International Journal of Education and Science Research Review

E-ISSN 2348-6457

Volume-9, Issue-4 July-August- 2022

P-ISSN 2349-1817

www.ijesrr.org Email- editor@ijesrr.org

DETECTION AND PREVENTION OF WEB APPLICATION FROM SQLI AND XSS

Mrs. M. Angelin Rosy
Assistant Professor
Master of Computer Applications
Er.PerumalManimekalai College of Engineering
angel_rosym@yahoo.co.in
9944579754

P. Chaithra
II MCA
Master of Computer Applications
Er.PerumalManimekalai College of Engineering
chaithranaidu301@gmail.com
9500933479

ABSTRACT:

In the global age of information web applications arethe backbone of the village. In this day it is the era of computers which technology in which a world becomes onevillage. In this day it is the era of computers which interconnects various business organizations which uses the internet for their financial transactions, educational endeavors, and countless other activities. Design Science Research Methodology and A rigorous survey have been conducted and consequently, comparative analysis of various detection and prevention techniques is done with respect to various types of attacks.in current research various Hashing algorithm for the detection and prevention of SQLIA are analyzed and few tested. From the survey of various papers, it is found that the SQL Injection and Cross-site Scripting (XSS) attacks are most powerful in today's web application. Hence, the big challenge became to secure such a website against attack via the Internet. The issue of SQLIA still exists and has become a giant online threat to many companies who became a victim of SQLIA vulnerability and cost them a lot of money. The main purpose of this study was designing improved hashing algorithm for Detecting and Preventing SQLIA & XSS approach has been implemented successfully and fully able to fix SQLIA and XSS vulnerabilities. SHA512 hashing algorithm is used to build a good, secure cryptographic hash function by developing a good compression function in which each input bit affects as many output bits as possible. The results obtained by the implementation and evaluation are measured in number of test cases that produce more robust and reliable SQLIA and XSS prevention mechanism.

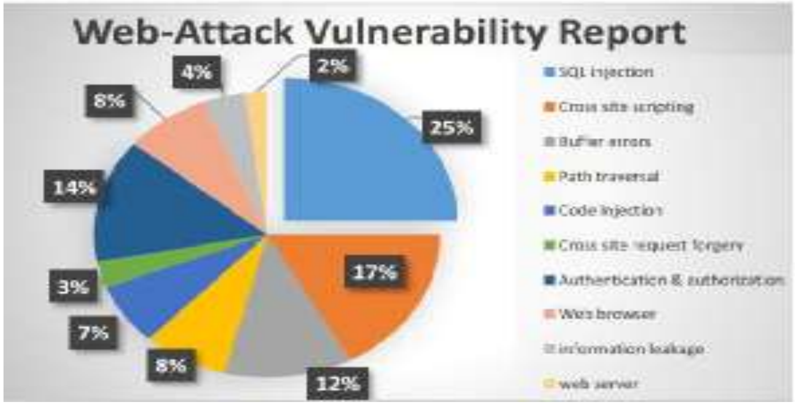
Keywords: Web Application Security, SQL Injection, SQLIA, XSS, Vulnerabilities, Hashing Technique.

I. INTRODUCTION:

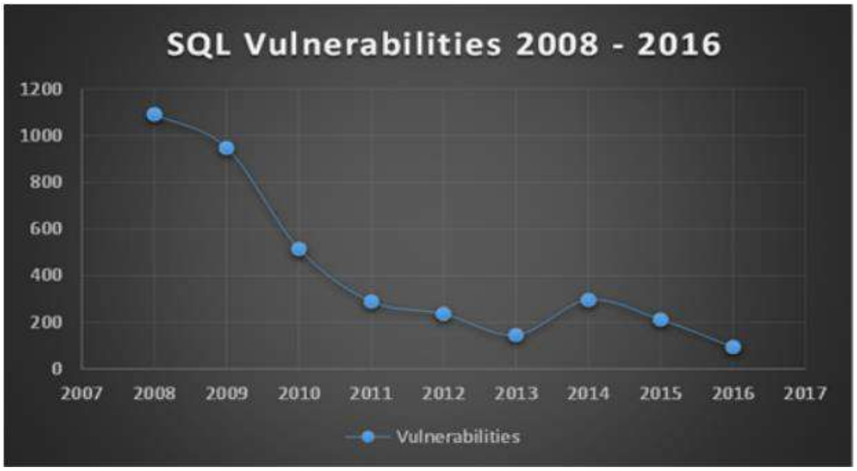
The Internet and web applications are the modern day workplace and business ground contributing to driving the world economy. At the same time, hackers and attackers have developed a parallel underground economy of hacking into web applications and stealing a large amount of sensitive business-critical information with malicious intent. Among the various security threats a web application is exposed to, SQL Injection Attack (SQLIA) and cross-site scripting (XSS) has taken the forefront. It has prevailed as a popular attack method. Using a SQL injection attack, an attacker can extract, modify or destroy the back end database of web applications. The simplicity of attacking a web application using SQL injection and abundance of vulnerable applications on the Internet has largely contributed to widespread data breach incidents.

II. LITERATURE REVIEW:

The first ever public disclosure of SQL injection attack vulnerability was documented by Jeff Forristal under the alias rain.forest.puppy (RFP) in a hacker ezine named Phrack Magazine released on December25, 1998. The relative prevalence of some of these attacks is shown in the figure below. This data was obtained from HACKING & TRICKS, a blog about Hacking &Computer Security by Nirav Desai, in an entry from January 8th, 2013. Clearly, SQL Injection comprises a significant fraction (approximately 25%) of all web-based attacks.

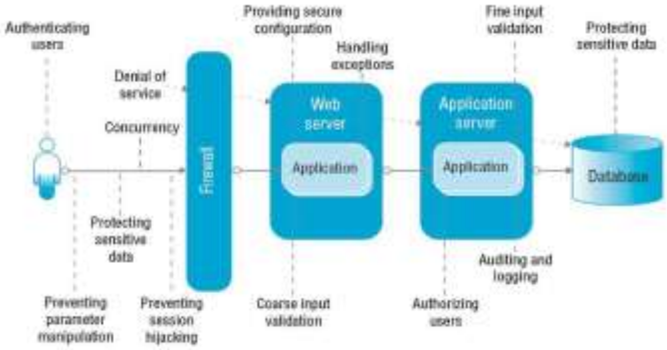


We understand from the following figure researches are conducted in every year and the problem is very severity in web technology. The National Vulnerability Database (‘NVD’) is a database managed by the National Institute of Standards and Technology (‘NIST)’. It is a database containing vulnerabilities that are based on the Common Vulnerabilities and Exposures (‘CVE’) dictionary[22,23] this shows that That number has fortunately dropped over the last few years. However, vulnerabilities continuously get reported and the problem persists even with newly developed applications.



A. Web application and Vulnerabilities:

Most of the current web applications use RDBMS (Relational Database Management Systems). Sensitive information like credit card, social security, and financial records are stored in these databases. Usually, programmers who write these programs are unaware of a technique for writing secure code. They would focus on implementing desired functionalities andwould focus less on security aspects. They would focus onimplementing desired functionalities and would focus lesson security aspects. This results in vulnerabilities in webapplications. Vulnerabilities allow an attacker to target thisweb application and obtain valuable information. Attackerswould send SQL (Structured Query language) to interactwith RDBMS servers or modify existing SQL to retrieveunauthorized information without any authentication.There is a large amount of literature available covering webapplication vulnerabilities and associated common attacks .Although the types of vulnerabilities and attacksare known since a long time, there is no single solution tomitigate all of them and the limited security support offeredby web application frameworks and the popularity andgrowing complexity of web applications have made themmore prone to attacks than ever .Common webapplication vulnerabilities can be classified into three typesInjection Vulnerabilities, Business LogicVulnerabilities, and Session Management Vulnerabilities.



B. SQL Injection Attack:

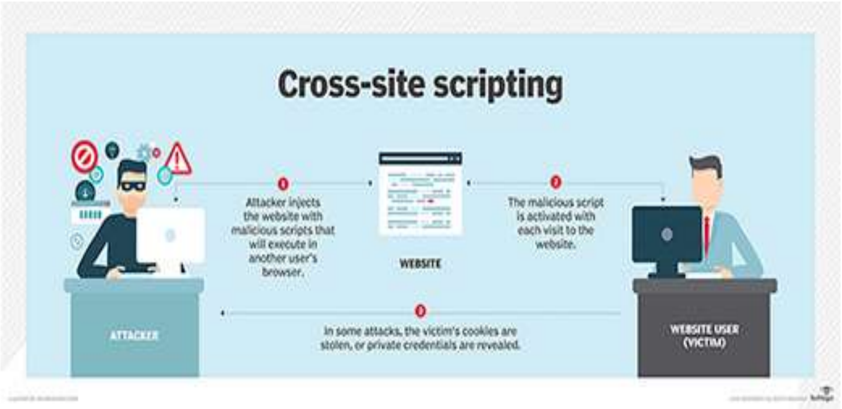
SQL injection attacks have many different forms andfunctions depending on the sources, goals, SQL specifictechniques, and platform configurations. The earliestformal classification of SQL injection attacks was presented by Helford. Which was based on theinjection mechanism and attack intent. Following their footprints, Sun et al. proposed a more elaborate modelof SQL injection attacks considering assets, threats, andCountermeasures in an attempt to establish a semanticrelationship between the different classes of attacks.

- **Tautological Attacks**: In logic, a tautology is a formulathat evaluates to true in every possible interpretation. Themain goal of a tautological SQL injection attack is tochange the conditional in the WHERE clause of thedynamic SQL query so that it always evaluates to trueirrespective of the original condition given by theprogrammer.

- **UNION based Attacks:**In SQL, the UNION keyword is used to combine the result from multiple SELECTstatements into a single result set. In UNION-based SQLinjection attacks, the attacker injects an additional queryusing the UNION keyword to bring data from other tablesor columns into the result set so that they can be displayedon the web page. For a UNION query to be valid, bothqueries must have the same number of columns of the same data types. This information is previously collected by theattacker through error messages from illegal or incorrectquery attacks.

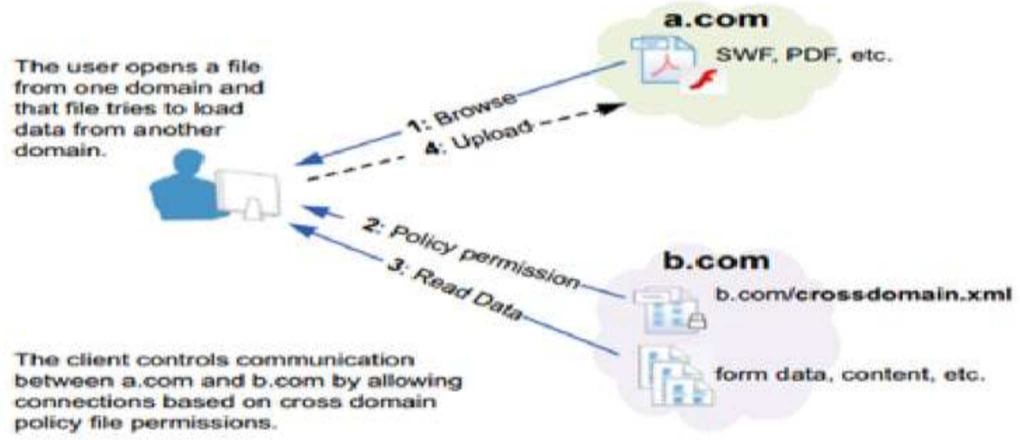
C. Cross-Site Scripting:

Cross-site scripting is a prominent threat in web-basedapplication, caused through a malicious input to theapplication. Cross-Site Scripting (XSS) attacks are a typeof injection, in which malicious scripts are injected intootherwise benign and trusted websites. XSS attacks occurwhen an attacker uses a web application to send maliciouscode, generally in the form of a browser side script, to adifferent end user. Flaws that allow these attacks tosucceed are quiteWidespread and occur anywhere a web application usesinput from a user within the output it generates withoutvalidating or encoding it. XSS is a new commonvulnerability which can let hackers inject the code into theoutput application of web page which will be sent to avisitor's web browser and then, the code which wasinjected will execute automatically or steal the sensitiveinformation from the visits input.


Cross-site scripting
1
Attacker injects the website with malicious scripts that will execute in another user's browser.
WEBSITE
2
The malicious script is activated with each visit to the website.
3
In some attacks, the victim's cookies are stolen, or private credentials are revealed.
ATTACKER
WEBSITE USER (VICTIM)

D. Related Work:

The study follows the guidelines for conducting Systematic Literature Review (SLR) by Kitchenham and Charters and Kitchenham et al.whichare widely used in theSoftware Engineering domain. Theguidelines are alsoapplicable in general toother areas of Computer Science and Engineering. TheSQL and XSS injection attack problem has long gained theattention of the research community and significant workon detection and prevention of SQL injection attacks hasbeen done in over a decade. However, very few comprehensive literature surveys have been conducted. Wehave conducted a systematic literature survey of thequalitative research works on prevention and detection ofSQL injection attacks published since 2002 up to late 2017.“Preventing SQL Injection Attacks in StoredProcedures”, they also provided a novel approach to shield the stored procedures from attack and detect SQL injection.


a.com
SWF, PDF, etc.
The user opens a file from one domain and that file tries to load data from another domain.
1: Browse
4: Upload
2: Policy permission
3: Read Data
b.com
b.com/crossdomain.xml
form data, content, etc.
The client controls communication between a.com and b.com by allowing connections based on cross domain policy file permissions.

III. METHODOLOGY:

Design Science Research Methodology (DSRM) approachhas been selected. We have preferred this methodology bythree objectives it is consistent with prior literature, itprovides a nominal process model for doing DS research,and it provides a mental model for presenting andevaluating DS research in security area. Moreover, DSRM include constructs, models, methods, and instantiations and therefore, it is reasonable to use DSRM as the proposed artifact is a model which is intended to provide guidance on how to prevent SQLI and XSS injection attack in web applications performed objectively.

- Inclusion and exclusion criteria
- B. Searching strategy
- Searching
- D. Obtaining and Assessing
- E. Critical Evaluation
- F. The algorithm used in this study Secure hash algorithm

IV. CONCEPTUAL MODEL:

He SHA512 i.e. Secure Hash Algorithm is based on the concept of hash function. The basic idea of a hash functionis that it takes a variable length message as input andproduces a fixed length message as output which can alsobe called as hash or message-digest.

- Requirements of Prevention and Detection System
  - False Positive (FP)
  - True Positive (TP)
  - False Negative (FN)
  - True Negative (TN)

V. CONCLUSION:

This study implements a technique to prevent and detect SQLIA and XSS. The implementation used hashingtechnique which is computationally light. The proposedtechnique effectively prevents and detects SQLIA andXSS attack without much over head on the application.In this work parameterized queries are used to avoidblind SQLIAs. The hashing technique and theparameterized queries have been tested against 65 testcases, here the application behaved as predicted againstall these test cases that showed we have meet ourobjectives and the developed framework is capable ofprevent and detect SQLIA and XSS attack. This paperanalyzed the problems that current Web VulnerabilityScanners are facing when trying to detect XSSvulnerabilities, as reported in recent research it wasfound that the vulnerability scanners are a promisingmechanism to fight the XSS vulnerabilities in webapplications.

REFERENCES:

1. ZhendongSu, Gary Wassermann. University of California, Davis.“The Essence ofCommand Injection Attacks in Web Applications.” RetrievedJanuary 11, 2009, fromhttp://portal.acm.org.
2. Merlo, Ettore, Letarte, Dominic, Antoniol&Giuliano.”AutomatedProtection of Applications against SQL-injection Attacks.”Software Maintenanceand Reengineering, 11th European Conference IEEE CNF.
3. B. Kitchenham, O. P. Brereton, D. Budgen, M. Turner, J. Bailey,and S. Linkman, “Systematic Literature Reviews in SoftwareEngineering–A Systematic Literature Review,” Information andSoftware Technology, vol. 51, no. 1, pp. 7–15. Elsevier, 2009.
4. K. Peffers, T. Tuunanen, M. A. Rothenberger and S. Chatterjee, "Adesign science research methodology for information systemsresearch," Journal of management information systems, vol. 24, pp.45-77, 2007.
5. A. Hevner and S. Chatterjee, "Design science research ininformation systems," in Design research in information systems,Springer, 2010, pp. 9-22.
6. Oates, B.J, (2006). “Reviewing the literature” – ResearchingInformation Systems and Computing. London, England: SAGEPublications Ltd.
7. Backman, Lars. "Why is security still an issue? A study comparingdevelopers’ software security awareness to existing vulnerabilities insoftware applications." (2018).
8. Bissyandé, Tegawendé F., et al. "Vulnerabilities of GovernmentWebsites in a Developing Country–The Case of BurkinaFaso." International Conference on e Infrastructure and e-Servicesfor Developing Countries. Springer, Cham, 2015.
9. B.SHAH, C., and PANCHAL, D. R. “Secured hash algorithm-1”:review paper. In International Journal for advance research inengineering and technology (Oct 2014), pp.1–6.
10. Temeiza, Q., Temeiza, M., and Itmazi, J. “A novel method forpreventing SQL injection using sha-1 algorithm and syntaxawareness.”In 2017 Joint International Conference on Informationand Communication Technologies for Education and Training andInternational Conference on Computing in Arabic (ICCA-TICET)(Aug 2017), pp. 1–4.
11. D. Litchfield, “Remote Web Application Disassembly with ODBCError Messages,” Blackhat Asia, 1,[Online]. Available:www.blackhat.com/ presentations/bh-asia-01/litchfield/litchfield.doc
12. C. Anley, “Advanced SQL Injection in SQL Server Applications,”NGSSoftware Insight Security Research (NISR), Next GenerationSecurity Software Ltd,Online].Available:http://www.cgisecurity.com/lib/advanced_sql_injection.pdf

13. C. Anley, “(more) Advanced SQL Injection in SQL ServerApplications (White Paper),” NGSSoftware Insight SecurityResearch (NISR), Next Generation Security Software Ltd, [Online].Available: http://www.cgisecurity.com/lib/more_ advanced_sql_injection.pdf
14. TrustWave, “Trustwave 2011 Global Security Report,” [Online]. Available:https://www.trustwave.com/Resources/Library/Documents/2011- Trustwave-Global-Security-Report/?dl=1.
15. Nirav Desai “HACKING&TRICK” a Blog about Hacking &Computer Security https://tipstrickshack.blogspot.com/2013/01/listof-vulnerability-its-tutorial.html.