International Journal of Current Engineering and Technology
E-ISSN 2277 – 4106, P-ISSN 2347 – 5161

Available at http://inpressco.com/category/ijcet

Research Article

An Analysis of SQL Injection Prevention using the Algorithms RSA, RC4 and RC5

Chinchu M.M†*, Asha Yeldose† and Deepa.S.Kumar‡

Christ Knowledge City, MG University, Kerala, India
‡College of Engineering, Munnar, India

Accepted 25 Nov 2015, Available online 02 Dec 2015, **Vol.5, No.6 (Dec 2015)**

Abstract

The SQL injection prevention techniques now available are not not sufficient for detecting and preventing SQL injections. So in order to efficiently prevent SQL injection attacks Blow Fish and RSA and RC4 and RC5 algorithms can be implemented which is Blow Fish is used for encryption of authentication fields and RSA RC4 and RC5 algorithm is used for query encryption. The URL is encrypted using Blowfish Algorithm. The SQL injection attacks happen in the Login phase. So the access will be provided to verified users only. That is, at the time of creation of the user account, a user key is generated for every user where the user name and password at the time of login is encrypted by Blowfish encryption .Then the corresponding query generated is encrypted using 3 algorithms RC4,RSA,RC5 technique at different levels of the total encryption process. The access is provided by the server after confirming the user's authenticity. On server side the encrypted data will be decrypted using the user key. The decrypted data will be checked and if the user is genuine, further access will be granted to the database. The RSA, RC4, RC5 encryption will work as a protective cover for the SQL query generated by the user at the clients end. The time and space complexity of both the RSA ,RC4 and RC5 Algorithm is evaluated. The security of all the algorithms are also evaluated

Keywords: *SQLInjection, prevention, ASCII, encryption, decryption*

1. Introduction

Any vulnerability in web applications makes unauthorised users to obtain access to private and confidential information. A survey was conducted about the attacks in web applications in 2014 by OWSAP in which SQL Injection was ranked second. An attacker can take advantage of the security flaws in the web applications and the programming flaws can pass malicious SQL statements through a web application which can explore the back end database.This exposure can lead to many types of information leakage and can either modify or delete the entire database.

Web applications have made our life very simple. Nowadays the web applications cover a variety of of daily needs. A large number of electronic transactions, including e-commerce, e-banking, e- voting, e-learning, and ehealth among others, can be conducted online at any time and from any place . In all these web applications there will be exposure of hackings attempts and security related problem. SQL injection today are the most common indirect attack technique against information stored in databases for web applications and can affect the secrecy, integrity and availability of web applications. SQL injection occurs when an attacker inserts malicious SQL code into an SQL query by manipulating data input into an application.

This kind of vulnerability is a serious threat to any web application that reads input from users and uses it to build and execute SQL queries in the database and extract information. With SQL injection, the attacker can run arbitrary SQL queries which can lead to extracting sensitive customer and order information from online applications, or she/he can bypass strong security mechanisms compromising the back-end databases and the data server file system. This types of attacks are very common now d a days.Any high security system will be compromised by the attacks.Any professionally experienced programmers are not able to cover all type of SQL Injection attacks

In order to perform SQL injection hacking, all an attacker needs is a web browser and a degree of guess work to find important table and field database names, which explains why SQL injection is one of the most common application layer attacks currently being used

*Corresponding author **Chinchu M.M** is a M.Tech Scholar; **Asha Yeldose** is working as Assistant Professor and **Deepa.S.Kumar** as Professor

on the Internet. Although there has recently been a great deal of attention devoted to the problem of SQL injection vulnerabilities, many proposed solutions fail to address all types of SQL injection attacks. The SQL Injections are of many types.

1. Tautology attacks
2. Union Queries
3. Piggy backed queries
4. End of Line Attacks

Tautology-based attacks are among the simplest and best known types of SQLIAs. The general goal of a tautology based attack is to inject SQL tokens that cause the querys conditional statement to always evaluate to true.

Union queries are a type of SQLIA that can be used by an attacker to achieve a goal, which can cause otherwise legitimate queries to return additional data. In this type of SQLIA, attackers inject a statement of the form UNION ¡ injected query ¿ . By suitably defining ¡ injected query ¿ , attackers can retrieve information from a specified table.

Piggy backed queries appends additional queries to the original query string. If the attack is successful, the database receives and executes a query string that contains multiple distinct queries

End of Line attacks add an extra end of line which will execute the query without password and can retrieve all types of information.

2. Related Work

The authors Konstantinos Kemalis and Theodoros Tzouraman proposed a novel methodology in 2008,to detect exploitations of SQL injection vulnerabilities in web applications [Kemalis et.al,2008]. The new approach uses security specifications that describe the intended syntactic structure of SQL statements that are produced by the application. SQL statements which does not match to the specifications are considered as security violations and their execution is blocked. The detection technique is based on the assumption that injected SQL commands have differences in their structure with regard to the expected SQL commands that are originally built the web application. Therefore, if the intended structure of the expected SQL commands has been explicitly pre-determined, it is possible to detect malicious modifications that alter this structure.

The authors R.Ezumalai and G.Akhila in 2009 suggested a paper [R.Ezumalai *et al*, 2009] ,against SQLIAs which is based on Signature based approach, which has been used to address security problems related to input validation. This approach describes three modules which are used to detect the security issues.Monitoring module has got the statement from web application which can decide whether it can send the statement pattern approach to database for execution. Analysis module uses Hirschberg algorithm to compare the statement from the specifications. Specifications comprise the predefined keywords and send analyze module for comparisons. It analyzes the comparisons as well as database transaction. If it finds any suspicious activity, it acts as an active agent to stop the transaction and audit the attacks. If both analysis module and auditing module has satisfied, it provides transaction.

The author MeiJunjin suggested a technique that user input might take a circuitous path from the user interface through one or more methods that may or may not be input filter methods [Meijujin et.al 2005], and ultimately to the SQL command to be executed. This approach traces the flow of the input values that are used for a SQL query by using the AMNESIA SQL query model and string argument instrumentation. Based on the input flow analysis, generate a test attack input for the method arguments used to construct a SQL query. Then generate test cases with an existing JUnit test case generation tool, JCrasher, and modify the test input with attack input. To help programmers to easily identify vulnerable locations in the program, this approach generates a colored call graph indicating secure and vulnerable methods

The authors Mayank Namdev , Fehreen Hasan, Gaurav Shrivastava after studying the various SQL injection prevention techniques in 2009,a technique is proposed in which implement a mechanism, that detect and prevent the SQL injections by incorporating the techniques of HASHING and ENCRYPTION[Mayank et.al, 2009]. The concept behind implemented system is simple: instead of relying on users permissions, implement cumbersome defensive coding techniques with which, can detect and prevent the SQL injections and provide security to the web application.

The proposed method simply works on the HASHING methods for the secure login technique. In which, computes the hash value of the username and passwords for any user and store it in database table along with the simple username and passwords. The proposed implemented system is unable to handle all the SQL injection attacks, but it can prevent tautology attacks, union based query attacks and illegal structured query attacks.

The authors Debabrata Kar and Suvasini panigrahi suggested an approach in 2013, which is a novel query transformation scheme that transforms a query into its structural form [Debarata et.al, 2013] instead of the parameterized form. It is assumed that any query generated by the website is equally vulnerable to some form of injection attack. An attacker will try various means to alter the structure (and hence the behavior) of dynamically generated queries by inserting SQL keywords, special characters and alpha-numeric values. Therefore, except for the SQL keywords and

special characters, identifier tokens in the query such as table and column names along with the parameter values are irrelevant as far as the structural form of the query is concerned. Hide the table and column names from the two parameterized queries

3. Proposed System

A method is proposed and an analysis is done for preventing SQL injection attacks by combining well-known encryption techniques Blowfish (Symmetric key encryption) ,RSA (Asymmetric key encryption) and RC4,RC5 algorithms at different levels of the proposed model. In the proposed scheme, access to the database will be provided by the server only to the authenticated users. If a new user wants to access the database he will have to register himself with the server. During the registration process, it requires every new user to provide username and password. Along with its regular process of checking the availability of user name, on successful completion, the server generates user key which is ASCII value of the password and with the key username and password is encrypted and stored.The corresponding query is then encrypted by using RSA, RC5 and RC4 Algorithms. The performance is evaluated by means of security, time, space complexity.The URL is also encrypted using AES Algorithm.

The information of the registered users is stored in the server database, in the user table with other user details like encrypted user name, password and the key. Once the user is registered he can access the database by requesting a log-in to the server and the server responds with the access control once the request is validated. There are two stages in the process

- Access Request Process
- Access Grant Process

3.1 Access Request Process

Access is provided to the database by the server after verifying users authentication. Every new user is required to register with the database first. Existing users can directly access the database by providing their username and password. Following are the steps:
Registration

- A new user has to register himself/herself with the database server.
- For registration a user has to provide valid username and password of minimum 4 characters.
- ASCII value of the password provided by the user is stored in the user authentication table, which will act as a key for blowfish encryption and decryption.

Login

- User enters username and password.
- Password is converted into ASCII.
- Username and password is encrypted using the ASCII value of the password as a key, by blowfish encryption,.
- An SQL query is the generated for the user request with username, password and encrypted username and password.
- SQL query is then encrypted with RSA,RC4 in different systems technique encryption using a public key for further security.
- Encrypted query is then sent over to the server. Since a two level encryption technique is used in this scheme the level of security is very high. The server end has to decrypt and verify users authentication and provide further access to the database.

3.2 Access Grant Process

Server provides access to the requested user only after the verification is complete.

Following are the steps included

- Server receives encrypted data
- Data is decrypted to get SQL query using a private server key..
- An SQL query is the generated for the user request with username, password and encrypted username and password
- SQL query is then encrypted with RSA,RC4 in different systems technique encryption using a public key for further security.
- Encrypted query is then sent over to the server. Since a two level encryption technique is used in this scheme the level of security is very high. The server end has to decrypt and verify users authentication and provide further access to the database.

From the decrypted SQL query the username and password are used to fetch the key (hexadecimal value of password) stored in the user authentication table.

- Key is used to decrypt the encrypted username and password in the query.
- If the decrypted user name and password matches any of the authorized users in the user table stored in the server then access is granted to the user or else the query is rejected

3.3 URL Encryption

The username and password in the URL is encrypted using AES Algorithm

- Username and password is entered
- The corresponding URL is encrypted using AES. URL Encryption

4. Algorithms

At the time of creation of the user account, a user key is generated for every user where the user name and password at the time of login is encrypted by Blowfish encryption and RSA technique at different levels of the total encryption process. The access is provided by the server after confirming the users authenticity. On server side the encrypted data will be decrypted using the user key. The decrypted data will be checked and if the user is genuine, further access will be granted to the database. The RSA encryption will work as a protective cover for the SQL query generated by the user at the clients end.

RSA Algorithm

Key Generation

• Choose two distinct prime numbers p and q.
• Find n such that n = pq.n will be used as the modulus for both the public and private keys.
• Find the totient of n, (n) • (n)=(p-1)(q-1)
Choose an e such that 1 ¡ e ¡ (n), and such that e and (n) share no divisors other than 1 (e and (n) are relatively prime).e is kept as the public key exponent.
• Determine d (using modular arithmetic) which satisfies the congruence relation de 1 (mod (n)).

Encryption

• Person A transmits his/her public key (modulus n and exponent e) to Person B, keeping his/her private key secret.
• When Person B wishes to send the message ″M″ to Person A, he first converts M to an integer such that 0 ¡ m ¡ n by using agreed upon reversible protocol known as a padding scheme.
• Person B computes, with Person A's public key information, the ciphertext c corresponding to c me (mod n).
• Person B now sends message ″M″ in ciphertext, or c, to Person A. Decryption
– Person A recovers m from c by using his/her private key exponent, d, by the computation m cd (mod n). – Given m, Person A can recover the original message ″M″ by reversing the padding scheme.

Blow Fish Algorithm

Generating the Subkeys

The subkeys are calculated using the Blowfish algorithm:

- Initialize first the P-array and then the four S-boxes, in order, with a fixed string. This string consists of the hexadecimal digits of pi (less the initial 3): P1 = 0x243f6a88, P2 = 0x85a308d3, P3 = 0x13198a2e, P4 = 0x03707344, etc.
- XOR P1 with the first 32 bits of the key, XOR P2 with the second 32-bits of the key, and so on for all bits of the key (possibly up to P14). Repeatedly cycle through the key bits until the entire Parray has been XORed with key bits. (For every short key, there is at least one equivalent longer key; for example, if A is a 64-bit key, then AA, AAA, etc., are equivalent keys.)
- Encrypt the all-zero string with the Blowfish algorithm, using the subkeys described in steps (1) and (2). – Replace P1 and P2 with the output of step (3).
- Encrypt the output of step (3) using the Blowfish algorithm with the modified subkeys. – P3 and P4 with the output of step (5). –
- the process, replacing all entries of the P array, and then all four S-boxes in order, with the output of the continuously changing Blowfish algorithm

RC4 Algorihtm

RC4 is a stream cipher, symmetric key algorithm. The same algorithm is used for both encryption and decryption as the data stream is simply XORed with the generated key sequence. The key stream is completely independent of the plaintext used. It uses a variable length key from 1 to 256 bit to initialize a 256-bit state table. The state table is used for subsequent generation of pseudo-random bits and then to generate a pseudo-random stream which is XORed with the plaintext to give the ciphertext. The algorithm can be broken into two stages: initialization, and operation. In the initialization stage the 256-bit state table, S is populated, using the key, K as a seed. Once the state table is setup, it continues to be modified in a regular pattern as data is encrypted. The steps for RC4 encryption algorithm is as follows:

• o be encrypted and the selected key.
• Create two string arrays.
• Initiate one array with numbers from 0 to 255.
• Fill the other array with the selected key.
• Randomize the first array depending on the array of the key.
• Randomize the first array within itself to generate the final key stream.
• XOR the final key stream with the data to be encrypted to give cipher text

RC5 Algorithm

RC5 should be a symmetric block cipher The same secret cryptographic key is used for encryption and for

decryption.The plaintext and ciphertext are fixed length bit sequences blocks.RC5 should be suitable for hardware or software This means that RC5 should use only computational primitive operations commonly found on typical microprocessors. RC5 should be fast This more or less implies that RC5 be word oriented the basic computational operations should be operators that work on full words of data at a time.RC5 should be adaptable to processors of different word lengths For example as bit processors become available it should be possible for RC5 to exploit their longer word length. Therefore the number w of bits in a word is a parameter of RC5 different choices of this parameter result in different RC5 algorithms.

RC5 Uses 3 primitive operations

- Twos complement addition of words.
- Bitwise Exclusive OR of words
- A left rotation of words

AES Algorithm

AES is based on a design principle known as a substitution-permutation network, combination of both substitution and permutation, and is fast in both software and hardware. Unlike its predecessor DES, AES does not use a Feistel network. AES is a variant of Rijndael which has a fixed block size of 128 bits, and a key size of 128, 192, or 256 bits. By contrast, the Rijndael specification per se is specified with block and key sizes that may be any multiple of 32 bits, both with a minimum of 128 and a maximum of 256 bits.AES operates on a 44 column-major order matrix of bytes, termed the state, although some versions of Rijndael have a larger block size and have additional columns in the state. Most AES calculations are done in a special finite field...

Module Implementation

The main modules are

- Login
- Encryption
- Decryption

Login In login module the user can enter into the web application with a username and password**.**new users can enter into the system with the help of new member signup

Encryption The username and password is encrypted using the Blow fish algorithm**.**This is stored in the database**.**The generated query for the corresponding username and password is encrypted with the help of RSA,RC4 Algorithms**.**Also the URL is encrypted using Blowfish Algorithm

Decryption The query is first decrypted using RSA,RC4 algorithm**.**Then we get the encrypted user name and password**.**This is checked with the database whether it exists in the database**.**

5. Results

In the system the username and password is not stored as such.A dual security mechanism is provided.The username and password is first encrypted using the Blowfish Algorithm.The query for corresponding is encrypted using RSA and RC4 algorithm.The encryption is done in jsp page.Decryption is done in database.The system provides a dual security where there is no chance of SQL Injection attacks.

Performance Evaluation

The performance of the system is evaluated by the encryption time and decryption time of algorithms used.Its noted that the RSA algorithm takes more time when compared to RC4, RC5 algorithm.

Performance evaluation for Blowfish Algorithm

Username	Password	Encryption time	Decryption time
Chinchu	chinchu	428ns	428ns
Ajiths	Ajiths	418ns	418ns

Performance evaluation for RC4 Algorithm

Query	Encryption time	Decryption time
Encrypted query for username and password =chinchu	528ns	528ns
Encrypted query for username and password = ajiths	518ns	518ns

Performance Evaluation for RC5 Algorithm

Query	Encryption time	Decryption time
Encrypted query for username and password =chinchu	428ns	528ns
Encrypted query for username and password = ajiths	418ns	518ns

Performance Evaluation for RSA Algorithm

Query	Encryption time	Decryption time
Encrypted query for username and password =chinchu	20528ns	20528ns
Encrypted query for username and password = ajiths	20518ns	20518ns

Conclusion

Sql injection is one of the top security threats.The injection can alter, add, delete information in the database or it can delete the entire database.There are many types of Sql injection found.Of those the topmost is the tautology attacks ,end of line attacks.All these attacks can be prevented by the proper sanitization of username and password ie the authentication fields. Sql injection prevention using RSA algorithm is a light weight approach to convert sql queries into encrypted form and apply Blowfish encryption to store these queries.The ASCII value is used as the private and only the key values corresponding to the username and passoword is stored in the database.Only if the generated key value matches with the one stored in database one will be authenticated. It can prevent all types of sql injection attacks. It can also be applied in any platform and in any database.The query is encrypted using both RC4 and RSA algorithms.The evaluation states that RSA takes more time than RC4 Algorithm

Future Work

In order to efficiently prevent SQL injection attacks and Direct db attacks secret key sharing can be used in which a proxy server will be created and the sql query will be splitted in the proxy server. This will be distributed in the form of shares and this will be stored in the database.The sharing key is unknown. So that the attacker cannot know where exactly the query is.So he cannot attack the database directly thus preventing Directdb attacks.

References

Mahima Srivastava, Algorithm to prevent back end database against SQL Injection attacks, IEEE, 2014.

William G.J Halfond, Jeremy Viegas and Alessandro Orso (2008), Specification Based approach for SQL injection detection, ACM.

Meijung (2009), An approach for SQL injection detection, IEEE.

R.Ezumalai (2009), An Combinatorial approach for SQL injection detection, IEEE.

Meijung (2012), An Novel approach for SQL injection detection using hashing and encryption, IJSCIT.

Meijung (2013), An approach for SQL injection detection using query transformation and hashing, IEEE.

William G.J Halfond, Jeremy Viegas and Alessandro Orso (2005), Amnesia :SQL injection prevention, IEEE.

S.Fouzal Hidhayal,Angelina Geethal (2012), Intrusion Detection against SQL Injection Attacks Using a Reverse Password , AIRJCCET

William G.J Halfond, Jeremy Viegas and Alessandro Orso (2008), WASP:Protecting Web Applications Using positive Tainting and Syntax-Aware Evaluation, ACM

William G.J Halfond, Jeremy Viegas and Alessandro Orso (2013), logical trees: An essential method of parsing SQL statements, IEEE.