



# A Survey on SQL Injection

Guide Name: Prof.S.V.Wankhede

1] **First Author Name:-**Sajid Shaikh

**Email:** - pranavpawar645@gmail.com

**Collage Name:-**Singad Academy of  
Engineering Pune

3] **Third Author Name:-**Utkarsh Thakur

**Collage Name:** - Singad Academy of  
Engineering Pune

**Email:-**imsajidshaikh30@gmail.com

2] **Second Author Name:-**Pranav Pawar

**Email:** - utkarshthakur414@gmail.com

**Collage Name:** - Singad Academy of  
Engineering Pune.

**Abstract:** - SQL Injection (SQLi) could also be a kind of an injection attack that makes it possible to execute malicious SQL statements. These statements control a database server behind an internet application. Attackers can use SQL Injection vulnerabilities to bypass application security measures. They can go around authentication and authorization of an internet page or web application and retrieve the content of the entire SQL database. They can also use SQL Injection to feature, modify, and delete records within the database

**Keyword:-**Cloud computing, SQL injection attack (SQLiA), Two fish encryption and decryption, Deep learning, code injection, intrusion detection, supervised learning, SQL injection, XSS, JAVA, JavaScript.

**Introduction:** - Code Injection attacks such as SQL Injection and Cross-Site Scripting (XSS) are among the major threats for today's web applications and systems. This paper proposes our project, a deep learning-based intrusion detection systems against web-based code injection attacks. Our projects main novelty consists in adopting a Convolutional Deep Neural Network and in improving its effectiveness via a tailored pre-processing stage which encodes SQL/XSS-related symbols into type/value pairs. Numerical experiments performed on real-world datasets for both SQL and XSS attacks show that, with an identical training and with a same neural network shape, our projects type/value encoding improves the detection rate..

**Literature Survey:** - Every smallest service on the web is formed available through web applications. Services like online shopping, online banking industry, e-booking system for railways or

airlines and lots of more are all available at the doorstep with the assistance of internet. With these numerous applications comes the majority amount of knowledge they store on day to day in their backend databases. The issue is the way to store the info in an efficient and secure manner in order that it shouldn't be misused. Nowadays most of the applications use Cloud storage for this purpose. But is that the data really secured on Cloud? With the increasing cases of cyber-attacks one can find out the solution to the present question. The attacks like code injection attacks, denial of service attack, spoofing, phishing attack, http flood attack et al. are a number of the main attacks challenging the protection of those applications. One of the main harmful attacks is SQL injection attacks (SQLinAs). There are several detection and prevention techniques for an equivalent yet the applications are highly susceptible to SQLin As in links used that spread to her own page making it seem as she liked it.

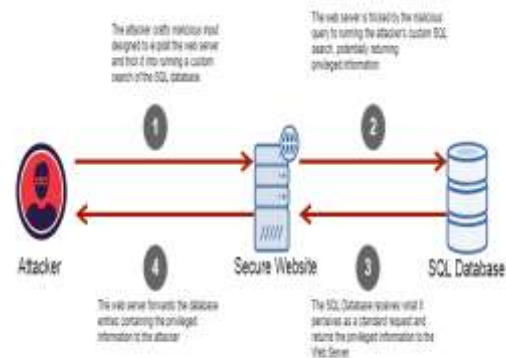
### Algorithms:-

- ❖ **Twofish Encryption Algorithm:** Twofish is a symmetric block cipher with a size of 128 bits and key size length up to 256 bits. Twofish is connected to the earlier block cipher Blowfish. Two fish's characteristic features are the use of pre-computed key-dependent S-boxes, and a comparatively complex key schedule. One half of an n-bit key is use as the definite key of encryption and the other part of the n-bit key is used to adjust the encryption algorithm (key-dependent S-boxes). Twofish borrows some elements from other intends, Twofish has a Feistel structure like Data Encryption Standard. Twofish also utilizes a Maximum Distance divisible matrix. Twofish is a Feistel network. It means that in each

round, half of the text block is driven through an F function, and next XORed with the further part of the text block.

$$F: \{0, 1\}^{n/2} \times \{0, 1\}^N \rightarrow \{0, 1\}^{n/2}$$

### Figure:-



### ❖ Aho-Corasick Algorithm:-

Security of network frameworks is obtaining a lot of essential as users confidential and personal information are being controlled on-line and find hacked systematically. The protection of a machine structure is changed off at the purpose once an intermission happens because it might originate information thieving or developer creating the machine structures a lot of vulnerable. Activity strategy for checking SQL Injection Attack (SQLIA) exploitation Aho-Corasick pattern matching computation. Main focus of this paper is on positive tainting thus detection makes it straight forward. The rule objective is intrusion detection. Investigations exhibit that planned system has higher recognition rate than existing structure.

### ❖ Knuth-Morris-Pratt Algorithm:-

Structured Query Language (SQL) injection and cross-site scripting remain a major threat to data-driven web applications. Instances where hackers obtain unrestricted access to back-end database of web applications so as to steal, edit, and destroy

confidential data are increasing. Therefore, measures must be put in place to curtail the growing threats of SQL injection and XSS attacks. This study presents a technique for detecting and preventing these threats using Knuth-Morris-Pratt (KMP) string matching algorithm. The algorithm was used to match user's input string with the stored pattern of the injection string in order to detect any malicious code. The implementation was carried out using PHP scripting language and Apache XAMPP Server. The security level of the technique was measured using different test cases of SQL injection, cross-site scripting (XSS), and encoded injection attacks. Results obtained revealed that the proposed technique was able to successfully detect and prevent the attacks, log the attack entry in the database, block the system using its mac address, and also generate a warning message. Therefore, the proposed technique proved to be more effective in detecting and preventing SQL injection and XSS attacks.

### Advantages:-

- Retrieving hidden data, where you can modify an SQL query to return additional results.
- Subverting application logic, where you can change a query to interfere with the application's logic.
- UNION attacks, where you can retrieve data from different database tables.
- Examining the database, where you can extract information about the version and structure of the database.
- Blind SQL injection, where the results of a query you control are not returned in the application's responses.

### Disadvantages:-

SQL injection attacks pose a serious security threat to organizations. A successful SQL injection attack can result in confidential data being deleted, lost or stolen; websites being defaced; unauthorized access to systems or accounts and, ultimately, compromise of individual machines or entire networks. Twenty years after its discovery, SQL injection remains a top database security concern.

### Conclusion:-

We have presented a survey and comparison of current techniques for detecting and preventing SQLIAs. To perform this evaluation, we first identified the various types of SQLIAs known to date. We then evaluated the considered techniques in terms of their ability to detect and/or prevent such attacks. We also studied the different mechanisms through which SQLIAs can be introduced into an application and identified which techniques were able to handle which mechanisms. Lastly, we summarized the deployment requirements of each technique and evaluated to what extent its detection and prevention mechanisms could be fully automated. Our evaluation found several general trends in the result.

### Reference:-

- [1] S. Deering and R. Hinden, IETF RFC2460, Internet Protocol, Version 6, 1998, <http://www.ietf.org/rfc/rfc2460.txt>.

- [2] M. Boucadair, J. Grimault, P. Levis, A. Villefranque, and P. Morand, "Anticipate IPv4 address exhaustion: a critical challenge for internet survival," in Proceedings of the 1st International Conference on Evolving Internet (INTERNET '09), pp. 27–32, Cannes La Bocca, France, August 2009.
- [3] M. Gunn, "War dialing," 2002.
- [4] Wikipedia, "War dialing," 2013, [http://en.wikipedia.org/wiki/War\\_dialing](http://en.wikipedia.org/wiki/War_dialing).
- [5] R. Oppliger, "Security at the internet layer," *Computer*, vol. 31, no. 9, pp. 43–47, 1998.
- [6] S. Weber and L. Cheng, "A survey of anycast in IPv6 networks," *IEEE Communications Magazine*, vol. 42, no. 1, pp. 127–132, 2004.
- [7] E. Fong and V. Okun, "Web application scanners: definitions and functions," in Proceedings of the 40th Annual Hawaii International Conference on System Sciences (HICSS '07), Waikoloa, Hawaii, USA, January 2007.
- [8] X. Fu, X. Lu, B. Peltsverger, S. Chen, K. Qian, and L. Tao, "A static analysis framework for detecting SQL injection vulnerabilities," in Proceedings of the 31st Annual International Computer Software and Applications Conference (COMPSAC '07), pp. 87–96, Beijing, China, July 2007.
- [9] J. Bau, E. Bursztein, D. Gupta, and J. Mitchell, "State of the art: automated black-box web application vulnerability testing," in Proceedings of the IEEE Symposium on Security and Privacy (SP '10), pp. 332–345, Oakland, Calif, USA, May 2010.
- [10] G. Pant, P. Srinivasan, and F. Menczer, *Crawling the Web*, 2004.
- [11] A. Heydon and M. Najork, "Mercator: a scalable, extensible web crawler," *World Wide Web*, vol. 2, no. 4, pp. 219–229, 1999 [12] HackTrix, "Stay away from malicious Facebook apps," 2013 [Online]. Available: <http://bit.ly/b6gWn5>
- [12] H. Y. Kao, S. H. Lin, J. M. Ho, and M. S. Chen, "Mining web informative structures and contents based on entropy analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 1, pp. 41–55, 2004.
- [13] I. S. Altingovde and O. Ulusoy, "Exploiting interclass rules for " focused crawling," *IEEE Intelligent Systems*, vol. 19, no. 6, pp. 66– 73, 2004.
- [14] V. Shkapenyuk and T. Suel, "Design and implementation of a high-performance distributed web crawler," in Proceedings of the 18th International Conference on Data Engineering, pp. 357– 368, San Jose, Calif, USA, March 2002.
- [15] C. Castillo, "Effective web crawling," *Computer Science*, The University of Chile in fulfillment: ACM SIGIR Forum, 2004.