A STUDY ON SQL INJECTION TECHNIQUES


Assistant Professor, Department of Computer Science and Engineering, Srinivasa Ramanujan Centre, Sastra University, Kumbakonam, Tamil Nadu, India.

Email: rubidhadevi@src.sastra.edu

Received on: 15.10.2016
Accepted on: 12.11.2016

Abstract

In modern days, cyber threats and attacks are triggered to corrupt or steal the information of a person in huge volume of data from different lines of businesses. Across the globe, nowadays it became mandatory to protect the database from security related attacks. SQL injection is a familiar and most vulnerable threat which may exploit the entire database of any organization irrespective whether it is a private organization or a government sector, where code is injected in a web page. This code injection technique is used to attack data-driven web applications or applications. A SQL statement will be altered in such a manner, which goes with ALWAYS TRUE as constraint. This study paper is prepared to give a comprehensive coverage about topics like basics of SQL Injection, types, recent attacks as a case study. This survey will not be complete, if we miss out to learn the algorithms, being used as a base to trigger vulnerability in this internet connected world; which in turn exploits the database and exposes top secrets. Tautology SQL injection – one of the code injection techniques is widely used as a data – driven attack as per the security related literatures and causes severe damage to the organizational data banks.

Keywords - 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[1] 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 [2] which generate the list of vulnerable web pages. Once the webpage is detected the
attacker start to steal the data using SQL Injection attack. Webpage detection is mainly 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 the term SQL, Query, Injection really 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.

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 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 out of 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 Consequences

The consequences[3] are very high since the database consists of sensitive information. We can categorize the cost based on Authorization, Authentication, Data Confidentiality and Data Integrity.

2.5 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.
2.5.1 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

2.5.2 Vulnerabilities

i) Improper validation done in a webpage leads to exploitation

ii) Privileged access for particular account

2.5.3 Asserts

i) Database fingerprint

ii) Schema

iii) Data

iv) Network

3. Types of SQL Injection Attacks

There are numerous SQL Injection attacks and it is performed sequentially or in combinatorial. Table-1 shows examples for each types of attack and its illustration is given below.

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.

**Union Query:**

Result – Different dataset is returned from the Database

Tricks applied – SQLInjected query remains safe by joining the keyword ‘union’

**Illegal/Logically Incorrect Queries:**

Result – Error message with useful debugging information

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

**Piggybacked Queries:**

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

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

**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)

**Stored procedure:**

Result – remote commands, denial of service is performed

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

**Table 1:** Types of SQL Injection with Example.

<table>
<thead>
<tr>
<th>S.No</th>
<th>SQL Injection Attack Types</th>
<th>Purpose</th>
<th>Example Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Tautologies</td>
<td>Bypassing authentication</td>
<td>Select * from userdet where uid='abcd' and pwd =’a’ or ‘3’=’3’</td>
</tr>
<tr>
<td>2</td>
<td>Union</td>
<td>Extracting Data</td>
<td>Select * from userdet where uid='' union select * from details -- and pwd='a';</td>
</tr>
<tr>
<td>3</td>
<td>Illegal/ logical incorrect queries</td>
<td>Identify injectable parameters</td>
<td>SELECT * FROM students WHERE username = 'ddd'' AND password =</td>
</tr>
</tbody>
</table>
4. Piggybacked Queries

Extract different dataset

```
SELECT Rno FROM St
WHERE login = 'abc' AND pass = ''; DROP table St --
```


5. Inference

Determining Database Schema

```
SELECT name, email FROM members WHERE id=1;
IF SYSTEM_USER='sa' SELECT 1/0 ELSE SELECT 5
```

6. Stored Procedure

Executing remote commands

```
SELECT Eid, Ename FROM Employee
WHERE Ename LIKE '8' or '8' = '8';
EXEC master.dbo. xp_cmdshell 'dir'--
```

4. Related Work

Table-2 shows various techniques used for SQL injection prevention and detection which was developed on or before the year 2010. Even though there are number of tools and techniques available, data breach happens at high level.

Apart from this there are more detection and prevention methods which are discussed briefly.

a) Runtime monitoring technique developed by Ramya Dharam [4] was presented and evaluated to detect and prevent tautologies attack in web applications. Their view is that the Pre-deployment testing techniques alone will not detect the attack post deployment testing is also necessary which is used to identify valid/illegal execution paths using basis path and overflow.

b) Kanchana Natarajan et.al proposed secure algorithm named SQL Injection Free (SQL-IF) based on dynamic technique. The complete task of this algorithm is to check for special character, Boolean keywords and keywords used in SQL query. Detection is done based on union attack and illegal/logical query attack.

c) Zhongding Dong et.al had defined a new role called “smart-driver” which is located between the database and the web portal. All types of information is sent to the user along with a random number for authorized user identity purpose and to protect their data from the attacker. Entire processing is performed by smart-driver. Every time when user access data from the database unique identifier is given to the user. Even though the attacker has cracked the identity number initially, the next step is protected with the newly generated random number which is issued to the user alone. Prevention is done with the help of this smart-driver.

d) Sailor Pratik, Prof. Jaydeep Gheewala [7] proposed new methodology which prevents all types of SQL injection. It is done by blocking the common keywords used by the attackers like ‘or’, ‘union’, script’, etc. Number of comparisons
it raises false alarm also.

Table 2: Techniques for SQL Prevention and Detection before 2011.

<table>
<thead>
<tr>
<th>S.No</th>
<th>Techniques</th>
<th>Attack Prevention</th>
<th>Attack Detection</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>AMNESIA [20, 21]</td>
<td>Automatic</td>
<td>Automatic</td>
</tr>
<tr>
<td>4</td>
<td>CSSE [31]</td>
<td>Automatic</td>
<td>Automatic</td>
</tr>
<tr>
<td>5</td>
<td>DIWeDa [30]</td>
<td>N/A</td>
<td>Automatic</td>
</tr>
<tr>
<td>6</td>
<td>IDS [28]</td>
<td>Report generate</td>
<td>Automatic</td>
</tr>
<tr>
<td>7</td>
<td>JDBC Checker [14, 15]</td>
<td>Code modification suggested</td>
<td>Automatic</td>
</tr>
<tr>
<td>8</td>
<td>Positive Tainting [23]</td>
<td>Automatic</td>
<td>Automatic</td>
</tr>
<tr>
<td>9</td>
<td>SQLCheck [18]</td>
<td>Automatic</td>
<td>Partially automatic</td>
</tr>
<tr>
<td>10</td>
<td>SQLIPA [9]</td>
<td>Partially automatic</td>
<td>Automatic</td>
</tr>
<tr>
<td>11</td>
<td>SAFELI [29]</td>
<td>N/A</td>
<td>Partially automatic</td>
</tr>
<tr>
<td>12</td>
<td>SQLrand [19]</td>
<td>Automatic</td>
<td>Automatic</td>
</tr>
<tr>
<td>13</td>
<td>SQL Prevent [26]</td>
<td>Automatic</td>
<td>Automatic</td>
</tr>
<tr>
<td>14</td>
<td>SecuriFly [27]</td>
<td>Automatic</td>
<td>Automatic</td>
</tr>
<tr>
<td>16</td>
<td>SQLDOM [22]</td>
<td>Automatic</td>
<td>Automatic</td>
</tr>
</tbody>
</table>
e) Tejinderdeep Singh Kalsi et.al developed a novel approach for detection and prevention mechanism for SQL injection which is occurred due to dynamic statements. Static Pattern matching algorithm is used for detection and prevention of SQL injection attack. Using static pattern matching, the user generated SQL queries are cross checked with the static pattern list. If the pattern matches exactly then injected queries are detected otherwise dynamic phase is used. In dynamic phase score value of the SQL query is calculated with the threshold value. Suppose if the threshold value is high, then the new injected anomaly pattern will be stored in static pattern list.

f) Lee et.al [10] has developed an approach for detecting the attack based on static and dynamic analysis i.e. SQL query attribute values are removed from the webpage during the runtime which is considered as dynamic analysis and then the query is compared with the predefined SQL query which is analyzed earlier and this process comes under static analysis. This method can be implemented in all types of applications connected with the database. All types of SQL injection can be easily detected in this method except real time SQL attacks.

g) Data Sanitization technique was used to develop perfect Security engine which has been developed by Kirti Randhe [33] for SQL injection prevention and Cross site scripting attacks. In this method, user input is tested twice before entering the database. First one is to use “Reverse proxy technique” for sanitizing the input given by the user which may attack the database in future. Second, the input is sent to the proxy server before entering the application server, where data cleansing algorithm is triggered to make the user input sanitized. They have used three modules SQL Injection Preventer, Cross Site Scripting Preventer and Analysis Module. In the first two modules the client IP addresses are analyzed. If the attack persists from the same IP address more than three times then the IP address is blocked or otherwise it is sent to the third module where analysis is done to find the intrusion attack.

<table>
<thead>
<tr>
<th>Authors</th>
<th>Technique</th>
<th>SQL Injection types Prevented</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ramya Dharam et.al [4]</td>
<td>Pre-Deployment &amp; Post-Deployment Technique</td>
<td>Tautology attack</td>
</tr>
<tr>
<td>Kanchana Natarajan et.al [5]</td>
<td>SQL-IF Algorithm</td>
<td>Union, Illegal/Logical attack</td>
</tr>
</tbody>
</table>
There are numerous detection and prevention techniques developed in later 2012. In this paper we have reviewed some handful of paper where their techniques are stated out in brief. Table-3 shows the techniques used by respective authors for preventing SQL Injection types.

5. Conclusion

Mostly intermediate layer is used to accept input from the user through web application. To build this layer scripting languages are used. So to exploit the database attacker uses SQL Queries. To confuse this layer SQL Queries are reshaped by the attackers. In this paper we have focused on those reshaped SQL Queries. We have discussed about SQL Injection attack and its various prevention and detection mechanisms used before and after 2011. We can conclude that even there are more number of security measures developed there are also equal number of exploitation done.

5. References

1. The Open Web Application Security Project, “OWASP Top ten project”
   


   International Journal of Advancements in Computing Technology Volume 3, Number 7, August 201110.4156/ijact.vol3.issue7.11

<table>
<thead>
<tr>
<th>Authors</th>
<th>Technique</th>
<th>Prevention Techniques</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zhongding Dong et al [6]</td>
<td>Smart Driver</td>
<td>Tautology, Union, Illegal/Logical, Piggybacked.</td>
</tr>
<tr>
<td>Sailor Pratik et al [7]</td>
<td>Common Keyword Blocking</td>
<td>All types of injection partially</td>
</tr>
<tr>
<td>Tejinderdeep Singh Kalsi et.al[8]</td>
<td>Static Pattern Matching Algorithm</td>
<td>Stored Procedures</td>
</tr>
<tr>
<td>Lee et.al [10]</td>
<td>Combined Static &amp; Dynamic Analysis</td>
<td>Tautology, Union, Illegal/Logical, Piggybacked.</td>
</tr>
<tr>
<td>Kirti Randhe [33]</td>
<td>Data Sanitization Technique</td>
<td>Cross site scripting and All types of SQL Injection</td>
</tr>
</tbody>
</table>

5. “Generation of SQL-injection free secure algorithm to detect and prevent SQL-injection attacks” Kanchana Natarajan, Sarala Subramani, ISSN: 2212-0173 2012 Published by Elsevier Ltd. doi: 10.1016/j.protcy.2012.05.129


