ISSN: 0975-766X
CODEN: IJPTFI
Available Online through
Research Article
www.ijptonline.com

A STUDY ON SQL INJECTION TECHNIQUES

[1]Rubidha Devi.D*, [2]R.Venkatesan, [3]Raghuraman.K
[1, 2, 3]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 – SQL Injected 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.

S.No	**SQL Injection Attack Types**	**Purpose**	**Example Code**
1	Tautologies	Bypassing authentication	Select * from userdet where uid='abcd' and pwd ='a' or **'3'='3'**
2	Union	Extracting Data	Select * from userdet where uid='' **union** select * from details -- and pwd='a';
3	Illegal/ logical incorrect queries	Identify injectable parameters	SELECT * FROM students WHERE username = **'ddd"'** AND password =

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

with existing methods are done to prove that their proposed methodology is comparatively good. But, the problem is, it raises false alarm also.

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

S.No	Techniques	Attack Prevention	Attack Detection
1	AMNESIA[20,21]	Automatic	Automatic
2	Automated Approaches[32]	Automatic	Automatic
3	CANDID[11]	Automatic	Automatic
4	CSSE[31]	Automatic	Automatic
5	DIWeDa [30]	N/A	Automatic
6	IDS[28]	Report generate	Automatic
7	JDBC Checker[14,15]	Code modification suggested	Automatic
8	Positive Tainting[23]	Automatic	Automatic
9	SQLCheck [18]	Automatic	Partially automatic
10	SQLIPA[9]	Partially automatic	Automatic
11	SAFELI [29]	N/A	Partially automatic
12	SQLrand[19]	Automatic	Automatic
13	SQL Prevent [26]	Automatic	Automatic
14	SecuriFly[27]	Automatic	Automatic
15	Security Gateway[16]	Automatic	Detailed manual Specification
16	SQLDOM[22]	Automatic	Automatic
17	Swaddler [24]	Automatic	Automatic
18	Tautology Checker [25]	Code modification suggested	Automatic
19	WAVES[13]	Report generate	Automatic

20	WebSSARI[17]	Partially automatic	Automatic

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 3: Techniques used for SQL Injection Prevention.

Authors	**Technique**	**SQL Injection types Prevented**
Ramya Dharam et.al [4]	Pre-Deployment & Post-Deployment Technique	Tautology attack
Kanchana Natarajan et.al [5]	SQL-IF Algorithm	Union, Illegal/Logical attack

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

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" https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project.
2. Maciejak, D., Lovet, G.: Botnet-Powered SQL Injection Attacks: A Deeper Look Within. In: Virus Bulletin Conference, pp. 286–288 (September 2009). https://www.virusbulletin.com/conference/vb2009/abstracts/botnet-powered-sql-injection-attacks-deeper-look-within/
3. Atefeh Tajpour, Suhaimi Ibrahim, Maslin Masrom, "SQL Injection Detection and Prevention Techniques" International Journal of Advancements in Computing Technology Volume 3, Number 7, August 201110.4156/ijact.vol3.issue7.11

4. "Runtime monitors for Tautology based SQL Injection attacks", Ramya Dharam, Sajjan G.Shiva, "International Journal of Cyber Security and Digital Forensics (IJCSDF) I (3):189-203 The Society of Digital Information and Wireless Communication (SDIWC) 2012 (ISSN: 2305-0012)

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

6. "A Smart-driver Based Method for Preventing SQL Injection Attacks", Zhongding Dong, Yun Liu,Guixun Luo and Sumeng Diao, International Journal of Security and Its Applications Vol.8, No.2 (2014), pp.67-76, http://dx.doi.org/10.14257/ijsia.2014.8.2.07 ISSN: 1738-9976 IJSIA

7. Pratik H Sailor, Prof. Jaydeep Gheewala. "Detection and Prevention of SQL Injection Attacks", International Journal of Engineering Development and Research (IJEDR), ISSN: 2321-9939, Vol.2, Issue 2, and pp.2660-2666, June 2014, Available: http://www.ijedr.org/papers/IJEDR1402215.pdf

8. "Detection and Prevention Of Sql Injection Attacks Using Novel Method In Web Applications", Tejinderdeep Singh Kalsi, Navjot Kaur, Int J Adv Engg Tech/Vol. VI/Issue IV/Oct.-Dec.,2015/11-15, E-ISSN 0976-3945

9. S. Ali, SK. Shahzad and H. Javed, "SQLIPA: An Authentication Mechanism against SQL Injection", European Journal of Scientific Research ISSN 1450-216X Vol.38 No.4 (2009), pp 604-611.

10. Lee, Inyong, Soonki Jeong, Sangsoo Yeo, and Jongsub Moon. "A novel method for SQL injection attack detection based on removing SQL query attribute values." Mathematical and Computer Modelling, Volume 55, Issues 1–2, January 2012, Pages 58–68, 0895-7177/$

11. P. Bisht, P. Madhusudan, and V. N. Venkatakrishnan, "CANDID: Dynamic Candidate Evaluations for Automatic Prevention of SQL Injection Attacks", ACM Transaction on information System Security, pp.1–39, 2010.

12. Kemalis, K. and T. Tzouramanis, "SQL-IDS: A Specification-based Approach for SQL V. Nithya, IJECS Volume 2 Issue 4 April, 2013 Page No. 886-905 Page 902 injection Detection", SAC'08. Fortaleza, Ceará, Brazil, ACM, pp.2153 2158, 2008.

13. Y. Huang, S. Huang, T. Lin, and C. Tsai, "Web Application Security Assessment by Fault Injection and Behavior Monitoring", in Proceedings of the 11th International World Wide Web Conference (WWW 03), May 2003.

14. C. Gould, Z. Su, and P. Devanbu. JDBC Checker, "A Static Analysis Tool for QL/JDBC Applications", in Proceedings of the 26th International Conference on Software Engineering (ICSE04) Formal Demos, pp 697–698, 2004.

15. C. Gould, Z. Su, and P. Devanbu, "Static Checking of Dynamically Generated Queries in Database Applications", in Proceedings of the 26th International Conference on Software Engineering (ICSE 04), pages 645–654, 2004.

16. D. Scott and R. Sharp, "Abstracting Application-level Web Security", in Proceedings of the 11th International Conference on the World Wide Web (WWW 2002), pages 396–407, 2002.

17. Y. Huang, F. Yu, C. Hang, C. H. Tsai, D. T. Lee, and S. Y. Kuo, "Securing Web Application Code by Static Analysis and Runtime Protection ", in proceedings of the 12th International World Wide Web Conference (www 04), May 2004.

18. Z. Su and G. Wassermann "The essence of command injection attacks in web applications", In ACM Symposium on Principles of Programming Languages (POPL'2006), Jan.2006.

19. S. W. Boyd and A. D. Keromytis, "SQLrand: Preventing SQL Injection Attacks", in Proceedings of the 2nd Applied Cryptography and Network Security Conference, pages 292–302, Jun. 2004.

20. W. G. Halfond and A. Orso, "AMNESIA: Analysis and Monitoring for Neutralizing SQL-Injection Attacks", in Proceedings of the IEEE and ACM International Conference on Automated Software Engineering (ASE 2005), Long Beach, CA, USA, Nov 2005.

21. W. G. Halfond and A. Orso, "Combining Static Analysis and Runtime Monitoring to Counter SQL-Injection Attacks", in Proceedings of the Third International ICSE Workshop on Dynamic Analysis (WODA 2005), pages 22–28, St. Louis, MO, USA, May 2005.

22. McClure, and I.H. Kruger, "SQL DOM: compile time checking of dynamic SQL statements," Software Engineering, ICSE 2005, Proceedings. 27th International Conference on, pp. 88- 96, 15-21 May 2005.

23. William G. Halfond, Alessandro Orso, "Using Positive Tainting and Syntax Aware Evaluation to Counter SQL Injection Attacks", V. Nithya, IJECS Volume 2 Issue 4 April, 2013 Page No. 886-905 Page 903 14th ACM SIGSOFT international symposium on Foundations of software engineering, ACM. pp: 175 – 185, 2006.

24. Macro Cova, Davide Balzarotti. Swaddler:" An Approach for the Anomaly-based Detection of State Violations in Web Applications", Recent Advances in Intrusion Detection, Proceedings, volume: 4637 Pages: 63-86 Published: 2007.

25. G. Wassermann, Z. Su, "An analysis framework for security in web applications," In: Proceedings of the FSE Workshop on Specification and Verification of Component-Based Systems, SAVCBS,, pp. 70–78, 2004.

26. P.Grazie., PhD, "SQL Prevent thesis", University of British Columbia (UBC) Vancouver, Canada, 2008.

27. M. Martin, B. Livshits, and M. S. Lam., "Finding Application Errors and Security Flaws Using PQL: A Program Query Language" ACM SIGPLAN Notices, Volume: 40, Issue: 10 Pages: 365-383, 2005.

28. F. Valeur, D. Mutz, and G. Vigna., " A Learning-Based Approach to the Detection of SQL Attacks" in Proceedings of the Conference on Detection of Intrusions and Malware and Vulnerability Assessment (DIMVA), Vienna, Austria, July 2005.

29. X. Fu, X. Lu, B. Peltsverger, S. Chen, K. Qian, and L. Tao., " A Static Analysis Framework for Detecting SQL Injection Vulnerabilities",COMPSAC 2007, pp.87-96, 24-27 July 2007

30. A. Roichman, E. Gudes, "DIWeDa - Detecting Intrusions in WebDatabases". In: Atluri, V. (ed.) DAS 2008. LNCS, vol. 5094, pp. 313–329. Springer, Heidelberg (2008).

31. Tadeusz Pietraszek and Dhris Vanden Berghe., "Defending against Injection Attacks through Context-Sensitive String Evaluation", Proceedings of Recent Advances in Intrusion Detection (RAID2005).

32. Mei Junjin, "An Approach for SQL Injection Vulnerability Detection," Proc. of the 6th Int. Conf. on Information Technology: New Generations, Las Vegas, Nevada, pp. 1411-1414, Apr. 2009.

33. Kirti Randhe1, Vishal Mogal2, "Security Engine for prevention of SQL Injection and CSS Attacks using Data Sanitization Technique", International Journal of Innovative Research in Computer and Communication Engineering (An ISO 3297: 2007 Certified Organization)Vol. 3, Issue 6, June 2015.