A Survey of SQL Injection Attacks, Their Methods, and Prevention Techniques

Mahmood A. Al-Shareeda
National Advanced IPv6 Center
Universiti Sains Malaysia
Penang, Malaysia
alshareeda022@gmail.com

Selvakumar Manickam
National Advanced IPv6 Center
Universiti Sains Malaysia
Penang, Malaysia
selva@usm.my

Sari Ali Sari
Computer Science and Information Technology
Universiti Tun Hussein Onn Malaysia
Johor, Malaysia
alsarisari46@gmail.com

2022 International Conference on Data Science and Intelligent Computing (ICDSIC) | 979-8-3503-3488-3/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICDSIC56987.2022.10075706

Abstract—The vast majority of web applications' databases are vulnerable to SQL Query Injection Attacks, which let clients directly insert sensitive data. They carry out their operations by inserting nefarious SQL Injection Query codes into the client-side web API, which allows them to retrieve all the confidential and sensitive data from the database. SQL injection is a technique where web attackers post the malicious SQL injection Query, occupying the full admin login access of the web database, for malicious Input data modifications or deletion of the existing user User's Information. The goal of this technique is to change the structure and behavior of the query that the computer programmer has proposed. The examination of SQL injections, which target a Web application's front end to get access to its back-end database, It will address the implications, categorization, and techniques of these attacks. Then, two tactics have chosen to defend the database against SQL injection assaults. As a consequence, the design and development of any new SQL injection attacks for web application may use this work as a guide and reference

Keywords—SQL Prevention Technique; SQL injection attacks, SQL attacks; Web application security, password-Based Cryptography, ASCII based string matching

I. Introduction

More than three-quarters of scanned websites contained vulnerabilities in 2016, according to the most recent Symantec Internet Security Threat Report published in April 2017 [1]. Injections are the first serious security threats to websites, according to the most recent OWASP Top 10 report [2]–[6], with an average of more than 229,000 being discovered daily in 2016.

A general overview of the structure of database-driven Web applications must be presented in order to comprehend how SQL injection occurs because databases are a common requirement for most websites [7]–[9]. All known database servers can be accessed using the structured query language (SQL), which is a standard computer language for managing relational databases and manipulating data [10]–[15].

As depicted in Fig.1, a database-driven Web application essentially consists of three tiers: a presentation tier, which includes a Web browser like Internet Explorer, Google Chrome, or Firefox; a logic tier, which includes a programming language like C#, ASP, or PHP; and a storage tier, which includes a database like (MS SQL Server, MySQL, Oracle) [16]–[20]. The middle tier (the logic tier) receives requests from the front-end (the presentation tier) and responds by querying and updating the back-end

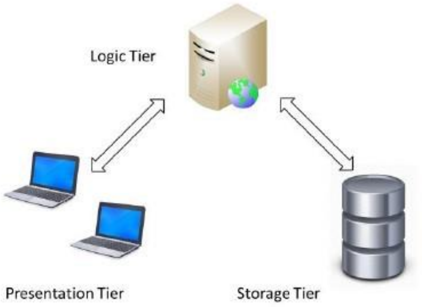


Fig. 1. The Web application's three layers.

By introducing malicious code into the front-end input parameters that are eventually given to the back-end database for processing and execution, an attacker can change a SQL statement and cause a SQL injection attack [21].

The rest of this paper is structured as follows. Section II provide the background of this paper. Section III explains the attack method of SQL injection. Section IV explains prevention against the SQL injection attacks. Finally, Section V provides the conclusion of this work.

II. II. BACKGROUND

A. SQL INJECTION ATTACK CONSEQUENCES

By virtue of its fundamental features, a SQL injection attack can target any database system [22]:

- Authentication: In a SQL injection attack, logging into the system is feasible without using the proper username and password.

- Confidentiality: A SQL injection attack aims to steal sensitive data, such as financial or personal information, which compromises confidentiality.
- Integrity and Availability: The loss of Integrity and/or Availability results from a SQL injection attack since it has the power to modify and/or destroy the data that is stored in the database.

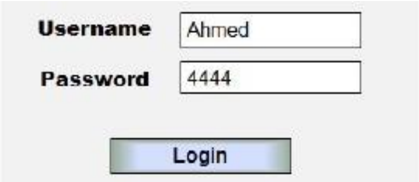
B. *SQL INJECTION ATTACK CLASSIFICATION*

Depending on the attacker’s goal, SQL injection attacks can be categorised in a number of ways [23]:

- Performing Denial of Service: The system will shut down as a result of the attacker injecting some SQL instructions for locking or removing database tables.
- Extracting, adding or changing data: An attempt is made to extract data from the back-end database by the attacker, who has the capacity to add or modify data values.
- Bypassing authentication: The hacker attempts to get beyond the database’s authentication procedures.
- Exploring database schema: In order to successfully extract data values from the database, the attacker tries to understand the database schema, including the names of the tables, columns, and data types.
- Database fingerprinting: To get ready for different types of attacks, the attacker tries to determine the kind and version of the database.

III. ATTACK METHODS

Depending on how the attack is carried out, there are a few different types of SQL injection attack strategies [24]. Assume Fig.2 represents a login screen for a website where a user must input their username and password to demonstrate that their account information has already been recorded in a backend database.



Username Ahmed

Password 4444

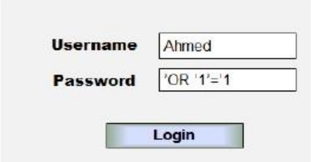
Fig. 2. website’s login page.

The username and password are compared by the login script, which then dynamically constructs a SQL statement that returns a record set:

Select ac _id from user table where username=’Ahmed’ and password=’4444’;

being aware that the database table user table contains all user account information We’ll now give a succinct illustration of each assault method:

- Logically incorrect query: By introducing logically flawed requests, the attacker can generate logical problems, type mismatches, or syntax issues that enable him or her identify table names, column names, and data types. This approach is typically employed by the attacker to obtain further information.
- Tautology: In this method, the WHERE condition is injected with the phrase ”the value entered in the input field is always true,” as seen in Fig.6. Select ac id from user table where username=” and password OR ’1’=’1’;

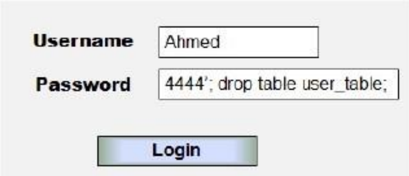


Username Ahmed

Password 'OR '1'='1

Fig. 3. Technique of tautology.

- Piggy-backed query: Using the query delimiter (;) as shown in Fig.4, this method includes additional queries in the value entered in the input field.
- Select ac id from user table where username=’Ahmed’ and password=’4444’; drop table user table;

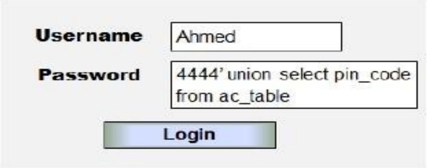


Username Ahmed

Password 4444'; drop table user_table;

Fig. 4. Technique of Piggy-backed query.

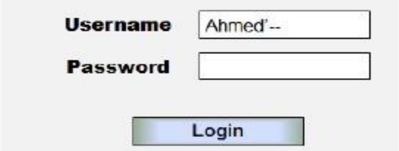
- Union query: As seen in Fig.5, this method uses the term UNION as part of the value entered into the input field to retrieve data pertaining to additional tables from the same database. Recognizing that ac table is a separate table that contains all user account information intended to be seen only to the database administrator. Select ac id from user table where username=’Ahmed’ and password=’4444’ union select pin code from ac table’;
- End of line comment: This method treats the remainder of the query as a comment, as seen in Fig.6, by ending the value supplied in the input field with a symbol. Select ac id from user table where username=’Ahmed’– and password=”;



Username Ahmed

Password 4444' union select pin_code from ac_table

Fig. 5. Technique of Union Query.



Username Ahmed'--

Password

Fig. 6. Technique of end of line comment.

The techniques described above are the most common types of SQL injections, but there are others that are not covered in this survey [25], [26].

IV. PREVENTION TECHNIQUES

By making a general survey about the used methods for preventing SQL injection vulnerabilities, we can find some simple techniques such as using of prepared statements with parameterized queries, using of stored procedures, white list input validation, and escaping all user supplied input [27]. SQL injections countermeasures are classified into three main categories [28]:

- Penetration testing.
- Defense mechanism deployment.
- Secure implementation.

In this survey, we'll concentrate on two techniques that alter the back-end database by adding additional columns to the same table or by creating new database tables, however other hybrid and complicated ways have been established for better results [29]–[31].

A. Prevention of SQL injection attacks using ASCII based string matching

Indrani Balasundaram and E. Ramaraj [32] presented this technique; it is a data-based validation in static and runtime validation to protect the Web application. It recommends that all private information be encoded in ASCII code and stored in a single database table; in our previous example of the login Web site, this would be table I. In order to authenticate a user when they attempt to log into the database, the values of the input fields should be translated into ASCII codes. Each acceptable username and password should be stored in the database table as an ASCII code.

TABLE I. COMPARISON BETWEEN DIFFERENT CRAWLERS

Username Username ASCII	Password	Password ASCII
Ahmed 65104109101100	4444	52525252
Sara 839711497	6666	54545454
Lim 76105109	8888	56565656

The shortcomings of this approach were identified by Mahima Srivastava [33]. Since SQL injections can occur anywhere that user input is required through input fields, they affect more than just authentications. As a result, storing those values and their ASCII codes requires double the amount of storage space. She discovered a novel strategy that is carried out in three stages. The first step is entering the data into the database. All information is converted to ASCII code before being stored there, and each value must be separated using a comma (,) as in the login Web application from our previous example as shown in Table II. The second phase is authentication, and the implementation algorithm is shown in Fig.7. Information retrieval, which takes place in the third phase, entails reshaping all requested material from ASCII code to its original form before it is shown to the user.

TABLE II. COMPARISON BETWEEN DIFFERENT CRAWLERS

Username ASCII	Password ASCII
65104109101100	52525252
839711497	54545454
76105109	56565656

B. Prevention of SQL injection attacks using Password-Based Cryptography

For secure identity authentication in Legacy Database Systems, Juanita Blue, Eoghan Furey, and Joan Condell present this approach [34]. The ”Salt table” is a new table that is added to the user table in the database using this technique. It has columns for storing the username, salt string, and protected password for each user account, while the main user table has columns for storing the username and a string of characters generated at random, as shown in Fig.8.

The outdated plaintext passwords are first salted with a random string and then encrypted/hashed using a one-way, well-proven cryptographic hash function, like SHA-256 or SHA-512. The output of this hash function should then be rehashed for 1000 times, as shown in Fig.9. If a SQL injection attack succeeds in extracting all usernames and passwords from the Salt table and User database, the attacker would have to un-hash and un-salt each password string for an illogically high number of repetitions, which would be a highly laborious and time-consuming procedure.

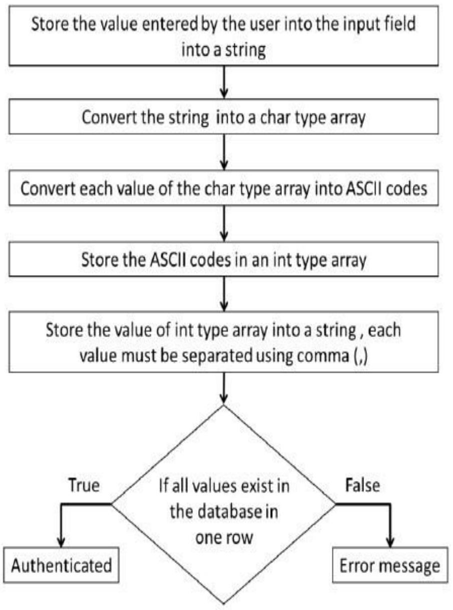

Store the value entered by the user into the input field into a string
Convert the string into a char type array
Convert each value of the char type array into ASCII codes
Store the ASCII codes in an int type array
Store the value of int type array into a string , each value must be separated using comma (,)
If all values exist in the database in one row
True
False
Authenticated
Error message

Fig. 7. Authentication implementation algorithm.

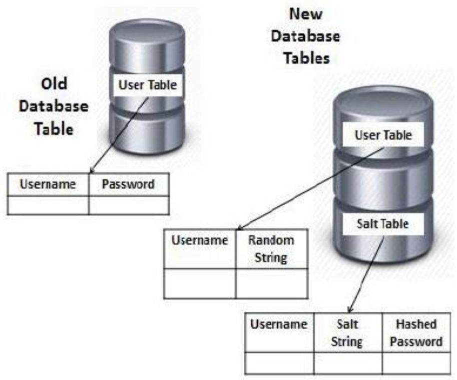


Fig. 8. Old and new databases that have been changed and contain Salt tables.

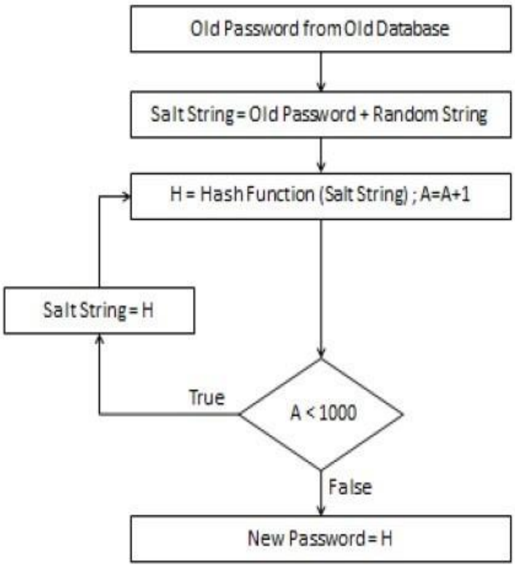

Old Password from Old Database
Salt String = Old Password + Random String
H = Hash Function (Salt String) ; A=A+1
Salt String = H
True
A < 1000
False
New Password = H

Fig. 9. algorithm to change a password from plaintext to hashed.

V. CONCLUSION

In this study, we explored SQL injection attacks and the typical approaches used in these assaults. Then two methods have been selected for mitigating database attacks: the first method turns all database entries into ASCII codes, while the second hashes plaintext passwords. The disadvantages of the first method include the possibility of reverse engineering if the attack is executed successfully and all of the ASCII codes are reconverted to their original plaintext. In contrast, the first method requires no additional space and the conversion process from plaintext to ASCII code takes a reasonable amount of time. Reverse engineering is highly difficult and time-consuming, assuming it is even possible. The second method requires extra space for the new Salt table and takes significantly longer to add random strings and hash the result for 1000 rounds. In future work, we extend this work by designing proposed solution in real environments. Additionally, the relevant related works should be reviewed as well

REFERENCES

[1] M. Fossi, G. Egan, K. Haley, E. Johnson, Trevor Mack,

[2] Teo Adams, Joseph Blackbird, Mo King Low, Debbie Mazurek, David´ McKinney, et al. Symantec internet security threat report trends for 2010. *Volume XVI*, 2011.

[3] M. A Al-Shareeda, M. Anbar, Iznan Husainy Hasbullah, and Selvakumar Manickam. Survey of authentication and privacy schemes in vehicular ad hoc networks. *IEEE Sensors Journal*, 21(2):2422–2433, 2020.

[4] M. A Al-Shareeda, M. Anbar, Selvakumar Manickam, and Ali A Yassin. Vppcs: Vanet-based privacy-preserving communication scheme. *IEEE Access*, 8:150914–150928, 2020.

[5] Y. Mahmood Hussain, HO Hanoosh, Z Zakaria, Fahad Taha Al-

[6] M. Ali Saare Dhief, M. Muhammad Jawad, A. Hamza Omran, and A. Abdulateef Abdulbari. Smartphone's off grid communication network by using arduino microcontroller and microstrip antenna. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, 19(4):1100–1106, 2021.

[7] M. A Al-Shareeda, M. Anbar, M. A Alazzawi, Selvakumar Manickam, and Ahmed Shakir Al-Hiti. Lswbvm: A lightweight security without using batch verification method scheme for a vehicle ad hoc network. IEEE Access, 8:170507–170518, 2020.

[8] D. Wichers. Owasp top-10 2013. OWASP Foundation, February, 2013.

[9] M. Maad Hamdi, L. Audah, S. Abduljabbar Rashid, and Mahmood Al Shareeda. Techniques of early incident detection and traffic monitoring centre in vanets: A review. J. Commun., 15(12):896–904, 2020.

[10] M.A Al-shareeda, M. Anbar, Iznan H Hasbullah, Selvakumar Manickam, Nibras Abdullah, and Mustafa Maad Hamdi. Review of prevention schemes for replay attack in vehicular ad hoc networks (vanets). In 2020 IEEE 3rd International Conference on Information Communication and Signal Processing (ICICSP), pages 394–398. IEEE, 2020.

[11] M. Maad Hamdi, A. Shamil Mustafa, H. Falih Mahd, Mohammed Salah Abood, Chanakya Kumar, and Mahmood A Alshareeda. Performance analysis of qos in manet based on ieee 802.11 b. In 2020 IEEE international conference for innovation in technology (INOCON), pages 1–5. IEEE, 2020.

[12] M. A Alazzawi, Hasanain AH Al-behadili, Mohsin N Srayyih Almalki, Aqeel Luaibi Challoob, and Mahmood A Al-shareeda. Id-ppa: robust identity-based privacy-preserving authentication scheme for a vehicular ad-hoc network. In International Conference on Advances in Cyber Security, pages 80–94. Springer, 2020.

[13] M. A Al-Shareeda, Mohammed Anbar, Selvakumar Manickam, Ayman Khalil, and Iznan Husainy Hasbullah. Security and privacy schemes in vehicular ad-hoc network with identity-based cryptography approach: A survey. IEEE Access, 9:121522–121531, 2021.

[14] S.Yue Wong, A. Hussain, and Murtaja Ali Saare. A survey analysis: Students'perceptions of using simulation game as learning tool. ASEAN Engineering Journal, 12(1):105–110, 2022.

[15] M. A Al-Shareeda, M. Anbar, S. Manickam, and I. H Hasbullah. Se-cppa: A secure and efficient conditional privacy-preserving authentication scheme in vehicular ad-hoc networks. Sensors, 21(24):8206, 2021.

[16] R. MehmoodGondal, S. Anwar Lashari, M. Ali Saare, and Sari Ali Sari. A hybrid de-noising method for mammogram images. Indonesian Journal of Electrical Engineering and Computer Science, 21(3):1435–1443, 2021.

[17] M. A Al-Shareeda, M. Anbar, S. Manickam, and I.H Hasbullah. Towards identity-based conditional privacypreserving authentication scheme for vehicular ad hoc networks. IEEE Access, 2021.

[18] M.A Al-shareeda, M. Anbar, S. Manickam, Iznan H Hasbullah, Ayman Khalil, Murtadha A Alazzawi, and

[19] A. Shakir Al-Hiti. Proposed efficient conditional privacy-preserving authentication scheme for v2v and v2i communications based on elliptic curve cryptography in vehicular ad hoc networks. In International Conference on Advances in Cyber Security, pages 588–603. Springer, 2020.

[20] T. Anwar Lashari, E.Amin, Sana Anwar Lashari, Murtaja Ali Saare, and Saima Anwar Lashari. Development of a web portal 'ikigai'to assess the psychological well-being of university students. In 2020 IEEE 7th International Conference on Engineering Technologies and Applied Sciences (ICETAS), pages 1–6. IEEE, 2020.

[21] M. A Al-Shareeda, S. Manickam, Badiea Abdulkarem Mohammed, Zeyad Ghaleb Al-Mekhlafi, Amjad Qtaish, Abdullah J Alzahrani, Gharbi Alshammari, Amer A Sallam, and Khalil Almekhlafi. Chebyshev polynomial-based scheme for resisting side-channel attacks in 5g-enabled vehicular networks. Applied Sciences, 12(12):5939, 2022.

[22] M. A Al-Shareeda, S. Manickam, B. Abdulkarem Mohammed, Zeyad Ghaleb Al-Mekhlafi, Amjad Qtaish, Abdullah J Alzahrani, Gharbi Alshammari, Amer A Sallam, and Khalil Almekhlafi. Cm-cppa: Chaotic map-based conditional privacy-preserving authentication scheme in 5g-enabled vehicular networks. Sensors, 22(13):5026, 2022.

[23] M. A Al-Shareeda and S. Manickam. Security methods in internet of vehicles. arXiv preprint arXiv:2207.05269, 2022.

[24] J. Clarke. SQL injection attacks and defense. Elsevier, 2009.

[25] A. Tajpour, S. Ibrahim, and M. Masrom. Sql injection detection and prevention techniques. International Journal of Advancements in Computing Technology, 3(7):82–91, 2011.

[26] H. Dehariya, P. Kumar Shukla, and M. Ahirwar. A survey on detection and prevention techniques for sql injection attacks. *International Journal of Wireless and Microwave Technologies*, 6(6):72– 79, 2016.

[27] K. Umar, A. Bakar Md Sultan, H. Zulzalil, Novia Admodisastro, and Mohd Taufik. Sql injection attack roadmap and fusion. *Indian Journal of Science and Technology*, 9(28):1–8, 2016.

[28] Ammar Alazab and Ansam Khresiat. New strategy for mitigating of sql injection attack. *International Journal of Computer Applications*, 154(11), 2016.

[29] J. Puneet Singh. Analysis of sql injection detection techniques. *arXiv preprint arXiv:1605.02796*, 2016.

[30] Chat Room. Sql injection. *database*, 4(20):51, 2022.

[31] H.-Chuan Huang, Z.Kai Zhang, Hao-Wen Cheng, and Shiuhpyng Winston Shieh. Web application security: threats, countermeasures, and pitfalls. *Computer*, 50(6):81–85, 2017.

[32] L. Khin Shar and H. Beng Kuan Tan. Defeating sql injection. *Computer*, 46(3):69–77, 2012.

[33] A. Ghafarian. A hybrid method for detection and prevention of sql injection attacks. In *2017 Computing Conference*, pages 833–838.

[34] IEEE, 2017.

[35] U. Upadhyay and Girish Khilari. Sql injection avoidance for protected database with ascii using snort and honeypot. In 2016 International Conference on Advanced Communication Control and Computing Technologies (ICACCCT), pages 596–599. IEEE, 2016.

[36] I. Balasundaram and E Ramaraj. An efficient technique for detection and prevention of sql injection attack using ascii based string matching. Procedia Engineering, 30:183–190, 2012.

[37] M. Srivastava. Algorithm to prevent back end database against sql injection attacks. In 2014 International conference on computing for sustainable global development (INDIACom), pages 754–757. IEEE, 2014.

[38] J. Blue, E.Furey, and J. Condell. A novel approach for secure identity authentication in legacy database systems. In 2017 28th Irish Signals and Systems Conference (ISSC), pages 1–6. IEEE, 2017.