IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 3, April 2022
Impact Factor: 6.252

SQL Injection

Prof. P. S. Gawali[1], Sajid Shaikh[2], Pranav Pawar[3], Utkarsh Thakur[4]
Guide, Department of Computer Engineering[1]
Students, Department of Computer Engineering[1,2,3]
Singad Academy of Engineering Pune, Maharashtra, India
imsajidshaikh30@gmail.com[2], pranavpawar645@gmail.com[3], utkarshthakur414@gmail.com[4]

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.*

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

I. 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.

II. PROPOSED METHODOLOGY

The code injection attack can be split into two groups: Binary attack and Source Code attack. Most well-known types of the Source Code injection vulnerabilities are the SQL Injection, PHP injection, and JavaScript injection. The code can be injected through visible or hidden input fields, manually, automatically, or through uploaded or addressed files. If the code is injected into the system command line, this attack will be defined as the command injection. Binary Code injection can happen via shell code injection into the executable file input and cause stack or heap overflow.

III. 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.

3.1 Algorithm

A. Twofish Encryption Algorithm

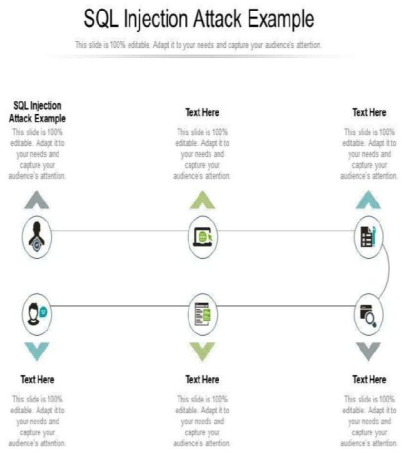
Twofish is a symmetric block cipher with a size of 128 bits and key size length up to 256 bits. Twofish is connected to the earlier block cipher Blowfish. Two fish's characteristic features are the use of pre-computed key-dependent S-boxes, and a comparatively complex key schedule. One half of an n-bit key is use as the definite key of encryption and the other

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 3, April 2022
Impact Factor: 6.252

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 X {0, 1 }N →{0, 1}n/2

B. Figure



that the proposed technique was able to successfully detect and prevent the attacks, log the attack entry the database, block the system using its mac address.

3.2 Mathematical Model

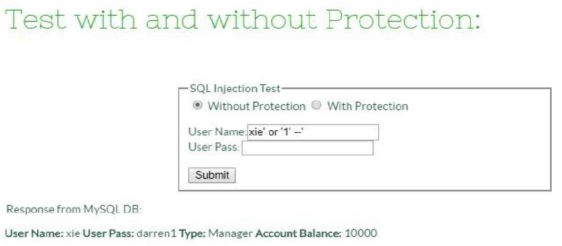
A. Static Analysis

Static analysis analyzes the SQL query sentences of web applications to detect and Prevent SQL injection attacks. It also requires rewriting of web applications. The focus of the static analysis method is to validate the user input type in order to reduce the chances of SQL(JSA) library to validate the user input type dynamically and prevent SQL injection attacks.

B. Dynamic Analysis

Dynamic analysis analyzes the response from a web application after scanning it. A scan means to send every kind of input to the target and receive the response. Unlike static analysis, it can locate vulnerabilities from SQL injection attacks without making any modifications to web applications. Paros [10], which is an open source program, finds not only SQL injection attacks, but also other vulnerabilities within the web application. Paros is not effective because it uses predetermined attack codes to scan and determines the success or fail with the HTTP response.

3.3 Implementation



Output SQL Injection without Protection

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 3, April 2022
Impact Factor: 6.252

Output Record SQL Injection with Protection

3.4 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.

3.5 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.

IV. 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.

REFERENCES

[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.
[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.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 3, April 2022
Impact Factor: 6.252

[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.