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

جامعة محمد بوضياف - المسيلة
Université Mohamed Boudiaf - M'sila

Order N: .....

UNIVERSITY OF M'SILA

FACULTY OF MATHEMATICS AND COMPUTER SCIENCE

Information Communication Sciences and Technology Department

Dissertation submitted in partial fulfillment of the requirements for the Degree of MASTER

Domain: Mathematics and Computer Science
Branch: Computer Science
Specialty: Information Communication and Technology

By: BOUKAROUI HADJER

TOPIC

An information theoretic approach to detect SQLI Intrusion

Publicly defended: / /2015 before a Jury composed of:

Ms.SAOUDI LALIA	**University of M'sila**	**Supervisor**
Dr.L.Belabdelouahab-Fernini	**University of M'sila**	**Supervisor**
.....	**University of M'sila**	**Chair**
.....	**University of M'sila**	**Examiner**
.....	**University of M'sila**	**Examiner**

Academic Year: 2014 /2015

Table of content

General introduction

1.1. Context

The people who say they know about SQLIA, they knew and saw simple examples of what this type of attack can do, but they probably did know that SQLIA is one of the most devastating vulnerability that can infect a business. It can also steal a sensitive information, secure data, like a credit card number, personnel information. It can also lead to gain an administrative access to an application and manipulate it as they want.

1.2. Statement of the Problem:

SQL injection attacks pose a serious security threat to Web applications, they allow attackers to obtain unrestricted access to the databases underlying the applications and to the potentially sensitive information these databases contain. Although researchers and practitioners have proposed various methods to address the SQL injection problem, current approaches either fail to address the full scope of the problem or have limitations that prevent their use and adoption. Many researchers and practitioners are familiar with only a subset of the wide range of techniques available to attackers who are trying to take advantage of SQL injection vulnerabilities. As a consequence, many solutions proposed in the literature address only some of the issues related to SQL injection.

Because of that, developers go to implement SQL injection detection system (SQLIDS). The system monitors web-based applications and detects SQL injection attacks in real time. We report some preliminary experimental results over several SQL injection attacks that show that the proposed query-specific detection allows the system to perform focused analysis at negligible computational overhead without producing false positives or false negatives. Therefore, the new approach is very efficient in practice.

1.3. Objectives

1.3.1. General Objectives

The subject of this work is to develop a prototype tool that detects SQLIA using the information theoretic approach. The mean goal is to improve the detection of the new SQLIA's.

1.3.2. Specific Objectives

The intent of this work is to improve the implementation of Hossain Shahriar's approach that detects SQLIAs using the information theory. Our goals are:

- To improve the extraction and the detection of SQLIA to a dynamic way, without need to instrument the source code.
- To make this application work on several applications that was written in different programming language, which means answering this question:
  - How to extract the SQL query from the source code
  - How to manage the way this queries are written (format, space…..)
  - How to calculate the number of tokens in each query
  - How to calculate the entropy of every query

1.4. Methodology

Our implementation of this detection tool is divided in two phases: static and dynamic phases.

In the static one, we extract the queries from the dynamic pages which mean a problem of getting the dynamic pages, then, calculate the number of its tokens, which put's the problem of the way the query was written with. To resolve this problem, we made an SQL parser.

Then, we calculate the number of repetition of every single token, and the entropy. In the dynamic phase, which means when the query executes, we got the inputs of the user and we place them in the query and recalculate the entropy. Then we made a comparison between the first and the second value of the entropy, if it has been decreased or increased, then, we deal with the query that contains an injection.

1.5. Report Outline

This work divided in four chapters:

1.5.1. Chapter 01

The first chapter contains everything about SQLIAs; we present the seven types of injection attacks based on the mechanisms that used and the intents of the attacks, every type with an example and an explanation.

1.5.2. Chapter 02

In the second chapter, we summarize how to make a web application secure against SQLIA, using IDSs and its technologies, Then, we give the most common technologies developed to detect and prevent the SQLIA. And at the end, we list the approaches that used the theoretic information as a base of their security work.

1.5.3. Chapter 03

In the third chapter we define the information theoretic and its different concepts specially entropy which is the basic element of it, then, we define the mathematical formula to calculate it.

The second part of this chapter is about the application of "the detection of SQLIA's based on information theoretic "approach

1.5.4. Chapter 04

In the fourth chapter we are going to explain more about the implementation of our tool using this approach. First of all, this java tool is implemented using eclipse, which is an integrated development environment for java. And we developed a dynamic website we called "security group", as a security group that inform the users about its different products, as well as everything new in the security field. This web site is developed for the purpose of testing the SQLIA's and the SQLI detector and we used the Dreamweaver for the design, and the Notepad for the code.

After that we define the different packages and programming languages, as well as the implementation of the web site and java application

1.5.2. Chapter 02

In the second chapter, we summarize how to make a web application secure against SQLIA, using IDSs and its technologies, Then, we give the most common technologies developed to detect and prevent the SQLIA. And at the end, we list the approaches that used the theoretic information as a base of their security work.

1.5.3. Chapter 03

In the third chapter we define the information theoretic and its different concepts specially entropy which is the basic element of it, then, we define the mathematical formula to calculate it.

The second part of this chapter is about the application of "the detection of SQLIA's based on information theoretic "approach

1.5.4. Chapter 04

In the fourth chapter we are going to explain more about the implementation of our tool using this approach. First of all, this java tool is implemented using eclipse, which is an integrated development environment for java. And we developed a dynamic website we called "security group", as a security group that inform the users about its different products, as well as everything new in the security field. This web site is developed for the purpose of testing the SQLIA's and the SQLI detector and we used the Dreamweaver for the design, and the Notepad for the code.

After that we define the different packages and programming languages, as well as the implementation of the web site and java application

General Conclusion

Despite secure programming practices and many proactive countermeasures available, most of the web-based applications still suffer from SQLI vulnerabilities. Exploitations of SQLI vulnerabilities result in unwanted program behaviors and loss and leakage of confidential information of end users. Thus, SQLI mitigation needs to be considered seriously. This work detected SQLI attacks using an information-theoretic approach. We defined entropy as a measure to understand the complexity of the static query written by programmers. When a maliciously input successfully alters the static nature of the query, the complexity value changes. Thus, we rely on comparing the statically computed entropies with that of dynamically computed entropies. The deviation indicates the presence of SQLI in a query. The approach has been found to be effective for a set of vulnerable programs implemented in PHP. The approach allowed us to identify unknown SQLI attacks.

Our approach comprised two phases: In the static one, we explained how we extracted the queries and how we calculated the number of their tokens, the number of repetition of every single token, and the entropy. In the dynamic phase, which means when the query executes, we explained how we got the inputs of the user, placed them in the query and recalculated the entropy. Then we made comparison between the first and the second value of the entropy, whether it has been decreased or increased, then, the query contained an injection.

Actually, we did implement SQL injection detector based on this approach using the java language. We did test it in a web site that we did implement and the application has been effective on the detection of SQLIA's

Our future work includes testing the application on several web applications. We also plan to apply our developed to detect other web-based applications written in other programming languages.

Bibliography

[1]. **Justin, Clarke.** *SQL Injection Attacks and Defense Second Edition.* USA : Elsevier, 2012. 978-1-59749-963-7

[2]. **Tamada, Srinivas.** sql-injection.html. *9lessons.info.* [En ligne] 15 December 2008. [Citation : 02 janvier 2015.] http://www.9lessons.info/2008/12/sql-injection.html.

[3].a Survey on SQL Injection attacks, their Detection and Prevention Techniques. **Nithya, V and all.** 4, s.l. : International Journal Of Engineering And Computer Science, 4 April, 2013, Vol. 2. 2319-7242.

[4]. **Rouabah, Abdelbasset, Elyass, Kacem et Sid, Mohammed.** *sql injection and svs "simple vulnerability scanner".* alger : s.n.

[5].William G.J., Halfond, Jeremy, Viegas and Orso, Alessandro. *A Classification of SQL Injection Attacks.* Georgia : College of Computing Georgia Institute of Technology, 2006.

[6]. **Wu, Tzeyoung Max.** *Intrusion detection systems.* s.l. : Information Assurance Technology Analysis Center (IATAC), 25-09-2009. 0704-0188.

[7]. **SANS Institute.** *Understanding Intrusion Detection Systems.* s.l. : SANS Institute Reading Room site, 2001.

[8]. **Carl Gould, Zhendong Su, and Premkumar Devanbu.** *JDBC Checker: A Static Analysis Tool for SQL/JDBC Applications.* California : Department of Computer Science University of California, Davis.

[9].Gary Wassermann, Zhendong Su. *An Analysis Framework for Security in Web Applications.* s.l. : Department of Computer Science University of California, Davis.

[10]. *AMNESIA: Analysis and Monitoring for NEutralizing SQL Injection Attacks.* **Orso, W. Halfond and A.** Long Beach, California, USA. : In Proceedings of the 20th IEEE/ACM International Conference on Automated Software Engineering, November 7–11, 2005.

[11]. *The Essence of Command Injection Attacks in Web Applications.* **Wassermann, Z. Su and G.** Charleston, South Carolina : In Proceedings of the 33rd Symposium on Principles of Programming languages, January 2006.

[12]. *Securing Web Application Code by Static Analysis and Runtime Protection.* **all, Yao-Wen Huang and.** New York, USA. : ACM, May 17–22, 2004. 1-58113-844-X/04/0005.

[13]. *Automatically Hardening Web Applications Using Precise Tainting information.* **all, Nguyen-Tuong and.** Virginia,USA : In Twentieth IFIP International Information Security Conference (SEC 2005)., May 2015. CCR-0092945.

[14]. *Defending Against Injection Attacks through Context-Sensitive String Evaluation.* **Berghe, T. Pietraszek and C. V.** s.l. : In Proceedings of Recent Advances in Intrusion Detection (RAID2005), 2005.

[15]. *Dynamic Taint Propagation for JAVA.* **V. Haldar, D. Chandra, and M. Franz.** s.l. : Proceedings 21st Annual Computer Security Applications Conference, Dec. 2005.

[16]. **Livshits, Benjamin, Martin, Michael et L, Monica S.** *securifly: Runtime Protection and Recovery from Web Application Vulnerabilities.* s.l. : Stanford University, September 22, 2006.

[17]. *Compile Time Checking of Dynamic SQL Statements.* **McClure, R. et Kruger, I.** pages 88–96, St. Louis, Missouri, USA : In Proceedings of the 27th International Conference on Software Engineering (ICSE 05), May 15–21, 2005.

[18]. *Safe Query Objects: Statically Typed Objects as Remotely Executable Queries.* **Cook, W. R. et Rai, S.** s.l. : In Proceedings of the 27th International Conference on Software Engineering (ICSE 2005), 2005.

[19]. *Abstracting Application-level Web Security.* **Scott, D. et Sharp, R.** pages 396–407, s.l. : In Proceedings of the 11th International Conference on the World Wide Web (WWW 2002), 2002.

[20]. *SQLrand: Preventing SQL injection attacks.* **Boyd, S. W. et Keromytis, A. D.** pages 292–302, s.l. : In Proceedings of the 2nd Applied Cryptography and Network Security (ACNS) Conference, June 2014.

[21]. *An Approach to Detection of SQL Injection Attack Based on Dynamic Query Matching.* **Das, Debasish, Sharma, Utpal et Bhattacharyya, D.K.** 25, s.l. : International Journal of Computer Applications, 2010, Vol. 1. 0975 - 8887.

[22]. *Rule based Detection of SQL Injection Attack.* **Das, Debasish, Sharma, Utpal et Bhattacharyya, D. K.** 19, s.l. : International Journal of Computer Applications, April 2012, Vol. 43. 0975 – 8887.

[23]. 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.

[24]. **Kemalis, Konstantinos et Tzouramanis, Theodoros.** SQL-IDS: A Specification-based Approach for SQL-Injection Detection. Brazil : Department of Information & Communication Systems Engineering,University of the Aegean,, 2008.

[25].S. Khayam, ''Worm Detection at Network Endpoints Using Information-Theoretic Traffic Perturbations,'' *Proc. of the IEEE International Conference on Communications (ICC)*, Beijing, China, May 2008, pp. 1561 - 1565.

[26].W. Yu, X. Wang, D. Xuan, and D. Lee, ''Effective Detection of Active Worms with Varying Scan Rate,'' Proc. of IEEE Securecomm and Workshops, Baltimore, MD, USA, August 2006, pp. 1-10.

[27] W. Lee and D. Xiang, ''Information-Theoretic Measures for Anomaly Detection,'' Proceedings of the IEEE Symposium on Security and Privacy, Oakland, California, May 2001, pp.130-143.

[28].G. Gu, P. Fogla, D. Dagon, W. Lee, and B. Skoric, ''Towards an Information-Theoretic Framework for Analyzing Intrusion Detection Systems,'' Proceedings of the 11th European Symposium Research Computer Security (ESORICS 2006), Hamburg, Germany, September 2006, pp. 527-546.

[29]. Information-Theoretic Detection of SQL Injection Attacks. **Shahriar, Hossain et Zulkernine, Mohammad.** 40-47, Canada : School of Computing Queen's University, Kingston, Canada in procceding IEEE 14th International Symposium on High-Assurance Systems Engineering, 2012. DOI 10.1109/HASE.2012.31.

[30]. **Desurvire, Emmanuel.** *Classical and Quantum Information Theory.* Cambridge : Cambridge University Press, 2009. 9780521881715.

[31]. **Stylesheet, Author.** *Information Theory.* University College London United Kingdom : ENCYCLOPEDIA OF COGNITIVE SCIENCE, 2000.

[32]. **Lowe, Doug.** *Java™ All-in-One Desk Reference For Dummies.* canada : Wiley Publishing, Inc., 2005. ISBN-13: 978-0-7645-8961-4.

[33]. *CREATING WEB PAGES WITH Dreamweaver.* s.l. : For additional help,UNC Pembroke, 2001.

[34]. **Edited by The HSQL Development Group, Blaine Simpson, and Fred Toussi.** *HyperSQL User Guide.* s.l. : The HSQL Development Group, 2002. 2011-03-13 22:44:24 -0400.

[35]. jsoup-parser-simplement-du-html-en-java.html. *eric-pidoux.* [En ligne] 25 juillet 2012. [Citation : 29 05 2015.] http://www.eric-pidoux.com/actu/jsoup-parser-simplement-du-html-en-java.

[36]. **Norvell, Theodore S.** *The JavaCC FAQ.* s.l. : Computer and Electrical Engineering Memorial University of Newfoundland, June 19, 2007.

ABSTRACT

SQL Injection (SQLI) is a widespread vulnerability commonly found in web-based programs. Exploitations of SQL injection vulnerabilities lead to harmful consequences such as authentication bypassing and leakage of sensitive personal information. Therefore, SQLI needs to be mitigated to protect end users. In this work, we present an approach to detect SQLI attacks based on information theory. We compute the entropy of each query present in a program accessed before program deployment. During the program execution time, when an SQL query is invoked, we compute the entropy again to identify any change in the entropy measure for that query. The approach then relies on the assumption that dynamic queries with attack inputs result in increased or decreased level of entropy. In contrast, a dynamic query with benign inputs does not result in any change of entropy value.

Keywords: SQL injection, software vulnerability, information theory, entropy.

Résumé

L'Injection SQL (SQLI) est une vulnérabilité répandue trouvée couramment dans les programmes basés sur le Web. Les exploitations de vulnérabilités d'injection SQL mènent à des conséquences néfastes telles que l'authentification contournement et la fuite d'informations personnelles sensibles. Par conséquent, SQLI doit être atténuée afin de protéger les utilisateurs finaux. Dans ce travail, nous présentons une approche pour détecter les attaques de SQLI basées sur la théorie de l'information. Nous calculons l'entropie de chaque requête présente dans un programme consultée avant le déploiement du programme. Au cours du programme le temps d'exécution, lors d'une requête SQL est invoqué, nous calculons à nouveau l'entropie pour identifier tout changement dans la mesure de l'entropie pour cette requête. L'approche repose donc sur l'hypothèse que les requêtes dynamiques avec des entrées d'attaque entraînent une augmentation ou une diminution du niveau de l'entropie. En revanche, une requête dynamique avec des entrées bénignes ne conduit pas à un changement de la valeur de l'entropie.

Mots-clés: injection SQL, vulnérabilité du logiciel, théorie de l'information, l'entropie.

ملخص

ال SQL Injection (SQLI) هو عبارة عن واحدة من أكثر الثغرات انتشارا والتي نجدها في برامج Web ، استغلال هذه الثغرات يؤدي إلى عواقب وخيمة مثل تجاوز المصادقة و تسريب المعلومات الشخصية الحساسة. لذلك يجب تعديل SQLI لحماية المستخدمين النهائيين. في هذا العمل نقدم طريقة لكشف ال SQLIA's بالاعتماد على information theory ،نقوم بحساب ال entropy الخاص بكل query يظهر في التطبيق ، وحين إستدعاءه نقوم بحساب ال entopy مرة أخري لتحديد التغير في القيمة، ذلك على حساب أن إدخال المعلومات التي تحتوي على SQLI سيتسبب في تزايد أو تناقص ال entropy بنسبة كبيرة، و إدخال المعلومات الصحيحة لن يتسبب في أي تغير.

الكلمات المفتاحية: SQL injection ، entopy ، Web ،SQLIA ، information theory ، query .