I.J. Wireless and Microwave Technologies, **2021, 3, 16-21**
Published Online June 2021 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijwmt.2021.03.03

SQL Injection Detection Tools Advantages and Drawbacks

Hazem M. Harb[a], Derar Eleyan[b], Amna Eleyan[c]
Palestine Technical University, Kadoorie[a], Department of applied computing, Palestine Technical University Kadoorie [b], Manchester Metropolitan University, Department of Computing and Mathematics [c]
Email: h.m.harb@students.ptuk.edu.ps, d.eleyan@ptuk.edu.ps , a.eleyan@mmu.ac.uk c

Received: 05 January 2021; Accepted: 20 February 2021; Published: 08 June 2021

Abstract: SQL injection attack is a major threat to web application security. It has been rated as one of the most dangerous vulnerabilities for a web-based application. Based on the Open Web Application Security Project (OWASP), it is measured as one of the top ten. Many types of research have been made to face this attack either by preventing the threat or at least detecting it. We aim in this paper to give an overview of the SQL injection (SQLI) attack and classify these attacks and prevention and detection tools. We introduce the most current techniques and tools that are used to prevent and detect SQLI and highlight their strengths and weaknesses.

Index Terms: SQLI, Web-based application, prevention and detection tools, Static analysis, Dynamic analysis

1. Introduction

During the last few past years, SQL injection (SQLI) attacks have been encountered as one of the most dangerous types of attacks to web-based systems and are ranked number one among the Open Web Application Security Project's (OWASP) top vulnerabilities [1].

According to The National Vulnerability Database (NVD), it found that 7 percent of the total reported incidents in 2011 were caused by SQL injections [2]. Moreover, it was discovered that an average of 71 SQL injection attacks is attempted hourly. The first SQLI was detected in Feb 2002 when a young man retrieved more than 200 thousand name and credit card numbers from the Guess.com database [3]. The bad consequence of this SQL injection is a hacker can gain access to the database information easily.

Many researchers wrote about SQLI attacks and prevention detection tools. Every year a new technique and approach were proposed to prevent this kind of attack. Through the sections of this paper, we provide an overview of SQLI types and SQLI detection and prevention tools. In addition, we present a classification of some tools based on the underlying systems that are applied in. The strengths and weaknesses of these tools are listed. Finally, we propose the importance and the role of fuzzy testing on SQLI techniques.

2. Related Works

Web-based applications are very common these days. The main thread for this type of application is an SQL injection (SQLI) attack. Many researchers write about tools and techniques to prevent or detect SQLI attacks. There are different classifications of these techniques based on different criteria such as types and systems that are attacked.

A. Type of SQLI attacks

Table 1. SQLI Attack classical types

Name of the attack	Definition	Goals
Tautology	Simply a statement that is always true. In this kind of attack, an attacker tries to inject the main code with malicious code that returns true in all states. [6]	Bypass authentication, extract data.
Piggybacked Query	Inject (add) new statements to the original query, In this case, the database first runs the basic (main) query then the additional statements will be executed. [7]	Extract, modify or add data, execute the remote command on Database.
Logically Incorrect	Attacker get benefit from the error message that returned by the database. Used this information to gather vulnerable parameters in the application or database. [6]	Find injectable parameters, database finger-printing.

Union query	Inserting extra statement to the original query by using union command or ';'.The output result will be from both queries. [8]	Bypass authentication, extract data.
Stored Procedure	The attacker tried to reach the procedure which is stored in the database, add malicious code on it, and run it directly. [9, 10]	Perform privilege escalation, Perform DOS, run a remote command.
Inference	Using an Inference attack enables the attacker to change the behavior of a database or application. This type of attack can be classified into two well-known techniques, which are: Blind injection and timing attack [6]	Change behavior of database.
Alternate Encoding	The attacker modifies the injection query via using alternate encodings, such as hexadecimal, ASCII, and Unicode. By this attacker try to escape from any developer filter on the application. [8]	Bypass authentication, extract data.

Table 1 shows the classical types of attacks to SQLI attack according to many researchers. These days, the attacker developed new techniques to attack the database. The following are a summary of them: [9]

1. Fast Flux SQL Injection Attack:

It's a hybrid attack that combined SQLI and phishing. Depending on social engineering the attacker tries to get sensitive information from a user through a third party. Attacker to protect its criminal assets, the operator of phishing websites started using the Fast Flux technique. Fast Flux is a Domain Name Server technique to hide phishing and malware distribution sites behind an ever-changing network of the compromised Hos.

2. Compounded SQL Injection Attack:

It's a mixture of two or more attacks at the same time. This kind of attack developed as a result of new prevention and detection tools. It aims to beat this tool and reach the target database. In this attack, the attacker used an SQLI attack with other Web application attacks at the same time. Such as:

- SQL Injection + Insufficient Authentication.
- SQLIA using Cross-Domain Policies of Rich Internet application (RIA).
- SQL Injection + XSS (Cross-Site Scripting)
- SQL Injection + DDoS (Distributed Denial of Service) Attacks

3. SQLI Attack Detection Tools:

Some many techniques and tools used to detect SQLI. These tools are classified based on the type of attack (classical or modern). In the next section, we will briefly talk about these techniques.

A. Detection tools for SQLI (classical Type): [8]

Table 2. Tools deal with classical types of attacks

Type of attack / Used Technique	Tautologies	Logically Incorrect Queries	Union Query	Piggy Query	Stored Procedures	Inference	Alternate Encoding
AMENSIA	✓	✓	✓	✓	×	✓	✓
SAFELI	×	✓	✓	✓	✓	✓	✓
WASP	✓	✓	✓	✓	✓	✓	✓
R-WASP	✓	✓	✓	✓	×	✓	✓
RT-WASP	✓	✓	✓	✓	✓	✓	✓
SecuriFly	++	++	++	++	++	++	++
JDBC-Checker	++	++	++	++	++	++	++
CANDID	++	++	++	++	++	++	++
Swaddler	++	++	++	++	++	++	++
DIWeDa	×	×	×	×	×	✓	×
Positive Tainting	✓	✓	✓	✓	✓	✓	✓
SQL Prevent	✓	✓	✓	✓	✓	✓	✓
SQLIPA	✓	×	×	×	×	×	×
PDO	✓	×	✓	×	×	×	×
Automated Approaches	✓	✓	✓	✓	✓	✓	×

Table 2 shows the mentioned detection tools didn't deal with a modern type of SQLI Attack. All of them were tested on classical types of attack. Where (✓) means the tool's proof efficiency against this type of attack, (×) means the tool didn't work, while '++' means the tool needed help from another tool to work. This classification didn't mention type off application or type of database or if they affect or not on the tool.

B. B.2: Detection and Prevention tools for SQLI (Modern Type):

There is also another tool that is designed especially for facing and detecting the new type of SQLI Attack which can be found in the following table [8]:

Table 3- Tools Deal with Modern Type

SQLI types / Techniques	Fast Flux SQLIA	SQLI+XSS	SQLI+DNS Hijacking	SQLIA using Cross-Domain Policies	SQLI + DDos	SQLI+ Insufficient Authentication
Fast Flux Monitor	D	×	D	×	×	×
Machine learning methods	D	✓	×	×	D	×
SQLMap	N	D	++	×	D	D
Crypto-Graphical Hash Function	×	×	×	×	×	P
Noxes tool	×	++		P	×	×
Ardilla tool	×	D	×	D	×	×
Session Shield	×	✓	++	×	×	++

The symbols which are used in Table 3 have the following meaning:

D: Technique used for detection only.
P: Technique used for prevention only.
++: depicts the incompleteness which means another method has to be applied to achieve complete detection and prevention.
✓: Technique is used for both Detection and Prevention for the modern type of SQL Injections.
×: Techniques or tools do not correspond to modern SQL Injection (In terms of Prevention and Detection).

As seen from Table 3 and from reading further researches there are many tools and techniques were presented to detect and prevent SQL Injection Attack. Previous researchers' works on specific tools and techniques. Some others present a comparison in terms of a survey to describe different techniques and tools to detect and prevent SQLI attacks.

C. Grouping Techniques:

As mentioned above there are many techniques and tools used to prevent or detect SQLI tools. There was different testing done on these tools. This testing methodology can be divided into:

Static analysis: which means testing and evaluating the code but not running the application.
Dynamic analysis (Run-Time): This means testing and examining the tools during the running of the application.

In the next section, this research will display and analyze results for some techniques based on the type of testing (static or dynamic) analysis. Displaying weakness and strength for each one of them.

D. Static analysis technique:

Table 4. Static analysis tools

Technique	**Advantages** [10]	**Drawbacks** [10]
SAFELI: (MSLI) to detect SQL Injection Attacks. [11]	This approach can find and detect vulnerabilities that can't be found by black-box testing.	Should add code transformation in MSLI. Work on Microsoft platform only.
Mining input sanitization patterns for predicting SQLIVS. [12]	85% reported a detection rate	High false positive
An algorithmic approach for replacing insecure SQL statements in the code with a secure one. [13]	94% reported accuracy	Not active for Java- Code transformation Batch queries can't be detected efficiently.
Automated fix Generator SQLIAs [14]	Fully automated. High Efficiency	Should be supported by code transformation. Overhear is high Limited to PHP.
Automatic creation of SQL injection attacks for uncovering SQL injection vulnerabilities. [15]	There is no runtime overhead. No need to modify the target system.	Sometimes generate false positives. Limited to PHP and MYSQL
Signature and auditing method to prevent SQLIAs using [16]	Show low overhead in execution. No need for code transformation	It's restricted to web applications only.
SQL DOM [17]	Show high coverage all over SQLIA (Except Stored Procedure)	Cost high overhead on the system. Need a new model in programming.

A method for hunting SQL injection vulnerabilities. [18]	Effective, the high detection rate - Very low overhead	Complex involves different stages
An anomaly-based system that uses different detection models to detect unknown attacks. [19]	Low overhead	- Limited to PHP - Coverage problems - Generates false-negative results
ASSIST (Automatic and Static SQL Injection Sanitization Tool. [20]	Effective, low overhead, high detection rate.	Used for JAVA only, need code transformation, produce false positive and true negatives in some cases.
WebSSAR [21] : Check Input Validation against preconditioning	Good for the toy system.	Some preconditions may not be accurately expressed for some filters.

4. Dynamic Analysis Techniques

Table 5. Dynamic Analysis tools

Technique	**Advantages** [10]	**Drawbacks** [10]
SQLIMW, a middleware specifically designated for the detection of SQL injection attacks. [22]	Efficient, transparent to the programmer	Limited coverage, bet environment is a single-sign-on system
SVM for prediction of SQLIA [23]	Low overhead High detection rate	Produce false positive in some cases
TransSQL [24]	Applicable for a different platform, easy to deploy	High latency (because every query need to be executed twice)
Security testing schemes based on automatic test case generating and simulated tests. [25]	Automated, effective	The problem in coverage(attack library should be improved)
SQL-IDS (injection detection system) [26]	No need to change on code, low overhead, high coverage	Limited to java
Artificial Neural Network-based web application firewall for SQL injection. [27]	Independent in a matter of platform	Erroneous results may be produced.

There are other detection and prevention tools. The main evaluation criteria for tools are the success rate to detect SQLI attacks and latencies or overhead that are caused to the system by applying this method. Other factors that can be taken such as platform, language support, and code coverage.

A new classification is also done base on a system that used the techniques. The researchers in [10] classified the system into three subcategories and done analysis based on that.

5. Analysis

Based on the above overview and analysis we can say that there is no perfect and fully controlled technique to stop and prevent SQLI attacks. Many techniques are showing drawbacks in some languages and can't be used alone. During this section, the researcher will present an approach that may help to mitigate or stop SQLI Attacks on a web-based system. The main idea behind this approach to isolate the database server from the application. On the other hand rewrite the error message that are returned and hiding technical information that can be used by the hacker. Mainly the login form is connected directly to the database server. Adding a new verification field before displaying the login form will add value for checking. This field is not connected to the database. Whitelist values can be added to this field, if the value entered matches the value in the whitelist then you can proceed to log in form. Log in form contains username field and password field. The developer could add a whitelist on a username before continuing to enter the password and connecting to the database. Inside the application, when generating a report, for example, the user should input some data to the field and this generate query to retrieve data on the database side. Parse tree or tokenization should be applied to split the data that was entered and eliminate or stop the generation of the query if finding any of the reserved words using in the input fields.

The major challenge is the different structure of different languages used in developing web-based system. More analysis and researches should be done to test if this applies to all languages or not.

6. Conclusion and Future Work

This paper has provided an overview of SQLI attacks and their types. The paper has investigated different approaches and techniques used to prevent and detect SQLI attacks. The web-based system is increasing rapidly. Many databases were used to build web- based application. As seen from the above analysis, no single technique can guarantee 100 percent control to counter and detect SQLI attacks. Most problem attacks are in a stored procedure. This because of the nature of SQL queries and their dynamic. This dynamic could be changed during execution time. The best practice is in writing code and does best from a developer not to let vulnerabilities field for hackers to get in their database. This means the protection phase starts from writing code. To save time and cost using fuzzy testing can be a

good approach before putting the code on production. Another challenge is using different languages and different database systems in developing web-based systems. Each language has its privacy and weakness. There is no general approach that can be applied to all languages with the same efficiency. The described approach above tried to isolate the database server from the input form. Doing as much as a possible check before connecting the application to the system. This could be good to prevent SQLI attacks. More testing and investigating should be done to come up with exact results about that.

Also, more analysis and deep researches of the current approaches are needed to test and examine the different aspects of the techniques (e.g., detection rate, overhead, and false-positive rate, etc.). Efforts must also be placed on developing general criteria for the assessment of these approaches

Acknowledgment

The authors wish to thank Palestine Technical University-Kadoorie (PTUK) for supporting this research work as part of PTUK research fund.

References

[1] O. t. t. s. vulnerabilities, "OWASP," [Online]. Available: http://www.owasp.org/ index.php/Top_10. [Accessed 22 12 2020].
[2] "National Vulnerability Database," [Online]. Available: http://nvd.nist.gov. [Accessed 28 11 2020].
[3] K. Poulsen, "securityfocus," [Online]. Available: https://www.securityfocus.com/news/5968. [Accessed 5 12 2020].
[4] A. K. Ammar Alazab, "New strategy for Mitigating of SQL Injection Attack," International Journal of Computer Application, vol. 154, no. 11, November 2016, 2016.
[5] M. Muthuraja, "SQLIA Detection and Prevention using Parse Tree with Query Tokenization," in International Research Journal of Engineering and Technology (IRJET), Maharashtra, India, 2020.
[6] A. A. a. A. Khresiat, "New Strategy for Mitigating of SQL Injection Attack," International Journal of Computer Applications, vol. 154, pp. 1-10, 2016.
[7] B. A. a. E. O.-M. a. Z. Qin, "SQL injection attack detection using fingerprints and pattern matching technique," 2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS), pp. 583-587, 2017.
[8] M. F. Y. Zainab S. Alwan, "Detection and Prevention of SQL Injection Attack: A survey," International Journal of Computer Science and Mobile Computing, vol. 6, no. 8, August 2017, pp. 5-17, 2017.
[9] J. P. Singh, "Analysis of SQL Injection Detection Techniques," Theoretical and Applied Informatics, vol. 28, pp. 37-55, 2016.
[10] M. N. R. A. Mamdouh Alenezi, "SQL injection attacks countermeasures assessments," Indonesian Journal of Electrical Engineering and Computer Science, vol. 21, no. 2, pp. 1121-1131, Feb 2021.
[11] X. L. B. P. S. C. K. Q. a. L. T. Xiang Fu, " "A Static Analysis Framework For Detecting SQL Injection Vulnerabilities," Proc. of the 31st International Conf. on Computer Software and Applications, COMPSAC, vol. 07, pp. 87-96, 2007.
[12] R. M. a. P. Frankl, " "Preventing SQL Injection through Automatic Query Sanitization with ASSIST"," TAV-WEB (EPTCS), vol. 35, pp. 27-38, 210.
[13] L. W. a. T. X. S. Thomas, "On Automated Prepared Statement Generation to Remove SQL Injection Vulnerabilities," Information and Software Technology, vol. 51, no. 3, pp. 589-598, 2009.
[14] F. D. a. M. Sherriff, "Automated Fix Generator for SQL Injection Attacks"," Proc. of the 19th International Symposium on Software Reliability Engineering, pp. 311-312, 2008.
[15] P. G. K. J. a. M. E. A. Kieyzun, " "Automatic creation of SQL Injection and cross-site scripting attacks,"," Proc. of the IEEE 31st International Conference on Software Engineering, ICSE, 2009.
[16] a. G. A. R. Ezumalai, " "Combinatorial Approach for Preventing SQL Injection Attacks"," Proc. of the International Advance Computing Conference (IACC '09), IEEE Computer Society, pp. 1212-1217, 2009.
[17] R. A. M. a. I. H. Kruger, "SQL DOM: compile-time checking of dynamic SQL statements"," Proc. of the 27th International Conference on Software Engineering, ICSE'05, pp. 88-96, 2005.
[18] J. X. K. L. Y. Z. J. Y. W. Tian, "Research on mock attack testing for SQL injection vulnerability in multi-defense level web applications," Proc. of the 2nd International Conference on Information Science and Engineering (ICISE'10), pp. 1-5, 2010.
[19] D. M. a. G. V. F. Valeur, "A Learning-Based Approach to the Detection of SQL Attacks"," Proc. of the Conference on Detection of Intrusions and Malware Vulnerability Assessment (DIMVA), Vienna, Austria, 2005.
[20] L. W. G. W. D. Z. a. Y. Y. X. Wang, "Hidden web crawling for SQL injection detection," the 3rd IEEE International Conference on Broadband Network and Multimedia Technology (IC-BNMT'10), pp. 14-18, 2010.
[21] P. K. S. M. A. Harish Dehariya, "A Survey on Detection and Prevention Techniques for SQL Injection Attack," International Journal of Wireless and Microwave Technologies(IJWMT), vol. 6, no. 6, pp. 72-79, 2016.
[22] C. M. X. a. J. M. G. Jiao, "SQLIMW: A New Mechanism against SQL-injection,"," International Conference on Computer Science & Service System, pp. 1178-1180, 2012.
[23] R. R. a. S. Shrivastav, " "SQL injection attack Detection using SVM," " International Journal of Computer Applications", pp. 1-4, 2012.
[24] C. L. S. C. Y. H. H. H. a. F. H. K. Zhang, " "TransSQL: A Translation and Validation-based Solution for SQL-Injection Attacks," Proc. of the First International Conference on Robot, Vision and Signal processing, pp. 248-251, 2011.
[25] Y. H. a. N. Zhihong, "A database security testing scheme of the web application"," Proc. of the 4th International Conference on Computer Science & Education, no. 09, pp. 953-955, 2009.
[26] K. K. a. T. Tzouramanis, "SQL-IDS: A Specification-based approach for SQL-injection Detection," Proc. of the 2008 ACM symposium on Applied Computing, 2008.

[27] M. Asaad, "Artificial Neural Network based Web Application Firewall for SQL Injection," World Academy of Science, Engineering & Technology., vol. 64, pp. 12-21, 2010.
[28] R. J. J. V.nithya, "A survey on SQL Injection attacks, their Detection, and Prevention Techniques," International Journal oF Engineering and Computer Science ISSN: 2319-7242, vol. 2, no. 4 April 2013, pp. 886-905, 2013.
[29] C. M. X. a. J. M. G. Jiao, "SQLIMW: A New Mechanism against SQL-injection,".

Authors' Profiles

Hazem Harb: Bachelor Degree in Computer Science from UAEU in 2003. At present, he is pursuing his Master of Science in Cybercrimes & Digital Evidence Analysis at Palestine Technical University- Kadoorie.

Professor Derar Eleyan currently works as the Vice President, Palestine Technical University- Kadoorie. He researches Software Engineering and Information Systems (Business Informatics). Their most recent publication is 'Conceptualizing a Model for the Effect of Information Culture on Electronic Commerce Adoption'.

Amna Eleyan is working as a Lecturer at Manchester Metropolitan University, Department of Computing and Mathematics. She was rewarded her PhD in Software Engineering in the field of Web Services from the University of Manchester. She is a Fellow member of the Higher Education Academy 'HEA'. Her research interests include Telecommunications and Computer Networks, Distributed Systems, Web Services, Internet of Things 'IoT' applications in smart home and smart healthcare, Block chain, Chaos theory for image encryption and privacy preservation in Vehicular Ad Hoc Networks 'VANET'. She has participated at the International Symposium on Networks, Computers and Communications, ISNCC 2016/17/18/19/20, as a Publicity chair, a chair of the "PhD Student Forum and a chair of the track 'Grid and Social Computing'.

How to cite this paper: Hazem M. Harb, Derar Eleyan, Amna Eleyan, " SQL Injection Detection Tools Advantages and Drawbacks", International Journal of Wireless and Microwave Technologies(IJWMT), Vol.11, No.3, pp. 16-21, 2021.DOI: 10.5815/ijwmt.2021.03.03