A novel method for SQL Injection Prevention

Janardhanan Y
Department of Computer Science andEngineering
SSIET, Coimbatore

Dhinesh S R
Department of Computer Science andEngineering
SSIET, Coimbatore

Manda Santhosh Kumar
Department of Computer Science andEngineering
SSIET, Coimbatore

Arun Anoop M
Assistant Professor
Department of Computer Science andEngineering

Sri Shakthi Institute of Engineering & Technology, Coimbatore, Tamilnadu, India,
arunanoopm@gmail.com

Abstract— **Web security may be a set of procedures, practices, and technologies for shielding internet servers, web users, and their encompassing organizations. Security protects you against surprising behavior. Most internet applications have essential bugs (faults) moving their security, that makes them prone to attacks by hackers and arranged crime. to forestall these security issues from occurring it's of utmost importance to grasp the everyday software system faults. Applications written with robust typewritten languages have a smaller range of reported vulnerabilities and exploits. we have a tendency to had to think about additional robust typewritten applications to get a good quantity of vulnerabilities when put next to the weak typewritten. per our findings, weak typewritten are the popular targets for the event of exploits. we have a tendency to conjointly determined that one fault kind was answerable for most of the protection issues analyzed. the foremost relevant fault sorts analyzed were totally careful providing enough data for the definition of vulnerability fault models. The planned methodology permits gathering the knowledge on common mistakes that developers ought to avoid. to grasp however these vulnerabilities are extremely exploited by hackers, this paper conjointly presents associate analysis of the ASCII text file of the scripts wont to attack them. the result may be wont to train software system developers and code inspectors within the detection of such faults and also the Digital signature is generated to avoid such attacks.**

Keywords—

I. Introduction

Web application security is that the method of securing confidential knowledge hold on on-line from unauthorized access and modification. this is often accomplished by imposing tight policy measures. Security threats will compromise the info} hold on by a corporation is hackers with malicious intentions try and gain access to sensitive information. internet security could be a set of procedures, practices, and technologies for safeguarding internet servers, web users, and their close organizations. Security protects you against sudden behavior. Most info systems and business applications engineered today have an online front and that they got to be universally out there to shoppers, employees, and partners round the world, because the digital economy is changing into additional and additional current within the world economy. These internet applications, which may be accessed from anyplace, become therefore wide exposed that any existing security vulnerability can most likely be uncovered and exploited by hackers. the safety of internet applications becomes a significant concern and it's receiving additional and additional attention from governments, firms, and therefore the analysis community. Attackers conjointly followed the move to {the internet|the online|the net} and intrinsically quite 1/2 current laptop security threats and vulnerabilities have an effect on web applications. the most analysis goal is to know the standard software package faults that area unit behind the bulk of internet application vulnerabilities, taking into consideration totally different programming languages. to know the connexion of those sorts of vulnerabilities for the attackers, the paper conjointly analyzes the code wont to exploit them. To characterize the categories of software package faults in a very set of real internet applications. every patch was inspected full to assemble the precise characteristics of the code that was answerable for the safety downside Associate in Nursingd classified them per an adaptation of the orthogonal defect classification (ODC).

II. Literature Survey

P Anbalagan “**Towards a Unifying Approach in Understanding Security Problems”.**

Security issues, vulnerabilities or faults and security exploits like attacks, failures square measure a set of the overall class of package faults and failures. we have a tendency to gift a model of relationships between security issues and their exploits within the field evaluated however promptly a project team fixes security issues, i.e., whether or not there's any backlog in fixing security issues or not, a project team’s response to security issues that have older failures (exploits) and people that stay fallow within the field.

III. Overview of Architectures

IV. Cryptography for IoT

Cryptography is a method for safeguarding the data by converting into ciphertext. There are different types of security measures and concepts for each layer of IoT [2] .The cryptography algorithms are mainly used for securing the data. This is a technique which encrypts the data needed to be secured to ciphertext during transmitting data in the network. The Cryptographic is alienated into two approaches namely the symmetric and asymmetric ciphers.

In the Symmetric key encryption method, for both encryption and the decryption process the same key is used. This process takes less time for encryption and decryption processes.

In the Asymmetric key encryption method, the algorithm uses public key and the private key for encryption and decryption process respectively. This type of encryption method provides a good mechanism for sharing the key. However, this method takes more time and the length of the key should not be large.

The IoT based device uses the sensor nodes, RFID, smart card has less memory and runs on less power. The standard cryptography algorithms cannot be used here because of high memory usage and low performance. Hence to overcome the above-mentioned problems of the IoT devices, lightweight cryptography algorithms are introduced which gives almost equal level of security and good performance [5].

V. POSSIBLE ATTACKS ON IOT

The communications of IoT based device happens through Internet, which is a public network. Hence this device becomes vulnerable to different types of attacks which cause interruption in the process [3]. Some of the possible types of attacks are as mentioned below:

DOS. The DOS is an attack which stops the service provided by that machine by keeping the network busy [2]. This attack will completely stop the working as it gets too many service requests at one time leading the machine to crash. This attack makes the machine too busy for authorized party's communication. On network layer the Denial of Service causes spoofing. In spoofing, a message is sent by a spite which is repeatedly used to generate a high traffic on the communication network.

Work Hole. In this type of attack the attacker records bit from the network, channels them in a particular planned way to the other site, and then retransmits them into the network.

The Eavesdropping. In this attack the attacker secretly listening to the private conversation without their permission. The attacker can also change the messages during the conversation between two parties.

The Man-in-Middle. The attack is quite similar to eavesdropping.[2] This attack is done during the communication session between two parties. In this attack the attacker enclosures into the communication and tries to get the information or any other move which may harm the sender and the receiver. The communicating parties can still think that they have been receiving legit messages.

The Fabrication. A fabrication attack seeks to create illegal or false information in a system. These fabricated data are directly inserted or modified in authentic data. The attackers use this to make a compromised system.

VI. Security encounters

The security challenges encountered in the IoTs are:

Integrity of data. Integrity of data is basically a complete consistency as well as the accuracy of the data. The integrity of the data is implemented during the design of the database process using the procedures and rules with specific standards. But due to wrong integrity measure’s data is captured by the third party. The third party (Malicious nodes) can be injected into the network with wrong information which will harm the entire system.

Confidentiality of Data. The confidentiality of data can be implemented in different types. The access control and data encryption are most commonly used procedures to ensure confidentiality with security. Data encryption is the method of securing the data and the access control is one of the methods to control access to system data based on identity who is trying to access the data.

Availability of Data. The data availability is essential and required to assure the quality service to the users. The third party attack makes the resources unattainable to handlers by keeping the server busy. The data can be made available by creating different routes for transmission of the data. By making different routes, if one route is attacked an alternate route can be used for data transmission and this will increase the possibility of attack detection.

The Verification and Approval. The verification is basically an authentication which means confirming the identity of the user and approval means giving access to the authenticated user. Attackers usually try to get the access to system using the liabilities in the verification and approval system. The third party can bypass the mechanisms of verification and approval and can deploy spiteful action on the system. By using the RBAC (Role Based Access Control) which uses the systematic access control protocol the attacks can be reduced.

VII. Comparison of various algorithms

The lightweight block ciphers are compared with respect to their different key sizes with structure, function, Block size, Rounds, Cycles, Areas and Throughput in the below table. The compared algorithms use two types of structures such as SPN (Substitution Permutation Network) and PSNR (Linear- feedback shift register). SPN makes use of S-box and P-box mainly. The PRESENT, RECTANGLE, KLEIN, LED,
PRINT, KATAN and KTANTAN are the algorithms with various key sizes. RECTANGLE with 128bits has a larger

area compared to all the algorithms. KLEIN 64 has a high throughput of the compared algorithms with various key sizes.

The below graph shows various algorithms with their different key sizes compared with the memory consumed by the algorithm in terms of area.

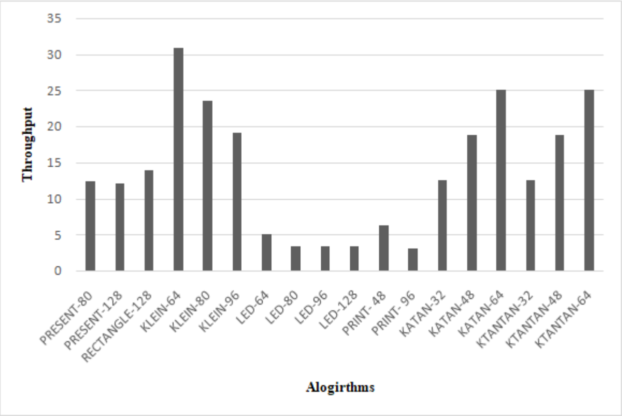

Fig. 2. Throughput of Algorithms

VIII. CONCLUSION

The technique we used in this project was able to correctly identify all attacks as SQLIAs, while allowing all legitimate queries to be performed. In other words, for the cases considered, our technique generated no false positives and no false negatives. The lack of false positives and false negatives is promising and provides evidence of the viability of the technique. In our study, we did not compare our results with alternative approaches against SQLIAs because most of the existing automated approaches address only a subset of the possible SQLIAs.

The results may be related to the specific subjects considered and may not generalize to other web applications. Our empirical evaluation, performed on commercial applications using a large number of realistic attacks, shows that proposed method is a highly effective technique for detecting and preventing SQLIAs.

References

1. Andrews, M.: Guest Editor's Introduction: The State of Web Security. IEEE Security and Privacy, 4, 4, 14--15 (2006)
2. Janot, E.: SQLDOM4J: Preventing SQL Injections in Object-Oriented Applications. Master thesis, Concordia University College of Alberta (2008), http://waziboo.com/thesis
3. McClure, R., Krüger, I.: SQL DOM: Compile Time Checking of Dynamic SQL Statements. In: 27th IEEE International Conference on Software Engineering, pp. 88--96. IEEE Press, New York (2005)
4. Power, R.: 2002 CSI/FBI Computer Crime and Security Survey. Computer Security Issues & Trends, 8, 1, 1--22 (2002)
5. OWASP Top Ten 2007, http://www.owasp.org/images/e/e8/OWASP_Top_10_2007.pdf
6. OWASP Foundation: SQL Injection, http://www.owasp.org/index.php/SQL_injection
7. Chapela, V.: Advanced SQL Injection, http://www.owasp.org/images/7/74/Advanced_SQL_Injection.ppt
8. Halfond, W., Viegas, J., Orso, A.: A Classification of SQL-Injection Attacks and Countermeasures. In: IEEE International Symposium on Secure Software Engineering (2006)
9. Huang, Y., Yu, F., Hang, C., Tsai, C., Lee, D., Kuo, S.: Securing Web Application Code by Static Analysis and Runtime Protection. In: Di Nitto, E., Murphy, A.L. (eds.) 13th international conference on World Wide Web, pp. 40--52. ACM, New York (2004)
10. Boyd, S., Keromytis, A.: SQLrand: Preventing SQL Injection Attacks. In: Nagel, W.E., Walter, W.V., Lehner, W. (eds.) ACNS 2004. LNCS, vol. 3089, pp. 292--304. Springer, Heidelberg (2004)
11. Buehrer, G., Weide, B.W., Sivilotti, P.A.: Using Parse Tree Validation to Prevent SQL Injection Attacks. In: Di Nitto, E., Murphy, A.L. (eds.) 5th International Workshop on Software Engineering and Middleware, pp. 106--113. ACM, New York (2005)
12. Halfond, W., Orso, A.: Preventing SQL Injection Attacks Using AMNESIA. In: Di Nitto, E., Murphy, A.L. (eds.) 28th ACM/IEEE International Conference on Software Engineering, pp. 795--798. ACM, New York (2006)
13. Su, Z., Wassermann, G.: The Essence of Command Injection Attacks in Web Applications. ACM SIGPLAN Notice 41, 1, 372--382
14. Cook, W., Rai, S.: Safe Query Objects: Statically Typed Objects as Remotely Executable Queries. In: Di Nitto, E., Murphy, A.L. (eds.) 27th ACM/IEEE International Conference on Software Engineering, pp. 97--106. ACM, New York (2005)
15. Cole, L.: AppSecInc to Launch Database Security Suite. Database Journal (2007), http://www.databasejournal.com/news/article.php/3657096
16. Ristic, I.: Web Application Firewalls: When Are They Useful?. OWASP AppSec Europe 2006, http://owasp.org/images/9/9c/OWASPAppSecEU2006_WAFs_WhenAreTheyUseful.ppt
17. OWASP Foundation: Preventing SQL Injection in Java, http://www.owasp.org/index.php/Preventing_SQL_Injection_in_Java
18. Oracle TopLink, http://oracle.com/technology/products/ias/toplink
19. Oracle Fusion Middleware Developer's Guide for Oracle TopLink – Using a SQLCall, http://oracle.com/technology/products/ias/toplink/doc/11110/devguide/qrybas.htm#CIHEBF

ID
20. Kost, S.: An Introduction to SQL Injection Attacks for Oracle Developers, http://www.net-security.org/dl/articles/IntegrigyIntrotoSQLInjectionAttacks.pdf