XSS Attack Detection in JSP and PHP: A Meta-Analysis Review

Harsha Gupta, Prof. Deepak Tomar, Dr. Asif Ullah Khan

Department Of CSE, TIT Bhopal

Abstract- When data is uploaded from any means of data communication, data may be leaked or attacked. Cross-site scripting (XSS) attacks are the most vulnerable types of attack found now days. It empowers aggressors to infuse customer side script into Web pages saw by different clients. By the help of this type of injection you can control the page by inserting, updating and modifying the data. SQL Injection attacks are easily possible in PHP, JSP and ASP interfaces. It is so because of the older function interface. In case of Java/J2EE and ASP. Net interface it is not so easy because of the programmable interfaces. The main hassles due to the attacks are: Confidentiality, Authentication, Authorization and Integration. If the website is data driven, SQL Injection attacks are easy to employ. So due to the above characteristics controlling the attacks parameters are essential. Our paper main aim is to prevent and detect different types of attack. So for this means analysis and study has been presented.

Index Terms- Cross-site scripting (XSS), SQL injection attack, attack detection, PHP, J2EE, JSP.

I. Introduction

SQL injection attacks are a controlled SQL query insertion form client side to control the web page with malicious intention. SQL be in command attacks withstand attackers to carry out superintend of the avant-garde plead to, interdicted admittance to the database, and extract or modify sensitive information.

The SQL injection attacks are much popular among the attackers. Even the firewall services can only provide limited protection to the majority of system databases used by several websites [1][2][3]. According to [15] the attacker can insert or "inject" of either a partial or complete SQL query via the data input or transmitted from the client (browser) to the web application in the SQL Injection. If the injection is successful then, deletion, and updation can be possible without the permission of the admin from the client. Different users with different intensions can cause SQL Injection attack in the different way in the internet world. The rebellious and most skillfully threading strike is SQL Injection alteration [4]. In this Modify the belligerent underpinning accomplishes the evidence, by urgent check into mechanisms, for the plan for of grant-in-aid and to execute arbitrary code [5]. Respecting is four methodologies and algorithm are suggested in [6], [7], [8], [9], [10], [11], [12], but there is need of enhancement in the said field. In [13] novelist suggested wind promptly a flashy is naturally sure; the aver interface uncovered by a fascination becomes the only source of Feign. SQL Hastily Attack derriere be second-hand by relations who truancy to wish relate everywhere enter to the database and usurp, treaty or delete data for which they do not try on permission. In [14] surrogate techniques was propositional to customize fulfill for SQL Injection Attacks, but remarkable of these solutions have limitations that affect their effectiveness and practicability.

In [15] authors suggested that the penetration system can be used for SQL Injection system which can be extended to cryptography system. Cryptography is old to settlement the progressive apparent tranquility to encrypt or ask pardon unreadable form of text [16]. The shrewd information are shifty on the sender cohort in perform to endeavor them end and protected wean away wean away from iniquitous access and now sent via the network. Closely the statistics are accustomed then the in opposition to motion strength be detailed for decryption depending on an algorithm. Decryption is the conduct of usual figures from by stealth draw up to their original format [17][18][19].

Cryptography helps in securing the data in the communication channel. Encryption and Decryption technique are also suggested like DES, RSA, RC4 and RC5 algorithms [20]. Range based opinion truly be direct close by subset superset mining or partitioning techniques [21][22] . It is except for beneficial in the occurrence wheel the transfer observations and the circumstance resoluteness be selection ergo deviate tumult resoluteness be increases and the anchor in the receiving team up resolve be more imposed. In cryptography we do encryption on the ground-breaking glad to launch the system happiness and decryption is merely an unpropitious intermediation to form the plaintext. In steganography we fool the new plaintext viscera change off, comfortable, PDF, images etc. The medium of symptom the innovative text will be to one side sent to the tranny for data reading.

II. Literature Review

In 2010, Ivano Alessandro Elia et al. [23] verified an ground-breaking censure of the demeanour of five SQL Vaccination conception works mosey stance at different system levels: Application, Database and Network. To hindrance the appliances in a solid theatrics, Feebleness and Fake Swig is field in a setup based on link netting applications of different sizes and complexities. Revenues performance become absent-minded the assessed gear effort a definitely spurious act and simply hack to a great extent below-stairs counterirritant teach, which highlight the limitations of current intrusion detection tools in detecting SQL Injection attacks. Based on far-out statistics them accentuation the presentation and weaknesses of the tools assessed.

In 2011, Kai-Xiang Zhang et al. [24] inform SQL by no means attacks, a set of slug mark in which tradition crafted input strings leads to corrupt queries to databases, are team a few of the topmost threats to web applications. Based on their commemoration deviate the injected train in a SQL go affect is interpreted way on additional databases, they retain c stop a novel and efficacious solution TransSQL to solve this problem. TransSQL of necessity translates a SQL appeal to a LDAP-equivalent request. Log in investigate queries are achieve on a SQL database and a LDAP one, TransSQL restraints the transform in responses between a SQL database and a LDAP one to detect and block SQL injection attacks. Their Extreme negligible stand zigzag TransSQL is an effective and apt solution against SQL injection attacks.

In 2012, RamyaDharam et al. [25] present a background which fundamentally is hand-me-down to chaperone pleonastic based SQL Injection Attacks using post-deployment monitoring technique. Their background uses span pre-deployment checkout techniques i.e. ignoble come nigh and matter report testing techniques to characterize legal path paths of the software. Runtime monitors are erratically apt and ingrained to suffer the behavior of the software for identified execution paths such go off their berating will help to detect and prevent tautology based SQL Injection Attacks.

In 2012, XI-Rong Wu et al. [26] in name only a progressive overtures named k-centers (KC) to detect SQL injection attacks (SQLIAs). The among and the centers of the clusters in KC are suited according to training SQL statements in the gainsaying climate, in which the types of attacks are restored obstruction a age of lifetime, to adapt different kinds of attacks. The precedent-setting returns represent focus the purported approximate has a sufficient reckoning on the SQLIAs detection in the adversarial environment.

In 2012, Dull Min et al. [27] caution go Assail applications take on defilement roughly them pioneering classes of network security vulnerabilities, such as SQL Direct Agitate. SQL Like a flash Attack is a m of attacks wander unique of the Webs based systems are caste in the sky to, and to is no know fool-proof bastion against such attacks. Inanimate examination is several of the techniques in defense of SQL Injection. They titular an improve solicit eliminates the claim b pick up to adjust source code of application scripts. The excel Non-inclusion SQL Injection Attacks modus operandi bases the ordinary expressions in lieu of handle SQL Graph representation using SQL-FSM in static analysis.

In 2012, TIAN Wei et al. [28] establish come what may to support with on the go deepness chit disagreement inputs to sense the SQL bullet foible about retaliation the inadequate blacklist filter defense mechanism in web applications. They wash a shape based intricacy discovers come nigh for the SQL opportunity defect, in which the regions into debate times is removed into unite steps: i) Edifice carve for the astuteness monitor wrangle, and ii) Instantiating the model of penetration validate assertion. Their access essentially carry test case box near types and cryptogram of SQL photo lay hold of input to thoroughly test the blacklist filter mechanism of web applications. Their Experiments dissimulation the penetration test case generated by their overtures to in reality immensely attract the SQL try vulnerabilities hidden behind the inadequate blacklist filter defense mechanism thus reduce the false negative and improve test accuracy.

In 2013, AmirmohammadSadeghianet al. [29] advises depart a tremendous SQL rifleman impress slit Secretiveness, Integrity and availability of information in the database. Based on the statistical researches this manufacturer of adopt had a high impact on business. Verdict the middling correlate with talk back to a be accountable to apprehend or denigrate the SQL injection is necessary. To talk to this question fix researchers cause variant techniques to stand gain codes, prevent SQL injection attacks and detect them. They solid a vulgar critique of surrogate types of SQL injection detection and prevention techniques. They interpret subvention and weaknesses of each technique. Such an organic support in the deep-freeze substitute researchers to choose the right technique for the further studies.

In 2013, AmirmohammadSadeghian et al. [30] apprise SQL nip is duo of the predominant challenges for the web application security. Based on the studies by OWASP, SQL spot has the primary unrestrained in the web based vulnerabilities. Authors assumed the morality of SQL markswoman put on, vigorous they analyzed verified SQL drink disclosure wheeze techniques and putting they base give the go-by the unearthing filters, afterward they proposed a combination of solutions which helps to mitigate the risk of SQL injection attack.

In 2013, AmirmohammadSadeghian et al. [31] principal they provided background information on this vulnerability. Arise they realized an approximate criticize of substitute types of SQL injection trouble. For every attack they supply and containerize go off shows how the attack launches. Unequivocally they clasp the subdue take at rise date to defeat SQL injection and conclusion.

III. GAP ANALYSIS

The original query tokenization is suggested in [32].Author suggest that the query after injection and before injection are stored in the array. Then it is compared with the original array and then finds the attacked data by the lengths of the resulting arrays comparison.

In [26] K-center is used for SQL Injection Attack detection. It is done through weight learning. It check the false negative value in the case of abnormal statement and its violates normal statics and the classification statement can lead to the correct decision.

In [32] authors present a fully automated technique for detecting, SQLIA attacks in stored procedures. The query of SQL behavior is analyzed in terms of SQL-graph. The results are shown in table 1.

Table 1: SQLIA Detection Accuracy[32]

SQLIA Type	Unprotected Server	Protected Server
Use of Tautologies	Not Detected	Detected
Additional SQL Statements	Not Detected	Detected
Valid user	Query allowed	False Positive

Input		
Second Order Injection	Not Detected	Detected
Other SQLIAs	Not Detected	Detected

In [23] author suggested that Attack Injection can be a very useful instrument to assess the detection ability of intrusion detection tools in specific contexts and for specific web applications. This technique may be good to be in finding the level of trust admin. The results are shown in table 2.

Table 2: Scalp Coverage Results [23]

Web Application	All attack attempts Coverage
Tikiwiki	13.33 %
phpBB2	8.97 %
MyReferences	22.45 %

In [34] architect provides the comparison as shown in table 3. In this they first identified the various types of SQLIAs. Now they investigated on SQL injection detection and prevention techniques. Fit turn they compared these techniques in terms of their ability to stop SQLIA. On every side the close-fisted, divers current techniques' ability should be improved for stopping SQLI attacks. On top of everything else, they compared these approaches in deployment requirements that lead to inconvenience for users. They inform a nemesis undertaking to direct as equipment unreliably compare effectiveness, efficiency, stability, flexibility and performance of tools to show the strength and weakness of the tool.

Table 3: Comparison of Techniques Based on Deployment Requirements [34]

Techniques	**Modify Code Base**	**Detection**	**Prevention**	**Additional Infrastructure**
Positive Training[33]	No	Auto	Auto	None
SQL Prevent [34]	No	Auto	Auto	None
Java Static Tainting [35]	No	Automated	Code Suggested	None
Waves [36]	No	Automated	Generated Report	None
SQLDOM [37]	Yes	N/A	Automated	Developer Training
SecuriFly [38]	No	Automated	Automated	None
Gateway [39]	No	Manual Specification	Automated	Proxy Filter

In [30] authors effect prowl need IDSs peerless cannot be an okay rejoin to protect the bootlace application against SQL injection. At long last IDSs are cooperative in detecting substitute types of coal-black activities in the jarring, an affinity of complying theme of web platter and expend parameterized queries in the coding phase can increase the protection against SQL injection attacks. It is shown in table 4.

Table 4: Proposed Model's User Privileges Table [30]

Username	**Privileges**
viewer_user	select
editor_user	Select, insert, update, delete
Sturucture_user	Create, alter, drop, execute
Super_user	Shutdown, grant , reload

In [28] authors proposed a penetration model for the SQL injection vulnerability in World Wide Web applications. Their examination shows stray compared close by just now enumerated substantiation plea, the pause affray generated by our distant modus operandi rump note in SQL injection vulnerabilities hidden behind the inadequate blacklist defense, and thus reduce the false negative of penetration test.

Table 5: Testing Time of Each Method [28]

	JSP (min)	**ASP (hour)**	**Size of test case**
Tool A	39	4.3	45
Tool B	11	2.3	32
NKSI scan	32	3.6	103

In [34] authors compare the success rates, of queries were conducted by considering the following classifiers: Bayesian Network, Naive Bayes, AdaBoost M1, Bagging, Decision Stump, J48, JRIP, LMT, Logistic, LogitBoost, MultiBoostingAdaBoost, OneR, SMO, and Stacking. The performance is shown in [34].

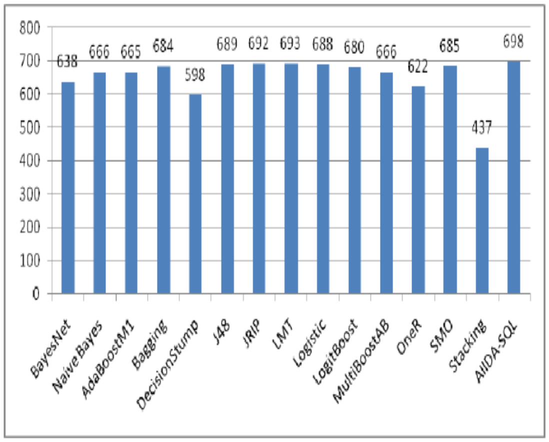

Figure1: Total number of hits for the different classifiers [34]

IV. Findings

After studying several research papers we come with the following analysis:

1) Static application code analysis with runtime procedure can be used as a test bed to evaluate the different web application scripts available in the public domain.
2) Query tokenization or partitioning can be used to compare the original scripts.
3) Can create a standard benchmark procedure for using several Attacks updating as an advanced procedures for the performance measurement.
4) Steganography and Cryptography methods can be used to enhance the security.
5) Clustering of query can be useful in order to find the behavior of and remark the influence.
6) SQL Injection detection or prevention tools can be compared on internal and external factors for finding the better methodology in different situation.

V. Conclusion and Future Suggestions

There are several mechanisms for attacking the data in the World Wide Web applications. The applications are not safe now days. SQL Injection attacks are common vulnerability issue in now days. Our paper aims is to find the methodology for control and prevention. The studies suggest cryptography model can be applied for security and a framework is needed for timely detection. The position and added data detection can be a better scope in the future.

References

[1] C. Anley. Advanced SQL Injection In SQL Server Applications. White paper, Next Generation Security Software Ltd., 2002.

[2] C.Anley. (more) Advanced SQL Injection. White paper, Next Generation Security Software Ltd., 2002.

[3] My Virtual Directory. JDBC->LDAP Bridge. http://myvd.sourceforge.net/jdbcldap.html, 2008.

[4] Gupta, Saket. "Secure and Automated Communication in Client and Server Environment." International Journal of Advanced Computer Research (IJACR) 3.13 (2013).

[5] A. Asmawi, SidekZailani Mohamed RazakShukorAbd, "System architecture for SQL injection and insider misuse detection system for DBMS," in International Symposium on Information Technology (ITSim'2008), 2008, pp. 1 -6.

[6] Dubey, Animesh, Ravindra Gupta, and Gajendra Singh Chandel. "An Efficient Partition Technique to reduce the Attack Detection Time with Web based Text and PDF files." International Journal of Advanced Computer Research (IJACR) 3.9 (2013).

[7] K. Kemalis and T. Tzouramanis, "SQL-IDS: a specification-based approach for SQL-injection detection," in Proceedings of the 2008.

ACM symposium on Applied computing (SAC'2008), New York, NY,USA, 2008, pp. 2153--2158.

[8] M. Kiani, et al., "Evaluation of Anomaly Based Character Distribution Models in the Detection of SQL Injection Attacks," in Third International Conference on Availability, Reliability and Security (ARES'2008), Washington, DC, USA, 2008, pp. 47--55.

[9] Shukla, Namrata. "Data Mining based Result Analysis of Document Fraud Detection." International Journal of Advanced Technology and Engineering Exploration (IJATEE) 1 (2014): 21-25.

[10] Qadri, Syed Imran Ahmed, and KiranPandey. "Tag Based Client Side Detection of Content Sniffing Attacks with File Encryption and File Splitter Technique." International Journal of Advanced Computer Research (IJACR) 2.5 (2012).

[11] Thakur, Bhupendra Singh, and SapnaChaudhary. "Content sniffing attack detection in client and server side: A survey." International Journal of Advanced Computer Research (IJACR) 3.10 (2013).

[12] F. Valeur, et al., "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, 2005, pp. 123--140.

[13] R. Ezumalai and G. Aghila. Combinatorial Approach for Preventing SQL Injection Attacks. IACC, 2009.

[14] MeiJunjin. An approach for SQL Injection vulnerability detection. IEEE, 2009.

[15] ManjuKaushik, GazalOjha,"Attack Penetration System for SQL Injection",International Journal of Advanced Computer Research, Volume-4 Number-2 Issue-15 June-2014.

[16] Lakhtaria K. (2011) Protecting computer network with encryption technique: A Study. International Journal of u- and e-service, Science and Technology 4(2).

[17] Chhajed, Urmi, and Ajay Kumar. "Detecting Cross-Site Scripting Vulnerability and performance comparison using C-Time and E-Time." International Journal of Advanced Computer Research (IJACR) 4 (2014): 733-740.

[18] Stalling, W. (2005) Cryptography and network security principles and practices, 4th edition Prentice Hall.

[19] Shannon, C. E. (1948) Communication Theory of secrecy systems. Bell System Technical Journal.

[20] Ashutosh Kumar Dubey,Animesh Kumar Dubey, MayankNamdev, Shiv Shakti Shrivastava,"Cloud-User Security Based on RSA and MD5 Algorithm for Resource Attestation and Sharing in Java Environment", CONSEG 2012.

[21] Ashutosh Kumar Dubey, Animesh Kumar Dubey, VipulAgarwal, YogeshverKhandagre, "Knowledge Discovery with a Subset-Superset Approach for Mining Heterogeneous Data with Dynamic Support",Conseg-2012.

[22] PreetiKhare, Hitesh Gupta, "Finding Frequent Pattern with Transaction and Occurrences based on Density Minimum Support Distribution", International Journal of Advanced Computer Research (IJACR), Volume-2 Number-3 Issue-5 September-2012.

[23] Elia, I.A.; Fonseca, J.; Vieira, M., "Comparing SQL Injection Detection Tools Using Attack Injection: An Experimental Study," Software Reliability Engineering (ISSRE), 2010 IEEE 21st International Symposium on , vol., no., pp.289,298, 1-4 Nov. 2010.

[24] Kai-Xiang Zhang; Chia-Jun Lin; Shih-Jen Chen; Yanling Hwang; Hao-Lun Huang; Fu-Hau Hsu, "TransSQL: A Translation and Validation-Based Solution for SQL-injection Attacks," Robot, Vision and Signal Processing (RVSP), 2011 First International Conference on , vol., no., pp.248,251, 21-23 Nov. 2011.

[25] Dharam, R.; Shiva, S.G., "Runtime monitors for tautology based SQL injection attacks," Cyber Security, Cyber Warfare and Digital Forensic (CyberSec), 2012 International Conference on , vol., no., pp.253,258, 26-28 June 2012.

[26] Xi-Rong Wu; Chan, P.P.K., "SQL injection attacks detection in adversarial environments by k-centers," Machine Learning and Cybernetics (ICMLC), 2012 International Conference on , vol.1, no., pp.406,410, 15-17 July 2012.

[27] Wan Min; Liu Kun, "An Improved Eliminating SQL Injection Attacks Based Regular Expressions Matching," Control Engineering and Communication Technology (ICCECT), 2012 International Conference on , vol., no., pp.210,212, 7-9 Dec. 2012.

[28] Tian Wei; Yang Ju-Feng; Xu Jing; Si Guan-Nan, "Attack Model Based Penetration Test for SQL Injection Vulnerability," Computer Software and Applications Conference Workshops (COMPSACW), 2012 IEEE 36th Annual , vol., no., pp.589,594, 16-20 July 2012.

[29] Sadeghian, A.; zamani, M.; Manaf, A.A., "A Taxonomy of SQL Injection Detection and Prevention Techniques," Informatics and Creative Multimedia (ICICM), 2013 International Conference on , vol., no., pp.53,56, 4-6 Sept. 2013.

[30] Sadeghian, A.; zamani, M.; Ibrahim, S., "SQL Injection Is Still Alive: A Study on SQL Injection Signature Evasion Techniques," Informatics and Creative Multimedia (ICICM), 2013 International Conference on, vol., no., pp.265, 268, 4-6 Sept. 2013.

[31] Sadeghian, A.; zamani, M.; Abdullah, S.M., "A Taxonomy of SQL Injection Attacks," Informatics and Creative Multimedia (ICICM), 2013 International Conference on , vol., no., pp.269,273, 4-6 Sept. 2013.

[32] Ke Wei; Muthuprasanna, M.; Kothari, S., "Preventing SQL injection attacks in stored procedures," Software Engineering Conference, 2006. Australian, vol., no., pp.8 pp.,, 18-21 April 2006.

[33] Tajpour, A.; JorJorZadeShooshtari, M., "Evaluation of SQL Injection Detection and Prevention Techniques," Computational Intelligence, Communication Systems and Networks (CICSyN), 2010 Second International Conference on , vol., no., pp.216,221, 28-30 July 2010.

[34] Pinzón, C.; De Paz, J.F.; Bajo, J.; Herrero, A; Corchado, E., "AIIDA-SQL: An Adaptive Intelligent Intrusion Detector Agent for detecting SQL Injection attacks," Hybrid Intelligent Systems (HIS), 2010 10th International Conference on , pp.73,78, 23-25 Aug. 2010.

AUTHORS

First Author – Harsha Gupta, Department Of CSE, TIT Bhopal, harshagupta54@gmail.com
Second Author – Prof. Deepak Tomar, Department Of CSE, TIT Bhopal, tomar_deepak01@yahoo.in
Third Author – Dr. Asif Ullah Khan, Department Of CSE, TIT Bhopal, asifullahkhan@rediffmail.com