Assignment11 - Security Issues

Name: HARSHITHA YEDDULA

Database security is one of the primary things to maintain and it can either be threat which is present internally or externally. SQL injection is an attack that can be used to gain, eliminate or change data information in any data driven system in a web or non web based system. Because of the diversity of its methods, defense methods can't prevent such attacks. The attacks gives the attackers unlimited access to the underlying systems. When an attacker inserts new SQL keywords or operators into a SQL query, a SQL Injection Attack (SQLIA) occurs. Many alternative input channels can be used to introduce malicious SQL statements into a vulnerable program and many techniques have been developed to prevent sql injections attacks. The input from the user is often obtained from submissions of the forms that are uploaded to the Web application via GET or POST methods in most SQLIAs that attacks the Web applications. In most cases, web applications can access the user input contained in these requests just like any other variable in the environment. But there is no whole system to prevent these kind of attacks. The privileges of the user isused while performing these kind of attacks.

The attacks can happen through different methods from servers and engines of the database. The patches should be fixed with regular updates or else it is more vulnerable to the attacks and it may be the first point of entry for the attackers. Strong validation and monitoring should be handled periodically. Reporting the error from client should not be allowed and the users with permissions be given access. Sensitive data can be protected by means of encryption.

There are a lot of aspects of solution and it includes Static analysis and Runtime analysis. Many architectures are used to organize data systems. Three tier architecture is the common architecture in these systems.The approach detects error based on run time and prevention methods. Also it provides additional defense layer and prevents un related code execution. Database fabrication and hacking are prevented from inside and outside of the database server systems. In second-order injections, attackers plant malicious inputs in a system or database to cause a SQLIA when that input is utilized later.

Several approach stages can be followed to reduce malicious sql queries and it includes replication of system databases and analyzing database behaviors. Also by checking simple sql syntaxes, redirecting queries, virtual and SQLIA detection can also act as protection stages. The variables of server are used by web applications for a variety of purposes, including tracking usage statistics and recognizing browsing trends. A SQL injection vulnerability might be generated if these variables are recorded to a database without being sanitized. Since the attackers can tamper with the HTTP methods and network headers,

This paper discussed about the hybrid technique that is a combination of static and runtime analysis to prevent the injection attack. VB.Net is employed in this simulation application to test different 250 queries and it covers all sql injection attack gateways.It covers all hybrid techniques of SQLIA attacks and prevents the attacks applied directly through databse and many built in functions prevents such attacks. A tautology-based attack's basic objective is to inject code into one or more conditional statements, causing them to always evaluate to true. The impact of this attack is determined on how the query's results are used within the application. Bypassing login pages and extracting data are the most prevalent applications. An attacker uses an injectable field in the WHERE conditional of a query to do this sort of injection. When you transform a conditional into a tautology, all of the rows in the database table that the query is targeting are returned. In general, an attacker must examine not only the injectable/vulnerable parameters, but also the code structures that assess the query results in order for a tautology-based attack to succeed. In most cases, the attack is a surprise. Insufficient input validation is the fundamental cause of SQL injection vulnerabilities. As a result, the most easy way for addressing these flaws is to implement appropriate defensive coding standards.

SQLIAs are carried out by injecting instructions into a string or a numeric argument. Many attacks may be avoided with just a basic check of such inputs. The use of meta-characters to mislead the SQL parser into reading user input as SQL tokens is a common way to inject into a string argument. Developers should provide input validation procedures that distinguish between good and incorrect input. These input sources can be leveraged by an attacker to introduce a SQLIA if they are utilized to build a query. To put it another way, all input sources must be examined.

Detection techniques identifies all places in a Web application that can be utilized to inject SQLIAs using a Web crawler.It then creates assaults based on a list of patterns and attack tactics that target such places. Although static code checks were not designed with the goal of identifying and preventing generic SQLIAs, they can be used to protect against attacks that take advantage of type incompatibilities in a dynamically created query string. One of the fundamental causes of SQLIA vulnerabilities in programming is incorrect type checking of input, which JDBC-Checker can identify. However, because most SQLIAs are composed of syntactically and type-correct queries, our method would miss more broad SQLIAs.

Connection between two articles:

It is more important than ever to provide adequate security. Both studies aim to uncover security issues and offer numerous solutions for assuring restricted, secure access to database contents while maintaining data integrity, consistency, and overall quality.

Key Takeaways:

Future works stated in this paper includes to decrease the overall time delay detected after SQLIA attacks.The key takeaways includes, the methods to detect the injection attacks and other prevention methods were explained and discussed.

Are there statements that you do not agree with?

Should approaches must be used or only a few will enough, and if so, what factors will determine which strategy is chosen over the others.

References: Atoum, J. O., & Qaralleh, A. J. (2014). A hybrid technique for SQL injection attacks detection and prevention. International Journal of Database Management Systems, 6(1), 21

Halfond, W. G., Viegas, J., & Orso, A. (2006, March). A classification of SQL-injection attacks and countermeasures. In Proceedings of the IEEE international symposium on secure software engineering (Vol. 1, pp. 13-15). IEEE