International Research Journal of Computer Science (IRJCS)
Issue 12, Volume 3 (December 2016)
ISSN: 2393-9842
www.irjcs.com

SQL INJECTION PREVENTION TECHNIQUEUE

Shaji. N.Raj
Assistant professor ,SAS SNDP Yogam college Konni,
Kerala Research Scholar

Abstract - Web application security has important role in development of website. Application level vulnerabilities are main issues of web security. Security has become an important issue in Web Services application, which has been widely concerned.SQL injection is the most common attack used to collect data from database. This paper introduces web application level vulnerabilities named SQL injection and prevention by key value encoding. In key value encoding method create a key for each data and store this key along with original data.

Keywords-Threat, web applications; vulnerability; SQL injection; security protection

I. INTRODUCTION

There are different types of threat are affected by a website. Today attack is related to web based attack. Web application consortium classifies the attack in different types. To ensure the security of web applications, the most important thing is to thoroughly study the vulnerabilities of them. Because only after all the attack methods and vulnerabilities are well understood, we can take effective actions to prevent them. Therefore, exploring the characteristic, origination, and principle of attacks on web applications are especially important. SQL injection is a technique where malicious users can inject SQL commands into an SQL statement, via web page input. Injected SQL commands can alter SQL statement and compromise the security of a web application. For this SQL injection example use a typical login page where users enter their credentials to login to a website or private portal.

When a user submits a username and password, the web application uses these credentials in an SQL query. This SQL query is sent to the backend database to be executed and depending on the result of the query, the website determines if the credentials are correct or not, thus allowing the user to access the portal or denying access.

For example if the username is "abc" and the password is "123", the web application sends an SQL query similar to the one below to the database to verify the credentials:

SELECT * FROM Users WHERE name = 'abc' AND password = '123'
Suppose a malicious user enters something like '1' OR '1' = '1 instead of the username and anything else as password.

In this case the SQL query will look like the below:
SELECT * FROM Users WHERE name = '1' OR '1' = '1 AND password = 'sss'

The above SQL statement will always return a true because:
Since the database returned a true value, the malicious user was able to trick the web application and manages to gain access to a logged in session without the need to guess the credentials. This Type SQL injection vulnerability can also be used to retrieve further data from the database, such as table names and their content.

RELATED WORK

1.AN AUTHENTICATION MECHANISM TO PREVENT SQL INJECTION ATTACK (INDRANI BALASUNDRAM,E.RAMRAJ 2011)

[4] Indrani Balasundran and E.Ramaraj present an authentication mechanism to prevent SQLIA using advanced encryption standard(AES).Here encrypted username and password is used to improve the authentication process..This model have three phases such as Registration phase, Login phase and Verification phase.In registration phase user must select a username and password .But in server it store in three filed include an addition filed named user secrete key.In login phase user name and password is encrypted by using AES by applying user secret key. then the query will send to the server,

R=select * from table where username='username'
AND password='password'
AND encrypted username='encrypt username'
AND encrypted password='encrypt pass'
In verification phase server verify the username and password. If user name and password matches the username and password can be decrypted from query by using this key.

2.EFFICIENT METHOD FOR PREVENTING SQL INJECTION ATTACK ON WEB APPLICATION USING ENCRYPTION AND TOKENIZATION (2014)

S.Anjugam and A.Murugan presented a light weight method to prevent sql injection attack by using tokenization technique to convert sql queries into token and encrypted using AES algorithm This include a. Encryption and decryption b) Query tokenization c) Comparison of dynamic table
The tokenization is applied on the input query by detecting single quotes double quotes etc.These tokens are stored in a dynamic table at client side. The table name, field name and data are encrypted. Then both are send to the server side. At the server side input query is decrypted and turn into tokens and stored in a dynamic table. If both the dynamic table are same, it prevents from SQL injection.

3.RUNTIME MONITORING TECHNIQUE TO HANDLE TAUTOLOGY BASED SQL INJECTION ATTACK (2012)

[6] Ramya dharma aand Sajjan G Shiva propose a frame work called Runtime monitoring frame work which perform runtime monitoring of web application during its post deployment to detect and prevent SQLIA.The frame work uses two pre deployment testing techniques such as basis data flow to identify a minimal set of all valid execution path of application. Runtime monitors are then developed and integrated to perform runtime monitoring of applications during its post deployment. This system first uses a software repository which consists of a collection of documents related to requirements, security specification and source code etc.to find the critical variable. Critical variables are those which interact with external world by accepting user input. A combination of basis path and dataflow testing technique is then used to find all valid execution paths that the critical variables can take during their life time in the application

4. EVOLUTION OF WEB SECURITY MECHANISMS USING VULNERABILITY & ATTACK INJECTION (2014)

Jose Fonseca,Macro Vieria and Henrique Maderia propose a methodology and a prototype tool to evaluate we application security mechanism. It based on the idea that injecting realistic vulnerabilities in a web application and attacking them automatically. This project supported by the project "ICIS Intelligent computing in the Internet of Services". The VAIT tool allows the automation of entire process. This method focuses on SQLIA and cross site scripting. The attack injection uses two external probes one for HTTP communication and other for database communication. These probes monitor the HTTP and SQL data exchanged. The attack injection mechanism is aware of important inner working of application while running. This system includes, Preparation stage,Vulnerability injection stage,Attack stage

5.A NOVEL METHOD FOR SQL INJECTION ATTACK DETECTION BASED ON REMOVING SQL QUERY ATTRIBUTE VALUES (2012)

Inyong LEE [7],Soonki Jeong ,Sangsoo Yeo and Jongsub Moon proposed a method removes the value of an SQL query attribute of WebPages when parameter are submitted and then compare it with predefined one. This method used static and dynamic analysis. This method detect SQL injection attack by comparing static SQL queries with dynamically generated queries after removing the attribute values.

6.DEFEATING SQL INJECTION USING DATA CLEANSING ALGORITHM (2015)

A code injection detection tool is used in this method. This algorithm has three phases,
Query detector
Script detector
DC Algorithm
All http request coming through the client side will transfer to CIDT,when any malicious content is found it execution will be blocked. The user input will be taken and checked for any misbehaved query in the input. The script detector take input from query detector and scan it for URL scripts. The DC algorithm will attempt to clean the request. This system is proposed by Digambar Patil.

PROPOSED MODEL

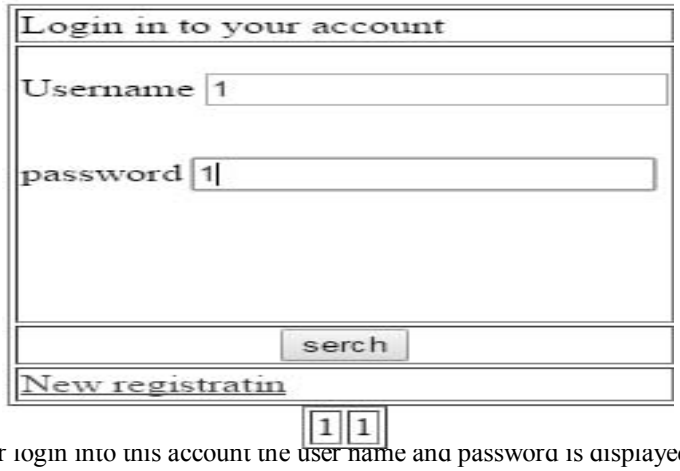
An SQL injection attack occurs when an unlawful user access to database by using SQL query. Code injection is done by injecting malicious code in the database.SQL statement is string and they are changed dynamically in web application based on user input.For example a login name form an intruder can simply type some code in the input form he can access the database very easily. Normally username and password are stored in a table and if a user try to login the input value is compared with the table value. Normal table structure and code is given below.

If a user try to login username and password is checked with existing value.

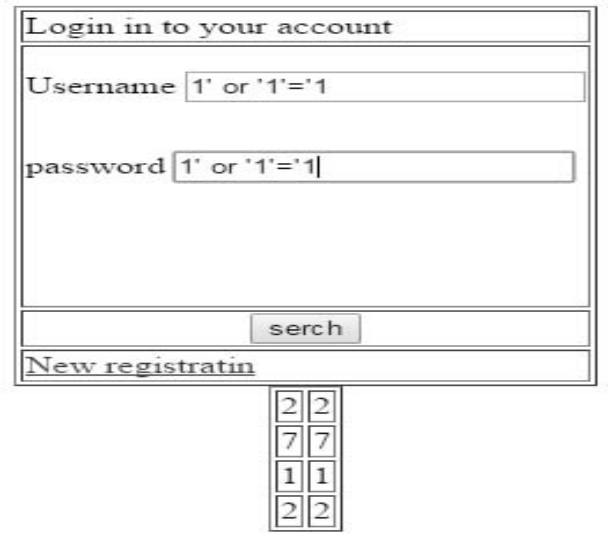
```
if(isset($_POST['serch']))
{$username=$_POST['use'];
$password=$_POST['pas'];
```



```
echo "<table border=1 align='center'>";
$se="select * from student where (USERNAME='$username' and PASSWORD='$password')";
$sd=mysql_query($se);
while($r=mysql_fetch_array($sd))
{          echo "<tr>";       echo "<td>".$r['USERNAME']."</td>";
echo "<td>".$r['PASSWORD']."<br>";
echo "</tr>";
}
```



In above code when a user login into this account the user name and password is displayed. If a user give user name as 1' or'1'=1' and password as 1'or'1'='1, then all the data from database will be displayed.



In proposed system a new approach is used to store the data in a database. Our solution consists of four steps .
Step 1. An additional key is generated.
Step2. This key is stored in an additional column
Step3 .When login into the form first check for username and password.
Step3.If user name and password is correct then check with key value.
Step4. If key value is correct only login into the form

USERNAME	PASSWORD	KEY
2	2	22
7	7	77
1	1	11

An additional column is used to store the data in a coded form. Username and password field are merged and stored in the third column named key.PHP code used to store additional field are given below.

```
mysql_select_db("sas");
if(isset($_POST['sub']))
{$username=$_POST['use'];
$password=$_POST['pas'];
$key=$username.$password;
$sq="insert into secure values('$username','$password','$key')";
mysql_query($sq);
```

When a user login into the form first check for user name and password then check for key.If key match is found correct we can only login in to the page. The table structure is given below.

The PHP code is given below,

```
mysql_select_db("sas");
if(isset($_POST['serch']))
{$username=$_POST['use'];
$password=$_POST['pas'];
$key=$username.$password;
echo "<table border=1 align='center'>";
$se="select * from secure where (USERNAME='$username' and PASSWORD='$password')";
$sd=mysql_query($se);
while($r=mysql_fetch_arry$sd))
{                    if($r['KEY']==$key)
{
echo "<tr>"
echo "<td>".$r['USERNAME']."</td>";
echo "<td>".$r['PASSWORD']."<br>";
echo "</tr>";}}
```

When an request come to the webserver the server will send the request to SQL injection prevention algorithem,which check for the key value .A user only know about the username and password ,internally the key generation and checking has been done. Thus an intruder cannot access the data in the database.

CONCLUSION

In this paper we proposed SQL injection prevention by key coding method. This model creates a coded key to store data. The key value is generated as the combination of all table value. These models filters the input data and prevent SQL injection attack. This model validates the input form information but also detect the key value matching. The server side verifies the user privilege.

REFERENCES

[[1].http://resources.infosecinstitute.com/sql-injections-introduction/1
[2] http://www.w3schools.com/sql/sql_injection.asp
[3].http://projects.webappsec.org/w/page/13246989/Web%20Application%20Security%20Statistics#Summary
[4]Indrani Balasundram.E.Ramraj ,2011 International Journel of computer application(0975-8887) Volume 19 NO 1 April 2011
[5].S.Anjugam and A.Murugan,2014 International Journal of advanced Research in computer science and software engineering(ISSN 2277-128x) Volume 4 issue 4 April 2014

