Infys Systems
you dream it, we build it!!!

www.infys.net
info@infys.net
+91-731-2570516

Understanding Sql Injection

-Hardik Shah

Infys Systems
you dream it, we build it!!!
www.infys.net
info@infys.net
+91-731-2570516

Understanding SQL Injection

Introduction:-

SQL injection is a technique used by a malicious user to gain illegal access on the remote machines through the web applications vulnerability. The basic idea behind this technique is to run the sql query which was not intended to run by a programmer. This technique is heavily relay on the logical operations like AND, OR.UNION etc. if this technique is used properly a malicious user can get complete access on a web server. If the application is creating SQL strings naively on the fly (dynamic queries) and then running them, it can create some real surprises as we see later on.

How it performed:-

This vulnerability occurs due to lack of proper validation of user entered data in web applications. It may be possible that the programmer is a newcomer and has lack of understanding of such kind of attacks. But in many cases I have seen most of the time programmers are too lazy to consider and apply proper security checks. Most of the programmer believes that client or end user will always give correct input to the application. They even check for some minor validations like empty string or null values etc but they never think of the fact that a user can insert a specially crafted query which reveals all the important information of your machines. With the outsourcing boom many companies started and they have less experienced programmer so such kind of attacks heavily exists in today's web applications. If we take a simple example of a login page, then generally programmer's uses this pseudo code (assume that the database server is MS-Sql server):-

```
query="select * from userinfo where username='  "&strUser&"'   and
password='"&strPass&"'  "
 strCheck=GetQueryResult(query)
  if strCheck="" then
    bool loginflg=False
  else
    bool loginflg=true
  end if
```

This query works fine without any problems if user enters correct characters. But suppose a malicious user enter following:-

username=test
password=' or 1=1—

Now the above query will become:-

query="select * from userinfo where username='test' and password='' or 1=1--'"

-- Symbol denotes the comment in sql server. Hence in the MS-Sql server everything after the -- is ignored.

So this query is actually becomes something like this:-

select * from userinfo where username='test' and password='' or 1=1

We can break this query in two portions like bellow:-
p=>username='test' and password=''
q=>1=1
So we can write it as pVq

Now from the Boolean algebra we know that in V(OR) operation the result will be true if any of the value is true. As here the value of q is always true as 1 is always equal to 1, hence the value of this entire expression(pVq) is always returned as TRUE.
So the query becomes (I replaced with p, q for ease of reading)

query="select * from userinfo where pVq"

Now as discussed above the pVq is always true hence the query will select all the records in the current table. But generally programmer takes one record for login hence the username becomes the username of the first record.

Consider following table:-
No. username password
1 test test

2 temp temp

In this table if above query is executed then username becomes test.
Hence on executing the above malformed query a malicious user can bypass authentication mechanism of the web application.

But this is only one thing among several endless options which an intruder can use. By using specially crafted query a user can retrieve the entire database schema of your application or he can upload/download any file or he can get any other info such as credit card numbers stored in the database, can delete the user, add new user etc.

Different types of attacks:-

1) SELECT-UNION:-

Union operation permits combining two results. So by using this option a user can retrieve any sensitive information from the database. In case of the above query we mentioned suppose a user enters following in the userid and password field:-

username=test
password=' or 1=1 union select top 1 TABLE_NAME from INFORMATION_SCHEMA.TABLES--

On execution of this query the database engine will give an error something like this:-

Microsoft OLE DB Provider for ODBC Drivers error '80040e07' [Microsoft][ODBC SQL Server Driver][SQL Server]Syntax error converting the nvarchar value 'userinfo' to a column of data type int. /testpage.aspx, line 25

Now from the above mentioned error it is clear that the table name is userinfo.
After determining the table name user need to find the column name in the table. So he can enter following values:-
username=test
password=' or 1=1 union select top 1 COLUMN_NAME from INFORMATION_SCHEMA.COLUMNS where TABLE_NAME='userinfo' –

Output

Microsoft OLE DB Provider for ODBC Drivers error '80040e07'

[Microsoft][ODBC SQL Server Driver][SQL Server]Syntax error converting the nvarchar
value 'login_id' to a column of data type int.

/testpage.aspx, line 25

The above error message shows that the first field or column in userinfo is login_id. To get the next column name will type
username=test
password=' or 1=1 union select top 1 COLUMN_NAME from INFORMATION_SCHEMA.COLUMNS where TABLE_NAME='userinfo' where COLUMN_NAME not in('login_id' --)

Output:

Microsoft OLE DB Provider for ODBC Drivers error '80040e07'

[Microsoft][ODBC SQL Server Driver][SQL Server]Syntax error converting the nvarchar
value 'login_name' to a column of data type int.

/testpage.aspx, line 25

so by using this a user can gain the information about the tables,username,passwords etc.

2)SELECT-INSERT:-
With the insert keyword a user can easily add new records in the database.
Given that we know the partial structure of the members table, we can try adding a new record to the table: if this works, we'll simply be able to login directly with our newly-inserted credentials. Look at the following query:-

```
SELECT email, passwd, login_id, full_name
  FROM members
```

```
WHERE email = 'test@test.com';
    INSERT INTO members ('email','passwd','login_id','full_name')
    VALUES ('test@test.com','hello','test','test');--';
```

based on the implementation and database permissions this query may work and one can login by using his user name "test" and password "test". This might failed as suppose there is a different table which contains access right to the user and other stuff or may be the web application user doesn't have insert permission on user table.

3) Select Update:-

As discussed above some times the select insert may fail depending on various conditions. In that case using the forgot password button seems a nice way to getting in to the system.

```
SELECT email, passwd, login_id, full_name
  FROM members
WHERE email = 'test@test.com';
    UPDATE members
    SET email = 'malicious_user_mail@mail.com'
    WHERE email = 'test@test.com';
```

This are the genral techniques used. but based on the combination of the various sql keywords say "LIKE","CREATE","DROP","WHERE" etc one can perform various different kind of attacks.

Built In Stored Procedure:-

Another technique in case of sqlserver is of using sql server's stored procedure. A default installation of sql server contains many stored procedure which a malicious user can easily misuse. Some of them are:-

xp_cmdshell

Microsoft's SQLServer supports a stored procedure xp_cmdshell that permits what amounts to arbitrary command execution, and if this is permitted to the web user, complete compromise of the web server is possible.

if xp_cmdshell is enabled then a malicious user can run any arbitery command on the web server . Although Access to xp_cmdshell is usually limited to administrative accounts, but it's possible to grant it to lesser users. not to mention many sql server installation runs with default user sa and blank password.

How to save web applications from sql injection attacks:-

You can see this attacks works on many sites easily. There are many programmers who never validate the data properly. If you are a web application developer thenyoumust need to secure your application from such attacks.following are the proposed solutions by which you can avoid such attacks;-

1) Data santinization:-we need to remove any unwanted characters say ' , " , ; or -- from the user input. Allowing this character may allow sql injection attacks on your website. But some times you need to allow certain special character say ' like name can be O'Reilly.

2) Limit database permissions:-we also need to make sure that we give only necessary permission to the user. Allowing unrestricted access to the user may cause trouble as we discussed above a malicious user can use built in stored procedure to perform the attacks.

3) Use stored procedure:-if possible, use properly formatted stored procedure instead of using dynamic queries in your web applications. It will reduce the chance of such attacks.

4) Use quote function: - we can use built in functions like magic_quote in case of PHP to properly format the user input. This will prevent such attacks.

-Hardik Shah