1' or '1' = '1'; EXEC master.dbo. xp_cmdshell 'dir'--'

4. Impacts of SQL Injection Vulnerability

Using SQL injection Attacker can gain unauthorized access of application and can compromise with confidentiality and integrity of database. An adversary can steal sensitive information which is stored in databases used by vulnerable programs or applications such as user credentials, personal information, bank information or transaction records. SQL injection vulnerabilities should never be left open; they must be patched in all circumstances. If the authentication or authorization aspects of an application is affected an attacker may be able to login as any other user, such as an administrator which leads to privilege escalation.

5. Mitigation of SQL Injection Vulnerability

SQL injection has been use for extracting data and bypass authentication for long time and, as such, there are multiple ways to patch these vulnerabilities. In short, any data that comes from a third-party source or users such as user input should not be trusted and assumed to be malicious. Technically, user input validation can be done through the following techniques & methods.

· Client-side AND most importantly server-side validation through the use of a whitelist · Use Prepared SQL Statements with parameterized queries
· Use stored SQL procedures
· Escape (a type of encoding) all user-supplied input.

Organizations or companies should view into implementing a Systems Development Life Cycle (SDLC) policy that acquires secure coding practices while ensuring static and dynamic code analysis is regularly performed. A final penetration test should also be considered important to going live the application to ensure all issues have been addressed.

When sanitizing and validating user input ensure a whitelist is used, Instead of a black list. A blacklist requires eliminate every single character that can be malicious. Ensure all whitelisting and sanitization is performed on the client-side as well as server-side.

Many times SQL injection can be prevented by using parameterized queries instead of string concatenation within the query. Parameterized queries can be used for any condition where untrusted input comes from user as data within the query, including the WHERE clause and values in an INSERT or UPDATE statement. They cannot be used to handle untrusted input in other parts of the query, such as table names, or the ORDER BY clause.

For prevent SQL injection attacks, the string that is used in the query must always be a hard-coded constant, and must never contain any variable data or content from any origin. we should use something called prepared statements which uses bound parameters. Prepared Statements do not combine variables with SQL strings which comes as user input, so it is not possible for an attacker to modify or alter the SQL statement.

References

[1]. https://www.slideshare.net/pallavibiswas1/sql-injection-16984356
[2]. https://www.slideshare.net/AnoopT3/sql-injection-69621267?from_action=save
[3]. https://www.slideshare.net/helloanand/sql-injection-13537064
[4]. https://en.wikipedia.org/wiki/SQL_injection
[5]. https://www.acunetix.com/websitesecurity/sql-injection/
[6]. https://www.guru99.com/learn-sql-injection-with-practical-example.html
[7]. https://portswigger.net/web-security/sql-injection