

PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH



MOHAMED BOUDIAF UNIVERSITY - M'SILA
FACULTY OF MATHEMATICS AND
COMPUTER SCIENCE

COMPUTER SCIENCE DEPARTMENT



DISSERTATION

**Submitted in partial fulfillment of the requirements for the Degree
of MASTER**

Domain: Mathematics and Computer Science

Branch: Computer Science

Specialty: Information Technology and Communication

By: DRIHEM ABDALLAH

TOPIC

New Detection Method for SQL injection

Publicly defended: 31 / 06 /2016 before a Jury composed of :

Chikouche
S.Bouhouita
M.Kamel

University of M'sila
University of M'sila
University of M'sila

President
Supervisor
Examiner

Academic Year: 2015 /2016

CONTENTS

INTRODUCTION GENERAL	1
INTRODUCTION GENERAL	2
CHAPITRE I - WEB SECURITY	3
1. Introduction	4
1.1. What is security	5
1.2. The foundations of security.....	6
1.3. Threats, vulnerabilities, and attacks defined.....	7
1.4. How to implement safety assessment	7
1.4.1. Asset Classification	8
1.4.2. Threat analysis.....	10
1.4.3. Risk analysis	10
1.4.4. Design of security programs	13
2. Web security.....	13
2.1. Targets and impacts of web attacks	13
2.2. The web application	14
2.3. 96% of Tested Applications Have Vulnerabilities (Report of CENZIC).....	14
2.4. Top 10 OWASP	15
2.5. Understanding the dangers of insecure web applications	16
2.6. Rise of web security	18
2.7. Improving Security in Web Applications	19
2.8. Web application security lifecycle (S-SDLC).....	19
2.8.1. Secure development.....	20
2.8.2. Secure deployment.....	20
2.8.3. Secure operation	21
2.9. Common detectable application vulnerabilities.....	21
3. Conclusion	22
CHAPITRE II - SQL INJECTION	23
1. Introduction	24
2-Understanding SQL Injection	24
3-Understanding How It Happens	27
3.1. Example of Incorrectly Handled Escape Characters	27
4 - Finding SQL Injection	28
4.1. Testing by Inference	28

4.2. Identifying data entry	29
4.3. GET Requests	29
4.4. POST Requests	30
4.5. Browser modification extensions	30
4.6. Proxy servers	31
4.7. Workflow of Web request and Web response.....	31
4.8. Database Errors	32
5. Differentiating Numbers and Strings	33
5.1. Example:	33
6. Inline SQL Injection.....	33
6.1. Injecting Strings Inline	34
6.2. Injecting Numeric Values Inline	35
7. Terminating SQL Injection	35
7.1. Database Comment Syntax.....	36
7.2. How Using Comments to terminate SQL statements	36
8. Conclusion.....	38
CHAPITRE III - THE APPROACH PROPOSED	39
1. Introduction	40
2. Definition of SQLIA	40
3. SQL Injection Attacks (SQLIA) Process	40
4. Types of SQLIA.....	41
4.1. Tautologies	41
4.2. Piggy-backed Query	41
4.3. Logically Incorrect.....	42
4.4. Union query	42
4.5. Stored Procedure.....	43
4.6. Inference	43
4.7. Alternate Encodings.....	44
5. The proposed approach	45
6. The method – (Method Detector)	45
6.1. Example	46
6.2. Queries	46
6.3. Links.....	46
6.4. Results of the test	47
6.5. Analyze of the results.....	47
7. Related work	47

7.1. SWADDLER	47
7.2. SQL Prevent	47
7.3. SQL Lrand	48
7.4. SAFELI	48
7.5. SQLIPA	48
7.6. SQLCheck	48
7.7. IDS (Intrusion Detection Systems)	49
7.8. Detection based on removing SQL query attribute values	49
7.9. DIWeDa	49
7.10. Context Sensitive String Evaluation (CSSE)	50
7.11. CANDID	50
7.12. Automated Approaches	50
7.13. AMNESIA	51
8. Comparison of SQL Injection detection techniques with respect to attack types	51
9. Percentage of SQL Injection Detection Techniques	52
10. Conclusion	53
CHAPTER IV ANALYSIS, RESULTS AND EXPERIMENTAL	54
1. Introduction	55
2. Platform	55
2.1. Visual Studio	55
2.2. C#	55
2.3. XAMPP	55
2.4. PHP	56
2.5. MySQL	56
3. Experimentation	56
4. The Sites of experimentation	56
4.1. The BTS Lab	56
4.1.1. Presentation	56
4.1.2. Architecture	57
4.2. TEST Site	57
4.2.1. Presentation	57
4.2.2. Architecture	57
4.3. Buggy Web Application (bWAPP)	58
4.3.1. Presentation	58
4.3.2. Architecture	58
5. The scanners used in the experiment	58

5.1. Netsparker	58
5.2. OWASP ZAP 2.4.2	58
6. Experiment of the Method	59
6.1. bWAPP	59
6.1.1. Queries	59
6.1.2. Links (bWAPP)	59
6.1.3. Method	59
6.1.4. Test method with queries.....	59
6.1.5. Bwapp – SQL Injection – Bypass Authentication.....	60
6.1.5.1 Queries	60
6.1.5.2 Links	60
6.1.5.3 Method	60
6.1.5.4 Test method with queries	60
6.1.6 Bwapp – SQL Injection – Blind – Boolean- Based.....	61
6.1.6.1 Queries	61
6.1.6.2 Links	61
6.1.6.3 Method	61
6.1.6.4 Test method with queries	61
6.2. BTS Lab Site	62
6.2.1 Queries	62
6.2.2 Links.....	62
6.2.3 Method	62
6.2.4 Test method with queries.....	62
7. Comparison the method with different scanner	63
7.1. List of the URL (TEST Site)	63
7.2. Low Security.....	63
7.3. Medium Security.....	64
7.4. High Security	65
7.5 The result of the vulnerability (12 page vulnerable).....	66
7.6 The result of False positive/the vulnerability (12 page vulnerable).....	66
8. Conclusion	67
CONCLUSION GENERAL.....	68
CONCLUSION GENERAL	69
REFERENCES	70

INTRODUCTION GENERAL

SQL injection is the vulnerability That results when you give an attacker the ability to influence the Structured Query Language (SQL) queries that an application passes to a back-end database. By being able to influence what is passed to the database, the attacker can leverage the syntax and capabilities of SQL itself, as well as the power and flexibility of supporting database functionality and operating system functionality available to the database. SQL injection is not a vulnerability that exclusively affects Web applications; any code that accepts input from an untrusted source and then uses that input to form dynamic SQL statements could be vulnerable (e.g., “fat client” applications in a client/server architecture).

A lot of works have been engaged to remedy to SQL attacks, the goal is to build a scanner that can reveal the vulnerability included in the web sites. Several approaches are proposed; they can be categorized. In our research we focus on the structure of the web page rather than the content, so, we proposed a method based on the variation of the number of the tags between the original page and the resulted one after being injected by an SQL attack.

The experiments realized to evaluate the proposed detection SQL injection method is done on web sites proposed by the researchers' community in the area. Three scanners are considered in order to appreciate the performance of our scanner. The results reveal a better performance.

The dissertation is structured as follow, In the first chapter we have seen a view in web security is presented and the major threats are discussed, In the second chapter a view in SQL injection, what is SQL injection, understanding SQL Injection, how it happens and more things about SQL injection is a full and comprehensive definition, In the third chapter we discuss the proposed approach is motivated and explained and all SQL Injection various types of attacks are explained, and a survey of the famous detection methods are listed, In the last chapter is like a series of experiments are done to evaluate the performance of the method.

CONCLUSION GENERAL

SQL injection is a serious threat with deep dangerous consequences on the integrity and security of web applications. In our work, we proposed a method designed upon the structure of the web page cached by its number of tags. The detection is based on the statement that “the behavior of any SQL attack changes the number of tags from the original page to the injected one”.

This statement reveals being a good one, because the detection performance is highly appreciated over the well-known scanners used by the community of researchers and professionals. a series of experiments have shown this fact.

[4] Justin Clarke, SQL Injection Attacks and Defense, Rachel Rounselotti , July 2, 2012

Journals and Survey:

[5] Purbhar Y Janc , M.S.Chauhan, SQLIA: Detection And Prevention Techniques A Survey, iotjournals, 2, 2013, pp. 56-60

[6] V. Nithya, E.Regan , L.vijayaraghavan, A Survey on SQL Injection attacks, their Detection and Prevention Techniques, iotjournals, Volume 2, April, 2013 pp. 886-903

Guide:

[7] The Ten Most Critical Web Application Security Risks, OWASP, 2013

[8] A Guide to Building Secure Web Applications and Web Services, OWASP

[9] Web Application Security a Beginner's guide, Bryan Sullivan Vincent Liu,

Report:

[10] Application Vulnerability Trends Report, CEN2IC, 2014

Website:

[11] OWASP, www.owasp.org

REFERENCES

Books :

[1] Hanqing Wu and Liz Zhao, **Web Security: A WhiteHat Perspective**, by Auerbach Publications, April 6, 2015

[2] John Wiley & Sons, Ltd, **Web Application Security for Dummies**, by Qualys, West Sussex PO19 8SQ England, 2011.

[3] Michael E. Whitman, Herbert J. Mattord, **Principles of Information Security**, Kennesaw State University, 20 Channel Center Boston, MA 02210 USA, 2012.

[4] Justin clarke, **SQL Injection Attacks and Defense**, Rachel Roumelioti , July 2, 2012.

Journals and Survey:

[5] Pushkar Y.Jane , M.S.Chaudhari, SQLIA: Detection And Prevention Techniques: A Survey, iosrjournals, 2, 2013, pp. 56-60.

[6] V. Nithya,R.Regan , J.vijayaraghavan, A Survey on SQL Injection attacks, their Detection and Prevention Techniques, iosrjournals, Volume 2, April, 2013 pp, 886-905

Guide:

[7] **The Ten Most Critical Web Application Security Risks**, OWASP, 2013

[8] **A Guide to Building Secure Web Applications and Web Services**, OWASP.

[9] **Web Application Security a Beginner's guide**, Bryan Sullivan Vincent Liu,

Rapport:

[10] **Application Vulnerability Trends Report**, CENZIC, 2014

Website:

[11] OWASP, www.owasp.org.

BIBLIOGRAPHY

[12] OWASP Zap Scanner, 2015

[13] Netsparker Web Application Security Scanner, Version 3.5.

[14] BTS Pentesting Lab Site.

[15] bWAPP Site.

Abstract

In this work we propose a new algorithm for detecting SQL injections in web applications, which is a serious and dangerous issue. The proposed method considers the structure of the web page and particularly the number of its tags. A series of experimentations are done to valid the performance of "T-Scan" the scanner built upon the proposed method. The results confirm its high detection performance over the famous known scanners.

Résumé

Dans ce travail, on propose un nouvel algorithme pour la détection des injections SQL au sein des applications web, qui est un sujet délicat et dangereux. La méthode proposée considère la structure de la page web et particulièrement son nombre de tags. De sérieuses expérimentations sont réalisées afin de valider la performance de « T-Scan » qui est le scanner conçu à partir de la méthode proposée. Les résultats confirment sa performance de détection par rapport aux scanners célèbres connus.

ملخص

في هذا العمل، نقترح خوارزمية جديدة للكشف عن حقن SQL في تطبيقات الويب، والتي هي قضية حساسة وخطيرة. تعتبر الطريقة المقترحة على هيكل صفحة على الوأب، وخاصة في عدد من العلامات. أجريت تجارب للتحقق من أداء "Tscan" الذي هو كاشف يعمل على هذه الخوارزمية المقترحة. النتائج تؤكد أداء الكاشف مقارنة مع الكواشف العالمية المعروفة.